Computer Science • 23(1) 2022 https://doi.org/10.7494/csci.2022.23.1.4570

Filip Malawski

IMMERSIVE FEEDBACK IN FENCING TRAINING USING MIXED REALITY

Abstract *During sports training, providing athletes with real-time feedback that is based on the automatic analysis of motion is both useful and challenging. In this work, a novel system that is based on mixed reality is proposed and verified. The system allows for immersive and real-time visual feedback in fencing training. Novel methods have been introduced for 3D blade tracking from a single RGB camera, creating weapon-action models by recording the actions of a coach and evaluating the trainee's performance against these models. Augmented reality glasses with see-through displays are employed, and a method for coordinate mapping between the virtual and real environments is proposed; this will allow for the provision of real-time visual cues and feedback by overlaying virtual trajectories on the real-world view. The system has been verified experimentally in fencing bladework training (with the supervision of a fencing coach). The results indicate that the proposed system allows novice fencers to perform their exercises more precisely.*

Keywords mixed reality, augmented reality, object tracking, real time, immersive

Citation Computer Science 23(1) 2022: 37–62

1. Introduction

Recently, modern technologies have been applied in many sports disciplines in order to measure the performance of athletes and provide them with useful feedback [33]. In team sports player detection, tracking, and identification are investigated [25], along with ball detection [21]; these provide information that is useful (mostly for a coach). In individual sports, the real-time feedback that is provided for an athlete is most beneficial, as it allows him/her to improve their performed actions during training. The timing and body angle consistency of a gymnast during a pommel horse routine were evaluated with the help of Kinect, with the results displayed on a screen [31]. Dedicated methods were presented for rapid feedback in three disciplines: rowing (by displaying plots from an accelerometer), table tennis (by detecting and visualizing the ball's impact positions), and the biathlon (by measuring and visualizing the motion of the barrel of one's rifle just before and after a shot) [3]. These approaches provide real-time sports analysis; however, presenting such results on a computer screen is not very convenient for a training athlete. In this paper, we consider providing real-time feedback in fencing by using a novel approach; namely, by employing mixed reality (MR). MR has been successfully applied in aiding surgeons [20] and engineers [30]; therefore, we believe it can be of great value to athletes as well.

In fencing, there are two main elements: footwork (which corresponds to how a fencer moves [19, 24]), and bladework (which corresponds to performing weapon techniques). Bladework analysis is very challenging due to the high precision of one's performed actions as well as the fast motion of one's blade. Weapon action classification was performed by using kinematic data that was acquired by a motion-capture system [26]. The presented methods are able to recognize a number of parry and thrust actions with high efficiency. The kinematic determinants of a weapon's velocity during a thrusting lunge have been identified in the work of [6]. The authors of [7] distinguished proper and improper weapon action executions by applying a neural network to inertial signals that were segmented with dynamic time warping (DTW).

None of the studies that discuss the analysis of bladework in fencing have considered real-time applications. The goal of this work was to develop a system that would allow us to 1) analyze weapon training in real time, 2) evaluate action performance, and 3) provide real-time immersive visual cues and feedback by employing MR. In particular, MR is used to enhance a fencer's view with virtually generated weapon trajectories, which aid in properly performed weapon actions during his/her training. Previously, 2D blade tracking and bladework action analysis and evaluation was considered in [23]. In this work, we propose a 3D tracking method and introduce real-time MR-based visual feedback, including real-virtual coordinate mapping and calibration methods. To the best to our knowledge, no similar approach has been presented in the literature thus far.

2. Background

The discussion of the fencing background that is provided in this section is based on the works of Prof. Czajkowski [12] as well as consultations with the fencers who cooperated in the development of the proposed system [2].

2.1. Weapon training in fencing

There are two types of weapons used in fencing – thrusting, and cutting. With the former, only thrusts score points, while with the latter, both cuts and thrusts can score points. Thrusting weapons require more precision, as even slight differences in the positioning or rotation of a blade can result in a weapon action being successful or not. For this reason, automatic training support is most beneficial for thrusting weapons; therefore, this type of weapon is considered in this work. In sports fencing, these include foils and epees.

There are several stages of training weapon actions. At first, novice fencers simply try to repeat motions that are presented by a coach. Next, an action is practiced with a partner, who provides interaction (for instance, attack-parry); then, sequences of actions are performed in which both participants are practicing. Next, the exercises are performed with varying timings and distances; finally, these actions are practiced during sparring. The goal of the system that is presented in this work is to mainly provide support for the initial stages of weapon action training, as the process of reaching at least a medium level in weapon handling is very time-consuming and toilsome for novice fencers.

Trajectories are an intuitive manner of understanding how a weapon action should be performed. When novice fencers observe a weapon motion that is presented by a coach, they try to remember the trajectory of the tip of the blade. During their training, they try to repeat this; however, without constant supervision from a coach (who usually needs to share his/her attention among multiple students), they lack the necessary feedback for correcting their performances of the action. The goal of the proposed system is to provide these students with real-time trajectory-based feedback and enable efficient training in the absence of the coach.

2.2. Object tracking

Classical methods for tracking objects in videos include the detection of spatio-temporal interest points (e.g., SIFT [22] and SURF [4]), histograms of oriented gradients (HOG) [13], or describing a motion with an optical flow [5]. Recently, deep-learning approaches have become popular [9]. In sports, specific problems can be addressed, such as player tracking in teams sports [29] or ball tracking (e.g., in volleyball [8]). However, tracking a weapon in fencing is significantly different. Such a blade is thin, it moves very fast, and it is made of steel (which is reflective); these factors make it very difficult to track. Moreover, the 3D tracking of position and rotation are needed for employing mixed reality and providing proper visual feedback for fencers.

Depth estimation is typically achieved with depth sensors [32]; however, our initial experiments with the Kinect depth sensor showed that it is not suitable for tracking a thin blade. Methods for 3D tracking from a single RGB camera have been proposed in the literature for body-pose [27] and hand-pose estimation [28]. Even though these are based on body and hand models, they are not applicable for fast-moving blades. Therefore, we propose a novel method in this work that employs active markers to track the 3D positions and rotations of a blade from a single RGB camera.

2.3. Augmented and mixed reality

Augmented reality (AR) enriches the perception of the real world with virtually added information. A popular approach for employing AR is to display virtual objects on the camera feed from the real world; e.g., on a smartphone [35]. A significantly more immersive experience is provided when using AR glasses with see-through displays; these allow for the overlaying of virtual objects not in a camera feed but directly in the user's field of view. AR glasses can be employed to display 3D objects that fit spatially into the real world, thereby creating a mixed reality. The distinguishing feature of MR is that one can see virtual objects as if they were part of his/her surrounding environment and possibly interact with them in some manner [18]. In this work, AR glasses are employed to display the trajectories of weapon actions in front of a training fencer. Coordinate mapping is performed between the virtual and real worlds; therefore, a trajectory is drawn along the tip of the blade during the weapon's movement, providing an accurate visualization of the motion. Similarly, virtually added trajectories of model weapon actions constitute visual cues on how to perform bladework exercises. A comparison of the trajectories of the performed and model actions allows for evaluation and feedback; therefore, the MR view is employed to aid the fencing training. It is worth noting that, while hand gestures and touch or haptic devices [10,34] are typically used in MR, such interactions are performed with the sports weapon in this case. An important challenge in this work was to perform tracking, analysis, and MR visualization in real time.

3. Methods

3.1. Architecture

Due to the utilization of AR glasses with real-virtual coordinate mapping, it is possible to create the following training routine. First, a coach performs multiple repetitions of an action that are recorded and used to build a model trajectory. Then, the model trajectory is displayed in front of the training fencer with the AR glasses, who follows the displayed trajectory with the tip of the blade. The evaluation of the action is performed based on the differences between the model and the 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. Due to the fast motion of the thin blade and the

low quality of the video stream, LED markers are used (as these can be detected relatively easily).

The architecture of the proposed system is based on employing the following devices: a double-LED marker (placed near the tip of the blade), AR glasses (for a training fencer), and a laptop (which is wirelessly connected to the glasses); see Figure 1. The camera in the AR glasses is used for detecting the marker in real time; this enables the tracking of 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 our experiments with the prototype system that was implemented in this work; however, it would be possible for the system to operate without a laptop. This would require the implementation of the control interface on the glasses themselves or on a mobile device. An additional benefit of using a laptop is that it allows one to preview the tracked trajectories on the screen, therefore providing feedback for a coach as well.

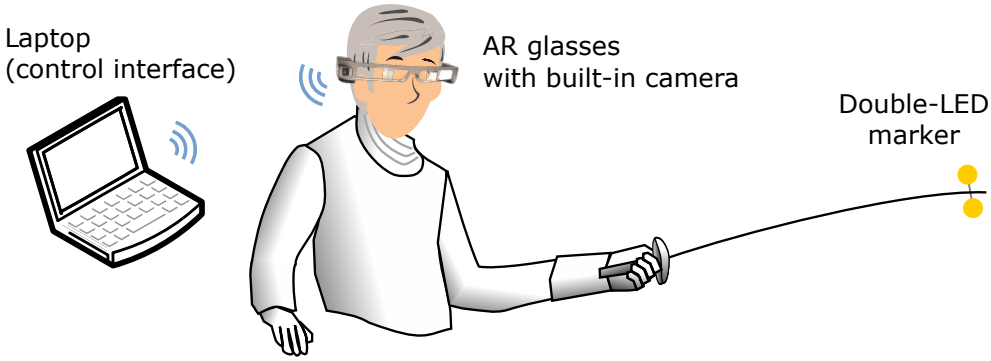

Figure 1. Architecture of proposed system

3.2. Blade tracking

Based on consultations with fencing experts, three parameters describe the motion of a weapon's 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 is most relevant for offensive actions in terms of evading an opponent's blade and hitting an intended target area. The second and third are most relevant for defensive actions, where the proper positioning and rotation of a blade are required so that one's parry actions are effective. In this work, only two of these parameters are tracked during a fencers' training; this is due to the limitation of the employed devices. Although the built-in camera in the AR glasses has a wide-angle lens, its view is not sufficient for reliably tracking the base of a blade – during weapon training, 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 that are provided by the proposed system are still useful for bladework-training support. Also, this limitation

results directly from the capabilities of the used device and not from the proposed method itself. If the camera lens had a wider angle, it would be possible to implement the tracking of an additional marker and the visualization of another trajectory.

The relatively low quality of the video stream that is provided by the built-in camera makes the tracking of the weapon blade difficult. The fast motions result in blurry images, particularly under poor lighting conditions (which are typical for training halls); this is due to the relatively long exposure times that are required by the light-sensitive matrix in order to capture a video frame. For this reason, state--of-the-art object-tracking algorithms are not applicable in this case. Instead, LED markers are mounted on the blade, and the camera is set to low exposure; this results in the captured images (other than the light sources) being dark. 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, the images are less blurry because the exposure times are shorter.

Using a double-LED marker enables depth and rotation tracking. The marker that is mounted at the tip of the blade has two LEDs (approx. 5 cm from each other) that are placed perpendicularly to the blade (see Fig. 2). Based on the relative positions of the LEDs in a captured image, it is possible to estimate both depth and rotation. The marker itself is very simple and low-cost, as it contains two LEDs that are soldered to a CR2032 battery case (see Fig. 2).

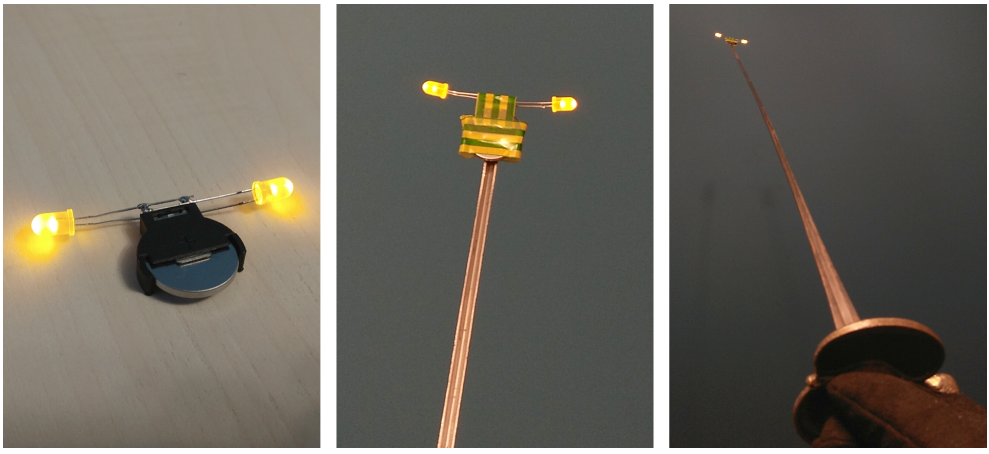

Figure 2. Double-LED marker (left) mounted on tip of blade (middle); entire weapon view (right)

The detection of the LEDs in the captured images is performed in two steps. First, the image is converted to grayscale, and a threshold operation that creates a binary image is applied in which all of the pixels above the threshold are white and all of the others are black (see Fig. 3). The threshold is selected automatically during a short calibration procedure – the user places the marker in the camera's view, and

the lowest and highest thresholds are found for which the number of detected LEDs is correct. The final threshold is set closer to the lower threshold in order to capture the marker during fast motions (when the LEDs are blurry). Alternatively, the threshold can be set manually in a dedicated display mode where all of the pixels that are above the threshold are shown.

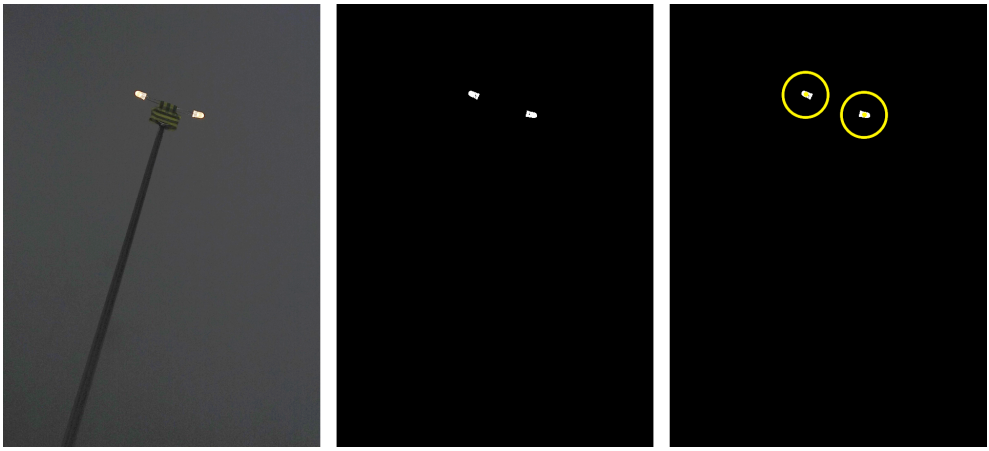

Figure 3. Marker detection: camera is set to low exposition setting (left); thresholding operation is applied (middle); LED positions (keypoints) are detected after clustering (right)

The second step of the LED detection consists of a clustering. Once a binary image is created, clusters of white pixels are found; these correspond to the markers. The algorithm runs through all of the pixels in the image, and it marks each found white pixel as a new cluster and performs a region-growing operation [1]. When more than two clusters are found, only the largest two are considered; the additional small clusters that correspond to separate pixels that are not connected to the main two clusters are then discarded. When fewer than two clusters are found, the frame is discarded. The proposed tracking method is solely based on detection; therefore, any spatio-temporal dependencies between consecutive frames are not considered. This approach is sufficient when no other light sources are present in the camera's field of view (which was the case in the experiments).

The centers of the detected clusters are referred to as keypoints; each corresponds to a single LED (see Fig. 3). 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 is to the camera, the greater the pixel distance is between the keypoints (see Fig. 4). By performing a one-time calibration, it is possible to map the pixel distance between the keypoints to the actual distance between the marker and the camera. In a limited depth range, the relationship is approximately linear. The rotation is given by the angle between the line that connects the two keypoints and the horizontal axis of

the camera (see Fig. 4). Due to the view angle of the camera, this is not exactly the same as the rotation that is measured relative to the ground; however, since both the model learning and training evaluation depend on the rotation that is measured with this method, the perceived rotation is more relevant than the rotation relative to the ground is.

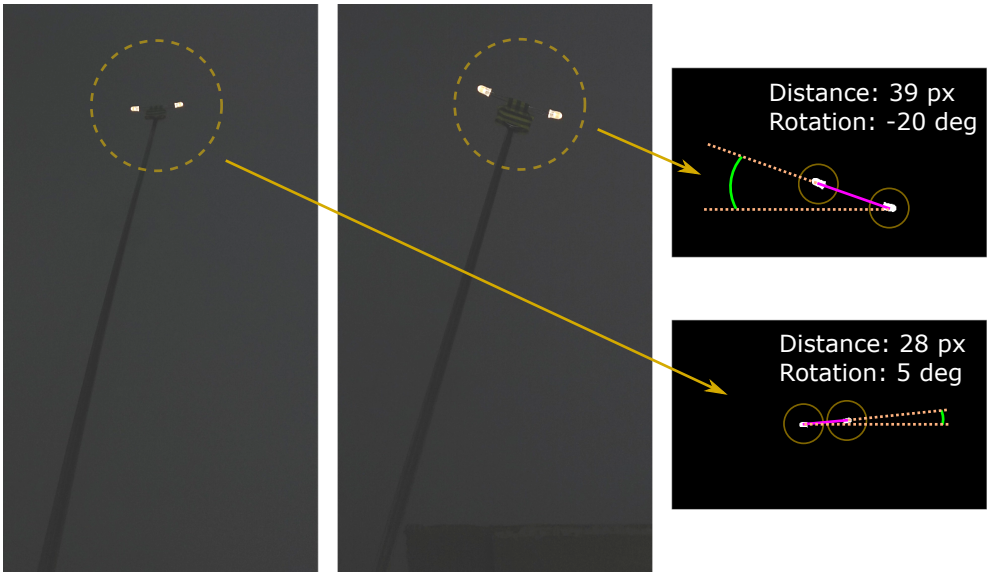

Figure 4. Depth and rotation estimation based on detection of relative positions of two LEDs; pixel distance between LEDs corresponds to depth – smaller distance (left) indicates that tip of blade is further away

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

3.3. Action models

Due to the fast motion of the blade and a camera acquisition rate that is equal to 30 Hz, detecting the double-LED marker provides only a sparse sampling of the blade tip's trajectory (see Fig. 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 visualization and evaluation purposes, dense trajectories with a common length are required; therefore, cubic spline interpolation [14] is applied, which provides densely sampled trajectories with constant arbitrarily chosen numbers of points (see Fig. 5 [right]).

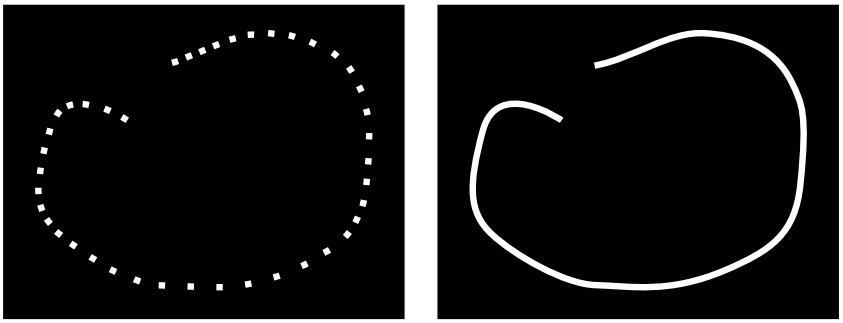

Figure 5. Tracking of tip of blade: detected marker positions (left); interpolated trajectory (right)

The use of interpolated trajectories allows us to calculate a model trajectory for the action and employ a point-by-point comparison with the trajectories that are recorded during the bladework training; however, a common starting point is also required. For this reason, building a model of the action begins with defining the starting point. The fencer simply places the tip of the blade in the desired position and saves the starting point by using the control interface. During both the model-learning and training evaluations, each repetition of an action begins with moving the tip of the blade to the starting point that is displayed in the AR glasses.

In order to facilitate the use of the system, the action repetitions are detected automatically. A finite-state machine (FSM) is employed (see Fig. 6). The initial state is *detection* in which the tip of the blade is detected; however, the trajectories are not recorded. Once the tip of the blade is moved to the displayed starting point, the state changes to *in position* and a timer is set that changes the state to *ready* after one second. Then, the system waits for the blade movement to start; this triggers the transition to the *recording* state. Each state is indicated to the user by changing the color of the virtual starting point marker.

While an action is performed, a simplified trajectory is displayed in the AR glasses by connecting the detected points with straight lines. When the fencer stops moving his/her blade, the interpolated trajectory is displayed instead. The next repetition of the action begins when the tip of the blade is moved again to the starting point. The tip of the blade is considered to start moving when the average displacement of the detected points in the previous 20 frames is above a predefined threshold (which was chosen experimentally), taking into account that the tip of the blade is not perfectly still even when the weapon is not moved voluntarily.

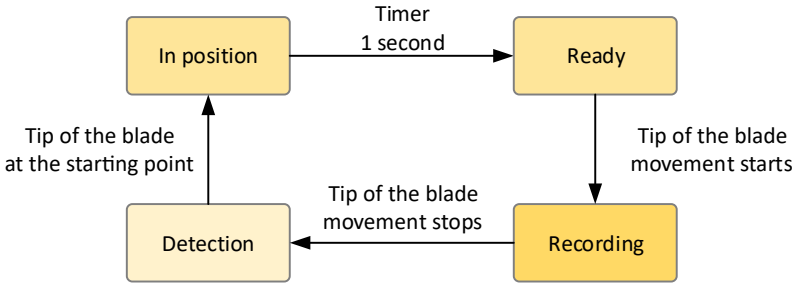

Figure 6. Finite state machine used to automatically detect action repetitions

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

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

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

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

In the training-evaluation mode, the recorded model of action is loaded, and the mean trajectory is displayed; then, the fencer tries to repeat the action. Both the current and model trajectories are visible; therefore, instant visual feedback is provided. Once a repetition of the practiced action is completed, 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 from the model point and the standard deviation for this point (which is also stored in the model). The edge values are at 100% when the distance is lower than the standard deviation and at 0% when the distance is greater than twice the standard deviation. Given that c_i is the *i-th* point in the current trajectory, the a_i accuracy of this point is computed as follows:

$$\forall i \in 1, ..., L, a_i = 100 - ((min(max(\frac{m_i - c_i}{s_i}, 1), 2) - 1) \cdot 100) \tag{3}$$

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 the xy plane were eventually employed. The rotation is evaluated in a similar manner. For the purpose of visualization, the rotation is also color-coded.

3.4. Mixed reality

In the proposed system, visual cues and feedback are provided to training fencers by employing AR glasses. An Epson Moverio BT-300 device was used in this work [16]; it included lightweight AR glasses connected by a wire to a small Android-based processing unit, which is also used for the control (see Fig. 7). The glasses have a see-through display for each eye – a user can see both the surrounding environment and the displayed content. The black color in the generated image is transparent in the glasses; therefore, a seamless mixture of real and virtual views is possible. The glasses operate at a 1280 x 720 resolution, have a refresh rate of 30 Hz, and provide a 23-degree field of view. A five-megapixel camera is built into the glasses (on the right side). The processing unit has a 1.44 GHz quad-core Intel Atom processor, 2 GB of RAM, and 16 GB of internal memory. It is also equipped with a touchpad and several control keys that act as the user interface.

Figure 7. Epson Moverio BT-300 AR device, including control and processing unit (left); AR glasses with see-through displays (right)

The glasses can display different images for each eye, therefore simulating a 3D perception of the generated objects. Among other things, human depth perception is determined on the basis of stereo vision [15]. Since our eyes perceive objects from slightly different points, each eye receives slightly different images; the depth of any perceived object is estimated based on the disparity between the right and left images. This mechanism can be used to generate 3D virtual scenes by displaying an image of a scene that is generated from a shifted viewpoint for each eye [11]. In the proposed system, this concept was implemented with the OpenGL ES [17] by repositioning the virtual camera when generating distinct images for the right and left eyes. While other techniques for introducing depth perception exist, stereo vision is the most effective for relatively close distances (up to 2 m) [15].

The main difficulty of creating a mixed real-virtual view with the AR glasses is to generate virtual objects in their proper positions relative to the real environment.

In order to do this, 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 that is placed at the tip of the weapon's blade; therefore, the system needs to calculate the 3D position of each point in the virtual trajectory based on the marker position that is 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. 8). In order to make this possible, a dedicated calibration procedure is proposed.

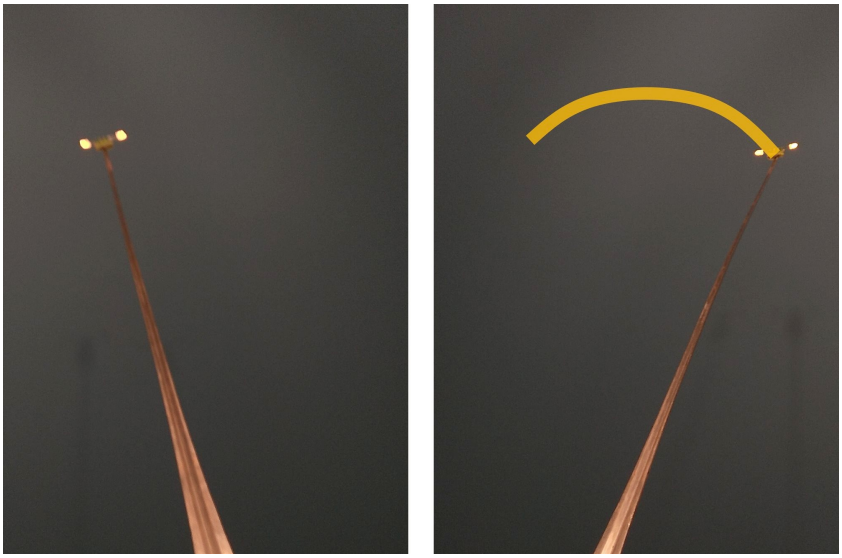

Figure 8. Expected mixed real-virtual view; virtually generated trajectory should match motion of blade tip in real-world view

The 3D position of the marker (corresponding to the tip of the blade) that is seen by the camera is defined by the x and y coordinates of the middle point between the two keypoints that are detected in the image as well as the pixel distance between them (which corresponds to the depth) (see Section 3.2). The virtual object coordinates are given to the OpenGL ES visualization engine in an arbitrarily defined coordinate system. Based on the position of the virtual camera, the engine calculates the projection of the object in the images that are displayed for the right and left eyes. In order to create the mapping between the two coordinate systems, the user manually matches the real and virtual objects at 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 in the AR glasses, and the user is asked to move the tip of the blade to this position and then click a button on the control interface; this saves the calibration data for this point (which includes both virtual and real coordinates).

In order to calculate all of 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 it is for distant objects; therefore, the translation of the coordinates from the real to the virtual world on the xy plane depends on the z distance of the real object (so, the calibration starts with the depth). The virtual marker is displayed in the middle of the screen at a close distance (for the first calibration point) and at a far distance (for the second calibration point). These distances roughly correspond to the typical distance of the tip of the blade before and after an arm extension (which are approx. 130 and 170 cm, respectively). Two calibration points allow the user to find the parameters for the linear equation that describes the relationship between the virtual and real depth coordinates. Given that z_{rc} represents the depth in the real-world coordinate system for the close distance, z_{rf} is the depth in the real-world coordinate system for the far distance, z_{vc} is the depth in the virtual-world coordinate system for the close distance, and z_{vf} is the depth in the virtual-world coordinate system for the far distance, the a_z and b_z parameters are computed from a pair of equations:

$$z_{vc} = a_z \cdot z_{rc} + b_z; \tag{4}$$

$$z_{vf} = a_z \cdot z_{rf} + b_z. \tag{5}$$

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

$$z_v = a_z \cdot z_r + b_z. \tag{6}$$

Next, four calibration points are gathered for the close distance, and another four are gathered for the far distance. The four points are in the middle of each of the left, right, top, and bottom edges of the displayed area (see Fig. 9). Therefore, four sets of parameters for the linear equations are calculated that correspond to the horizontal and vertical coordinates at the close and far distances: (a_{xc}, b_{xc}), (a_{yc}, b_{yc}), (a_{xf}, b_{xf}), and (a_{yf}, b_{yf}). Given that x_{rc}, y_{rc}, x_{rf}, and y_{rf} represent the x and y real-world coordinates at the close and far distances and x_{vc}, y_{vc}, x_{vf}, and y_{vf} are the x and y virtual-world coordinates at the close and far distances, the resulting equations are as follows:

$$x_{vc} = a_{xc} \cdot x_{rc} + b_{xc}; \tag{7}$$

$$y_{vc} = a_{yc} \cdot y_{rc} + b_{yc}; \tag{8}$$

$$x_{vf} = a_{xf} \cdot x_{rf} + b_{xf}; \tag{9}$$

$$y_{vf} = a_{yf} \cdot y_{rf} + b_{yf}. \tag{10}$$

Equations 7, 8, 9, and 10 allow us to compute the x and y virtual coordinates at only the close or far distances (see Fig. 9). General equations for computing the x_v and y_v coordinates at any distance can be expressed as follows:

$$x_v = a_x \cdot x_r + b_x; \tag{11}$$

$$y_v = a_y \cdot y_r + b_y, \tag{12}$$

where

$$w = (z_v - z_{vc})/(z_{vf} - z_{vc}); \tag{13}$$

$$a_x = a_{xc} \cdot (1 - w) + a_{xf} \cdot w; \tag{14}$$

$$a_y = a_{yc} \cdot (1 - w) + a_{yf} \cdot w; \tag{15}$$

$$b_x = b_{xc} \cdot (1 - w) + b_{xf} \cdot w; \tag{16}$$

$$b_y = b_{yc} \cdot (1 - w) + b_{yf} \cdot w. \tag{17}$$

Once the calibration procedure is finished, the virtual depth is calculated with the depth equation (Eq. 6). The parameters for the horizontal and vertical equations (Eqs. 14, 15, 16, and 17) are then determined based on the depth weight (Eq. 13). Finally, the virtual x and y coordinates are computed (Eqs. 11 and 12).

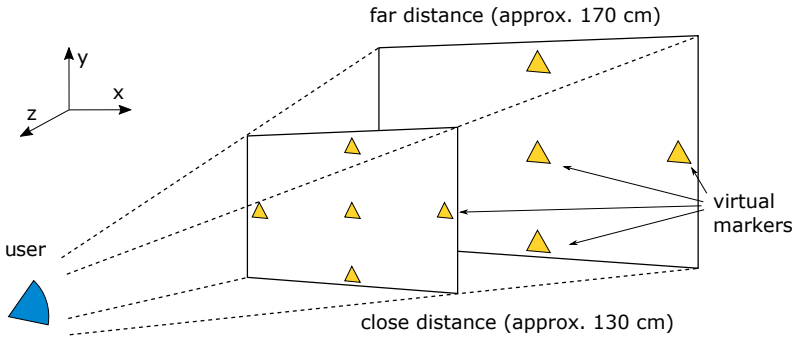

Figure 9. Calibration points are gathered at two distances (close and far) by displaying virtual markers at ten different positions that user must match to physical weapon

With the obtained coordination mapping that was calculated during the calibration procedure, it is possible to display a virtually generated trajectory that follows the tip of a blade. During their training, fencers will 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 real-time immersive feedback). Once a fencer finishes an action, the system compares the current trajectory with the model (see Section 3.3) and displays numerical evaluation scores for both the trajectory and the rotation. The rotation is additionally color-coded using a green-to-red transition.

4. Experimental results

4.1. 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 that were performed with a weapon that was equipped with the double-LED marker and were captured with the camera that was built into the AR glasses (with a resolution of 640 × 480). At each location, no other light sources nor reflections were present. In all of the video frames, the positions of the LEDs were manually labeled in order to provide the ground truth. The threshold for automatic detection was set automatically with the proposed calibration procedure (see Section 3.2). The position of the tip of the blade, the pixel distance between the LEDs, and the rotation were computed for both the 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 1. The detection rate is presented as well; this was computed as the percentage of frames in which both LEDs were detected correctly. An LED is considered to be detected correctly when the difference between the automatically found position and the ground truth is less than half the distance between the two LEDs in the ground truth.

Table 1

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

Parameter	Location 1 (Univer. lab.)	Location 2 (Train. Hall A)	Location 3 (Train. Hall B)	Average
No. 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 greater than 1 pixel in all of the recordings. The distance between the LEDs was estimated to have an average error of a little more than half a pixel. For reference, the image size was 640 × 480, and the pixel distance between the LEDs was typically between 20 to 30 pixels. The average rotation value error was less than 1 degree. The results indicate that the proposed detection method provided high accuracy; there were only a few frames in which the LEDs were not located correctly due to the very fast motion (this resulted in very blurry images). Typically, weapon actions are not performed at maximum speeds during training; therefore, this issue will rarely occur.

The proposed method for the depth estimation (which is based on the pixel distance between the LEDs in a captured image) was verified in the following manner. The AR glasses were set to focus on a measuring tape that was placed on the floor, and the weapon with the double-LED marker was moved along the tape at 10-cm intervals (see Fig. 10). The pixel distances between the LEDs in the captured images were computed automatically using the proposed detection method. Figure 11 presents the LEDs' relative pixel distances plotted against their actual distances from the camera. In the measured range (which was 50 to 170 cm), the dependency was non-linear due to the wide-angle lens in the camera of the AR glasses (see Fig. 11a). However, the tip of a blade is typically between 130 and 170 cm during bladework training; for such a small range, the dependency can be approximated with a linear function (see Fig. 11b). This indicates, that the proposed depth-estimation method is appropriate for the considered scenario.

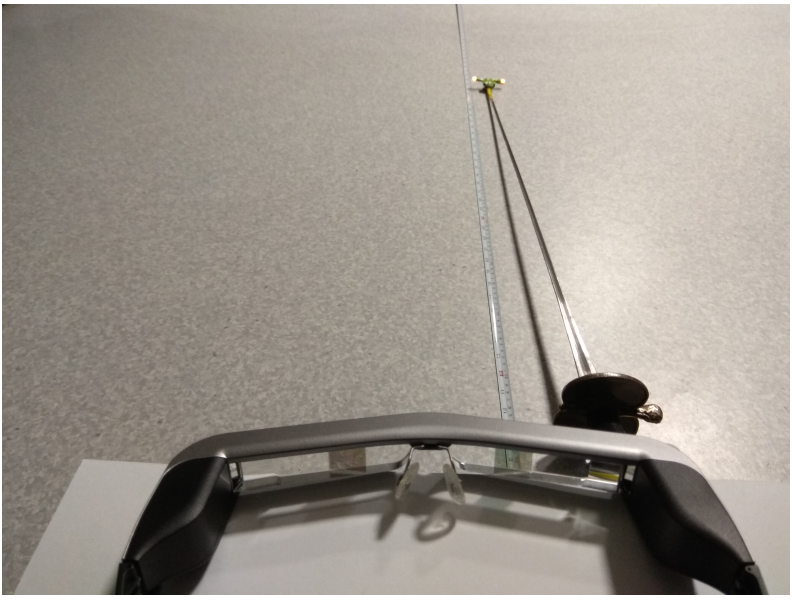

Figure 10. Setup for depth-estimation experiments

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

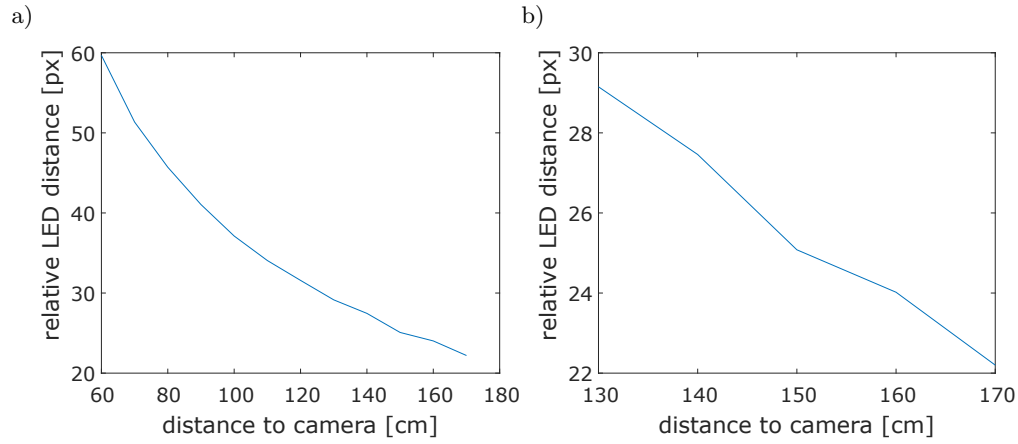

Figure 11. Relative pixel distance of detected LEDs plotted against their actual distances from camera: full measured range (a); typical range for bladework training (b)

4.2. Action models

The purpose of the action models is to provide visual cues as well as to allow for a numerical evaluation of bladework training. The goal of this numerical evaluation is to award 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 bladework training, it comes down to human perception. Since the system is supposed to provide feedback close to that of a coach, the result of the evaluation of the trajectories should be consistent with the coach's evaluation. Therefore, a dedicated tool was implemented in order to verify the proposed method for the action evaluation; this allows for simulating actions by drawing them with a mouse on a computer screen. The sixth-to-fourth-parry motion (see Fig. 12) was chosen for this evaluation, as it is one of the most commonly used weapon actions and can be performed without forward motion (therefore, the trajectory can be evaluated in 2D). The model for the action was created based on 30 repetitions of drawing it with a mouse. The mean trajectory is presented in red in Figure 13, with the standard deviation indicated by the circles drawn around the selected points. Several correct and incorrect actions were drawn and compared to the model trajectory (as presented in Figure 13). Based on this experiment, it was concluded that the automatic estimation of the trajectory similarity corresponds to human assessment.

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 was rather convenient and easy to use and allows users to train without any additional control interface. Based on the experiments with the fencers, one other feature was added to the system;

namely, a model editor (which allows for choosing which of the recorded repetitions should be included in the final model).

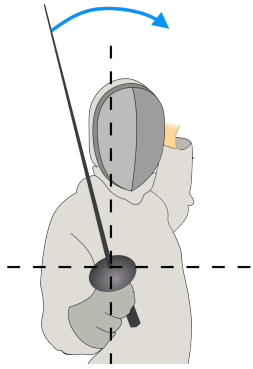

Figure 12. Sixth-to-fourth parry action in fencing

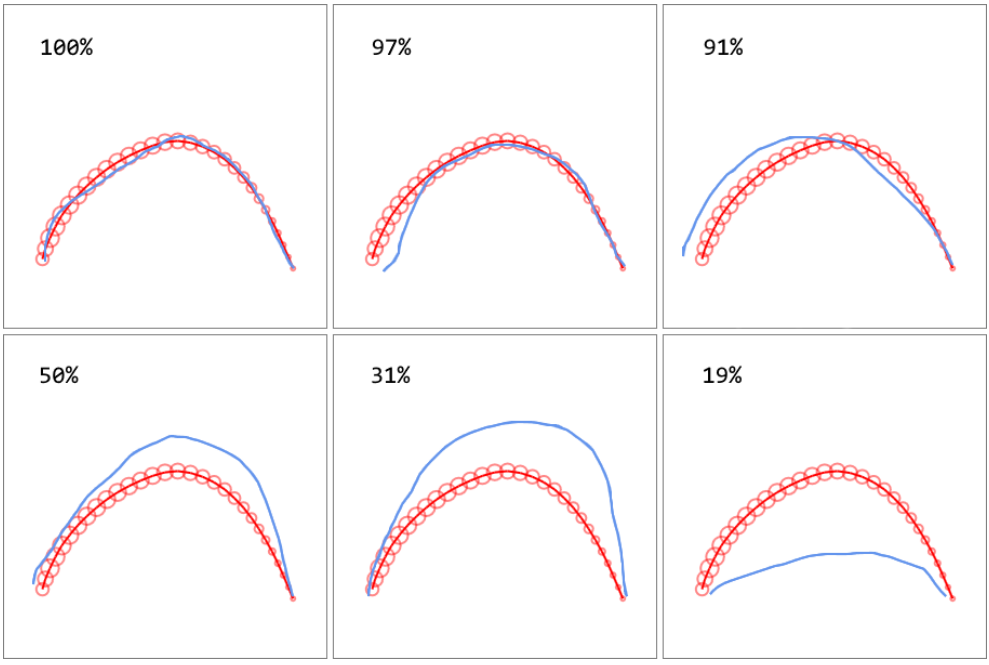

Figure 13. Verification of automatic evaluation of similarity of trajectories: red lines indicate model trajectory; red circles indicate standard deviation; blue lines indicate trajectories of practiced actions (average percentage similarity is provided)

4.3. Mixed reality

The purpose of employing a mixed reality was to provide real-time visual feedback for the training of fencers by creating a mixed virtual-real view that incorporated virtually generated trajectories that were properly aligned with the real-world weapon. The mixed view (captured with a camera that peered through the AR glasses) is presented in Figure 14. The experiments for evaluating the mixed reality included two stages: the first considered the calibration procedure, while the second regarded the usefulness of the proposed system in fencing bladework training.

Figure 14. Actual mixed real-virtual view displayed on AR glasses – virtually generated trajectory is overlayed on real-world view

The calibration procedure for the coordinate mapping was performed by three people. Each of these individuals was then asked to assess the accuracy of the blade tracking when using each of the three calibrated coordinate mappings. All three people chose the same coordinate mapping as being the most accurate; this indicated that the calibration process was not user-specific. By repeating the calibration procedure several times, it was revealed that the most important factor in the calibration was 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 of the participants were asked to assess the accuracy of the tracking of the tip of the blade (first in static positions, and then during movements). A virtual marker

was displayed in the estimated position of the tip of the blade. For the horizontal direction in the static positions, the virtual marker was always present between the LEDs (mostly in the middle, but 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 sometimes appeared slightly too close or too far away. Also, the depth tracking was less stable, as the virtual marker oscillated slightly (even when the blade was not moving). This was due to the slight differences in the keypoint--position estimation between frames, which resulted from the noise in the camera's video stream. In a test recording that lasted 30 seconds (in which the blade was stationary), the variance of the estimated keypoint distance was approx. one pixel (which corresponded to 5 cm of depth).

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

The next step of the evaluation was to verify whether the visual clues and feedback that were provided by the system resulted in performing bladework training more accurately. At first, both defensive and offensive actions were considered for the evaluation; however, the initial experiments showed that the depth perception of the displayed trajectories was very limited (as they were flat, therefore providing little 3D context). The stereo vision itself was not sufficient to provide relevant depth-change perception in this case. The estimation of the depth was still important in order to provide a proper mixed real-virtual view, but this was not used for the evaluation of the exercises. For this reason, the trajectories were only compared in 2D and (separately) with regard to the rotation. Therefore, the actions were limited to defensive ones, since these did not include significant movements along the depth direction. The defensive actions (parries) are defined by two imaginary lines that divide the action field to upper and lower (above and below the hilt) as well as inner and outer (inner on the front side of the fencer, and outer on his/her back side). Based on the position of the blade in one the four areas that are defined by these lines, the parry positions are denoted as follows: sixth (upper-outer area), fourth (upper-inner area), eighth (lower-outer area), and seventh (lower-inner area). Parry actions are defined as changes between parry positions; see Fig. 12 (for example, the sixth-to-fourth parry action). For the experiments, four of the most commonly used parries were chosen: the sixth-to-fourth, fourth-to-sixth, sixth-to-eighth, and sixth-to-seventh.

Eight people participated in the experiments – none had any prior experience in fencing, as the goal was to verify whether the proposed system would allow novice fencers to better perform their exercises. The procedure for the experiment was as

follows. First, a person was introduced to the system with the simple exercise of making a straight line action with a weapon by following the displayed model trajectory. Once the person became comfortable with using the system, the actual exercises started. For each parry action, a person was showed how this action should be performed by a fencing coach. Then, the person was asked to perform the action a few times without the proposed system to get comfortable with the exercise. At this point, the coach was allowed to help the person. Next, the person performed repetitions of the action with the proposed system set to tracking only – a starting point was displayed, but the model trajectories and performance scores were hidden (even thought the system tracked and scored the actions). Finally, the person performed the exercise with the support of the system (which displayed the color-coded model trajectory) as well as the evaluation scores for both the blade trajectory and rotation. Each person performed a total of 30 repetitions of each exercise (15 with the support of the system, and 15 without). This number of repetitions was selected due to the fact that, in the initial experiments, more repetitions resulted in too much fatigue (which could impact the results). A coach was present during the exercises – he was not allowed to interfere at this point but was supposed to observe the differences in the performances of the actions with and without the support of the system.

The results of the evaluation (regarding the trajectory and rotation scores) are presented in Tables 2 and 3, respectively. With regard to the trajectory, a significant improvement in the obtained scores can be observed in almost all of the cases. Only in two cases were the scores with and without support similar; in two other cases, the scores without the system were higher. The average improvements of the scores when using the system was 15.00–17.37 for the first three exercises and 8.25 for the final one. The sixth-to-seventh parry action is easier than the other ones, as it requires a simpler motion; therefore, the participants performed this exercise better (even without the support of the proposed system).

Table 2

Mean trajectory scores obtained by participants with (w/s) and without (wo/s) support of proposed system; last row contains average improvements for each parry action

Person	Sixth-to-fourth		Fourth-to-sixth		Sixth-to-eighth		Sixth-to-seventh	
	wo/s	**w/s**	wo/s	**w/s**	wo/s	**w/s**	wo/s	**w/s**
1	10	**37**	10	**55**	16	**29**	32	**43**
2	9	**30**	12	**28**	7	**19**	7	**14**
3	13	**36**	10	**25**	16	**29**	11	**21**
4	55	**56**	65	**45**	19	**42**	24	**32**
5	15	**32**	11	**25**	28	**29**	25	**18**
6	7	**34**	21	**56**	17	**56**	39	**60**
7	21	**35**	15	**31**	21	**28**	35	**43**
8	31	**40**	13	**29**	14	**26**	17	**25**
average improvement	17.37		17.13		15.00		8.25	

Table 3
Mean rotation scores obtained by participants with (w/s) and without (wo/s) support of proposed system; last row contains average improvements for each parry action

Person	Sixth-to-fourth		Fourth-to-sixth		Sixth-to-eighth		Sixth-to-seventh	
	wo/s	**w/s**	wo/s	**w/s**	wo/s	**w/s**	wo/s	**w/s**
1	31	**36**	27	**63**	26	**29**	62	**84**
2	14	**49**	30	**31**	16	**21**	31	**51**
3	33	**47**	25	**29**	24	**21**	67	**66**
4	63	**64**	76	**72**	24	**23**	40	**55**
5	28	**39**	13	**32**	12	**20**	32	**63**
6	67	**41**	54	**65**	18	**39**	42	**71**
7	21	**35**	19	**31**	17	**18**	33	**40**
8	60	**61**	48	**54**	25	**24**	59	**72**
average improvement	6.86		10.63		4.12		17.00	

This was the reason for the lesser improvement. With regard to the rotation average, the improvements in scores per actions ranged from 4.12 to 17.00. Those scores that were obtained with the support of the system were mostly better, although no improvement or even worse scores were observed in several cases. Also, the improvements in the rotation scores were smaller than in the case of the trajectory score. This was most likely due to the fact that the participants were more focused on the trajectories.

The weapon training was also evaluated by a coach. The most important observation was that, when using the system, all of the participants performed the actions more correctly than typical novice fencers. Novice fencers tend to perform the parry motion too widely and with significantly too much or too little rotation. The visual cues for the rotation and movement range that were provided by the proposed system largely prevented such errors in our experiments. However, it was also stated that the motion of the base of the blade should be corrected with the system as well. As discussed in Section 3.2, the motion of the base of the blade is one of the key factors in weapon-action performance; however, this was not considered by the proposed system due to hardware limitations (the insufficient field of view of the camera in the AR glasses). Finally, it was suggested by the coach that, for the offensive actions, the depth could be color-coded (similar to rotation in defensive actions). This may be investigated in future work.

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 is displayed is limited and, therefore, does not allow users to perform bladework actions in the full range of the bladework. This problem can be addressed by merely employing different AR glasses with a larger display. It was also

pointed out that, with the double-LED marker at the tip of a blade, it was not possible to practice some of the weapon actions with a partner. This problem could also be addressed by employing a better AR device that is capable of capturing the video stream with a wider angle that would allow us to mount two LEDs on the weapon's guard and only a single LED at the tip of the blade. A potential issue was a lack of compensating for head movement, which can introduce an apparent motion of the blade. No head movement is typically present during basic stationary bladework training, and no such problems occurred in the experiments. However, this issue would need to be addressed if the system were to be used in more advanced training that included footwork.

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

5. Conclusions

This goal of this work was to provide real-time visual feedback in fencing bladework training. An innovative approach was presented by employing a mixed-reality view of real weapon and generated trajectories. The proposed methods included the tracking of the 3D positions and rotations of a blade, learning weapon-action models, evaluating bladework performance based on the learned models, and mapping coordinates in the mixed reality. The proof-of-concept system was evaluated in several experiments, which included the participation of a fencing coach and eight people with no prior fencing experience. The proposed methods proved to be suitable for aiding bladework training in the case of novice fencers. The system provides real-time visual feedback without distracting the athlete with a separate screen, thereby allowing for efficient training. The main limitations of the system were due to the employed AR device. In future work, employing better AR glasses is expected to improve the proposed system by extending the field of view (due to a larger display) and enhancing weapon tracking (due to a wide-angle high-quality camera).

Acknowledgements

This work was supported by National Science Center, Poland, under Research Grant 2016/21/N/ST6/00553. We would like to thank the coaches from Aramis Fencing School (aramis.pl) for their help in developing and evaluating the system. We would also like to acknowledge sources of the graphics used in this paper: Vectorpocket (freepik.com/vectorpocket) and Wikipedia (commons.wikimedia.org/wiki)

References

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

[2] Aramis Fencing School: https://aramis.pl/, 2020. Last access on Sep 2020.

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

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

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

[6] Bottoms L., Greenhalgh A., Sinclair J.: Kinematic determinants of weapon velocity during the fencing lunge in experienced épée fencers, *Acta of Bioengineering and Biomechanics*, vol. 15(4), pp. 109–113, 2013. doi: 10.5277/abb130414.

[7] 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, *Motricidade*, vol. 14(4), pp. 94–102, 2018.

[8] 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, *Multimedia Tools and Applications*, vol. 60(3), pp. 641–667, 2012.

[9] Ciaparrone G., Sánchez F.L., Tabik S., Troiano L., Tagliaferri R., Herrera F.: Deep learning in video multi-object tracking: A survey, *Neurocomputing*, vol. 381, pp. 61–88, 2020.

[10] Cosco F., Garre C., Bruno F., Muzzupappa M., Otaduy M.A.: Visuo-haptic mixed reality with unobstructed tool-hand integration, *IEEE Transactions on Visualization and Computer Graphics*, vol. 19(1), pp. 159–172, 2012.

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

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

[13] Dalal N., Triggs B.: Histograms of oriented gradients for human detection. In: *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 1, pp. 886–893, IEEE, 2005.

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

[15] El Jamiy F., Marsh R.: Survey on depth perception in head mounted displays: distance estimation in virtual reality, augmented reality, and mixed reality, *IET Image Processing*, vol. 13(5), pp. 707–712, 2019.

[16] Epson BT-300 smart glasses: https://www.epson.eu/products/see-through-mobile-viewer/moverio-bt-300?productfinder=bt300, 2020. Last access on Sep 2020.

[17] ES O.: OpenGL ES, https://www.khronos.org/opengles/, 2020. Last access on Sep 2020.

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

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

[20] Kersten-Oertel M., Jannin P., Collins D.L.: DVV: a taxonomy for mixed reality visualization in image guided surgery, *IEEE Transactions on Visualization and Computer Graphics*, vol. 18(2), pp. 332–352, 2011.

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

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

[23] Malawski F.: Real-Time First Person Perspective Tracking and Feedback System for Weapon Practice Support in Fencing. In: *APPIS*, pp. 79–88, 2018.

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

[25] Manafifard M., Ebadi H., Moghaddam H.A.: A survey on player tracking in soccer videos, *Computer Vision and Image Understanding*, vol. 159, pp. 19–46, 2017.

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

[27] Mehta D., Sridhar S., Sotnychenko O., Rhodin H., Shafiei M., Seidel H.P., Xu W., Casas D., Theobalt C.: Vnect: Real-time 3d human pose estimation with a single RGB camera, *ACM Transactions on Graphics (TOG)*, vol. 36(4), pp. 1–14, 2017.

[28] Mueller F., Bernard F., Sotnychenko O., Mehta D., Sridhar S., Casas D., Theobalt C.: GANerated Hands for Real-Time 3D Hand Tracking from Monocular RGB. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 49–59, 2018.

[29] Parisot P., De Vleeschouwer C.: Scene-specific classifier for effective and efficient team sport players detection from a single calibrated camera, *Computer Vision and Image Understanding*, vol. 159, pp. 74–88, 2017.

[30] Regenbrecht H., Baratoff G., Wilke W.: Augmented reality projects in the automotive and aerospace industries, *IEEE Computer Graphics and Applications*, vol. 25(6), pp. 48–56, 2005.

[31] Reily B., Zhang H., Hoff W.: Real-time gymnast detection and performance analysis with a portable 3D camera, *Computer Vision and Image Understanding*, vol. 159, pp. 154–163, 2016.

[32] Santos dos Júnior J.G., Monte Lima do J.P.S.: Particle swarm optimization for 3D object tracking in RGB-D images, *Computers & Graphics*, vol. 76, pp. 167–180, 2018.

[33] Thomas G., Gade R., Moeslund T.B., Carr P., Hilton A.: Computer vision for sports: Current applications and research topics, *Computer Vision and Image Understanding*, vol. 159, pp. 3–18, 2017.

[34] Wang P., Bai X., Billinghurst M., Zhang S., Han D., Sun M., Wang Z., Lv H., Han S.: Haptic Feedback Helps Me? A VR-SAR Remote Collaborative System with Tangible Interaction, *International Journal of Human–Computer Interaction*, vol. 36(13), pp. 1242–1257, 2020. doi: 10.1080/10447318.2020.1732140.

[35] Yovcheva Z., Buhalis D., Gatzidis C., Elzakker van C.P.: Empirical Evaluation of Smartphone Augmented Reality Browsers in an Urban Tourism Destination Context, *International Journal of Mobile Human Computer Interaction (IJMHCI)*, vol. 6(2), pp. 10–31, 2014.

Affiliations

Filip Malawski
AGH University of Science and Technology, Institute of Computer Science, Krakow, Poland, fmal@agh.edu.pl, ORCID ID: https://orcid.org/0000-0003-0796-1253

Received: 30.11.2021
Revised: 20.02.2022
Accepted: 20.02.2022