ZUR ERLANGUNG DES AKADEMISCHEN GRADES DES DOKTORS DER NATURWISSENSCHAFTEN (DR. RER. NAT.) AN DER TECHNISCHEN FAKULTÄT DER UNIVERSITÄT BIELEFELD

MOTOR LEARNING IN VIRTUAL REALITY: FROM MOTION TO AUGMENTED FEEDBACK

VORGELEGT VON

FELIX HÜLSMANN

BIELEFELD, 2019

VERSICHERUNG

Hiermit versichere ich,

- dass mir die geltende Promotionsordnung der Fakultät bekannt ist,
- dass ich die Dissertation selbst angefertigt habe, keine Textabschnitte von Dritten oder eigenen Prüfungsarbeiten ohne Kennzeichnung übernommen und alle benutzten Hilfsmittel und Quellen in meiner Arbeit angegeben habe,
- dass Dritte weder unmittelbar noch mittelbar geldwerte Leistungen von mir für Vermittlungstätigkeiten oder für Arbeiten erhalten haben, die im Zusammenhang mit dem Inhalt der vorgelegten Dissertation stehen,
- dass ich die Dissertation noch nicht als Prüfungsarbeit für eine staatliche oder andere wissenschaftliche Prüfung eingereicht habe und
- dass ich keine gleiche, oder in wesentlichen Teilen ähnliche oder eine andere Abhandlung bei einer anderen Hochschule als Dissertation eingereicht habe.

Bielefeld, 2019

Felix Hülsmann

Printed on non-aging paper in compliance with DIN-ISO 9706.

ACKNOWLEDGMENTS

I am deeply thankful to all the people who supported me during the time I was working on this thesis. Without the help of all the people who told me not to give up and who helped me to maintain my motivation, this thesis would not have been possible. I also thank all the people in my scientific environment for the fruitful discussions and support. I especially thank Prof. Mario Botsch for teaching me to always strive for the best possible contribution and to always head for the maximum. Further, I like to thank Prof. Ipke Wachsmuth for arousing my interest in science.

This work was supported by the Cluster of Excellence Cognitive Interaction Technology "CITEC" (EXC 277) at Bielefeld University, which is funded by the German Research Foundation (DFG).

ABSTRACT

Sports and fitness exercises are an important factor in health improvement. The acquisition of new movements—*motor learning*—and the improvement of techniques for already learned ones are a vital part of sports training. Ideally, this part is supervised and supported by coaches. They know how to correctly perform specific exercises and how to prevent typical movement errors. However, coaches are not always available or do not have enough time to fully supervise training sessions. Virtual reality (VR) is an ideal medium to support motor learning in the absence of coaches. VR systems could supervise performed movements, visualize movement patterns, and identify errors that are performed by a trainee. Further, feedback could be provided that even extends the possibilities of coaching in the real world. Still, core concepts that form the basis of effective coaching applications in VR are not yet fully developed. In order to diminish this gap, we focus on the processing of kinematic data as one of the core components for motor learning. Based on the processing of kinematic data in real-time, a coaching system can supervise a trainee and provide varieties of multi-modal feedback strategies.

For motor learning, this thesis explores the development of core concepts based on the usage of kinematic data in three areas. First, the movement that is performed by a trainee must be observed and visualized in real-time. The observation can be achieved by state-of-the-art motion capture techniques. Concerning the visualization, in the real world, trainees can observe their own performance in mirrors. We use a virtual mirror as a paradigm to allow trainees to observe their own movement in a natural way. A well established feedback strategy from real-world coaching, namely improvement via observation of a target performance, is transfered into the virtual mirror paradigm. Second, a system that focuses on motor learning should be able to assess the performance that it observes. For instance, typical errors in a trainee's performance must be detected as soon as possible in order to react in an effective way. Third, the motor learning environment should be able to provide suitable feedback strategies based on detected errors. In this thesis, real-time feedback based on error detection is integrated inside a coaching cycle that is inspired from real-world coaching. In a final evaluation, all the concepts are brought together in a VR coaching system. We demonstrate that this system is able to help trainees in improving their motor performance with respect to specific error patterns. Finally, based on the results throughout the thesis, helpful guidelines in order to develop effective environments for motor learning in VR are proposed.

CONTENTS

1 INTRODUCTION

Sports training is an ideal way to increase fitness and health [WNB06]. For a sports training to be effective, learning of new motor tasks and improvement in already learned ones are vital. Often it is useful and much more efficient to learn motor tasks with the help of a coach. The coach provides the trainee with information on whether a given motor task was executed correctly and what kind of errors were made. Further, the coach makes suggestions and proposes guidelines on how to improve a motor performance. However, coaches are not always available: In gyms a coach typically has only a limited amount of time to supervise a trainee. Most of the time, especially when training at home, trainees are on their own. When training alone without a coach, the motion performed by a trainee remains uncorrected. This can lead to potential motor errors that are reinforced due to training an incorrect course of movement. If a technical environment exists that supervises a trainee's motion and that is suitable to provide helpful feedback, a high quality of training could be preserved even in periods of time where no coach is available. Here, the extensive capabilities of virtual reality (VR) seem to be an ideal candidate to facilitate and boost the learning process [DHS18; Neu+18; RK05; Sch+14]. Coaching environments for motor learning are becoming a more and more popular research topic in VR [Cha+11; Kok+15; Kya+15; Neu+18; Sig+15]. Such environments provide the opportunity to introduce innovative types of augmented feedback that can even exceed real world opportunities, such as augmented feedback strategies or multi-sensory stimuli [Chu+03; Sig+13; Sig+15]. In their review, Miles et al. especially highlight the flexibility of VR environments and their possibility to provide extra information as a reason to use them in sports training [Mil+12]. VR motion capture systems are able to obtain objective kinematic data of a trainee in real-time. Further, they allow highly individualized training sessions [DHS18] while maintaining the ability of being precise and highly reproducible [Neu+18]. An effective setup for motor learning in sports might even be used in the context of rehabilitation. For the periods of time where no rehabilitation coach is available, trainees could continue training and the coach could check the system's report afterwards.

However, there exist some pitfalls in the context of motor learning in VR. One category of pitfalls are technical issues such as the latency induced by typical VR environments [Mil+12]. Further problems refer to inconclusive results in the field of motor learning in VR [Mil+12] and the unclear impact of specific manipulations and feedback strategies [Sig+13]. As indicated by Neumann et al. in their review on interactive VR in sports, a theoretical framework concerning VR application for sports would be desireable [Neu+18]. Current approaches focus mainly on technical properties of the environment (e.g., [Wal+16]), or the analysis of performed motion (e.g., [Bev+18; BOL17]), or on specific feedback strategies (e.g., [Cha+11; Rah+18; Sig+15]). Properties of the system that are not directly in the focus of a specific contribution are typically oversimplified or just not discussed at all. This finally leads to results that can be inconclusive and impossible to reproduce.

This work aims at reducing such gaps in existing literature. To this end, we

consider motion and kinematic data as a core part of VR applications for motor learning. Consequently, we focus on motion data as a basis in order to develop core concepts for VR motor learning applications. We follow an integrated approach that consists of mainly three steps. First, a motor learning system must be capable to *observe* and to *visualize* motion data. The observation needs to be performed by motion capture systems, a visualization can consist of mapping the recorded motion on a trainee's virtual avatar. Here, we focus on a virtual mirror paradigm to allow trainees to observe their own movement in a realistic way. Our results indicate that motion that is observed by our system and that is visualized by using virtual avatars can be used as a basis to help novices in adapting their motor performance. To this end, we transfer a well established feedback strategy from real-world coaching, namely improvement via observation [AP13; AP14; MBZ76; RP11], into the virtual mirror paradigm. Second, a coaching system needs to assess the trainee's performance of a specific exercise — in the following called motor action — in order to *detect the occurrence of typical error patterns* performed by a trainee. Third, results from steps one and two need to be combined to *generate feedback* to the trainee. All three steps need to be performed online, already during a trainee's performance in a closed-loop interaction. We propose an integrated pipeline towards the online classification of typical errors in a trainee's performance and the generation of augmented feedback based on the properties of the learned classifier. We demonstrate that our detection of typical errors in motor performances works with a high accuracy and that we can, based on detected errors, provide online feedback to a trainee. In a user study, we show that the final system that integrates all components developed in this thesis can help trainees in improving their performances with respect to the error patterns the system coaches. Throughout this thesis, we develop and evaluate a VR environment for motor learning from the very first steps, namely the basic VR system (see Chapter 2), to its final application (see Chapter 6). See Figure 1.1 for an impression of how our resulting setup is used by a trainee in order to learn the squat. Based on our findings, we finally propose guidelines in order to develop effective environments for motor learning in VR.

Holden suggest that VR systems that target motor learning should be developed not only from a technical point of view, but also based on knowledge provided from an interdisciplinary point of view [Hol05]. Consequently, we develop our core concepts for motor learning in VR, in addition to knowledge from computer science, also based on literature from the field of motor learning and via collaborating with experts from multiple disciplines such as movement science and psychology. For transparency reasons, at the beginning of each chapter, the contribution of the author of this thesis to the contents of the chapter as well as the contributions of his collaborators are summarized.

This thesis is structured as follows: In Chapter 2, we first investigate requirements for VR applications for motor learning and evaluate state-of-the art systems with respect to these requirements. Then, based on these results, we present an environment that satisfies the requirements. Chapter 3 contains an experiment that has been performed in our motor learning environment. This experiment indicates that novices are able to adapt towards a skilled performance when watching an own avatar together

Figure 1.1: A trainee interacts with the motor learning environment proposed in this thesis.

with a superimposed skilled performance during practice. We show that improvement depends on the perspective chosen to display the superimposed performance. In the subsequent Chapter 4 we develop an extension of open-end Dynamic Time Warping in order to accurately align a movement that is currently performed by a trainee with a reference movement. In Chapter 5, we propose a pipeline towards the classification of errors performed by a trainee in order to provide online feedback. Here we propose two options, one rule-based option that does not require training data as well as a data-driven option. All findings and components are combined to a final coaching environment in Chapter 6. This environment is evaluated in a user study. As an outlook, in chapter 7, we provide an approach towards a portable low-cost version of our setup. We provide information on the typical pitfalls and drawbacks of such a system based on consumer hardware as compared to our final state-of-the-art environment. Chapter 8 summarizes this thesis and contains guidelines based on the findings of the single chapters. Finally, limitations of the work proposed in this thesis as well as possible directions of future work are addressed. In this work, we mainly use the squat as a test case. The squat is a full-body motor action that is used in the context of rehabilitation [BSR11; Esc01] as well as for sports training [Esc01]. When executed by novice trainees, various error patterns can be observed. Further, in some cases, we demonstrate the quality of our results using Tai Chi pushes and lateral raises.

The main contributions of this thesis are:

- We investigate requirements (hardware as well as software) in order to develop an effective VR environment for motor learning. Based on these findings, we evaluate state-of-the art approaches and demonstrate how to implement such an environment.
- We transfer an important paradigm in the field of motor learning, namely performing an exercise together with a skilled subject, to our VR environment and demonstrate its effectiveness.
- We propose a pipeline towards the classification of errors in a trainee's motor performance that is able to automatically generate feedback based on the properties of the underlying classifier.
- We combine the findings and concepts that are presented in this thesis in an integrated final coaching environment. We demonstrate the ability of this environment to support trainees in improving their motor performances.
- In our outlook, we propose an approach on how to set up a down-scaled, low-cost consumer environment. In a pilot experiment, we show that it can be, despite from typical issues of such an environment, able to help people in improving their motor performances for specific fields of application.

Parts of this thesis have been published in the following publications:

de Kok, I., Hough, J., Hülsmann, F., Botsch, M., Schlangen, D., & Kopp, S. (2015). A multimodal system for real-time action instruction in motor skill learning. In Proceedings of the 2015 ACM International Conference on Multimodal Interaction (pp. 355–362).

Waltemate, T., Hülsmann, F., Pfeiffer, T., Kopp, S., & Botsch, M. (2015). Realizing a low-latency virtual reality environment for motor learning. In Proceedings of the 21st ACM Symposium on Virtual Reality Software and Technology (pp. 139–147).

Hülsmann, F., Frank, C., Schack, T., Kopp, S., & Botsch, M. (2016). Multi-level analysis of motor actions as a basis for effective coaching in virtual reality. In Proceedings of the 10th International Symposium on Computer Science in Sports (ISCSS) (pp. 211–214). Springer.

Hülsmann, F., Kopp, S., Richter, A., & Botsch, M. (2017). Accurate online alignment of human motor performances. In Proceedings of the Tenth International Conference on Motion in Games (pp. 7:1–7:6). ACM.

Hülsmann, F., Göpfert, J. P., Hammer, B., Kopp, S., & Botsch, M. (2018). Classification of motor errors to provide real-time feedback for sports coaching in virtual reality — A case study in squats and Tai Chi pushes. Computers & Graphics, 76, pp. 47–59.

Hülsmann, F., Frank, C., Senna, I., Ernst, M., Schack, T., & Botsch, M. (2019). Superimposed skilled performance in a virtual mirror improves motor performance and cognitive representation of a full-body motor action. Frontiers in Robotics and AI, 6, pp. 43:1–43:17

The following publications are not part of this thesis, however, they were developed based on findings, technology and/or data that emerged from contributions of this thesis:

Waltemate, T., Senna, I., Hülsmann, F., Rohde, M., Kopp, S., Ernst, M., & Botsch, M. (2016). The impact of latency on perceptual judgments and motor performance in closed-loop interaction in virtual reality. In Proceedings of the 22nd ACM Symposium on Virtual Reality Software and Technology (pp. 27–35).

Hosseini, B., Hülsmann, F., Botsch, M., & Hammer, B. (2016). Non-negative kernel sparse coding for the analysis of motion data. In International Conference on Artificial Neural Networks (pp. 506–514). Springer.

de Kok, I., Hülsmann, F., Waltemate, T., Frank, C., Hough, J., Pfeiffer, T., Schlangen, D., Schack, T., Botsch, M., and Kopp, S. (2017). The Intelligent Coaching Space: A Demonstration. In International Conference on Intelligent Virtual Agents (pp. 105–108). Springer.

2 LOW LATENCY ENVIRONMENT

Published in: [Wal+15]

A VR environment can be equipped with high-precision sensors and can employ various feedback channels. Thus, the extensive capabilities of VR seem to be an ideal candidate to facilitate and boost the process of motor learning [RK05; Sch+14]. Highly precise sensors gather data of the trainee, which are analyzed in real time, in order to provide directed purposeful feedback over various channels. This feedback can either be given after the movement execution or—more interestingly and more useful—during the execution. Especially in the latter case it is important to ensure that the feedback is precisely timed, so that it is presented exactly when it is relevant. As a consequence, the environment has to be highly controlled, i.e., properties like the end-to-end latency or tracking robustness either must be controlled or at least have to be taken into account. It thus seems necessary to report such basic properties of a system in every research addressing issues of motor learning in VR. This would allow researchers to compare systems and to reproduce studies more reliably.

At the time being, no general guidelines for VR environments targeted at motor learning seem to exist. Furthermore, for many systems described in literature, no sufficient information on relevant aspects such as end-to-end latency, robustness of the motion capture system, et cetera is given. Thus, when building up a new VR environment, one is faced with a vast number of potential techniques and technologies, but a well-informed choice is hardly possible.

This chapter deals with improving this situation by

1. providing general requirements towards VR systems for motor learning,
2. evaluating and assessing state-of-the-art techniques and technologies,
3. presenting a system built according to the aforementioned requirements,
4. providing latency measurements of the virtual environment and giving hints on how to reduce latency.

A minimal VR system for motor learning would consist of components for motion capturing, pre-processing of motion data, motion analysis, feedback generation, rendering, and display technology (see Figure 2.1). As rendering and motion capturing are the backbones of the VR environment for motor learning, we focus on these two components in this chapter.

In the following, we start by developing general requirements towards motor learning in VR applications (Section 2.1). After discussing related motor learning approaches in Section 2.2, we present in Section 2.3 the essentials of our low-latency VR environment, while also assessing particular state-of-the-art techniques and technologies for motion capturing and real-time rendering. In Section 2.4, we present an evaluation of our system. In Section 2.5 we explain how we use our environment to record training data for the motion analysis proposed in Chapter 4 and Chapter 5. Finally, we summarize and conclude this chapter in Section 2.6.

My Contribution *The VR environment for motor learning presented in this chapter was developed in close cooperation with Thomas Waltemate. I contributed via the evaluation and*

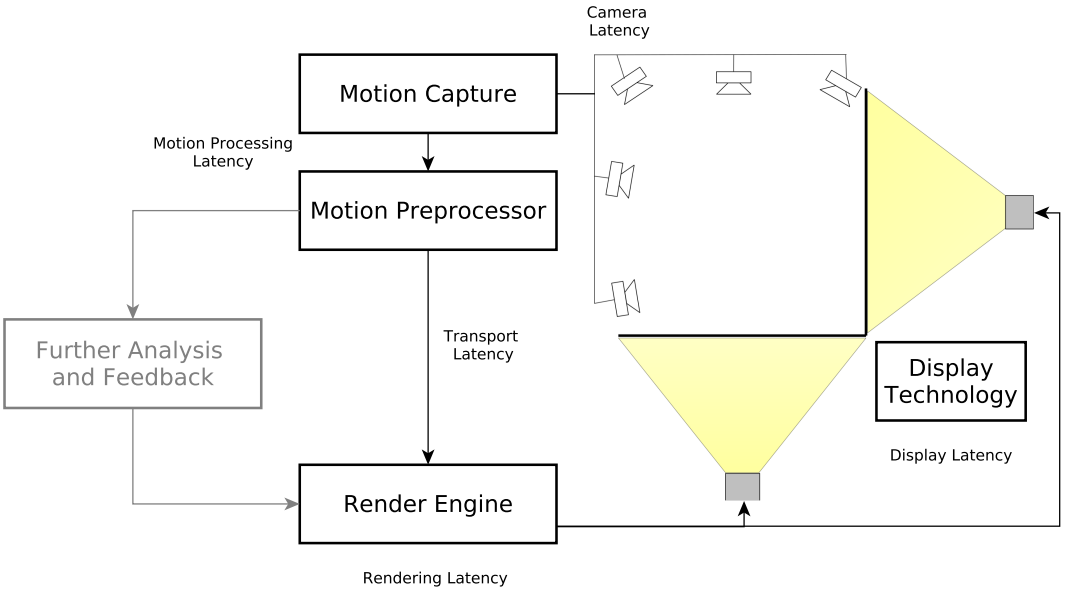


Figure 2.1: A minimal architecture for a VR environment for motor learning combines a motion capturing system, motion processing (e.g., for re-targeting or motion analysis), as well as a render engine for high-fidelity character rendering.

selection of the motion tracking systems. Further, I worked on motion preprocessing. Additionally, I contributed to setting up the whole system and to the compilation of the requirements. Moreover, I adapted the original pendulum-based approach for latency measurement. The measurements were done together with Thomas Waltemate. Thomas Waltemate developed the render engine and the techniques to visualize the virtual mirror. Additionally, he contributed to setting up the whole system and the compilation of the requirements. The data sets presented here were recorded in collaboration with Irene Senna, Cornelia Frank, Iwan de Kok, and Julian Hough. I developed the setup for data recording, preprocessing and annotation.

2.1 REQUIREMENTS

In this section we develop requirements necessary for an efficient motor learning system in VR. Many researchers already pointed out some of the most crucial requirements for VR applications in general: For instance, Bierbaum et al. [Bie00] provide an overview including general features like low latency, high frame rate, tracking robustness, but also engineering requirements such as extensibility and hardware abstraction. To our knowledge, this has not yet been done for VR systems specialized on motor learning. In the following we therefore carve out the most important requirements.

R1: Feedback on one's own motion

As a first requirement, users have to be able to verify the correct execution of a given motor task by getting feedback of whatever kind. This feedback should be as intuitive

as possible, and one of the most intuitive ways is to let users observe their own motion by viewing their own body.

In real world scenarios, like fitness or dance studios, self-monitoring is usually achieved through a mirror. Thus, it seems desirable to provide mirror-like feedback in VR training environments as well [Häm04]. This is inter alia motivated by findings of Chua et al., who found that none of their proposed layouts of students and teachers could improve upon a standard face-to-face configuration—similar to that of a mirror—when learning Tai Chi [Chu+03]. A *virtual* mirror, as planned in our setup, may serve multiple purposes: it may show the optimal performance, just as a teacher would, to guide the performance of the trainee; it can simply reflect the real performance of the trainee to support self-monitoring; or it could add augmentations to the real performance, e.g., emphasizing errors. Finally, it serves as a perfect base for further feedback strategies. Besides face-to-face layouts, a third person view could also improve training results, as has been recently shown by Covaci et al. [COM14]. In summary it can be said that self-monitoring is an essential ingredient on the way towards meaningful visual feedback.

R2: Low latency and high frame rate

The times in which any response delay below 1 s was considered acceptable—as had been suggested by Shneiderman [Shn84]—are long over. Work in the area of system response time suggests that delays in the range of 80–100 ms will not be noticeable by the majority of users. In a study by Mauve et al. users did not notice network lags below 120 ms, thus, depending on the applications, tolerable latencies might be even higher than 80–100 ms [Mau+04]. Gutwin showed that in a simple coordination task a delay of 200 ms already significantly increased the error rates [Gut02]. However, the examples used in such studies are primarily targeting human-machine interaction or manipulation of objects and do not address the issue of self-perception and self-monitoring.

Research on the effects of latency in the context of virtual environments has been primarily focused in research on distributed virtual environments. In this context, Roberts et al. define the time required to present the user's actions back to the user as local latency [RSS95]. In a study on collaborative virtual environments, Park et al. show that with increased latencies, humans adopt a move-and-wait strategy, waiting several seconds to let their views synchronize, before continuing performing their tasks [PK99]. They showed that in such setups jitter had a larger impact on collaborative performance than latency. The development of similar strategies has to be avoided in our target scenario, as it would hamper with the natural flow of movements.

Regarding display latencies, it has been shown that trained users are able to detect a latency of perspective adaptation of about 15 ms in a HMD-based study [Man+04]. In CAVE- or Powerwall-based VR systems, latency is less critical, as the projection screens remain stationary. However, a highly responsive system is important in terms of task performance and presence in VR environments in general [Mee+03]. In particular for HMDs a high frame rate is also important. We thus want to separate the requirements regarding latency induced by the display technology (HMD or projection-based) from

the requirements regarding a low-latent update of visual feedback, e.g., of a figure animated via motion capturing. In our description we are focusing on the latter.

For *feedback on one's own motion* (R1) in a virtual mirror, latency-induced effects could be reduced since humans can use motor prediction to adapt to delayed sensory feedback [HHW09; KV12; RE12]. Still, having more complex feedback strategies and an augmented virtual mirror in mind, low latency will become even more important for precise presentation of feedback (e.g., an avatar pointing at erroneous parts of the user's body during movement execution).

However, no fixed rules concerning the maximum allowed level of latency in VR motor learning applications exist. Literature suggests values of 150 ms for controlling characters in computer games, since higher latencies are already directly noticeable for untrained users and affect players in several ways [JNS12]. Meehan et al. showed that decreasing the latency from 90 ms to 50 ms already affects presence in virtual environments [Mee+03]. MacKenzie et al. used Fitt's tapping task to investigate the influence of latency on performance: They found that the performance of participants is reduced when being exposed to a latency of 75 ms or higher [MW93]. According to Ware et al. even a latency of 70 ms already affects performance in a VR reaching task [WB94]. In a non-VR tapping task Jota et al. found that performance improves only little using latencies below 25 ms [Jot+13]. Even for latencies below 50 ms, only a very slight improvement was measured. Improvements in latencies below 40 ms were not even noticed by most untrained participants. In our own research, we evaluated the impact of different levels of latency of an avatar visualized inside a virtual mirror [Wal+16]. To this end, we used the system described in this chapter. Here, we observed an awareness of participants towards latencies in the interval between 75 ms and 125 ms. Further, such a latency worsened participants' motor performance. Agency and ownership were affected for latencies above 125 ms [Wal+16].

In conclusion, it seems to be desirable to reach the lowest possible latency. However, a corridor of end-to-end latencies between 40 ms and 70 ms seems to be still acceptable, depending on the specific application. The latency of a VR system is composed of four components (cf. Figure 2.1):

- Latency in motion capture (cameras, preprocessing)
- Transport latency
- Visualization latency (rendering, displays)

In order to reduce the system's end-to-end latency, all these components of latency should be minimized.

R3: Minimal level of disturbance

To guarantee a natural and intuitive training, the user should be able to move freely, at least regarding the movements that are relevant for the motions to be trained. Thus the hardware attached to the user has to be as unobtrusive as possible, since otherwise the user would not be able to use her full range of motion. For instance, the use of long and stiff wires as well as heavy components should be prevented if possible: Participants should perform the motor actions as they would in a real training scenario. Besides issues of naturalness of movements, obtrusive hardware could also make

the optimal perception of the virtual environment more difficult [WS98]. Therefore motion capturing system and VR environment have to be chosen to offer a reasonable compromise between tracking precision, immersion, and obtrusiveness.

R4: Robust tracking

Many typical sports exercises include movements during which parts of the body are occluded for outside-in tracking systems. The motion capture system has to be as robust as possible against such kind of occlusions, where single or multiple markers might get lost. If the tracking is not robust enough, it might require a re-calibration of the human that is to be tracked. Thus, the training has to be interrupted and cannot be continued until the re-calibration is performed. The training is severely affected by such a re-calibration procedure to re-align tracking: If this happens, the naturalness of the application, as for instance demanded by Witmer and Singer [WS98], would be significantly reduced.

Summary

We developed these requirements with a focus on VR applications for a large spectrum of motor learning. Some of the proposed requirements depend on the field of application. For instance, if a VR environment is only used in terms of motivational aspects, feedback on the performed motion does not need to be as elaborate as when a new motor task is learned. A higher level of disturbance might be tolerable if, for instance, obtained feedback mechanisms (e.g., provided by haptic feedback devices) are helpful enough to override the negative effects. Further, tracking does not need to be maximally robust if an application only needs rough information on the performed motor task (for instance step counting in a running simulation). However, from all requirements, low latency (R2) is always crucial. If the latency is high, relevant factors for the VR experience decay (e.g., agency, ownership) and undesired effects such as simulator sickness arise [Mil+12; Pot98; Wal+16]. We therefore argue that the latencies of a VR environment for motor learning should be reported whenever presenting results of studies conducted in such a setup. This is important to exclude high latency as a potential side effect in the conducted experiments. More generally, when the exact specifications of an environment are known, the results of future experiments become better comparable as well as more reproducible. Further, for a motor learning environment that does not only target a specific narrow application, but a broad variety of different motor actions and various ways to provide feedback at multiple levels of trainees' expertise, we also suggest to comply with the other requirements.

2.2 RELATED APPROACHES

This section gives a short overview of state-of-the-art approaches to motor learning systems in VR with respect to requirements developed above. In the following, these are referenced as R1–R4.

Smeddinck et al. present a training system that covers a large range of human movements [Sme+14]. The system aims at improving motor performance for Parkinson's disease patients. Participants can monitor their own motion visualized through a coarsely rendered skeleton. Furthermore, the movement of the instructor can also be monitored, depending on experimental condition. The authors evaluate the effect of different abstractions of instruction presentations on motor performance. A Microsoft Kinect camera was used for motion capturing. The authors fulfill R1 (feedback on one's own motion), but did not provide any information on system latency (R2). However, given the latencies of the Kinect sensor, they can be expected to be well above 100 ms. Requirement R3 can be considered fulfilled as no hardware has to be attached to the user for Kinect-based motion tracking. The overall tracking robustness can be assumed to be sufficient for the task of rehabilitation for Parkinson's disease patients and the employed set of simple movements. However, using a Kinect camera might not be fast and robust enough for more complex motions (R4).

A yoga training game with a special focus on visually impaired people is presented by Rector et al. [RBK13; Rec+17]. They focus on spoken feedback to help trainees to reach a desired yoga posture. To get information about the performed movement, they also employ a Kinect camera. As the system targets visually impaired people, requirement R1, which demands for feedback on one's own motion can be seen as fulfilled via the provided spoken feedback. Indeed, the authors do not give any information on the system's latency (R2), which might be important to counter-steer over- and under-shooting movements caused by a high latency. For example, the system could state "Lean forward" based on a delayed measurement, although the user already exceeded the desired angle. Yet, yoga movements are typically rather slow, such that a high latency might only slightly influence the given task. Requirement R3, which requires a minimal level of disturbance, is fulfilled due to the marker-less Kinect tracking. Concerning the robustness of the tracking for the desired type of motion (R4), no information is given. It can be assumed that the authors chose postures that are easy to track with the Kinect camera and do not require too many changes in user orientation or self-occlusions of body-parts.

A highly specialized training system for rowing in VR is presented by Sigrist et al. [Sig+15]. The user is placed in a modified boat, surrounded by projection walls. An extended version of the rowing blade is visualized and superimposed by the optimal blade position. Furthermore, the authors employ auditory feedback, which consists of a sonified oar blade and a sound which is played when the blade enters the virtual water. Haptic feedback is applied via resistance torques against the user's movement as soon as the user's blade moves away from the target position. Virtual self-monitoring (R1) is only possible via observing the virtual oar blade. Concerning latency and frame rate (R2), no information on the overall latency is given. Only the update rate of the projectors (> 30 Hz), movement sonification (30 Hz), and the frequency of the haptic device (1000 Hz) is described. The Unity engine is used to render the virtual ocean and the motion of the oar blade. Requirement R3 is satisfied as no additional hardware except from headphones has to be attached to the user and he/she is located inside a real boat. The tracking can be assumed to be sufficiently robust (R4), since the tracking task is not very complex. Another VR system that focuses on rowing

is proposed by Arndt et al. [APV18]. They use a HMD and place the trainee on a real rowing machine. The movement of the rowing machine is tracked and used to animate virtual rowing blades. The subjects rows through virtual water. Requirement R1 is only partially fulfilled, as only the movement of the rowing blades is tracked and visualized in the virtual environment. However, we can assume that this very simple tracking task leads to a verly low latency (R2), even though the latency is not explicitly reported. Concerning disturbing hardware (R3), only the HMD is attached to the user. Despite from this, the rowing machine itself is a realistic real-world device. Also the tracking can be assumed to be robust (R4) as only the rowing machine itself is used. However, no specific information is provided. In the best case, we can assume that this system meets our requirements, however, they are not explicitly stated and can only be tried to be extracted from the textual description and images of the system. Further, the systems proposed by Arndt et al. and Sigrist et al. are very specific systems only capable of training rowing. A direct transfer of the environments proposed in [APV18; Sig+15] towards more complex full-body movements seems not to be possible.

Covaci et al. [COM14] present a training system that aims at high-precision tasks such as the basketball free throw. The system is located in a CAVE environment, hence the ball has to be attached to a special construction to prevent the walls from damage. The ball and the user are tracked by a Vicon MX motion capture system. Directly after throwing the ball, the system calculates the trajectory of the ball and visualizes the throw. The users can monitor their own motion (R1) either in first- or in third-person perspective. The third-person perspective can also be overlaid with the correct trajectory of the ball. The system's shutter glasses run at 30 Hz per eye, the motion capture system has a frequency of 120 Hz. Information on the system's latency is not stated (R2). In a user study, the authors showed that the overall latency did not disturb the users. Requirement R3 (minimal disturbance) was evaluated via questionnaires: The interaction was stated as natural by participants, such that R3 can be considered fulfilled. The tracking is described as being robust (R4) and the calculation of the ball trajectory leads to correct results in 87.5 % of 500 trials.

Cannavò et al. also propose a system to train the basketball free throw [Can+18]. To this end, they combined ball tracking via the Kinect 2 camera with full-body motion capture via a motion capture suit by Perception Neuron and the HTC Vive trackers. To display the virtual environment, a HTC Vive is used as display device and a game engine performs the rendering. As the system mainly displays the trainee's hands R1 can only considered as partially fulfilled. Concerning latency (R2), no information is provided. Requirement R3 is not fulfilled, as multiple hardware devices (HMD, HTC Vive trackers, etc.) are attached to the trainee. On the other hand, due to the combination of Perception Neuron motion capture suit and HTC Vive trackers, a robust tracking of the trainee's motion can be assumed (R4).

To summarize, many different approaches towards VR motor learning exist. However, information on end-to-end latency is only rarely given. Hence results are difficult to compare, e.g., concerning the achieved levels of performance, and it is difficult to exactly replicate experiments. Furthermore, some systems use sensors unable to provide a robust tracking for a broad set of possible motor actions. To the best of our knowledge, no approach published until now aims at providing a general, highly con-

trolled, efficient training environment that satisfies the above mentioned requirements and provides information on end-to-end latency. This work tries to fill this gap via description, discussion, and evaluation of state-of-the-art techniques, leading to an exemplary realization of a system that satisfies the stated requirements. Furthermore, we provide information on the system's end-to-end latency, which enables replication, comparison, and assessment of future experiments to be performed in this particular VR system.

2.3 REALIZATION OF LOW-LATENCY ENVIRONMENT

This section describes our hardware setup, provides an assessment of state-of-the-art techniques for building a low-latency VR environment for motor learning, and finally presents our design choices and developments for this particular task. Figure 2.1 depicts the architecture of our system. It consists of three major parts: (i) display technology, (ii) render engine and (iii) motion capturing system / motion preprocessing.

To display our virtual world, we use a CAVE environment. This ensures a *minimal level of disturbance* (R3), since the equipment attached to the user is limited to a pair of tracked 3D glasses. These glasses are usually much lighter and smaller than a full-sized HMD and there are no cables attached to the user. Moreover, the user is still able to see her own physical body and thus gets *feedback on her own motion* (R1) without any additional equipment. The still slightly narrow field of view of available HMDs impedes self monitoring by looking at one's own (virtually rendered) body: The user has to make larger head movements, especially when looking down along one's own body, which then may interfere with the training goals. In particular training situations, in which head and neck orientation and/or movements are essential, the additional weight imposed by the HMD also influences the trainee's posture compared to the optimal natural posture.

Our two-sided CAVE (L-Shape, 3 m $\times$ 2.3 m for each side) has a resolution of 2100 $\times$ 1600 pixels per side. Each side is driven by two projectors with INFITEC filters to enable passive stereoscopic vision by utilizing wavelength division. Both walls (floor and front) use back-projections. The four projectors are driven by a single computer (2 Intel Xeon CPU E5-2609 @2.4 GHz, 16 GB Ram, 2 Nvidia Quadro K5000 GPUs).

Our virtual world consists of the following components: a virtual fitness room with a virtual mirror mounted on the front wall. The user is placed in front of this mirror and her motions are mapped onto a generic avatar visible in the mirror. This effectively generates a virtual reflection of the user's motions, which further enhances the fulfillment of the requirement for *feedback on one's own motion* (R1). The user's motions are captured by an optical motion tracking system mounted at the top and the sides of the CAVE. Motion data is streamed into the motion preprocessor, which prepares the data for its use in the render engine and additional software packages for further analysis and feedback generation. The render engine then visualizes the scene while adapting the camera perspective(s) according to the user's head and animates the virtual character in the mirror using the full-body tracking data.

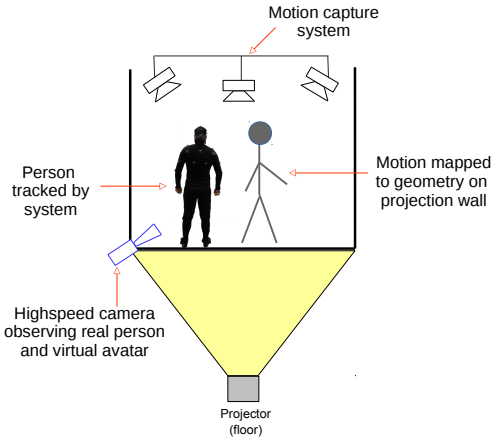


Figure 2.2: Latency measurement: Motion of the person inside the CAVE is tracked and directly mapped on the virtual character. A high-speed camera records both: real person and virtual character. Furthermore, it adds a time-stamp to the video. Later on, the number of milliseconds between the real person reaching a turning point and the virtual character reaching the turning point is determined.

In order to evaluate and compare the overall end-to-end latency of different rendering and tracking approaches, we adopted and extended a well-established latency measurement approach [FS14; LSG91; Ste08]: Typically, a pendulum is placed inside the tracking area, and the tracking data is visualized on a display behind the pendulum. A high speed camera records both the swinging real pendulum and the virtual pendulum on the screen. Afterwards, the recording is analyzed by hand, and the time-offset between the real and virtual pendulum is the end-to-end latency of the overall system. The following individual system latencies add up to the total latency: tracking latency, network latency, rendering latency, and display latency. The simple and periodic movement of a pendulum allows the application of automatic evaluation techniques [FS14]. However, in our case, we are not only interested in the end-to-end latency of a single marker tracked by the system, but in the latency induced by (more complex) full-body motion capture. Hence we replace the pendulum by a human standing in the center of the CAVE, who is fully tracked and instructed to move one arm up and down. The tracked motions are mapped onto the virtual character, which is rendered on the front screen (see Figure 2.2). The scene is again recorded by a high-speed camera (170 Hz), and the video is analyzed by hand (see video in supplementary material of [Wal+15]). To reduce errors due to manual labelling, we average latency results over 30 trials.

In the following we first discuss our rendering solution, before presenting the full-body motion capturing approach.

2.3.1 *Real-Time Rendering*

Stereoscopic visualization in a CAVE requires to render two images (left/right eye) for each projection wall (floor and front in our case). Thus, the rendering framework must be capable of rendering multiple views per frame while still keeping up to the stated requirements: We satisfy requirement R1 (*feedback on one's own motion*) by visualizing the movements of the participant through a virtual character in a virtual mirror. Requirement R2 (*low latency and high frame rate*) then mainly depends on the chosen hardware and software solution for real-time multi-view rendering.

System	Price / Camera	Camera Res.	Max. FPS	Used FPS	Latency	SD
Vicon T20	20.000 EUR	1600 × 1280	500 Hz	100 Hz 240 Hz 500 Hz	54.9 ms 44.7 ms 38 ms	13.18 ms 10.6 ms 8.4 ms
OptiTrack Prime 13W	2.500 EUR	1280 × 1024	240 Hz	100 Hz 240 Hz	59.7 ms 41 ms	12.3 ms 9.9 ms
OptiTrack Flex 100	600 EUR	640 × 480	100 Hz	100 Hz	65.5 ms	21 ms
Microsoft Kinect 2	150 EUR	512 × 424	30 Hz	30 Hz	98.8 ms	19.17 ms

Table 2.1: Comparison of end-to-end latencies of the different motion capturing systems (averaged over 30 measurements), also listing price per camera, camera resolution, as well as the maximum and the employed frame rates.

In terms of hardware, we identified that multi-pipe rendering on a single computer reduces the latency as compared to distributed rendering [Wal+15]. Concerning the software, we identified a custom-developed render engine as being able to best suit our requirements as such a self-developed approach allows to fully control data storage and data flow in order to minimize latency [Wal+15]. We minimize computational cost as we offload expensive computations to the available GPUs. Further, we animate the virtual character using the very efficient and simple linear blend skinning [Jac+14], performed on the GPU. The virtual mirror is implemented by first rendering the scene, including the animated character, from the mirrored perspective of the user. The content of the resulting framebuffer is then mapped as a texture onto the mirror geometry in the scene (see Figure 2.6). To animate the character, we stream the pre-processed motion data from the motion preprocessor to the render engine using a network interface (compare Figure 2.1). The data is received asynchronously and is then directly used to update the posture of the character. This updates the transformation matrices in CPU memory, and a simple version counter approach is used to keep the data on the GPUs up-to-date. In terms of lighting we use the simple Phong lighting model, and we apply shadow mapping to the character and other objects in the scene (see Figure 2.6). The resulting render engine provides all necessary features for our VR motor learning environment, while maintaining a slim software design and flexibility. Further details on the realization of the rendering can be found in [Wal+15].

2.3.2 *Motion Capture*

Full-body motion capture is necessary to provide real-time augmented feedback on motor performance. In the following, we give an overview of state-of-the-art motion tracking approaches and assess them with respect to the aforementioned requirements.

Evaluation of Existing Approaches

To track full-body motion, the distinction between *outside-in* and *inside-out* approaches is important. For outside-in approaches the actual capturing devices are placed at fixed positions outside the tracking area. The inside-out approach works the other way around, for instance by attaching inertial trackers or cameras to the user (e.g., the Lighthouse tracking system shipped with the HTC Vive). Although the outside-in approach has to deal with occlusions, it has the important advantage that no sensitive and/or heavy devices have to be attached to the user. Furthermore, outside-in approaches do not suffer from drift due to time-integration of sensor data, and they provide the exact location of the user. Since we want to avoid attaching disturbing hardware on the user (R3) we only take outside-in approaches into account in the following.

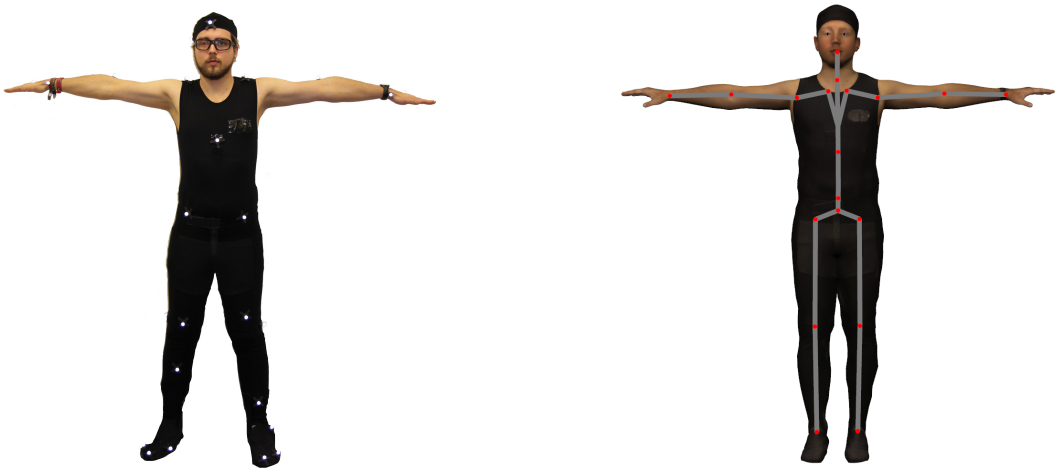
For these systems, the next distinction is between *marker-based* and *marker-less* approaches. Many commercially available systems exist for both approaches, such as the marker-based systems Vicon and OptiTrack, or the marker-less systems Microsoft Kinect and Organic Motion. The advantage of marker-less systems seems obvious: No hardware has to be attached to the participants, thereby reducing setup time significantly and minimizing user disturbance. However, as already pointed out in requirement R2, a low latency is crucial. This criterion is not yet satisfied by state-of-the-art approaches in marker-less motion capture such as [Cao+17; Mat+18; Meh+17a; Meh+17b; Wei+16]. For instance, OpenPose, as one of the top approaches [Cao+17; Wei+16], estimates 3D joint data with framerates of less than 4 fps in a multi-camera setup [1]. Even faster approaches, as for instance the one presented by Mehta et al., reach frame rates of not more than 30 fps only for the image processing part. In addition, this approach suffers from noisy joint estimations [Meh+17b]. Further, marker-less systems often depend more on lighting conditions and a good view of the participant. Consequently, we decide to focus on fast accurate marker-based system. Nevertheless, we also analyzed the marker-less Kinect sensor, since this device can also operate in rather dark environments and is used in many related approaches towards motor performance training (e.g., [RBK13; Sme+14]).

For the marker-based systems, one can use *active* or *passive* markers. Passive markers simply reflect the infrared light emitted by the tracking cameras. The tracking system captures a set of markers, which then have to be consistently labeled. As soon as markers get lost and re-appear later on, the labeling step can produce errors. Active markers, as used in systems like PhaseSpace[2] avoid the labeling problem by emitting light at a unique frequency. The disadvantage of active markers is that they require more service and are more prone to get damaged during experiments. Furthermore, the marker suits are more difficult to clean. Additionally, active motion capture suits are often less comfortable to wear than suits for passive markers (R3).

Thus we decided to focus on outside-in tracking systems based on passive markers. We analyzed and compared the Vicon T20 system and the OptiTrack systems Flex 100 and Prime 13W. These systems require a motion capture suit with attached markers,

1 These values are reported in the repository of the project (https://github.com/CMU-Perceptual-Computing-Lab/openpose, last visited: 2018-11-12)

2 http://www.phasespace.com



(a) Marker placement. (b) Skeleton representation.

Figure 2.3: Marker setup and reconstructed skeleton representation.

or having the markers attached directly to the human skin. As motion capture suit any tightly fitting sports clothing can be used as long as it does not contain reflective materials. Thus these systems satisfy requirement R3. We evaluate the end-to-end latency and update rate (R2) of the different tracking systems using the latency measurement approach described in Section 2.3. In order to focus on the tracking latency, we only rendered a simple stick figure (at about 280 fps). Table 2.1 summarizes the resulting latencies for Vicon T20, OptiTrack Prime 13W, OptiTrack Flex 100, and Microsoft Kinect.

Concerning the robustness of the tracking (R4), the marker-based systems meet our demands: For most basic movements and exercises (e.g., squats, walking around, jumping), the user is tracked without the need for re-calibration or returning to the T-Pose during a session. In contrast, the tracking robustness for the Kinect camera was worse: Here, many kinds of exercises, e.g. squats, cannot be tracked reliably due to occluded body parts.

Realization

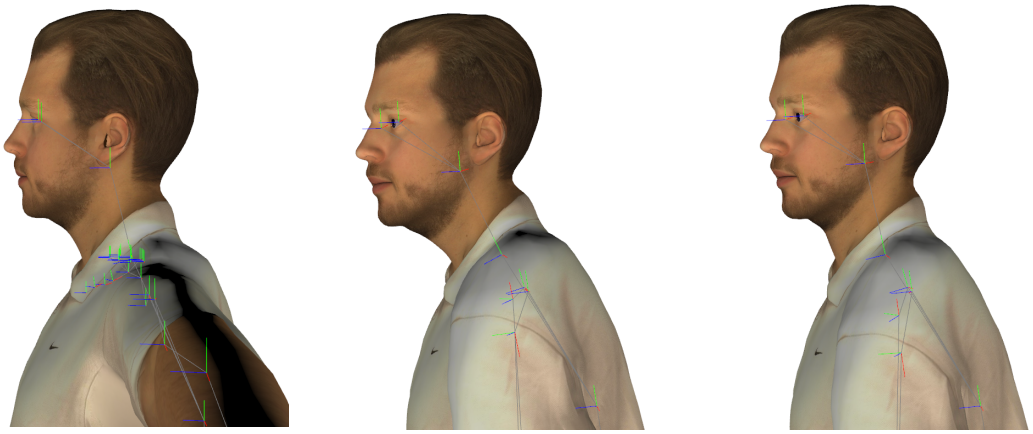
Based on the benchmark results shown in Table 2.1, we decided to use an OptiTrack Prime 13W system with 10 cameras. This marker-based solution is a good compromise as long as there is no marker-less option of similar performance and robustness. The Microsoft Kinect was excluded due to its high latency (R2) and problems in dealing with occluded body parts (R4). The Vicon cameras' advantage in terms of temporal and spatial resolution did not justify the much higher price for our field of application. We decided to use the Prime 13W system instead of the Flex 100 cameras because of the wider field of view (82° vs. 58°) and the higher temporal and spatial resolution.

The cameras are arranged in a way that allows an almost failure-free tracking. Participants are equipped with a marker suit of up to 44 markers for accurate skeleton tracking (see Figure 2.3). We use a self-designed marker suit that is a combination of the trousers provided by OptiTrack and a sleeveless sports top equipped with velcro to attach the markers. The markers on the arms and the hands are attached

directly to the skin to prevent noise which can be induced due to slipping markers. Typically, we do not cover the arms, as our setup is designed for performing sports movements which tend to induce sweating. This would be especially uncomfortable when wearing sleeves and non-sports clothing. Our marker layout is based on the 41-marker layout specified by OptiTrack. Three additional markers on the back can be added to this marker setup in order to capture the bending of the spine in more detail. The system provides joint positions and rotations, which are used to animate the virtual mirror character, for feature extraction (e.g., calculation of movement direction, speed, acceleration), and for motion analysis. The joint angles denote the rotation of a joint with respect to its parent. To animate the virtual character, we map the translation of the root joint (the hips) and the joints angles obtained from OptiTrack on the virtual character. To this end, we use a mapping table between the joint names provided by OptiTrack and the joint we use for our virtual characters, which are named according to the H-Anim standard [ISO05]. When using the 41-marker setup, we use 19 joints to animate the character (see Figure 2.3b), when using the extended one with 44 markers, we use 21 joints. Many motion capture systems provide rotations for each joint with respect to its parent and with respect to a rest pose. Typically, virtual characters that shall be animated can have different rest poses. In these cases, it is required to determine the offset from the rest posture of the motion capture system to the rest posture of the virtual character. The inverse of this offset can then be applied to the motion capture data. The OptiTrack system uses the T-Pose as rest pose. The characters that we animate are typically also placed in T-Pose (cf. Figure 2.3b). Consequently, the joint rotations obtained from OptiTrack could be directly mapped to the virtual character. However, in some cases, characters do not stand perfectly in the T-Pose. For instance, the character displayed in Figure 2.4a is obtained from a 3D scan and has an slightly arched neck. If the person with the slightly arched neck that has been used to create the virtual character is now tracked in the motion capture environment and the observed data is mapped to the character, without any preprocessing, the neck is arched even more (see Figure 2.4b). Such a behavior is typically undesired. Consequently we perform a correction of the motion capture data with respect to the target character per default. For correction, we first determine the offset from the rest pose of the motion capture system to the rest pose of the character in a preprocessing step. This is performed for each joint. Next, when animating the character, we multiply the inverse of this offset to each joint rotation as obtained from the motion capture system. An alternative would be to place the character directly in the rest pose of the motion capture system. However, this would complicate the animation of the character if another motion capture system should be used. In addition to this preprocessing of the motion capture data, we manipulate the root translation such that the animated character does neither penetrate the floor nor flies due to a mismatch in limb lengths between tracked subject and rendered avatar.

2.4 BENCHMARK

To evaluate the influence of rendering options on latency, we evaluated different quality levels to find the best trade-off between quality and performance. The same



(a) Rest pose of virtual character. (b) Uncorrected posture mapped on character. (c) Posture after motion correction.

Figure 2.4: Correction of the motion data to match the actual posture. The character has a slightly arched neck by default (a). The motion capture system also tracks a sligthly arched neck. This data is then mapped on the character leading to an overarched neck (b). In (c) our mechanism for posture correction is applied.

Render Quality	**Fps**	**Latency**	**Std. Dev.**
Stick figure	690	36 ms	9 ms
Low resolution	114	54 ms	9 ms
Low resolution + Shadows	88	60 ms	10 ms
High resolution	86	62 ms	12 ms
High resolution + Shadows	62	81 ms	14 ms

Table 2.2: Latency and performance values for different rendering qualities (mean value of 30 tries and standard deviation). Rendering a minimalist stick figure without a virtual environment, or rendering the full gym scene and the virtual mirror, but using either a low-resolution (20 k triangles) or high-resolution (135 k triangles) virtual character, with optional shadow mapping.

Figure 2.5: Example frames from one of the latency test videos (highest quality character with shadows). The left image shows the user's arm approaching the lowest point. The image in the middle shows the turning point of the real arm, the picture on the right shows the turning point of the virtual arm, while the real arm already moves upwards.

measurement procedure as described in Section 2.3 was used for 30 trials, and we report mean latency values and standard deviations. The virtual scene used for the tests consists of a virtual fitness studio (about 100 k triangles) including the virtual mirror (see Figure 2.6).

The results are listed in Table 2.2. For our high-resolution character (135 k triangles), we observed a latency of 81 ms at 62 fps when using shadow mapping. Without shadows, a latency of 62 ms at 86 fps was measured. Using the low-resolution character (20 k triangles) reduces the latency to 60 ms at 88 fps (with shadows) and 54 ms at 114 fps (without shadows). As a baseline test, we also rendered a simple stick figure without the surrounding fitness studio, which resulted in a latency of 36 ms at 690 fps. We conducted an additional test which consists of a single marker attached to a pendulum instead of a tracked human. The pendulum was visualized as a box inside the CAVE. Here, we observed a latency of 32 ms (SD=9 ms).

Figure 2.6: The visual quality achieved in the final system, including artificial shadows for the trainee.

Our end-to-end latency consists of the individual latencies of the cameras (approx. 4 ms according to manufacturer), of the tracking software, the motion preprocessing (approx. 2 ms), the network communication (approx. 1 ms), as well as rendering, synchronization, and display hardware (approx. 19 ms according to manufacturer). Figure 2.5 shows exemplary frames of the recording filmed by the high-speed camera, showing the experiment using the high resolution character and real-time shadows.

Figure 2.6 shows a photograph of a user reflected in the virtual mirror inside our environment. Please also see the video in the supplementary material of [Wal+15], which shows a user interacting with the system using the low-resolution character. The second part gives a glance on the latency measurement procedure recorded using a high resolution character with shadows.

2.5 DATA RECORDING

Environments for motor learning require data to obtain information on how movements are conducted correctly and to train and evaluate data-driven algorithms towards motion processing. This section describes that data that was recorded for this thesis. Motion data was obtained using the same OptiTrack motion capture system (10 Prime 13W cameras) as introduced in Chapter 2.3.2. The usage of such a marker-based system, which is a well evaluated standard procedure in biomechanical analysis,

allows us to obtain highly precise motion capture data, that also covers fine-grained errors and variations in motor performances. In this thesis, mainly data sets for two motor actions, namely squats and Tai Chi pushes, are used. The squat data consists of $N = 96$ squats from 50 subjects. The Tai Chi push data consists of $N = 120$ Tai Chi pushes from 24 subjects.

The recorded data can be described as follows. Each repetition of a motor action is represented as a sequence of single postures (frames), called *trajectory*, of features which describe the movement of the human body. The motion capture system outputs kinematic features for $k = 19$ joints based on 41 markers (see Figure 2.3) per frame at 120 Hz. Each frame consists of k joint rotations as well as k joint positions. Joint rotations are represented as quaternions $\mathbf{q}_1, \ldots, \mathbf{q}_k$. Each quaternion denotes the rotation of a joint with respect to its parent. The root rotation $\mathbf{q}_1$ describes the rotation of the root with respect to its rotation at the beginning of the movement. As root joint we use the hips. The joint positions are represented by vectors $\mathbf{t}_1, \ldots, \mathbf{t}_k \in \mathbb{R}^3$. Each $\mathbf{t}$ denotes the translation relative to the position of the root joint at the beginning of the movement. Further, we use joint angles as Euler angles, calculated from the quaternion representation, which correspond to flexion/extension, abduction/adduction and twist of the corresponding joint. The data set, together with the annotations that we describe in the following chapters, is publicly available via http://doi.org/10.4119/unibi/2930611.

2.6 CONCLUSION

First, we developed and motivated requirements for VR motor learning. We examined state-of-the-art techniques and technologies for motion capturing and rendering with respect to these requirements, and propose a low-latency environment based on the most promising components or approaches. In terms of rendering, a single-PC multi-pipe approach was shown to achieve a lower latency than even a minimal render cluster using two nodes. Our slim custom-designed render engine maps all expensive computations to the GPUs and parallelizes well. For full-body motion capture, we decided to use the marker-based outside-in OptiTrack system. The resulting system provides a virtual environment with a mirror and a high quality character, and it can serve as a solid base for further developments and experiments in VR motor learning.

Using the 20k-triangle character, our system meets the stated requirements: The user is able to monitor his own motion in the virtual mirror (R1). The overall latency of the system is at around 60 ms, which is comparable or better then related systems (R2). The graphics engine runs at 88 fps, feeding four channels with 2100×1600 pixels each, which is sufficient to perceive smooth images. Requirement R3 is also satisfied as users only have to wear passive stereo glasses and tight clothing with attached markers. Of course marker-less motion tracking would be the ideal solution, but to our experience the available solutions are not fast or robust enough in a CAVE environment. Requirement R4 can also be considered as satisfied, as shown in the accompanying video published in [Wal+15].

This chapter gives guidelines on how to develop a VR environment usable for motor learning experiments. Inherent variables of our system as well as possible

alternative approaches have been evaluated and compared. This information should support reproducibility and increase comparability of experiments.

Our proposed system lacks portability, since the display technology as well as the motion tracking system are fixed installations. A portable system could be achieved by using components like a commodity depth sensor (e.g., Kinect) or inertial trackers for motion tracking and a HMD for visualization. However, any configuration of that sort will have the problems presented here. Still, developing a portable system for motor learning that can be used at home or in a small clinic is an interesting challenge. In this context, it is to be evaluated how the more obtrusive display hardware (HMDs) influences participants' performance of motor actions and their ability of motor learning. A first attempt that goes into this direction is presented in Chapter 7.

Another direction of future research is motivated by the usability of the virtual environment. To attach motion capture markers to subjects is time-consuming. Recent approaches from pattern recognition and computer vision are able to extract the human posture from video images. Developing approaches towards accurate marker-less and low-latency motion capture is promising to advance the field of sports coaching in virtual environments.

The environment proposed here will serve as the basic environment for all steps conducted in the following. In the next chapter, we demonstrate that our environment can be used to help novices to adapt their motor performance. To this end, we conduct an experiment motivated from sports science. We will evaluate whether participants are able to adapt their own performance towards a skilled movement that is superimposed on their own movement in the virtual mirror.

3 IMPROVEMENT VIA OBSERVATION: SUPERIMPOSED SKILLED PERFORMANCE

Published in:
[Hül+19]

Feedback is essential for skill acquisition as it helps identifying and correcting performance errors. Concurrent types of feedback have shown to be especially beneficial for novices. Moreover, watching skilled performance helps novices to acquire a motor skill, and this effect depends on the perspective taken by the observer. To date, however, the impact of watching one's own performance together with full-body superimposition of skilled performance, either from the front or from the side, remains to be explored. Here we used the VR setup developed in Chapter 2 and we asked novices to perform squat movements in front of a virtual mirror. Participants were assigned to one of three concurrent visual feedback groups: participants either watched their own avatar performing full-body movements or were presented with the movement of a skilled individual superimposed on their own performance, either from a frontal or from a side view. Motor performance and cognitive representation were measured in order to track changes in movement quality as well as motor memory across time. Consistent with our hypotheses, results showed an advantage of the groups that observed their own avatar performing the squat movements together with the superimposed skilled performance for some of the investigated parameters, depending on perspective. Specifically, for the deepest point of the squat, participants that watched from the front adapted their height, while those that watched from the side adapted their backward movement. In a control experiment, we ruled out that the observed improvements were due to the mere fact of performing the squat movements per se—irrespective of the type of visual feedback. The present findings indicate that it can be beneficial for novices to watch oneself together with a skilled performance during practice, and that improvement depends on the perspective chosen. Further, we show that the environment proposed in Chapter 2 can be applied in the context of motor learning in VR.

My Contribution *The work presented here was done in close collaboration with Irene Senna and Cornelia Frank. I designed, conducted and evaluated the experiment together with both of them. Further, I contributed by implementing the experimental setup. The statistical analysis of the results concerning the motor performance was done by Irene Senna. The analysis concerning the cognitive representation was done by Cornelia Frank.*

3.1 RELATED APPROACHES

Feedback is essential for skill acquisition as it delivers performance-related information and can help to identify potential errors and to implement corrections needed for performance improvement [MA12; Mag01]. While task-intrinsic feedback relates to information available as a result of task execution, augmented feedback is used to convey any kind of extra information in addition to task-intrinsic feedback. In sport

settings, for instance, when it comes to learning a new motor skill that requires the execution of complex full-body movements, looking at a mirror offers visual feedback or receiving instructions from a coach offers verbal feedback. So far, augmented feedback has proven to speed up the learning process and to help acquire a skill (for reviews, see [HF04; MA12; Mag01; Sig+13]. Nowadays, VR can be used as a tool for guiding and boosting motor learning: indeed, a virtual environment offers the opportunity to introduce innovative types of augmented feedback that exceed real world opportunities (e.g., [Chu+03; Sig+13; Sig+15; TSB97]). For instance, in the real world the learner can compare her own performance as seen in a mirror to the coaches' demonstration of an optimal performance. To do so, the learner must map the own performance to the target performance. This requires some cognitive effort: the learner has to switch between looking at herself in the mirror and looking at the coach, while trying to infer what might be wrong with the movement during its execution. Instead, in VR this effort can be reduced by showing the target performance superimposed on the learner's performance during execution. Here we developed a virtual-reality system for the learning and coaching of full-body movements, and we investigated different kinds of real-time augmented visual feedback to foster learning of correct squat movements. Accordingly, the purpose of the present study was to provide online visual feedback through a virtual mirror and to examine the influence of a skilled performance superimposed on the learner's performance on motor performance and learning. In particular, we aimed to explore whether the novice participants would tend to spontaneously adjust their movements in order to match them with the correct ones, and whether this would be more effective than watching one's own performance alone. To this end, we mapped the participant's performance to a virtual avatar and showed this performance in a virtual mirror during the execution of a squat movement. At the same time, we showed the performance of a skilled individual mapped onto a second virtual character, superimposed over the participant's avatar. The different feedback was delivered from different points of view for the participants (i.e., from the front or the side). We investigated the effectiveness of these different kinds of visual feedback on motor performance and cognitive representation of the squat.

The impact of augmented feedback on motor learning is highly dependent on the characteristics of the feedback provided, such as the timing of information. Visual feedback can either be provided as terminal feedback after task execution, such as video replays, or as concurrent feedback during task execution, such as a mirror image. Particularly for novices, being new to a skill, concurrent visual feedback has shown to be effective, as it guides the learner whilst executing the motor action [MBW07; Sig+13; SSW84; TSB97]. Along these lines, any task that is novel to the learner is a challenge [GL04], and thus concurrent visual feedback is appropriate for novices who do not yet have a representation of the skill in an early phase of learning [FKS18; FLS13; HF02; HF04]. Consequently, we chose to provide concurrent feedback rather than terminal feedback in the present study.

Apart from the timing used to provide visual feedback, the content of the visual feedback (i.e., what exactly is shown to participants) is of high relevance [MA12]. From research on observation and modeling (for reviews, see [ARS14; MLS12], the type of model (defined as an example to imitate) shown during practice has proven to

be a critical variable for motor learning [AP14; MBZ76]. Specifically, it has been shown that watching successful performance promotes motor learning [MBZ76]. Moreover, mixing successful performance as provided by an expert model and unsuccessful performance as provided by a novice has proven to be extremely effective for motor learning [AP13; AP14; RP11]. For instance, Andrieux et al. found that watching both a novice and an expert model in an alternate fashion favors motor learning as compared to watching either type of model alone [AP14]. This combination of skilled and unskilled performance can help novices to combine descriptive and prescriptive knowledge of performance and thus information on movement quality of what is and what should be. Thus, in order to assist a novice with learning a motor skill, concurrent visual feedback together by providing both information on one's own movement together with a skilled performance might be most effective for motor skill acquisition. Consequently, we chose to present two virtual characters at the same time during the acquisition of the squat.

To date, several studies have used VR to investigate the influence of observing one's own and/ or a skilled performance on subsequent motor performance and motor learning [And+13; Bur+11; Chu+03; COM14; Hoa+16; Sig+15; Tan+15; TSB97]. In these studies, the skilled performance is either visualized as an overlay on top of the participant's movement (e.g., [Sig+15]) or visualized on a virtual character next to the participant (e.g., [Chu+03]). Sigrist et al. examined concurrent visual feedback in a VR-based rowing simulator, comparing visual feedback to different types of multimodal feedback [Sig+15]. In their visual feedback condition, the target movement of the oar was visualized as an overlay on top of the participant's oar. Depending on the deviation from the target, the transparency of the target oar was manipulated. This feedback was complemented by a trace of the subject's trajectory when the error became too large. The authors observed improvements in spatial error as well as in temporal error for all conditions, including unimodal visual feedback. However, the authors showed a movement implying the skilled use of a tool (i.e., the oar) involving only one body part (i.e., the arm) superimposed on the participant's performance together with additional information (i.e., trace visualization, changes in opacity) in the visual feedback condition. Given this design, the mere effect of the superimposition cannot be interpreted from their findings, and whether this generalizes to full-body movement remains unclear. With regard to full-body movement, Chua et al. investigated the impact of several visual feedback strategies on Tai Chi performance, two of which entailed the concurrent superimposition of a virtual character executing a skilled performance [Chu+03]. In their study, superimposing the virtual teacher on the participant's virtual body did not lead to any effect. Thus, whether superimposing a skilled performance on that of the participant's virtual body is beneficial to motor learning during the execution of full-body movements is still unclear. To the best of our knowledge, among the few studies that focus on full-body movements (e.g., [Bur+11; Chu+03; Hoa+16]), no systematic investigation of this feedback strategy on motor learning exists that allows to determine the mere effect of a superimposed skilled performance. Nonetheless, most sports require the execution of full-body movements. Consequently, we chose a full-body movement (i.e., the squat) to investigate the influence of superimposing a skilled performance

on one's own performance. We did this in comparison to watching oneself only, by including a no-superimposition control group.

A further factor that is important in motor learning is the participant's viewing perspective, as it determines which perceptual information can be picked up by the observer for subsequent action execution [SN85]. For many exercises, the crucial aspects of the movement cannot be well observed from a first-person perspective. For instance, common errors while practicing squats, involve wrong weight distribution or bending the back in a wrong way. In a real environment, such as a gym, a person can have a side perspective of the movement from a mirror only when turning the head, which would imply a wrong posture for the squat. As opposed to the real world, virtual environments allow for changes in perspective [COM14; Hoa+16; Sal+10]. For instance, while in the real-world participants watch themselves in a mirror looking at their own performance from a natural perspective, artificial rotations in VR allow for different perspectives, such as watching oneself from the side whilst standing frontal to the mirror. To the best of our knowledge, while some studies investigate different perspectives (e.g., [COM14; Sal+10]), and even though combinations of different perspectives with overlays exist (e.g., [Hoa+16]), there is no investigation of varying perspectives together with full-body superimposition of skilled performance. Consequently, we chose to examine the influence of perspective whilst practicing together with a skilled performance superimposed on one's own performance. Specifically, the superimposition is, in an additional condition of our experiment, enriched by a rotated perspective in the virtual mirror: Participants performing in front of a virtual mirror observe their own movement together with the skilled performance from the side. Such rotation of the image might offer the advantage of making it possible to watch body parts that are crucial in the execution of the squat, and that are not visible from a frontal (and natural) point of view, allowing for an easier error correction. On the other hand, the rotation in perspective might interfere with the performance, by requiring the subjects to perform a mental rotation of the image, which might have a detrimental effect in sensorimotor learning, instead of facilitating it.

In addition to measuring learning by means of motor performance, we measured the underlying cognitive representations to assess changes in motor memory as reflected by modifications in representation structures. According to the cognitive action architecture approach (CAA-A; for an overview, see [Sch04; SM06]), motor actions are hierarchically organized across cognitive and motor levels and are represented in memory as well-integrated representational networks. These cognitive representations of motor actions are formed by units compiled of body postures/movement components and associated sensory consequences, known as basic action concepts (BACs; see [Sch12]) that are encoded in long-term memory and guide motor skill execution [Lan+13; SM06]. Learning, according to the CAA-A, is reflected by modifications in the relations and the groupings of BACs and the respective representation structure, and thus by functional changes in representational networks of complex action in long-term memory (e.g., [Sch03; Sch04; SR13]. Together with performance improvements, novices' representations have been shown to become functionally more organized following physical practice [FLS13] and mental types of practice, such as motor imagery [Fra+14] and action observation [FKS18; KFS17]. The impact

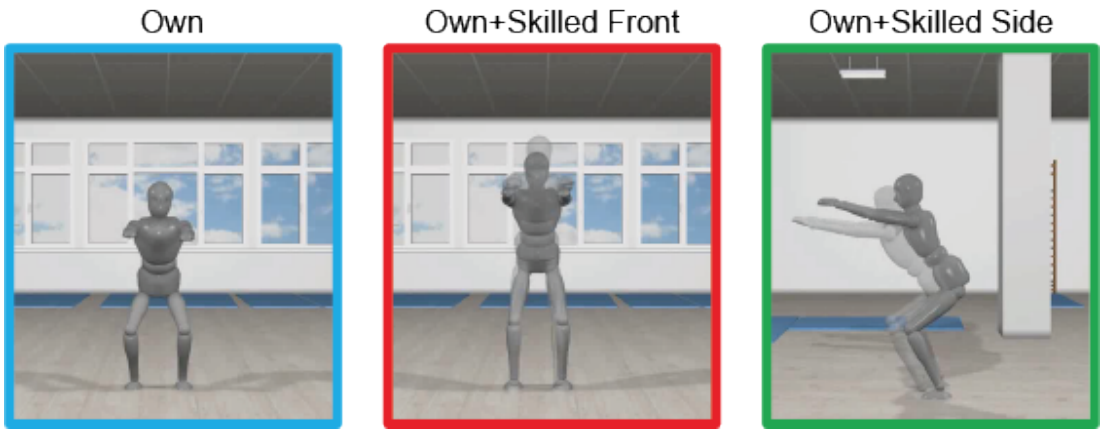


Figure 3.1: Conditions. During acquisition, participants were provided with different visual feedback: one group of participants observed only the own avatar. A second group observed the skilled performance superimposed onto the own avatar from a frontal perspective. A third group watched the skilled performance superimposed over their own avatar from a side view.

of VR-based augmented feedback on the development of cognitive representations, however, remains to be explored. Consequently, we chose to measure participants' cognitive representations, in addition to motor performance, to learn about the impact of different types of concurrent visual feedback on the formation of representational networks in motor memory.

To summarize, the purpose of the present study was to investigate the influence of superimposing a skilled performance on one's own performance on *motor performance* and on the development of *cognitive representations* in long-term memory. Moreover, we also investigated the impact of the different kinds of feedback on *subjective judgments* (i.e., the experience participants had with the virtual characters), by means of questionnaires. This was realized using the environment proposed in Chapter 2. Specifically, concurrent visual feedback was provided such that participants watched their own performance of a full-body movement in front of a virtual mirror, or with a superimposed skilled performance, either from the front or from the side (cf. Figure 3.1). We hypothesized that superimposing a skilled performance would lead to better motor performance and more developed cognitive representations compared to watching one's own performance alone. Furthermore, we expected an influence of perspective on these two variables.

3.2 EXPERIMENT 1

3.2.1 *Materials and Methods*

Three groups of novices (between-subject design) performed squats inside a virtual environment while obtaining, depending on the experimental condition, concurrent visual feedback on their motor performance and their cognitive representation. Concurrent visual feedback was provided such that participants watched their own

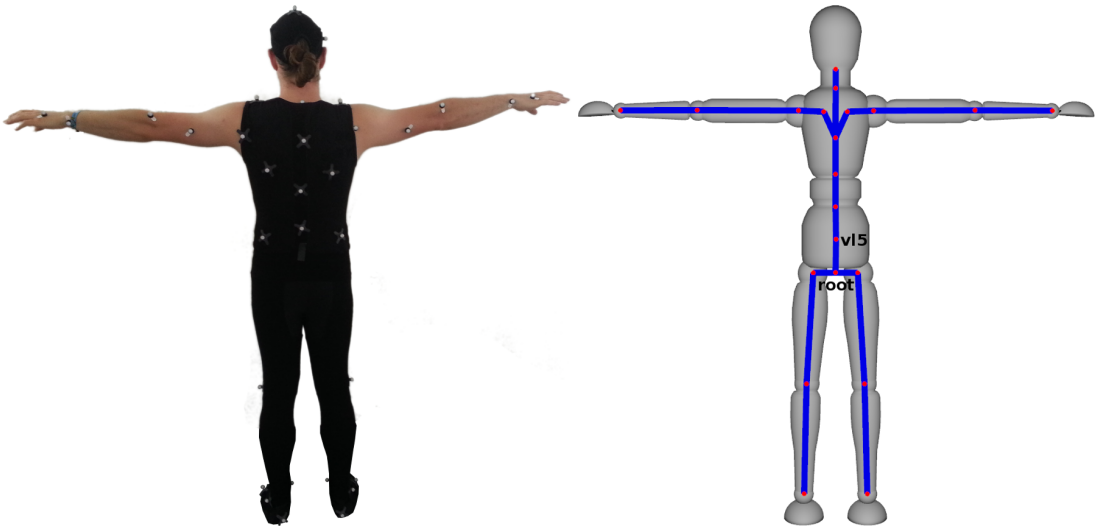


Figure 3.2: Marker setup and reconstructed skeleton representation. Joints that are specifically used for the kinematic analysis of this experiment are named.

performance of a full-body movement in front of a virtual mirror (Own), or with a superimposed skilled performance, either from the front (Own+skilledFront) or from the side (Own+skilledSide). We investigated the impact of the different kinds of feedback on motor performance, cognitive representation, and subjective judgments.

Participants

Thirty-five naïve participants (21 males, mean age $M = 26.3$, standard deviation $SD = 4.4$) took part in the study. Four further participants were tested, but their data were not included in the analyses due to technical issues during the experimental session. All participants were novices with respect to the squat movement: They had never attended a professional training of the exercise before and had never trained the squat on a regular basis. Further, they did not have any theoretical information on how to execute a correct performance. All participants were taller than 1.6 m and spoke German fluently. Participants provided written informed consent and got paid 6 euros per hour for their participation. The study was conducted in accordance with the Declaration of Helsinki and had ethical approval from the ethics committee of Bielefeld University.

Apparatus

We used the system that is described in Chapter 2 for the experiment. For motion tracking, we used a marker setup based on 44 markers to reconstruct the movement of 21 joints (see Figure 3.2). Depending on the phase of the experiment, the virtual room the participants are placed in was equipped with either a black plane or a virtual mirror in front of the participant (Figure 3.1). If the mirror was shown, it reflected the virtual room as well as a virtual avatar of the participant. This avatar had the appearance of a wooden stick figure with a per-limb scaling according to the participant's limb lengths. The avatar was animated in real time using the information from the motion capture

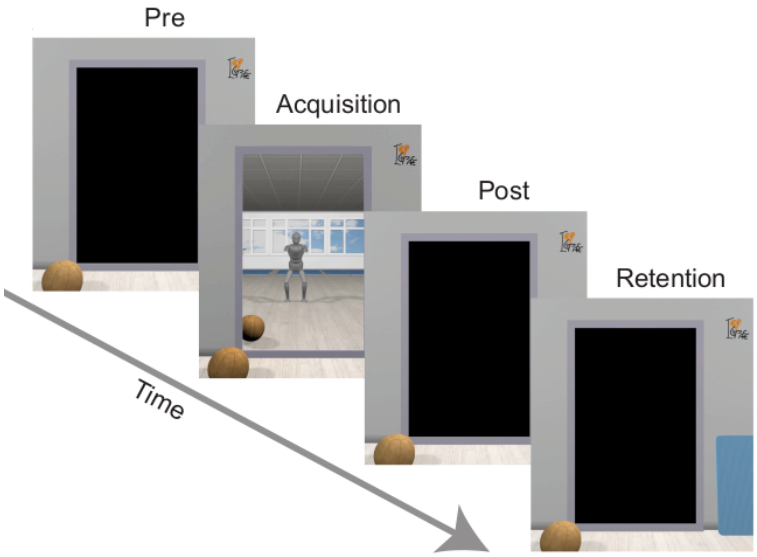


Figure 3.3: Procedure. The experiment consisted of different phases, and participants were asked to perform squat movements.

system. The rendering engine ran at around 88 fps. According to the measurements from Chapter 2, the latency of the setup is approx. 60 ms. An experiment that involves the same experimental setting and a similar task showed that participants maintain a high level of perceived simultaneity for such a low latency [Wal+16].

3.2.2 *Procedure*

The experiment consists of four phases (cf. Figure 3.3): pre-test, acquisition, post-test and retention-test. Pre-test, acquisition phase as well as the post-test took place on the first day and lasted approximately two hours. The retention-test took place on the day after and lasted around one hour. Participants were assigned to one of three groups: Own ($n = 12$), Own+skilledFront ($n = 11$), Own+skilledSide ($n = 12$), which differed in the content of concurrent visual feedback provided in the acquisition phase (see 'Acquisition phase' section and Figure 3.1).

Pre-test First, we handed out the main instructions for the overall experiment as well as a consent form. In the next step, participants filled in questionnaires for demographic data and simulator sickness [Ken+93]. Then, we equipped participants with 3D glasses and asked them to enter the CAVE and to stand on a marker on the floor of the virtual room. Participants were instructed to carefully observe a virtual character performing a skilled squat twice. The skilled squat was a recording of a skilled athlete (8 years of experience in practicing the squat for 2-3 times per week). Participants were asked not to move while watching the prerecorded performance. Next, the participants left the CAVE and performed a splitting task to measure their cognitive representation of the squat movement in long-term memory (structural dimensional analysis of mental representations; SDA-M, see [Sch12]; for details, see

Chapter 3.2.3). First, participants were introduced to the splitting task and the general setup. In order to ensure comprehension of the concepts, each was explained to the participants in random order. After having received general instructions on how to complete the splitting task, the splitting task (i.e., first step of the SDA-M) proceeded as follows: while one BAC is permanently shown on a screen (i.e., the anchor), the remaining concepts are presented one after another in randomized order. For each of the concepts being displayed together with the anchor, participants were asked to decide whether the two concepts would relate to one another during movement execution or not. Once the participants have finished a list of concepts, another concept took the anchor position and the procedure continued. This procedure resulted in 240 yes/ no decisions overall used as a basis for further structural dimensional analysis of mental representations (see data analysis section). Once each BAC had been compared to the remaining ones, the splitting task was completed. After completion of the splitting task, participants put on a motion capture suit and were equipped with motion capture markers. Next, participants were instructed to perform a single squat. We used this performance to instruct participants to reach approximately the desired depth (around 100 degrees). This step aimed at preventing them from performing the movement too deep, which would have put too much strain on their knees. In the next step, participants were equipped again with the 3D glasses and were instructed to orient themselves towards the disabled black mirror on the front wall of the CAVE while standing on the marked position on the floor. Again, a virtual character demonstrated the prerecorded skilled squat for two consecutive times, and after that disappeared from the screen. Then, the participants' initial squat performance was recorded. We instructed them to perform the movement as similarly as possible to the recording of the skilled person they had previously seen with respect to body postures and temporal aspects. Participants themselves started the recording procedure by performing a T-Pose. Then, they had to perform ten repetitions of a single squat in two sets of five repetitions each. Instructions on when to start the squat together with countdowns (from 5 to 0, with 0 representing the go-signal) were presented in textual form on the disabled black mirror in front of the user.

Acquisition phase In order to familiarize with the environment, participants were asked to move freely for 45s in in the center of the CAVE, while watching their own avatar in the virtual mirror. After familiarization, participants performed 6 sets of 5 squats each. Depending on their experimental condition, they performed the motor task under different concurrent visual feedback conditions (Figure 3.1). Participants in condition Own observed their own avatar in the mirror during the squat. Those in condition Own+skilledFront observed their own avatar as in the Own condition together with a second virtual character superimposed on their own. The second character performed the skilled performance as demonstrated before the pretest and was scaled in the same way as the participant's avatar and displayed slightly transparent. Participants in condition Own+skilledSide observed the same scene as the ones in condition Own+skilledFront, but with the mirror image rotated by 90 degrees around the vertical axis. Thus, they saw their own performance as well as the skilled performance from the side. The participants in conditions Own+skilledFront and

Own+skilledSide were informed that they were going to observe the movement of the skilled person. All participants were again instructed to perform the exercise as similarly as possible to the performance of the skilled person as shown during pre-test.

Post-test The procedure in the post-test was the same as in the pre-test. Afterwards, questionnaires on simulator sickness and about participants' experience in the virtual environment were filled out. Participants in conditions Own+skilledFront and Own+skilledSide were asked to answer questions related to the avatar twice, once for their own avatar and once for the virtual character that was used to display the skilled performance. Finally, the experimenter removed the markers and participants pulled off the motion capture suit.

Retention-test The procedure in the retention-test was the same as in the pre- and post-test. The retention-test took place one day after. First, participants put on the marker suit and markers were attached again. We used photos of the subject as well as the calibration data inside the motion capture system from the day before to verify the positioning of the markers. After having performed 10 squats, participants put off the marker suit and performed the splitting task again in order to measure their final cognitive representation of the squat.

3.2.3 *Data Analysis*

Motor Performance

Motor performance was measured using motion capture data. Based on these data, we focused on (a) spatial and temporal comparison of the whole movement with a skilled movement based on Dynamic Time Warping (DTW), (b) comparison of several kinematic variables at the deepest point of the movement to the skilled movement, and (c) number of principal components required to specify participants' movements. The general procedure of DTW ist explained in detail in Chapter 4. See Appendix A.1 for the specific details on how our measures for motor performance were calculated. Concerning the kinematic variables at the deepest point, we focus on five measures. The first two are based on a simplified center of mass (com) that is determined based on the centroid of the joint positions either on the sagittal plane (back vs. front) or on the frontal plane (up vs. down). Two further measures are based on the position of the hips (root joint, see Figure 3.2), also on the sagittal plane or on the frontal plane. The fifth measure compares the flexion of the back, based on the angle of joint vl5 (see Figure 3.2). We chose these variables as they are, from an applied point of view, critical aspects for correctly performing the squat.

For each parameter that is based on a comparison to the skilled performance, a repeated-measures analysis of variance (ANOVA) was conducted with phase (pre-test, acquisition, post-test, retention-test) as within-subject factor and group (Own, Own+skilledFront, Own+skilledSide) as between-subject factor. For all analyses, the level of significance was set at $p < 0.05$. Post-hoc comparisons were run with a

Bonferroni correction. In case of sphericity violation, the Greenhouse-Geisser correction for repeated measures was applied.

Cognitive Representation

In order to measure the participants' cognitive representation of the squat in long-term memory by way of psychometric data, structural dimensional analysis of mental representation (SDA-M; [Sch12]) was employed. The SDA-M serves to determine relations between basic action concepts (BACs) and as such to outline the structure of one's cognitive representation. Representation structures typically develop toward more complex ones after practice [FLS13], and evidence of such development would be a marker for functional changes in long-term memory. For the specific purpose of the present study, a pre-determined set of 16 concepts was used (see Table 3.1), each relating to a particular movement phase: preparation phase (BAC 1-3), going-down/ main phase (BAC 4-10), going-up/ attenuation phase (BAC 11-12), or relating to typical error patterns (BAC 13-16).

Based on individual distance scalings between BACs as obtained from the splitting procedure (cf. SDA-M; for more details, see [Sch12]), a hierarchical cluster analysis ($\alpha = .05; d_{crit} = 3.41$) was performed to outline the structure of the cognitive representation for each group and each phase. An analysis of invariance within- and between-groups served to compare different cluster solutions ([LL92; Sch12]), and thus to track the change in cognitive representation structures. According to Schack, cluster solutions are variant, that is significantly different, for $\lambda < 0.68$, while two cluster solutions are invariant for $\lambda \geq 0.68$ [Sch12]. In addition, the similarity between representation structures and a reference structure reflecting well the different movement phases was examined. For this analysis of similarity, Adjusted Rand Indices (ARI; [SE09]) were calculated for each group and time of measurement in comparison to a reference, in order to rank similarity of mean group tree diagrams relative to a reference. Indices between -1 (cluster solutions are different) and 1 (cluster solutions are the same) mark the degree of similarity. This analysis served to ensure whether the change in mental representation structures reflected a functional development toward an expert structure.

Subjective Judgments (Questionnaires)

Questionnaires. Questionnaires were used to measure simulator sickness [Ken+93], and the experience with regards to the VR set-up in terms of sense of agency and ownership toward their own avatar, perceived latency of the avatar, its anatomical plausibility, and two control questions (see Table 3.2). Questions for the second questionnaire were answered on a 7-point Likert scale, ranging from -3 to 3 (+3 indicated maximum agreement).

To test for the presence of simulator sickness induced by the system, we compared the responses to each item of the simulator sickness questionnaire between the first and the second presentation of the questionnaire using the Wilcoxon Signed rank test. Moreover, for each item and group we calculated the mean differences between post- and pre-test scores and compared those differences across the different groups

by means of the Kruskal–Wallis one-way analysis of variance. For the experience questionnaire (see Table 3.2), we calculated the mean response in each item and group. For questions relating to the participant's avatar, differences across the three groups were tested by means of the Kruskal–Wallis one-way analysis of variance. In case of significant results, post hoc comparisons were calculated by means of Wilcoxon rank sum test. For questions relating to the character that was used to display the skilled performance, Wilcoxon Signed rank test were used in each item of the experience questionnaire to test whether each response significantly differed from zero.

3.2.4 *Results*

Motor Performance

Results for motor performance variables are displayed in Figure 3.4. The ANOVA run on the temporal error based on DTW revealed significant main effect of phase ($F_{3,96} = 29.74, p < 0.0001$). Planned comparisons showed that the temporal error, reported in frames, decreased in acquisition ($M = 1.09, SD = 0.76; p < 0.0001$), post-test ($M = 1.88, SD = 0.9; p = 0.016$) and retention ($M = 1.94, SD = 0.95; p = 0.049$), as compared to the pre-test ($M = 2.3, SD = 0.97$). A significant phase by group interaction ($F_{6,96} = 3.19, p = 0.007$) showed that temporal error diminished in the Own+skilledFront group ($M = 0.66, SD = 0.15$) and the Own+skilledSide group ($M = 0.81, SD = 0.44$) in the acquisition phase as compared to pre-test (Own+skilledFront: $M = 2.62, SD = 0.88, p < 0.0001$, Own+skilledSide: $M = 1.93, SD = 0.74, p = 0.003$), post-test (Own+skilledFront: $M = 1.91, SD = 0.92, p < 0.0001$; Own+skilledSide: $M = 1.61, SD = 0.84, p = 0.003$), and retention (Own+skilledFront: $M = 1.99, SD =$

Table 3.1: Basic Action Concepts (BACs) of the squat.

	Basic Action Concept (BAC)	**Phase/Errors**
1	Stance shoulder width	Preparation
2	Toes slightly rotated outwards	
3	Upright posture	
4	Bend legs	Main phase
5	Push bottom backward	
6	Keep upright posture	
7	Knees remain behind toes	
8	Knees remain in same axis as feet and hips	
9	Heels remain on ground	
10	Knee angle 100°	
11	Push hips forward	Attenuation
12	Extend legs	
13	Push knees forward	Error patterns
14	Knees point inwards	
15	Heels leave the ground	
16	Bend upper back	

0.92, $p < 0.0001$; Own+skilledFront: $M = 1.65, SD = 0.92, p = 0.023$). The group factor was not significant ($F_{2.32} = 2.19, p = 0.13$).

The ANOVA on the spatial error based on DTW showed a decrease in all groups ($F_{3,96} = 3.8, p = 0.013$) for acquisition ($M = 1.05, SD = 0.39$), as compared to pre-test ($M = 1.35, SD = 0.44; p = 0.014$). The main effect of group ($F_{2,32} = 1.76, p = 0.19$) and the group by phase interaction ($F_{6,96} = 1.3, p = 0.28$) were not significant.

The ANOVA performed on the deviation of the center of mass at the deepest point in the sagittal plane, which is reported in meter, revealed a significant main effect of phase ($F_{3,96} = 5.21, p = 0.002$). The error decreased in the acquisition phase ($M = 0.059, SD = 0.047; p < 0.001$), post-test ($M = 0.06, SD = 0.048; p = 0.0017$), and retention phase ($M = 0.06, SD = 0.05; p = 0.04$) as compared to the pre-test phase ($M = 0.08, SD = 0.05$). Moreover, the analysis showed a significant phase by group interaction ($F_{6,96} = 4.82, p = 0.0002$). Post hoc test revealed a significant reduction of the performance error in the Own+skilledSide group only, for which motor performance improved in acquisition ($M = 0.05, SD = 0.035; p < 0.0001$), post-test ($M = 0.06, SD = 0.04; p = 0.007$), and retention phases ($M = 0.05, SD = 0.04; p = 0.002$), as compared to the pre-test ($M = 0.09, SD = 0.05$). The group factor was not significant ($F_{2,32} = 1.32, p = 0.28$).

The ANOVA on the deviation of the center of mass at the deepest point in the frontal plane, reported in meter, showed a significant main effect of phase ($F_{3,96} = 8.99, p < 0.001$). Error performance was smaller in acquisition ($M = 0.04, SD = 0.036$) as compared to pre-test ($M = 0.08, SD = 0.05; p < 0.0001$) and retention ($M = 0.07, SD = 0.048; p = 0.018$). The phase by group interaction was significant ($F_{6,96} = 2.32, p = 0.039$): error performance decreased in the acquisition phase of the Own+skilledFront group ($M = 0.03, SD = 0.015$), as compared to pre-test ($M = 0.09, SD = 0.03$), post-test ($M = 0.07, SD = 0.04$), and retention ($M = 0.07, SD = 0.04$). The group factor was not significant ($F_{2,32} = 0.62, p = 0.54$).

The analysis performed on the deviation of the root position in the sagittal plane, reported in meter, revealed a significant main effect of phase ($F_{3,96} = 3.2, p = 0.046$). In all groups, performance error decreased in acquisition phase ($M = 0.02, SD = 0.013$) and in post-test ($M = 0.025, SD = 0.013$) as compared to pre-test ($M = 0.03, SD = 0.027; p = 0.03$). The main effect of group and the group by phase interaction ($F_{6,96} = 1.31, p = 0.26$) were not significant.

The analysis performed on the deviation of the root position in the frontal plane, also reported in meter, showed a significant effect of phase ($F_{3,96} = 8.6, p < 0.0001$). In all groups, motor performance improved in acquisition ($M = 0.04, SD = 0.029; p < 0.0001$), post-test ($M = 0.06, SD = 0.03; p = 0.016$), and retention ($M = 0.06, SD = 0.038; p = 0.067$) as compared to pre ($M = 0.08, SD = 0.05$). The main effect of group ($F_{2,32} = 0.25, p = 0.78$) and the group by phase interaction ($F_{6,96} = 0.6, p = 0.26$) were not significant.

Similarly, the analysis on the deviation of the angle between hips and upper body (vl5), as reported in degrees, showed a significant effect of phase ($F_{3,96} = 4.5, p = 0.005$). In each group, the error decreased in acquisition ($M = 18.1, SD = 6.97$) and retention ($M = 18.1, SD = 8.6, p = 0.013$) as compared to pre-test ($M = 21.1, SD = 5.3$). The main effect of group ($F_{2,32} = 0.57, p = 0.57$) and the group by phase interaction

($F_{6,96} = 1.4, p = 0.22$) were not significant. The three groups did not differ in their ability to perform the squat before the beginning of the experimental session, as shown by the lack of significant differences in the pre-test phase across groups in each of the aforementioned parameters ($p > 0.8$ for all p).

The Friedman test on the number of principal components showed a significant effect of phase in the Own+skilledFront group only ($X^2(2) = 7.72, p = 0.02$). Pairwise comparison with the Wilcoxon signed-rank test (Bonferroni correction) revealed that the number of principal components decreased in the Own+skilledFront group in the post-test ($M = 2.6, SD = 1.29$) as compared to the pre-test phase ($M = 4.36, SD = 1.29, z = -2.45, p = 0.014$). The effect was not maintained in the retention phase ($M = 4, SD = 1.7, z = -0.6, p = 0.55$). The Friedman test was not significant in the Own group ($X^2(2) = 2.48, p = 0.29$), where the number of component did not significantly change across pre-test ($M = 3.5, SD = 1.38$), post-test ($M = 3.83, SD = 1.59$) and retention ($M = 4.2, SD = 1.54$). Similarly, the number of principal components did not change in the Own+skilledSide group (pre-test: $M = 3.17, SD = 1.4$; post-test: $M = 3.33, SD = 1.67$; retention: $M = 4.2, SD = 1.6; X^2(2) = 1.65, p = 0.44$).

Cognitive Representation

Mean group tree diagrams for pre- and retention-test and for the different conditions are displayed in Figure 3.5. For the Own group, the tree diagrams revealed one cluster consisting of several concepts of all three movement phases for both pre-test $[1, 3, 6, 8, 12]$ and retention-test $[3, 6, 8, 12]$. The tree diagrams of the Own+skilledFront group showed a similar cluster for pre-test $[3, 6, 8, 12]$, and three clusters of two concepts each for retention-test $[1, 6][3, 12][4, 13]$. A similar tree diagram was evident for the Own+skilledSide group at pre-test $[1, 3, 6, 8, 12][4, 13]$, but at retention-test it revealed more structured clusters $[1, 3, 8][4, 5, 13]$. That is, while for the Own and the Own+skilledFront groups, concepts of different movement phases were grouped together after practice, distinct groupings corresponding to distinct movement phases became evident for the Own+skilledSide group. Analyses of invariance revealed variance across times of measurement for two of the three groups. Specifically, the cluster solutions across time were variant for the Own+skilledFront group ($\lambda = 0.34$) and the Own+skilledSide group ($\lambda = 0.63$), but not for the Own group ($\lambda = 0.95$). This shows that the overall structure of cluster solutions changed over time for the conditions in which participants watched their own avatar together with that of a skilled person during movement execution, but not for the condition in which participants watched their own avatar only. Furthermore, adjusted rand indices indicated increasing similarity to the reference for the Own+skilledSide group from pre-test ($ARI = -0.05$) to retention-test ($ARI = 0.03$), emphasizing that the mean tree diagram of the group watching the own avatar together with a skilled performance from the rotated perspective revealed a more functional structure after the intervention. In contrast, similarity for the Own group remained stable from pre-test ($ARI = -0.03$) to retention-test ($ARI = -0.03$), and decreased slightly for the Own+skilledFront from pre-test ($ARI = -0.03$) to retention-test ($ARI = -0.05$). From these group comparisons, novices' representations changed during learning when watching their own avatar together with that of a skilled person, but not when watching their own

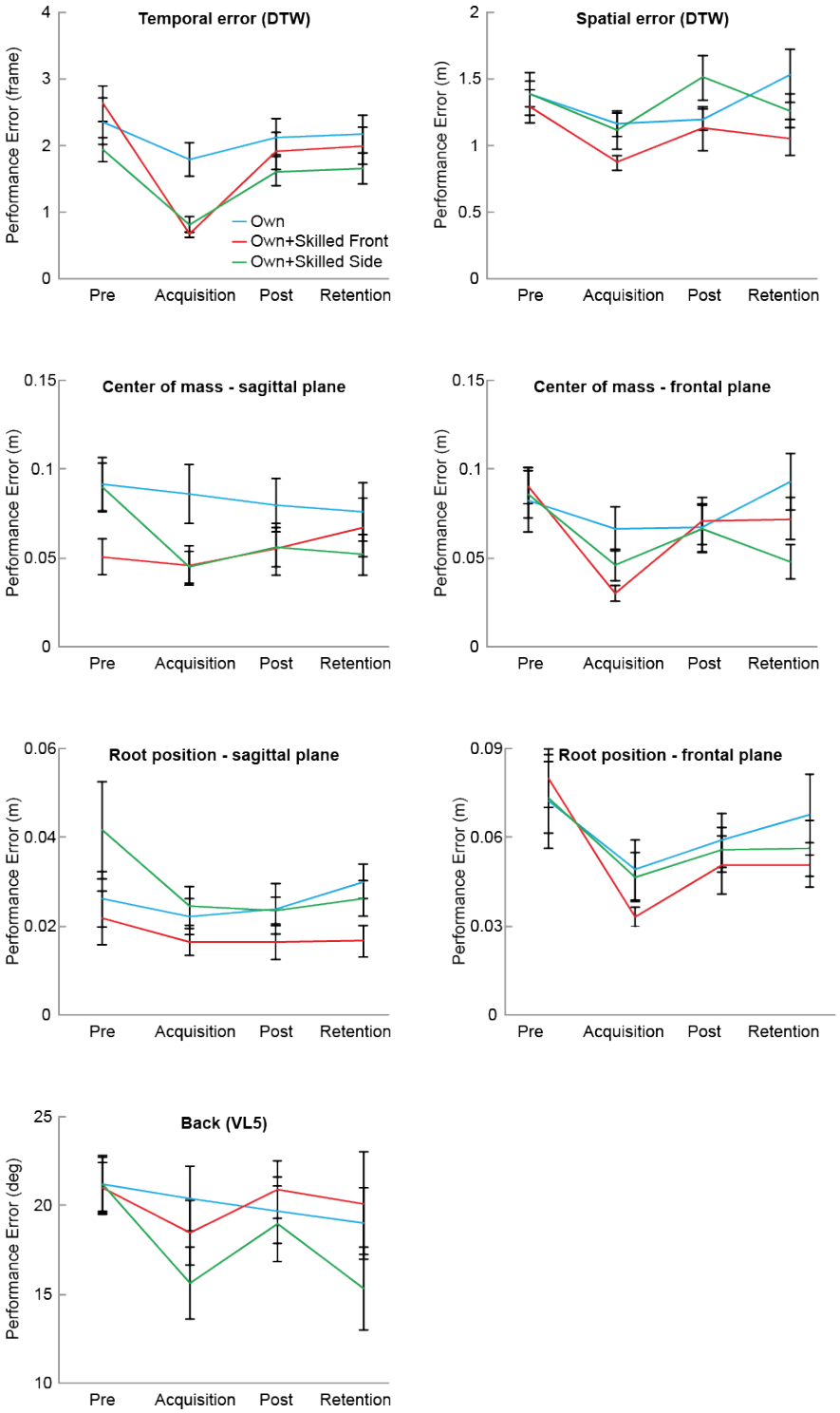


Figure 3.4: Motor performance results. Each graph shows the effect of the visual feedback provided to the different groups on each parameter used to evaluate motor performance.

avatar only. Particularly, most functional representation structures were evident after having watched their own avatar together with a superimposed skilled performance from the side.

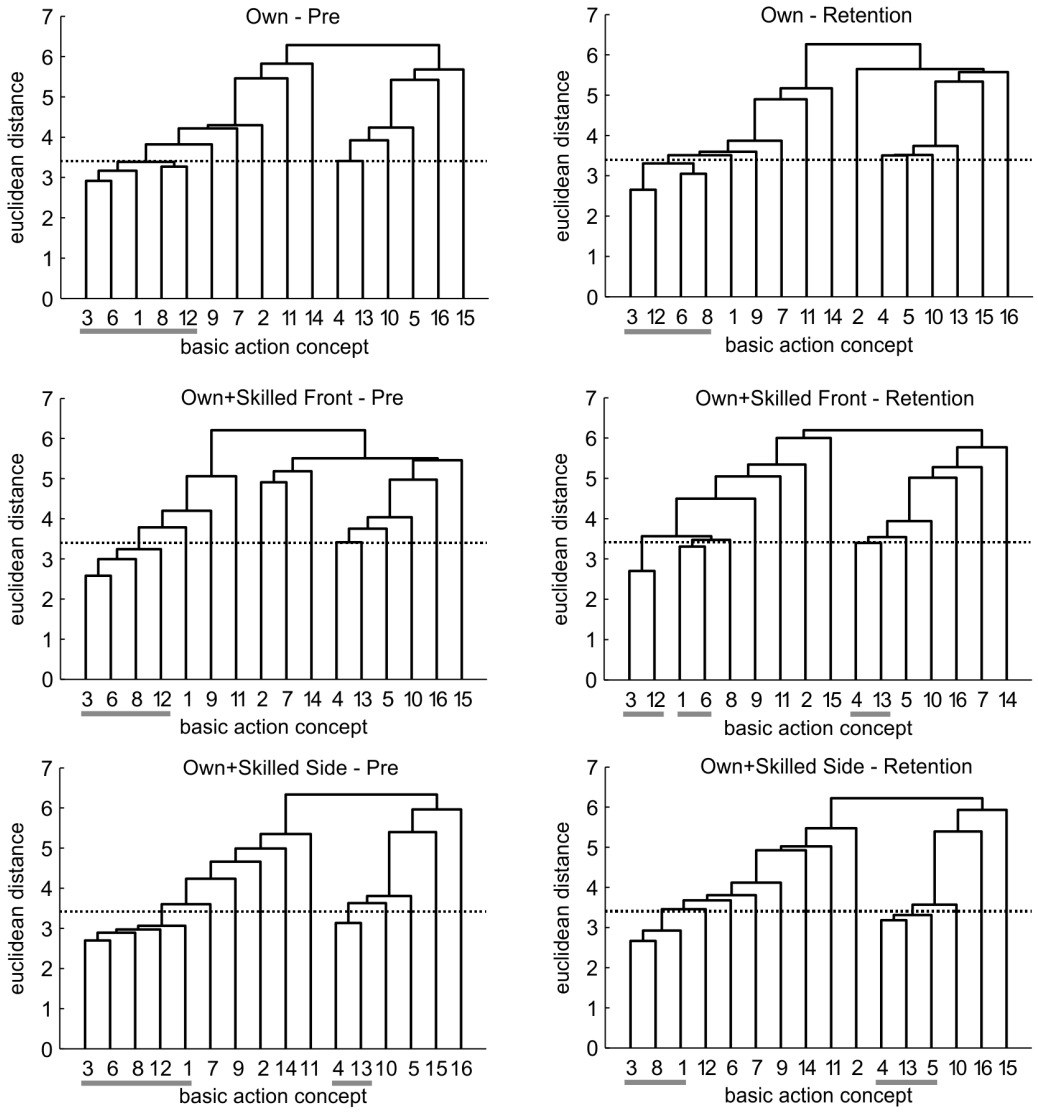


Figure 3.5: Cognitive Representation. Mean group tree diagrams for Experiment 1 displaying the three visual feedback groups for pre- and retention-test. For each tree diagram, the numbers on the x-axis relate to one particular BAC (for the list of BACs, see Table 3.1). The numbers on the y-axis display Euclidean distances. The lower the Euclidean distance between BACs, the closer the BACs are. The horizontal dotted line marks the critical value d_{crit} for a given α-level ($d_{crit} = 3.41; \alpha = 0.05$). Horizontal grey lines on the bottom mark clusters.

Subjective Judgments (Questionnaires)

Participants did not show simulator sickness after taking part in the experiment ($p > 0.37$ for all p). Moreover, the three groups did not differ in their post-pre-values in any item ($p > 0.1$ for all p). Results of the experience questionnaire are summarized in Table 3.2. The three groups had analogous sense of agency, ownership, perceived latency, and plausibility toward their own avatar, as shown by the lack of significant differences across groups in all items of the experience questionnaire ($p > 0.11$ for all p). Similarly, the Own+skilledFront group and the Own+skilledSide did not differ

in any item relating to the virtual character that was used to display the skilled performance (Wilcoxon rank sum test, $p > 0.48$ for all p). Overall, participants in all groups rated the movements of both virtual characters (own and skilled) as plausible. Moreover, they reported a high sense of agency toward their own avatar, and a low latency in the movements of their own avatar, as shown by the Wilcoxon Signed rank test against zero in each item (see Table 3.2). On contrary, they did not report sense of agency, ownership, nor a low latency as respect to their own movement toward the skilled character, as shown by negative values significantly differing from zero (see Table 3.2).

3.3 EXPERIMENT 2

To rule out the possibility that some of the improvements observed in Experiment 1 (i.e., error reduction in the motor performance and changes in the cognitive representation) were due to the mere fact of performing the squat movements per se — irrespective of the type of training received in the acquisition phase — we ran a control experiment. In Experiment 2 participants were presented with a disabled black mirror instead of the virtual mirror during the acquisition phase. If performing repetitive squats in the present experimental design without any visual feedback were enough to induce improvements in the motor and cognitive performances, we should find differences in acquisition, post-test and/or retention, as compared to the pre-test.

3.3.1 *Participants*

Twelve naïve participants (3 males, mean age $M = 27.33$, standard deviation $SD = 6.6$) took part in the study. Participants' selection criteria were the same as in Experiment 1. None of the participants of Experiment 1 took part in Experiment 2.

3.3.2 *Task and Procedure*

Task and procedure were the same as in Experiment 1. The main difference was that participants executed the squat movements during the acquisition phase in front of the same disabled black mirror they saw during pre-test, acquisition, post-test and retention. Moreover, given that participants were not presented with any virtual characters (i.e., the virtual mirror was black), participants did not fill in the questionnaire presented in Experiment 1 regarding their experiences with the avatars. Only the questionnaire on motion sickness was filled in, and participants did not show any sign of simulator sickness after taking part in the experiment ($p > 0.06$ for all p).

Questionnaire item	**Own Avatar**			**Skilled Avatar**	
	Own	Own+skilledFront	Own+skilledSide	Own+skilledFront	Own+skilledSide
The avatar's movements were caused by mine (Agency)	$2.8 \pm 0.11 * **$	$2.9 \pm 0.09 * **$	$2.4 \pm 0.23 * *$	$-1.6 \pm 0.62 * *$	$-1.1 \pm 0.66 * *$
I felt like the avatar was my own body (Ownership)	0.3 ± 0.59	0.8 ± 0.46	0.2 ± 0.61	$-2.3 \pm 0.45 * **$	$-1.6 \pm 0.6 * *$
The avatar moved as soon as I moved (Latency)	$2.4 \pm 0.23 * **$	$2.4 \pm 0.28 * *$	$2 \pm 0.39 * *$	$-1.7 \pm 0.56 * *$	$-1 \pm 0.8*$
The movement of the avatar seemed plausible (Plausibility)	$2 \pm 0.23 * **$	$2.2 \pm 0.23 * **$	$1.6 \pm 0.51*$	$2.1 \pm 0.28 * *$	$1.7 \pm 0.36 * *$
I felt as if I had more than one body (Control quest. 1)	$-2.6 \pm 0.34 * **$	$-2.3 \pm 0.33 * *$	$-2.1 \pm 0.4 * *$	$-2.3 \pm 0.43 * **$	$-2.3 \pm 0.35 * **$
I felt as if the virtual avatar would move to me (Control quest. 2)	$-2.7 \pm 0.18 * **$	$-1.7 \pm 0.59*$	$-2.2 \pm 0.32 * **$	$-2.2 \pm 0.44 * **$	$-2.4 \pm 0.19 * **$

Table 3.2: Questionnaire. Mean and standard deviation in the questionnaire items investigating participants' experience toward the virtual characters (own avatar and the character used to display the skilled performance) in the three groups are reported. The scale ranged from -3 to + 3 (+3 indicated maximum agreement). Asterisks indicate items significantly different from 0 (*** $p < 0.001$, ** $p < 0.01$, * $p < 0.05$).

3.3.3 *Results*

Motor Performance

For each parameter, a repeated-measures analysis of variance (ANOVA) was conducted with phase (pre-test, acquisition, post-test, retention-test) as within-subject factor. The analyses revealed a significant main effect of phase only for the back (vl5) ($F_{3,96} = 5.618, p = 0.003$). Post hoc comparisons revealed a decrease of the performance error, reported in degrees, in the retention phase ($M = 12.83, SD = 1.85$) as compared to the pre-test ($M = 17.87, SD = 2.87, p = 0.013$), and the post-test ($M = 18.36, SD = 2.58, p = 0.016$). In all the other parameters, the main effect of phase was not significant ($p > 0.16$ for all p), showing that—for most of the tested parameters—the mere execution of squat movements in the absence of visual feedback was not enough to induce improvements in the motor performance. Moreover, we ran a Friedman test on the number of principal components in the pre-test, post-test, and retention phase. Results showed a significant increase of the principal component number ($X^2(2) = 8.71, p = 0.013$) in the post-test ($M = 8.08, SD = 3.6$) as compared to both the pre-test ($M = 3.92, SD = 1.2, z = -2.55 p = 0.018$) and the retention phase ($M = 4.42, SD = 1.44, z = -2.68, p = 0.0047$).

Cognitive Representation

The tree diagram for pre-test revealed two clusters, one cluster pertaining to both preparation and main phase [1368] and one cluster pertaining to the main phase [45]. For retention-test, the diagram was composed of two clusters, one involving concepts of two movement phases [36], and one including an error pattern [413]. Analyses of invariance revealed variance across times of measurement ($\lambda = 0.56$), indicating that the overall structure of cluster solutions changed over time. However, adjusted rand indices displayed decreasing similarity to the reference from pre-test ($ARI = 0.07$) to retention-test ($ARI = -0.03$). This indicates that the mean tree diagram changed to a more dysfunctional structure after the intervention.

3.4 DISCUSSION

In the present study we used our state-of-the-art VR system for the coaching of full-body motor actions to provide on-line visual feedback during the learning of a sport technique. We compared the effectiveness of three different types of visual feedback in the acquisition of a proper squat technique: the participant's avatar during the execution of squat trials was presented either alone (Own) or with the superimposed character used to display a skilled performance, either from the front (Own+skilledFront) or the side (Own+skilledSide) view. Results showed an advantage of the groups observing their own avatar performing the squat movements together with the skilled performance over the view of their own avatar alone. In Experiment 2, which investigated squat acquisition without any visual feedback, we found a slight tendency of performance to get even worse. In Experiment 1, participants tent to

adapt to the temporal aspects and the depth of the skilled movement. In particular, during the acquisition phase, the Own+skilledFront and the Own+skilledSide groups similarly adapted the timing of their performance to the skilled one. Regarding the center of mass at the deepest point, the Own+skilledFront group reduced the motor error for height during the acquisition phase. This finding showed an advantage of observing both virtual characters from a front view for correctly estimating how deep participants should go to perform a correct squat. For the center of mass at the deepest point on the sagittal plane, we found an advantage for the Own+skilledSide group in the phase acquisition over the other groups, which was maintained in retention phase. This means that if participants are presented with a side view of the two virtual characters, they can correctly learn how they should adjust their squat along the back-front axis. Thus, we observed changes in motor performance for aspects that could be perceived by the observer according to their particular viewing perspective [SN85]. In general, providing participants with the mere view of their avatar already decreased performance error as compared to the absence of visual feedback. Indeed, performance error decreased for the spatial comparison of participant's movement to the skilled movement (Dynamic Time Warping) and in the positioning of the hips (deepest point) in all groups provided with online feedback. Instead, practicing squat movements without any visual feedback (Experiment 2) did not significantly improve the overall motor performance. The participants who performed the squat movement in front of a disabled black mirror and in the absence of direct feedback reduced their error only for one single parameter (vl5; flexion of the lower back), similar to the other groups. Concerning the PCA analysis, we observed a reduction of the principal components at the end of the training for the participants in the Own+skilledFront group. Participants in the other feedback groups did not show any change in the PCA analysis before and after the training. Instead, and opposed to the Own+skilledFront group, executing the task in the absence of any direct feedback increased the number in the principal components. Similar to motor performance, the advantage of providing the avatar of the trainee together with that of an expert is also noticeable in participants' cognitive representation of the squat, as analyzed with the SDA-M [Sch12]. Participants in the Own+skilled groups revealed changes in cognitive representations of the squat after the acquisition phase. In particular, those who observed the two virtual characters from a side view showed a more structured cognitive representation, which lasted beyond the training session. Participants who watched their avatar only did not show any change in their cognitive structure. Similarly, and together with an increase in motor performance, previous studies have shown that representation structures develop toward more elaborate ones as a result of practice by execution [FLS13] as well as mental types of practice such as observation [FKS18; KFS17] or imagery [Fra+14].The finding that representations of those who performed the task in the absence of any visual feedback changed toward a more dysfunctional structure indicates that the absence of visual guidance might even lead to a deterioration of the cognitive representation. This together with the PCA findings suggests that performing movements without visual feedback might even be detrimental, as participants in the present study get worse both in the functional groupings of action concepts in memory (i.e., their cognitive

representation) and in the number of principal components constituting the overall movement. This is in line with the notion that changes on cognitive levels of action organization are linked to changes on the motor level. For instance, using a spatio-temporal kinematic decomposition of movement together with SDA-M for the full swing in expert golfers, Land et al. found a close link between movement kinematics and the structure of golfers' cognitive representation of the swing [Lan+13]. It is unlikely that the differences we found across groups result from differences in the way the avatars were perceived across groups. Specifically, according to participants' ratings, participants in all groups perceived avatar's movements as similarly plausible, having a very low latency, and inducing a similar sense of agency and ownership. Overall our study shows that observing the participant's own avatar together with the superimposed skilled performance displayed on a second virtual character can improve motor performance while practicing a full-body movement. Previous studies that focus on complex full free body movements (i.e., that are not restricted to one single body part, and not related to the use of tools) showed conflicting results. For instance, Chua et al. examined the effectiveness of a VR training for Tai Chi, which is a sport that—similarly to squat—requires the execution of slow full-body movements[Chu+03]. Performances of a skilled athlete who performed the to-be-learned motor action were shown together with the avatar of the subject. The authors tested several feedback conditions, but did not find any feedback-specific improvements [Chu+03]. Such lack of improvement, in contrast to our results, might well be explained with the high end-to-end latency in the setup used by Chua and colleagues (around 170 ms). Indeed, when participants are presented with a real-time feedback of their movements (e.g., when observing their own virtual avatar), a high end-to-end latency might affect the perceived temporal coherence of the scene, inducing a break-down in sense of agency and sense of ownership toward the virtual avatar and affecting motor performance [Fra+01; IA15; JNS12; LH09]. Instead, the setup that we used in the present study has a low end-to-end latency which might have allowed us, in contrast to [Chu+03], to observe improvements in motor performance. In a previous study we asked participants to perform a series of full-body movements, and we presented them with their own virtual avatar, whose performance was delayed between 45 and 350 ms [Wal+16]. We showed that, in our setup, awareness for delays significantly increases for an end-to-end latency above 75 ms. Further, a latency above 75 ms led to a gradual decay in motor performance. Perceptual aspects such as sense of agency and ownership were affected for a latency above 125 ms [Wal+16]. The latency in the setup discussed in [Chu+03] presents an end-to-end latency that according to our previous results would be enough to affect simultaneity perception, motor performance, sense of agency and ownership [Wal+16]. Therefore, the online feedback (i.e., participant's own avatar) would be perceived as significantly less simultaneous to the participants' movement as compared to our setup, and motor performance would drop with increasing delay.

In an analogous study, Burns et al. investigated karate learning by comparing a 'traditional group' (in which a teacher gave oral explanations and some practical examples of the movements), a group observing a video of a teacher performing a prerecorded example, and a virtual character showing an example of the gestures [Bur+11]. The

results of this related study showed no significant difference on the performance after training in the three groups. In contrast to Burns et al., in our study we showed that a VR environment providing low-latency visual feedback of the trainee's avatar together with that of an expert improves motor performance. Compared to conditions in which participants performed the task in the absence of visual feedback or just observing themselves in a virtual mirror, the participants can reduce motor error by directly comparing their performance to the target one. Directly comparing one's own to a skilled performance is a clear advantage as compared to what would happen in real training environments, in which learners are provided with instructions and visual examples by the coach, and subsequently have to repeat what they just observed in front of a mirror (or even in the absence of it). This process implies cognitive load: for instance, the learners have to retrieve the relevant information provided by the coach from memory. Moreover, this process is further complicated by the fact that a novice, who by definition has no experience with the to-be-learnt sport, does not know which the most common errors are, and to which body parts he/she should pay more attention in order to avoid such errors. Having the opportunity to directly compare the own avatar to that of a skilled individual offers an advantage that would not be possible in a real environment. Furthermore, the possibility of showing the two virtual characters from different points of view (e.g., from the front, or from the side, which would not be possible in a real environment) provides an additional gain. For instance, participants who observed the virtual characters from a side view were able not only to correct their motor performance (which is completely visible only from the side view) during the training, but also showed improved performance the day after in the retention phase, which indicates motor learning (i.e., the ability to maintain the practiced improvement over a period of time and without receiving further feedback; [KW12]. Overall, in the present study, participants only showed little learning that lasts over a day. Even if they tended to reduce the error in performance during the training, they tended not to preserve the improvement the day after. Only the participants who observed the virtual characters from a side view maintained their performance advantage with respect to the center of mass at the deepest point (sagittal plane) over the retention period. For the other parameters, the improvement in performance was not significantly maintained the day after. This might be mainly due to the fact that we used concurrent feedback during task execution, which is particularly effective for novices [Sig+13], but often leads to a dependency on the feedback [Sch+89; Sch91; WS90].

3.5 CONCLUSION

We used our VR setup to investigate the effectiveness of different kinds of augmented feedback in a motor learning scenario. We found that performing squats together with a superimposed motion of a skilled subject can increase a novice's performance. According to our results, observing both movements from different perspectives can further increase the performance depending on specific features of the movement. Performance was measured on two levels: the subject's motion during the experiment as well as the subject's cognitive representation, and subjective experience, of the task.

Our results demonstrate that our environment is suitable to induce an improvement in motor performances of novice athletes. However, our ways to provide feedback are still limited as the system does not have any information on the quality of the athlete's performance during runtime. Analyzing a full-body movement during runtime, in order to provide online feedback, is a complex task. Approaches towards offline and online motion analysis in this context frequently apply Dynamic Time Warping [ARB08; Pet+14; Xi+06; YB14]. In the following chapter, we propose an approach towards an optimized online alignment of human motor performances, based on online Dynamic Time Warping. This alignment will serve as the basis for the detection of typical errors in motor performances in the subsequent chapter.

4 ACCURATE ONLINE ALIGNMENT OF MOTOR PERFORMANCES

Published in:
[Hül+17]

Online algorithms for motion analysis and synthesis become highly important, as applications, such as virtual coaching environments, gain more and more popularity. However, numerous data-driven state-of-the-art algorithms for movement analysis and synthesis were originally developed to work offline. Many of them require a temporal alignment of an input motion with a reference trajectory as preliminary step [GP00; Krü+17; MC12]. This alignment is frequently achieved via Dynamic Time Warping (DTW). Even though DTW can be prone to outliers and noise [VKG02], DTW and its extensions provide compelling results in various applications. Unfortunately, DTW needs a whole trajectory to be completed before it can start calculating the optimal alignment. Thus, algorithms that rely on DTW can only provide results as soon as the input motion has been completed. If the alignment could be estimated earlier, ideally directly after an input frame is observed, such algorithms could already provide results online during the performance.

An extension of DTW, Open-End DTW (OE-DTW), has been shown to work for the alignment of trajectories in online scenarios [Tor+09], but unfortunately it can perform much worse than its offline counterpart. This is crucial: If the alignment fails, e.g., the algorithm decides that the final frame of an incomplete input motion still matches an early frame of the reference motion (cf. Figure 4.1), the whole alignment can become useless. In this case, all further steps that build on the aligned trajectories, such as motion classification, might fail.

In this work, we extend OE-DTW by path-length weighting together with joint weights based on evolutionary optimization to improve the alignment. We call the resulting algorithm Weight-Optimized Open-End DTW (WOOE-DTW). This algorithm is used as a basis to extend the classification pipeline proposed in the subsequent chapter to work online. Despite the fact that there is a large number of related work on DTW and Open-End DTW, we are the first who combine path-length weighting with evolutionary optimized joint weights in favor of an improved alignment performance of Open-End DTW. Joint weights already appear in the literature, however, these are often engineered based on prior knowledge of the movement of interest or are based on heuristics. We propose a data-driven optimization-based approach that can, additionally to the improved online DTW performance, even uncover insights on the movement of interest via the estimated joint weights. We demonstrate that for our test scenario, which contains recordings of 95 body-weight squats by 49 subjects, our algorithm WOOE-DTW clearly enhances the alignment performance compared to OE-DTW. Further, we also evaluate WOOE-DTW based on the Tai Chi data set.

My Contribution *The approach proposed in this chapter was developed in cooperation with Andreas Richter. I performed the analysis of the classic Dynamic Time Warping (DTW) algorithm and its extension, as well as the conception and development of the newly proposed extensions. Concerning the evolutionary optimization, I was supported by Andreas Richter who*

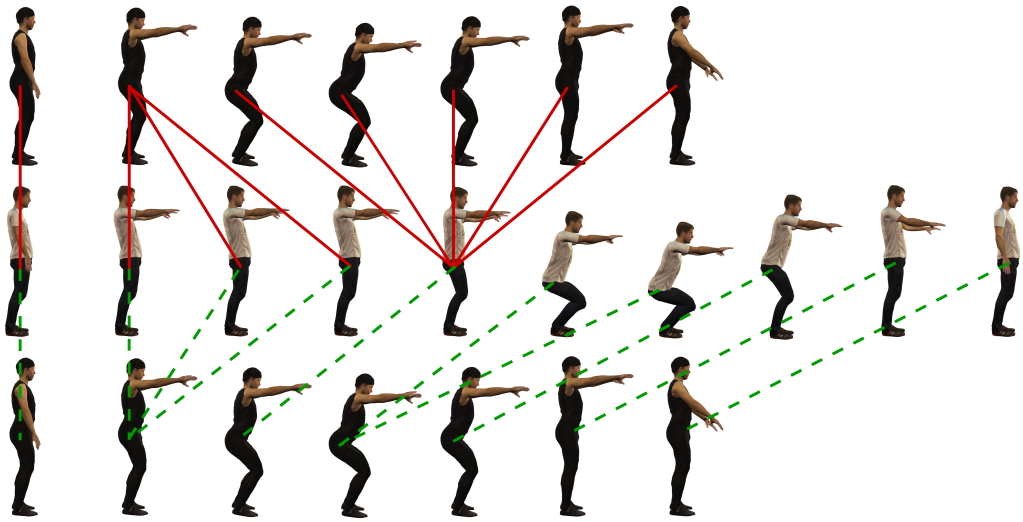


Figure 4.1: Bad (top) and good (bottom) alignment of a squat performance with a reference movement (middle). Note that the bad alignment estimates an early frame of the reference as the end frame. Nearly the whole input movement is mapped on one single reference frame. Using this alignment to warp the input into the timing of the reference would result in an incomplete movement. If the good correspondence would be used, the result would be a similar movement than the reference squat in time, but with the spatial properties of the input trajectory.

helped in setting up the optimization environment and provided knowledge on evolutionary optimization that helped during the development process.

4.1 RELATED APPROACHES

One way to improve the performance of OE-DTW is to weight individual features [Ari+14; CCK15; Cel+13; JJO11; Par+18; RDE11; Tan+18; Yua+18]. This weighting can reduce noise induced by unimportant features: If, for instance, two squat movements have to be aligned, any impact of, e.g., the rotation of the wrist should be minimized, since the rotation of the wrist has nothing to do with the performance of the squat. If this joint influences the alignment, it can only induce noise. Jeong et al. propose an approach to prevent DTW from aligning frames that belong to different phases of a repetitive movement [JJO11]. They introduce a penalty based on the temporal location of a frame. For the alignment of motor performances we mainly focus on the online analysis of the most recent single motor action. Thus we might not suffer from repetitive movement, for which penalties as described in [JJO11] would be wise to use. For an application in signature verification Parziale et al. build up on this approach [Par+18]. They use domain knowledge to introduce a stability criterion. Points that lie outside of stable regions are penalized. For hand writing, Tang et al. also integrate domain knowledge. Different weights are assigned to subsequences based on preprocessed information on the overall structure. Yuan et al. focus on the classification of time series data based on k-Nearest-Neighbor (kNN) [Yua+18]. To this end, they propose a locally weighted DTW that helps to achieve homogeneous neighborhoods for each class from which

the kNN classifier can profit. Other approaches introduce joint weights based on inter- as well as intra-class variability of the gesture of interest [RDE11]. In their evaluation, the weighted DTW improved the classification quality for gesture recognition. Arici et al. [Ari+14] and Celebi et al. [Cel+13] follow a similar approach. Weights for each joint are calculated for each gesture class of interest. The basic idea is to capture the contribution of each joint to the specific gesture. This contribution is quantified via the total displacement of the joint during the performance of a trained user. Furthermore, an additional meta parameter inside the weighting term is calculated based on maximization of a discriminant ratio with respect to different gesture classes. The authors' extensions of DTW increase the performance of gesture classification for their test cases. This approach is extended in [CCK15]: The authors introduce additional dynamic weights which are able to change over time. To summarize, related approaches to feature weighting mainly focus on the overall movement, inter-class differences for classification, or the variance of a feature. Variance-based approaches seem most promising as they do not require a specific classification task to be linked with the alignment. However such an approach prevents to account for important joints whose movement is rather small. Additionally, unimportant joints that move mostly non-functional could be higher ranked than the important ones. Instead of variance-based weights, we propose an approach that uses a suitable optimization of DTW weights. To this end, we introduce an error measure for DTW alignments and use it to optimize weights instead of requiring the data to be linked to a classification task.

Another problem of DTW is its bias against temporal shifts in the warping function [AF13; Dix05; SC78]. This bias is normally unwanted. As a solution, some related approaches propose specific penalties that reward temporal shifts [Dix05]. Unfortunately, this has a major drawback: Even when a warp without any shift would be correct, the penalty could induce the algorithm to prefer a different mapping. To make DTW independent from assumptions on the movements' timing, we apply path-length weighting [AF13; MGB09]. This approach slightly increases the computational effort, but avoids the bias of DTW. In contrast to Anguera et al. [AF13], we do not privilege specific directions of the alignment path. Furthermore, we weight each feature on the whole temporal axis equally, wheres in [AF13] later frames implicitly gain more weight for DTW. In favor of performing path-length weighting, Muscariello et al. [MGB09] introduce a specific weight matrix to store local path length weights as well as a matrix that stores possible alignment path lengths. In our approach, we only require one additional matrix that stores the paths' lengths.

4.2 DOMAIN AND DATA SET

Motor actions can be divided into smaller homogeneous subsequences that we call *movement segments*. For the squat, these are for instance "preparation", "going down", "is down", "going up" and "wrap-up". For the Tai Chi push, these are "preparation", "push", "pushed", "retraction", "wrap-up". If we know, for each frame of a trajectory, to which movement segment it corresponds, we can use this information to assess the quality of an alignment of this trajectory to another trajectory. To this end, we

only need to test whether frames that correspond to a specific movement segment in trajectory 1 were aligned to frames of the same movement segment of trajectory 2. Consequently, we extend the data sets described in Chapter 2.5 by an annotation of movement segments.

For all our steps performed in this chapter, we use cross-validation (CV) with 5 folds. Our plots contain averaged results. We ensured that no data from any recorded subject contained in a specific training set is contained in the corresponding test set. This enables us to test the generalization to new subjects, which is especially crucial as performances can vary much between subjects.

4.3 ONLINE TEMPORAL ALIGNMENT

Dynamic Time Warping (DTW) establishes a frame-to-frame correspondence between two trajectories. This is achieved via finding the optimal alignment path between both, according to an integrated per-frame distance measure. The following subsection describes the standard DTW algorithm [Mül07, p. 69].

Let T_1 and T_2 be motion capture trajectories. Here, $|T_1|$ and $|T_2|$ denote the number of frames in the trajectories T_1 and T_2. To establish the correspondence between T_1 and T_2, DTW first calculates the $|T_1| \times |T_2|$ per-frame distance matrix $\mathbf{M}$. Each element $\mathbf{M}(i,j)$ contains the distance between frame i of T_1 and frame j of T_2. We define this distance as the summed distance between all quaternions $\mathbf{q}_1, \ldots, \mathbf{q}_k$ (cf. Chapter 2.5) of these frames. As quaternion distance, we use the inner product as evaluated in [Huy09] and construct each entry in $\mathbf{M}$ as follows:

$$\mathbf{M}(i,j) = \sum_{d=1}^{k} (1 - |\mathbf{q}_{i,d} \cdot \mathbf{q}_{j,d}|). \tag{4.1}$$

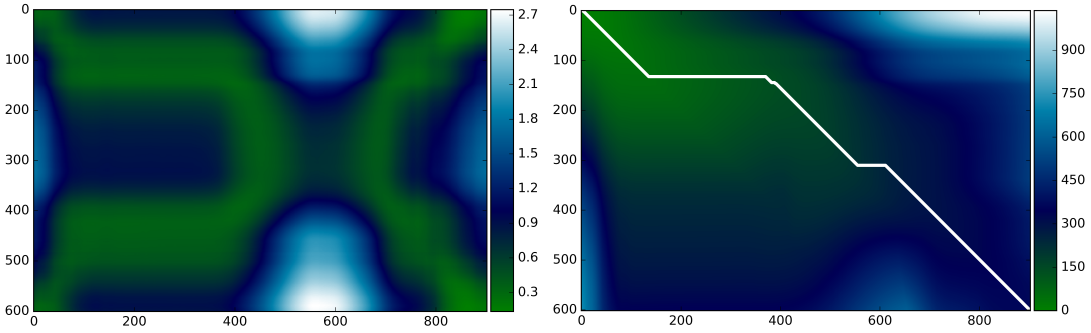
Here, k is the number of joints (cf. Chapter 2.5). The rotation of joint d at position i in T_1 is denoted by $\mathbf{q}_{i,d}$ and $\mathbf{q}_{j,d}$ is the rotation of joint d at position j in T_2. Note that the operator $\cdot$ denotes the dot products of two vectors, not the multiplication of quaternions [Huy09]. DTW now finds the alignment path with minimal costs from start $\mathbf{M}(1,1)$ to end $\mathbf{M}(|T_1|,|T_2|)$ through the matrix based on dynamic programming. See [Mül07, p. 69] for a formal definition of the alignment path. To obtain the alignment path, one calculates first the $(|T_1|+1) \times (|T_2|+1)$ matrix $\mathbf{D}$ which accumulates the minimal costs on possible paths. In the accumulated cost matrix, $\mathbf{D}(i,j)$ corresponds to $\mathbf{M}(i-1,j-1)$ in the local cost matrix. The accumulated cost matrix $\mathbf{D}$ is initialized as follows:

$$\mathbf{D}(i,j) = \begin{cases} 0, & \text{if } i = 1 \text{ and } j = 1 \\ \infty, & \text{otherwise} \end{cases}.$$

The entries of $\mathbf{D}$ are iteratively updated as follows:

$$\mathbf{D}(i,j) = \mathbf{M}(i-1,j-1) + \min\{\mathbf{D}(i-1,j-1), \mathbf{D}(i-1,j), \mathbf{D}(i,j-1)\} \tag{4.2}$$

for $2 \leq i \leq |T_1|+1, 2 \leq j \leq |T_2|+1$. The alignment path is traced back via minimizing the accumulated error in each step, starting from $\mathbf{D}(|T_1|+1,|T_2|+1)$ [Mül07, p. 73]. See Figure 4.2a for a visualization of a per-frame-distance matrix and Figure 4.2b for



(a) Per-frame-distance matrix $\mathbf{M}$ which contains the posture-wise distances according to Equation 4.1.

(b) Accumulated cost matrix $\mathbf{D}$ according to Equation 4.2 together with the alignment path.

Figure 4.2: These matrices are obtained from DTW on squat trajectories. The axis' labels indicate the frame number of the corresponding trajectories. Green denotes a smaller distance, blue and white denote a larger distance.

the corresponding accumulated cost matrix together with the calculated alignment path. To warp trajectory T_1 to the timing of T_2, we select the corresponding frame in T_1 according to the calculated alignment for each frame in T_2. If multiple frames are aligned to a frame in T_2, we select the one that is in the middle of these frames on the temporal axis.

As DTW needs two complete trajectories to calculate an alignment it cannot work online. This can be bypassed by using Open-End DTW (OE-DTW) [Tor+09]. OE-DTW allows to align a prefix T_1 of a query trajectory with a complete reference trajectory T_2. It yields a warp as well as an estimation of which frame in the reference matches the last frame of T_1. Thus, the backtracing step in OE-DTW does not start from $\mathbf{D}(|T_1|+1, |T_2|+1)$, but from $\mathbf{D}(|T_1|+1, \Omega)$, with

$$\Omega = \arg\min_j \mathbf{D}(|T_1|+1, j), \text{ where } 2 \leq j \leq |T_2|+1.$$

Consequently, $\Omega - 1$ is the frame in the reference trajectory that matches the last frame of the incomplete trajectory T_1. To calculate OE-DTW for a new incoming motion frame, we only have to update the last row of the cost matrix $\mathbf{M}$ as well as the last row of the accumulated cost matrix $\mathbf{D}$. For implementation reasons, we start with the calculation as soon as the input motion consists of at least three frames.

We compare the alignment quality of OE-DTW to the quality of the offline alignment. To estimate the alignment error, we make use of the annotated movement segments: When T_1 is warped to the timing of the reference T_2, we check, for each pair on the alignment path, whether the frames of T_1 and T_2 belong to the same movement segment. If this is the case, the error value of that pair is 0. For each pair of frames where this is not the case, we calculate the offset to the next frame on T_2 that is annotated with the desired movement segment. This distance can be described as the alignment error per frame-pair. We normalize this distance via dividing by the length of T_2. The maximum error value per frame that is theoretically possible is thus 1. Finally, the average alignment error over the whole path is returned. Figure 4.3 contains the results of the comparison between standard offline DTW and

OE-DTW for the squats and for the Tai Chi pushes. In both cases, we observe that the alignment quality of OE-DTW is inferior to the one by standard offline DTW. However, the overall alignment error of the squats is much higher than the one of the Tai Chi pushes. Also the difference between OE-DTW and the offline variant is much more dominant for the squat. In case of the Tai Chi pushes, if only the first 40 % of the input trajectories are known, the alignments are nearly correct. However, for the squat, in some cases, the alignment even becomes completely degenerated: Starting from one specific frame of the input trajectory, all further frames are warped to the same frame of the reference trajectory. Such an alignment is visualized in Figure 4.4a. In the following, we describe and evaluate our proposed extensions to improve the alignment performance.

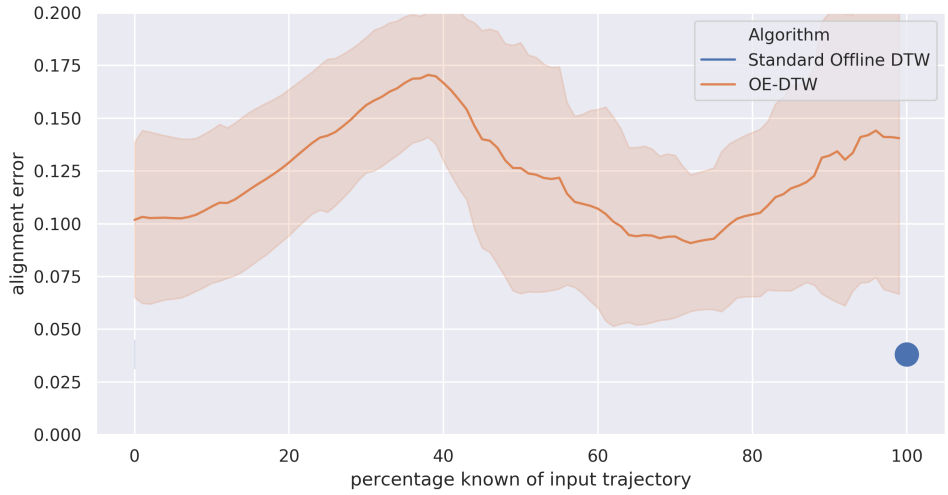
4.3.1 *Path-length weighting*

The formulation of the accumulated cost matrix is biased towards shorter paths [AF13; Dix05; SC78]: The shorter the path, the smaller the accumulated error. This bias would make sense, if shorter paths would represent — in general — better alignments than longer ones. However, this is not the case: Let us consider the alignment of two performances of the same motor action that are similar in the spatial domain, but differ in timing. One of these trajectories is performed with a specific speed. Now, the more similar the speed of the other trajectory is, the shorter is the alignment path for the standard DTW: It would stay mainly on the diagonal of $\mathbf{D}$. However, if the performance is paused, e.g., because the performing subject has to think about how to continue, an optimal alignment path must leave the diagonal to account for the change in timing. If we allow the algorithm to prefer shorter paths, which means preferring less deviation in timing, it would tend to stay on the diagonal. To make DTW independent from assumptions on the movements' timing, we apply path-length weighting via adapting Equation (4.2) as follows:

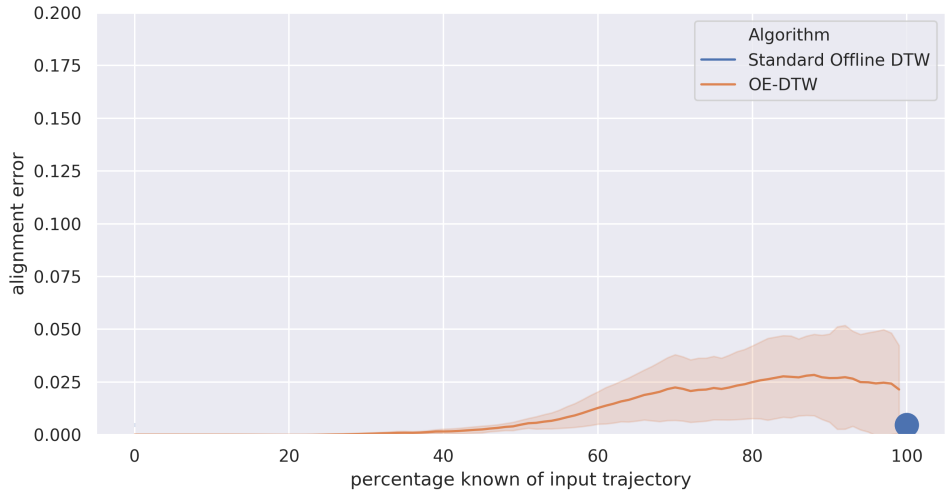
$$\mathbf{D}(i,j) = \mathbf{M}(i-1,j-1) + \mathbf{D}\left(\underset{(k,l)}{\arg\min}\left(\frac{\mathbf{D}(k,l)+\mathbf{M}(i-1,j-1)}{\mathbf{L}(k,l)+1}\right)\right), \quad \text{where } (k,l) \in \{(i-1,j-1),(i-1,j),(i,j-1)\}. \tag{4.3}$$

Matrix $\mathbf{L}$ contains the path-lengths of each optimal path. It is updated together with $\mathbf{D}(i,j)$ based on the just calculated values for k and l. After calculating $\mathbf{D}$ and $\mathbf{L}$, we determine the optimal path via backtracing from $\mathbf{D}(|T_1|+1,\Omega)$. In each step, we divide all examined cells of the accumulated cost matrix $\mathbf{D}$ by their corresponding path-lengths from $\mathbf{L}$ and select the one with the smallest result.

Figure 4.5 displays the alignment quality of OE-DTW with path-length weighting compared to standard OE-DTW. Additionally, we calculate the alignment when extending OE-DTW with a diagonal penalty [Dix05]. We observe a positive impact of path-length weighting on the alignment of both motor actions, the squats as well as the Tai Chi pushes. When removing the weights or when using a penalty factor for diagonal steps the accuracy decreases.

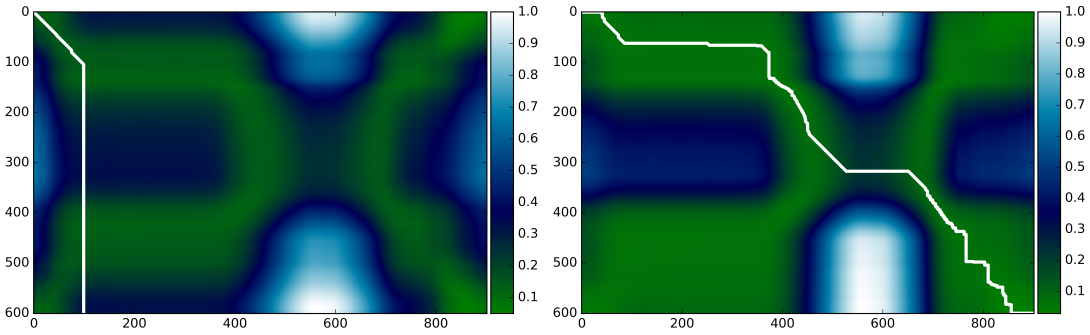


(a) Squat.



(b) Tai Chi push.

Figure 4.3: Comparison of the alignment error of standard DTW and OE-DTW averaged over all folds. Standard DTW only provides results as soon as the whole input trajectory is known, OE-DTW can already provide alignments earlier. For OE-DTW, the plot shows the mean values over all cross-validation folds together with the standard deviation. For Standard DTW, the mean value over the cross-validation folds is marked.



(a) Failure of OE-DTW alignment: After a certain early input frame, all remaining input frames are matched to the same reference frame.

(b) WOOE-DTW significantly improves the alignment quality for the same two trajectories.

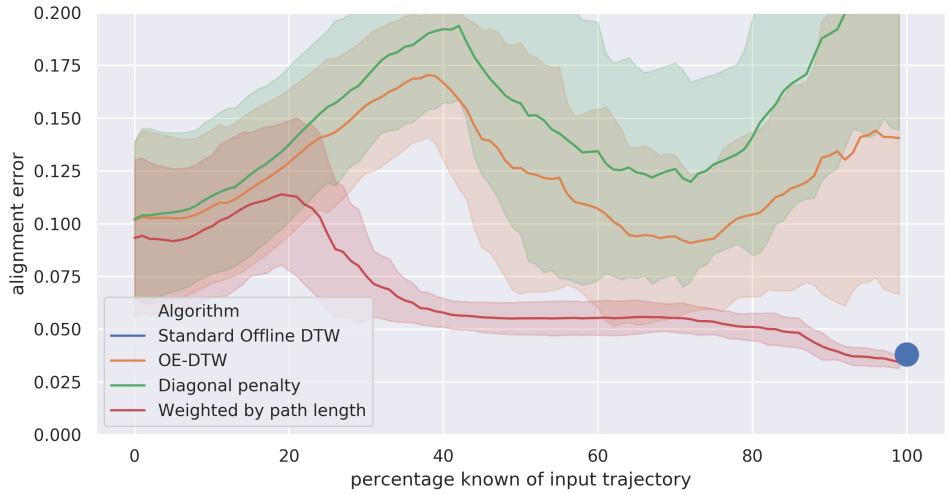
Figure 4.4: Comparison of exemplary alignments based on OE-DTW and WOOE-DTW. The image displays the local cost matrix $\mathbf{M}$ together with the alignment path estimated by the two DTW variants. Both cost matrices are normalized to the same interval. The axis' labels indicate the frame numbers of the corresponding trajectories.

4.3.2 *Evolutionary-weighted DTW*

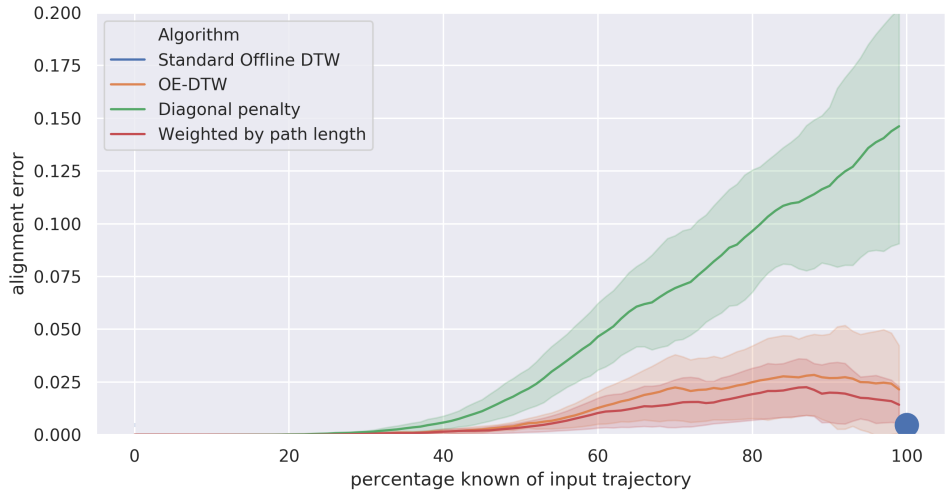
DTW uses an equal weighting for all joints in Equation (4.1). However, for a given motor action, certain joints are more important for the alignment than others: A motor action such as the squat, for instance, mainly depends on the motion of the legs. For DTW, non-functional motion in other joints, such as the wrists, has the same impact on the alignment as these important joints. Thus, if the motion in the legs is only minimal, but the wrists move a lot, they would dominate the alignment, although an optimal alignment would prefer a simultaneous motion in the legs. Intuitively, one would thus increase the weight of the important joints. Thus, we incorporate the joints' importance for warping using a weight vector $\mathbf{w}$ with an entry w_d for each joint, where $d \in \{1, \ldots, k\}$. Consequently the DTW cost matrix $\mathbf{M}$ from Equation (4.1) is adapted as follows:

$$\mathbf{M}(i,j) = \sum_{d=1}^{k} w_d (1 - |\mathbf{q}_{i,d} \cdot \mathbf{q}_{j,d}|). \tag{4.4}$$

In a naive approach, we would now weight the joints we consider important (e.g., the legs for the squat) more than the unimportant ones (e.g., the wrists). Depending on the type of motor action, this could require a huge effort in manually adjusting the weights. Instead, we propose a data-driven approach to find the appropriate weights. Related approaches for a better joint weighting [Ari+14; Cel+13] often quantify the joints' importance via their overall contribution to the motor action. The contribution is quantified e.g., by calculating the variance of the joints' features in training recordings of the motor action of interest. However, these joint weights are prone to a high amount of noise in some of the involved joints. Further, in a variance-based approach, no information on whether large changes are functional and relevant for specific movement segments is integrated. We aim at a goal-directed approach which optimizes the joint weights $\mathbf{w}$ by a minimization of the alignment error. In order to



(a) Squat.



(b) Tai Chi push.

Figure 4.5: Impact of path-length weighting on the average alignment error of OE-DTW. For online versions of DTW, the plot shows the mean values over all cross-validation folds together with the standard deviation. For Standard DTW, the mean value over the cross-validation folds is marked.

optimize $\mathbf{w}$ efficiently, we use a simpler version of the alignment error described in the previous section via just counting the correctly and incorrectly aligned frame pairs. The alignment error for optimization is quantified as

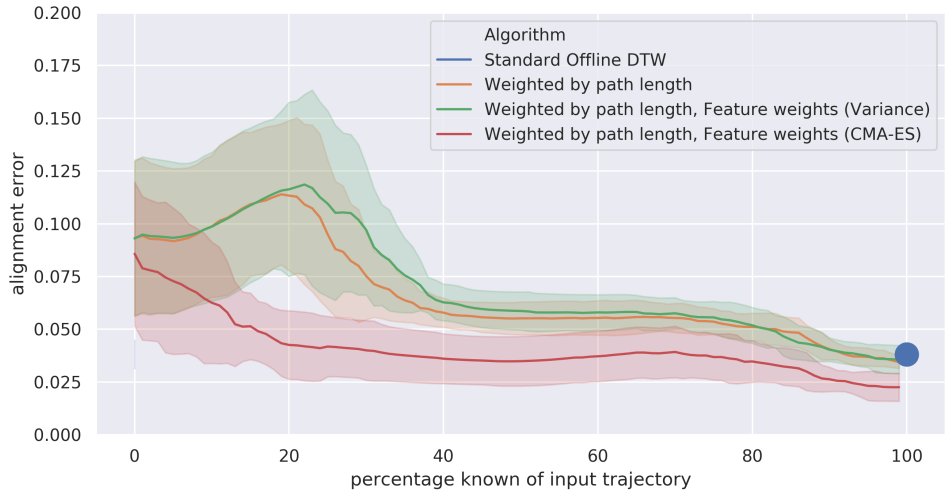
$$\frac{\#IA}{\#CA + \#IA}, \tag{4.5}$$

where $\#CA$ is the number of frames on the alignment path that are aligned to the correct movement segment, $\#IA$ is the number of incorrectly aligned frames, according to the annotations.

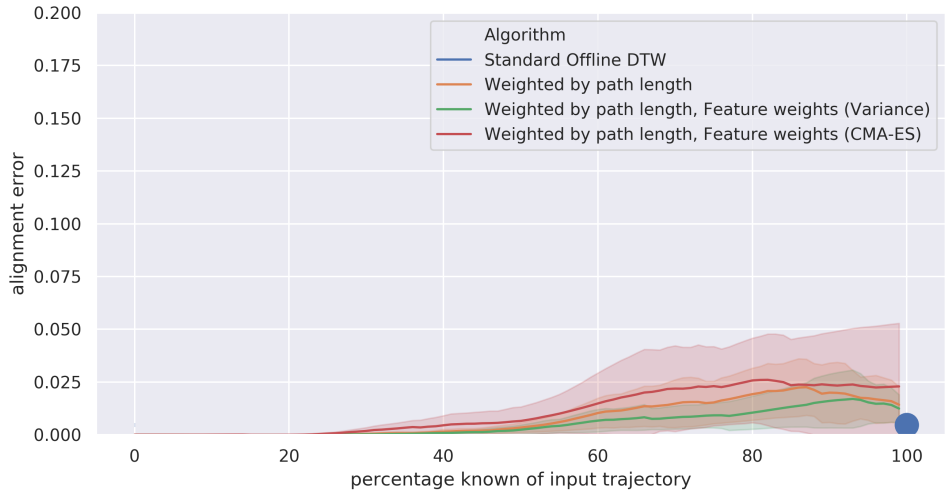
For each CV fold, we have to optimize the weights. As we cannot directly compute derivatives of the DTW process, we use a gradient-free method: Covariance Matrix Adaptation Evolution Strategy (CMA-ES) [HO97]. An advantage of evolutionary algorithms, besides not requiring a gradient, is their low susceptibility to ending-up in local optima. We decide to use CMA-ES, as the calculation of our error term is still expensive and CMA-ES only needs a comparatively low number of error evaluations during the optimization. A further advantage of CMA-ES is the small number of parameters: We only have to set initial weights $\mathbf{w}$ ($w_d = 1$ for all d), an initial step size (0.02), as well as the desired population size (parent population: 6, offspring population: 12). After preliminary tests, we decided to use 300 iterations for optimization.

CMA-ES needs a fitness function to rate the quality of each individual. To this end, we perform our weighted OE-DTW for each training trajectory and a reference trajectory, based on the weights to be evaluated. Then, we calculate the alignment error based on Equation (4.5) for all training trajectories and sum-up the results. For more details on how CMA-ES works, we refer to [Han16]. We use the CMA-ES implementation from the Shark library in version 3.1.0, which is a reference implementation of [Han16].

We extend OE-DTW with optimized joint weights and path-length weighting. We call the resulting algorithm Weight-Optimized Open-End DTW (WOOE-DTW). As baseline to analyze the influence of optimized joint weights on the alignment, we use OE-DTW with path-length weighting. Additionally, we compare our results to another type of feature weighting related to approaches such as [Ari+14; Cel+13; RDE11]: We quantify the influence of each joint on the motor action using its averaged variance over the whole movement: For each training trajectory, we calculate the variance of roll, pitch, and yaw of each joint over time. This variance is normalized by the maximum variance of all these features of the given trajectory. We then calculate the average of each feature over all training trajectories. For each joint, the corresponding weight is the maximum value for roll, pitch and yaw of the joint. For the OE-DTW that uses these variance-based weights, we also use path-length weighting. See Figure 4.6 for the results. Our optimized joint weights clearly achieve the best alignment quality for the squats, especially when only small prefixes of the input trajectory are known. However, for the Tai Chi pushes, the optimized joint weights fail to improve the alignment quality. One reason could be that the initial alignment is already very good and if only small prefixes of the Tai Chi pushes are known, the alignment is nearly perfect. Consequently the CMA-ES might not be able to find weights that can sufficiently improve the alignment.



(a) Squat.



(b) Tai Chi push.

Figure 4.6: Averaged impact of evolutionary optimized joint weights on the alignment error. For online versions of DTW, the plot shows the mean values over all cross-validation folds together with the standard deviation. For Standard DTW, the mean value over the cross-validation folds is marked.

For time measurements, we used a machine with Intel Core i7-7700K 4.2 GHz. The time needed to update the DTW matrices for a new frame only depends on the size of the reference trajectory and is constant during the whole process. The time to calculate the optimal path after the matrices are filled depends on the size of the alignment path which is always smaller than the sum of the lengths of the input trajectories. For the squat, on average, we need approximately 2.5 ms (4 ms for the Tai Chi push) for the alignment if WOOE-DTW sees 20 % of the motor action used as input, approximately 4 ms (9 ms for the Tai Chi push) if 60 % are available, and less than 6 ms (14 ms for the Tai Chi push) when WOOE-DTW knows the whole trajectory.

4.4 DISCUSSION AND CONCLUSION

In this chapter, we propose an extension of Open-End DTW to improve the online alignment of a reference movement with an incoming motion stream. We demonstrate that the alignment quality of Open-End DTW can strongly fall behind the alignment of the offline DTW. To explain this behavior, we carve out two reasons. One is the preference of DTW for shorter paths. We demonstrate that simple penalties for diagonal paths in the accumulated weight matrix, which are suggested in related literature, do not necessarily improve the alignment quality. To circumvent the bias of DTW, we propose path-length weighting and show its positive impact on the alignment quality. The other drawback of OE-DTW is the equal weighing of all joints: In real-world scenarios, some joints are more important for certain motor actions than others, which is not considered by DTW. We exploit our annotated training data to optimize weights for each joint using evolutionary optimization. We show that, for the squat, our extension WOOE-DTW improves the alignment quality of OE-DTW and beats variance-based joint weights. We reach a high alignment score that nearly reaches the performance of offline DTW. This shows that WOOE-DTW can provide good alignments even for such heterogeneous data as the performance of fitness exercises. However, for the Tai Chi push, only the path length weighting leads to an improvement. Our second extension, the evolutionary optimized joint weights, fail to improve the alignment quality. One reason could be the originally already very good alignment quality of the Tai Chi pushes that might be much harder to further improve. See the supplementary video of [Hül+17] for a comparison of the warps obtained by standard OE-DTW and WOOE-DTW and Figure 4.4 for an examplary comparison of warps obtained from OE-DTW and WOOE-DTW.

As WOOE-DTW calculates an online correspondence between the input prefix and a given reference, online segmentation of the input motion comes for free. Labels of the reference trajectory can directly be transferred to the input. When using the proposed approach for an online stream of arbitrary motion data, preceding to starting the WOOE-DTW calculation, the beginning of the motor action of interest must be detected. This can be, for instance, achieved via state-machine-based segmentation as proposed in the following chapter or by sliding-window-based classification [Cao+04].

Some approaches such as [VMM09] and [Car+15], which aim at the alignment of motion capture data, rely on algorithms other than DTW, partly to be less prone to noise and outliers. However, these two related approaches rely on parameters that have to be adjusted manually whereas our implementation does not have such critical parameters that directly affect alignment performance. Still, it might generally be interesting to compare the performance of these approaches to WOOE-DTW on our heterogeneous data set: This data is, as it consists of motion capture data, noisy and it contains outliers, as the motor actions can be performed with different styles.

Our extension of DTW requires manual labeling of training data. Indeed, the labels are as simple as “movement segment starts”. In our test scenario, we already obtain high accuracies using less than 100 annotated example movements per motor action. One limitation of our results is that we only use the squat and the Tai Chi push for evaluation. Even though both are comparatively complex motor actions and are

used in related approaches, further test cases with different kinds of motor actions and synthetically generated data would be desirable to strengthen our results. We assume that different motor actions will lead to similar results, depending on the initial alignment quality: We did not use any squat-/ or Tai Chi push-specific heuristics or tuning, but instead propose a data-driven optimization. Evaluating our approach for new motor actions only requires to annotate the movement segments in new training data and to run the evolutionary optimization to obtain the weights for the new motor action.

In the future, an integration and evaluation of WOOE-DTW in combination with other optimizations of DTW is worthwhile, such as Derivative DTW [KP01], Sakoe-Chuba Band [SC78], and Fast DTW [SC07]. Furthermore, the representation of the motion capture data itself is worth evaluating. In our work, we rely on raw data. Heloir et al. propose a PCA-based representation [Hel+06]. Although this requires the crucial part of the movement to be covered by the PCA, such an approach can be worth using for specific types of motor actions. Another representation that can be worth evaluating is based on Self-Organizing Maps [Den+11].

The proposed WOOE-DTW can lead to an improved alignment of a motor performance with a given reference trajectory. In the next chapter, we propose a pipeline to detect a trainee's errors during exercise that is designed to automatically generate feedback for the trainee. We propose a data-driven as well as a rule-based pipeline. The data-driven approach uses the online alignment proposed in this chapter to classify typical errors of the trainee already while the trainee performs an exercise.

5 CLASSIFICATION OF MOTOR ERRORS TO PROVIDE REAL-TIME FEEDBACK

Published in:
[Hül+16; Hül+18; Kok+15]

Obviously, feedback on the trainee's performance is crucial for the success of coaching systems. A coaching system has to assess the quality of the motor action performed by a trainee, and communicate this information in terms of feedback. Often, algorithms developed in the context of sports coaching either focus on the assessment of the performed motion, or on the generation of feedback. In this chapter, we propose an integrated pipeline that performs the detection of typical motor errors and provides results that are directly interpretable in terms of automatically generated augmented visual feedback. Further, results can be linked to already existing verbal feedback strategies.

In order to develop such an integrated solution, specific requirements, additional to a high classification quality, hold for the assessment of the trainee's performance:

R1 Connectable to existing feedback strategies: A coaching system should spot the occurrence of typical errors in the trainee's performance that can be linked to feedback strategies that have already been established by coaches in the real-world.

R2 Real-time: Whether feedback at early stages of the movement should be provided must be determined by the applied coaching strategies. However, to provide the coaching system a maximal range of applicability, components that assess the motor performance should deliver their results as soon as possible. If, for

Figure 5.1: In our real-time VR coaching environment, a trainee performs exercises while being observed by a virtual coach. Our algorithm provides the virtual coach with the information necessary to apply his feedback strategies in an online manner.

instance, the starting posture of a motor action is already problematic, the system should be able to intervene, to prevent the trainee from performing potentially problematic movement patterns, or even from hurting herself. For an analysis of real-world coaching and timing for the squat, I refer to [Hou+15; Kok+14; Kok+16].

R3 Interpretability: The classification process should be transparent and interpretable. It ideally provides information on the classified errors that can be used to generate augmented feedback in the virtual environment. Furthermore, an interpretable classifier gives experts the ability to verify whether the classifier works in a plausible way.

R4 Conservative size of data sets: Recording high quality training data and recruiting experts to perform data annotation is time consuming and expensive. Thus, the system should be able to deal with limited data sets to ensure practical usefulness of the coaching system.

R5 Minimal manual work: Manual work is expensive and reduces the usefulness of developed approaches in real-world applications. The classifier should require as few as possible manually coded expert knowledge.

Research in the area of VR that focuses on these aspects, thus keeps in mind the ideal integration of the kinematic movement analysis in a VR coaching system, would advance the field of sports and rehabilitation coaching in virtual environments. To this end, the contributions presented in this chapter are as follows:

- We develop a hierarchical representation of motion that serves as a basis for a *rule-based classification* of motion data which requires only a minimal amount of training data.
- We propose a second approach, an interpretable and real-time pipeline towards the *data-driven classification* of error patterns in motor performances. It uses a reference-based Dynamic Time Warping of movement prefixes as a basis for a feature selection using Random Forest (RF). The selected features are in a final step classified by Support Vector Machines (SVM).
- We demonstrate that the proposed pipeline can automatically generate real-time augmented feedback based on a trainee's motion. Further, the pipeline is integrated within our VR coaching environment (see Chapter 2) and connected to verbal as well as augmented visual feedback.

Both, our hierarchical representation together with the rule-based classification, as well as the data-driven pipeline, receive skeleton data (joint rotations, joint positions; cf. Chapter 2.5) as input to provide classification results, as well as augmented visual feedback in real-time. Due to using skeleton data, the pipeline can be applied in combination with various motion capture systems, as they typically output kinematic features for the tracked subject's joints. We demonstrate our hierarchical representation of motion as well as the rule-based classifier based on the squat movement as a test case. The data-driven approach is evaluated based on two data sets. They consist of body-weight squats and Tai Chi push movements. Based on these data sets, we show the ability of our pipeline to beat the popular classifier kNN-DTW (k-nearest-neighbor

Dynamic Time Warping) that has been found to be difficult to beat for typical time series classification tasks as shown in [BL14; Xi+06]. Further, we compare our pipeline to a recent neural-network-based approach to human activity recognition [Núñ+18]. The proposed pipeline does not only provide better classification results, but is also better suited to generate augmented visual feedback. Figure 5.1 shows a trainee and a virtual coach inside our setup. The coach provides feedback based on the results of this pipeline (see also the video in the supplementary material of [Hül+18]). Finally, we end this chapter with a discussion on which of our developed classifiers is most suitable depending on the context of application.

My Contribution *This chapter is based on the work of three publications. In [Hül+16; Kok+15] the hierarchical motion representation and the rule-based analysis was presented. In [Hül+16], in addition, its combination with an analysis of the trainee's mental representation of the motor action was presented. The hierarchical representation and the state-based analysis were developed by myself. My coauthor Cornelia Frank worked on the integration of the trainee's mental representation of the task which is not included in this thesis. In [Kok+15] the analysis is integrated in two studies: A pilot study served as a proof-of-concept demonstration of integrating the analysis inside the overall system. The other study used this analysis as an input for verbal feedback which was provided by a virtual coach. I developed the parts of the article that contain the description of the analysis as well as its integration into the coaching system. Further, I planned and realized the pilot study. I supported Iwan de Kok and Julian Hough in conducting the main study. Iwan de Kok and Julian Hough developed the virtual coach and planned and conducted the main study. Both studies are shortly described in Appendix A.3 and Appendix A.4 of this thesis. The article [Hül+18] contains the data-driven pipeline and the comparisons to related approaches. The development of the requirements, the conceptualization and development our approaches, as well as the comparison to related approaches were performed by myself. Jan Philip Göpfert provided knowledge on neural networks and their design and provided support for the comparison of our pipeline to the neural-network-based approach.*

5.1 RELATED APPROACHES

To assess the quality of human motor performances, two main approaches have been applied. The first approach (Section 5.1.1) is to engineer a highly specialized method, e.g., for the evaluation of feedback strategies for a very specific type of motor action. Often, a model for specific performance patterns is manually designed drawing from expert knowledge. The second direction (Section 5.1.2) consists in using more general, data-based approaches that have already been used in the context of motor learning and motion assessment. In Section 5.1.3, we focus on more general approaches from machine learning that have not been typically used in the field.

5.1.1 *Specific, Manually Designed Approaches*

Houmanfar et al. use a manually designed scoring function to represent patients' performance changes in a rehabilitation setting [HKK16]. Even though this approach provides compelling results in the field of application, no detailed information on occurred error patterns is gained, which would be necessary for the application of complex coaching strategies. Other approaches make use of rule-based systems to detect the occurrence of certain error patterns. In the context of yoga training, Rector et al. define optimal yoga poses [RBK13; Rec+17]. However, this approach is only based on static postures and does not take the whole motion trajectory into account. Hachaj et al. propose an approach to detect movements based on a rule-based system called gesture description language [HO14]. Here, the overall movement is detected via specifying subsequent key postures. However, Hachaj et al. do not specifiy a way to detect erroneous parts of movements [HO14]. Zhao et al. went further and propose a rule-based system to detect rehabilitation exercises and to measure occuring errors in the trainees' performances [Zha+17]. Rules are proposed that on the one hand quantify errors in terms of deviations from desired poses and on the other hand define specific parts of the movement that a trainee must pass through. However, error patterns that consist of the coexistence of specific patterns are not possible to implement. For instance even a simple error pattern such as going down too deep during a squat depends on the interplay of the knee angles of both legs. If the squat is performed, for instance, in an asymmetric way, one leg might violate the constraint while the other does not. Consequently the rules for an error pattern that specifies a too deep squat must be active as soon as one leg is flexed too much. However, for an inverse error pattern (reaching the desired depth), the movement should only be considered correct if both involved knee angles meet the desired angle. Further, the proposed approach cannot directly deal with errors that are only relevant during specific parts of the movement. This can be the case if, for instance, a body part should be held in a specific orientation during the first phases of the movement, but must be moved to a completely different orientation during the middle part. Finally, the proposed system does not allow for a decoupling between required movement segments and errors in motor performances. A deviation from a predefined movement part is considered as an error, however in some cases, such deviations could occur just due to different styles in performing the motion that do not impact the correctness of the performance as demonstrated by the authors in their evaluation. One major advantage of rule-based systems is their real-time capability (R2). Specific feedback strategies, linked to typical error patterns, can be applied immediately (R1) and the rules can be directly interpreted by experts (R3). Nearly no training data is needed (R4). Further, the results are deterministic. If the rules are correct and exhaustive, and the motion capture system works properly, an incorrect classification is unlikely to occur. However, the rules are designed manually which violates (R5). It is mostly not trivial—even when interviewing sports coaches—to obtain exact information about which features are significant or where to draw the border between a correct and an incorrect movement. And even if it is possible, the design of rules requires enormous manual effort. For each motor action and for each type of error, a detailed investigation on how to describe

the motor action and the error has to be performed. For complex error patterns, this quickly becomes infeasible. Still, for simpler patterns, and especially if no or only very few data is available, such an approach might be a worthwhile alternative. However, to our knowledge, no existing approach combines rules that define the course of a movement as well as error patterns that quantify typical errors that can occur during specific parts of the movement. Existing error patterns typically rely on deviations with respect to single joints and do not allow for an interplay between multiple features. Consequently, as one contribution of this chapter, we chose to develop an approach towards such a rule-based system. Here, we go beyond approaches such as the one proposed in [Zha+17], by distinguishing between movement segmentation and hierarchical error detection. Still, as such an approach is clearly not generally usable, we will also focus on approaches that automatically learn most of their information from data.

5.1.2 *Data-based Approaches for Performance Assessment*

Taylor et al. classify error patterns in rehabilitation exercises using a combination of rule-based segmentation and Adaptive Boosting on a set of manually defined features [Tay+10]. In a within-subject cross validation, the authors obtain highly convincing results. However, classification performance decreases significantly when generalizing to new subjects. Furthermore, the design of feature sets requires additional manual work. Yurtman et al. proposed an extension of Dynamic Time Warping (DTW) that is able to detect multiple occurrences of multiple exercise types in trajectories as well as to classify error patterns [YB14]. Classification is performed by comparing the just performed motion to pre-recorded templates and then selecting the best matching one similar to 1-nearest-neighbor Dynamic Time Warping (1NN-DTW). Combinations of multiple error patterns cannot be considered as long as they are not included as individually pre-recorded templates. Further, the authors did not test for inter-subject performance. Another prototype-based approach was described by Parisi et al. who propose a recursive neural network for the assessment of sports motion [PMW16]. As indicator for motion quality, the system compares the performed motion to the desired continuation of an exercise. Single-subject evaluation leads to very high accuracies, whereas tests with multiple subjects lead to a high number of false positives. O'Reilly et al. use a neural network classifier to differentiate between correct and incorrect performances of squats and to classify error patterns [ORe+15]. A leave-one-out cross validation resulted in an accuracy of 80 % to distinguish between correct and incorrect, but only in an accuracy of 57 % for the classification of error patterns. Similar experiments were conducted by Giggins et al. [GKC13; GSC14]. Brock et al. go further and propose an approach based on Convolutional Neural Networks (CNN) to classify errors in ski jumps [BOL17]. The proposed architectures obtain classification accuracy results from 69 % to 94 %. For some of the error patterns, the CNNs provide better results than classic methods such as SVM and Hidden Markov Model (HMM). However, the proposed classifier is only able to classify the performed movement as soon as it has been completely finished. Based on rehabilitation movements, Bevilacqua et al. evaluated different types of

standard classifiers (logistic regression, SVM, adaptive boosting, RF, decision trees) to distinguish between a good or an erroneous performance [Bev+18]. Movements were preprocessed, and manually defined features such as mean, skewness et cetera were determined. Best classification results (accuracies from 73 % to 98 %) were obtained using the SVMs and the RF. However, due to the features used for classification that are defined on the whole trajectory, classification can only be performed as soon as the movement is finished. Using weight shifting exercises, Vonstad et al. also extracted features from complete trajectories and classified whether the exercise was conducted correctly or incorrectly [Von+18]. Here, three classifiers (SVM, RF, kNN) were applied. The authors stated results of on average 95 % to 99 % accuracy per classifier. However, similar to [Bev+18], classification was only possible after completion of an exercise and no classification of specific error patterns was performed. Kianifar et al. present an approach towards distinguishing between good, moderate, and bad performances of squat movements [Kia+16]. They use a feature vector based on manually designed features, such as skewness and range, whose dimensionality is reduced using Sparse Principal Component Analysis (SPCA). Decision Trees are used for classification. The presented approach is only able to distinguish between three coarse classes of quality and cannot spot single error patterns. In addition, manual effort is needed for feature preparation. Furthermore, SPCA is an unsupervised algorithm, which searches for a set of sparse principal components that cover as much as possible of the variance inside the data [ZHT06]. This is problematic as most of the variance could be induced due to individual differences rather than performance errors. This might be especially risky for sports movements that can differ considerably between subjects.

Overall, the data-based approaches employed in the context of sports and rehabilitation applications have multiple weaknesses. First, many of them are unable to provide classification results before the movement of interest has been completely finished. In addition, three weaknesses in terms of classification performance are typical. First, it is often not analyzed how well the trained classifiers generalize to new subjects. Some of the addressed approaches require the system be re-trained for each user. This procedure can rarely be applied to real world coaching applications as subjects are often physically not able to provide all the required training data. Second, the motor actions and error patterns are often rather simple. Some systems only distinguish between, e.g., “good” or “bad” for a motor action that only involves a very small number of joints. Especially algorithms that use comparisons with prototypes will perform worse on more subtle errors or more complex movements when performing multi-subject evaluation as shown in [PMW16; Tay+10]. Here, different styles and differences between subjects might predominate differences induced by movement patterns underlying the motor errors. This holds especially as many types of complex sports movements can be executed correctly yet with different individual styles [HSG15]. Furthermore, an analysis that only relies on an overall deviation from a prerecorded desired performance, including task-irrelevant deviations, is non-optimal when aiming at improving the trainee’s performance [LT07; Sig+13]. One reason is that some muscle groups are often less requested than others. This makes these body parts less relevant when trying to successfully execute a movement.

5.1.3 *General Approaches for Human Activity Recognition*

Indeed, the classification of errors in motor performances is a special case of time series classification. In this area, ground-breaking work was performed by Wilson et al., who used HMMs for the recognition of gestures [WB99]. Other methods are based on decision trees [RA04], SVMs [WC04], or Multi-Layer Perceptrons (MLP) [NAM01]. DTW is usually applied to temporally align two recorded trajectories. As a pseudo-metric combined with a subsequent classification, DTW has a highly positive impact on motion classification [ARB08; Pet+14; Xi+06]. Xi et al. provide an extensive review comparing a large set of available classification methods, such as HMMs, MLPs, and decision trees on time series data [Xi+06]. They show that no tested classifier is able to beat a combination of DTW and 1-Nearest-Neighbor (1NN-DTW). 1NN-DTW compares the query trajectory to each available training trajectory using DTW as distance measure. Then the most similar training trajectory is used to predict the label of the query trajectory. The superiority of this approach in comparison with other classifiers, such as Random Forests, SVM, Bayes Networks, et cetera, is supported by work from Bagnall et al. [BL14]. Likewise, Yurtman et al. achieved good classification results using a method similar to 1NN-DTW, which, however, was limited to simple movement patterns and was not evaluated with respect to generalization to new subjects [YB14].

Recently neural networks have been frequently used in the related field of skeleton-based human activity recognition. They typically reach a high classification performance, especially for large training data sets. Recurrent Neural Networks (RNNs) allow for an online recognition of motor actions. For instance Li et al. propose a tree-like hierarchy of RNNs to distinguish between actions learned on thousands of sequences. [Liu+17a] focus on Long Short-Term Memory (LSTM) networks with trust gates to model temporal properties of the data. The classifier proposed by Liu et al. works on a combination of video and skeleton data [Liu+18a]. Here, data is preprocessed by convolutional layers to generate higher level features. The classification is then performed by an LSTM network and a combination of classification and regression layer. Other approaches focus on adding attention on temporal or spatial aspects into LSTM networks [Han+18; Liu+17b; Liu+18b; Zha+18]. For instance Liu et al. propose context-aware attention LSTM networks to allow the network to focus on informative joints for a specific motor action [Liu+17b; Liu+18b]. This is achieved via combining Spatio-Temporal LSTM layers with a dedicated global context memory. Han et al. go into a similar direction via enriching the LSTM architecture by a global spatial and temporal attention model to quantify different contributions of specific joints that vary over time [Han+18]. Other approaches focus on the deep preprocessing of the input to improve the classification performance. For instance Weng et al. propose a combination of convolutional neural network (CNN) and LSTM [Wen+18]. The CNN preprocesses the joint data and the LSTM covers the temporal aspects of the movement. A recent approach by Núñez et al. performs spatial as well as temporal preprocessing of the input to improve the classification performance [Núñ+18]. A CNN is used to preprocess the input, however it does not only process the data on the spatial, but also on the temporal temporal domain to generate higher-level features

that contain relevant information for the classification task. This information is then passed to a LSTM network to account for a larger temporal context.

Concerning the data-driven approaches from Section 5.1.2 (approaches that are already used in the field of performance assessment) and Section 5.1.3 (approaches from the more general field of human activity recognition), two approaches seem most suitable for our needs: kNN-DTW as well as the combination of CNNs with LSTMs (from now on called CNN-LSTM). The combination of nearest-neighbor classifiers with DTW is popular and difficult to beat in classic sequence classification [BL14; Xi+06]. Furthermore, related approaches have already been successfully used for the assessment of human motor performances [YB14]. CNN-LSTM has recently been proposed in the field of human activity recognition [Núñ+18]. Although the approach has not been demonstrated to work for subtle patterns such as errors in motor performances, the preprocessing step based on CNNs seems promising as it can be expected to learn the relevant features for specific error patterns. Furthermore, the well established field of CNNs provides methods to estimate the saliency of specific features of the input of the classifier [SVZ13], which would increase the interpretability (R3) of the approach. Both approaches can be linked to existing feedback strategies, as they are—given a sufficient classification quality—able to classify typical error patterns (R1). Further, they are fully data-driven and thus require only few manual work (R5). CNN-LSTM can be expected to work in real-time (R2). Further, this approach is described as being able to work even for small data sets (R4). However, both approaches have weaknesses: kNN-DTW suffers from high computational costs and CNN-LSTM can be suspected to suffer from few training data. Consequently, we propose a new pipeline towards error classification in sports exercises that is expected to suffer less from these issues. We demonstrate that our pipeline outperforms kNN-DTW as well as CNN-LSTM with respect to the quality of the classification as well as the classifiers' interpretability in terms of automatically generate augmented visual feedback. In the following, after shortly describing the data sets we used for the experiments, we first describe our rule-based approach in Chapter 5.3 followed by the data-driven approach in Chapter 5.4.

5.2 DOMAIN AND DATA SET

To build data sets for training and testing, we first identify error patterns for the squat as well as for the Tai Chi push via consulting coaches (squat: 14 coaches, on median 9 years of experience. Tai Chi push: 1 coach, 14 years of experience), literature (e.g., [CLS08], [CLH03]), as well as videos from coaching sessions (for the squat only, partly from corpus described in [Kok+14], partly recorded in our own lab). The recordings presented in Chapter 2.5 ($N = 96$ squats from 50 subjects, $N = 120$ Tai Chi pushes from 24 subjects) were annotated by an expert for the presence of any of the error patterns. The expert had to add an intensity rating for each error as well as confidence ratings for each decision. These ratings were combined into a score in the interval $[0, 1]$ by averaging. Only ratings with a score above 0.5 were used for the experiment. We selected the error patterns that appeared with a frequency of at least 15 positive and negative examples for training. The resulting patterns and their frequency in the

(a) Hips do not initiate the movement.

(b) Incorrect weight distribution.

(c) Hollow back.

(d) Too deep.

(e) Arched neck.

(f) Feet distance not sufficient.

(g) Wrong dynamics and feet distance not sufficient.

(h) Wrong dynamics and legs extended at end.

(i) Not symmetric and feet distance not sufficient.

Figure 5.2: The images depict examples and symptoms for error patterns of the squat mapped on a virtual character. These images can only provide a rough overview of how the errors could look like. Specific occurrences can deviate and require information on the rest of the movement. The pattern “knees tremble sideways” is not visualized as it is difficult to depict aspects of this error pattern in one single image.

training data are listed in Table 5.1 and Table 5.2. Figure 5.2 and Figure 5.3 provide a visual overview of the errors from typical recordings mapped on a virtual character.

5.3 HIERARCHICAL STATE-BASED MULTI-LEVEL ANALYSIS

Published in: [Hül+16; Kok+15]

The availability of annotated data that can be used to train algorithms towards classification of typical errors is limited. Consequently, approaches that require only few or even no training data are desirable. Ideally, we can exploit expert knowledge (e.g., obtained via interviewing coaches) to compile formal descriptions of kinematic features that describe typical errors. However, approaches based on already available knowledge are often oversimplified. These approaches perform the analysis of motor actions with respect to only a subset of particular aspects of motor actions: Often only simple features of the motion (e.g., joint angles) are considered and other important features such as relationships between joints or errors that are only relevant during specific phases of an exercise cannot be integrated into the model (cf. [RBK13; Zha+17]). As a detailed analysis is helpful for further steps, e.g., giving helpful feedback, this is one of the gaps we aim to diminish in this work. We develop a hierarchical representation that forms the basis of an online analysis of motor performances.

We propose a hierarchical representation that consists of four levels (cf. Figure 5.4). The highest level denotes the motor action that is performed. In the context of motor learning, this can be exercises, such as the squat. The next level consists of movement segments, which divide the motor action into smaller homogeneous sub sequences. For the squat, we use a subdivision into “preparation”, “going-down”, “is-down”, “going-up” and “wrap-up”. The next lowest level contains high-level features, for instance relationships between joints, information on velocity, or representations that differ from the representation of the raw data (such as Euler angles). On the lowest level, we have raw kinematic data (e.g., joint angles) obtained by motion capture (cf. Chapter 2.5). This representation has the advantage to allow focusing on error patterns on an extensible feature set which are especially relevant for single parts of

Table 5.1: Analyzed error patterns in the execution of a squat (cf. data from [Kok+14]). The numbers denote the quantity of incorrect and correct executions of the squat in our data, with respect to the corresponding pattern.

Performance Error Pattern	#Erroneous	#Correct
arched neck	33	29
feet distance not sufficient	45	33
hips do not initiate movement	23	51
hollow back	34	42
incorrect weight distribution	51	16
knees tremble sideways	23	33
legs extended at end	42	38
not symmetric	17	46
too deep	51	34
wrong dynamics	61	27

(a) Incorrect weight distribution.

(b) Right foot is incorrectly rotated.

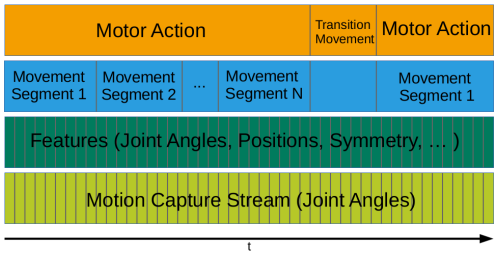
(c) Arms return at chest height.

(d) Knee is in front of the toes.

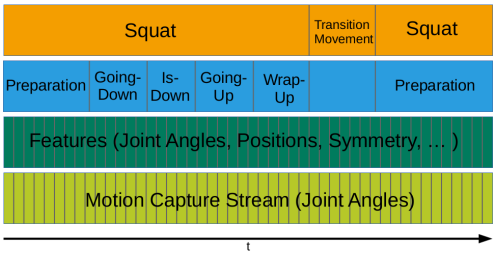
(e) The left leg is moved forward.

(f) Arms and legs are asynchronous.

Figure 5.3: The images depict examples and symptoms for error patterns of the Tai Chi push mapped onto a virtual character. These images can only provide a rough overview of how the errors could look like. Specific occurrences can deviate and require information on the rest of the movement. The patterns "non-uniform movement" and "asynchronous" are not visualized as it is difficult to depict aspects of these error patterns in one single image.



(a) General hierarchical motion representation.



(b) Exemplary representation for the squat.

Figure 5.4: Motion representation.

the action. Figure 5.4b describes the hierarchy exemplary for the squat. We build on this hierarchy to perform an efficient analysis of the performed motion. Section 5.3.1 describes how movement segments and motor actions are detected based on the lower levels of the hierarchy. Section 5.3.2 then describes the detection of performed errors based on knowledge on current motor action and movement segment. Here, we focus on the squat as an exemplary exercise. Developing detectors for further movements, such as the Tai Chi push would be theoretically possible, but is not done due to the time consuming extraction and evaluation of the movement-specific rules.

5.3.1 *Detection of Motor Action and Movement Segments*

Our real-time performance analysis is based on rules that describe the desired motion. This kind of analysis is highly efficient and allows a direct interpretation of the results in terms of performance flaws. For each type of action and movement segment, a list of relevant features is manually specified. Then, key-postures for the segments are defined, ideally via using manual analysis of recorded video and/or tracking data. To detect a single action, the system has to detect a posture similar enough to one of these key-postures. Motion segmentation works via using a state machine: Each motor action and its movement segments are represented as states. As soon as a posture inside a manually defined interval around the first key posture of the first segment of a motor action is detected, the analyzer switches its state. If the next posture is still valid for the current state the state machine remains at its state. If the posture belongs to the first key posture of the next movement segment, it switches its state to the next state. Otherwise, it assumes that the motor action has been aborted or has been incorrectly detected. Then it returns to the idle state. The current state reflects the current motor action and the current movement segment. See Figure 5.5 for an exemplary visualization of such a state machine. The next section describes how the information on current motor action and movement segment can be used as a basis for error detection.

Table 5.2: Analyzed error patterns in the execution of a Tai Chi push. The numbers denote the quantity of incorrect and correct executions of the Tai Chi push in our data, with respect to the corresponding pattern.

Performance	#Erroneous	#Correct
non-uniform movement	47	21
left leg moves forward	67	52
knee too much in front	23	65
incorrect weight distribution	17	64
backmost foot incorrectly rotated	39	65
asynchronous	35	39
arms return at chest height	16	73

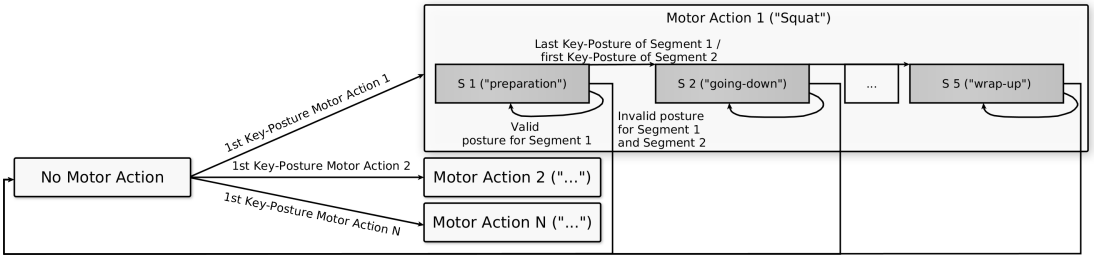


Figure 5.5: Exemplary state machine used to determine current action and movement segment for the squat.

5.3.2 *Detection of Performed Errors*

We are interested in providing advice on how to correct the movement. For example, to prevent the user from incorrectly distributing the weight, a possible way to provide feedback is to instruct the user to move their buttocks back. Thus, we would like to make use of a grounded specification of possible error patterns directly connected to the implications they have for the overall movement, and provide strategies to prevent the error. During different motor actions and different movement segments, different error patterns can occur. A posture or a part of a movement that is judged as erroneous during one movement segment does not need to lead to a movement error during another movement segment. Consequently, the motion detection takes into account the current movement segment to determine on which error types to focus. Error patterns, as obtained in Chapter 5.2, are formalized using at least one rule per pattern, which describes, e.g., the violation of specified constraints. Multiple rules can be combined to allow for an interplay between several features involved into the appearance of an error pattern. Each rule returns a quantitative error value for the incoming motion. The rules are developed based on literature (e.g., [CLS08; Esc01]) and information obtained from experts. Here, also observations from recorded data of correctly or incorrectly performed actions were taken into account. We define two types of rules:

Type 1 This rule becomes active (returns an error value) as soon as a given condition is violated (e.g., the bending of the neck during a squat performance exceeds a given interval).

Type 2 This rule stays active (returns an error value) as long as a given condition is not satisfied (e.g., the user does not go down deep enough during the whole squat).

As the development of rules to specify error patterns is time-consuming, we focus on a subset of error patterns that are important for the squat and easy to describe with rules and higher-level features. The ideal intervals can either be learned from data or be specified manually based on information from literature and experts. The following error patterns for non-optimal performances in the motor skill during squats are detected and can later be connected to verbal feedback:

Arched neck (Type 1) The angle of the skull-base and the curvature at the cervical vertebra are important to detect this pattern: If their angle gets too large, the pattern is activated and the largest deviation to the allowed interval of the joints is returned.

Feet distance not sufficient (Type 1) If the distance between the feet becomes too small, this rule is activated.

Hollow back (Type 1) To estimate a hollow back, the pattern takes into account the curvature of the thoracic vertebra. If this angle becomes too small (large value below zero), the rule becomes active.

Incorrect weight distribution (Type 1) One indicator for this error is that the knees are in front of the toes. In our experiments, this error is detected via observing the angle of shin and ankle. The error pattern returns the largest deviation to the allowed posture the trainee performed during the squat.

Knees tremble sideways (Type 1) During the movement, the knees tremble to the sides. We describe this error based on the change in the position of the knees from frame t to frame $t + 1$.

Not symmetric (Type 1) Parts of the left and the right side of the body are not in symmetry. We describe this error based on a higher-level feature that calculates the symmetry of a posture as the averaged quaternion distance between the rotations of the right and the mirrored rotations of the left side of the sagittal plane.

Too deep (Type 1) If the angle between the upper and the lower leg becomes too small, this error pattern becomes active.

Not deep enough (Type 2) In addition to the error patterns listed in Table 5.1, we specify a pattern that is used to determine if subjects went down deep enough to reach the desired depth of a squat. The goal is to achieve an angle of 100 degrees in the thigh position compared to the user's rest pose. This pattern is active until the user reaches the target joint angle. The return value quantifies the minimal deviation to the 100 degrees, the trainee reached during the squat.

Rules can be active for the whole squat (e.g., the pattern "arched neck") or only during specific phases of the squat (e.g., the pattern "not deep enough" is only active during "going-down" and "is-down"). This approach to the online analysis of motor performance allows a fast detection (approx. 1 ms) and does not need large amounts of annotated training data. Further, the rules can be used to extract augmented feedback such as color highlights of the joints that are involved in the detection of an error pattern. This is demonstrated in Chapter 6 for a combination of the error patterns "too deep" and "not deep enough". However, our manually hand-crafted classifiers are time-consuming to develop and thus violate requirement (R5), which demands few manual work. Further it is not possible to develop these hand-crafted classifiers for all types of errors. However, if they are available, they can mark a kind of gold standard to which a data-driven classifier can be compared. Additionally, if only a limited amount or nearly no training data is available they can be used as a first classifier until a sufficient amount of training data for data-driven systems is available. See

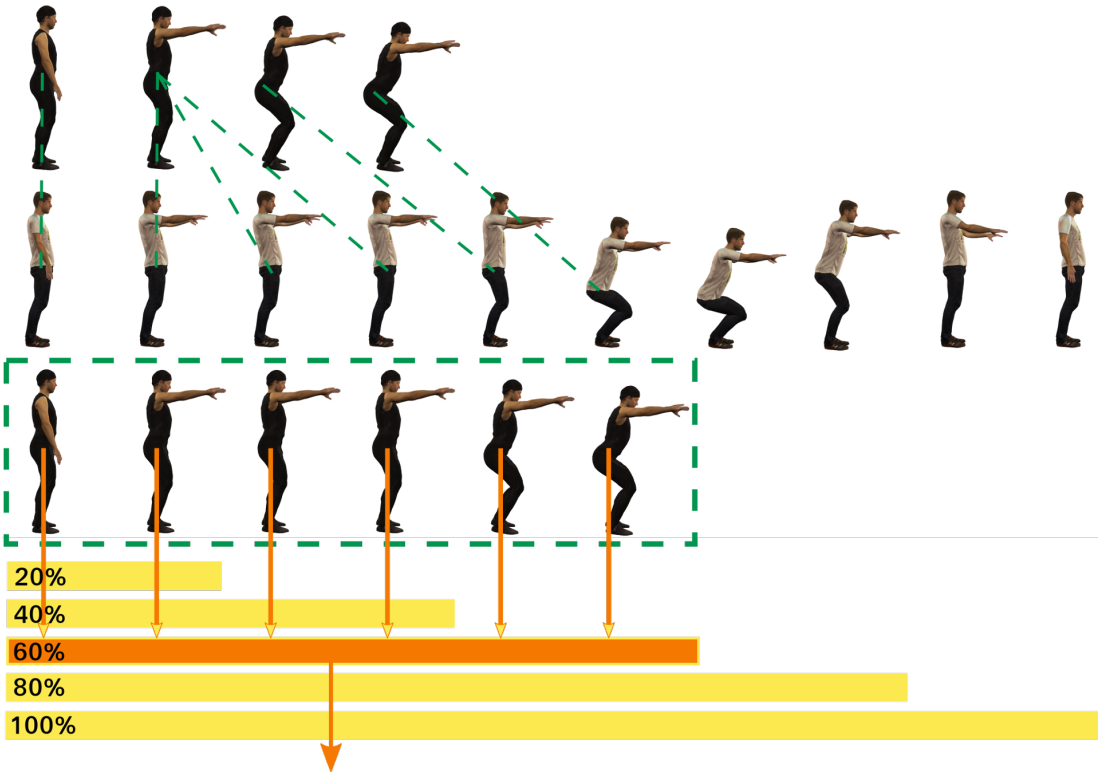


Figure 5.6: Online classification of error patterns: The trainee (upper row) has nearly reached the deepest point of the squat. The input trajectory is brought into correspondence (green dashed lines) with the reference using WOOE-DTW. Then the input is warped to the timing of the reference. The new warped trajectory (green box) corresponds to the first 60 percent of the reference trajectory. Thus, the classifier that is responsible for the first 0 % to 60 % of the reference is selected (orange) and performs the classification.

Chapter 5.5 for a quantitative evaluation of rule-based classifiers and a comparison to the classification quality of data-driven approaches. A discussion of reasonable scopes of applications of these detectors, also in comparison to data-driven approaches, can be found at the end of this chapter.

5.4 DATA-DRIVEN ANALYSIS

Published in:
[Hül+18]

5.4.1 *Classification*

Our classification pipeline is trained on the data described in Section 5.2. It learns a classifier for each error pattern, considering each training trajectory as one data point with the label *pattern occurs* or *pattern does not occur*. In the final application, the pipeline receives a stream of frames of skeleton data from a motion capture system and outputs a label w.r.t. each error pattern. As we use skeleton data, our pipeline

is highly flexible. The architecture is not restricted to specific input data, but can also be used with various motion capture algorithms, such as marker-based, but also marker-less ones, for instance [Cao+17; Meh+17a; Meh+17b; Wei+16].

In order to develop a preferably simple classifier that satisfies all our requirements, we rely on Support Vector Machines (SVMs). They are one of the most successful machine learning algorithms in general [Fer+14]. According to a recent review by Cust et al., SVMs are one of the most commonly used approaches for the classification of sports movements [Cus+18]. Additionally, they are fast and especially linear kernel SVMs are easy to interpret. For classification, the SVM only has to determine on which side of a hyperplane an input query lies. More technical and analytical information concerning SVMs can be found in [Bis06, p. 325].

In the context of motion trajectories, SVMs cannot be directly applied as they require input vectors of a fixed size. In order to represent all data on a canonical time line of fixed size, we exploit the general similarity between the trajectories that all represent the same motor action. We use DTW to warp all training and input trajectories into the timing of a fixed reference trajectory T_r. As reference trajectory, any arbitrary recording of the motor action of interest can be used. However, if the reference trajectory is too short (i.e., a very fast movement), information from the original trajectory can get lost due to the warping. Further, the selected trajectory should be prototypical for the motor action of interest and should contain as few as possible noise and tracking errors. For this thesis, the reference trajectories for the squat and for the Tai Chi push were determined manually after observing some candidate trajectories. For each frame t of T_r, the corresponding frame in the to-be-warped trajectory is extracted. Next, for these frames, we extract all joint angles in Euler angle representation as well as the joint positions. The resulting feature vector thus has size $6|T_r|k$, where $|T_r|$ is the number of frames of the reference trajectory and k the number of joints. We have $k = 19$ and $|T_r| = 902$ for the squat movement and $|T_r| = 782$ for the Tai Chi push.

The feature vector of size $6|T_r|k$ comprises many irrelevant features. For instance, we intuitively do not consider the rotation of the wrist to be related to having a straight back. The SVM classifier might suffer from this high number of irrelevant features as shown by Weston et al. [Wes+00] and Chen and Lin [CL06]. According to their results, we assume a robust feature selection method to be able to help improving classifier performance. A good introduction into the area of feature selection methods can be found in [GE03]. In the past, Random Forests (RF) have often demonstrated to lead to good feature selection results [CL06; GMS17; GPT10; Sve+04]. We use Random Forests as they tend to lead to especially good results for small sample sizes and a large number of features. Random Forests are based on Decision Trees, which learn a hierarchical set of rules to distinguish between classes. Thereby, they implicitly weight the importance of each feature. See [Bre01] for more analytical information on Random Forests. An in-depth analysis of the theoretical background and the statistical properties of Random Forests can be found in [Bia12]. Random Forests could be directly applied as classifiers, however classification using Random Forests leads to high computational cost, as all trees in the forest must be considered. Thus, we use a feature selection based on Random Forests as preprocessing for the SVM-based

classification during training. We train one Random Forest for each error pattern on the feature vectors extracted after DTW. To train the trees, we use the Gini impurity as criterion to optimize the decision rules [Bre+84]. As break condition for growing, we require all leaves to contain only a single class or less than two samples. We observed a number of 200 trees to lead to good results. For each error pattern, the Random Forest assigns an importance value to each feature via averaging the relative importance of the feature in each decision tree. Following an idea of Bi et al. [Bi+03], we add 10 random features to each frame before performing the feature weighting. The average of their importance values is used as threshold to discard irrelevant features. For the squat, this leads to 570 features on average per error pattern (from originally over 100,000 features). For the Tai Chi push, we end up with about 500 features. We use the implementation of Random Forests that is provided by scikit-learn [Ped+11] in version 0.17.1. For each error pattern, we train one two-class SVM with linear kernel on the selected features, which are standardized via scaling to unit variance and removing the mean. The implementation of the SVM is provided by scikit-learn. Formally, the classification is finally performed via evaluating the sign of:

$$\mathbf{w}^T \mathrm{fs}(\mathrm{warp}(\mathbf{T_x})) + b, \tag{5.1}$$

where $\mathbf{w}$ is the weight vector which specifies the orientation of the decision plane of the SVM and b is the bias which specifies the location of the decision surface. These parameters are trained by the SVM in the final step. The warping of input trajectory $\mathbf{T_x}$ into the timing of the reference trajectory $\mathbf{T_r}$ is denoted by warp and fs denotes the selection of the relevant features and the scaling required for the SVM classifier.

Due to the classic DTW, this classifier only starts the classification as soon as an exercise has been finished. In order to obtain a real-time classification, we provide two extensions to this procedure: First, we use Weight-Optimized Open-End DTW (WOOE-DTW), as proposed in Chapter 4, to make the temporal alignment work online. As a second extension, we train multiple classifiers on prefixes of our training data, to be able to select the best matching classifier for each point in time. In more detail, the training works as follows. First, all training trajectories are warped into the timing of the reference trajectory. Then, for each error pattern, we train the above classifiers on prefixes of the training trajectories in 5 % steps. The online classification looks as follows. A trainee has performed a part of the exercise, the input prefix. We warp this input prefix into the timing of the reference trajectory using WOOE-DTW. WOOE-DTW returns, additional to the alignment, the percentage c of the reference that corresponds to the input prefix. If $c \in [5\%, 10\%)$, we select the first of our classifiers, if $c \in [10\%, 15\%)$, we select the second, and so on. We apply the classifier on the part of the warped input that matches the prefix of the reference we used for training. See Figure 5.6 for a visualization of the classification procedure.

5.4.2 *Visual Augmented Feedback*

We provide feedback in terms of a visual augmentation of the trainee's avatar. Body parts that are related to a just performed error are highlighted in red. The manual selection of the important body parts as well as the point in time when they typically

contribute to an error is a time-consuming task. Consequently, we aim at extracting a visual highlight mask that provides temporal as well as spatial information using feature importance from our classification pipeline.

Our pipeline can easily be used to generate feedback, as the specification of the hyperplane of the linear SVM can be interpreted as importance values for each feature at each time step. The separating hyperplane is expressed by $\mathbf{w}^T\mathbf{x} + b = 0$, where $\mathbf{x}$ is the input. The components of $\mathbf{w}$ can be interpreted as importance values assigned to each feature. Based on this information, a visual highlight mask for each error pattern is calculated offline after training. It can then be applied inside the coaching application as soon as an error is detected.

First, joint importance is determined in two steps. The first one performs denoising for each joint. If a joint is considered important at a specific time step, but the temporal neighborhood is considered not important, the importance value is set to zero. Afterwards, for each joint, its importance values are summed-up over time leading to joint weights $\omega_j(k)$. Next, we calculate the final highlight mask and, as this mask can be precomputed, we smooth it to obtain better looking highlights. We set the values for all joints to zero whose joint importance $\omega_j(k)$ is smaller than 20 % of the largest value in $\omega_j(k)$. Then, for each frame, we sum-up all joint weights to obtain frame weights $\omega_f(t)$. These provide us with information on which point in time is in general important for the error pattern of interest. The frame weights are smoothed via applying two closing masks followed by an erosion mask. The final highlight mask $h(t,k)$ for each spatial feature k and each frame t (with respect to the canonical timeline) is then calculated by

$$h(t,k) = \begin{cases} 0, & \text{if } \omega_f(t)\omega_j(k)y(t) = 0 \\ 1, & \text{otherwise} \end{cases}. \tag{5.2}$$

Here, $y(t)$ denotes the binary label estimated by the classifier at frame t.

5.5 EVALUATION AND COMPARISONS OF CLASSIFIERS

5.5.1 *Classification*

We applied a 5-fold cross validation that aims at between-subjects testing and similar proportions of positive and negative labels in the folds as compared to the overall data set. We measure classification quality in terms of accuracy and F1 scores for the point in time where the classifier knows the whole input trajectory. Additionally to presenting the average classification performance per motor action and per error pattern, we check for significance on the level of error patterns. To this end, we perform a pairwise comparison of the classification success for all test trajectories by using the Wilcoxon signed-rank test with Bonferroni correction. The time measurements were conducted on a machine with Intel CPU Core i7-7700K 4.2 Ghz.

We compare our rule-based and data-driven approaches to kNN-DTW and CNN-LSTM. The overall classification quality of all tested data-driven approaches is visualized in Figure 5.7 for the squat and Figure 5.8 for the Tai Chi push. Concerning the

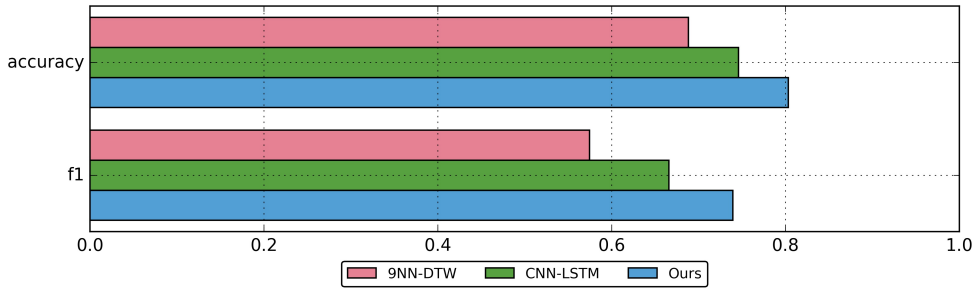


Figure 5.7: Averaged scores of the data-driven classifiers on the squat data set.

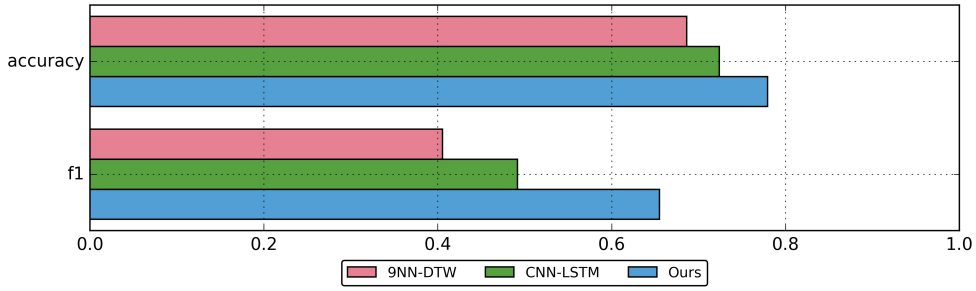


Figure 5.8: Averaged scores of the data-driven classifiers on the Tai Chi push data set.

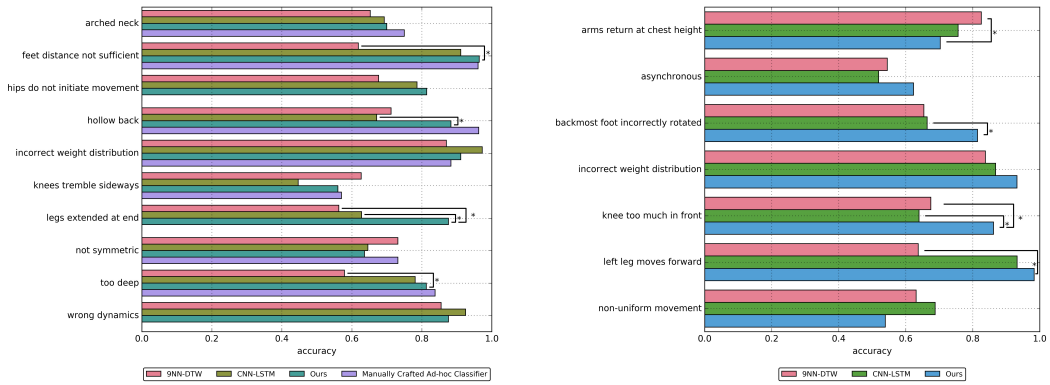


Figure 5.9: Accuracies of the classifiers for the squat (left) and the Tai Chi push (right) for each error pattern. Significant differences ($p < 0.05$) are marked with a star ($*$).

data-driven approaches, for both data sets, the worst results are obtained by kNN-DTW. The best results are obtained by our own classifier. CNN-LSTM lies in between. Additionally to the summarized classification quality of the data-driven approaches, we provide results for the individual error patterns. The accuracies for the single error patterns can be found in Figure 5.9. Concerning our data-driven pipeline, for all error patterns of the squat together, the warp as well as the classification itself need 5.2 ms on average. For the Tai Chi push, we need on average 6.6 ms to perform the single warp as well as the final classification for all error patterns. The timings for the other approaches are presented in the subsequent paragraphs.

Comparison to Rule-Based Classifiers

For the squat, we developed hand-crafted rule-based classifiers (cf. Section 5.3) for some of the error patterns. For the pattern "not symmetric", we defined the symmetry of a posture as the averaged quaternion distance between the rotations of the right and the mirrored rotations of the left side of the sagittal plane. To capture the trembling of the knees, we extract the lateral movement of the knees. For the other error patterns, we used manually selected joints and simple relationships between them as input. For the manually selected and preprocessed input features, we learn separating hyperplanes using a linear SVM based on the same cross validation folds as used in the experiments before. Our data-driven pipeline reaches a performance in a range similar to the results of the rule-based classifiers. However, our data-driven pipeline needs much less manual work and is not only restricted to posture-based patterns, but also takes the current point in time of an input motion into account. For the patterns "knees tremble sideways" and "not symmetric", which are not well classified by the data-driven approaches, results indicate that even manually crafted rules do not lead to better results. For the pattern "knees tremble sideways" we obtain an accuracy of 0.57 which is close to the accuracy of 0.56 obtained by our own data-driven pipeline. For the pattern "not symmetric", the manually crafted classifier obtains an accuracy of already 0.73 instead of 0.64, however the F1 score is zero. See Appendix A.2 for F1 scores for each error pattern.

Comparison to KNN-DTW

KNN-DTW is the combination of k-nearest-neighbors (kNN) as classification algorithm with Dynamic Time Warping (DTW) as distance measure. For an input query, kNN searches for the K data points that are most similar to the input. Then it returns their label, using majority vote. In order to classify a new query trajectory, kNN-DTW performs DTW with all trajectories, and then, for each error pattern of interest, returns the label of the closest trajectories that are annotated with respect to this error pattern. We use the DTW with path-length weighting as described in Chapter 4. For kNN, we select $K = 9$, as we observed this value to lead to best results (see Appendix A.2).

For the squat, 9NN-DTW leads to a classification performance of on average $accuracy = 0.69$, $F1 = 0.57$, wheras our pipeline reaches $accuracy = 0.8$, $F1 = 0.74$. Our pipeline leads to better accuracies than 9NN-DTW in eight of the ten error patterns. The differences are significant for the patterns "legs extended at end" ($p < 0.001$), "feet distance not sufficient" ($p < 0.001$), and "too deep" ($p = 0.003$). We observe a trend towards significance for the patterns "hollow back" ($p = 0.07$) and "hips do not initiate movement" ($p = 0.08$). Concerning the Tai Chi push, 9NN-DTW reaches $accuracy = 0.69$, $F1 = 0.41$ compared to $accuracy = 0.78$, $F1 = 0.65$. The accuracies of our pipeline are better in five of seven patterns. We observe significant differences between our pipeline and 9NN-DTW for the pattern "arms return at chest height" ($p = 0.03$, this is the only case, where one of the other approaches performs significantly better than our pipeline), "left leg moves forward" ($p < 0.001$), and "knee too much in front" ($p = 0.005$). We observe trends for the patterns "incorrect weight distribution" ($p = 0.08$) and "backmost foot incorrectly rotated" ($p = 0.09$).

For the squat as well as for Tai Chi, 9NN-DTW needs multiple seconds to calculate all necessary DTWs for the comparison.

Comparison to CNN-LSTM

The combination of Convolutional Neural Networks (CNN) and a Long Short-Term Memory (LSTM), as described by Núñez et al., is especially designed for the classification of human motion capture data [Núñ+18]. We therefore compare to their approach and give a brief description below. Fore more analytical insights and experiments on architecture and parameters, as well as figures that visualize the architecture, we refer to the original paper [Núñ+18]. Basic information on the underlying properties of CNNs and LSTMs can be found in [Goo+16].

The input movement is first processed by the CNN. The CNN learns a higher level representation of motion on the spatial as well as on the temporal domain via spatio-temporal convolution. Next, the preprocessed feature map is handled by the LSTM which covers the broader temporal context. The CNN proposed by Núñez et al. consists of six alternating Convolutional (ReLU activation; filter sizes: 20, 50, 100; kernel sizes: 3, 2, 3) and Pooling layers. The LSTM consists of 100 units. The training of the complete network, CNN-LSTM, consists of two steps. In the first step the weights of the CNN are pre-trained. The CNN is not yet connected to the LSTM, but to two densly connected layers (300 units, 100 units, ReLu), followed by an output layer with sigmoid activation. Time windows are separately fed into the network together with the label of the corresponding trajectory. The network is trained for 100 epochs with a batch size of 200. Next, as suggested by Núñez et al., the dense layers are cut off and the pretrained CNN is connected to the LSTM. Complete recordings of movements that consist of a sequence of time windows are now used as input. The new network is trained for 500 epochs with a batch size of 16 using Adadelta [Zei12]. According to [Núñ+18], due to the two-stage training, higher accuracies can be achieved compared to training the final network in one step. In our implementation, we use a window size of $T = 20$ according to the experiments performed in [Núñ+18] and we use a time shift of 10 that led to good results in our experiments. As input, we use joint translations as they led to better results than the combination of of translations and angles. We implemented the networks using Tensorflow [Mar+15] in version 1.6.0 and Keras [1] in version 2.1.5.

For the squat, CNN-LSTM leads to a classification performance of on average $accuracy = 0.75$, $F1 = 0.67$, wheras our pipeline reaches $accuracy = 0.8$, $F1 = 0.74$. For the squat, our pipeline leads to better accuracies than CNN-LSTM in seven of the ten error patterns. These differences are significant for the patterns "hollow back" ($p = 0.01$) and "legs extended at end" ($p < 0.001$). Concerning the Tai Chi push, CNN-LSTM reaches $accuracy = 0.72$, $F1 = 0.49$ compared to $accuracy = 0.78$, $F1 = 0.65$. The accuracies of our pipeline are better in five of seven patterns. Among them, "knee too much in front" ($p = 0.004$) and "backmost foot incorrectly rotated" ($p = 0.03$) lead to significant differences. For both motor actions, no error pattern is significantly better classified by CNN-LSTM than by our pipeline. CNN-LSTM needs approximately 8 ms

1 https://keras.io

for the classification of all error patterns of the squat. For the Tai Chi push, it needs around 7 ms on average.

Summary

In our summary, we only focus on the data-driven approaches as the rule-based ad-hoc classifiers need a high amount of manual work (R5) and as it is problematic to design them for all of the error patters. For the squat, our pipeline leads to best accuracies in six of the ten error patterns. In two cases, CNN-LSTM leads to best results, in two cases 9NN-DTW obtains best scores. Concerning the Tai Chi data set, our pipeline leads to best accuracies in five of seven patterns. One pattern is best classified by 9NN-DTW, one is best classified by CNN-LSTM.

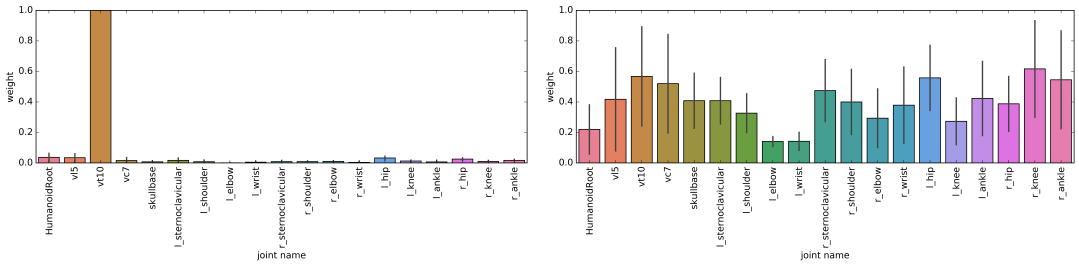
When using our own pipeline, we obtain the best averaged classification performance, followed by the CNN-LSTM, followed by kNN-DTW. More results concerning the evaluation of our pipeline can be found in Appendix A.2. All three approaches allow for the application of already existing feedback strategies linked to specific error patterns (R1). As soon as an error is detected, the corresponding feedback strategy can be triggered. The CNN-LSTM as well as our new pipeline work in real-time (R2). This is not the case for kNN-DTW, as the time needed for classification depends on the size of the training set, and is already large for one single comparison. All data-driven approaches work with small data sets (R4) and require only few manual work (R5), namely the labeling and recording of the training data. We will focus on the evaluation of the interpretability (R3) in terms of visual augmented feedback that can be generated in the next section.

5.5.2 *Visual Augmented Feedback*

We provide a comparison of visual feedback obtained by our pipeline to visual feedback we extract from the CNN-LSTM-based approach. First, we describe how the latter can be used to generate the desired highlight masks. For neural networks, saliency maps have been established to provide information on the importance of features in the input data [SVZ13]. They are calculated via deriving the output w.r.t. the input. We use the implementation provided by keras-vis [2] in version 0.4.1. As the input data for the CNN-LSTM-based approach consists of trajectories with different lengths and different timings, we cannot pre-process a fixed visual highlight mask for each classifier, but calculate the saliences for each input. We map the saliences to highlights if the error of interest is classified for the given point in time. Preprocessing is not performed as the saliency depends on the input movement a trainee performs, and these movements are of different lengths and have different timings.

In our evaluation, we first focus on the spatial dimension, namely the joints. We examine the joint importance values exemplary for the error patterns “hollow back” and “incorrect weight distribution”. For the “hollow back”, a straight posture of the back is important. In our body model, the flexion of the back is specified by joint vt10. Its flexion approximates the angle between the lower part of the upper back

2 https://raghakot.github.io/keras-vis/



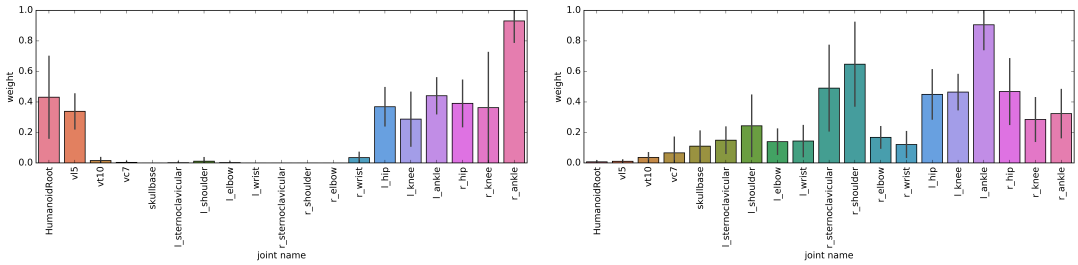
(a) Selection by our pipeline.

(b) Selection by the CNN-LSTM-based approach.

Figure 5.10: Comparison of selected joints for the error pattern "hollow back".

(thoracic spine) and the upper part of the lower back (lumbar spine). The error pattern "incorrect weight distribution" occurs, if the knees and the hips move too much to the anterior. Based on [CLS08], it is required that the knees are kept in line with the toes. Consequently, the whole lower part of the body can directly contribute to the error "incorrect weight distribution". Results for our pipeline are obtained during training and averaged over all cross validation folds. For the CNN-LSTM-based approach, we calculate the importance for all joints for each test trajectory that is correctly classified as erroneous. The resulting importance values are then averaged. For the "hollow back", Figure 5.10a contains the joint importances obtained by our pipeline and Figure 5.10b contains the results for the CNN-LSTM. The results for the pattern "incorrect weight distribution" can be found in Figure 5.11. For both patterns, there are joints that are similarly pointed out as important by both approaches, however, the results for the CNN-LSTM are less clear and tend to highlight joints that are not important for the given error pattern. Concerning the "incorrect weight distribution", the joints which obtained high values by our pipeline are mostly in the lower parts of the body which is in line with the theoretical information on the error patterns. These joints are mostly also selected by the CNN-LSTM, however, here, also parts of the upper body, such as sternoclavicular and shoulder are considered as important. This is problematic in terms of feedback, as the posture of the upper body w.r.t. the error pattern depends on the subject's proportions. A coach might want the subject to focus on the lower part and to automatically move the upper part in a suitable way to maintain a stable stand. Concerning the "hollow back", our pipeline selects exactly the joint that is important for the error pattern from a theoretical point of view, namely vt10. Concerning CNN-LSTM, also other less important joints, such as many joints in the lower body, obtain high values. To summarize, the joints selected by our pipeline are clearer and more suitable for visualization. Further, the joints selected by CNN-LSTM depend on the just performed movement, so selected features could vary for different inputs.

Next, we compare the overall quality of augmented feedback generated by both approaches. See Figure 5.12 and Figure 5.13 for exemplary visual highlights generated by both approaches. For the "hollow back" (cf. Figure 5.12a, 5.12b), the CNN-LSTM-based approach selects not only the back for the given example, but also joints in the lower part of the body, whereas our pipeline selects exactly the most relevant features,



(a) Selection by our pipeline.

(b) Selection by the CNN-LSTM-based approach.

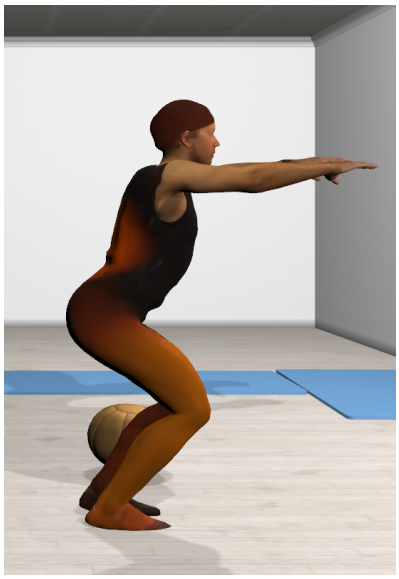
Figure 5.11: Comparison of selected joints for the error pattern "incorrect weight distribution".

namely the back. Concerning the "incorrect weight distribution" (cf. Figure 5.12c, 5.12d), the features selected by CNN-LSTM (left leg) for the input are a subset of the important features, however, other relevant joints, such as the right leg as well as the hips, are not selected. In contrast, our pipeline provides a much clearer highlight of the important joints. Concerning the error pattern "too deep" (cf. Figure 5.13), our comparison demonstrates, that even if the joints selected by CNN-LSTM are reasonable, the timing of the feedback can be problematic. Here, the highlight for the given trajectory is shortly activated already at the beginning of the movement, thus at a point in time that does not have a direct impact on the depth. In contrast, the highlights extracted from our pipeline are visible exactly when the subject is approaching the deepest point of the movement.

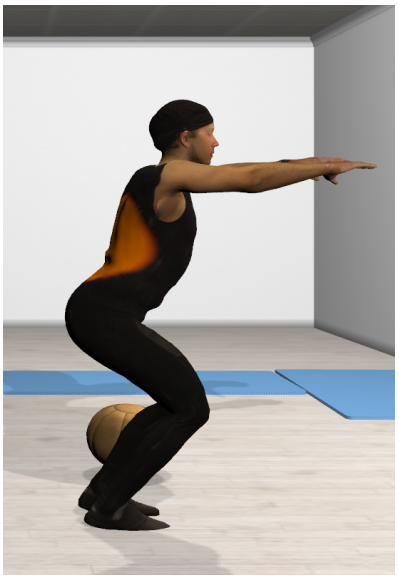
In summary, the highlights generated by our pipeline are more meaningful compared to the ones generated by CNN-LSTM. Additionally, the complete highlight mask can be precomputed and, if desired, manually checked for obvious errors (e.g., an activation of highlights for error patterns such as "hollow back" that occur at a point in time that is not sufficiently related to the error itself). When relying on the CNN-LSTM, highlight masks for single performances can work sufficiently well, whereas the highlight for other movements is problematic. Consequently, our pipeline better satisfies requirement (R3), the interpretability of the classifier. See the video in the supplementary material of [Hül+18] for exemplary visualizations of automatically generated augmented feedback for complete movements. Figure 5.14 gives an impression of how visual feedback provided by our virtual coach can look like.

5.6 DISCUSSION AND CONCLUSION

The focus of this chapter is on the assessment of motion performed by a trainee in a sports coaching environment in VR, using the squat and the Tai Chi push as test case. We had a special focus on the combination of error detection with the automatic generation of augmented feedback. To this end, we carved out proper requirements. Based on these requirements, we proposed two approaches that work online and provide results already while a trainee performs a motor action. The first one (see Chapter 5.3) can be used when no or only a little amount of training data is available.



(a) "hollow back" (CNN-LSTM): Additional to the back, the legs are undesirably highlighted.

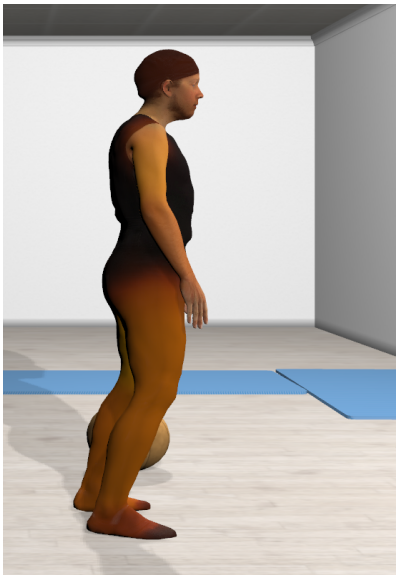


(b) "hollow back" (ours): The crucial part (the back) is highlighted.

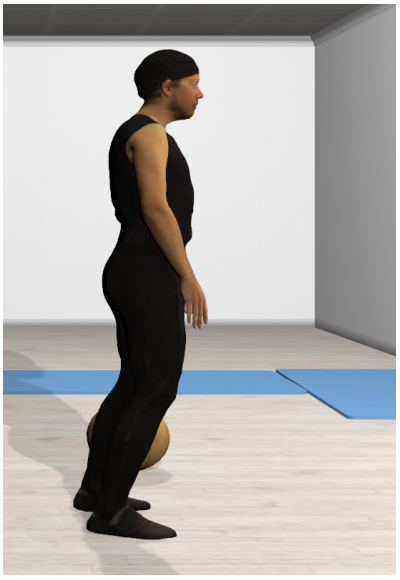
(c) "incorrect weight distribution" (CNN-LSTM): Only parts (left leg) of the relevant joints are highlighted.

(d) "incorrect weight distribution" (ours): The relevant parts in the lower body are highlighted.

Figure 5.12: Comparison of feedback generated by CNN-LSTM (a, c) and our pipeline (b, d). In (a, b) feedback for the error "hollow back" is visualized, in (c, d) the feedback for the error "incorrect weight distribution" is shown. If multiple error patterns occur at the same time, highlights are only shown for one of them.



(a) CNN-LSTM: Highlights are already visualized for a short period of time at the beginning of the movement which is undesired.



(b) Ours: As the point in time is not relevant for the error pattern, no highlights are shown.

(c) CNN-LSTM: Highlights are correctly enabled (highlights on the left side of the body are slightly brighter than on the right side).

(d) Ours: The highlights are correctly enabled.

Figure 5.13: Pattern "too deep": Comparison of feedback generated by CNN-LSTM and our pipeline at two different time steps. The beginning of the movement (a, b) is not relevant for the error pattern and should thus contain no highlights. The other time step (c, d) is relevant and should thus contain highlights.

(a) The hollow back is highlighted on the avatar inside the virtual mirror. The perspective of the mirror image is rotated to enable the user to observe his errors without the need to change his body's orientation.

(b) The user's last squat is replayed in the virtual mirror. The perspective of the mirror is rotated to enable the user to observe how the knees move in front of his toes which is one of the indications for the error pattern "incorrect weight distribution".

Figure 5.14: Desktop rendering of some of the feedback mechanisms that can be applied by the virtual coach.

Based on information obtained from experts and literature, rules can be designed in order to detect error patterns in a trainee's performance. The obtained rule-based classifiers can be used for instance to trigger specific feedback strategies. This could be, for instance, augmented feedback directly generated from the classifier: Based on the joints that contribute most to a rule, color highlights could inform a trainee that a specific error pattern has been observed. Even though rule-based classifiers are expected to lead to ideal results in case of very simple error patterns (e.g., in cases where only single joints are involved, as for instance for the error pattern "hollow-back"), this approach has a major drawback: The design of rules is a time-consuming task that is also prone to errors during the design of the rule and the selection of critical values. Consequently, rule-based classifiers can only be used for specific error patterns that are easy enough to formalize in a manually-designed rule set. Therefore we proposed a second data-driven approach. To this end, we introduced a new pipeline that satisfies our requirements and consists of two main parts: The classification of motor errors and the automatic generation of augmented feedback. We demonstrate that our pipeline is able to beat kNN-DTW as well as a recent neural-network-based approach [Núñ+18] in terms of classification performance and generated augmented visual feedback. Further, in cases where rule-based classifiers are appropriate and can be designed, it reaches a classification performance that is close to this gold standard. Our data-driven pipeline has been specifically designed to treat the special properties of motion data in order to classify typical errors in real-time. Consequently, known properties of the problem, such as the temporal warping or the feature selection, are covered by the architecture of the pipeline. The neural-network-based approach needs to learn most of these properties from the training data which could explain the superior performance of our pipeline. For the evaluation, we use two motor tasks, namely the squat and the Tai Chi push. The squat and Tai Chi push data sets used in this publication are publicly available via the DOI: *10.4119/unibi/2930611*. Table 5.3 lists the ability of the data-driven as well as of the rule-based pipeline to comply with our requirements. It can be used to guide the decision when it comes to setting-up a

Pipeline	Sufficient quality	R1: Link to Feedback	R2: RT	R3: Interpretable	R4: data	R5: little man. work
rule-based	yes*	yes	yes	yes	no data	no
data-driven	yes	yes	yes	yes	few data	yes

Table 5.3: Comparison of our rule-based and our data-driven pipeline with respect to the requirements that were carved out at the beginning of this chapter. R1: Connectable to existing feedback strategies, R2: Real-time, R3: Interpretability, R4: Conservative size of data sets, R5: Minimal manual work.
* if an error pattern can be easily described with rules, the classification quality is expected to be ideal. However, some error patterns are hard or impossible to describe with a bearable amount of work.

classification system for a motor learning applications.

Even though general classification performance of our pipeline is high, the performance is not convincing specifically for two error patterns for the squat and one for the Tai Chi push. The pattern "arms return at chest height" is classified with a very low F1 score ($F1 = 0.1$). A possible reason could be the immense imbalance between positive (16) and negative (73) examples in combination with the fairly complex error pattern. Concerning the squat, the error pattern "not symmetric" is detected with F1 scores only slightly above 0.4. This error pattern is annotated in trajectories where some joints are not symmetric between the left and the right side of the body. As this can occur in almost all joints and all phases of the movement, the feature selection cannot easily spot those features of interest that are relevant. For the other problematic pattern "knees tremble sideways" our results look similar. This pattern describes a very subtle movement. Also, it can spread temporarily: Exactly the frames that are problematic for subject A can be correct for subject B and vice versa. Finally, the number of trembles can be different for different subjects which also makes classification harder. Focusing on such patterns that are hard to classify, is a reasonable direction of future work, as here even a hand-crafted ad-hoc classifier was unable to obtain good classification results. One possible solution could be a combination of more complex higher-level features within our pipeline. Concerning the generated augmented feedback, note that the feedback we generate can only work if the classifier itself performs well. For error patterns such as "not symmetric" or "knees tremble sideways", the classifier is unreliable, thus also the selected features have no explanatory power.

A limitation of our pipeline is that temporal properties of the movements are not covered directly. However, for motor actions where the user's timing has an influence on whether certain errors occur, temporal information could be included via adding velocity as well as information on the warping function extracted from DTW. The list of error patterns and the annotated training data for the Tai Chi movement is only based on information from a single, albeit experienced coach and on literature. Taking into account information from more experts could further improve the developed model. Another interesting focus of future work could be the application of our pipeline to further challenging motor actions, such as dancing or martial arts. As we specifically designed our pipeline with a focus on dealing with error classification in

sports movements, we would assume similar results due to the general properties of the data. To enhance the overall performance of the classifier, one direction of future work could be improvements in the single components of the pipeline, for instance concerning extensions of DTW as well as an evaluation of further approaches towards feature selection such as the ones described in [Li+17a]. Concerning the augmented feedback, we even do not always need classification in order to provide feedback. In cases where just the attention of the trainee needs to be guided to the crucial parts of the movement with respect to a certain error pattern, we only need the first part of the pipeline, the temporal warping. Then, we could highlight the important joints based on Equation 5.2. One aspect of future work is to further investigate when to provide which amount of augmented feedback in order to help trainees in improving their motor performance.

In this chapter, we proposed two pipelines to detect a trainee's errors during exercise that are designed to automatically generate feedback for the trainee. However, we did not yet demonstrate whether these pipelines can be used in a coaching session to help athletes in improving their motor performances. In the next chapter, we combine all the components developed until now in a final system to provide athletes with an effective coaching session for the squat. This system is then evaluated in terms of a user study to demonstrate the successful integration and the interplay between our classification and feedback pipelines and the environment proposed in Chapter 2.

6 FULLY INTEGRATED ENVIRONMENT: COMPLETE COACHING CYCLE

First, a successful development of motor learning applications in VR requires concepts for the core parts of coaching applications. Especially the general environment (see Chapter 2) as well as suitable approaches to analyze a trainee's motion (see Chapter 5) are of high relevance. However, it must be evaluated whether such components can be used to induce an improvement in motor performances (e.g., via conducting user studies). Further, it is necessary to investigate whether and how single components can be combined to result in a final effective motor learning system. In an extensive experiment (see Chapter 3), we demonstrated that the environment proposed in Chapter 2 can be successfully used in the field of motor performance improvement. However, we did not yet combine it with the concepts developed in Chapter 5 in order to assess the trainee's performance during training. To this end, we first conducted two pilot studies. In one study, we demonstrated that participants already reacted to simple textual feedback that was generated based on the detection of errors in their performance. Details concerning this study can be found in the Appendix A.3. In a second pilot study, we connected the detection of error patterns with verbal feedback provided by a virtual coach. Here, we noticed that some participants were already able to adapt their performance in reaction to very simple verbal instructions by our virtual coach (see Appendix A.4). Still, we did not yet combine all the components developed in this work which are the coaching environment, both ways to classify errors in motor performances, the ability to generate visual feedback based on our classifiers, as well as further techniques to provide feedback. In this chapter, we combine and partly extend all these components in a final coaching environment. Our goal is to develop a system that combines all ways of feedback we can provide in the technically most ideal environment that is possible in our setup. The final application is able to control a complete coaching session for the squat. Multiple error patterns are addressed. The system decides on the basis of a typical real-world coaching cycle [Kok+14] as well as on the trainee's performance when to provide which feedback. To evaluate whether our combined setup is able to help people in improving their motor performance for the squat, we performed a user study in which we compared to the same system at an earlier stage without any provided feedback and only a simple generic avatar.

The contribution of this section is as follows:

- We verify that a combination of multiple ways to provide feedback and all the components developed in this work is possible.
- We demonstrate that the resulting system can lead to an improvement of motor performance demonstrated for the squat.

In our evaluation, we again use the squat movement with a focus on the error patterns "incorrect depth", "incorrect weight distribution", and "wrong dynamics".

My Contribution *The work presented here was done with some support from Cornelia Frank. To control the virtual coach, I used the coach system developed by Iwan de Kok and Julian Hough [Kok+15] (see Appendix A.4) as a basis and extended it according to the needs of this environment. Concerning the integration of the rendering, I was supported by Thomas Waltemate. I was responsible for the integration of all further technical components inside the final environment. I also developed the concept of the coaching session and the experimental design, based on the design developed in Chapter 3. Cornelia Frank supported in answering questions related to sports science in terms of timing of feedback and the importance of specific error patterns.*

6.1 REALIZATION

This section describes the improvements in the technical environment as well as the coaching cycle for our final coaching system.

6.1.1 *Technical Environment*

Our system is based on the environment already proposed in Chapter 2. To provide trainees with a realistic mirror image, we use fully animatable virtual avatars that have been automatically created from 3D scans according to [Ach+17]. For the virtual coach, we also use a character that has been created based on a 3D scan. All exercise-related movements that are performed by the virtual coach during a training session have been prerecorded. To get an ideal match between recorded performance and virtual character, the same person who provided the 3D scan performed the example movements. This person (age: 31 years) is an experienced athlete who trains the squat for around 5 years. To animate the coach, as well as for the decision making process and speech, we build upon the setup proposed in [Kok+15] as well as in Appendix A.4. See Figure 6.2 for a photo of a person who interacts with the environment. The system detects the error patterns "incorrect depth", "incorrect weight distribution", and "wrong dynamics". These are error patterns that have already been used in the preceding experiments and are expected to occur with a high frequency (cf. data set used in Section 5). Further we are able to classify these errors with a high accuracy (cf. Chapter 5.5). Concerning the error pattern "incorrect depth", it is clear which joints indicate the desired depth of the movement, namely the flexion of the knees. Further, squats should not reach 90 degrees flexion to prevent trainees from injuries due to incorrect techniques in combination with too much strain of the knees (cf. [Esc01]). Consequently, we use a rule-based classifier (see Chapter 5.3) to account for an incorrect depth of performed movements. The other error patterns ("incorrect weight distribution" and "wrong dynamics") are hard to model via hand-crafted rules. Consequently, we use the data-driven classifiers (see Chapter 5.4). We include multiple feedback strategies. First, we integrate color highlights that can be directly obtained from the classifiers for the error patterns "incorrect weight distribution" and "incorrect depth". For "incorrect depth", the highlighted body parts correspond to the joints that are used for the rule based classification (namely the flexion of the

knees). For "incorrect weight distribution", we use the highlight mask as described in Chapter 5.4.2. Further, we use replays of incorrect and correct performances in the virtual mirror, superimposed skilled performances in the virtual mirror (cf. Chapter 3), rotated virtual mirror (cf. Chapter 3), demonstrations performed by the coach, as well as verbal feedback.

6.1.2 *Coaching*

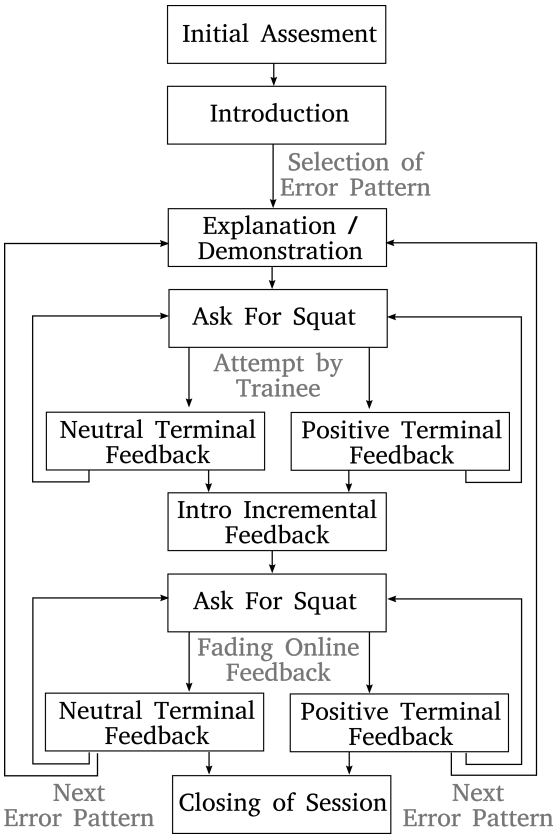


Figure 6.1: Coaching cycle that we use in our experiment.

There exist arbitrary many possibilities on how to structure coaching sessions. We learned from a pilot study on verbal feedback (cf. Appendix A.4) that addressing multiple error patterns simultaneously can be problematic. To obtain an overall good result and to avoid such issues, we chose to rely on a coaching cycle similar to the ones used in real-world coaching situations as described in [Kok+14]. After an initial assessment of the trainee's performance, a coach selects an error pattern to address and explains it to the trainee. After a demonstration and hints on how to improve the performance, the trainee is asked to perform the motor action of interest. The coach can provide feedback and further explanations during the movement (online feedback), but also after the movement has been finished (terminal feedback). Finally the coaching cycle is repeated until the coach selects a new error pattern of interest or the coaching session is terminated on behalf of the coach. To coach switches to a new error pattern only if the current error pattern of interest did not appear for a given number of consecutive trials.

Concerning the timing of feedback, we follow the coaching cycle proposed in [Kok+14] and include terminal as well as online feedback. As people tend to get dependent on feedback if they obtain it during every single trial [SW97], we chose to provide fading feedback. This means, that the system decides, depending on the trainee's performance, whether to provide feedback or not to provide feedback. The system then observes whether the trainee maintains a good performance when not obtaining feedback anymore. From our pilot study on verbal feedback (see Appendix A.4), we learned that sufficiently much detail is needed for the instructions to be understandable and to allow a trainee to improve the performance. Consequently, we provide

our virtual coach with the ability to explain various types of errors and to provide concurrent as well as terminal feedback together with some motivational utterances. The trainee gets precise information on how to prevent the motor errors of interest. In addition, we enable the virtual coach to provide information on specific types of errors that goes beyond simple verbal feedback. The coaching system can trigger replays of errors performed by a trainee together with explanations from the coach. Further the coach can demonstrate the correct performance together with some further information on how to prevent a specific error.

From the experiment presented in Chapter 3, we learned that superimposing a subject's performance with the performance of a skilled athlete can lead to improvements in terms of performing the movements more similar to the skilled athlete. From this experiment, we also learned that changes in perspective, such as viewing one's own avatar from the side, can further help with respect to motor learning. Consequently we apply these two ways to provide feedback (superimposed skilled performance, view from the side) in cases where they seem to be appropriate. In literature, color highlights and changes in appearance of a virtual character are often proposed [Hoa+16; PMW16; Sig+15; UKR14; VBG13]. Consequently, we chose to also include color highlights in our coaching application. The training for a specific error pattern is stopped as soon as four correct squats with respect to this pattern have been conducted in a row. If an error has been completely coached, the last four squats are taken into account to select the next error pattern that is to be coached according to the description above. The overall training ends as soon as either a maximum of 35 squats have been performed, or if all error patterns have been successfully coached, or if four squats are performed without any error in a row. As soon as the system decides to end the training, the coach says goodbye and provides a last standardized sentence with motivational feedback. In the following paragraphs, the coaching cycle as well as the provided feedback is described in detail. The coaching cycle is visualized in Figure 6.1. See Appendix A.5 for an overview of the utterances provided by the virtual coach. The supplementary video of [Hül+18] and the figures of Chapter 5 provide an overview on how some of the applied feedback strategies look like.

Introduction Phase At the beginning of the coaching session, the virtual coach welcomes the subject. Then, from an initial assessment performed before the actual coaching session (10 squats, see Chapter 3.2.2), the system selects the first pattern to be coached. For each pattern the number of occurrences during the initial assessment is counted. The pattern that was active most is selected. If two patterns have occurred for the same amount of times, the pattern to be coached is selected as follows: highest priority: "incorrect depth", middle priority: "incorrect weight distribution", lowest priority: "wrong dynamics". After the pattern of interest has been selected, the coach explains the error pattern. Then, if either the pattern "incorrect depth" or the pattern "wrong dynamics" is selected, the coach demonstrates a correct performance together with information on how to improve. For the pattern "incorrect weight distribution", the coach initiates a replay of the worst squat performed by the athlete in the virtual mirror, followed by a replay of a better performance, mapped on the subject's body, also in the virtual mirror. In addition, the coach provides information on how to

prevent the error pattern.

Terminal Feedback Phase After this initial explanation and demonstration, the athlete is asked to perform a squat. After this squat has been completed, terminal verbal feedback is provided by the coach. This is done for three squats in a row. If a correct squat has been performed or if the last squat was better then the previous one, the coach acknowledges this with a short utterance. For the pattern "incorrect depth", verbal feedback is provided whether the last squat was too deep or not deep enough. During the blocks with terminal feedback, the mirror is only rotated for the error "incorrect weight distribution". Even though, also the other error patterns can be observed easier from the side, in these cases, the error is also visible from the front, so we do not rotate the mirror directly at the beginning to prevent overloading subjects with too much information at the same time.

Incremental Feedback Phase After the block with terminal feedback is completed, the coach switches to faded concurrent feedback. Feedback is first provided online during each squat performance. Then, as soon as an improvement is observed over the last three squats, a single squat without obtaining feedback has to be performed. Then, feedback is provided again. If the subject is able to further improve, feedback is disabled for two squats. This procedure is repeated until the pattern is completely coached. We use faded feedback to prevent the athlete from becoming dependent on the feedback. If this would happen, the athlete would improve during acquisition. However, later on, in the post-test, the performance could drop back to a level similar to the pre-test [SW97]. The actual feedback looks as follows. For the pattern "incorrect depth" verbal feedback is combined with visual feedback. The coach instructs the athlete during going down via asking to go deeper. As soon as the desired depth is reached, the coach asks the participant to stop and to go up again. Together with the verbal feedback, the legs of the athlete are colored in red until the desired depth is reached. As soon as the athlete goes down too deep the legs are again colored. The opacity of the color is a linear function of the intensity of the error. In addition, after the squat, the coach tells the athlete whether the squat was too deep or not deep enough and provides motivational feedback (e.g., via saying, in German, "You are on the right track"). For the pattern "incorrect weight distribution", the highlight mask obtained from the classifier (cf. Section 5.4) is used to provide feedback. As soon as the error occurs, the corresponding joints are highlighted. See Section 5.4 for further details on how the highlight is calculated. For the error pattern "wrong dynamics", a replay of a correct movement with respect to this error pattern is shown as overlay on the virtual avatar of the athlete. The virtual coach asks the athlete to perform the movement as similarly as possible to the overlay with respect to the simultaneity of the arms and the legs. For all error patterns, during the repetitions with feedback, the mirror is rotated.

Figure 6.2 shows an athlete who interacts with our coaching environment. The avatar of the athlete as well as the coach character are reconstructed from 3D scans.

Figure 6.2: An athlete interacts with our coaching environment.

6.2 EXPERIMENT

To evaluate the effectiveness of the developed system, we perform an experiment with a 2×2 mixed factorial design. We use testing time (pre-test, before training with our system; post-test, after training with our system) as within-subject factor and condition (feedback, baseline) as between-subject factor. Overall, we had 41 subjects in two groups (21 males; age $M = 25.9$, $SD = 4.1$; group feedback: $n = 21$; group baseline: $n = 20$). We recorded additional 6 subjects, but did not include them for technical reasons. The baseline group had already been recorded in earlier experiments. Participants provided written informed consent and got paid for their participation. The study was conducted in accordance with the Declaration of Helsinki, and had ethical approval from the ethics committee of Bielefeld University.

We measured the improvement between pre- and post-test with respect to each error pattern. For the rule-based error pattern "incorrect depth", we measured the error value in degrees. For the other error patterns, the error was quantified in terms of the obtained distance to the decision plane of the linear SVM classifier. Similar to the experiments presented before, feedback was provided between the pre- and the post-test. For the full system group, we used our full system in this acquisition phase. For the baseline group, in the acquisition phase, we used an earlier version of the system that used a stick figure instead of a 3D scan of the subject (cf. experiment presented in Chapter 3). Despite from the virtual mirror, no feedback was provided to participants in the baseline group. These participants just performed 30 squats in front of the virtual mirror. A timer informed them when to perform a squat. The overall procedure was similar to the one used in the afore described experiments. Participants in the full system group were 3D scanned at the beginning of the procedure.

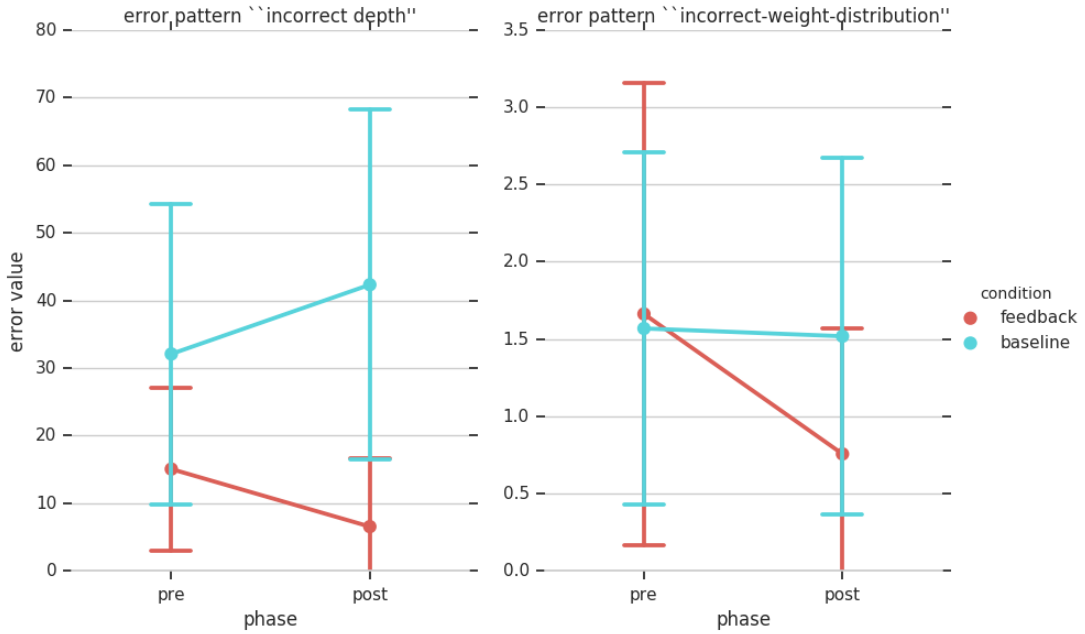


Figure 6.3: Motor performance results for the error pattern "incorrect weight distribution" (distance to decision plane) and "incorrect depth" (degrees). The graphs show the effect of the feedback provided to the different groups on motor performance.

6.3 RESULTS AND DISCUSSION

For each parameter, a two-way mixed analysis of variance (ANOVA) was conducted with phase (pre-test, post-test) as within-subject factor and group (feedback, baseline) as between-subject factor. For all analyses, the level of significance was set at $p < 0.05$. We restricted our analysis to the patterns "incorrect depth" as well as "incorrect weight distribution". We did not analyze the pattern "wrong movement dynamics" as it was coached for 3 participants only. For the pattern incorrect weight distribution, the error values indicate the distance to the decision hyper plane (cf. Section 5.4), for the pattern "incorrect depth", the error values are provided in degrees. The results can be found in Figure 6.3.

For the pattern "incorrect weight distribution" results showed no significant main effect of condition ($F_{1,39} = 0.932$, $p = 0.34$, $\eta_p^2 = 0.023$) on performance. For phase, the main effect was significant ($F_{1,39} = 10.78, p = 0.002, \eta_p^2 = 0.217$). Participants performed better in the post-test ($M = 1.13, SD = 1.08$) than in the pre-test ($M = 1.61, SD = 1.35$). There was a significant interaction between condition and phase ($F_{1,39} = 8.266, p = 0.007, \eta_p^2 = 0.175$). Descriptive statistics showed that subjects who obtained feedback obtained better scores in the post-test ($M = 0.76, SD = 0.83$) as compared to the pre-test ($M = 1.66, SD = 1.53$). Subjects in the baseline group showed similar results for post- and pre-test ($M_{pre} = 1.57, SD = 1.17; M_{post} = 1.52, SD = 1.18$). Consequently, for the pattern "incorrect weight distribution", the feedback as provided by our system is able to help participants improving their squat performance.

For the pattern "incorrect depth" results showed a significant main effect of condition ($F_{1,39} = 23.65, p < 0.001, \eta_p^2 = 0.378$) on performance. Subjects in the baseline

group ($M = 37.2, SD = 24.96$) performed worse than subjects in the feedback group ($M = 10.79, SD = 12.05$). For phase, results showed no significant main effect ($F_{1,39} = 0.066, p = 0.799, \eta_p^2 = 0.002$). There was a significant interaction between condition and phase ($F_{1,39} = 14.4, p < 0.001, \eta_p^2 = 0.27$). Descriptive statistics showed that subjects who obtained feedback obtained better scores in the post-test ($M = 6.53, SD = 10.32$) as compared to the pre-test ($M = 15.05, SD = 12.37$). Subjects who were in the baseline group did not improve ($M_{pre} = 32.07, SD = 22.78; M_{post} = 42.32, SD = 26.55$). Consequently, for the pattern "incorrect depth", the feedback as provided by our system is able to help participants improving their squat performance.

6.4 CONCLUSION

Our results indicate that our approach towards a combination of all components developed in this work is possible and has been successfully conducted. Further, results also indicate that the developed system is able to help athletes in improving their squat performance. For the error patterns "incorrect weight distribution" and "incorrect depth", we observe a significant improvement over time as compared to the baseline group. Further, both ways we use to classify error patterns in motor performances (rule-based as well as data-driven) are demonstrated as being usable in coaching environments.

In this work, we demonstrated an effective system for motor learning in VR. However, the equipment that is necessary for our setup is expensive and not portable. Being able to scale down parts of our work is desirable. Consequently, in a short outlook, we will propose a scaled down application that uses the same architecture and parts of our full system, but is portable and only uses low-cost hardware. We use this environment to evaluate the effectiveness of augmented feedback in terms of color highlights that are shown on the athlete's avatar. Finally, we will discuss in which situations a scaled down environment can be considered for motor learning applications.

7 OUTLOOK: PORTABLE ENVIRONMENT

Live demo of parts of this environment published in: [Kok+17]

While our environment as described in Chapter 2 is a state-of-the-art setup with low latency and robust motion capture, it is complex and experiments are time consuming. For instance, the marker setup alone takes 20 to 30 minutes per participant. Further, the environment cannot be transported, e.g., to gyms where participants could use the environment to train specific tasks. Despite the fact that properties such as low latency and high robustness are important for VR environments that target at motor learning, minimal portable setups, e.g., a combination of Microsoft Kinect and consumer HMDs have strong advantages: Trainees do not need to visit a specific location where the setup is installed, but could train at home. Further, such setups are much cheaper, so that they could even be used in practice and for long term experiments (for instance when participants are allowed to train with the system at home on a regular basis). In addition, when using consumer motion capture (e.g., the Microsoft Kinect), time for preparing the subject for experiments can be reduced as attaching markers to the participants is not necessary anymore. Here, also other marker-less ready-to-use approaches such as the system developed by The Captury [1] could be applied. However such commercial solutions are typically expensive. Another option would be to build upon techniques such as the ones proposed in [Cao+17; Mat+18; Meh+17a; Meh+17b; Wei+16]. Still, devices such as the Kinect are cheap, can be easily integrated, and already proved to be suitable for an at-home training as for instance the Kinect is originally developed for the integration of full-body motion in video games. Concerning the display technology, even though HMDs are becoming more popular, they are, compared to other techniques, such as 2D screens or large scale installations seldom used in the context of sports training and motor learning [Neu+18]. Some possible reasons are that they might be impractical for specific exercises (e.g., jumping), head movements might be uncomfortable due to cables attached to the HMD and sweating becomes a problem when performing exertive tasks [Neu+18]. However, HMDs provide stereo 3D which is not the case for many standard 2D screens. They can be expected to be much more immersive than simple screens. In an experiment conducted by Waltemate et al., HMDs provide even better results than an L-shaped CAVE environment in terms of important variables such as body ownership and presence [Wal+18]. Consequently, as HMDs become more lightweight and more and more consumer systems for motion capture are developed, more systems emerge that show a positive impact of combinations of consumer techniques on improving motor performance.

This chapter provides a short outlook on how some of the components developed in this thesis could be scaled down to be used in a consumer environment for specific well-defined applications. The resulting environment is used in a pilot experiment in order to show possible promising developments. Further, we use this pilot experiment

1 http://thecaptury.com/

to investigate the ability of augmented feedback in terms of color highlights to be suitable to improve trainees' motor performance. Our contribution to the state of the art is as follows:

- We demonstrate that it is possible to set up a portable version of our coaching environment based on consumer hardware if some of the requirements proposed in Chapter 2 carry less weight for the specific field of application. Further, we show that this reduced environment can already be used to reduce motor errors in specific cases.
- We demonstrate the effectiveness of augmented feedback in terms of color highlights for specific motor errors in a pilot experiment.

As a motor action to evaluate our environment, we use the lateral lift. In this chapter, portable means that all parts of the system can be easily detached and set up in a short period of time at a new place. We demonstrated the portability of all hardware parts of the environment described in the following during the demo session of the International Conference on Intelligent Virtual Agents in 2017 [Kok+17].

My Contribution *The work presented in this chapter was done in collaboration with Thomas Waltemate, Holger Bienek, Yannic Wietler, Robert Feldhans, and Iwan de Kok. I developed the concepts for the overall design of the environment together with Thomas Waltemate. Further, I developed the design for the application as well as for the experiment. In addition, I implemented parts of the procedure of the experiment. Yannic Wietler implemented the integration of the Kinect camera, Robert Feldhans implemented the integration of the HMD, and Holger Bienek implemented extensions of the Kinect setup as well as the integration of the provided feedback. Further, he developed the controller of the experiment and conducted the experiment. Some adaptations that were needed to implement the feedback inside the renderer were implemented by Thomas Waltemate. The virtual coach that is addressed in the emerging live demo [Kok+17] was developed by Iwan de Kok, but is not described in this chapter.*

7.1 RELATED WORK

This section first summarizes related work in the field of portable and consumer setups towards motor learning. In the second paragraph, we focus on augmented feedback in terms of color highlights and changes in appearance. Arndt et al. propose a VR environment for rowing [APV18]. Participants sit on a rowing machine and wear a HMD that visualizes the trainees movement inside a lake combined with information on the trainee's performance visualized in the display. A more complex setup that also contains tracking of motion that is mapped on a virtual avatar is propsed by Han et al. who use a combination of Oculus Rift, Myo sensors, and Leap Motion to guide people performing specific arm movements [Han+16]. Hoang et al. perform learning of Tai Chi movements via a combination of color highlights and imitating the movement of a trainee in an environment based on Oculus Rift and Kinect[Hoa+16]. Cannavò et al. develop a system towards the training of the basketball free throw [Can+18]. They combine HTC Vive, Perception Neuron motion capture suit, HTC Vive trackers as well as the Kinect 2 camera. For rendering, they use a game engine. The field of

application of these approaches is quite restricted: They are either developed for very specific tasks (e.g., rowing, basketball free throw, arm movements) [APV18; Can+18; Han+16], or do not comprise a complex analysis of the performed motion that can be connected to various ways to provide feedback. However, These articles suggest that systems even with minimal technical equipment such as HMD and Kinect tend to be usable for very specific use cases in the field of motor learning, even though they do not fulfill all requirements developed in Chapter 2. Even though the main focus of our work is on investigating the basics of how motion data can be used to support motor learning, we rate it as helpful for further research to get a first glance on how our state-of-the-art environment could be scaled down to a consumer setup that might even be used in practice. Consequently, we chose to develop a minimal version of our training environment. This environment might not reach the high quality standards and requirements developed in Chapter 2, but it might already be able to support people in learning specific motor actions. We conduct a pilot experiment to verify whether down-scaling of our environment is possible at least for specific fields of application. As we know that low latency and robustness are important for motor learning applications (cf. Chapter 2), we chose to focus on a task that can be expected to be minimally prone to these properties. We chose to conduct our experiment using the lateral raise as motor action.

Color highlights and changes in appearance (e.g., changes in opacity) have been frequently used in the field of motor learning in VR (e.g., [Hoa+16; PMW16; Sig+15; UKR14; VBG13]. For instance, [VBG13] combine demonstrations of the target movement, color highlights, as well as further visual feedback in order to improve motor performance. However, they do not evaluate whether their system is able to induce motor learning. Ukita et al. propse a combination of, amongst others, colored joint overlays, side-by-side views and animated joints that automatically move into the right direction [UKR14]. An evaluation of the proposed feedback has not been conducted. Parisi et al. suggest a way to automatically extract color highlights from a trained classifier. However, despite problems concerning the classification quality between-subjects, they do not evaluate the quality of the obtained color highlights neither do they evaluate whether these highlights can support trainees in improving their motor performance [PMW16]. In other cases, the impact of combinations of various feedback methods on motor performance has been evaluated. For instance, Sigrist et al. changed the transparency of a rowing oar depending on its distance to the ideal performance. Further, the ideal performance was visualized via a trace and a superimposed target performance [Sig+15]. Hoang et al. superimpose a target performance on top of the trainee's performance together with switching the color of the parts of the avatars as soon as its movement is similar enough to the target movement [Hoa+16]. In these examples the provided feedback indicates to be able to help subjects in improving their motor performance. However, the authors do not separately evaluate the effectiveness of all parts of the provided feedback. Consequently, it is unclear how much the color highlights, the superimposed target movement or some other features of the movement influence the improvement of the subjects performance.

We chose to evaluate one specific way to provide feedback, namely the use of color highlights to guide the subject. For this experiment, we used a 2×2 mixed factorial

design. Participants were randomly assigned to one of two groups: feedback (subjects obtained feedback in terms of color highlights) and no feedback (subjects performed the same tasks as in the feedback condition, but without obtaining feedback in terms of color highlights). We compared the initial performance before obtaining feedback to the performance after obtaining feedback.

7.2 REALIZATION

To realize the portable version of the environment presented in Chapter 2, we use a similar architecture. We replace the marker-based motion capture system with a depth camera (Microsoft Kinect 2) and we replace the CAVE with a head-mounted display (HTC Vive). One single machine (Intel Xeon CPU E5-1620 @3.6 GHz, 8 GB Ram, Nividia GeForce GTX 1080) is used for all parts of our pipeline. The origin of the virtual environment is placed below the Kinect camera on the floor level during calibration of the HMD. To determine the height of the Kinect camera, we place a Vive controller on top of the camera and use the Lighthouse tracking system to determine its location. Due to the slow Kinect camera which is the bottleneck of this setup, according to the measurements in Chapter 2, the latency of this setup can be expected as being around 100 ms. When transfering the motion capture data as provided by the Kinect camera to a virtual character, the drawbacks of the relatively inaccurate motion capture can easily observed. For instance joints tend to flicker as their angle cannot be robustly estimated (see Figure 7.1a). Sometimes joints such as the hands completely flip, e.g., due to occlusions. Consequently, we chose not to animate a preprocessed virtual avatar. Instead, inspired by Beck et al. [Bec+13], we stream the point cloud of the subject as obtained from the Kinect as avatar of the subject (see Figure 7.1b). Similar to the environment described in Chapter 2, participants were placed in a virtual gym in front of a virtual mirror.

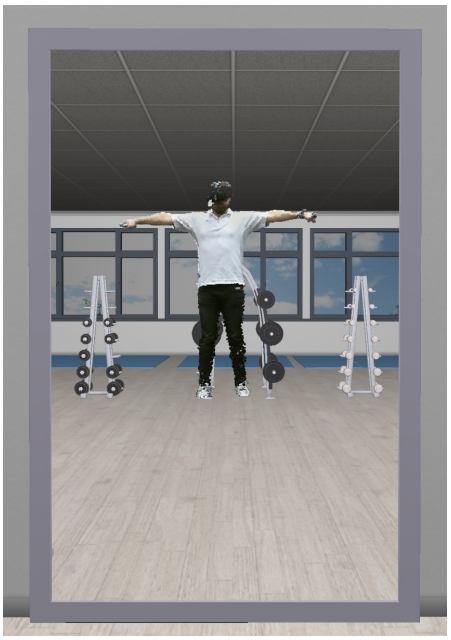
Still the visual quality of the obtained scene is inferior to the one obtained in the CAVE, especially due to the flickering of the extracted point cloud. Also the latency is clearly inferior. However, the time for conducting experiments is greatly reduced as marker placement is not necessary anymore. Further, subjects can wear their everyday clothes and do not need to put on the motion capture suit.

7.3 PILOT EXPERIMENT

7.3.1 *Task and Feedback*

In order to be compatible with the portable version of our environment, especially with the Kinect tracking, we restrict the motor action to be slow, executed frontally oriented and containing few possibilities for occlusions. We chose to focus on the lateral raise as it fully satisfies these constraints. In order to prevent overstressing, we removed as much weight as possible from the barbells. This results in a weight of 2.2 kg participants had to lift with each arm. We decide in favor of an error pattern that can easily be observed from the front: “incorrect height” of the arms. This pattern

(a) 3D scan animated based on the data obtained from the Kinect camera.



(b) Point cloud streamed from the data obtained from the Kinect.

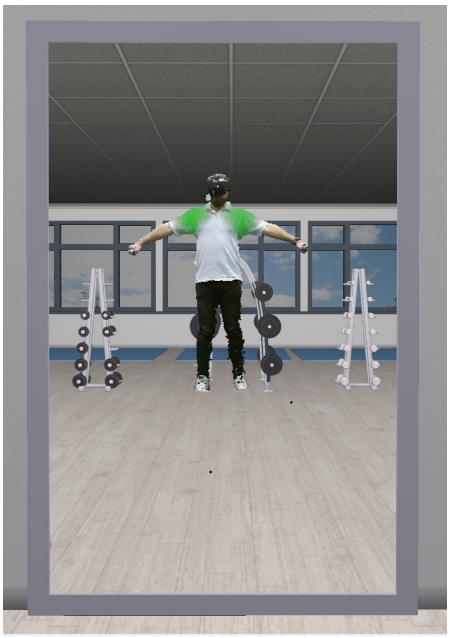
Figure 7.1: Comparison of using a skinned virtual character driven by the joint angles obtained from the Kinect vs. directly using a 3D point cloud as obtained from the Kinect camera.

is a combination of rule-based classifiers (cf. Section 5.3) for the sub patterns arms "not high enough" and "too high". To provide feedback, the shoulders of the subject's avatar are highlighted in green if "not high enough" is detected and in red in case of the pattern "too high". As we do not use a 3D animated mesh as avatar, but only the streamed point cloud, we have no information on which points in the point cloud belong to which joint. To be able to highlight plausible points in the point cloud that belong to the shoulder, we first estimate the subject's joint lengths based on the information provided by the Kinect and calculate the translation of the subject's joints. Next we define a radius around the joints that are responsible for the error patterns of interest (the left and the right shoulder) and highlight all points in the point cloud around the given radius as soon as an error pattern is violated (see Figure 7.2). The calculation of the points within the given radius is performed in the shader.

7.3.2 *Procedure and Measures*

Fourteen participants (9 males; age $M = 26.43$, $SD = 4.57$) took part in the study. One further participant was recorded, but had to be excluded due to technical reasons. Participants provided written informed consent. The study was conducted in accordance with the Declaration of Helsinki, and had ethical approval from the ethics committee of Bielefeld University.

The experiment consists of three phases: Pre-test, acquisition and post-test. All phases took place on one day subsequently to each other. Participants were divided



(a) The arms are not high enough. Consequently the shoulders are colored in green.

(b) The arms are too high. Consequently the shoulders are colored in red.

Figure 7.2: Color highlight which we provide inside the portable version of our training environment.

into two groups (group feedback: $n = 8$; group baseline: $n = 6$), which differed in the quality of the feedback provided in the acquisition phase.

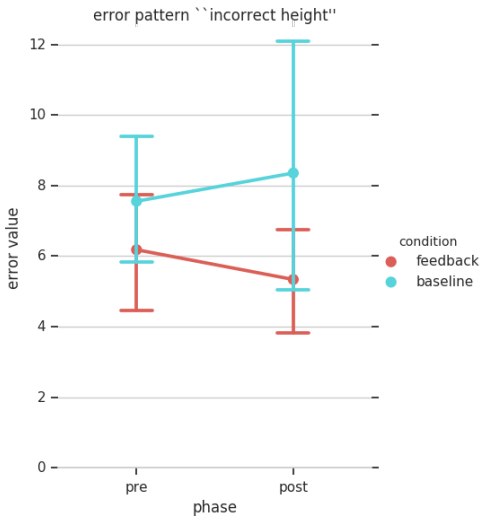


Figure 7.3: The graph shows the effect of the visual feedback provided to the different groups on the overall error.

First, we welcomed the participants and handed out the main instructions for the overall experiment as well as a consent form. In the next step, participants filled in questionnaires. Then, they put on the HMD and the barbells were put into the subjects hand. In the virtual environment, a virtual character demonstrated the desired movement. Then, the pre-test began. Subjects were asked to perform 10 lifts in sets of 5. A countdown presented as text indicated when to start the movement. During the pre-test, the mirror was disabled so participants were unable to observe their own performance. After the pre-test, the acquisition phase began and participants were asked to perform 25 lifts in 5 sets. Again a countdown was used to indicate when to perform the movement. Participants in the group "baseline" only observed their avatar in the mirror during practice. Subjects in the group "feedback" obtained guided augmented feedback in terms of color highlights to reach the desired height of the

arms. See Figure 7.2 for an overview of the provided feedback. After the acquisition phase, the post-test took place. Participants performed 10 lifts in sets of 5 using the same procedure as in the pre-test. Finally, the subjects filled in the post questionnaire. The whole procedure took around 30 minutes.

To measure the subjects' performance in terms of the error pattern "incorrect height", we used the classifiers proposed in Chapter 5.3 for error detection. We consider a performance as ideal when having the arms parallel to the floor. The error pattern "incorrect height" combines the error patterns "not high enough" and "too high". The classifiers measure the deviation of the rotation of the arms from the desired target posture. As subjectives measures, we focus on

- feeling that the avatar represented the subject's movement ("How strong was your feeling that the virtual avatar mirrored your own motion?"; related to agency, cf. [Gal00]).
- improvement ("How would you rate your improvement?").

These questions were answered on 5-point scales where -2 is the lowest possible value and $+2$ the maximum possible value.

7.3.3 *Results and Discussion*

For each parameter of the motor performance, a two-way mixed analysis of variance (ANOVA) was conducted with phase (pre-test, post-test) as within-subject factor and group (feedback, baseline) as between-subject factor. For all analyses, the level of significance was set at $p < 0.05$. For the error pattern "incorrect height", as the combination of the error patterns "not high enough" and "too high", we did not observe any significant effects. There was neither a main effect on condition ($F_{1,12} = 2.8, p = 0.12, \eta_p^2 = 0.189$) nor on phase ($F_{1,12} = 0.02, p = 0.9, \eta_p^2 = 0.001$). Further we did not observe any interaction between condition and phase ($F_{1,12} = 0.55, p = 0.47, \eta_p^2 = 0.044$). The results for the overall error is summarized in Figure 7.3. When looking closer on the data, evaluations for the single error patterns "too high" and "not high enough" (cf. Figure 7.4) indicate a possible reason for this missing effect: Subjects might tend to overshoot. For the pattern "too high" results showed that there was a trend in the main effect of condition ($F_{1,12} = 4.33, p = 0.06, \eta_p^2 = 0.265$) on performance with subjects in the feedback condition ($M = 2.86, SD = 3.57$) performing slightly better overall than subjects in the baseline group ($M = 6.21, SD = 4.81$). In addition, there was also a trend in the main effect of phase on performance error ($F_{1,12} = 3.99, p = 0.07, \eta_p^2 = 0.244$), with participants performing better in the post-test ($M = 3.20, SD = 5.30$) than in the pre-test ($M = 5.40, SD = 3.10$). There was a significant interaction between condition and phase ($F_{1,12} = 10.73, p = 0.007, \eta_p^2 = 0.472$). Descriptive statistics showed that subjects who obtained feedback obtained better scores in the post-test ($M = 0.18, SD = 0.22$) as compared to the pre-test ($M = 5.54, SD = 3.28$). Subjects who did not obtain feedback showed a reversed pattern ($M_{pre} = 5.20, SD = 3.13; M_{post} = 7.23, SD = 6.23$). For the pattern "not high enough" results showed no significant main effect in condition ($F_{1,12} = 1.16, p = 0.30, \eta_p^2 = 0.088$). However, there was a significant main effect in phase ($F_{1,12} = 10.39, p =$

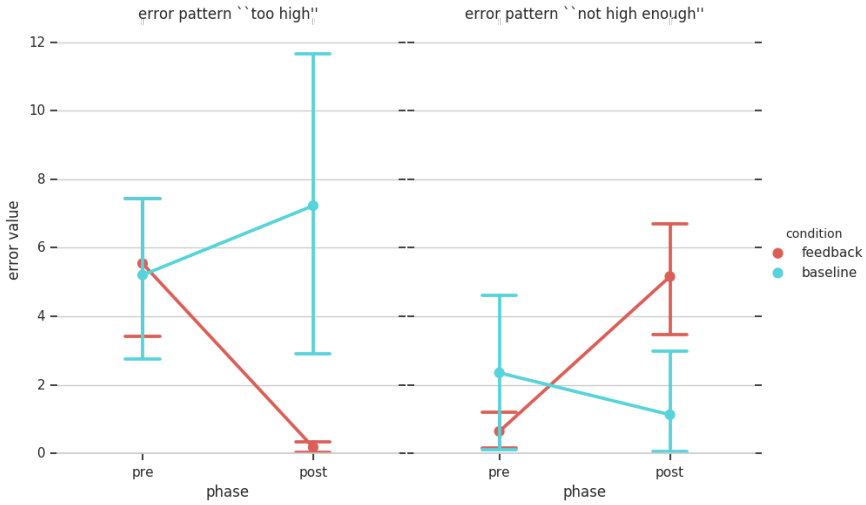


Figure 7.4: Each plot shows the effect of the visual feedback provided to the different groups on the errors "too high" and "not high enough". Error values are provided in degrees.

$0.007, \eta_p^2 = 0.464$), indicating that subjects in the post-test performed worse ($M = 3.43, SD = 3.08$) than in the pre-test ($M = 1.37, SD = 2.36$). There was a significant interaction between condition and phase ($F_{1,12} = 19.79, p = 0.0008, \eta_p^2 = 0.623$). Here, descriptive statistics revealed that subjects who obtained feedback obtained worse scores in the post-test ($M = 5.16, SD = 2.48$) than in the pre-test ($M = 0.64, SD = 0.81$), whereas subjects who were in the baseline condition obtained better scores in the post-test ($M = 1.13, SD = 2.20$) as compared to the pre-test ($M = 2.35, SD = 3.40$). See Figure 7.4 for an overview for the error patterns "too high" and "not high enough".

To summarize, when looking at the error "too high" subjects appear to profit from obtaining our feedback. However, they tend to become worse for the opposite pattern "not high enough". A possible explanation for these effects might be an overshooting behavior when trying to improve the performance for the pattern "not high enough". This might be especially true, as participants first (during the moving-up phase) obtained the feedback for pattern "not high enough". We assume that this effect is not induced due to general fatigue, as subjects in the baseline group did not become worse with respect to this error. One reason of this effect might be the high latency. This is in line with recent research stating that latencies between 75 ms and 125 ms can already negatively impact motor performance [Wal+16].

To analyze the perceived improvement of the participants, we calculated the mean response in every group. Differences across the two groups were tested by means of the Mann-Whitney-U-Test. We did not observe any significant results ($p = 0.34$). Concerning the participants feeling, that their avatar represented their movement, we obtained good results ($M = 1.6$, $SD = 0.83$).

7.4 CONCLUSION

We demonstrated that it is possible to set up a portable version of our motor learning environment. According to our results, this portable environment can lead to a successful reduction of motor errors. In our experiment, we applied concurrent color feedback to guide subjects to improve their performances with respect to two error patterns, “not high enough” and “too high”. Subjects who obtained color feedback and who raised their arms too high, were able to improve with respect to this error. However, results indicate overshooting for the opposite error pattern “not high enough”. Further, we only tested the setup in a scenario that is expected to be least vulnerable to the drawbacks of the setup, namely high latency, unstable motion capture, as well as sweating. As a conclusion, our motor learning environment can be scaled down to a consumer version. However, for such an environment that is based on current consumer devices, the impact of possible drawbacks due to parts of the system that do not fulfill the general requirements (cf. Chapter 2) must be carefully considered, depending on the field of application. For non-frontally-oriented, complex, or faster tasks that involve the whole body we suggest to use our default high performance state-of-the-art environment as described in Chapter 2.

8 DISCUSSION AND CONCLUSION

In this thesis, we developed core concepts for VR environments for motor learning based on motion and kinematic data. To this end, we first carved-out a set of requirements. First, we suggest that trainees should receive feedback on their own motion. Further, the system should have a low latency and a high frame rate. The level of disturbance a trainee is exposed to should be minimized. Also, motion capture systems should provide a robust tracking. After establishing these requirements, we proposed an exemplary environment that is able to meet the demands. To this end, we first evaluated suitable hardware and software. We finally proposed a solution towards a motor learning environment that consists of a combination of a marker-based outside-in motion capture system, a suitable preprocessing and mapping of the motion capture data to a trainee's virtual avatar, as well as a L-Shaped CAVE environment that is driven by a render engine tailored to our needs. In an experiment, we demonstrated that this environment can be used to help trainees in improving their motor performances. More precisely, the experiment was designed to evaluate whether superimposing a skilled movement on top of the trainee's movement can lead trainees to adapt their movement in order to match the skilled one. We further evaluated whether the perspective from which the trainees observed their overlay had an influence on the parameters the trainees improved. Our findings indicate that it can be beneficial for novices to watch oneself together with a skilled performance during practice and that improvement depends on the perspective chosen. Consequently, in order to develop an ideal motor learning environment, we suggest that such an environment should be able to make use of learning via observation as a feedback strategy.

In the next step, we move towards the question on how to exploit the online motion capture data to detect typical errors in a trainee's performance. We require this error detection to work online, already during a trainee's performance. However, before focusing on the assessment of the trainee's performance, we identify the need to obtain information on the current timing of a performed exercise with respect to a reference timing. To this end, we extend the well known algorithm open-end DTW to achieve a more accurate alignment of an incomplete motion streamed by the motion capture system to a reference movement. The extension consists of path-length weighting together with evolutionary optimized joint weights. In order to include open-end DTW in a motor learning environment, we suggest, for each motor action of interest, to evaluate initial performances based on standard open-end DTW and to extend it whenever necessary based on our suggested extensions. Next, we proposed a pipeline towards the detection of typical error patterns and the generation of feedback based on the underlying classifiers. To this end, we warp a trainee's movement into the timing of a reference movement by using our extended open-end DTW. Then, we extract a feature vector from the warped movement. This feature vector is reduced by a feature selection mask that is trained by a Random Forest. Finally, the resulting reduced feature vector is classified by a linear SVM. We demonstrate that our pipeline

improves over the state of the art in terms of quality of the classification as well as the ability to automatically generate feedback based on the learned classifier. In addition to this data-driven pipeline, we suggest an optional rule-based pipeline. Rules are designed from expert knowledge as well as based on literature. This approach is especially suitable in cases where either no or only few training data is available or in cases where an error pattern can easily be described using rules. As a suggestion for an ideal motor learning environment, we suggest to use the following strategy: If no training data is available and the error patterns of interest are sufficiently easy to formalize, rule-based detection as proposed in Chapter 5.3 could be used. For more complex setups and when having labeled training data, we propose to include three steps: alignment of performed motion in order to extract a feature vector, reduction of the feature vector via a supervised feature selection, and finally the usage of a simple classifier that allows the extraction of information in order to directly generate feedback (see Chapter 5.4).

In a final study (see Chapter 6), we combined our findings and concepts. To this end, we implemented a coaching cycle with a similar structure as applied in real-world coaching sessions inside our environment. We used two data-driven and one rule-based classifier and linked them to various possible options to provide feedback. In our study, we demonstrated that our system is able to help people in improving their motor performances with respect to error patterns that the system can address. Based on our evaluations in that chapter, we suggest to use the concepts developed in this thesis as a tool box in order to develop an ideal motor learning environment.

Finally, we indicated how to scale down parts of the environment towards a consumer system (see Chapter 7). The resulting prototype uses a depth camera for motion capture and a HMD for the visualization. In a pilot experiment, we demonstrated that this system can be able to help trainees in improving their performance, however it has some drawbacks in terms of robustness of motion capture as well as latency. We suggest that, depending on the field of application, it must be carefully considered whether to take the risk of an environment that does not fully satisfy the suggestions for motor learning environment or whether to use a state-of-the-art, but also expensive and more complex environment.

To summarize, in this thesis, we developed core concepts for motor learning environments with a special focus on the handling of kinematic data. We developed these concepts starting from general requirements and suggestions, over an exemplary implementation of a state-of-the-art environment until the assessment of the performed motion and the generation of suitable feedback to the trainee. Our findings and suggestions can be used by researchers and developers to set-up effective motor learning environments to help people in improving their motor performances, but also to investigate the effectiveness of various feedback strategies on motor learning.

LIMITATIONS AND FUTURE WORK

Despite the contributions, this thesis of course has some limitations and also leaves some space for future work in the field. We will go through these points part by part. In Chapter 2, we suggested to state information on possible latency of VR systems in

all publications to make findings easier to reproduce. However, the way we measure latency is time consuming. The setup must be filmed and these films are, later on, analyzed by hand. This leaves space for improvement. An automatic annotation of these measurements would reduce the overall effort. Consequently, with such an approach at hand, researchers might be more willing to measure and to report their end-to-end latency. Concerning the suggestions and requirements we propose for motor learning environments, for some of them, it would be interesting to obtain information on the level to which they are needed. For instance it is not clear how robust exactly a motion capture system needs to be, nor how much hardware can be attached to a trainee until she is severely impaired in her performance. Concerning the impact of different levels of latency on perception and motor learning, experiments inside our environment have already been conducted in order to gain more knowledge on a necessary minimal end-to-end latency [Wal+16]. According to the results presented in [Wal+16], an end-to-end latency of more than 75 ms has been found to increase awareness for delays as well to decrease motor performance. A latency of more than 125 ms affects perceptual aspects such as sense of agency and ownership. Further details can be found in [Wal+16]. Experiments performed by Stauffert et al. demonstrated the negative impact of latency jitter in head tracking on user experience in terms of simulator sickness [SNL18]. A further direction of future work could be thus to evaluate the impact of latency jitter of full-body motion capture data inside motor learning environments. Concerning the environment itself, focusing on accurate marker-less and low-latency motion capture algorithms is promising to advance the field. Further, not only using tracking of the larger parts of the body, but also integrating hand tracking, face tracking, eye tracking, et cetera would increase the quality of our system. Our pipeline towards an online classification of motor errors and the generation of feedback could be further improved by focusing on the single components. For instance, the developed weight-optimized open-end DTW could be combined with other optimizations and extensions of DTW, such as Derivative DTW [KP01], Sakoe-Chuba Band [SC78], and Fast DTW [SC07]. The simple state-based detection of the beginning of a motor action could be extended for instance by sliding-window-based classification [Cao+04] or by new approaches such as [Núñ+18]. Here, also approaches that require a substantial amount of training data could be applied, as the acquisition of data that is only annotated with respect to the performed motor action, not necessarily with respect to specific error patterns, is comparably easy. Further, the features used in our hierarchical representation could be enriched by further higher-level features to detect more complex error patterns. For instance, temporal properties as obtained from the DTW could be included.

Concerning further experiments, our work paves the way to address multiple problems that are highly interesting. For instance, the influence of skill level on the effectiveness of specific ways to provide feedback is an important subject of possible future investigations. This also includes a coaching that gradually approaches the expert performance. Novices could be trained bit-by-bit towards more and more elaborated levels of expertise. New approaches towards the generation of feedback could be evaluated, for instance the exaggeration of errors based on just classified movements. Here, similar to the generated visual feedback, information from the learned classifiers could be used as a basis to manipulate the movement that is

Figure 8.1: In a prototypical extension of our system two people interact with their avatars inside our virtual environment. For the person on the left, the face, the body, as well as the hands are tracked. For the person on the right, we track the body and the hands.

performed by a trainee. However, not only the type of feedback is an important subject of further investigations, but also the amount of augmented feedback that is necessary to obtain the desired amount of motor learning. Further, when developing a virtual agent that act as a coach, a comparison to a real world coach would be desirable. One way to compare the performance obtained by an autonomous coach with the one of a real-world coach would be Wizard-of-Oz experiments. In such a setting, a trainee is placed in our environment and either interacts with the avatar of a real coach who is placed in another laboratory or with an autonomous virtual agent. Such a setting would allow us to gradually adapt specific properties of the system and to measure their impact on the trainee. This will be an important direction of future work that has already been approached with first prototypes (see Figure 8.1). Further, it would be interesting to evaluate how well our findings generalize on different types of motor actions and whether specific effects are moderated by the type of exercise a trainee learns. Finally, an evaluation of our environment in the field of rehabilitation would be a promising direction of future research. For instance, our pipeline could be extended in a way to deal with trainees that have specific limitations such as being unable to move certain parts of the body.

BIBLIOGRAPHY

[Ach+17] J. Achenbach, T. Waltemate, M. E. Latoschik, and M. Botsch. "Fast generation of realistic virtual humans". In: *ACM Symposium on Virtual Reality Software and Technology*. 2017, p. 12.

[AF13] X. Anguera and M. Ferrarons. "Memory efficient subsequence DTW for query-by-example spoken term detection". In: *Multimedia and Expo (ICME), 2013 IEEE Int. Conf. On*. 2013, pp. 1–6.

[And+13] F. Anderson, T. Grossman, J. Matejka, and G. Fitzmaurice. "YouMove: Enhancing Movement Training with an Augmented Reality Mirror". In: *Proc. of the 26th annual ACM symposium on user interface software and technology*. 2013, pp. 311–320.

[AP13] M. Andrieux and L. Proteau. "Observation learning of a motor task: who and when?" In: *Experimental brain research* 229.1 (2013), pp. 125–137.

[AP14] M. Andrieux and L. Proteau. "Mixed observation favors motor learning through better estimation of the model's performance". In: *Experimental brain research* 232.10 (2014), pp. 3121–3132.

[APV18] S. Arndt, A. Perkis, and J.-N. Voigt-Antons. "Using Virtual Reality and Head-Mounted Displays to Increase Performance in Rowing Workouts". In: *Proc. of the 1st International Workshop on Multimedia Content Analysis in Sports*. ACM. 2018, pp. 45–50.

[ARB08] K. Adistambha, C. H. Ritz, and I. S. Burnett. "Motion classification using dynamic time warping". In: *Multimedia Signal Processing, IEEE Workshop on*. 2008, pp. 622–627.

[Ari+14] A. Arici, S. Celebi, A. S. Aydin, and T. T. Temiz. "Robust gesture recognition using feature pre-processing and weighted dynamic time warping". In: *Multimedia Tools and Applications* 72.3 (2014), pp. 3045–3062.

[ARS14] D. I. Anderson, A. M. Rymal, and D. M. Ste-Marie. "Modeling and Feedback". In: ed. by A. G. Papaioannou and D. Hackfort. Routledge Companion to Sport and Exercise Psychology: Global perspectives and fundamental concepts. Routledge, 2014, pp. 272–288.

[Bec+13] S. Beck, A. Kunert, A. Kulik, and B. Froehlich. "Immersive group-to-group telepresence". In: *IEEE Transactions on Visualization and Computer Graphics* 19.4 (2013), pp. 616–625.

[Bev+18] A. Bevilacqua, B. Huang, R. Argent, B. Caulfield, and T. Kechadi. "Automatic Classification of Knee Rehabilitation Exercises Using a Single Inertial Sensor: a Case Study". In: *15th Int. Conf. on Wearable and Implantable Body Sensor Networks (BSN)*. IEEE. 2018, pp. 21–24.

[Bi+03] J. Bi, K. Bennett, M. Embrechts, C. Breneman, and M. Song. "Dimensionality reduction via sparse support vector machines". In: *Journal of Machine Learning Research* 3.Mar (2003), pp. 1229–1243.

[Bia12] G. Biau. "Analysis of a random forests model". In: *Journal of Machine Learning Research* 13.Apr (2012), pp. 1063–1095.

[Bie00] A. D. Bierbaum. "VR Juggler: A Virtual Platform for Virtual Reality Application Development". PhD thesis. Iowa State University, 2000, pp. 1–115.

[Bis06] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer New York, Inc., 2006.

[BL14] A. Bagnall and J. Lines. "An experimental evaluation of nearest neighbour time series classification". In: *arXiv preprint arXiv:1406.4757* (2014).

[BOL17] H. Brock, Y. Ohgi, and J. Lee. "Learning to judge like a human: convolutional networks for classification of ski jumping errors". In: *Proc. of the 2017 ACM International Symposium on Wearable Computers*. 2017, pp. 106–113.

[Bre+84] L. Breiman, J. Friedman, C. J. Stone, and R. A. Olshen. *Classification and regression trees*. CRC press, 1984.

[Bre01] L. Breiman. "Random forests". In: *Machine learning* 45.1 (2001), pp. 5–32.

[BSR11] R. Bailey, J. Selfe, and J. Richards. "The single leg squat test in the assessment of musculoskeletal function: a review". In: *Physiotherapy Practice and Research* 32.2 (2011), pp. 18–23.

[Bur+11] A.-M. Burns, R. Kulpa, A. Durny, B. Spanlang, M. Slater, and F. Multon. "Using virtual humans and computer animations to learn complex motor skills: a case study in karate". In: *SKILLS*. EDP Sciences. 2011, pp. 1–4.

[Can+18] A. Cannavò, F. G. Pratticò, G. Ministeri, and F. Lamberti. "A Movement Analysis System based on Immersive Virtual Reality and Wearable Technology for Sport Training". In: *Proc. of the 4th Int. Conf. on Virtual Reality*. ACM. 2018, pp. 26–31.

[Cao+04] D. Cao, O. T. Masoud, D. Boley, and N. Papanikolopoulos. "Online motion classification using support vector machines". In: *Int. Conf. On Robotics and Automation*. Vol. 3. IEEE. 2004, pp. 2291–2296.

[Cao+17] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh. "Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 7291–7299.

[Car+15] B. Caramiaux, N. Montecchio, A. Tanaka, and F. Bevilacqua. "Adaptive gesture recognition with variation estimation for interactive systems". In: *ACM Transactions on Interactive Intelligent Systems (TiiS)* 4.4 (2015), pp. 1–34.

[CCK15] H.-R. Choi, H. Y. Cho, and T. Y. Kim. "Dynamically weighted DTW for dynamic full-body gesture recognition". In: *Proc. of The Int. Conf. on Intelligent Systems and Image Processing*. 2015, pp. 126–129.

[Cel+13] S. Celebi, A. S. Aydin, T. T. Temiz, and T. Arici. "Gesture recognition using skeleton data with weighted dynamic time warping". In: *Proc. of VISAPP*. 2013, pp. 620–625.

[Cha+11] J. C. Chan, H. Leung, J. K. Tang, and T. Komura. "A virtual reality dance training system using motion capture technology". In: *IEEE Transactions on Learning Technologies* 4.2 (2011), pp. 187–195.

[Chu+03] P. T. Chua, R. Crivella, B. Daly, N. Hu, R. Schaaf, D. Ventura, T. Camill, J. Hodgins, and R. Pausch. "Training for physical tasks in virtual environments: Tai Chi". In: *Proc. of IEEE Virtual Reality*. 2003, pp. 87–94.

[CL06] Y.-W. Chen and C.-J. Lin. "Combining SVMs with Various Feature Selection Strategies". In: *Feature Extraction: Foundations and Applications*. Ed. by I. Guyon, M. Nikravesh, S. Gunn, and L. A. Zadeh. Vol. 207. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006. Chap. 12, pp. 315–324.

[CLH03] S. Chan, T. Luk, and Y. Hong. "Kinematic and electromyographic analysis of the push movement in Tai Chi". In: *British Journal of Sports Medicine* 37.4 (2003), pp. 339–344.

[CLS08] M. A. Clark, S. Lucett, and B. G. Sutton. *NASM essentials of personal fitness training*. Lippincott Williams & Wilkins, 2008.

[COM14] A. Covaci, A.-H. Olivier, and F. Multon. "Third person view and guidance for more natural motor behaviour in immersive basketball playing". In: *ACM Symposium on Virtual Reality Software and Technology*. 2014, pp. 55–64.

[Cus+18] E. E. Cust, A. J. Sweeting, K. Ball, and S. Robertson. "Machine and deep learning for sport-specific movement recognition: a systematic review of model development and performance". In: *Journal of sports sciences* (2018), pp. 1–33.

[Den+11] L. Deng, H. Leung, N. Gu, and Y. Yang. "Real-time mocap dance recognition for an interactive dancing game". In: *Computer Animation and Virtual Worlds* 22.2-3 (2011), pp. 229–237.

[DHS18] P. Düking, H.-C. Holmberg, and B. Sperlich. "The potential usefulness of virtual reality systems for athletes: A short SWOT analysis". In: *Frontiers in physiology* 9 (2018), pp. 1–4.

[Dix05] S. Dixon. "Live tracking of musical performances using on-line time warping". In: *Proc. of the 8th Int. Conf. on Digital Audio Effects*. 2005, pp. 92–97.

[Esc01] R. F. Escamilla. "Knee biomechanics of the dynamic squat exercise". In: *Medicine and science in sports and exercise* 33.1 (2001), pp. 127–141.

[Fer+14] M. Fernández-Delgado, E. Cernadas, S. Barro, and D. Amorim. "Do we need hundreds of classifiers to solve real world classification problems". In: *Journal of Machine Learning Research* 15.1 (2014), pp. 3133–3181.

[FKS18] C. Frank, T. Kim, and T. Schack. "Observational Practice Promotes Action-Related Order-Formation in Long-Term Memory: Investigating Action Observation and the Development of Cognitive Representation in Complex Motor Action". In: *Journal of Motor Learning and Development* 6.1 (2018), pp. 53–72.

[FLS13] C. Frank, W. M. Land, and T. Schack. "Mental representation and learning: the influence of practice on the development of mental representation structure in complex action". In: *Psychology of Sport and Exercise* 14.3 (2013), pp. 353–361.

[Fra+01] N. Franck, C. Farrer, N. Georgieff, M. Marie-Cardine, J. Daléry, T. d'Amato, and M. Jeannerod. "Defective recognition of one's own actions in patients with schizophrenia". In: *American Journal of Psychiatry* 158.3 (2001), pp. 454–459.

[Fra+14] C. Frank, W. M. Land, C. Popp, and T. Schack. "Mental representation and mental practice: experimental investigation on the functional links between motor memory and motor imagery". In: *PLOS One* 9.4 (2014), e95175.

[FS14] S. Friston and A. Steed. "Measuring latency in virtual environments". In: *IEEE Transactions on Visualization and Computer Graphics* 20.4 (2014), pp. 616–625.

[FW13] J. Fröhlich and I. Wachsmuth. "The visual, the auditory and the haptic–a user study on combining modalities in virtual worlds". In: *Int. Conf. on Virtual, Augmented and Mixed Reality*. Springer. 2013, pp. 159–168.

[Gal00] S. Gallagher. "Philosophical conceptions of the self: implications for cognitive science". In: *Trends in cognitive sciences* 4.1 (2000), pp. 14–21.

[GE03] I. Guyon and A. Elisseeff. "An introduction to variable and feature selection". In: *Journal of machine learning research* 3.Mar (2003), pp. 1157–1182.

[GKC13] O. Giggins, D. Kelly, and B. Caulfield. "Evaluating rehabilitation exercise performance using a single inertial measurement unit". In: *Proc. of the Int. Conf. on Pervasive Computing Technologies for Healthcare*. 2013, pp. 49–56.

[GL04] M. A. Guadagnoli and T. D. Lee. "Challenge point: a framework for conceptualizing the effects of various practice conditions in motor learning". In: *Journal of motor behavior* 36.2 (2004), pp. 212–224.

[GMS17] B. Gregorutti, B. Michel, and P. Saint-Pierre. "Correlation and variable importance in random forests". In: *Statistics and Computing* 27.3 (2017), pp. 659–678.

[Goo+16] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio. *Deep learning*. Vol. 1. MIT press Cambridge, 2016.

[GP00] M. A. Giese and T. Poggio. "Morphable models for the analysis and synthesis of complex motion patterns". In: *International Journal of Computer Vision* 38.1 (2000), pp. 59–73.

[GPT10] R. Genuer, J.-M. Poggi, and C. Tuleau-Malot. "Variable selection using random forests". In: *Pattern Recognition Letters* 31.14 (2010), pp. 2225–2236.

[GSC14] O. M. Giggins, K. T. Sweeney, and B. Caulfield. "Rehabilitation exercise assessment using inertial sensors: a cross-sectional analytical study". In: *Journal of Neuroengineering and Rehabilitation* 11.1 (2014), pp. 1–10.

[Gut02] C. Gutwin. "The Effects of Network Delays on Group Work in Real-Time Groupware". In: *Proc. of European Conference on Computer Supported Cooperative Work*. 2002, pp. 299–318.

[Häm04] P. Hämäläinen. "Interactive video mirrors for sports training". In: *Proc. of the third Nordic conference on Human-computer interaction*. ACM. 2004, pp. 199–202.

[Han+16] P.-H. Han, K.-W. Chen, C.-H. Hsieh, Y.-J. Huang, and Y.-P. Hung. "Ar-arm: Augmented visualization for guiding arm movement in the first-person perspective". In: *Proc. of the 7th Augmented Human Int. Conf. 2016*. ACM. 2016, pp. 1–4.

[Han+18] Y. Han, S.-L. Chung, A. Ambikapathi, J.-S. Chan, W.-Y. Lin, and S.-F. Su. "Robust Human Action Recognition Using Global Spatial-Temporal Attention for Human Skeleton Data". In: *2018 International Joint Conference on Neural Networks (IJCNN)*. IEEE. 2018, pp. 1–8.

[Han16] N. Hansen. "The CMA evolution strategy: A tutorial". In: *arXiv preprint arXiv:1604.00772* (2016).

[Hel+06] A. Heloir, N. Courty, S. Gibet, and F. Multon. "Temporal alignment of communicative gesture sequences". In: *Computer animation and virtual worlds* 17.3-4 (2006), pp. 347–357.

[HF02] N. J. Hodges and I. M. Franks. "Modelling coaching practice: the role of instruction and demonstration". In: *Journal of sports sciences* 20.10 (2002), pp. 793–811.

[HF04] N. J. Hodges and I. M. Franks. "The Nature of Feedback". In: ed. by I. M. Franks and M. Hughes. Notational analysis of sport: Systems for better coaching and performance in sport. Routledge, 2004, pp. 17–39.

[HHW09] J. Heron, J. V. Hanson, and D. Whitaker. "Effect before cause: supramodal recalibration of sensorimotor timing". In: *PLOS One* 4.11 (2009).

[HKK16] R. Houmanfar, M. Karg, and D. Kulić. "Movement Analysis of Rehabilitation Exercises: Distance Metrics for Measuring Patient Progress". In: *IEEE Systems Journal* 10.3 (2016), pp. 1014–1025. ISSN: 1932-8184.

[HO14] T. Hachaj and M. R. Ogiela. "Rule-based approach to recognizing human body poses and gestures in real time". In: *Multimedia Systems* 20.1 (2014), pp. 81–99.

[HO97] N. Hansen and A. Ostermeier. "Convergence properties of evolution strategies with the derandomized covariance matrix adaptation: The $(\mu/\mu_I, \lambda)$-CMA-ES". In: vol. 97. 1997, pp. 650–654.

[Hoa+16] T. N. Hoang, M. Reinoso, F. Vetere, and E. Tanin. "Onebody: Remote posture guidance system using first person view in virtual environment". In: *Proc. of the 9th Nordic Conference on Human-Computer Interaction*. ACM. 2016, p. 25.

[Hol05] M. K. Holden. "Virtual environments for motor rehabilitation". In: *Cyberpsychology & behavior* 8.3 (2005), pp. 187–211.

[Hou+15] J. Hough, I. de Kok, D. Schlangen, and S. Kopp. "Timing and grounding in motor skill coaching interaction: Consequences for the information state". In: *Proc. of the 19th SemDial Workshop on the Semantics and Pragmatics of Dialogue (goDIAL)*. 2015, pp. 86–94.

[HSG15] E.-J. Hossner, F. Schiebl, and U. Göhner. "A functional approach to movement analysis and error identification in sports and physical education". In: *Frontiers in Psychology* 6 (2015), pp. 1–12.

[Hül+16] F. Hülsmann, C. Frank, T. Schack, S. Kopp, and M. Botsch. "Multi-level analysis of motor actions as a basis for effective coaching in virtual reality". In: *Proc. of the 10th International Symposium on Computer Science in Sports (ISCSS)*. Springer. 2016, pp. 211–214.

[Hül+17] F. Hülsmann, A. Richter, S. Kopp, and M. Botsch. "Accurate online alignment of human motor performances". In: *Proc. of ACM Motion in Games*. Barcelona: ACM, 2017, pp. 7:1–7:6.

[Hül+18] F. Hülsmann, J. P. Göpfert, B. Hammer, S. Kopp, and M. Botsch. "Classification of motor errors to provide real-time feedback for sports coaching in virtual reality—A case study in squats and Tai Chi pushes". In: *Computers & Graphics* 76 (2018), pp. 47–59.

[Hül+19] F. Hülsmann, C. Frank, I. Senna, M. O. Ernst, T. Schack, and M. Botsch. "Superimposed skilled performance in a virtual mirror improves motor performance and cognitive representation of a full-body motor action". In: *Frontiers in Robotics and AI* 6 (2019), pp. 43:1–43:17.

[Huy09] D. Q. Huynh. "Metrics for 3D rotations: Comparison and analysis". In: *Journal of Mathematical Imaging and Vision* 35.2 (2009), pp. 155–164.

[IA15] S. Imaizumi and T. Asai. "Dissociation of agency and body ownership following visuomotor temporal recalibration". In: *Frontiers in integrative neuroscience* 9 (2015), pp. 1–10.

[ISO05] ISO19774. *Information technology — Computer graphics and image processing — Humanoid animation (H-Anim)*. Standard. Geneva, CH: International Organization for Standardization, 2005.

[Jac+14] A. Jacobson, Z. Deng, L. Kavan, and J. Lewis. "Skinning: Real-time Shape Deformation". In: *ACM SIGGRAPH Course Notes*. 2014.

[JJO11] Y.-S. Jeong, M. K. Jeong, and O. A. Omitaomu. "Weighted dynamic time warping for time series classification". In: *Pattern Recognition* 44.9 (2011), pp. 2231–2240.

[JNS12] S. Jörg, A. Normoyle, and A. Safonova. "How Responsiveness Affects Players' Perception in Digital Games". In: *Proc. of ACM Symposium on Applied Perception*. 2012, pp. 33–38.

[Jot+13] R. Jota, A. Ng, P. Dietz, and D. Wigdor. "How fast is fast enough? A study of the effects of latency in direct-touch pointing tasks". In: *Proc. of ACM SIGCHI Conference on Human Factors in Computing Systems*. 2013, pp. 2291–2300.

[Ken+93] R. S. Kennedy, N. E. Lane, K. S. Berbaum, and M. G. Lilienthal. "Simulator sickness questionnaire: An enhanced method for quantifying simulator sickness". In: *The international journal of aviation psychology* 3.3 (1993), pp. 203–220.

[KFS17] T. Kim, C. Frank, and T. Schack. "A systematic investigation of the effect of action observation training and motor imagery training on the development of mental representation structure and skill performance". In: *Frontiers in human neuroscience* 11 (2017), pp. 1–13.

[Kia+16] R. Kianifar, A. Lee, S. Raina, and D. Kulić. "Classification of squat quality with inertial measurement units in the single leg squat mobility test". In: *Annual Int. Conf. of the Engineering in Medicine and Biology Society (EMBC)*. IEEE. 2016, pp. 6273–6276.

[Kok+14] I. de Kok, J. Hough, C. Frank, D. Schlangen, and S. Kopp. "Dialogue structure of coaching sessions". In: *Proc. of the SemDial Workshop on the Semantics and Pragmatics of Dialogue (DialWatt)*. 2014, pp. 167–169.

[Kok+15] I. de Kok, J. Hough, F. Hülsmann, M. Botsch, D. Schlangen, and S. Kopp. "A multimodal system for real-time action instruction in motor skill learning". In: *Proc. of the Int. Conf. on Multimodal Interaction*. ACM. 2015, pp. 355–362.

[Kok+16] I. de Kok, J. Hough, D. Schlangen, and S. Kopp. "Deictic gestures in coaching interactions". In: *Proc. of the Workshop on Multimodal Analyses enabling Artificial Agents in Human-Machine Interaction*. ACM. 2016, pp. 10–14.

[Kok+17] I. de Kok, F. Hülsmann, T. Waltemate, C. Frank, J. Hough, T. Pfeiffer, D. Schlangen, T. Schack, M. Botsch, and S. Kopp. "The Intelligent Coaching Space: A Demonstration". In: *Int. Conf. on Intelligent Virtual Agents*. Springer. 2017, pp. 105–108.

[KP01] E. J. Keogh and M. J. Pazzani. "Derivative dynamic time warping". In: *Proc. of the Int. Conf. On Data Mining*. SIAM. 2001, pp. 1–11.

[Krü+17] B. Krüger, A. Vögele, T. Willig, A. Yao, R. Klein, and A. Weber. "Efficient unsupervised temporal segmentation of motion data". In: *IEEE Transactions on Multimedia* 19.4 (2017), pp. 797–812.

[KV12] M. Keetels and J. Vroomen. "Exposure to delayed visual feedback of the hand changes motor-sensory synchrony perception". In: *Experimental brain research* 219.4 (2012), pp. 431–440.

[KW12] S. S. Kantak and C. J. Winstein. "Learning–performance distinction and memory processes for motor skills: A focused review and perspective". In: *Behavioural brain research* 228.1 (2012), pp. 219–231.

[Kya+15] M. Kyan, G. Sun, H. Li, L. Zhong, P. Muneesawang, N. Dong, B. Elder, and L. Guan. "An approach to ballet dance training through MS kinect and visualization in a cave virtual reality environment". In: *ACM Transactions on Intelligent Systems and Technology (TIST)* 6.2 (2015), p. 23.

[Lan+13] W. Land, D. Volchenkov, B. E. Bläsing, and T. Schack. "From action representation to action execution: exploring the links between cognitive and biomechanical levels of motor control". In: *Frontiers in computational neuroscience* 7 (2013), pp. 1–14.

[LH09] M. R. Longo and P. Haggard. "Sense of agency primes manual motor responses". In: *Perception* 38.1 (2009), pp. 69–78.

[Li+17a] J. Li, K. Cheng, S. Wang, F. Morstatter, R. P. Trevino, J. Tang, and H. Liu. "Feature selection: A data perspective". In: *ACM Computing Surveys (CSUR)* 50.6 (2017), pp. 1–45.

[Li+17b] W. Li, L. Wen, M.-C. Chang, S. N. Lim, and S. Lyu. "Adaptive rnn tree for large-scale human action recognition". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 1444–1452.

[Liu+17a] J. Liu, A. Shahroudy, D. Xu, A. K. Chichung, and G. Wang. "Skeleton-based action recognition using spatio-temporal lstm network with trust gates". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2017), pp. 3007–3021.

[Liu+17b] J. Liu, G. Wang, P. Hu, L.-Y. Duan, and A. C. Kot. "Global context-aware attention lstm networks for 3d action recognition". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 1647–1656.

[Liu+18a] J. Liu, Y. Li, S. Song, J. Xing, C. Lan, and W. Zeng. "Multi-Modality Multi-Task Recurrent Neural Network for Online Action Detection". In: *IEEE Transactions on Circuits and Systems for Video Technology* (2018), pp. 1–14.

[Liu+18b] J. Liu, G. Wang, L.-Y. Duan, K. Abdiyeva, and A. C. Kot. "Skeleton-based human action recognition with global context-aware attention LSTM networks". In: *IEEE Transactions on Image Processing* 27.4 (2018), pp. 1586–1599.

[LL92] H.-J. Lander and K. Lange. "Eine differentialpsychologische Analyse begrifflich-strukturierten Wissens". In: *Zeitschrift für Psychologie mit Zeitschrift für angewandte Psychologie* 200.3 (1992), pp. 181–197.

[LSG91] J. Liang, C. Shaw, and M. Green. "On temporal-spatial realism in the virtual reality environment". In: *Proc. of ACM symposium on User interface software and technology*. 1991, pp. 19–25.

[LT07] D. Liu and E. Todorov. "Evidence for the flexible sensorimotor strategies predicted by optimal feedback control". In: *The Journal of Neuroscience* 27.35 (2007), pp. 9354–9368.

[MA12] R. A. Magill and D. I. Anderson. "The roles and uses of augmented feedback in motor skill acquisition." In: ed. by H. N. J. and W. A. M. Skill acquisition in sport: Research, theory and practice. Routledge, 2012, pp. 3–21.

[Mag01] R. A. Magill. "Augmented feedback in motor skill acquisition". In: ed. by R. N. Singer, H. A. Hausenblas, and J. C. M. Handbook of sport psychology. Wiley New York, 2001, pp. 86–114.

[Man+04] K. Mania, B. D. Adelstein, S. R. Ellis, and M. I. Hill. "Perceptual sensitivity to head tracking latency in virtual environments with varying degrees of scene complexity". In: *Proc. of ACM Symposium on Applied perception in graphics and visualization*. 2004, pp. 39–47.

[Mar+15] Martín Abadi et al. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. Software available from tensorflow.org. 2015.

[Mat+18] A. Mathis, P. Mamidanna, T. Abe, K. M. Cury, V. N. Murthy, M. W. Mathis, and M. Bethge. "Markerless tracking of user-defined features with deep learning". In: *arXiv preprint arXiv:1804.03142* (2018).

[Mau+04] M. Mauve, J. Vogel, V. Hilt, and W. Effelsberg. "Local-lag and timewarp: providing consistency for replicated continuous applications". In: *IEEE Transactions on Multimedia* 6.1 (2004), pp. 47–57.

[MBW07] F. Marschall, A. Bund, and J. Wiemeyer. "Does frequent augmented feedback really degrade learning? A meta-analysis". In: *Bewegung und Training* 1 (2007), pp. 75–86.

[MBZ76] R. Martens, L. Burwitz, and J. Zuckerman. "Modeling effects on motor performance". In: *Research Quarterly. American Alliance for Health, Physical Education and Recreation* 47.2 (1976), pp. 277–291.

[MC12] J. Min and J. Chai. "Motion graphs++: a compact generative model for semantic motion analysis and synthesis". In: *ACM Transactions on Graphics (TOG)* 31.6 (2012), pp. 1–12.

[Mee+03] M. Meehan, S. Razzaque, M. C. Whitton, and F. P. Brooks Jr. "Effect of latency on presence in stressful virtual environments". In: *Proc. of IEEE Virtual Reality*. 2003, pp. 141–148.

[Meh+17a] D. Mehta, H. Rhodin, D. Casas, P. Fua, O. Sotnychenko, W. Xu, and C. Theobalt. "Monocular 3d human pose estimation in the wild using improved cnn supervision". In: *Int. Conf. on 3D Vision (3DV)*. IEEE. 2017, pp. 506–516.

[Meh+17b] D. Mehta, S. Sridhar, O. Sotnychenko, H. Rhodin, M. Shafiei, H.-P. Seidel, W. Xu, D. Casas, and C. Theobalt. "Vnect: Real-time 3d human pose estimation with a single rgb camera". In: *ACM Transactions on Graphics (TOG)* 36.4 (2017), pp. 1–14.

[MGB09] A. Muscariello, G. Gravier, and F. Bimbot. "Variability tolerant audio motif discovery". In: *Advances in Multimedia Modeling* (2009), pp. 275–286.

[Mil+12] H. C. Miles, S. R. Pop, S. J. Watt, G. P. Lawrence, and N. W. John. "A review of virtual environments for training in ball sports". In: *Computers & Graphics* 36.6 (2012), pp. 714–726.

[MLS12] P. McCullagh, B. Law, and D. Ste-Marie. "Modeling and Performance". In: ed. by S. Murphy. The Oxford Handbook of Sport and Performance Psychology. Oxford University Press, 2012, pp. 250–272.

[Mül07] M. Müller. *Information retrieval for music and motion*. Springer Science & Business Media, 2007.

[MW93] I. S. MacKenzie and C. Ware. "Lag as a determinant of human performance in interactive systems". In: *Proc. of the ACM INTERACT'93 and CHI'93 conference on Human factors in computing systems*. 1993, pp. 488–493.

[NAM01] A. Nanopoulos, R. Alcock, and Y. Manolopoulos. "Feature-based classification of time-series data". In: *International Journal of Computer Research* 10.3 (2001), pp. 49–61.

[Neu+18] D. L. Neumann, R. L. Moffitt, P. R. Thomas, K. Loveday, D. P. Watling, C. L. Lombard, S. Antonova, and M. A. Tremeer. "A systematic review of the application of interactive virtual reality to sport". In: *Virtual Reality* 22.3 (2018), pp. 183–198.

[Núñ+18] J. C. Núñez, R. Cabido, J. J. Pantrigo, A. S. Montemayor, and J. F. Vélez. "Convolutional Neural Networks and Long Short-Term Memory for skeleton-based human activity and hand gesture recognition". In: *Pattern Recognition* 76 (2018), pp. 80–94.

[ORe+15] M. O'Reilly, D. Whelan, C. Chanialidis, N. Friel, E. Delahunt, T. Ward, and B. Caulfield. "Evaluating squat performance with a single inertial measurement unit". In: *Int. Conf. on Wearable and Implantable Body Sensor Networks (BSN)*. IEEE. 2015, pp. 1–6.

[Par+18] A. Parziale, M. Diaz, M. A. Ferrer, and A. Marcelli. "SM-DTW: Stability Modulated Dynamic Time Warping for signature verification". In: *Pattern Recognition Letters* (2018), pp. 1–11.

[Ped+11] F. Pedregosa et al. "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[Pet+14] F. Petitjean, G. Forestier, G. I. Webb, A. E. Nicholson, Y. Chen, and E. Keogh. "Dynamic time warping averaging of time series allows faster and more accurate classification". In: *Int. Conf. on Data Mining*. IEEE. 2014, pp. 470–479.

[PK99] K. S. Park and R. V. Kenyon. "Effects of network characteristics on human performance in a collaborative virtual environment". In: *Proc. of IEEE Virtual Reality*. 1999, pp. 104–111.

[PMW16] G. I. Parisi, S. Magg, and S. Wermter. "Human motion assessment in real time using recurrent self-organization". In: *International Symposium on Robot and Human Interactive Communication (RO-MAN)*. IEEE. 2016, pp. 71–76.

[Pot98] M. Potel. "Motion sick in cyberspace". In: *IEEE Computer Graphics and Applications* 18.1 (1998), pp. 16–21.

[RA04] J. J. Rodríguez and C. J. Alonso. "Interval and dynamic time warping-based decision trees". In: *Proc. of the ACM symposium on Applied computing*. 2004, pp. 548–552.

[Rah+18] S. Rahm, K. Wieser, D. E. Bauer, F. W. Waibel, D. C. Meyer, C. Gerber, and S. F. Fucentese. "Efficacy of standardized training on a virtual reality simulator to advance knee and shoulder arthroscopic motor skills". In: *BMC musculoskeletal disorders* 19.1 (2018), pp. 1–7.

[RBK13] K. Rector, C. L. Bennett, and J. A. Kientz. "Eyes-free yoga: an exergame using depth cameras for blind & low vision exercise". In: *Proc. of International ACM SIGACCESS Conference on Computers and Accessibility*. 2013, 12:1–12:8.

[RDE11] M. Reyes, G. Domínguez, and S. Escalera. "Feature weighting in dynamic time warping for gesture recognition in depth data". In: *Computer Vision Workshops (ICCV Workshops)*. IEEE. 2011, pp. 1182–1188.

[RE12] M. Rohde and M. O. Ernst. "To lead and to lag–forward and backward recalibration of perceived visuo-motor simultaneity". In: *Frontiers in psychology* 3 (2012).

[Rec+17] K. Rector, R. Vilardaga, L. Lansky, K. Lu, C. L. Bennett, R. E. Ladner, and J. A. Kientz. "Design and Real-World Evaluation of Eyes-Free Yoga: An Exergame for Blind and Low-Vision Exercise". In: *ACM Transactions on Accessible Computing (TACCESS)* 9.4 (2017), pp. 1–25.

[Rei+11] D. Reidsma, I. de Kok, D. Neiberg, S. C. Pammi, B. van Straalen, K. Truong, and H. van Welbergen. "Continuous interaction with a virtual human". In: *Journal on Multimodal User Interfaces* 4.2 (2011), pp. 97–118.

[RK05] A. Rizzo and G. Kim. "A SWOT analysis of the field of virtual reality rehabilitation and therapy". In: *Presence* 14.2 (2005), pp. 119–146.

[RP11] H. Rohbanfard and L. Proteau. "Learning through observation: a combination of expert and novice models favors learning". In: *Experimental brain research* 215.3-4 (2011), pp. 183–197.

[RSS95] D. J. Roberts, P. M. Sharkey, and P. Sandoz. "A real-time, predictive architecture for Distributed Virtual Reality". In: *Proc. of ACM SIGGRAPH Workshop on Simulation & Interaction in Virtual Environments*. 1995, pp. 1–10.

[Sal+10] P. Salamin, T. Tadi, O. Blanke, F. Vexo, and D. Thalmann. "Quantifying effects of exposure to the third and first-person perspectives in virtual-reality-based training". In: *IEEE Transactions on Learning Technologies* 3 (2010), pp. 272–276.

[SC07] S. Salvador and P. Chan. "Toward accurate dynamic time warping in linear time and space". In: *Intelligent Data Analysis* 11.5 (2007), pp. 561–580.

[SC78] H. Sakoe and S. Chiba. "Dynamic programming algorithm optimization for spoken word recognition". In: *IEEE transactions on acoustics, speech, and signal processing* 26.1 (1978), pp. 43–49.

[Sch+14] T. Schack, M. Bertollo, D. Koester, J. Maycock, and K. Essig. "Technological advancements in sport psychology". In: ed. by A. G. Papaioannou and D. Hackfort. Routledge Companion to Sport and Exercise Psychology: Global perspectives and fundamental concepts. Routledge, 2014, pp. 953–965.

[Sch+89] R. A. Schmidt, D. E. Young, S. Swinnen, and D. C. Shapiro. "Summary knowledge of results for skill acquisition: Support for the guidance hypothesis." In: *Journal of Experimental Psychology: Learning, Memory, and Cognition* 15.2 (1989), pp. 352–359.

[Sch03] T. Schack. "Kognition und Emotion". In: ed. by H. Mechling and J. Munzert. Handbuch Bewegungswissenschaft Bewegungslehre. Hofmann, 2003, pp. 313–330.

[Sch04] T. Schack. "The cognitive architecture of complex movement". In: *International journal of sport and exercise psychology* 2.4 (2004), pp. 403–438.

[Sch12] T. Schack. "Measuring mental representations". In: ed. by G. Tenenbaum, R. C. Eklund, and A. Kamata. Measurement in sport and exercise psychology. Human Kinetics, 2012, pp. 203–214.

[Sch91] R. A. Schmidt. "Frequent augmented feedback can degrade learning: Evidence and interpretations". In: *Tutorials in motor neuroscience*. Springer, 1991, pp. 59–75.

[SE09] J. M. Santos and M. Embrechts. "On the use of the adjusted rand index as a metric for evaluating supervised classification". In: *Int. Conf. on Artificial Neural Networks*. Springer. 2009, pp. 175–184.

[SH10] G. Skantze and A. Hjalmarsson. "Towards incremental speech generation in dialogue systems". In: *Proc. of the 11th Annual Meeting of the Special Interest Group on Discourse and Dialogue*. Association for Computational Linguistics. 2010, pp. 1–8.

[Shn84] B. Shneiderman. "Response Time and Display Rate in Human Performance with Computers". In: *ACM Computing Surveys* 16.3 (1984), pp. 265–285.

[Sig+13] R. Sigrist, G. Rauter, R. Riener, and P. Wolf. "Augmented visual, auditory, haptic, and multimodal feedback in motor learning: a review". In: *Psychonomic bulletin & review* 20.1 (2013), pp. 21–53.

[Sig+15] R. Sigrist, G. Rauter, L. Marchal-Crespo, R. Riener, and P. Wolf. "Sonification and haptic feedback in addition to visual feedback enhances complex motor task learning". In: *Experimental brain research* 233.3 (2015), pp. 909–925.

[SM06] T. Schack and F. Mechsner. "Representation of motor skills in human long-term memory". In: *Neuroscience letters* 391.3 (2006), pp. 77–81.

[Sme+14] J. D. Smeddinck, J. Voges, M. Herrlich, and R. Malaka. "Comparing modalities for kinesiatric exercise instruction". In: *ACM CHI'14 Extended Abstracts on Human Factors in Computing Systems*. 2014, pp. 2377–2382.

[SN85] D. Scully and K. Newell. "Observational-learning and the acquisition of motor-skills-toward a visual-perception perspective". In: *Journal of human movement studies* 11.4 (1985), pp. 169–186.

[SNL18] J.-P. Stauffert, F. Niebling, and M. E. Latoschik. "Effects of Latency Jitter on Simulator Sickness in a Search Task". In: *2018 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*. 2018, pp. 121–127.

[SR13] T. Schack and H. Ritter. "Representation and learning in motor action–bridges between experimental research and cognitive robotics". In: *New ideas in psychology* 31.3 (2013), pp. 258–269.

[SSW84] A. W. Salmoni, R. A. Schmidt, and C. B. Walter. "Knowledge of results and motor learning: a review and critical reappraisal." In: *Psychological bulletin* 95.3 (1984), pp. 235–244.

[Ste08] A. Steed. "A Simple Method for Estimating the Latency of Interactive, Real-Time Graphics Simulations". In: *ACM Symposium on Virtual Reality Software and Technology*. 2008, pp. 123–129.

[Sve+04] V. Svetnik, A. Liaw, C. Tong, and T. Wang. "Application of Breiman's random forest to modeling structure-activity relationships of pharmaceutical molecules". In: *International Workshop on Multiple Classifier Systems*. Springer. 2004, pp. 334–343.

[SVZ13] K. Simonyan, A. Vedaldi, and A. Zisserman. "Deep inside convolutional networks: Visualising image classification models and saliency maps". In: *arXiv preprint arXiv:1312.6034* (2013), pp. 1–8.

[SW97] R. A. Schmidt and G. Wulf. "Continuous concurrent feedback degrades skill learning: Implications for training and simulation". In: *Human factors* 39.4 (1997), pp. 509–525.

[Tan+15] R. Tang, X.-D. Yang, S. Bateman, J. Jorge, and A. Tang. "Physio@ Home: Exploring visual guidance and feedback techniques for physiotherapy exercises". In: *Proc. of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. 2015, pp. 4123–4132.

[Tan+18] J. Tang, H. Cheng, Y. Zhao, and H. Guo. "Structured dynamic time warping for continuous hand trajectory gesture recognition". In: *Pattern Recognition* 80 (2018), pp. 21–31.

[Tay+10] P. E. Taylor, G. J. Almeida, T. Kanade, and J. K. Hodgins. "Classifying human motion quality for knee osteoarthritis using accelerometers". In: *Annual Int. Conf. of the IEEE Engineering in Medicine and Biology*. 2010, pp. 339–343.

[Tor+09] P. Tormene, T. Giorgino, S. Quaglini, and M. Stefanelli. "Matching incomplete time series with dynamic time warping: an algorithm and an application to post-stroke rehabilitation". In: *Artificial intelligence in medicine* 45.1 (2009), pp. 11–34.

[TSB97] E. Todorov, R. Shadmehr, and E. Bizzi. "Augmented feedback presented in a virtual environment accelerates learning of a difficult motor task". In: *Journal of motor behavior* 29.2 (1997), pp. 147–158.

[UKR14] N. Ukita, D. Kaulen, and C. Rocker. "Towards an automatic motion coaching system". In: *Int. Conf. on Physiological Computing System*. 2014.

[Uso+00] M. Usoh, E. Catena, S. Arman, and M. Slater. "Using presence questionnaires in reality". In: *Presence: Teleoperators & Virtual Environments* 9.5 (2000), pp. 497–503.

[VBG13] E. Velloso, A. Bulling, and H. Gellersen. "MotionMA: motion modelling and analysis by demonstration". In: *Proc. of the SIGCHI Conference on Human Factors in Computing Systems*. ACM. 2013, pp. 1309–1318.

[Vil+07] H. Vilhjálmsson, N. Cantelmo, J. Cassell, N. E. Chafai, M. Kipp, S. Kopp, M. Mancini, S. Marsella, A. N. Marshall, C. Pelachaud, et al. "The behavior markup language: Recent developments and challenges". In: *International Workshop on Intelligent Virtual Agents*. Springer. 2007, pp. 99–111.

[VKG02] M. Vlachos, G. Kollios, and D. Gunopulos. "Discovering similar multidimensional trajectories". In: *18th Int. Conf. on Data Engineering*. IEEE. 2002, pp. 673–684.

[VMM09] N. Van Hanh, F. Merienne, and J. L. Martinez. "Effects of virtual avatar characteristics on performance of healthy subjects' training tasks". In: *Journal of CyberTherapy and Rehabilitation* 2.3 (2009), pp. 221–234.

[Von+18] E. K. Vonstad, X. Su, B. Vereijken, J. H. Nilsen, and K. Bach. "Classification of Movement Quality in a Weight-Shifting Exercise". In: *3rd International Workshop on Knowledge Discovery in Healthcare Data*. 2018, pp. 27–32.

[VYK14] H. Van Welbergen, R. Yaghoubzadeh, and S. Kopp. "AsapRealizer 2.0: The next steps in fluent behavior realization for ECAs". In: *Int. Conf. on Intelligent Virtual Agents*. Springer. 2014, pp. 449–462.

[Wal+15] T. Waltemate, F. Hülsmann, T. Pfeiffer, S. Kopp, and M. Botsch. "Realizing a low-latency virtual reality environment for motor learning". In: *ACM Symposium on Virtual Reality Software and Technology*. 2015, pp. 139–147.

[Wal+16] T. Waltemate, I. Senna, F. Hülsmann, M. Rohde, S. Kopp, M. Ernst, and M. Botsch. "The impact of latency on perceptual judgments and motor performance in closed-loop interaction in virtual reality". In: *ACM Symposium on Virtual Reality Software and Technology*. 2016, pp. 27–35.

[Wal+18] T. Waltemate, D. Gall, D. Roth, M. Botsch, and M. E. Latoschik. "The Impact of Avatar Personalization and Immersion on Virtual Body Ownership, Presence, and Emotional Response". In: *IEEE Transactions on Visualization and Computer Graphics* 24.4 (2018), pp. 1643–1652.

[WB94] C. Ware and R. Balakrishnan. "Reaching for objects in VR displays: lag and frame rate". In: *ACM Transactions on Computer-Human Interaction* 1.4 (1994), pp. 331–356.

[WB99] A. D. Wilson and A. F. Bobick. "Parametric hidden markov models for gesture recognition". In: *IEEE transactions on pattern analysis and machine intelligence* 21.9 (1999), pp. 884–900.

[WC04] Y. Wu and E. Y. Chang. "Distance-function design and fusion for sequence data". In: *Proc. of the thirteenth ACM Int. Conf. on Information and knowledge management*. 2004, pp. 324–333.

[Wei+16] S.-E. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh. "Convolutional pose machines". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 4724–4732.

[Wen+18] J. Weng, M. Liu, X. Jiang, and J. Yuan. "Deformable pose traversal convolution for 3d action and gesture recognition". In: *Europen Conference on Computer Vision (ECCV)*. Vol. 2. 5. 2018, pp. 142–157.

[Wes+00] J. Weston, S. Mukherjee, O. Chapelle, M. Pontil, T. Poggio, and V. Vapnik. "Feature selection for SVMs". In: *Advances in Neural Information Processing Systems (NIPS)* (2000), pp. 668–674.

[WNB06] D. E. Warburton, C. W. Nicol, and S. S. Bredin. "Health benefits of physical activity: the evidence". In: *Canadian medical association journal* 174.6 (2006), pp. 801–809.

[WS90] C. J. Winstein and R. A. Schmidt. "Reduced frequency of knowledge of results enhances motor skill learning." In: *Journal of Experimental Psychology: Learning, Memory, and Cognition* 16.4 (1990), p. 677.

[WS98] B. G. Witmer and M. J. Singer. "Measuring presence in virtual environments: A presence questionnaire". In: *Presence: Teleoperators and virtual environments* 7.3 (1998), pp. 225–240.

[Xi+06] X. Xi, E. Keogh, C. Shelton, L. Wei, and C. A. Ratanamahatana. "Fast time series classification using numerosity reduction". In: *Proc. of the 23rd Int. Conf. on Machine learning*. ACM. 2006, pp. 1033–1040.

[YB14] A. Yurtman and B. Barshan. "Automated evaluation of physical therapy exercises using multi-template dynamic time warping on wearable sensor signals". In: *Computer methods and programs in biomedicine* 117.2 (2014), pp. 189–207.

[Yua+18] J. Yuan, A. Douzal-Chouakria, S. V. Yazdi, and Z. Wang. "A large margin time series nearest neighbour classification under locally weighted time warps". In: *Knowledge and Information Systems* (2018), pp. 1–19.

[Zei12] M. D. Zeiler. "ADADELTA: an adaptive learning rate method". In: *arXiv preprint arXiv:1212.5701* (2012).

[Zha+17] W. Zhao, M. A. Reinthal, D. D. Espy, and X. Luo. "Rule-based human motion tracking for rehabilitation exercises: realtime assessment, feedback, and guidance". In: *IEEE Access* 5 (2017), pp. 21382–21394.

[Zha+18] P. Zhang, J. Xue, C. Lan, W. Zeng, Z. Gao, and N. Zheng. "Adding Attentiveness to the Neurons in Recurrent Neural Networks". In: *arXiv preprint arXiv:1807.04445* (2018).

[ZHT06] H. Zou, T. Hastie, and R. Tibshirani. "Sparse principal component analysis". In: *Journal of computational and graphical statistics* 15.2 (2006), pp. 265–286.

APPENDICES

A.1 DESCRIPTION OF ANALYSIS FOR CHAPTER 3

This appendix describes the parameters that we used to measure motor performance for the experiments described in Chapter 3.

A.1.1 *Temporal and Spatial Error based on Dynamic Time Warping (DTW)*

We use DTW to extract a spatial and a temporal error value for a participant's performance based on the comparison to the skilled performance. For a detailed explanation of DTW see Chapter 4. Here, we perform DTW based on the joint positions. We use all joints, but the root joint, as well as three joints in the back (spine markers placed at l2, t5, and t10). We exclude these joints as we would like to mainly focus on the movement of the extremities and the joints in the back tend to induce a high level of noise in our setup. The self-similarity Matrix $\mathbf{M}$ that is needed for DTW is constructed based the following distance function:

$$\mathbf{M}(i,j) = \sum_{d=1}^{k} ||\mathbf{t}_{participant,d}(i) - \mathbf{t}_{skilled,d}(j)||.$$

Each element (i,j) of this matrix corresponds to the distances between the postures in the trajectory of the participant $T_{participant}(i)$ and the skilled trajectory $T_{skilled}(j)$. Here, $\mathbf{t}_{participant,d}(i)$ denotes the translation of joint d at frame i of the movement of the participant, $\mathbf{t}_{skilled,d}(j)$ denotes the translation of the same joint at frame j of the skilled movement, and k denotes the number of joints. Based on dynamic programming, we determine an optimal path of corresponding frames through this matrix according to [Mül07, p. 69] (cf. Chapter 4).

We extract two features based on DTW: the temporal as well as the spatial error. The temporal error is calculated as follows: For each frame in a participant's movement, we calculate the change in the temporal offset from the performed movement of the skilled movement. Example: If frame 200 of the participant's movement maps to frame 210 of the skilled movement and frame 201 of the participant's movement maps to frame 215, the error at frame 201 is -4. Finally, we return the RMSE of these shifts. To be more specific, we perform the following calculation:

$$error_{temporal} = \left(\frac{1}{|T_{participant}|} \sum_{f=1}^{|T_{participant}|} \Big(\big(f - \mathrm{w}(f)\big) - \big((f-1) - \mathrm{w}(f-1)\big) \Big)^2 \right)^{\frac{1}{2}}.$$

Here, $\mathrm{w}(f)$ is the frame number of the skilled movement that is mapped on frame number f of the participant's movement according to the frame-wise correspondences calculated by DTW. If, according to the optimal path, multiple frames of the skilled movement map on the same frame of the participant's movement, we select the one

that is in the middle of these frames on the temporal axis. The spatial error is the averaged value of $\mathbf{M}$ on the optimal path:

$$error_{spatial} = \frac{1}{L} \sum_{\xi=1}^{L} \mathbf{M}(p_\xi).$$

Here, p specifies the optimal alignment path through $\mathbf{M}$ that is calculated by DTW. See [Mül07, p. 69] for a formal definition. Each entry p_ξ is a tuple $(i, j) \in p$ that contains the frame numbers i and j of the trajectories $T_{participant}$ and $T_{skilled}$ that correspond to each other, i.e., that lie on the optimal path. L denotes the length of the optimal path.

A.1.2 *Center of Mass at the Deepest Point*

We estimate a simplified center of mass based on the centroid of the joint positions. More specifically, the center of mass of a trajectory at the deepest point of the squat is calculated as follows:

$$\mathbf{com} = \frac{1}{k} \sum_{d=1}^{k} \mathbf{t}_d(f_{deepest}).$$

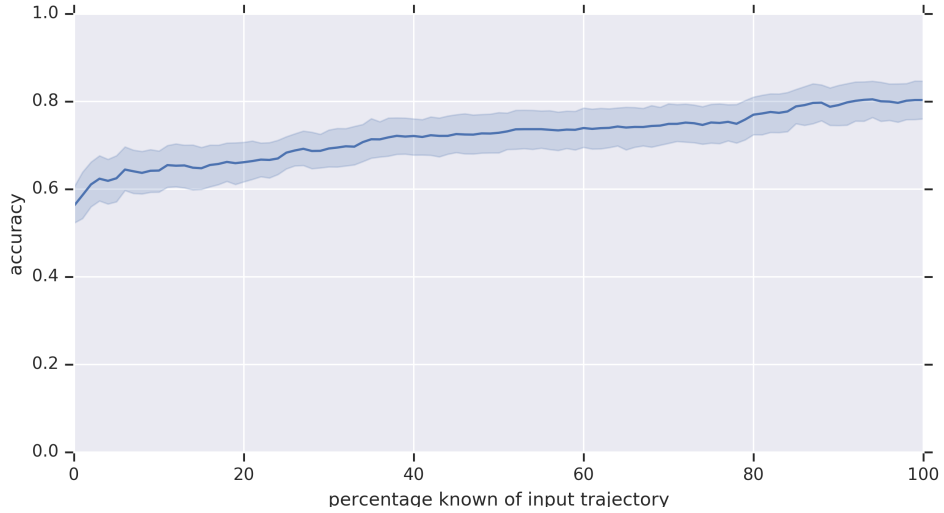
Here, k denotes the number of joints. $\mathbf{t}_d(f_{deepest})$ denotes the translation of joint d at the deepest point (frame $f_{deepest}$) of the squat.

A.1.3 *Principal Component Analysis*

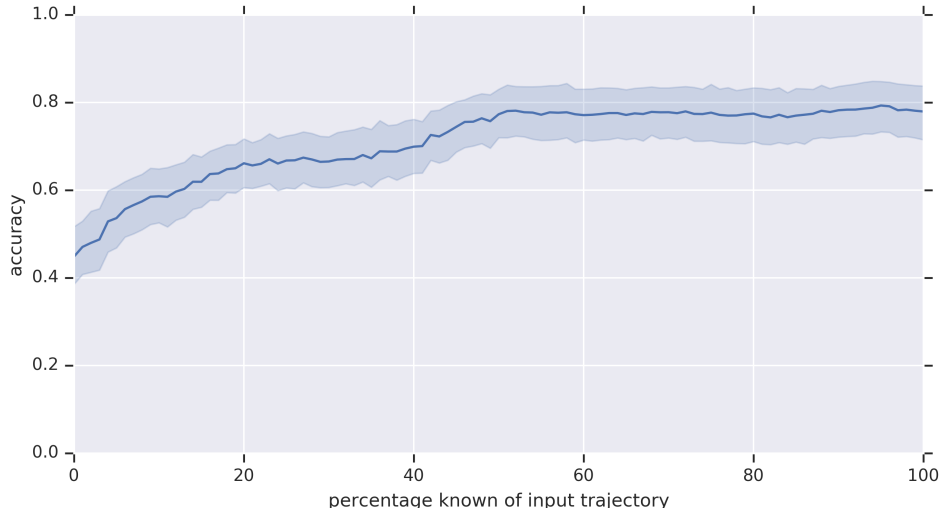
Principal Component Analysis (PCA) is typically used in the field of dimensionality reduction [Bis06]. For a given data set, it searches for a set of linear combinations that capture a given amount of variance inside the data. It reduces the high-dimensional data set into a smaller number of structural components. We determine the number of principal components needed to cover 85 % of the variance inside our data for each participant and each test phase (pre-test, post-test, retention-test). To focus only on the spatial properties, we first perform DTW between each trajectory of a participant T_i and the first trajectory of this participant in the given phase T_0. We use the correspondence path determined by DTW to warp each movement into the timing of T_0: For each frame of T_i, the corresponding frame in T_0 is extracted. Next, we construct a feature vector that consists of the joint translations of these frames. This vector has the length $3k|T_0|$, where k is the number of joints. $|T_0|$ is the length of trajectory T_0. Then we calculate the PCA based on the feature vectors for each participant and the test phases (pre-test, post-test, retention-test).

A.2 ADDITIONAL INFORMATION FOR CHAPTER 5: MORE RESULTS

This appendix provides additional results for Chapter 5. First, we provide information on the quality of the classification depending on the percentage of the input trajectory that is already known. Figure A.1 provides the results for the accuracy, Figure A.2 provides the results for the F1 score.



(a) Squat.

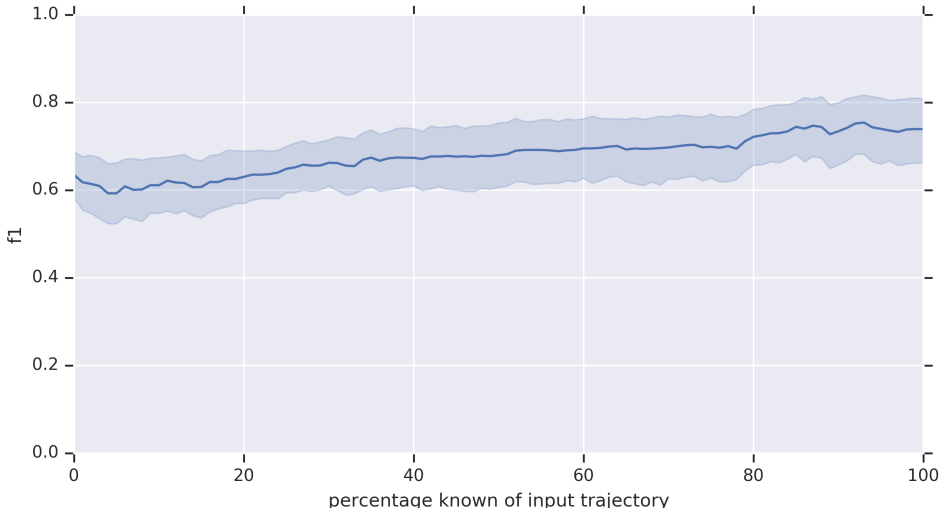


(b) Tai Chi push.

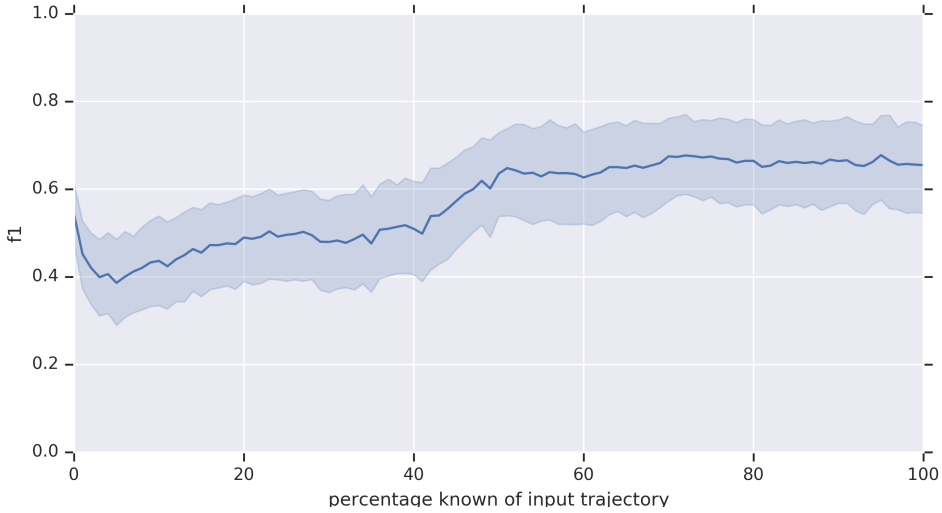
Figure A.1: Average accuracy of the classifier over all error patterns depending on the percentage of the input trajectory that is already known.

In addition to the accuracies for each error pattern depicted in Chapter 5, we show the f1 scores for each pattern in Figure A.3.

Further, we provide results on the comparison of different values for k in our baseline, the classification via kNN-DTW. Figure A.4 provides the accuracies for



(a) Squat.

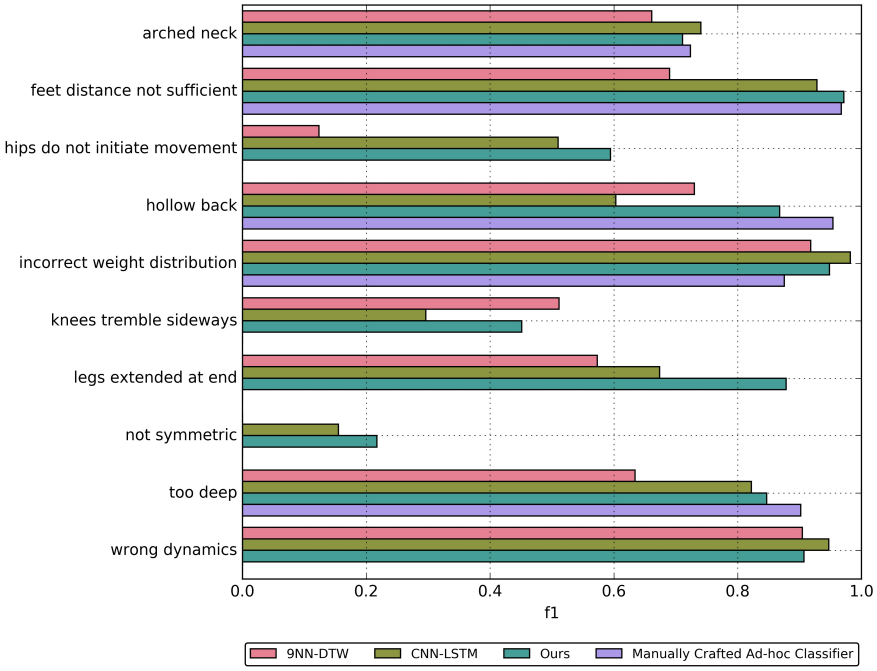


(b) Tai Chi push.

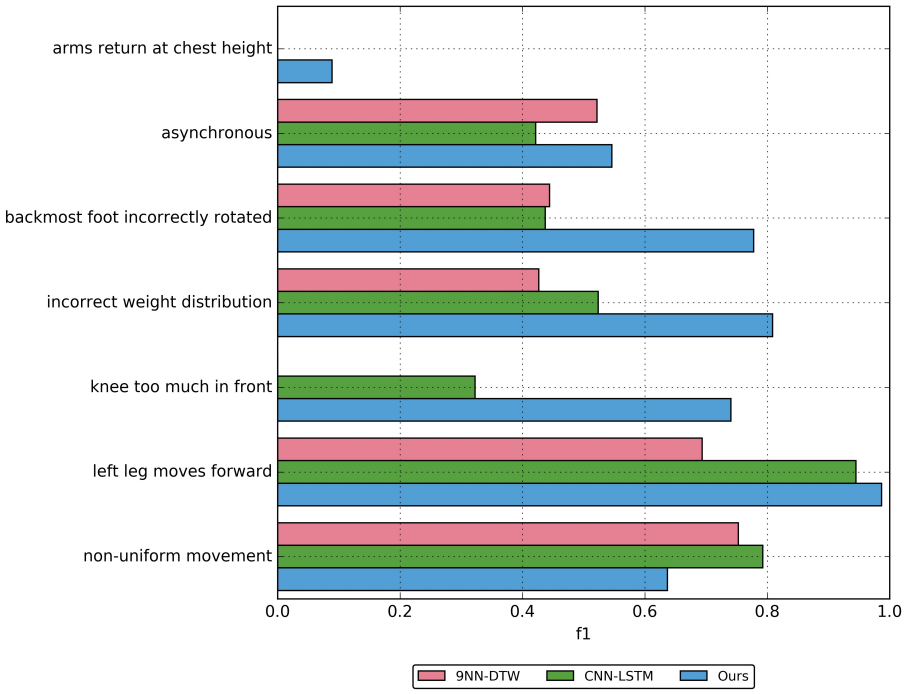
Figure A.2: Average F1 score of the classifier over all error patterns depending on the percentage of the input trajectory that is already known.

kNN-DTW with $k \in [1, \ldots, 11]$. We observe, that the best results can be observed for $k = 9$.

Figure A.5 shows the performance of our classifier when no feature selection is performed and the full feature vector of size $6|T_r|k$ is directly fed into the SVM.

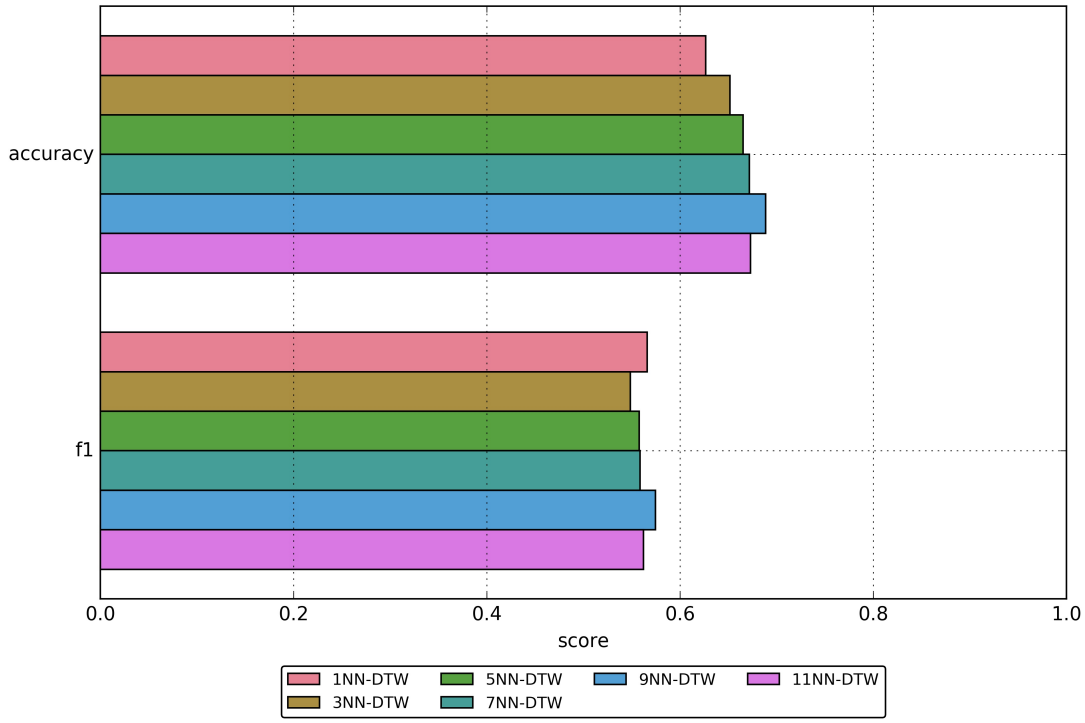


(a) Squat.

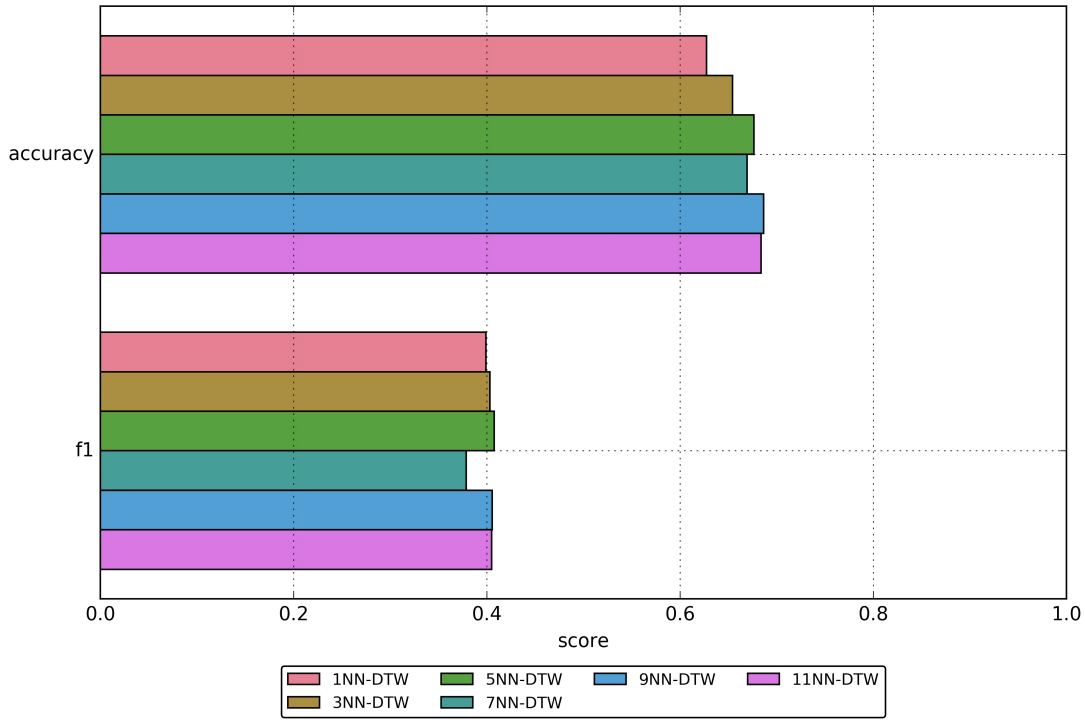


(b) Tai Chi push.

Figure A.3: F1 scores of the classifiers for each error pattern.

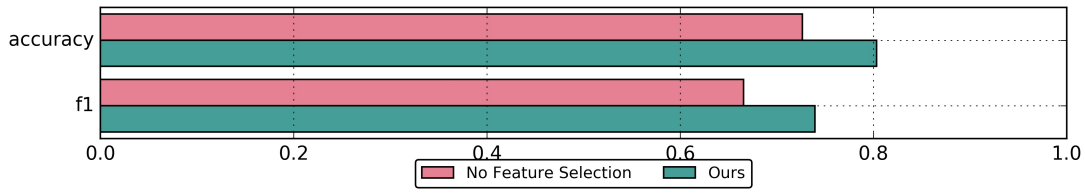


(a) Squat.

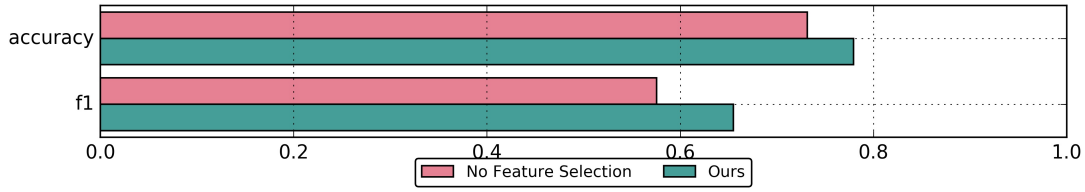


(b) Tai Chi push.

Figure A.4: Comparison of the impact of different values of k for kNN-DTW on the classification of typical errors in the squat and in the Tai Chi push. We provide results for accuracy and F1 score.



(a) Squat.



(b) Tai Chi push.

Figure A.5: Performance of our classifier when no feature selection is performed as compared to our full classifier. We provide results for accuracy and F1 score.

A.3 PILOT STUDY ON SIMPLE TEXTUAL FEEDBACK

Published in:
[Kok+15; Wal+15]

To investigate the usability of the core components in a pilot study, we developed a simple coaching application that offers squat training by exploration. Subjects performed squats in front of a virtual mirror and obtained, after each squat, textual information on the type of error they performed. In addition to task performance, we measured subjective ratings of the environment. Here, we focus on three measures: Simulator Sickness, Presence, and the athletes perceived control of their avatar.

To assess each performed squat, the hierarchical state-based analysis as described in Chapter 5 was used. The primary aim of this study is to rapidly evaluate whether our system is usable and promising for its application. Our main contributions are:

- Showing the combination of training environment (cf. Chapter 2) and classification of motor errors (cf. Chapter 5) in a simplistic way
- Demonstrating that the system can be applied in the context of motor learning. This is achieved via demonstrating participants' ability to reduce their motor error even when only obtaining highly simplistic feedback as well as via measuring the participants' subjective perception of the environment by using questionnaires.

My Contribution *This appendix contains the pilot study published in [Kok+15; Wal+15]. I planned, realized, conducted and evaluated the study. Thomas Waltemate realized necessary adaptations and extensions of the renderer.*

A.3.1 *Materials and Methods*

Twenty-three participants (15 female; age $M = 26.17$, $SD = 8.94$) with normal or corrected to normal vision took part in the study. Participants provided written

informed consent and got paid for their participation. The study was conducted in accordance with the Declaration of Helsinki, and had ethical approval from the ethics committee of the Bielefeld University.

For this pilot study, we used the setup described in Chapter 2. Every subject was placed in a virtual room with a virtual mirror in front of them. Inside the virtual mirror, the subject's avatar was shown. We rendered the high-resolution default character including shadow mapping (cf. Chapter 2) as avatar for all subjects. See Figure A.6 for an overview of the rendered environment.

Participants were welcomed, read a description of the experiment and filled in a consent form. Next general questionnaires were filled in. Afterwards subjects were equipped with the motion capture suit and the passive markers. Next, they were placed inside the CAVE and calibrated inside the motion capture system. Then, the experimenter showed a video of an expert performing a squat and informed subjects to especially focus on depth, the position of the knee as well as on the back. Afterwards, the actual training began. Participants were instructed to perform squats in six sets with a break in between. Each set ended as soon as a squat had been performed correctly or as soon as a given time limit had been reached. During each set, the mirror showed a red tint until the trainee succeeded in performing a correct squat. After the performance of a correct squat, the mirror changed its color to green. To depict the detection of a squat, the mirror flashed yellow. If an error pattern had been detected during a squat, one keyword for the specific pattern which had been explained to the participant beforehand (e.g., "neck") was displayed next to the mirror directly after the performance. The error patterns were prioritized as follows: The pattern "not deep enough" had the highest priority, followed by "incorrect weight distribution", followed by "straight neck" followed by "straight back" (combination of "hump" and "hollow back"). Between sets, participants had a short break. During this break, the mirror looses its color. The whole interaction with the system took around 5–6 minutes. Afterwards further questionnaires were filled in and participants were payed. The subjects' performance was calculated based on the hierarchical state-based analysis which also served as a basis for the feedback the subjects obtained after each squat. We compared each subject's performance between the first and the last set of squats during the experiment. To this end, we used the Wilcoxon signed-rank test. In addition to the subjects' performance, we also included questionnaires on presence (questions based on a modified version of the Slater, Usoh, Steed questionnaire (SUS) [Uso+00] used in [FW13]), Simulator Sickness [Ken+93], as well as further subjective ratings including an evaluation of perceived control (7-point

Figure A.6: The subject did not go down deep enough ("Kniewinkel"). Inside the virtual mirror, which blinks yellow directly after a squat, the subject's motion is mapped on an avatar.

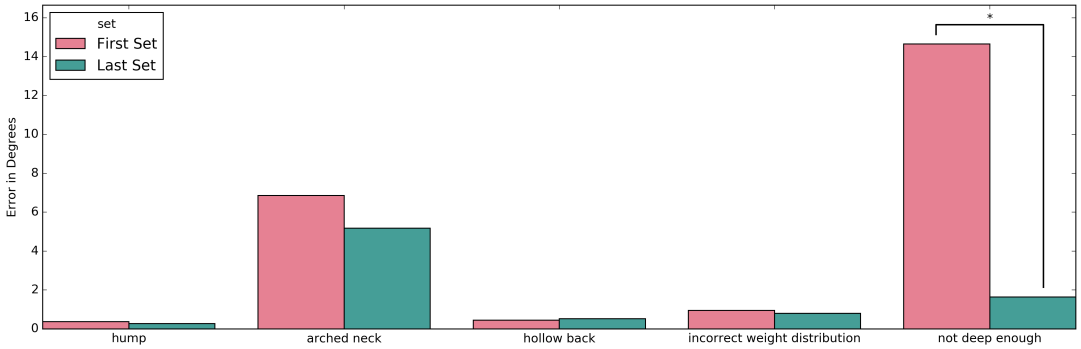


Figure A.7: Errors performed by the subjects summarized for the first and the last set of squats during the experiment.

Likert scale ranging from 0 (no control) to 6 (highest level of control). We evaluated the robustness of the motion capture setup via evaluating whether re-calibration was necessary during experiments after marker loss.

A.3.2 *Results and Discussion*

Participants were able to improve their performance from the first to the last set for the error pattern not deep enough ($M_{set1} = 14.65$, $M_{set6} = 1.64$, $p = 0.02$). For the pattern "hump", "arched neck" and "incorrect weight distribution" subjects became on average better, however these results were not significant. For the pattern "hollow back", subjects became slightly worse, however, these results were also not significant. See Figure A.7 for a summary of the performance-related results. Our results indicate that the subjects' performance with respect to the desired depth can be improved during to the interaction with our environment and the provided simplistic feedback. However, as this was only a pilot experiment, we did not compare to a control group. Consequently, we cannot point out the source of the improvement. Further, we did not test whether subjects were able to maintain their improved performance.

The degree of perceived control was measured using a 7-point scale ranging from -3 (no control) to 3 (highest level of control). We obtained a satisfying value of $M = 2$ ($SD = 1$). The results for presence were at an intermediate level ($M = 3.1$, $SD = 1.5$ on a scale from 0 to 6). The relatively low mean may have been due to the fact that this preliminary study used only visual feedback and a very simple virtual environment. According to the simulator sickness questionnaire, no increase of simulator sickness ($M_{pre} = 0.15$, $M_{post} = 0.13$), was induced due to the experiment. Concerning the robustness of the motion capture environment our results were also satisfying: Only one single time, too many markers were occluded which required a re-calibration of the participant. In all other trials, temporary loss of markers was compensated by the system.

A.3.3 *Conclusion*

Our results suggest that the virtual environment is technically sound for our field of application. We even obtained slight improvements in motor performance during the interaction with our system. However the feedback we provided was somewhat static, and not conductive to good interaction or enhancement of skill. For instance, we did not provide any information on error intensity and subjects needed to recall the information on how a good performance would look like from the beginning of the experiment. Further, the experimental setup was very simple as we did not perform any tests without feedback nor did we test a control group. In summary, the experiment allowed us to show that the environment is usable for our field of application and that we can expect it to be able to help subjects improving in performing motor actions. In the next steps, we evaluate suitable feedback strategies for motor learning in our setup and finally develop and evaluate a training scenario for squats that combines the information obtained in the subsequent chapters.

A.4 PILOT STUDY ON VERBAL FEEDBACK

Published in:
[Kok+15]

There has already been some work in systems that help users improve specific bodily movements, facilitating *motor skill learning*. While this has involved various types of auditory, visual and haptic feedback to optimize the learning gain of the user—see [Sig+13] for a review—little attention has been payed to generating *real-time instructions* as the motor skill is being attempted which uses comprehensive motion analysis, nor to the general verbal and gestural generation requirements of multimodal virtual coaching agents who could operate in such a domain with access to this detailed knowledge. In this pilot study, we address this unique challenge for motor skill coaching by virtual agents, using the intelligent coaching space environment presented in this thesis and introducing a virtual coach that can generate appropriate instructions as the motor skill is performed, based on the classification of typical error patterns (cf. Chapter 5.3). The coaching system that is presented here is used as a basis for the environment that is described in Chapter 6. This appendix only contains a shortened version of the original publication, see [Kok+15] for more details.

My Contribution *This appendix contains the study published in [Kok+15] as well as a description of the coaching system. I integrated my rule-based analysis (see Chapter 5.3) into the overall coaching system used in the study. I further supported Iwan de Kok and Julian Hough in conducting the study. Iwan de Kok and Julian Hough developed the virtual coach and planned and conducted the study. Further they performed the analysis of the results.*

A.4.1 *Apparatus*

For this study, we used the setup described in Chapter 2. We rendered the high-resolution default character (cf. Chapter 2) as avatar for all subjects. The virtual coach was located on the right hand side of the mirror. For both characters, we used the same model, but different textures for the clothes as well as different resolutions of the geometry. The coach character wore a shirt with the logo of our institution (CITEC), the subject's avatar wore dark clothes. For the coach we used a geometry of intermediate-resolution (around 80,000 triangles), for the subject's avatar, we used the low resolution geometry (around 20,000 triangles, cf. Chapter 2). See Figure A.8 for an overview of the rendered environment. The subjects' performance was calculated based on the hierarchical rule-based analysis. This information was send to the coach in real time. We focused on the error patterns Error patterns "not deep enough", "incorrect weight distribution", "arched neck" as well as "hollow back".

A.4.2 *Virtual Coach*

Our virtual coach aims to bring incremental situated coaching to our intelligent coaching space hitherto described. The software architecture of the Virtual Coach consists of three main components (cf. Figure A.9): The *Coaching Strategy Manager*

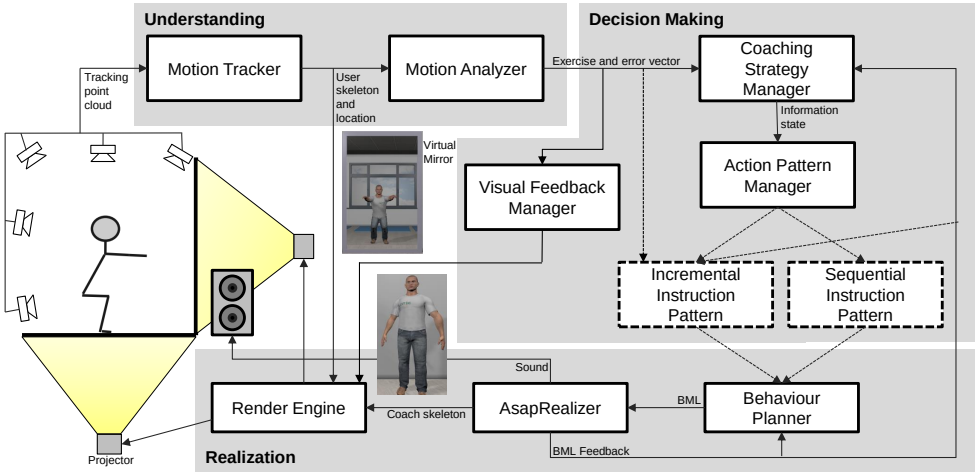


Figure A.9: The overall architecture of our intelligent coaching space. On the left the hardware setup is depicted, with wall and floor projection and motion capturing cameras. In our software we have understanding components interpreting the user's actions. Our decision makers comprise both visual feedback (middle part) and a virtual coach and realization components to communicate the decisions back to the user.

(CSM), which is responsible for the general structure of the coaching session, *Action Patterns*, which generate the behaviour to realize the plans the CSM decides upon, and finally the *Realizer*, which transfers the behavior to the Render Engine. In the following these components will be explained in more detail.

Figure A.8: Overview of the training environment.

The Coaching Strategy Manager (CSM) is responsible for making decisions about the overall structure of the interaction. It keeps track of the long term goal of teaching the motor skill and selects the next coaching action that maximizes its utility for achieving it. It is currently implemented as a finite state machine making decisions based on an information state. This information state is updated by processing the incoming user input, in this case the output of the Motion Analyzer, and also feedback from the Realizer, which informs the Coaching Strategy Manager on the status of its own behavior. The information state keeps track of how many squats have been performed by the user in the current interaction, the errors made during each squat and, which phase of the squat the user is currently in. The CSM makes a decision each time a new phase of the squat is detected by the Motion Analyzer or it has completed its previous coaching action.

For many actions a decision update rate of once every squat phase or every completed coaching action is too infrequent. To address this problem we introduce

the concept of *Action Patterns*. Action Patterns are dynamic software modules that can be created, activated, and/or stopped at run time. All Action Patterns are their own decision makers within their own expertise that are free to generate behavior fitting the constraints from earlier decision makers, typically the Coaching Strategy Manager (CSM). Each Action Pattern can create its own information flow links to all other parts of our system. For instance, the `Incremental Instruction` pattern directly listens to the output of the Motion Analyzer, bypassing the CSM (see Section A.4.3 for more details). Note that it can still be deactivated by the CSM if it decides on another action. All Action Patterns are available to the Action Pattern Manager. This manager keeps track of which Action Patterns are currently active and has the power to start and stop them if needed. Action Patterns produce behaviors described in the Behavior Markup Language (BML) [Vil+07]. In the current system each coaching act is implemented as its own Action Pattern. *Greeting*, *Introduction*, and *Closing* are lexicon-based Action Patterns where behavior is hard-coded. The different Action Patterns for *Instruction* are explained in more detail in Section A.4.3.

The BML blocks produced by Action Patterns are collected by the Behavior Planner. This Behavior Planner resolves potential conflicts between BML blocks produced by Action Patterns active in parallel, e.g., if two BML blocks want to use a certain body part of the coach at the same time. Currently our system is not rich enough such that many conflicts occur, and we simply delay BML blocks that cause conflicts, however we intend to increase the demand on the behavior planner in future development in this regard. The BML blocks are then realized by the AsapRealizer [VYK14]. It transforms the BML blocks into joint rotations and blend shapes which are passed on to the renderer, resulting in animation of the virtual coach character. The coach's speech is synthesized using the CereVoice Engine Text-to-Speech system (voice Nathan).

A.4.3 *Study*

In our corpus analysis we observed two instruction strategies from the coach which differ in timing: coaches giving their instructions either between squats (sequential instructions) or during squats (incremental instructions) depending on the situation. Sequential instructions between squats allow for more elaboration, while incremental instructions allow for precise timing information. In a user study we explore these instruction types to test whether our architecture can deliver both types successfully and also to gain insight into the user experience of each instruction type both subjectively and in terms of objective learning gain.

Instructions

The virtual coach addresses the error patterns using two types of instruction: *incremental*—the instructions are vocalized *during* the squat—and *sequential*—the instructions are vocalized *between* squats. We now briefly detail the interactive effect of these instructions on users and how they are realized in our architecture.

Incremental Instructions In the incremental instructions setting the virtual coach gives its instructions while the participant is doing the stroke (downward phase) of the squat. The instructions given are short, but occur as soon as the coach becomes aware of the error and has time to produce the instruction. These instructions were generated by an Action Pattern that takes as input the output of the Motion Analyzer at 120 fps—to detect errors—and the BML feedback from the Realizer—to know when a previous instruction is finished. Instructions were pre-planned [Rei+11], meaning that all possible instructions are already submitted to the Realizer in order to pre-process the text-to-speech. They would start playing once an activation signal has been sent to the Realizer.

When no errors occurred the coach would say the following default instructions: "Deeper. Go on. A bit more. A bit more...". It would do so until the error pattern "not deep enough" was no longer present. It would then interrupt this sequence by saying "Stop" as soon as possible, interrupting ongoing instructions. If one of the other two errors are detected it would selected that instruction over one of the default instructions, where "incorrect weight distribution"—instructed by saying "Hips back more"—had priority over "arched neck"—"Watch your neck", a priority observed in our corpus analysis.

After each squat the system would ask for another slow squat by saying "Okay. Give me another slow one." We ask for a slow squat in this configuration to allow the system to express more instructions. A regular squat only provides enough time to say "Deeper" and "Stop".

Sequential Instructions In the sequential instructions configuration the virtual coach gives its instructions after the participant completes the entire squat. These instructions were more verbose than the incremental instructions and were generated by an Action Pattern that takes as input from the Coaching Strategy Manager a summary of the squat, indicating which errors occurred in which phases. If errors occurred in the squat the coach would say between squats: "Okay. Give me one more, but this time (keep your neck straight / push your hips back more / go a bit deeper)[1]" or say: "Perfect. Give me one more like that" when no errors occurred.

Participants and Procedure

Our study had 16 participants (9 female, age $M = 26$). All but one participant had done squats before, 7 reported doing squats at least once a week. After a brief welcome the participants read an explanation of the study and signed a consent form for the data recordings. Then the participant put on the motion capturing suit and tracking markers were attached. When the participants first entered the CAVE a calibration session followed to ensure that all the markers were in place and the tracking was correctly configured. The participants were briefed again about the interactions that would follow. The participants interacted twice with our virtual coaching system, each time with a different instructions configuration. In each interaction the system would ask for a squat 20 times. The coach gives (*incremental* or *sequential*) instructions on

1 All three or only a subset were generated depending on the errors in the squat.

each uneven squat. The even squats are used to measure the performance. Between the two sessions the participants were allowed to take a break as long as they needed. The order of the experimental conditions was balanced between subjects. After the two interactions with the system a questionnaire (see A.4.3) was filled out.

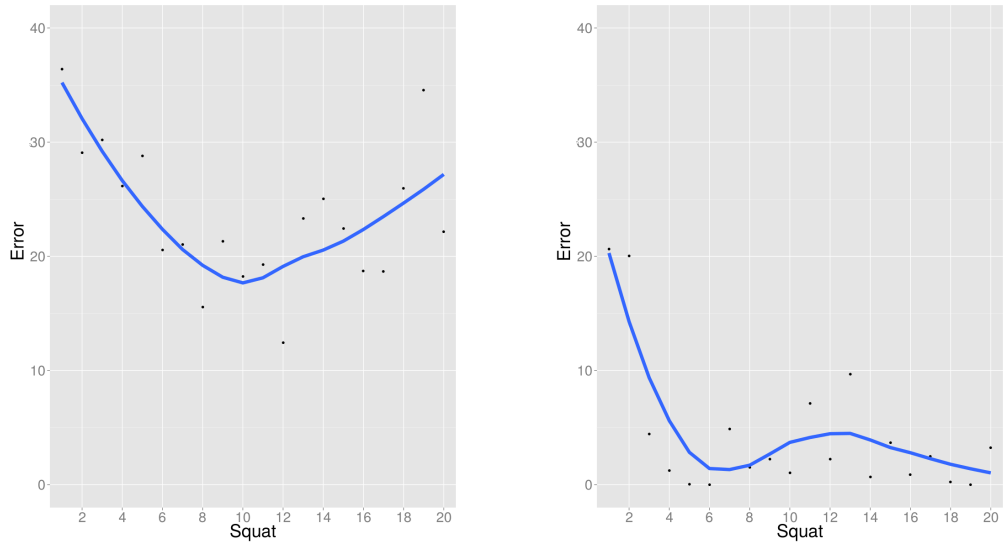
Measures

The questionnaire included items about demographics (gender, age), sport and squat experience (3 items), and 10 items asking to compare the two interactions in terms of several adjectives. These were 7-point Likert scale items with the low end being instruction DURING squats and the high end instruction AFTER squats. A value of 4 indicates no difference. The 10 adjectival properties used were: helpful, responsive, human-like, friendly, polite, efficient, clear, intelligent, tiring, and preferred. This list was inspired by the questionnaire used by Skantze and Hjalmarsson [SH10]. Finally there was an open feedback field where they could share their thoughts and remarks about the study. We also measured performance of the squats with the rule-based motion analysis explained in Section 5.3 applied. For each error pattern, one overall performance value was obtained, where a smaller value indicates a better performance. This was done for the error patterns "incorrect weight distribution", "not deep enough", and "arched neck".

A.4.4 *Results and Discussion*

In order to find out whether the instructions of our coaching system resulted in learning gain, for each error pattern we investigate whether the error was corrected in subsequent squat or severance of the violation decreased.

Only three participants performed the pattern "not deep enough" during squats (two in the *sequential* instructions condition, one during the *incremental* instructions). All of them were able to correct the error in the subsequent squats. Here, the feedback was quite detailed, combined with a precise instruction ("Go a bit deeper" or a clear "Stop") thus, all were able to fix the error. The pattern "incorrect weight distribution" was performed by many of the participants and most were unable to fix it. For the *incremental* instructions, most participants did not leave the coach enough time for expressing the relevant instruction, forcing the coach to generate "Stop" when the desired angle for the pattern "not deep enough" was reached. For the *sequential* instructions, some participants were able to improve their performance. Figure A.10b shows the development of the maximum error value of the squats of one participant who nearly managed to fix the error "incorrect weight distribution" by the end of the session. Some of the participants started reducing the error, but at some point the results became worse again (see Figure A.10a). Some participants complained that they were not informed about getting better, and thus lost motivation to try and improve. For "arched neck" we also observed no improvement. Participants were aware of the error and tried to fix it. However, the provided instruction was not detailed enough. Since no information was provided on whether the neck was over- or under-stretched, participants where unsure on how to fix the error.



(a) Breaking off of error reduction during learning, assumed due to exhaustion and lackingquantitative feedback.

(b) Error rate during the performance of squats for participant 5.

Figure A.10: Exemplary results for error pattern "incorrect weight distribution".

In summary, while promising, the formulation of the instructions should be improved significantly to result in guaranteed learning gain. We need to give more detailed instructions to help users identify their errors and improve their motor program schema. Another issue was that we tried to address three errors simultaneously. Especially in the *incremental* instructions configuration, this led to incomplete, insufficiently precise instructions given the time constraints of the condition (2–3 seconds for an average squat time). See [Kok+15] for the results concerning the questionnaires.

A.4.5 *Conclusion*

In this chapter we have introduced the challenging domain of motor skill acquisition by verbal feedback through the results of an empirical study and have demonstrated that our hard- and software architecture is capable of creating the closed-loop interaction that the domain requires. The system architecture was evaluated by users interacting with two different configurations of the system teaching the motor skill squats. The system gave *incremental* instructions on how to improve during the squat or *sequential* instructions after the squat. The instructions are not yet accurate or clear enough to result in learning gain for the more complex error patterns. For "not deep enough" both the sequential and incremental instructions were effective in correcting the rare occurrences of the error pattern. For "arched neck" and "incorrect weight distribution" the incremental instructions provided timing information on when the errors occurred, however without clear directive instructions on how to correct the error pattern, learning proved difficult. This was also the case in the sequential instructions. Despite the mixed results in terms of performance improvement, from a technical viewpoint the interactions were satisfactory, and we fulfill our desiderata of online

movement analysis, incrementality, and multimodality. The incremental instructions were delivered in a timely manner, such that corrections could be made during skill execution, an ability which was a likely factor in leading participants to perceive the incremental instruction setting as more intelligent (cf. [Kok+15]).

A.5 COACH UTTERANCES AS USED IN CHAPTER 6

This appendix contains the original utterances used by the virtual coach in the experiment described in Chapter 6. The coach speaks German. In this Appendix we provide an English translation in addition to the original German utterances.

The virtual coach introduces himself as follows: *Herzlich willkommen. Ich bin heute dein Trainer. Zusammen werden wir für ein paar Minuten Kniebeugen trainieren. Ich werde auf insgesamt drei Bewegungsmuster achten, die für die Kniebeuge wichtig sind. Fangen wir mit dem ersten Muster an.* — Welcome. I am your coach today. We will train the squat together for a few minutes. I will focus on three movement patterns that are important for the squat.

Then the coach switches to the terminal feedback phase. He describes the error pattern that will be discussed. Together with the explanation replays of prerecorded performances as well as incorrect performances by the trainee are used (see Chapter 6).

Error pattern "incorrect depth" *Die richtige Tiefe ist für Kniebeugen sehr wichtig. Wir trainieren die leichte Kniebeuge, bei der in den Knien der Winkel von 90 Grad nicht erreicht wird. Ich zeige dir einmal, wie tief die Kniebeuge sein muss.* — The correct depth is very important for the squat. We practice the lite squat that does not reach the angle of 90 degrees in the knees. I will show you how deep the squat should go.

Error pattern "incorrect weight distribution" *Wichtig ist, dass die Knie nicht zu sehr belastet werden. Bewegt sich die Hüfte zu weit nach vorne, wird die Belastung auf den Knien zu hoch. Das kann man am besten von der Seite beobachten. Dafür drehe ich den Spiegel, sodass du dich von der Seite sehen kannst. Ich werde dir jetzt eine Bewegung von dir zeigen, bei der die Hüfte und die Knie zu weit nach vorne kommen. Achte dabei besonders auf die Bewegung der Hüfte. Danach zeige ich dir wie die Bewegung im Idealfall aussehen sollte. Im Anschluss mach bitte selbst eine Kniebeuge. Achte dabei besonders darauf, die Hüfte nach hinten zu bewegen und die Knie hinter den Fußspitzen zu lassen.* — It is important that there is not too much strain on the knees. If the hips moves too much to the front, the strain in the knees becomes too much. This can be better observed from the side. To this end, I rotate the mirror, so you can observe yourself from the side. I will now show you a movement that has been performed by yourself, where the hips and the knees move too much to the front. Especially focus your attention on the movement of the hips. Afterwards, I show how the movement should look like ideally. Afterwards, please perform a squat by yourself. Put your attention on moving the hips backwards and leaving the knees behind the toes.

Error pattern "wrong dynamics" *Im Idealfall wird die Bewegung von Armen und Beinen synchron ausgeführt. Achte genau darauf wie ich die Bewegung mache. Arme und Beine bewegen sich synchron.* — Ideally, the movement is performed in synchrony between the arms and the legs. Pay attention to how I perform the movement. Arms and legs move in synchrony.

After the introduction of the error pattern, the coach asks the trainee to perform squats. If the performances becomes better, motivational sentences such as *Ja, das war besser. Noch eine!* (Yes, that was better. One more!) are verbalized. Otherwise, the coach just asks for the next squat with utterances such as *Und die nächste* (And the next one). For the error pattern "incorrect depth" information on the direction of the occurred error is presented with utterances such as *Und noch eine Kniebeuge, aber etwas weniger tief* (And one more squat, however slightly less deep). For all these minor utterances, the coach alters between multiple versions to make the coaching session more natural and less repetitive. The order in which utterances of the same type are selected is the same for all participants.
The following utterances are used to ask for a further squat:

- *Mach bitte noch eine Kniebeuge.* — Please perform one more squat.
- *Noch eine.* — One more.
- *Und noch eine.* — And one more.
- *Und die nächste.* — And the next one.
- *Die nächste.* — The next one.

If the coach decides to compliment the trainee's performance, one of the following utterances is selected:

- *Schon besser! Mach bitte noch eine Kniebeuge.* — Better! Please perform one more squat.
- *Das entwickelt sich in die richtige Richtung. Und noch eine Kniebeuge.* — You are on the right track. And one more squat.
- *Ja, das war besser. Noch eine.* — Yes, that was better. One more.
- *Schön! Und die nächste.* — Nice! And the next one.

The coach can select one of the following utterances to provide information on the direction of the error for the error pattern "incorrect depth":

- *So, mache jetzt bitte eine etwas [weniger tiefe, tiefere] Kniebeuge als eben.* — Please perform one more [deeper, less deeper] squat than the last time.
- *Und noch eine etwas tiefere Kniebeuge.* — And a slightly deeper squat.
- *Und noch eine Kniebeuge, aber etwas weniger tief.* — And one more squat, but slightly less deep.
- *Noch eine etwas [weniger tief, tiefer].* — One more [less deeper, deeper] squat
- *Und die nächste, aber etwas [weniger tief, tiefer].* — And the next one, but slightly [less deep, deeper].
- *Mach bitte noch eine etwas weniger tiefe Kniebeuge.* — Please perform one more less deeper squat.
- *Mach bitte noch eine Kniebeuge, aber etwas tiefer.* — Please perform one more squat, however slightly deeper.

After the terminal feedback phase has been finished, the coach switches to the incremental feedback phase.

Error pattern "incorrect depth" *Jetzt werde ich dich dabei unterstützen während der Kniebeuge die richtige Tiefe zu erreichen. Dafür werden deine Beine so lange eingefärbt, bis du die richtige Tiefe erreicht hast. Solltest du zu tief runtergehen, werden deine Beine erneut eingefärbt. Gleichzeitig werde ich dich mündlich instruieren, damit du die richtige Tiefe erreichst. Damit du die Tiefe besser beobachten kannst drehe ich den Spiegel. Mache die Kniebeuge bitte langsam und bewege dich so lange nach unten bis ich Stop sage!* — Now I will support you during the squat in order to reach the correct depth. To this end, your legs are highlighted until you reach the desired depth. If you go down too deep, your legs are highlighted again. At the same time, I will verbally instruct you to reach the correct depth. To enable you to better observe the depth, I rotate the mirror. Perform the squat slowly and go down until I say Stop!

Error pattern "incorrect weight distribution" *Bei deinen nächsten Bewegungen werde ich, wenn deine Gewichtsverteilung nicht passend ist, die besonders wichtigen Körperteile einfärben. Führe die Bewegung dann zu Ende und versuche dich bei der folgenden Kniebeuge zu verbessern. Los geht's. Mach bitte eine Kniebeuge.* — During the next movement, I am going to highlight the most important parts of the body, whenever your weight distribution is not okay. Then, continue the movement until the end and try to improve during the upcoming squats. Let's go. Please perform a squat.

Error pattern "wrong dynamics" *Im nächsten Schritt erhältst du Feedback während der Kniebeuge. Ein Geist wird im Spiegel gleichzeitig mit dir eine gute Bewegung ausführen. Versuche die Bewegung zusammen mit dem Geist auszuführen und dich an der Gleichzeitigkeit seiner Arm- und Beinbewegungen zu orientieren. Um dir das Betrachten der Gleichzeitigkeit von Armen und Beinen zu vereinfachen, drehe ich den Spiegel, sodass du dich von der Seite betrachten kannst. Mache jetzt bitte eine Kniebeuge.* — In the next step, you receive feedback during the squat. A ghost in the mirror will perform a good movement while you are performing the squat. Try to perform the movement together with the ghost and to orient yourself at the similarity of the ghost's motion of the arms and the legs. To simplify the observation of the synchrony of arms and legs, I rotate the mirror, so you can observe yourself from the side. Please perform a squat now.

In case of the feedback that is faded-out (see Chapter 6), the coach uses utterances such as *Das läuft super. Jetzt einmal ohne Rückmeldung vom System.* (That works out. Now, without feedback from the system). When the feedback is switched on again, the coach says *Und nochmal mit Unterstützung* or *Und nochmal mit Feedback* (And again with support/feedback). The following variations of motivational utterances before switching-off the feedback are used:

- *Jetzt probiere es nochmal ohne Hilfe aus.* — Now try it again without feedback.
- *Das läuft super. Jetzt einmal ohne Rückmeldung vom System.* — That works out nicely. Now without feedback from the system.
- *Gut gemacht. Und noch einmal alleine.* — Nice one. And now alone.
- *Sehr gut. Jetzt wieder alleine.* — Very good. And now alone, again.

- *Du bist auf dem richtigen Weg. Und nochmal ohne Hilfe.* — You are on the right track. And again without help.
- *Super! Und noch einmal alleine.* — Great! And now alone, again.
- *Schön. Jetzt nochmal ohne Hilfe von mir.* — Nice. Now without my help.
- *Fast perfekt. Und noch einmal alleine.* — Nearly perfect. And alone, again.
- Only "incorrect weight distribution": *Schön. Nochmal ohne Einfärbungen.* — Nice. Again without highlights.
- Only "incorrect weight distribution":*Gut! Und nochmal ohne Einfärbungen.* — Nice. And now without highlights.
- Only "incorrect depth": *Das läuft super. Jetzt einmal ohne Rückmeldung von mir.* — That's great. Now without my help.

When enough perfect performances were observed by the coach, he says *Das war perfekt* (That was perfect) and continues with the closing sentence. If this is not the case, but the desired maximum number of repetitions has been reached, the coach directly switches to the closing sentence: *Das war's. Ich hoffe ich konnte dir etwas helfen dein Wissen zur Kniebeuge zu erweitern. Du hast dich gut geschlagen. Bis zum nächsten Mal.* — That was it. I hope I was able to extend your knowledge concerning the squat. You did a great job. See you next time.