A Multimodal System for Real-Time Action Instruction in Motor Skill Learning

Iwan de Kok[1,2,4], Julian Hough[2,4], Felix Hülsmann[1,3,4], Mario Botsch[3,4],
David Schlangen[2,4], Stefan Kopp[1,4]
[1]Social Cognitive Systems Group, [2]Dialogue Systems Group, [3]Graphics & Geometry Group, [4]CITEC
Bielefeld University
idekok@techfak.uni-bielefeld.de

ABSTRACT

We present a multimodal coaching system that supports online motor skill learning. In this domain, closed-loop interaction between the movements of the user and the action instructions by the system is an essential requirement. To achieve this, the actions of the user need to be measured and evaluated and the system must be able to give corrective instructions on the ongoing performance. Timely delivery of these instructions, particularly during execution of the motor skill by the user, is thus of the highest importance. Based on the results of an empirical study on motor skill coaching, we analyze the requirements for an interactive coaching system and present an architecture that combines motion analysis, dialogue management, and virtual human animation in a motion tracking and 3D virtual reality hardware setup. In a preliminary study we demonstrate that the current system is capable of delivering the closed-loop interaction that is required in the motor skill learning domain.

Categories and Subject Descriptors

H.5.2 [**Information Interfaces and Presentation**]: User Interfaces—*Natural Language, Virtual Reality*

General Terms

Algorithms, Human Factors

Keywords

Coaching; motor skill learning; multimodal interaction; virtual reality; virtual human

1. INTRODUCTION

Artificial coaching systems that support users doing sport and exercise activities have been around for little over a decade. Most of these have been motivational in nature: some involving long-term motivation Human Robot Interaction [5] and others an artificial fitness instructor who gives instructions and encouragement within the course of a steady-state exercise like cycling [8, 17] or running [3].

There has also 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 [14] 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 paper we address this unique challenge for motor skill coaching by virtual agents, presenting an intelligent coaching space environment capable of analysing a coachee's movement and a virtual coach character that can generate appropriate instructions as the motor skill is performed. The paper is organized as follows: In Section 2 we describe the behavioural and processing requirements of a virtual coach based on our findings from human-human motor skill coaching interactions. In Section 3 we present the hardware and software architecture of our system. Section 4 describes an experiment we carry out addressing different types of instruction giving. The results of the experiment are presented and discussed in Section 5 and we conclude and present an outlook in Section 6.

2. REAL-TIME COACHING BEHAVIOUR IN MOTOR SKILL LEARNING

In the real-world domain of motor skill learning, human coaches exhibit unique online behaviours which present interesting requirements for a virtual coach. In Figure 2 we show a typical interaction found in a corpus study on a human-human coaching scenario wherein a coach trains a coachee to improve their ability at bodyweight squats (those without a weight or barbell)—full details of the study are reported in [6]. Here, we see a coach instructing on the stance width of the squat's preparation phase. In this interaction, the coach first places himself in parallel with the coachee, introducing the element of the skill they should carry out next in (A). He then demonstrates and describes the skill for the coachee in (B) and uses multimodal instructions (deictic gesture and speech) to communicate the desired stance width in (C). In (D), the coach *repairs* the coachee's stance width and instructs her to *adjust* her foot position with 'a bit closer together' until he is satisfied.

While giving feedback after a coachee skill attempt, or

ICMI 2015, November 9–13, 2015, Seattle, WA, USA.
Copyright is held by the owner/author(s). Publication rights licensed to ACM.
ACM 978-1-4503-3912-4/15/11 ...$15.00.
DOI: http://dx.doi.org/10.1145/2818346.2820746.

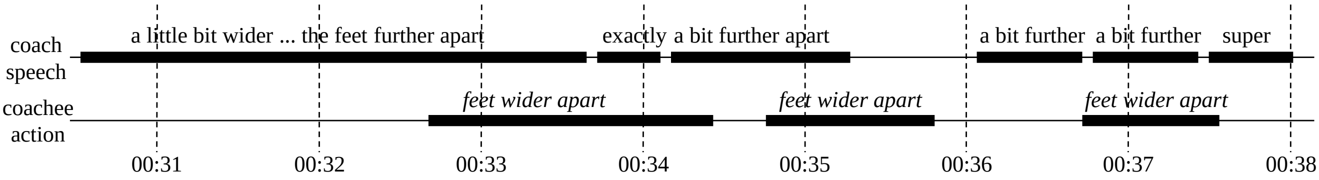

Figure 1: The critical timing of an adjust act by a human coach

even after a series of them, can be an effective strategy for skill improvement—see [14]—we see evidence for the regular need for these kinds of online incremental instructions in our corpus. We tagged each of the coach's utterances for a specialized coaching dialogue act and find that nearly 10% of them consist of these *adjust* acts. These are particularly challenging because of their time-critical nature—see Figure 1 for the temporal representation of a different adjust act where the coach, constantly monitoring the coachee's action, keeps incrementing his contribution with the adjunctive phrase 'a bit further' until the coachee has achieved the desired foot stance. Aside from adjustment cases, the interaction between coach and coachee actions is generally time-critical. We find the mean interval from the end of the instruction to the start of the skill attempt was negative for reactions to instructions for the going down phase of the squat at -0.274s (st.d.=1.204) and even more so for the going up phase at -0.410s (st.d.=0.510), meaning on average coachees were moving well before the end of the utterance. Coachees can take initiative and predict instruction completions easily, just as fine-grained initiative and prediction is common in other situated domains.

We assume the coach is attempting to induce in the coachee a *motor program schema* [13], and evidence as to whether the coachee has learned it or not is observed through their demonstration of the desired outcomes. As a squat has no easily tangible notion of success, particularly for novices, it is crucial that feedback on successful learning be relayed to the coachee to ground the fact it was successful. It is clear that a mixture of offline and online instructions for teaching a motor program schema can be used effectively, and the online instructions provide a particular challenge for a virtual coach.

2.1 Requirements for a virtual motor skill coaching system

Based on the evidence from real coaching interaction, several requirements for a realistic motor skill coaching agent become evident:

- Inherently *multimodal* in both directions—both in the understanding of user's movements and in generation via physical demonstration by the virtual coach. Motor skill acquisition requires detailed monitoring of the user's movements, and also requires demonstration of the skill [4]. The Virtual Agents community takes multimodal generation as a given [10], however systems involving learning gain for a motor skill largely rely on providing disembodied feedback [14].
- Detailed sports and movement science informed *online movement analysis* in the visual processing pipeline. The goal of the system is to reduce the difference between the observed action and the desired action, so the error analysis should be paramaterised by a user's potential.

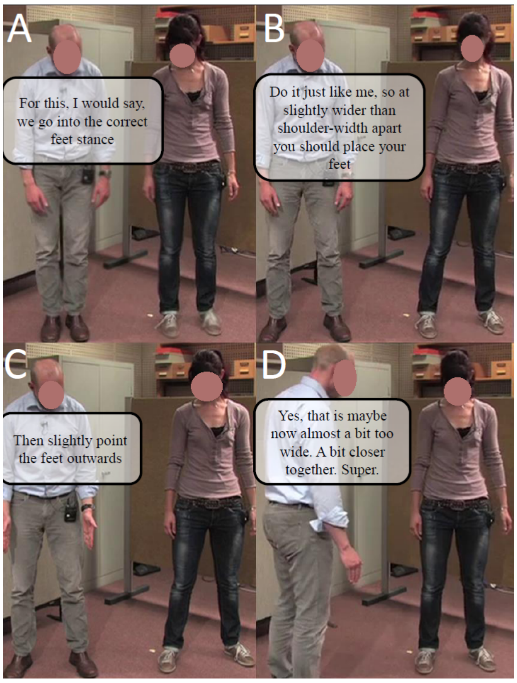
A
For this, I would say, we go into the correct feet stance
B
Do it just like me, so at slightly wider than shoulder-width apart you should place your feet
C
Then slightly point the feet outwards
D
Yes, that is maybe now almost a bit too wide. A bit closer together. Super.

Figure 2: A typical coaching interaction. After the coach introduces a skill phase multimodally in A-C, in D he repairs the coachee's skill attempt and *adjusts* her stance until satisfied.

- Motivational and relevant *feedback generation* to maximise the learning gain and naturalness of the coach.
- Tightly connected *low latency and incremental* processing components from input to output to move towards a *closed-loop system.* Most coaching systems only offer advice and motivation *after* the user's skill attempt, however we require this to be online and during the exercise to simulate a human coach's behaviour as described above. The low latency is in fact vital if the coach is to instruct on and correct a problematic phase of the skill at the relevant time.

It is the last requirement that we focus on in detail in this paper, whilst the others are made possible by our architecture as we will explain. Low latency and incrementality in understanding and generation is key to a motor skill coach, as the closed-loop between perceiving user actions and generating feedback is essential. If there is any latency in an online instruction, this may be interpreted by the coachee as referring to a different part of the movement than intended. Low latency incremental versions of interactive systems have been shown to be preferred to their non-incremental counterparts in simple domains [1, 15], and we use the insight that incremental systems allow fast and responsive behaviour [12, 10] as a point of departure for our architecture.

3. INTELLIGENT COACHING SPACE

We now describe the components that make up our intelligent coaching space. The hardware system consists of a CAVE environment and a motion capture system; the software components are composed of a rendering engine, motion analysis, and dialogue system—see Figure 3 for an overview.

Here the hardware and render engine (Section 3.1), understanding (Section 3.2) and decision making and realization components of the virtual coach character (Section 3.3) are explained in detail. A more extensive description together with a technical evaluation of latencies of the core system can be found in [20].

3.1 CAVE and Graphics Environment

The virtual coaching space is located inside a two-sided CAVE (L-Shape, $3\,m \times 2.3\,m$ for each side) with a resolution of 2100 x 1600 pixels per side.

Our Render Engine runs on a single computer equipped with two NVIDIA Quadro K5000 graphics cards. Rendering runs at approximately 60 fps supporting high quality character rendering, shadows, and post-processing and fulfills our low latency requirements.

In the virtual coaching space the user, equipped with passive 3D goggles, is located inside a virtual fitness room, and following the motivation for enhanced learning in [2], he/she stands in front of a *virtual mirror*. The system maps the user's motion in real time onto an avatar to effect a virtual reflection. The virtual coach is rendered adjacent to the mirror. The virtual world is capable of providing visual feedback on motor skill performance in two ways: users are able to observe their own movements inside the virtual mirror; and the tint of the mirror adapts depending on the observed performance. In our initial setup, feedback was also provided by a summary of the performance as text overlay inside the virtual world.

In the current setup, in line with our motivation of realistic and interactive virtual coaching, information on motor performance is presented using spoken feedback uttered by the virtual coach. Additionally, the mirror flashes for a short time interval to inform users about the successful recognition of a squat.

3.2 Motion Capture and Analysis

The CAVE is equipped with a 10-camera Prime 13W OptiTrack motion capture system. The Motion Tracker uses information obtained from passive markers attached to a motion capture suit to calculate 20 joint angles / positions of the user. This data is passed on to the Render Engine to display the user in the Virtual Mirror.

3.2.1 Motor Performance Analysis

We represent movements as a sequence of feature vectors. This sequence consist of motor actions (e.g. squats), which are connected by arbitrary transition movements. Motor actions are a sequence of Movement Primitives (MPs). A MP describes a homogeneous part of a more complex movement [21]. The real-time analysis system first segments the movement and thus determines which MP the coachee currently performs. Then, it determines the current quality of motor performance. We describe the system in detail in [7].

For each motor action, it is specified which features (e.g. joint angles) are relevant for the definition of the action and its MPs. For the squat, these are mainly the joints of the lower body. Then key-postures for the MPs are defined.

Motion segmentation works via using a state machine: Each motor action and its MPs are represented as states. As soon as a posture similar enough to the first key posture of the first MP of a motor action is detected, the analyzer switches its state. If the next posture is still valid for the current state it remains there. If the posture belongs to the first key posture of the next MP, it switches its state to the second MP. Otherwise, it assumes that the motor action has been aborted and returns to the idle state. The state of the motion analyzer thus reflects the current motor action and the current MP.

After determining the current action and current MP, the quality of the movement has to be assessed. For our application, a detection of performance errors via just comparing the performed MP to an optimal performance is not sufficient: this would lead to single performance values which just describe the overall deviation to the norm. Indeed, we are interested in providing advice on how to correct the movement. For example, to prevent the user from incorrectly distributing the weight, an appropriate 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. Hence, we make use of Prototypical Style Patterns (PSPs) described in [7]: A PSP describes movement styles considered as erroneous and is defined by at least one rule. Each rule returns a quantitative error value for the incoming motion. For each motor action—the squat in our case—a list of PSPs is identified. For our feedback system, we define two types of rules:

Type 1 This rule becomes active 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 as long as a given condition is not satisfied (e.g. the user does not go down deep enough during the whole squat).

The following PSPs for non-optimal performances in the motor skill during squats are detected and later connected to verbal feedback:

Not deep enough (Type 2) 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 coachee reached during the squat.

Too much strain on knees (Type 1) One indicator for this error is that the knees are in front of the toes. This error can also be detected via observing the angle of the shin or the ankle. The PSP returns the largest deviation to the allowed posture the coachee performed during the squat.

Neck not straight (Type 1) The angle of the skull-base and the angle at cervical vertebra 7 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.

All analysis results are transferred to the coaching system (see Section 3.3) in real time at a frequency of around $120\,Hz$, allowing the coach to start planning actions before

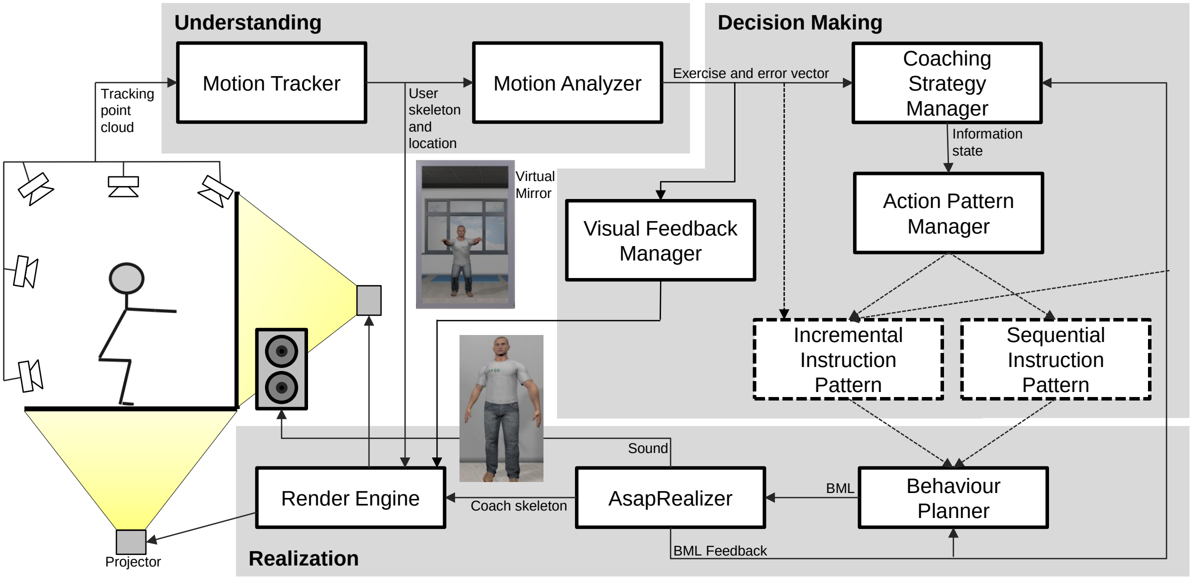

Figure 3: The overall architecture of our intelligent coaching space. On the left the hardware setup is depicted, with wall and floor projection and motion capturing cameras. In our software we have understanding components interpreting the user's actions. Our decision makers comprise both visual feedback (middle part) and a virtual coach and realization components to communicate the decisions back to the user.

they need to be realized in accordance with our incremental processing requirement as will be explained. Our approach to the online motor performance analysis allows a fast detection (*approx.* $1\,ms$ for our list of PSPs) and does not need large amounts of annotated training data.

3.2.2 Pilot Usability Evaluation

To investigate the usability of the core components in a pilot study, we developed a simple coaching application that offered squat training by exploration: The correct movement was shown to a trainee, who also received additional information such as which parts of the movement are of special importance (e.g. "Make sure to keep your neck straight"). The actual training took place in the above described virtual world, but without the virtual coach. Participants were instructed to perform squats in six sets with a break in between. The mirror showed a red tint until the trainee succeeded in performing a correct squat. To depict the detection of a squat, the mirror flashed yellow. If a PSP was been detected, 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. After the performance of a correct squat, the mirror changed its color to green. Each set ended as soon as a squat had been performed correctly according to the given rules or as soon as a given time limit had been reached.

Among others, simulator sickness (using the questionnaire by Kennedy et al. [9]) and presence (using a modified version of the Slater, Usoh, Steed questionnaire (SUS) [16]) were measured. 22 participants took part in the experiment, which took 5–6 minutes each. No increase of simulator sickness was reported. The results for presence were at an intermediate level (M = 3.1, st.d. = 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.

In terms of learning gain, for PSP `not deep enough`, participants were able to reduce the number of necessary squats to produce a correct performance significantly over the sets ($M_{Set1} = 1.8$, $M_{Set2} = 0.6$, $p<0.005$). For all other PSPs, participants did not learn to improve their performance.

These results suggest that the virtual environment is technically sound for measuring error analysis, however the feedback was somewhat static, and not conductive to good interaction or enhancement of skill. In line with our motivation from human coaching set out in Section 2, in our next development phase we introduce our virtual coach character into the coaching space to deliver more natural, human-like instructions during and after the performance of user motor skill attempts. The architecture which enables the virtual coach to deliver these instructions is explained in the following sections.

3.3 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: The *Coaching Strategy Manager (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 behaviour to the Render Engine. In the following sections these components will be explained in more detail.

3.3.1 Coaching Strategy Manager

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 behaviour.

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.[1] 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.

3.3.2 Action Patterns for Behaviour Generation

For many actions a decision update rate of once every squat phase or every completed coaching action is too infrequent to comply with the incremental system requirements expressed in Section 2. 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 behaviour 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 4.1.1 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 behaviours described in the Behaviour Markup Language (BML) [19].

In the current system each coaching act is implemented as its own Action Pattern. *Greeting*, *Introduction*, and *Closing* are lexicon-based Action Patterns where behaviour is hardcoded. The different Action Patterns for *Instruction* are explained in more detail in Section 4.1.

3.3.3 Behaviour Planning and Realization

The BML blocks produced by Action Patterns are collected by the Behaviour Planner. This Behaviour 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 behaviour planner in future development in this regard.

The BML blocks are then realized by the AsapRealizer [18]. 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).

4. EXPERIMENT

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 an experiment 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.

4.1 Instructions

Here we address three squat PSPs: `too much strain on knees`, `not deep enough`, and `neck not straight` errors (see Section 3.2.1) and the virtual coach addresses these errors 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.

4.1.1 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 Hz—to detect errors—and the BML feedback from the Realizer—to know when a previous instruction is finished. Instructions were pre-planned [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 PSP `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 `too much strain on knees`—instructed by saying "Hips back more"—had priority over `neck not straight`—"Watch your neck", a priority observed in our corpus analysis.

Figure 4 shows an example of the incremental instructions. At time point 1 the squat exercise is first detected by the motion analyzer. At that time `not deep enough` is the only error that is detected, so the action pattern vocalizes an instruction to correct this error, in this case "Deeper". When the coach is finished uttering the instruction and a minimum of 200 ms has passed the next instruction is vocalized (time point 2). At time point 3 the error `neck not straight` is detected. Note that this error is not vocalized until time point 4, after the previous instruction is finished and a minimum amount of silence between instructions has passed. In response to the instruction, the user corrected the neck error (see the adjusted neck angle in skeleton 5), so at time point 5 the coach continues encouraging the coachee to go deeper. At time point 6 the desired deepness of the squat is reached and "Stop" is uttered. Note that this interrupts the ongoing adjust instruction "A bit more".

After each squat the system would ask for another slow

[1] The squat is separated into a preparation phase (assuming the starting position), stroke (going down), strokehold (in the lowest position), and retraction phase (coming up).

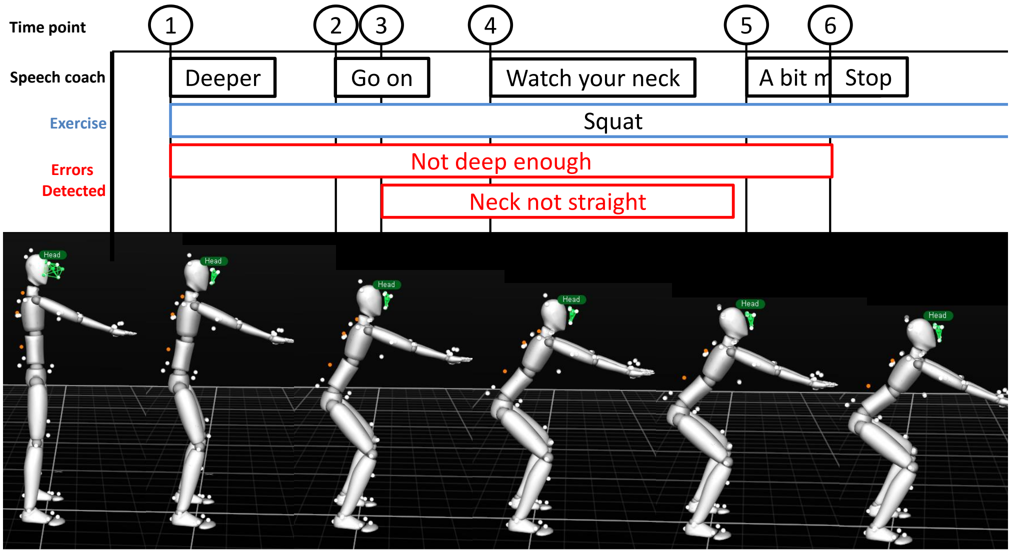

Figure 4: Incremental instructions generated by the virtual coach in response to a user's squat. As soon as a squat is detected (Time point 1) the coach utters instructions to correct the errors detected. Note how the user adjusts the neck angle in response to the instruction "Watch your neck". As soon as the stroke of the squat is completed (Time point 6) the "Stop" instruction interrupts any ongoing instructions.

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".

4.1.2 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)[2]" or say: "Perfect. Give me one more like that" when no errors occurred.

4.2 Participants and Procedure

Our experiment had 16 participants (9 female, 7 male, age 20–45, mean 26). Participants were recruited through university-wide advertisement and were paid 8 Euros or awarded course credit for their time. All but one 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 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 4.3) was filled out. In total the experiment lasted between 30 and 45 minutes, depending on calibration time.

4.3 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 [15]. Finally there was an open feedback field where they could share their thoughts and remarks about the experiment.

We also measured performance of the squats with the motion analysis explained in Section 3.2.1 applied. For each PSP, one overall performance value was obtained, where a smaller value indicates a better performance. This was done for the PSPs `too much strain on knees`, `not deep enough`, and `neck not straight`.

5. RESULTS AND DISCUSSION

5.1 Performance

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

[2] All three or only a subset were generated depending on the errors in the squat.

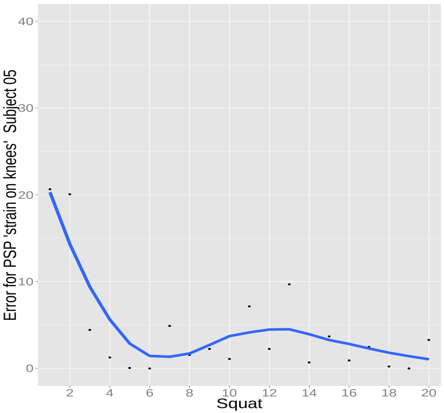

Figure 5: Error rate during the performance of squats for participant 5.

Only three participants performed PSP `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.

`Too much strain on the knees` 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 PSP `not deep enough` was reached. For the *sequential* instructions, some participants were able to improve their performance. Figure 5 shows the development of the maximum error value of the squats of one participant who nearly managed to fix PSP `too much strain on the knees` 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 6). Some participants complained that they were not informed about getting better, and thus lost motivation to try and improve.

For `neck not straight` we also observed no learning gain. Participants were aware of the error and tried to fix it (see Figure 4 for on example in the *incremental* instructions configuration). 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.

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).

5.2 Questionnaire

Table 1 presents the results of the questionnaire. The 7-point Likert scale values were condensed to preference for *incremental* (values 1–3), *no difference* (value 4) and *sequential* (values 5–7) (columns 2–4). The mean and standard deviation values are presented in the final column.

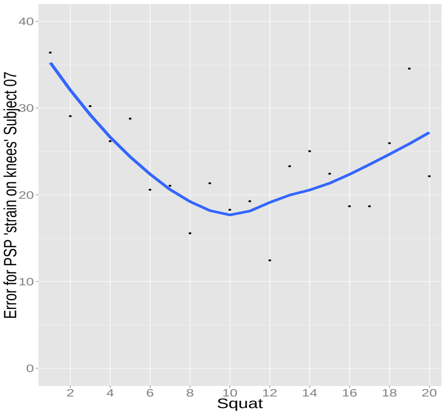

Figure 6: Breaking off of error reduction during learning, assumed due to exhaustion and lacking quantitative feedback.

	Preference for			
	Incr.	**No Diff**	**Seq.**	$\mu(\sigma)$
Helpful	9	3	4	3.38 (1.54)
Responsive	8	3	5	3.31 (1.92)
Humanlike	8	4	4	3.38 (1.70)
Friendly	4	6	6	4.19 (1.62)
Polite	4	6	6	4.38 (1.62)
Efficient	6	4	6	3.75 (1.93)
Clear	8	4	4	3.31 (1.82)
Intelligent	8	7	1	2.94 (1.50)
Tiring	9	3	4	3.43 (2.16)
Preference	7	2	7	3.94 (2.45)

Table 1: Results of the questionnaire.

The *incremental* instructions were found to be significantly more Intelligent ($B(9, 0.5), p < 0.05$). More participants also found that configuration more Helpful, Responsive, Humanlike and Clear. Most also found the *incremental* instructions more Tiring. We attribute this to the fact that the system asked for slower squats. The overall Preference was polarizing, half of them preferred *incremental* instruction, while the other half preferred *sequential* instructions. 9 out of 14 also chose the most extreme value (1 or 7) to express this preference.

6. CONCLUSION AND FUTURE WORK

In this paper we have introduced the challenging domain of motor skill learning through the results of an empirical study and have presented a hard- and software architecture 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 `neck not straight` and `too much` `strain on knees` 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 learning gain, 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 (see neck adjustment between skeleton 4 and 5 in Figure 4), an ability which was a likely factor in leading participants to perceive the incremental instruction setting as more intelligent. In future work we will explore more complex generation strategies, including the effect of our coach demonstrating a skill, to move towards truly interactive multimodal strategies for artificial coaching.

Acknowledgements

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

7. REFERENCES

[1] G. Aist, J. Allen, E. Campana, C. Gomez Gallo, S. Stoness, M. Swift, and M. Tanenhaus. Incremental dialogue system faster than and preferred to its nonincremental counterpart. In *Proceedings of the 29th Annual Conference of the Cognitive Science Society*, 2007.

[2] F. Anderson, T. Grossman, J. Matejka, and G. Fitzmaurice. Youmove: enhancing movement training with an augmented reality mirror. In *Proceedings of the 26th annual ACM symposium on User interface software and technology*, pages 311–320. ACM, 2013.

[3] F. Buttussi, L. Chittaro, and D. Nadalutti. Bringing mobile guides and fitness activities together: a solution based on an embodied virtual trainer. In *MobileHCI*, pages 29–36, 2006.

[4] I. de Kok, J. Hough, C. Frank, D. Schlangen, and S. Kopp. Dialogue structure of coaching sessions. In *Proceedings of the 18th SemDial Workshop on the Semantics and Pragmatics of Dialogue (DialWatt)*, pages 167–169, Edinburgh, September 2014.

[5] J. Fasola and M. J. Mataric. Using socially assistive human–robot interaction to motivate physical exercise for older adults. *Proceedings of the IEEE*, 100(8):2512–2526, 2012.

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

[7] 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 *International Symposium on Computer Science in Sport*, 2015.

[8] W. IJsselsteijn, Y. de Kort, R. Bonants, J. Westerink, and M. de Jager. Virtual Cycling: Effects of immersion and a virtual coach on motivation and presence in a home fitness application. In *Proceedings Virtual Reality Design and Evaluation Workshop*, pages 22–23, 2004.

[9] R. Kennedy, N. Lane, K. Berbaum, and M. Lilienthal. Simulator Sickness Questionnaire: An enhanced Method for Quantifiying Simulator Sickness. *The International Journal of Aviation Psychology*, 3(3):203–220, 1993.

[10] S. Kopp, H. van Welbergen, R. Yaghoubzadeh, and H. Buschmeier. An architecture for fluid real-time conversational agents: integrating incremental output generation and input processing. *Journal on Multimodal User Interfaces*, 8(1):97–108, 2014.

[11] D. Reidsma, K. Truong, H. van Welbergen, D. Neiberg, S. Pammi, I. de Kok, and B. V. Straalen. Continuous Interaction with a Virtual Human. *Journal on Multimodal User Interfaces*, 4(2):97–118, 2011.

[12] D. Schlangen and G. Skantze. A general, abstract model of incremental dialogue processing. *Dialogue and Discourse*, 2(1):83–111, 2011.

[13] R. A. Schmidt. A schema theory of discrete motor skill learning. *Psychological review*, 82(4):225, 1975.

[14] R. Sigrist, G. Rauter, R. Riener, and P. Wolf. Augmented visual, auditory, haptic, and multimodal feedback in motor learning: A review. *Psychonomic bulletin & review*, 20(1):21–53, 2013.

[15] G. Skantze and A. Hjalmarsson. Towards incremental speech generation in dialogue systems. In *Proceedings of the SIGDIAL 2010 Conference*, pages 1–8, Tokyo, Japan, Sept. 2010.

[16] M. Slater, M. Usoh, and A. Steed. Taking steps: the influence of a walking technique on presence in virtual reality. *ACM Transactions on Computer-Human Interaction*, 2(3):201–219, 1995.

[17] L. Sussenbach, N. Riether, S. Schneider, I. Berger, F. Kummert, I. Lutkebohle, and K. Pitsch. A robot as fitness companion: towards an interactive action-based motivation model. In *Robot and Human Interactive Communication, 2014 RO-MAN: The 23rd IEEE International Symposium on*, pages 286–293, 2014.

[18] H. van Welbergen, R. Yaghoubzadeh, and S. Kopp. AsapRealizer 2.0 : The Next Steps in Fluent Behavior Realization for ECAs. In *Intelligent Virtual Agents*, pages 449–462, 2014.

[19] H. Vilhjalmsson, N. Cantelmo, J. Cassell, N. E. Chafai, M. Kipp, S. Kopp, M. Mancini, S. Marsella, A. N. Marshall, C. Pelachaud, Z. Ruttkay, K. R. Thórisson, H. van Welbergen, and R. J. van der Werf. The behavior markup language: Recent developments and challenges. In *Intelligent Virtual Agents*, pages 99–111, 2007.

[20] T. Waltemate, F. Hülsmann, T. Pfeiffer, S. Kopp, and M. Botsch. Realizing a low-latency virtual reality environment for motor learning. Proceedings of ACM Symposium on Virtual Reality Software and Technology. ACM, 2015.

[21] A. Woch and R. Plamondon. Using the framework of the kinematic theory for the definition of a movement primitive. *Motor Control*, 8:547–557, 2004.