MotionMA: Motion Modelling and Analysis by Demonstration

Eduardo Velloso
Lancaster University
Lancaster, United Kingdom
e.velloso@lancaster.ac.uk

Andreas Bulling
Max Planck Institute for Informatics
Saarbrücken, Germany
andreas.bulling@acm.org

Hans Gellersen
Lancaster University
Lancaster, United Kingdom
hwg@comp.lancs.ac.uk

ABSTRACT
Particularly in sports or physical rehabilitation, users have to perform body movements in a specific manner for the exercises to be most effective. It remains a challenge for experts to specify how to perform such movements so that an automated system can analyse further performances of it. In a user study with 10 participants we show that experts' explicit estimates do not correspond to their performances. To address this issue we present *MotionMA*, a system that: (1) automatically extracts a model of movements demonstrated by one user, e.g. a trainer, (2) assesses the performance of other users repeating this movement in real time, and (3) provides real-time feedback on how to improve their performance. We evaluated the system in a second study in which 10 other participants used the system to demonstrate arbitrary movements. Our results demonstrate that MotionMA is able to extract an accurate movement model to spot mistakes and variations in movement execution.

Author Keywords
Activity Assessment; Weight Lifting; Motion Modelling; Real-Time User Feedback; Learning by Demonstration

ACM Classification Keywords
H.5.m Information Interfaces and Presentation (e.g. HCI): Miscellaneous

INTRODUCTION
Physical exercises, such as in sports or physiotherapy, require a specific execution to result in the desired training effect. Hence, experts such as personal trainers and physiotherapists need to communicate this knowledge to novices so that they can perform these movements properly. From informal observations and interviews with trainers we found that the communication between experts and novices can usually be described by a 3-step communication loop (see Figure 1): The expert first demonstrates how the movement should be performed and gives hints on what the novice should focus on.

CHI 2013, April 27–May 2, 2013, Paris, France.

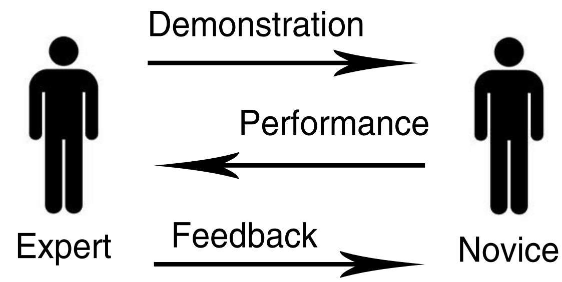

Figure 1: Bidirectional communication loop between expert and novice: The expert demonstrates a movement, which is repeated by the novice and improved according to the expert's feedback.

The novice then repeats the movement under the expert's supervision. Finally, the expert provides feedback on the quality of the execution and how it can be improved.

This communication loop works well if the novice is under direct supervision by the trainer, i.e. if both of them are colocated, but it breaks if personal supervision it not possible, e.g. at home. In these cases, novices so far have to rely on video recordings or on written descriptions and images of the exercises. While training using such descriptions is possible, this approach does not allow for real-time feedback and prevents novices to learn how close their execution is to the desired one and how they can improve it. In addition, written descriptions are typically high-level and qualitative and do not allow novices to quantitatively analyse their performance.

We present *MotionMA* (Motion Modelling and Analysis), a system to encode and communicate movement information and thus enable real-time feedback between spatially separated users. MotionMA allows expert users to specify movements by demonstration. The system then automatically extracts a model of the movement and generates a feedback interface that can be used by other users to repeat the movement and receive feedback on their performance. MotionMA comprises a method for extracting a quantitative model of a movement, e.g. performed by an expert; a method that uses the extracted model to evaluate a given execution, e.g. by a novice; and an approach to automatically generate a feedback interface from the model that helps novices to assess and improve their performance in real time.

RELATED WORK

Remote Coaching

Remote collaboration has been extensively explored in the field of Computer Supported Cooperative Work (CSCW). Similarly, our system supports knowledge transfer between experts and novices separated in space and time. Previous work related to physical activities includes virtual reality systems that put the user and trainer side by side for tai chi learning and dancing [13]. There's also been work on how to convey gestures remotely using voice combined with a projection [12] and a head-mounted camera and a near field display installed on users' helmets [10]. Video recordings of trainers performing exercises have been used extensively in several different mediums, ranging from video tapes to online streaming. More recently trainers are also able to recommend sets of exercises remotely using a wide range of smart phone apps and services like Fitocracy and Fitlink enable online collaboration among users and between users and trainers. These systems, however, can only go as far as routine prescription, without any means for users to assess their performances.

Motion Tracking and Analysis

In the sport sciences, a common method for analysing performance of exercises is to film the athletes and annotate the footage offline using a video digitisation system. An alternative is to use a motion tracking system to extract a skeleton of the athlete automatically. Such systems can be vision-based, usually with passive markers (Vicon, OptiTrack) or IMU (Inertial Measurement Unit)-based (XSens). More recently, depth cameras such as the Microsoft Kinect and the ASUS Xtion enabled consumer-level motion tracking applications, including fitness games that guide players and give feedback on their performance, such as Nike+ Kinect Training, Your Shape: Fitness Evolved and EA Sports Active 2. A drawback of these commercial systems is that their algorithms are hidden from the user and their exercises are pre-programmed, without the possibility to tailor the exercises to the user's needs. Technogym's strength machines can be augmented with the IsoControl hardware that provides feedback on range of motion, speed and resting time as well as repetition counting. No similar product exists for free weight lifting (i.e. using dumbbells and barbells).

Several research projects focused on using inertial measurement units to track and automatically assess physical exercises. For example, Chang et al. were able to tell weight lifting exercises apart using accelerometers on users' bodies but did not analyse the quality of individual executions [3]. Moeller et al. used the sensors in a mobile phone to analyse and provide feedback on exercises performed on a balance board [17]. The Kinect has also been used to improve the quality of movements. Kinerehab analyses users' performances of physical rehabilitation movements and provides feedback, but the exercises only included lifting both arms to the front, to the side and upwards [4]. Martin et al. developed a system that sets out to perform a real-time ergonomic analysis of industrial workers carrying and lifting objects in order to prevent musculoskeletal disorders [16]. Their system was limited to analysing static positions rather than dynamic movements or gestures. Moreover, neither system allows users to specify the movements by demonstration.

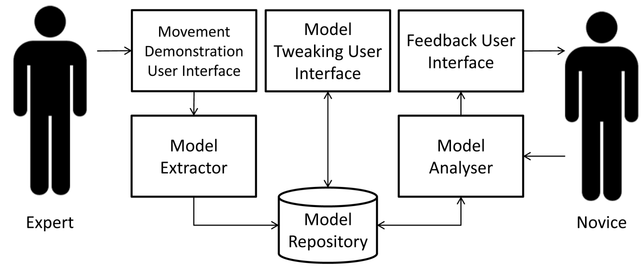

Figure 2: System architecture. Experts specify movements through the demonstration interface. The system extracts a model of the movement and stores it in a repository. The models can be visualised and edited in the tweaking interface. Novice users can then perform the movement, which is then compared with the model to provide feedback accordingly.

Programming by Demonstration

Programming by Demonstration (PbD) has been an active area of research since the early 80's [9]. Instead of hard-cording a system's behaviour, PbD aims to make it possible to program systems by having a user demonstrate to them how they should behave. Such systems aim at making their programming easier for the end-user, who does not need to learn a formal language to specify the system's behaviour [5, 14]. Application areas include robotics [2], software for children [6], text editing, gesture recognition [15], children's toys [19], and context-aware computing [7].

In this work, we use a PbD approach to model human movement and assess its quality of execution. Researchers in robotics extensively studied training of robots with human demonstrating certain activities [11, 20]. Hence, while previous works focused on using PbD for reproduction, prediction and recognition of movement, we use it to specify a model and analyse further performances of the same movement.

THE MOTIONMA SYSTEM

MotionMA is a system to communicate movement information in a way that people can convey a certain movement to someone else who is then able to monitor his own performance and receive feedback in an automated way (Figure 2). It allows users to model movements and assesses further performances of the same movement in comparison to the model.

In developing a system to support the communication of movement information, we tried to emulate the process described on Figure 1. This implies three main functionalities that the system should make available: (1) Allow users to specify movements; (2) Analyse performances of movements; (3) Provide feedback on performances.

All three functionalities are tightly coupled to each other. The feedback the system provides depends on the output of the analysis, which, in turn, strongly depends on the movement's model. At the core of the system is the movement's internal representation, i.e. its model. We wanted our model to be simple enough to be suitable for real-time analysis, but be able to convey enough detail to provide an accurate description of the movement. We also wanted it to be expressive enough to model a wide variety of movements and to be subject-independent, so that the same model can be used by different users. Moreover, we wanted it to allow for analysis approaches that provide relevant information, such as mistake spotting and improvement guidelines.

The standard in Kinesiology is to define motion in terms of the angles of each bones in relation to reference planes (sagittal, frontal and horizontal), which makes it suitable for people with different body measurements. In practice, however, we don't need all three angles, since the direction of a vector can be fully specified in 3D space with only two. Therefore, for each bone we define a spherical coordinate system with the origin at one of the joints and the zenith direction as the vertical at the global coordinate system. Our model is defined as a set of timestamped characteristic points for each bone in each rotation axis. A set of characteristic points is a minimum collection of points with which you could generate a approximation of the time series. Equation 1 shows the formal representation of our model, where the models M_θ and M_φ for each one of the 19 bones b in a skeleton $\boldsymbol{S}$ are the sets of n tuples (t_{θ_i}, θ_i) and m tuples $(t_{\varphi_j}, \varphi_j)$,with the timestamped values of the polar θ and azimuthal φ angles.

$$\forall b \quad \in \quad \boldsymbol{S} \begin{cases} M_\theta(b) = \{(t_{\theta_1}, \theta_1), ..., (t_{\theta_n}, \theta_n)\} \\ M_\varphi(b) = \{(t_{\varphi_1}, \varphi_1), ..., (t_{\varphi_m}, \varphi_m)\} \end{cases} \tag{1}$$

We don't use tuples with both angles at once, so n can be different than m. This allows us to analyse the movement in each axis independently, which makes it easier to provide improvement guidelines. For example, if a problem is detected at the azimuthal axis, we can guide the user to rotate the corresponding bone up or down, whereas if the problem is at the polar axis, the rotation would be in the left or right direction.

Using this model, we can compare two performances by comparing corresponding sets of points. First we need to compute the distances between each point in one set to each point in the other to match the points. By looking at each component of the difference vector, we can then infer whether each characteristic point was at the correct angle (y axis) at the correct time (x axis). Moreover, depending on the direction of this vector and the rotation axis, we can infer improvement guidelines. For example, if the difference between a point in a given performance and the corresponding point in a baseline performance for the azimuthal angle is positive in the x axis and negative in the y axis, we can infer that the user got that point too soon and at an angle too large. The magnitude of each component will tell how far off was he from the specification and whether an improvement guideline should be displayed. Also, we can tell genuine mistakes and normal variability by looking at the distribution of values in the demonstration and checking whether the value in the performance differs significantly from the ones in the demonstration.

ABILITY OF EXPERTS TO DESCRIBE EXERCISES

We first conducted a user study to investigate whether and if yes how well expert users are able to formalise a movement description from their knowledge and experience, i.e. to specify and estimate angles and speed of movements. For this first study we opted to investigate weight lifting exercises given that these exercises are well defined and structured but at the same time allow for plenty of variability and can be performed in the wrong way.

For this study, we chose three common weight lifting exercises, all of which novices were able to perform and that experts were able to instruct others on how to perform. The exercises were the Unilateral Dumbbell Biceps Curl, the Unilateral Dumbbell Lateral Raise and the Unilateral Dumbbell Triceps Overhead Extension. We recruited 10 male expert weight lifters, aged between 22 and 45 years (mean = 31.0, std = 8.38), heights ranging from 1.70 m to 1.89 m and weights ranging from 63.0 kg to 90.8 kg. Participants were recruited using posters distributed around the university campus and on two gym clubs. We made sure that all participants had at least three years of experience with weight lifting, ranging from that up to 25 years.

The experiment took place in a quiet indoor laboratory setting. Participants were asked to wear a belt, armband, gloves and hold a dumbbell on each of which were mounted 5 Xsens inertial measurement units as shown on Figure 3. These sensors were connected by wires in a daisy chain fashion to an Xsens XBus, which streamed the data over Bluetooth to a Linux laptop. Data recording and synchronisation was handled by the Context Recognition Network Toolbox (CRNT)[1]. Participants were asked to stand approximately 2 meters in front of a Kinect camera. A custom application streamed the skeleton data to the CRNT, as well as the labels and the frame number of the RGB camera. As ground truth we recorded videos using a camcorder mounted to the right side of the participants.

Upon arrival in the lab all participants were asked to sign a consent form and to answer a brief questionnaire regarding their previous experience with the specific exercises and weight lifting in general. They were then presented with a written description of the three exercises. Afterwards, they were guided through a structured interview by the experimental assistant, in which they were asked to provide the angles for certain joints of the body for each step of each exercise. Specifically, our goal was to obtain the values of angles that they considered to be ideal for each movement in their own understanding of how they these movements should be performed when exercising or coaching as defined in the written description. We were interested in finding for each joint, whether there was movement during the exercise, the initial and final angle, the acceptable tolerance and how long the movement should take. These are all standard measures in Kinesiology [8]. Since we are focusing on upper body movements, we inquired about arm flexion/extension/hyperextension, arm abduction/adduction, arm lateral/medial rotation, arm horizontal abduction/adduction,

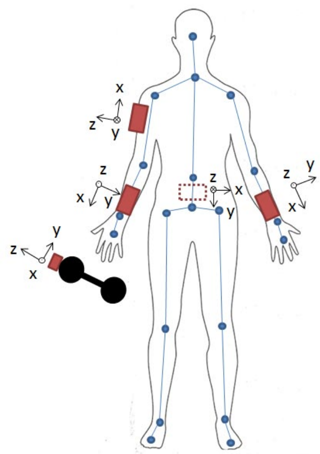

Figure 3: Sensing setup. Blue circles indicate the joints tracked by the Kinect and red rectangles indicate the positioning and orientation of the Xsens sensors.

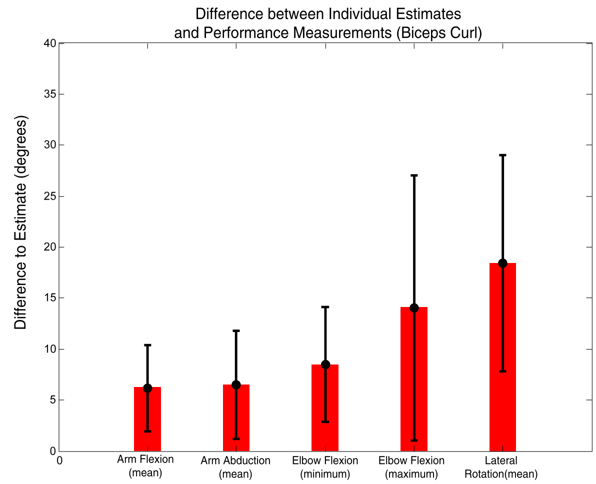

Figure 4: Average difference among all participants of each angle against the value suggested by the corresponding participant for that angle. The error bars represent standard deviations. For angles that are supposed to remain still we use the mean and for those which are supposed to vary we use the maximum and minimum.

elbow flexion/extension, wrist pronation/supination, wrist flexion/extension and ulnar/radial deviation. To make it clear to the interviewee what each movement meant, we provided a graphical diagram of each movement along with the corresponding angular axes. Also, participants were free to ask questions if in doubt about the what each movement meant.

After equipping our sensing system participants were asked to perform 10 repetitions of the Unilateral Dumbbell Biceps Curl, the Unilateral Dumbbell Lateral Raise, and the Unilateral Dumbbell Triceps Extension using three different weights (1.25kg, 3kg and 7kg), totalling 90 repetitions for each participant. The data was manually annotated according to the exercise, weight and participant for post-hoc analysis.

Results
We analysed the data by manually separating each repetition according to the plots and the frames in the video recordings. From each repetition, we extracted different information depending on the exercise. For the Biceps Curl, we extracted the maximum and minimum of the elbow flexion, the mean of the arm flexion, the mean of the arm abduction, the mean of the lateral rotation and the duration of each repetition. For the Lateral Raise, we extracted the maximum and minimum of the arm abduction, the mean of the arm flexion, the mean of the lateral rotation, the mean of the horizontal abduction, the mean of the elbow flexion and the duration of the repetition. Finally, for the Triceps Extension, we extracted the mean of the arm flexion, the mean of the arm abduction, the mean of the lateral rotation, the mean of the horizontal abduction, the minimum and maximum of the elbow flexion and the duration of the repetition. For each participant, we compared the distribution of measurements for each of these values to the answer given in the questionnaire with a One-Sample T-Test. With very few exceptions, the angles suggested in the interview were significantly different ($p < 0.05$) to the angles in the actual performance. This means that participants' estimates of the angles don't reflect their performance. Figure 4 shows the consolidated differences for all participants between each of the measured angles in the 30 performances of the Biceps Curl and the estimate given by the corresponding participant. Differences of the same order of magnitude were found for the other exercises. Figure 5 shows the average difference for all angles in all movements for each participant.

In order to find out whether the performances matched among different participants, we ran a One-Way ANOVA Test on each dataset and again the results were significantly different ($p < 0.05$), even though in a some cases the performance of a few different trainers would match within the group, as seen on a post-hoc Tukey analysis.

Discussion
The results point out that even for experienced weight lifters it is difficult to give accurate estimate of the angles in their own performance, even for movements that are simple and very well defined. Indeed, it has been long known that humans tend to overestimate acute angles and underestimate obtuse ones [18]. This evidences that specifying a movement by natural language and estimating precise angles by observation is difficult.

Moreover, even though they all knew the movements rather well from previous experience and were provided with a written description of their execution, performances varied significantly among different participants. This does not mean, however, that some of them did the exercises incorrectly, only that a wide variety of small possible variations in the performance of a given movement, which is evidence for the ambiguities of written descriptions of movements. Also, these variations occur according to the goals of the weight lifter. For example, while the standard description of a Biceps Curl might state to start the movement with your arm fully extended and

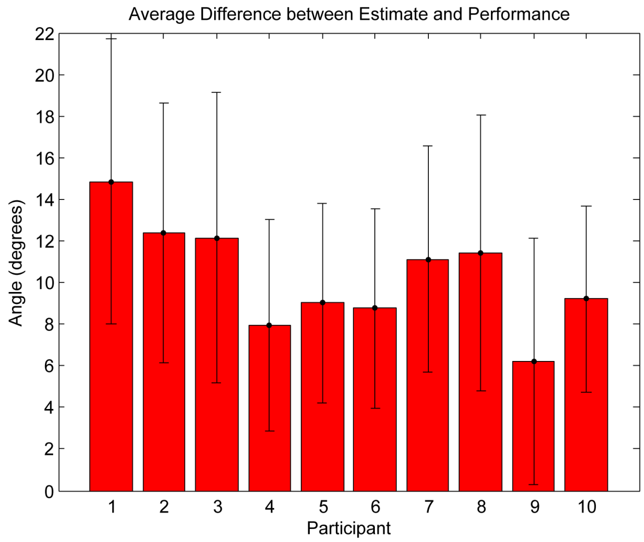

Figure 5: Average difference for each participant between the estimate and the measured angles among all angles. This study shows that users' estimates are in average 10.76 degrees off the measured angle, making it hard to rely on these estimates to evaluate performances.

to finish the movement with your arm fully flexed, a trainer might ask you to flex your arm only halfway to exercise specific muscle fibres. This does not make the second execution incorrect, only different to the first one, which indicates the need to be able to encode these variations into descriptions in a clear way.

If even experienced users have difficulty in explicitly specifying movement angles, we can assume that results would be even worse when considering a wider range of experience and complexity of movement. Moreover, even if they accurately provide this information, inputting these values into the system would prove to be a tiresome and demanding task, which reinforces the case for a more implicit way to obtain this information. Inspired by the observation of how sports trainers and physical rehabilitation professionals communicate movement in real-world scenarios, we developed an approach to extract the movement model by having the user demonstrate it.

MOVEMENT MODELLING BY DEMONSTRATION

Our approach to specifying the movement model draws inspiration from Programming by Demonstration (PbD) works. Our system allows the movement to be specified by allowing the user to demonstrate it, analogous to how professionals do in reality. While this minimises users' effort in inputting information, it arises new challenges, such as detecting the beginning and end of the activity, detecting the characteristic points of the movement and accounting for variations in the movement.

A problem shared by many PbD and Activity Recognition systems is defining the limits of the actual activity being demonstrated, i.e. detecting when it begins and when it ends. In

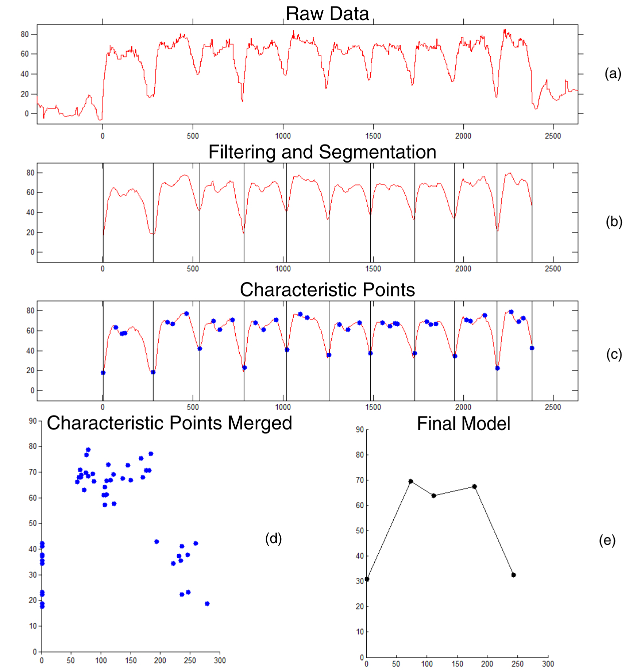

Figure 6: Model extraction from demonstration performance. The raw data (a) is filtered and by counting repetitions, the data is segmented (b). We then find the characteristic points for each repetition (c), merge them (d) and look for the centroids of the data clusters (e).

order to solve this problem, we take as input for our model extraction algorithm a dataset with the performance of 10 repetitions of the movement. Once we can detect each repetition, we can trim the demonstration data at either end.

Detecting the repetitions, however, is not a trivial task. Chang et al. counted repetitions of free-weights exercises by applying a strong low-pass filter and extracted features to train a Hidden Markov Model for individual exercises [3]. Because at demonstration time there is no previous data to be used for training, a machine learning approach is unsuitable for our goal. Instead, we ask the user to perform 10 repetitions of the movement and assume that we will find 10 cycles of a pattern in the data. Because during the movement some bones might be static while others move, we need to pick one axis of a bone to count the repetitions in all of them. We choose the bone and axis by counting peaks and valleys in every axis of every bone and looking for a dataset that gives us 10 repetitions with the largest amplitude of angles. The peak counting algorithm uses a strong low pass filter combined with an autocorrelation algorithm that looks for zero-derivatives on the data, with the mean of values as a threshold to eliminate noise and small variations. Once we have the repetition separation, we automatically have the duration of the movement, which we use in the analysis algorithm to analyse speed.

In order to find the characteristic points of the movement, our algorithm analyses each repetition separately. We do this by using a weaker low pass filter in every dataset and by looking

Figure 7: Demonstration interface. The user can see his skeleton overlaid on top of the colour image recorded by the Kinect. The recording is controlled by voice commands.

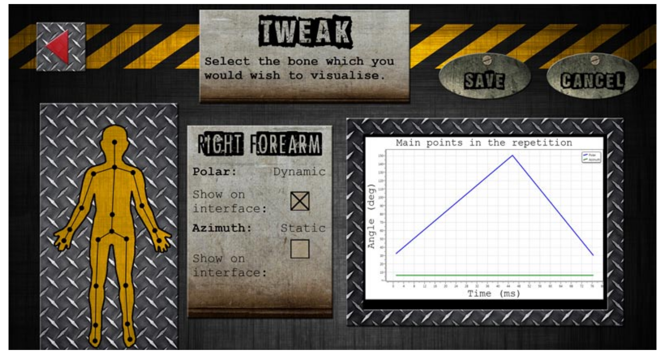

Figure 8: Tweaking interface. In this step, the user can visualise the extracted model for each bone and select what is to be monitored in the performance interface.

at zero-derivatives in the data. This gives us peaks, valleys and inflexion points, which works well to provide a general shape of the curve. At the end of this step, we have 10 sets of points that describe the same movement. We merge these sets to get a consolidated model. We accomplish this by looking for the centroids in the merged data using a k-means clustering algorithm. By using a consolidated model, we account for variations in individual repetitions of the movement. With this set of points at hand, we can make the distinction between static and dynamic axes and tag them accordingly. This information will be used in the analysis to analyse each dataset appropriately. Figure 6 shows the complete process for an example dataset.

The demonstration user interface is shown in Figure 7. The main application was programmed in C# and receives the data from the Kinect sensor directly through its SDK (version 1.5). From the main menu, the user selects 'Demonstrate' and is presented with the demonstration interface. Here he can see his own image in a virtual mirror overlaid with the skeleton tracked by the Kinect sensor. The user then stands in the starting position and issues a voice command to start the recording. He proceeds to perform 10 repetitions of the movement as consistently as possible. When he is finished he issues another voice command to stop the recording. The system then uses the extraction algorithm described in the previous section to generate a motion model. The user can then save or discard the generated model. The demonstration modelling is done in Matlab, using a COM automation interface to transfer data between the two applications.

In order to correct eventual mistakes in the extracted model and to improve its overall accuracy, the user can choose to tweak the model. The 'Tweak' interface is shown in Figure 8. The system displays a skeleton figure on which the user can select the bone he wishes to visualise. When a bone is selected, the system displays the plots of the model for each axis of the bone as well as whether each axis is dynamic or static. In this step, the user also selects which bones he would like to see in the feedback interface. This allows users to tailor the system to specific goals, such as balance or range of motion. He can then save or discard the changes.

In the example of the biceps curl, a user could use this step to make sure the model for both arms are the same as well as adjusting the range of motion in the y-axis. Also, he could make sure that the duration of the flexion and extension during the movement are correct by adjusting the position of the characteristic points in the x-axis.

MODEL ANALYSIS AND FEEDBACK

Once a motion model is extracted from a demonstration, we can use it to analyse further performances of this movement. Since the relevancy of the feedback is tightly coupled to the analysis outputs, we analyse each performance in different levels to allow for different kinds of feedback. Even though further research is necessary into how to actually provide this feedback, we tried to make our analysis approach as comprehensive as possible.

The input for our analysis algorithm is the same as in the demonstration, i.e. the spherical coordinates of each bone in the skeleton as well as the model extracted at the demonstration step. We analyse the data at three separate points in time, giving feedback accordingly: continuously, at the end of a repetition and at the end of a set of repetitions. Continuous feedback is given for rotation axes that are supposed to remain static. This means that the angles at these axes should remain still around a predetermined value, so they are monitored continuously looking for deviations that are flagged as soon as they are found. Dynamic axes are monitored at the end of each repetition. These datasets require the complete analysis, which is described in the introduction. The system buffers the values from the skeleton and runs the analysis when it detects the completion of an instance of the movement. After the user completes all repetitions, the system can analyse the dataset as a whole and provide a more detailed evaluation of the performance due to the reduced attention overload. In this work, we focus on the real-time feedback, i.e. the continuous analysis for static axes and the repetition analysis for dynamic bones.

The architecture of the analysis is comprised of the elements shown in Figure 9. The raw data from the tracking system is converted into spherical coordinates and depending on whether the dataset is static or dynamic it is analysed by a

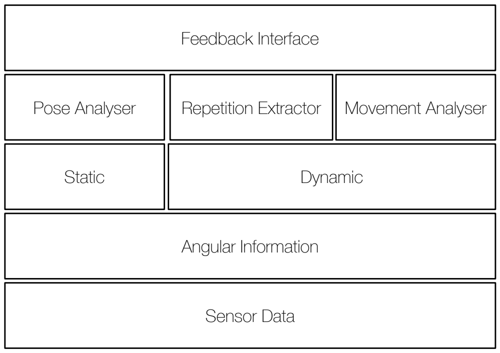

Figure 9: Analysis architecture. Data from the tracking system is converted to spherical coordinates, analysed by the appropriate component and the final analysis is displayed on the interface.

Pose Analyser or a Movement Analyser, respectively. The Movement Analyser is triggered by the Repetition Extractor. The analysis outputs are then showed in the feedback interface.

Much more work is needed to investigate what and when to display to the user. In this first attempt to answer this question we tried out different approaches to communicate movement: video recording of the demonstration; moving dials to monitor the adequate range of motion; traffic lights to monitor static axes and to indicate how to correct them; coloured skeleton, that changes the colour of each bone from green to red depending on its score; speed warning lights, that tells the user to speed up or slow down accordingly; repetition counter, that shows how many instances of the movement have been performed.

Figure 10 shows the user interface for the feedback system we implemented. The systems loads the model, as well as the interface elements selected by the user in the 'Tweak' step. If this step was skipped, the system selects the information to be displayed automatically, based on the variation detected in the model. The user stands in the starting position and issues a voice command to start the analysis. Even though further research into how to convey movement information in a meaningful way is necessary, we took the first steps in that direction by trying out different approaches. Information regarding static axes can be visualised in traffic lights that indicate whether the pose is correct and how to correct it. Dynamic axes can be visualised in dials that move together with the performance. The system also displays warnings when the performance is too fast or too slow and displays the repetition count. Also, the system displays the video feed as recorded in the demonstration session and the skeleton of the user as currently tracked. Each bone in this skeleton is coloured differently according to its score.

SYSTEM EVALUATION

With a working prototype that implements our approach at hand, we could evaluate it in a user study. The goals of this

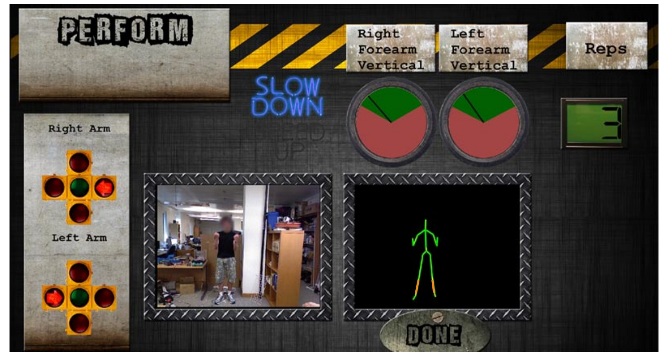

Figure 10: Performance interface. Information regarding static bones is displayed on traffic lights whilst the ranges of motion of dynamic bones are displayed on dials. The user can see the video recording of the demonstration and his own skeleton as tracked by the Kinect with each bone in a different colour depending on its score. The interface also displays the repetition count and warnings when the speed is too fast or too slow.

study were to evaluate the modelling and analysis. For this purposes, we recruited 10 participants aged between 24 and 41, of which one was female. They had different levels of experience with weight lifting ranging from none to experienced. This was to ensure that the modelling system was robust enough to handle different consistencies of performance. The study was carried out in two steps, each one of which was designed to evaluate each of our goals. This section describes our experimental procedure and results.

The study took place in a quiet laboratory setting and each session was carried out with a single participant. The experimental setup consisted of a 27" display with a Kinect sensor mounted on a tripod behind it. When using the system, participants would stand on a previously marked cross on the floor, approximately 2m away from the display and sensor. The system uses the Kinect as a recording device, so no experience with gestural interfaces was needed.

Experimental Procedure

We first evaluated the expressiveness and accuracy of our modelling approach. Each user was asked to think of a controlled and repeatable movement which they would model using our system. Before performing the actual movement, they were asked to provide a detailed written description of it, give it a name and describe five possible mistakes or variations that a person trying to learn the movement would most likely make. They were also asked which bones could be used to count the repetitions of this movement. Then, each participant recorded their movement using the demonstration interface.

We then displayed the extracted model in the "Tweak" interface and went through the model for each axis of each bone together with the participant in a structured interview fashion. We selected each bone, and the corresponding plots of each angle would appear on the screen, as well as the values for each step of the model. The user would then fill in a qualitative questionnaire where he would rank the accuracy of the model for each axis of each bone as well as whether the system marked the bone as static or dynamic correctly. They were also asked to rank the axis and bone that the model extraction algorithm used in order to count repetitions.

When evaluating the analysis, we wanted to find out (1) whether the system could recognise a good performance and (2) whether the system could spot the mistakes foreseen by the participants. We asked them to perform 10 repetitions of the movement as closely as possible to the demonstration performance and to keep an eye on every element in the feedback interface. After this performance, they filled in a questionnaire regarding how accurate the system was at counting repetitions, displaying the correct range of motion in the dials, showing a green light for static bones and indicating the correct speed. Then, users were asked to do 10 repetitions of each mistake and look for elements of the interface that would spot these variations. After each performance, users were asked to fill in a questionnaire regarding how accurately did the system spot each one of them. Finally, each participant filled in a general questionnaire regarding the overall accuracy of the model, the detection of correct and incorrect instances of the movement and the repetition counting. They were also prompted to point out positive and negative aspects of the system.

Results

Participants chose movements of a wide variety, from common strength exercises (Dead Lift, Lateral Raise, Biceps Curl) to some amusingly named body gestures (The Lawnmower, Robot Elevator, Circulation Agent). These included both upper and lower body movements and could all be considered controlled and repeatable.

When inquired about the accuracy of the model for each axis of each bone on a 5 point Likert scale, users rated it were very high (median = 5, mean = 4.7785, std = 0.7181) as shown on Figure 14.

Users were also asked to rate how well the system spotted each of the 5 mistakes they came up with. Figure 11 shows users' ratings of the mistake detection accuracy for all mistakes. Figure 12 shows users' responses regarding how accurate the system was in counting repetitions, in displaying the correct range of motion, in displaying the correct posture in the traffic lights and in displaying the correct speed in the speed signs.

In the end, users gave their overall impressions of the system. Figure 13 shows the answers to regarding how much they agree that: (1) the system was able to extract an accurate model of the demonstrated movement; (2) the system was able to detect a correct performance of the movement; (3) the system was able to count repetitions accurately; (4) the system was able to detect mistakes in the movement. Most responses were positive, with the exception of the one where due to the poor choice of the repetition counting dataset, the analysis did not work as expected.

Comments ranged from being very positive, recommending deployment in real world situations ("Very impressed with the ability to analyse body and and movement repetitions. I would like to see this implemented in gyms") to negative in the case where the system did not work as expected ("Counting repetitions was inaccurate. Feedback was confusing."). Some participants complained about the amount of information on the screen ("Too many things going on to look at!"), the limitations of the tracking system ("It cannot track hand opening/closing.") and limitations in the algorithm ("It didn't pick up when I was going in the wrong direction"). Most responses, however, complimented the interface ("I really like the interface!"), the gamification of the movement communication ("It looks like a cool game!") and how the feedback could be used to correct mistakes ("Feedback was easy to use as basis for correcting movement").

Discussion

In general, participants were very positive about both the modelling and the analysis, as the high scores show. While this does not prove that the extracted model is entirely accurate, it does reflect that the extracted model makes sense in terms of

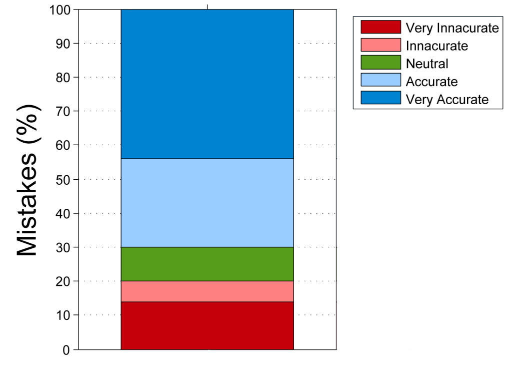

Figure 11: Each user was asked to rate the accuracy of the mistake detection for each of the 5 mistakes they came up with. This chart show that our system accurately detected around 70% of mistakes.

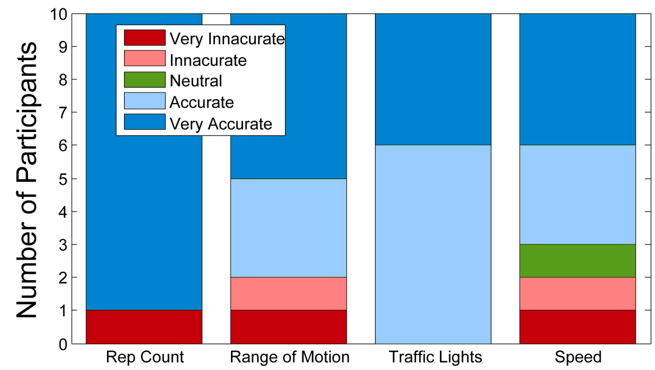

Figure 12: Users' responses regarding how accurately each interface element would indicate a correct execution when performing the movement in the same way as in the demonstration.

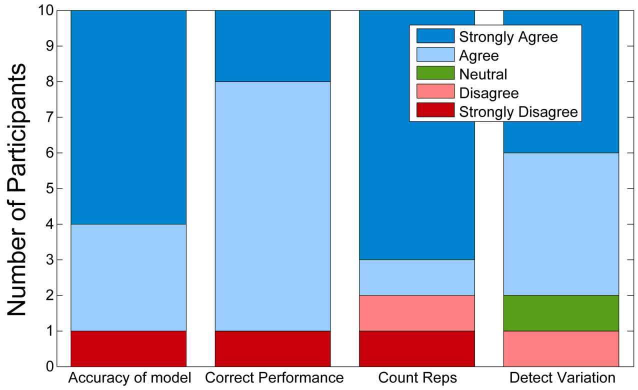

Figure 13: Users' responses regarding their perception of the system.

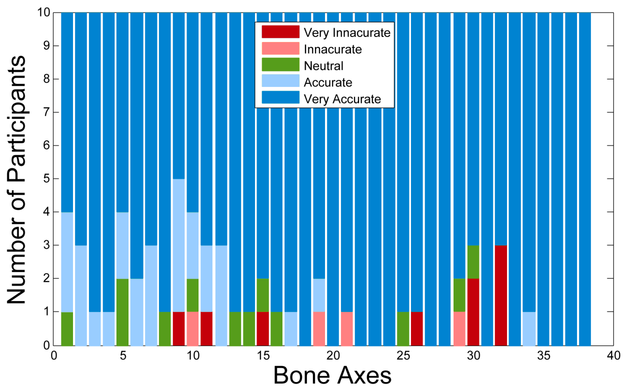

Figure 14: Users' responses regarding the accuracy of the model. Each column represents each of the 38 axes of bone movement (2 for each of the 19 bones).

ranges and general rotation of the bones to the users.

Even though users seemed to agree with the extracted model, factors such as the difficulty in interpreting plots and the previously mentioned difficulty in estimating angles make it necessary for us to obtain more evidence of its accuracy. Among the interface elements, the speed meter was the one that performed the worst. We posit that this happened due to the nature of the movement analysis. The speed is measured after a repetition of the movement is completed, by comparing how long the user took to complete it with the duration of the repetition in the model. Because the displayed speed always regards the previous repetition, we noticed that users would speed up or slow down accordingly but with no effect in the warning signs (which would only change after the repetition was completed), which generated some frustration. With the exception of one movement our repetition counting algorithm counted all 10 repetitions very accurately. The one case where it failed was due to a poor choice of the repetition counting dataset, which by consequence, produced a poor model overall.

Our results indicate that the system is able to extract an accurate model of controlled and repeatable movements and to generate automatically a feedback interface that provides feedback on the execution of the movement. The data in the user study also demonstrated some limitations in our approach. Whilst the analysis algorithm picked most mistakes, there is still plenty of room for improvement. The negative cases were due to the user not being able to recognise the system's feedback, limitations in the tracking system (for example, the Kinect can't tell whether the hand is open or closed, which was an element present in some mistakes), limitations in the algorithm (for example, the analysis algorithm analyses each bone individually, so it would not detect mistakes that had to do with the relation of the movement of one bone to another).

Limitations

Although the Kinect proved to be very accurate in tracking coarse movements, when limbs were pointed directly at the camera, or occluded by the body, the overall tracking was severely penalised. The tracking of hands and feet was not accurate enough to track some of the mistakes participants suggested. Also, due to the limitations of the tracking system , we limited the scope of this work to movements where the user was standing up and facing the camera, but there are a wide variety of movements in which the user is in positions that can't be tracked by the Kinect. Another limitation imposed by the tracking system is that we treat each bone as a vector in the 3D space, without taking into account the rotation of the bone around its length. This means that there is currently no support for detecting rotations such as pronation and supination of the wrist.

Another limitation regards our algorithmic approach. Our current implementation looks at the absolute orientation of the bones. Some of the mistakes participants suggested, however, in terms of the orientation of the bones in relation to one another. In future versions, we hope to add the support to model a hierarchy of bones that can be used to address this problem and suggest more valuable feedback.

Our evaluation has two main limitations. First, we only evaluated the algorithmic approach with a single user. Even though the system was designed for remote collaboration between multiple users, we still need to run another study to evaluate this aspect. The second limitation is the realism of the environment. In the real world, such system would be used by athletes and trainers or patients and physiotherapists. We evaluated the system for its expressiveness by letting the users choose their own movements, but further work is necessary to ensure that the system attends the requirements for specific application domains.

In this work, we explored the general communication process between experts and novices when transmitting movement information as described in Figure 1 and built a system that is a first step in implementing it, focusing on how experts can use it to model movements and configure feedback. Even though it was outside the scope of this paper, there is yet a great amount of work to be done in how effectively this information is actually conveyed to novices and how much the feedback impacts their performances.

Future Work

Since providing an unambiguous and accurate way to communicate motion information is quite an ambitious goal, there are many areas that could be explored in future work. In the lowest levels, there is the need to provide support for more sophisticated tracking approaches. In the algorithmic side, while our approach seemed to extract an accurate model and to perform an accurate analysis, there is plenty of room for improvement, including using more complex curve analysis techniques, such as dynamic time warping. Also, our current approach relies entirely on our algorithm for the extraction of the model. We would like to include support for including rule-based constraints so that users could also encode expert knowledge explicitly in the model.

The next step, however, is to close the loop between the expert and the user, by exploring the best way to provide feedback. Whilst this work has shown that it is possible to encode motion information that can be used to quantitatively analyse performances and to use this analysis to provide feedback, there is a huge range of possibilities of how this feedback may be. Topics of interest in this sense include multi-modal interaction, information architecture, semiotics, etc.

CONCLUSION

Due to the complexity of human motion, conveying movement information is still a difficult problem. In this work, we showed that even people with a lot of experience with very well defined movements can't give an accurate quantitative description of such movements. The need for an intuitive way of codifying motion information inspired us to develop an algorithmic approach to model and to analyse movements. We implemented a working prototype of a system that employs these algorithms to enable users to demonstrate movements and that generates automatically a feedback system for the demonstrated movement which can spot mistakes and guide users in improving their performances.

REFERENCES

1. Bannach, D., Lukowicz, P., and Amft, O. Rapid prototyping of activity recognition applications. *Pervasive Computing, IEEE 7*, 2 (2008), 22 –31.
2. Calinon, S. *Robot programming by demonstration*. CRC, 2009.
3. Chang, K., Chen, M., and Canny, J. Tracking free-weight exercises. In *Proc. of the 9th International Conference on Ubiquitous Computing*, Springer-Verlag (2007), 19–37.
4. Chang, Y., Chen, S., and Huang, J. A kinect-based system for physical rehabilitation: A pilot study for young adults with motor disabilities. *Research in developmental disabilities* (2011).
5. Cypher, A., and Halbert, D. *Watch what I do: programming by demonstration*. 1993.
6. Cypher, A., and Smith, D. Kidsim: end user programming of simulations. In *Proc. of the SIGCHI conference on Human factors in computing systems* (1995), 27–34.
7. Dey, A., Hamid, R., Beckmann, C., Li, I., and Hsu, D. a cappella: programming by demonstration of context-aware applications. In *Proc. of the SIGCHI conference on Human factors in computing systems* (2004), 33–40.
8. Floyd, R., and Thompson, C. *Manual of Structural Kinesiology*. McGraw-Hill Companies,Incorporated, 2011.
9. Halbert, D. *Programming by example*. PhD thesis, University of California, Berkeley, 1984.
10. Huang, W., and Alem, L. Supporting hand gestures in mobile remote collaboration: a usability evaluation. In *Proc. of the 25th BCS Conference on Human-Computer Interaction* (2011), 211–216.
11. Jeong, I.-W., Seo, Y.-H., and Yang, H. S. Effective humanoid motion generation based on programming-by-demonstration method for entertainment robotics. In *Proc. of the 16th International Conference on Virtual Systems and Multimedia* (oct. 2010), 289 –292.
12. Kirk, D., Rodden, T., and Fraser, D. Turn it this way: grounding collaborative action with remote gestures. In *Proc. of the SIGCHI conference on Human factors in computing systems* (2007), 1039–1048.
13. Kurillo, G., Bajcsy, R., Nahrsted, K., and Kreylos, O. Immersive 3d environment for remote collaboration and training of physical activities. In *Proc. of the Virtual Reality Conference* (2008), 269–270.
14. Lieberman, H. *Your wish is my command: Programming by example*. Morgan Kaufmann, 2001.
15. Lu, H., and Li, Y. Gesture coder: A tool for programming multi-touch gestures by demonstration. In *Proc. CHI* (2012).
16. Martin, C., Burkert, D., Choi, K., Wieczorek, N., McGregor, P., Herrmann, R., and Beling, P. A real-time ergonomic monitoring system using the microsoft kinect. In *Proc. of Systems and Information Design Symposium* (2012), 50–55.
17. Moeller, A., Roalter, L., Diewald, S., Kranz, M., Hammerla, N., Olivier, P., and Ploetz, T. Gymskill: A personal trainer for physical exercises. In *Proc. of the International Conference on Pervasive Computing and Communications* (2012).
18. Nundy, S., Lotto, B., Coppola, D., Shimpi, A., and Purves, D. Why are angles misperceived? *Proc. of the National Academy of Sciences 97*, 10 (2000), 5592.
19. Raffle, H. Topobo: programming by example to create complex behaviors. In *Proc. of the 9th International Conference of the Learning Sciences-Volume 2*, International Society of the Learning Sciences (2010), 126–127.
20. Skoglund, A., Iliev, B., and Palm, R. Programming-by-demonstration of reaching motions–a next-state-planner approach. *Robotics and Autonomous Systems 58*, 5 (2010), 607–621.