Reactive Video: Adaptive Video Playback Based on User Motion for Supporting Physical Activity

Christopher Clarke[1], Doga Cavdir[2], Patrick Chiu[3], Laurent Denoue[3], Don Kimber[3]
[1]Lancaster University - Lancaster, UK, chris.clarke@lancaster.ac.uk
[2]Stanford University - Stanford, CA, USA, cavdir@ccrma.stanford.edu
[3]FXPAL - Palo Alto, CA, USA, {chiu, denoue, kimber}@fxpal.com

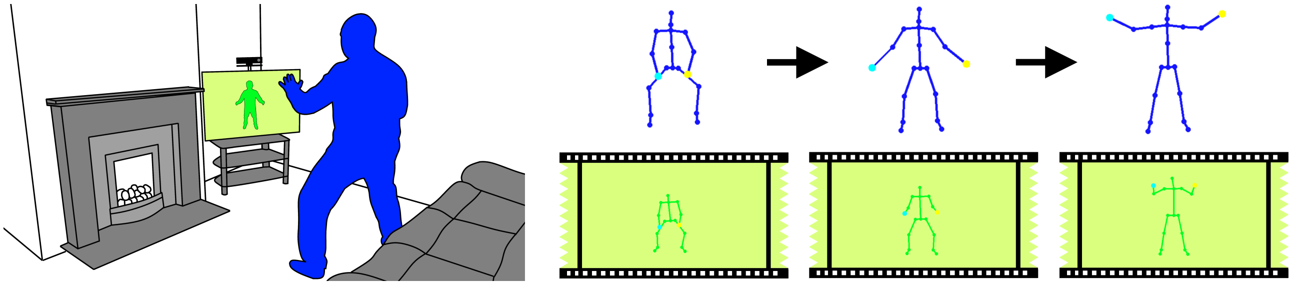

Figure 1. Reactive Video is a vision-based system which supports users when learning or practising physical activities using videos. At the core of the system is adaptive video playback, which is used to control the video by tracking the user's movement and adapting the play time accordingly.

ABSTRACT

Videos are a convenient platform to begin, maintain, or improve a fitness program or physical activity. Traditional video systems allow users to manipulate videos through specific user interface actions such as button clicks or mouse drags, but have no model of what the user is doing and are unable to adapt in useful ways. We present adaptive video playback, which seamlessly synchronises video playback with the user's movements, building upon the principle of direct manipulation video navigation. We implement adaptive video playback in Reactive Video, a vision-based system which supports users learning or practising a physical skill. The use of pre-existing videos removes the need to create bespoke content or specially authored videos, and the system can provide real-time guidance and feedback to better support users when learning new movements. Adaptive video playback using a discrete Bayes and particle filter are evaluated on a data set collected of participants performing tai chi and radio exercises. Results show that both approaches can accurately adapt to the user's movements, however reversing playback can be problematic.

Author Keywords

Physical activity; full body; direct manipulation; probabilistic.

CCS Concepts

•**Human-centered computing** → **Gestural input;**

INTRODUCTION

Technology enables users to readily practice and learn physical activities such as fitness programs, dancing, martial arts, or sports from the comfort of home. Since the 1980s, videos have helped users stay active by offering a cheaper, more convenient alternative to gym memberships, personal trainers, and expert coaching. In the past decade, game consoles have emerged as popular alternatives for fitness and dance programmes leveraging tracking technologies such as the Microsoft Kinect [32] or Wii Fit [36]. These provide entertaining, gamified experiences which can adapt to the user, however they require bespoke content to be created which limits the choice of content offered. In contrast, the proliferation of online video and streaming services provides access to a wide variety of physical activities offered by a diverse set of instructors.

Despite the abundance of videos available, video players offer limited support to users when performing physical activities. Unlike in-person instruction, a video can not provide feedback or adapt the pace and intensity of the physical movement or exercise to an individual user. For example, performing movements in slow motion is beneficial for learning by providing opportunities for self-analysis and developing timing of various components. It has also been shown to result in less physical strain for inexperienced users [8], and can increase strength compared with regular exercise [57]. For some users, keeping pace may be difficult due to physical constraints or because they are unfamiliar with the movements. Standard controls may be available to pause, replay, scrub, or set the play speed to a pre-defined value, however interaction is limited and often involves reaching for the remote, or the device itself, which breaks flow and immersion. Voice input offers a compelling hands-free alternative, however in isolation can only be used for issuance of explicit commands [7].

In this work, we present *adaptive video playback* which infers the pace at which a user is performing a physical activity and dynamically adjusts the play time so that the video reflects their movement. We implement this concept in *Reactive Video*, an interactive system which uses adaptive video playback to provide a more immersive experience for users when using videos for physical activities, and to better assist in learning new movements, see Figure 1. Adapting the playback of the video to the user is not straightforward because a user's internal intentions may not be accurately reflected in their movements, and is compounded further by sensing inaccuracies. Due to this, we investigate two probabilistic tracking approaches, a discrete Bayes and particle filter, which account for the uncertainties in the process. Using tai chi and radio exercises with varied tempos, we demonstrate how they can successfully adapt the playback based on user intention.

Reactive Video is designed to work with videos, and does not require bespoke content to be created. This also removes the need for complex or laborious authoring of content, and opens up the opportunity of using the wide variety of existing video content with the system. We demonstrate how pre-existing videos can be post-processed using state-of-the-art skeletal trackers to extract the instructor's poses for use with the system. In addition to adaptive video playback, the system can leverage the motion of the user and instructor to augment the video with configurable graphical overlays to provide real-time guidance and feedback to assist in learning new movements. Recording and logging capabilities also allow a user or trainer to view an activity with detailed feedback on both the user's pose error and tempo of the movements relative to the instructor.

We develop the contributions of this work as follows. First, we describe the design and implementation of Reactive Video and elaborate the different modes of the system, showing consideration for how users can initiate interaction or decouple from the adaptive playback, and how to offer guidance and feedback. We then discuss the challenges and requirements for adaptive video playback, and describe two probabilistic approaches used to overcome these. A data collection of participants performing different physical exercises is presented, which demonstrates the efficacy of the proposed approaches for accurately tracking user intentions. Finally, we discuss the causality dilemma which arises with Reactive Video, and directions for future work.

RELATED WORK

The concept of adapting playback to user movement was first introduced by Watanabe et al. with Synchronized Video [56]. We extend this by discussing the requirements of adaptive video playback in the context of learning new movements, explore probabilistic approaches which do not require prior training for individual videos, and demonstrate how pre-existing videos can be post-processed using state-of-the-art computer vision techniques without the need for fiducial markers. In addition to adaptive playback, Reactive Video augments videos with feedback and guidance, and we consider how users can activate or decouple from the adaptive playback element for use in real-world applications. Our work also builds upon previous research on technological tools for assisting users to learn or practice a physical activity, direct manipulation video navigation techniques, and motion correlation.

Technology-Based Exercise Tools

A range of work has explored how movements can be defined by an expert with a commercial depth sensor, and appropriate feedback presented to help correct user movements in real-time for training [53], learning a new skill [1], and rehabilitation [11, 46]. Anderson et al. discuss design guidelines for movement training systems based on previous literature which include [1]: reducing visual complexity [51], motivating the user [3], adapting guidance as the user learns [42], providing summary feedback [59], and allowing users to progress at their own pace [60]. They also propose to leverage domain knowledge in the authoring system, however our approach contrasts systems designed for specific physical activities, examples of which include ballet [49], tai chi [19], and weight-lifting [29].

Feedforward techniques are used to guide users and illustrate future movements. A seminal example of this is OctoPocus, which introduced the concept of dynamic guides which show motions paths of possible gestures [2]. These have inspired similar trajectory-based guidance for physical activities, such as cue ribbons in YouMove [1], or wedge visualisations in Physio@Home [46]. In addition, feedback can be used to indicate how well the user is performing a movement and can help to correct for errors, enhancing motor learning [43]. Different approaches have been studied for supporting physical activities by providing feedback and guidance, including multi-camera perspectives [46], augmented mirrors [1], light projectors [45], low-latency multimodal feedback [12], augmented reality headsets [9], and superimposed trainers in virtual reality [61]. These provide compelling solutions but often involve complex and/or expensive hardware setups and require bespoke content to be created or authored for the system. We contrast this in our approach by requiring basic hardware and providing the capability to use pre-existing videos without additional authoring.

Although the aforementioned approaches provide guidance and feedback for the user to correct their pose or tempo, none of them adapt the underlying content to the user in real-time. In the area of physical rehabilitation, virtual reality has been investigated as a means to provide self-adapting technologies to better support therapists and patients with different requirements [52]. Kallmann et al. adapt the pace of a virtual character performing physical movements in real-time by parameterising modifiable properties of a movement, such as the speed, amplitude, and duration of pauses [22]. In this approach, the properties of the exercise are adapted based on how well the user completed the previous repetition of the movement. Our proposed approach to adapting the content to the user is more generalizable, as not all motions can be parameterised, and doesn't rely on detecting repetitions which can be non-trivial [33].

Video Navigation

The proposed concept of adaptive video playback is inspired by direct manipulation video navigation (DMVN) techniques. Traditional video navigation involves manipulating a seeker bar which linearly controls the play time of the video. DMVN applies the direct manipulation metaphor to enable users to drag objects in the scene along their motion trajectories to affect playback. CyberCoaster first introduced the concept of DMVN [40], which later gained traction with automated techniques for motion extraction using object recognition and tracking, and optical flow-based methods [26, 13, 23, 16]. DMVN has also been adapted for touch-based mobile interaction [24, 35], navigation in 3D [34], manipulation of data visualisations [28], and most recently for navigation in spatial recordings [31]. In Empatheater, users must interact with the system at pre-defined events to continue playing the video [30]. This used motion gestures and jumping, and can be seen as a precursor to DMVN using full-body gestures. In this work we demonstrate the first full body movement implementation of DMVN, which focuses on adaptive playback as opposed to seeking to specific play times.

Uncertainty and Motion Correlation

Our work also builds upon the notion of continuous uncertain interaction [58] and synchrony, which has been used for interaction in HCI in the form of motion correlation [54]. Continuous uncertain interaction takes into account ambiguity due to sensing limitations and poor modelling of user behaviour [39, 58]. We adopt this approach of maintaining uncertainty in the interface when estimating play times. Motion correlation uses spatiotemporal matching of the user's input with the device's output for interaction, and has been used for selection [54], calibration [38], addressing gesture systems [15], and for bootstrapping spatial interaction with touchless gestures [10]. We use this as inspiration for seamlessly triggering the adaptive playback component.

REACTIVE VIDEO SYSTEM

The novel aspect of Reactive Video is the ability to adapt the video's playback based on the user's movements, such that the video appears to mirror them. *Control points* are joints in the body which are used to assess where in the video the system believes the user is, and adapt the playback accordingly. By default, we use both hands, however the control points can vary depending on the type of exercise being performed (e.g. one hand, hips and legs etc.). In the remainder of this section we discuss the design of the system based on the novel adaptive playback component (i.e. in learn or immerse mode).

User Modes

We developed four user modes to help better support users when performing a physical exercise using a video:

- *Watch:* The user passively watches to familiarise themselves with the movement, and can control the video using traditional or DMVN techniques using a mouse or touch input, similar to previous work [26, 13, 23]. For DMVN, dragging is limited to individual joints of the instructor in the video.
- *Imitate:* The user watches and tries to copy the movement. In this mode, the system records the user's movement and provides feedback afterwards on pose errors to help the user identify places in which they may have struggled, and areas on which to focus.
- *Learn:* The user is trying to learn a physical exercise. The system provides adaptive playback control, real-time feedforward to dynamically guide the users, and real-time feedback to help correct for pose errors.
- *Immerse:* The user is very familiar with the movement and performing it. The video provides adaptive playback control, but no feedback or feedforward mechanisms.

The first two modes correspond to how users may traditionally interact with an exercise video, with additional feedback provided in the *imitate* mode which can be used for assessment. The *learn* and *immerse* modes utilise the novel adaptive video playback proposed in this paper. Guidance for users can be helpful in the initial learning stages, however the *guidance hypothesis* states that a user may become overly reliant on the guidance given during the training phase, which in turn hinders learning of the underlying process [41]. When using the system to learn an exercise, we envisage that the user progresses through the user modes as they become more skilled, and thus reduce the guidance. The utility of such a system can be demonstrated with two scenarios.

Scenario 1: An elderly user would like to partake in their daily physical exercises for which they use a video for guidance. Due to their age they are unable to keep pace with the instructor in the video and take regular breaks. The adaptive nature of the playback using *immerse* mode means the video stays synchronised with their movements, and they feel less pressure to keep pace. Similarly, when they pause during a movement to take a break, the adaptive playback recognises this and pauses until they are ready to resume.

Scenario 2: A user recovering from a physical injury has been given a set of exercise videos to help their rehabilitation, however due to pain they find it hard to keep pace. As they are unfamiliar with the exercise, they can use the *watch* mode to familiarise themselves with the exercises. The *learn* mode supports the user by providing real-time guidance in the form of motion trails demonstrating how to perform the movement, whilst the adaptive playback allows them to go at their own pace and ensures the guidance relates to the most relevant part of the video.

Instructor Pose Extraction

Reactive Video requires a video containing the poses of the instructor for adaptive playback, and feedback and feedforward mechanisms. We developed two different approaches for this:

Bespoke recordings. The first method we implemented was to record physical exercises using a depth sensor capable of skeletal pose tracking, e.g. Microsoft Kinect, similar to previous work which used expert demonstrators to develop authoring tools for their systems [53, 46, 1].

Pre-existing videos. The second method utilises Reactive Video's ability to work with existing videos, without placing extra burden on content creators. We developed a program to post-process any video and extract instructor poses using OpenPose [5]. Post-processing existing videos also has the advantage of extracting poses from videos with higher frame rates and/or resolutions than are possible with most commercial depth sensors. However, most current skeletal pose trackers extract the pose in 2D, and thus downstream algorithms must cope with 2D data. The ability to extract 3D poses from 2D videos is an ongoing research topic with promising results for future implementations (e.g. [37]).

With either approach, there may be scope for additional authoring of the videos. In the presence of multiple people in a video, the instructor is by default the one closest to the centre of the screen. This may not always be the case, and users or creators may wish to define the instructor when multiple people are present. Trainers collaborating with a user may also wish to define the default control points to be used for a physical activity, annotate specific movements for later inspection, label repetitions, or select points at which adaptive video playback should be disabled. However, it is important to note additional authoring beyond pose extraction is not essential because the user has full control over the adaptive playback, and can switch it on and off as and when they require.

Intent to Interact

One of the main usability concerns of body movement-based sensing systems is how to address the system to initiate interaction [4]. We developed two approaches for activating the adaptive playback component in-situ:

Activation gestures are those which one would not expect the user to perform accidentally, and are thus reserved for starting an interaction [27, 20, 55]. These can be performed using different modalities, such as a specific body pose or voice command. Activation gestures can be invoked from anywhere in the scene, however they must be designed to reduce the risk of accidental activation. We found a simple body pose of hands together above the head was sufficient to activate the system reliably and afforded the user with explicit control.

Motion correlation involves signalling intent to interact by mimicking movement displayed on a screen [15, 10]. Traditionally the on-screen movement is synthetically generated for selection, however we utilise the instructor's inherent movement performing the physical exercise, and look for a correlation between the user and instructor's movements. Motion correlation results in a seamless transition for the user, and from a system perspective ensures they are following the movement at the beginning of the interaction.

Adaptive Playback Decoupling

A user may wish to disengage the adaptive playback for a temporary period, e.g. during rest between exercises in a fitness class, or because they have finished using it. The nature of the decoupling would ideally result in different behaviours from the system. In the case of resting between exercises, the video should continue to play at unit speed until the next exercise is ready, whereas in the case of attending to something else (e.g. a knock at the door) the system should pause and wait for the user to resume. One way of approaching this is to use simple voice commands, such as "pause" or "disconnect", or to incorporate authored elements of the video, e.g. parts in which adaptive playback is disengaged. We also consider how body movement could be used to infer intent to decouple for generic videos.

The pose error between the user and instructor can be used to assess whether the system should be decoupled (i.e. have they stopped following the exercise), based on a threshold of euclidean distance between the control points over a given time window. This is set generously to allow for errors in the following of the movements themselves. By default, the system resumes playback at unit play speed, unless the user moves out of a specific zone (in our case the sensor's field of view) in which case it pauses the video. It is also important to note that smaller, in-situ pauses are captured by the adaptive video playback component, such as when a user holds a pose temporarily to catch their breath (as they do not break the disconnect threshold).

Feedforward and Feedback Mechanisms

The learning mode uses feedforward and feedback mechanisms to help guide the user when performing a physical activity. Feedforward mechanisms relate to where and how the user should position themselves and can be used to help guide them through movements they may be unfamiliar with, or through complex sequences of motion. In contrast, feedback mechanisms can be used to indicate to the user how well they are performing the movements according to the video, and can be either real-time or summary. Feedback may be provided on both spatial (e.g. pose error) or temporal (e.g. play speed) aspects of the video. These elements are generalisable across videos as they focus on movements viewed as trajectories, and can be tailored by the user (e.g. guidance with no feedback).

Visual Feedback

The movement guide (Figure 2) shows movement trails, user skeleton, and control points. These are designed to be configurable such that the user can adapt the guidance to suit their needs, allowing for users to reduce visual complexity as they see fit [42, 51]. The user's skeleton is overlaid in real-time onto the instructor's to provide feedback on their alignment, and to help guide the user's movement relative to the instructors. The type of alignment used will affect the nature of the feedback. For example, one can align individual body parts to the instructor's skeleton to isolate and provide feedback on individual limbs. Skeleton alignment is also used for playback estimation, which we discuss in the next section, however the feedback provided to the user is independent of this.

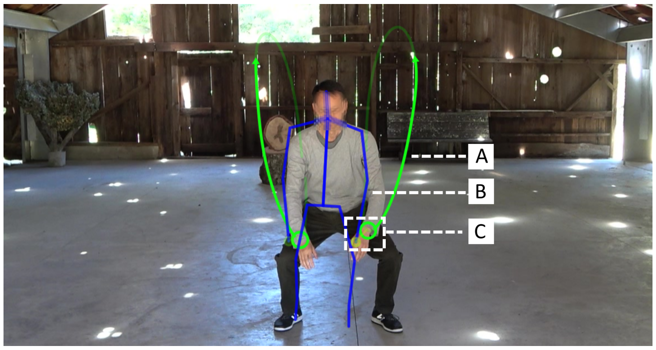

Figure 2. Visual feedback of Reactive Video showing (a) movement trails to guide the user which adapt to the tempo of the movement, (b) overlay of the user's skeleton, and (c) control points illustrating pose errors through colour change.

Movement trails illustrate both the forward and backward motion paths of the instructor in the video, and provides a visual cue of the upcoming movements. By default, these are configured to show movement two seconds in the future and two seconds in the past. The arc length of the movement trails indicates to the user the speed at which the movement takes place (i.e. larger the trail the quicker the movement). The movement trails originate from the control points, which also illustrate to the user which parts of the movement they should focus on. We use a colour gradient to indicate how close the user's control point is to the instructor's.

Audio Feedback

Changing the play speed of the video playback changes the audio too, which can be mitigated by techniques, such as *time stretching*, which change the speed or duration of an audio signal without affecting its pitch [14]. Auditory feedback has also been shown to be beneficial for motor learning tasks (see [43] for a review). We developed a complementary auditory feedback mode which could be played instead of the video's original audio track. This provides musical feedback to help the user correct pose alignment and speed of movement using qualities of the music's pitch and tempo respectively.

Summary Feedback

A user may record a session for later review, or to share with an expert for feedback in accordance with the *demonstrate, perform, feedback* cycle discussed in previous research [53, 1]. Upon completion of an exercise in recording mode, the post-video guide (Figure 3) provides feedback using a line graph which plots the pose error and play time of the video against time. Scrubbing along the graph acts as a slider to control both the user video and instructor video so that the user and/or expert can see at which part of the exercise is being attended to, and watch the user and expert movements side-by-side.

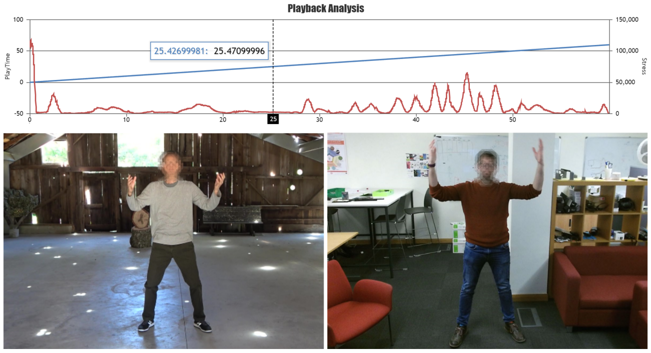

Figure 3. Summary feedback showing a line graph with the play time (blue) and pose errors (red). The instructor (left) and user (right) videos can be shown side-by-side and are controlled by scrubbing along the line graph (user and instructor zoomed in for illustration purposes).

IMPLEMENTATION

We built Reactive Video using a NodeJS server and Javascript client. One requirement was for a CPU-only implementation that did not require expensive GPU-based acceleration. For development we used the Microsoft Kinect v2 as a sensing device, and also implemented a PoseNet[1] version which only required a standard webcam. We found the speed/accuracy tradeoff of PoseNet made this approach only suitable for the simplest of movements.

The Kinect interfaces with the NodeJS server which sends the poses to the browser, whereas the browser directly accesses the webcam for the PoseNet approach. In both cases, the target videos are stored and accessed via the NodeJS server which sends the video to the browser as images. We found this approach more robust than seeking an HTML5 video in the browser. A Python test framework was developed for algorithm development which interfaced with the NodeJS server over SocketIO to control the output. Algorithms for adaptive playback could be implemented either directly in the browser, or by using the framework to take advantage of the extensive libraries available for Python.

For pose extraction we developed two programs for extracting movement from video. The first was a simple recording program which records from the Kinect and creates a JSON file with the relevant data. We also developed a program which takes any video as input and uses OpenPose to extract skeletal data to be used directly by Reactive Video. The program filters the skeletal data using SciPy's implementation of a zero-lag Butterworth filter to remove high frequency noise, the values of which we tailored depending on the video to be post-processed (e.g. frame rate, speed of motion).

ADAPTIVE VIDEO PLAYBACK

The core component of Reactive Video is that it can adapt the playback of a video to mimic the user's movements. This is achieved in three stages:

1. *Registration:* The user's skeleton is spatially aligned to the instructor's.
2. *Estimation:* Based on the user's movements we estimate which play time in the video corresponds to their intentions.
3. *Rendering:* We may wish to vary what to show given our estimation of where we believe the user is in the video.

We view the issue of determining where in the video the user is, as similar to that of curvilinear dragging in DMVN interfaces. Several approaches have been proposed for estimating the play time when the user controls video playback by dragging objects along their visual trajectories [26, 13, 23]. In essence, these approaches minimise a spatiotemporal distance metric based on the position of the cursor relative to the target trajectory, whilst accounting for temporal changes to minimise discontinuous jumps. During initial development we found these approaches did not work beyond the simplest of videos, due to a number of additional challenges posed by matching user movements to the video in the context of Reactive Video. In this section, we describe the requirements for adaptive playback, reflect on the challenges of replacing a mouse or touch input with body gestures in the proposed context, and introduce probabilistic approaches aimed at addressing these issues. We only consider the case of 2D matching due to the ability to post-process pre-existing videos to extract skeletal poses.

[1] PoseNet: https://github.com/tensorflow/tfjs-models/tree/master/posenet

Requirements and Challenges

Dragicevic et al. postulate five basic requirements for "correct" behaviour of curvilinear dragging [13]:

- *Multi-scale:* Both coarse- and fine-grained dragging should be supported.
- *Arc-length continuity:* Continuity should be maintained for self-intersecting trajectories, such that the play time does not jump at the intersection.
- *Directional continuity:* Continuity should be maintained for recurring movements and at cusps, where the forward and backwards trajectory are similar.
- *Proximity:* The offset between cursor and target should be minimal when the user stops dragging to minimise spatial indirection.
- *Responsiveness:* There should be no noticeable delay between user movement and interface response to minimise temporal indirection.

In addition to these requirements, using body movements to control playback in the context of Reactive Video presents several unique challenges:

- *Continuous Playback:* Curvilinear dragging is used for seeking to a position in previous DMVN applications. In contrast, adaptive video playback continuously updates the play time of the video, and therefore the requirement to reduce both small and large jumps in playback, and ensure a smooth output, is important for the user experience.
- *Temporal Ambiguities:* Temporal ambiguities, e.g. a prolonged pause in the video, can cause erratic jumps or are skipped altogether using previously proposed DMVN approaches. DragLocks addressed this for traditional DMVN applications by expanding the spatial point using a loop, or transitioning from spatial to time-based control around the ambiguity [25]. However, modifying the trajectory is unsuitable for Reactive Video because users should be able to intuitively navigate pauses without breaking the immersion.
- *User Errors:* As one of the goals of Reactive Video is to help users learn an exercise, there is increased chance that they may not be able to accurately mimic the exercise movements, creating uncertainty for play time estimation.
- *Multiple Control Points:* Previous approaches utilise a single control point, e.g. a cursor position, for object dragging. In Reactive Video, play time estimation may be based on multiple control points corresponding to different joints of the body. This exacerbates the issue of user error for complex movements.
- *Sensor Accuracy:* Users are precise and accurate using a mouse or with touch-based input, where mapping of input device to output space is well-defined. Body movement sensors do not have the same sensing capabilities, and mapping user input to the video can be affected by sensor placement and physiological differences between people.

Registration

Registering the user's pose to the instructor compensates for differences in camera perspectives between the input and recording devices, and physiological variability between the user and instructor. It is important for estimating the playback correctly, and can also be used for providing accurate feedback to the user in the learning mode. Registration involves a transformation using translation, scaling, and rotation, or a subset of these, and can be applied to all joints *globally*, or specific joints *locally*.

The type of registration affects how well the user needs to be synchronised with the video to affect playback. For example, consider comparing the hand position of two poses. If we transform the user's skeleton globally, the position of their hand is based on the position of the torso and legs, and therefore the pose errors in other parts of the body propagate to the hand. In contrast, if we transform the user's arm to the origin of the shoulder, then the movement can be isolated relative to the rest of the body. The type of registration used for playback estimation may differ to the alignment shown to the user as feedback (as in Figure 2). As we consider 2D poses, accurate detection of limb lengths, which has been used in previous work [1, 46], is non-trivial. We therefore implemented two different approaches for registration:

Transformation

The *transformation* registration applies a simple transformation of scale and translation globally to all joints of the body. The system calculates scaling values for the x-axis, S_x, based on distance between shoulders, and for the y-axis, S_y, based on the difference between user and instructor trunks, defined as the distance between the neck and the middle of the hips. The trunk proved to be a more stable measurement than using the legs, which we observed could be subject to tracking errors. Translation, t_x and t_y, is calculated as the difference between the mid-point of the shoulders. This requires the user to be globally in sync with the instructor's movement.

Anchored

We observed during development that most users found it difficult to consider the positioning of multiple body parts at once during compound movements involving both arm and leg movement. Rather than applying a global transformation, anchored registration anchors specific body parts to the torso of the instructor. The arms are anchored to the shoulder joints, the legs to the hips, and the head to the neck. Each limb is individually transformed onto its respective anchor point. This reduces the error that is propagated through the body (e.g. hands are out of place because torso is), and is advantageous for isolating the movement of control points. Scaling is calculated in the same way as the transformation approach.

Probabilistic Play Time Estimation

Previous approaches for DMVN rely on a single "best guess" of the play time and do not take into account uncertainties in the system, making them sensitive to jumps in play time estimation. In Reactive Video, uncertainty arises due to differences in the user's pose relative to the video as a result of misalignment of poses in the registration phase, sensor noise, or user error. Probabilistic approaches represent information as probability distributions over a range of guesses whilst taking all evidence into account, and are typically more robust in the face of uncertainties. We view the estimation of play time as a *position tracking* problem, where we assume we know the initial play time based on when the system was activated.

We propose two approaches, discrete Bayes and particle filter, both of which are capable of tracking multiple hypotheses of where we believe the user's movement corresponds to. This is important for directional continuity, where we may want to keep track of two hypotheses to see how they evolve, e.g. one representing the play speed going forward and one representing the play speed in reverse. In this paper, we explore simple probabilistic models with parameters not directly learnt from the data. Our motivation for this is to develop generic algorithms that work well across different videos because of the potential for Reactive Video to work with a wide variety of physical activities and exercise videos.

The instructor in the video is represented as a spatiotemporal trajectory, $\mathbf{Y}$, defined as a sequence of N poses: $\mathbf{Y} = \{\mathbf{y}(t_1), \mathbf{y}(t_2), ..., \mathbf{y}(t_n)\}$, where $\mathbf{y}(t_i)$ is a set of M joints which represent the control points at a given timestamp, t_i: $\mathbf{y}(t_i) = \{\mathbf{y}^1, \mathbf{y}^2, ..., \mathbf{y}^m\}(t_i)$, where $\mathbf{y}^k$ corresponds to the kth joint where $1 \leq k \leq M$, and represents the joint as a two-dimensional point in the camera's coordinate system. Given a user's current pose, $\mathbf{z}(t_j)$, consisting of m joints, at time j, our goal is to estimate which play time from the video best represents the user's intention.

Discrete Bayes Filter

A discrete Bayes filter recursively estimates a probability density function (PDF), $p(x_i^t|z,u)$, over a discrete latent state, x, conditioned on the history of observations, z, and actions, u [48]. The PDF is represented as a histogram across the latent state space, x, where each state is a frame in the video (i.e. discretised play time). The observations, z, are the user's pose, and the actions, u, represent a simplistic model of the user's intentions. To estimate which play time the user is in the video, we calculate the *posterior* probability using Bayes theorem:

$$p(x_i^t|z,u) = \frac{p(z|x_i).p(x_i^t)}{p(z)} \qquad (1)$$

An initial *prior* probability represents our starting belief of which frame the user is at. We then recursively estimate the position of the play time in two stages: prediction and update.

Initialisation: We initialise our prior belief with a point mass distribution that centres all probability mass on the initial play time at which the system was activated, and assign zero probability elsewhere (practically we assign very small probability).

Prediction: We use a very simple process model to represent how we anticipate the user will behave, in which we assume at each time step they will proceed with unit play speed. The uncertainty associated with this is represented with a Gaussian kernel of length N_p and variance σ_p^2. After applying unit play speed (i.e. shifting the histogram right), we convolve our new prior belief with this Gaussian kernel. We use this to predict the prior probability for each state estimate, x_i^t, using the theorem of total probability:

$$p(x_i^t) = \sum_j p(x_i^t|u_t, x_j^{t-1}).p(x_j^{t-1}) \qquad (2)$$

Where $p(x_i^t|u_t, x_j^{t-1})$ represents our process model, which is not focused on a single pose, but on a continuum of poses centred around the expected outcome (of unit play speed).

Update: The update step takes into account measurements from the user (i.e. their current pose) to update our state estimation. For each state, we compute a likelihood value which represents how likely it is that the user's pose matches each state. For this, we calculate the average Euclidean distance between each of the control points between the user's current pose and the target poses. Uncertainty due to sensor or user error is modelled using a Gaussian likelihood:

$$p(z|x_i) = \mathcal{N}(\frac{1}{M}\sum_j^M ||(z_j - y_j)||; 0, \sigma_L^2) \qquad (3)$$

Where $P(z|x_i)$ is the likelihood, $\mathcal{N}(x; \mu, \sigma^2)$ is a Gaussian with mean, μ, and variance, σ^2, M is the number of control points, z_j is the position of the jth control point from the measurement, y_j is the position of the jth control point from the pose corresponding to state x_i, and i is the frame index. We use *selective updating* which only updates those frames within a certain time distance, t_d of the currently estimated play time to reduce computational complexity [48]. The likelihood function is a simplification of the underlying measurement probability, and we would expect a more complicated function based on the complexity of the movements in the video, and the skill of the user, to yield better results.

Estimation: After normalisation, the estimated play time can be calculated from the posterior by taking the value with the highest probability – the maximum a posterior (MAP) prediction, or the expected value, $E[X]$:

$$E[X] = \sum_i p(x_i)x_i \qquad (4)$$

Where x_i is the ith state and $p(x_i)$ is the corresponding estimated probability.

Particle Filter

The second approach we developed was a particle filter (PF), which uses a set of particles (or samples) to represent the posterior distribution [21, 44]. Each particle has an associated weight which corresponds to the probability that it represents the true position of the video play time. PFs allow us to extend our state estimation to two dimensions, with which we can estimate both the play time and play speed, and model our states as continuous variables which change smoothly over time. In contrast to the discrete Bayes, in which the pauses in the video are navigated at unit play speed due to the process model, the play speed in the PF approach will be dependent on the user's behaviour prior to the pause due to the incorporation of play speed into the model. The disadvantage of PFs is the computational complexity required to track each particle. There are five main stages involved in the approach: initialisation, prediction, update, estimation, and resampling.

Initialisation: We first begin by initialising a set of N_p particles around where we think the play time should start – the play time at which the system was activated. We therefore draw from a prior distribution, where the initial play time of a particle, x_0, is randomly sampled from a Gaussian centred around the time at which the system was activated, t_0:

$$\mathcal{N}(x_0; t_0, \sigma_{x_0}^2) \qquad (5)$$

Likewise, the initial play speed, $\dot{x}_0$, of a particle is drawn from a Gaussian centred around unit play speed:

$$\mathscr{N}(\dot{x}_0; 1, \sigma^2_{\Delta_0}) \tag{6}$$

After the initialisation, we have no reason to favour one particle over another, therefore we assign a weight of $\frac{1}{N}$ for all particles, so that the sum of all weights equals one.

Prediction: The prediction stage incorporates the dynamics of the system and associated uncertainties. We assume that the play time is steady with some drift:

$$x_{t+1} = x_t + \dot{x}_t + a_x \tag{7}$$

Where x_t is the play time at time t, $\dot{x}_t$ the play time velocity, and a_x is a random variable drawn from a Gaussian distribution, $\mathscr{N}(a_x; 0, \sigma_x^2)$. We also assume that the play time velocity can also drift slowly, such that:

$$\dot{x}_{t+1} = \dot{x}_t + a_{\dot{x}} \tag{8}$$

Where $a_{\dot{x}}$ is a random variable drawn from a Gaussian distribution, $\mathscr{N}(a_x; 0, \sigma^2_\Delta)$.

Update: A weighting function is used to update the weights of the particles based on how well they, and the associated frames they represent, agree with the user's pose. Those particles whose frames are closer to the user's pose are weighted more highly. We use the same Gaussian likelihood used in equation 3 for the weighting function.

Estimation: We can calculate our estimate of the state based on either the particle with the largest weight, or based on the expected value of the particles once their weights have been normalised (using equation 4).

Resampling: At each iteration, there will be particles with low weights that are highly improbable. To avoid accumulating lots of improbable particles, resampling replaces particles with low probability with particles of higher probability. We use low variance resampling [48], in addition to drawing from our prior distribution as in the initialisation stage with an initial time of our current state estimate.

Rendering

The rendering stage can be used to ensure that the rendered frames appear smooth to the user. Real-time low-pass filtering techniques, such as the Kalman or One Euro filters [6], can reduce small fluctuations in play times due to user behaviour (e.g. small corrections), sensor errors in either target video or user, or erroneous estimates. The rendered frames need not correspond to the estimated play time which represent the user's intention. For example, if the temporal synchronisation with the video is important, the play time can be used to encourage the user to speed up or slow down. Alternatively, for learning movements an offset can be added to the play time such to ensure the video is always ahead of the user's movement, provided the proximity requirement is satisfied during pauses and slower movements.

EVALUATION

We undertook a controlled data collection to demonstrate the feasibility of adaptive video playback and evaluate the proposed probabilistic approaches. Participants were recorded mimicking movements against six different exercise videos. To assess the approaches' ability to track different play speeds, modified videos of each exercise were created to elicit controlled changes in tempo from participants. The recordings were then post-processed to assess the proposed approaches' outputted play times against the ground truths.

Participants

We recruited eight participants (M: 25.5, SD: 2.45) to take part in the data collection. Three participants were female, one of whom had tried tai chi before but did not practice on a regular basis. All participants were fit young adults who participated in various activities including cardio, weights, meditation, and dancing. None of the participants had knowledge of Reactive Video prior to the data collection.

Exercise Videos

Figure 4 shows the six physical exercises selected: four tai chi movements (A-D) and two Radio Exercises (E, F) – a form of warm-up callisthenics popular in Japan. The tai chi movements involved multiple repetitions of the same movement, whereas the radio exercises involved multiple elements, including arm stretching followed by the movements shown in Fig. 4. These videos were specifically chosen to be simple enough for inexperienced users to complete, whilst also containing challenging elements for the approaches. Exercises C and F contain movements which introduce directional ambiguity. Exercise D increases the likelihood of user error due to the complexity of the movement which requires synchrony between both arms. Exercise E increases sensor error which involves crossed arms with high speed movement. Exercise A involves subtle leg movements, and B is asymmetrical.

The exercise videos were filmed by two of the authors. Exercises A and B were recorded using a video recorder at 60 fps, with skeletons extracted using OpenPose and filtered with a fifth order Butterworth filter with critical frequency of 0.025 half-cycles per second. These were filmed at a height of 1.7 m. The remaining four videos were recorded at 30 fps using the Microsoft Kinect, recorded at a height of 1.8 m. Videos A-D last for 60 seconds, whereas E and F run for 30 seconds.

In addition to unit play speed, we included three variants with modified play speed to illicit different participant behaviour, see Figure 5(a). Changes in play time were controlled to provide a ground truth to assess the approaches, and were designed to be gradual rather than abrupt. The modifications to the instructor videos were presented in the following order for each exercise: no change (unit play speed), fast-pause-slow (FPS), slow-pause-fast (SPF), slow-reverse-fast (SRF).

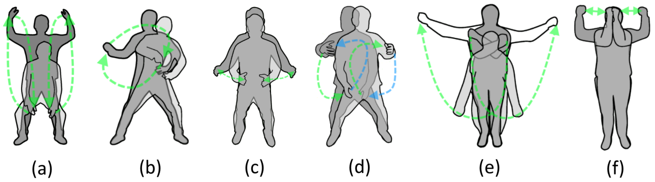

Figure 4. Illustration of the exercises used to evaluate the feasibility of the system: (a-d) Reciprocal tai chi movements, and (e-f) Radio exercises.

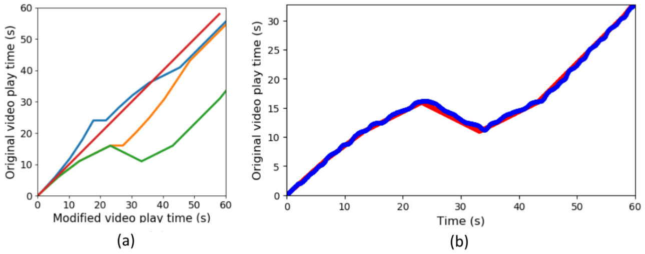

Figure 5. (a) Modified videos were created of each exercise to elicit controlled changes in tempo from the user: unit play speed (red), fast-pause-slow (blue), slow-pause-fast (orange), and slow-reverse-fast (green). (b) Example output for the slow-reverse-fast variant for P2 performing exercise B, showing the ground truth of the play times the participant was following on-screen (red), and the estimated play time according to the discrete Bayes filter (blue).

Apparatus

All participant exercises were recorded using a Microsoft Kinect V2. Participants were positioned 2.6 m away from the sensor in the centre of the optical axis. The Kinect was positioned at a height of 1.8 m located directly above a 60" smart TV. One participant performed the data collection in a different setting in which a 50" smart TV was used instead.

Data Collection Procedure

Participants were first handed a consent form and information sheet, prior to filling out demographic information. They were then introduced to the Reactive Video system and were told the purpose of the data collection was to record data of physical exercises which would be used for analysis of the proposed approaches. All participants were presented with exercises A-F in ascending order. Participants first watched the video to get an idea of the movements involved. Once they were happy with the movement, they performed an activation gesture which restarted the video and began the recording. Participants then mimicked the movement on-screen. For times when the target video paused, green text was displayed and the user was instructed to pause in their current pose.

Algorithm Parameters

In search for an estimation approach that works well across the different videos we varied the parameters for the probabilistic approaches. The position noise for both approaches is defined as a percentage of the instructor's trunk length, measured from the neck to the hips, to account for different scaling across videos. We used the following parameters for the algorithms:

Discrete Bayes filter: The length of the Gaussian kernel, N_p, used for convolution of the prior with the system dynamics was set to $5 \times \sigma_p^2$. As these values are index based (rather than time), the values were doubled for the 60 fps recordings. The values used were: selective updating (t_d) [2.5 s, 5.0 s], prior Gaussian kernel (σ_p^2) [1, 3, 5, 7], position noise (σ_L^2) [0.3, 0.4, 0.5], and both MAP and expected estimations of the posterior.

Particle filter: We used 1000 particles, and the Gaussian distributions used for the prior ($\sigma_{x_0}^2$, $\sigma_{\Delta_0}^2$) were set to σ_x^2 and σ_Δ^2 respectively. We varied the prior resampling rate [0%, 10%], play time noise (σ_x^2) [0.1, 0.05, 0.01], velocity noise (σ_Δ^2) [0.01, 0.005, 0.001], and position noise (σ_L^2) [0.3, 0.4, 0.5].

Table 1. Average play time difference (s) for the best parameters for the discrete Bayes filter and particle filter algorithms with anchored registration for each video and manipulation, averaged across participants.

Video	Algorithm	None	FPS	SPF	SRF	Ave.
A	Discrete Bayes	0.35	0.39	0.38	2.11	0.80
	Particle Filter	0.37	0.41	0.40	1.41	0.64
B	Discrete Bayes	0.21	0.41	0.33	1.50	0.61
	Particle Filter	0.24	0.43	0.36	0.32	0.34
C	Discrete Bayes	0.43	1.28	0.87	4.35	1.73
	Particle Filter	0.83	3.46	1.43	4.40	2.53
D	Discrete Bayes	0.57	0.72	0.72	0.72	0.68
	Particle Filter	0.46	0.74	0.75	0.67	0.65
E	Discrete Bayes	0.38	0.64	0.72	0.84	0.64
	Particle Filter	1.22	1.43	0.84	1.02	1.13
F	Discrete Bayes	0.43	0.61	0.67	0.79	0.63
	Particle Filter	0.69	0.66	0.85	1.27	0.87

Data Analysis

The recordings of the participants were played through the test framework as if they were a real-time data stream, where each recorded video of a participant is played against its respective original target video, see Fig. 5(b) for an example. We used both hands as the control points to estimate play time. The system does not activate until two seconds after the activation gesture, to account for the user re-positioning themselves, during which the target video is played at unit play speed. For each recording, we have the ground truth of the play times assuming the user perfectly followed the instructor in the videos. We define the best algorithm as the one which minimises the absolute difference between estimated play times and the ground truths. The participants may not be in perfect synchrony with the target video, thus even with a perfect algorithm we expect some leading or lagging.

Results

The discrete Bayes filter with anchored registration performed the best across all videos, with an average time difference between the estimated play times and ground truths of 0.85 s. The anchored registration was also best for the particle filter, achieving an average difference of 1.03 s. In both cases, anchored registration significantly outperformed the global transformation registration method (DB: 1.44 s, PF: 1.37 s). The best parameters for the discrete Bayes estimator were $t_d = 2.5s$, $\sigma_p^2 = 3$, $\sigma_L^2 = 0.5$, with the expected value of the posterior as the output. For the particle filter the best values were $\sigma_x^2 = 0.1$, $\sigma_\Delta^2 = 0.005$, $\sigma_L^2 = 0.5$, with 10% of particles resampled from the prior each iteration.

Analysis of the individual videos (Table 1) showed that the approaches were able to successfully adapt their playback across the videos, however the reverse variants were the most problematic to accurately track. In particular, we observe large differences with exercise C due to the high level of directional ambiguity. It is ill-defined as to which direction the algorithm should play when users reverse playback at the turning points, and we observed that the differences in play time were due to the tendency to play forward rather than reverse (by design).

We observed the Bayes filter performed better with directional ambiguity when the intention was to play forward, however this bias could hinder reverse playback. The particle filter outperformed the discrete Bayes in some exercises (A, B, D). It similarly struggled with the reverse playback variants, although to a lesser extent than the discrete Bayes filter.

DISCUSSION

Our results demonstrate how adaptive video playback can successfully track the user's intentions across different physical movements, and at different tempos. Reflecting on the challenges of adaptive video playback in the context of Reactive Video, the probabilistic approaches provide continuous play time estimations by taking into consideration the uncertainties between user and instructor poses which arise due to sensor inaccuracies and user error. They are able to gracefully cope with temporal ambiguities due to the underlying models which drive the video forward in the absence of any useful observations (i.e. no instructor movement), and their ability to track multiple hypotheses ensures directional continuity is maintained in the majority of cases tested. Reverse playback posed a difficulty for some videos, however we would not expect it to be used often, or for prolonged periods. We have demonstrated adaptive video playback based on the movement of both hands, however this is easily extendable to multiple control points.

Reactive Video leverages adaptive video playback to provide a highly deployable platform for enhancing the interactivity of traditional videos. The use of state-of-the-art skeletal trackers enables videos to be converted for use with Reactive Video, presenting the opportunity to use the thousands of available videos. Although we did not formally study the system in use, during development we gained several valuable insights from visitors to the research lab using the system:

- *Users tend to mirror the instructor.* We observed that most users of the system, including the participants of the data collection, naturally mirror the instructor's movements. However, some users preferred non-mirrored interaction.
- *Continuous feedback can enshroud learning of the underlying movements.* Similar to previous findings [46], we observed that even simple feedback offered to users on the pose error could be distracting. Users had a tendency to focus too much on aligning the dots on-screen, rather than focusing on the underlying movements.
- *Increasing the number of control points requires stricter matching between user and instructor.* When learning a new movement, we observed it was easier for users to start with fewer control points, before increasing them to learn the finer intricacies of the movement. For some exercises, some control points may be more important to a movement than others, which could be reflected by weighting them accordingly in the likelihood functions.
- *Explicit intent to interact is more useful.* Despite motion correlation presenting a seamless alternative to activation gestures, we found that it could be problematic if the user wanted to copy the movements on-screen without activating the system and was most suited for walk-up and use scenarios. However for prolonged use a more explicit form of starting, and decoupling from, the adaptive playback proved to be more useful. Using body movement for intent and decoupling can prove problematic because the system uses this as input, which can result in accidental (de)activations [27]. Alternative modalities, such as voice recognition showed promising results using Sphinx [50].
- *Rendering provides further opportunities for filtering.* The probabilistic approaches output smooth movement but can reflect the nuances in the way in which a user performs the movements. For example, a user may mimic slow, smooth tai chi movements in a more ballistic fashion. The use of rendering stage filtering can suppress these intricacies, but must be carefully tuned to avoid lag.

In traditional video, the user passively watches or copies the video, but can never take the lead. In Reactive Video, the user's action can lead and the video can follow, making the user the causal source. However, when users are learning a new skill they mimic the instructor. With Reactive Video this leads to a causality dilemma - a subtle circular feedback case where it is not clear who is leading, but which can create an immersive experience of connection with the teacher. The evaluation presented in this paper focused on the system's ability to track the user through the video, where the recordings are representative of a user who knows the exercise and is not relying on the instructor for cues. The unique capabilities of Reactive Video begs the question to what extent synchrony between user and instructor increases immersion and feelings of engagement, which have been shown in interpersonal settings and virtual environments [18, 17, 47].

Adaptive video playback invites the exploration of how it can be used to support users practice and learn movements beyond the limited range of exercises used in the evaluation (e.g. high speed movements involving occlusion), and to other contexts such as how-to videos [7]. Other skeletal trackers, e.g. OpenPose, output confidence of the joints which can be used as input for the approaches to reflect the sensor's capabilities. However, OpenPose is still computationally expensive, and even with GPU-support can not run in real-time. Our current implementation processes poses in 2D coordinate space, limiting the ability to control the content of the video in the z-dimension, and requiring the user's perspective to match the instructors. Although it is common for exercise videos to be filmed as if the user is in front of a mirror, the ability to match between different perspectives would enhance the capabilities of the system.

CONCLUSION

Reactive Video enhances the utility of existing videos designed for physical activities, requiring only minimal hardware. The system can post-process existing videos to extract instructor poses, opening up the opportunity to use thousands of existing videos. At the core of the system is adaptive video playback, which mirrors the user's movements and enables intuitive, immersive control of the video playback without breaking flow. Inspired by direct manipulation video navigation, our work demonstrates how the unique challenges posed by full body movement matching can be overcome using probabilistic approaches which take into account the uncertainties in the system, ensuring smooth playback.

REFERENCES

[1] Fraser Anderson, Tovi Grossman, Justin Matejka, and George Fitzmaurice. 2013. YouMove: enhancing movement training with an augmented reality mirror. In *Proceedings of the 26th annual ACM symposium on User interface software and technology*. 311–320.

[2] Olivier Bau and Wendy E Mackay. 2008. OctoPocus: a dynamic guide for learning gesture-based command sets. In *Proceedings of the 21st annual ACM symposium on User interface software and technology*. 37–46.

[3] Benjamin B Bederson. 2004. Interfaces for staying in the flow. *Ubiquity* 2004, September (2004), 1–1.

[4] Victoria Bellotti, Maribeth Back, W Keith Edwards, Rebecca E Grinter, Austin Henderson, and Cristina Lopes. 2002. Making sense of sensing systems: five questions for designers and researchers. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. 415–422.

[5] Z. Cao, G. Hidalgo Martinez, T. Simon, S. Wei, and Y. A. Sheikh. 2019. OpenPose: Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2019).

[6] Géry Casiez, Nicolas Roussel, and Daniel Vogel. 2012. 1€ filter: a simple speed-based low-pass filter for noisy input in interactive systems. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 2527–2530.

[7] Minsuk Chang, Anh Truong, Oliver Wang, Maneesh Agrawala, and Juho Kim. 2019. How to design voice based navigation for how-to videos. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. 1–11.

[8] Dale Chapman, Michael Newton, Paul Sacco, and Kazunori Nosaka. 2006. Greater muscle damage induced by fast versus slow velocity eccentric exercise. *International journal of sports medicine* 27, 08 (2006), 591–598.

[9] Winyu Chinthammit, Troy Merritt, Scott Pedersen, Andrew Williams, Denis Visentin, Robert Rowe, and Thomas Furness. 2014. Ghostman: augmented reality application for telerehabilitation and remote instruction of a novel motor skill. *BioMed research international* 2014 (2014).

[10] Christopher Clarke and Hans Gellersen. 2017. MatchPoint: Spontaneous Spatial Coupling of Body Movement for Touchless Pointing. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology*. 179–192.

[11] Alana Da Gama, Thiago Chaves, Lucas Figueiredo, and Veronica Teichrieb. 2012. Guidance and movement correction based on therapeutics movements for motor rehabilitation support systems. In *2012 14th symposium on virtual and augmented reality*. IEEE, 191–200.

[12] Iwan de Kok, Julian Hough, Felix Hülsmann, Mario Botsch, David Schlangen, and Stefan Kopp. 2015. A multimodal system for real-time action instruction in motor skill learning. In *Proceedings of the 2015 ACM on International Conference on Multimodal Interaction*. 355–362.

[13] Pierre Dragicevic, Gonzalo Ramos, Jacobo Bibliowitcz, Derek Nowrouzezahrai, Ravin Balakrishnan, and Karan Singh. 2008. Video browsing by direct manipulation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 237–246.

[14] Jonathan Driedger and Meinard Müller. 2016. A review of time-scale modification of music signals. *Applied Sciences* 6, 2 (2016), 57.

[15] Euan Freeman, Stephen Brewster, and Vuokko Lantz. 2016. Do that, there: an interaction technique for addressing in-air gesture systems. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. 2319–2331.

[16] Dan B Goldman, Chris Gonterman, Brian Curless, David Salesin, and Steven M Seitz. 2008. Video object annotation, navigation, and composition. In *Proceedings of the 21st annual ACM symposium on User interface software and technology*. 3–12.

[17] Joanna Hale and F De C Antonia. 2016. Testing the relationship between mimicry, trust and rapport in virtual reality conversations. *Scientific reports* 6 (2016), 35295.

[18] Joanna Hale and Antonia F de C Hamilton. 2016. Cognitive mechanisms for responding to mimicry from others. *Neuroscience & Biobehavioral Reviews* 63 (2016), 106–123.

[19] Ping-Hsuan Han, Yang-Sheng Chen, Yilun Zhong, Han-Lei Wang, and Yi-Ping Hung. 2017. My Tai-Chi coaches: an augmented-learning tool for practicing Tai-Chi Chuan. In *Proceedings of the 8th Augmented Human International Conference*. 1–4.

[20] Ken Hinckley, Patrick Baudisch, Gonzalo Ramos, and Francois Guimbretiere. 2005. Design and analysis of delimiters for selection-action pen gesture phrases in scriboli. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM, 451–460.

[21] Michael Isard and Andrew Blake. 1996. Contour tracking by stochastic propagation of conditional density. In *European conference on computer vision*. Springer, 343–356.

[22] Marcelo Kallmann, Carlo Camporesi, and Jay Han. 2015. Vr-assisted physical rehabilitation: Adapting to the needs of therapists and patients. In *Virtual realities*. Springer, 147–168.

[23] Thorsten Karrer, Malte Weiss, Eric Lee, and Jan Borchers. 2008. DRAGON: a direct manipulation interface for frame-accurate in-scene video navigation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 247–250.

[24] Thorsten Karrer, Moritz Wittenhagen, and Jan Borchers. 2009. Pocketdragon: a direct manipulation video navigation interface for mobile devices. In *Proceedings of the 11th International Conference on Human-Computer Interaction with Mobile Devices and Services*. 1–3.

[25] Thorsten Karrer, Moritz Wittenhagen, and Jan Borchers. 2012. DragLocks: handling temporal ambiguities in direct manipulation video navigation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 623–626.

[26] Don Kimber, Tony Dunnigan, Andreas Girgensohn, Frank Shipman, Thea Turner, and Tao Yang. 2007. Trailblazing: Video playback control by direct object manipulation. In *2007 IEEE International Conference on Multimedia and Expo*. IEEE, 1015–1018.

[27] Rick Kjeldsen and Jacob Hartman. 2001. Design Issues for Vision-based Computer Interaction Systems. In *Proceedings of the 2001 Workshop on Perceptive User Interfaces (PUI '01)*. ACM, New York, NY, USA, 1–8. DOI:http://dx.doi.org/10.1145/971478.971511

[28] Brittany Kondo and Christopher Collins. 2014. Dimpvis: Exploring time-varying information visualizations by direct manipulation. *IEEE transactions on visualization and computer graphics* 20, 12 (2014), 2003–2012.

[29] Yousef Kowsar, Masud Moshtaghi, Eduardo Velloso, Lars Kulik, and Christopher Leckie. 2016. Detecting unseen anomalies in weight training exercises. In *Proceedings of the 28th Australian Conference on Computer-Human Interaction*. 517–526.

[30] Myunghee Lee and Gerard J Kim. 2010. Empathetic video experience through timely multimodal interaction. In *International Conference on Multimodal Interfaces and the Workshop on Machine Learning for Multimodal Interaction*. 1–4.

[31] Klemen Lilija, Henning Pohl, and Kasper Hornbæk. 2020. Who Put That There? Temporal Navigation of Spatial Recordings by Direct Manipulation. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems (CHI '20)*. Association for Computing Machinery, New York, NY, USA, 1–11. DOI:http://dx.doi.org/10.1145/3313831.3376604

[32] Microsoft. 2020. Kinect for Windows. (2020). https://developer.microsoft.com/en-us/windows/kinect/

[33] Dan Morris, T Scott Saponas, Andrew Guillory, and Ilya Kelner. 2014. RecoFit: using a wearable sensor to find, recognize, and count repetitive exercises. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 3225–3234.

[34] Cuong Nguyen, Yuzhen Niu, and Feng Liu. 2013. Direct manipulation video navigation in 3D. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 1169–1172.

[35] Cuong Nguyen, Yuzhen Niu, and Feng Liu. 2014. Direct manipulation video navigation on touch screens. In *Proceedings of the 16th international conference on Human-computer interaction with mobile devices & services*. 273–282.

[36] Nintendo. 2020. Wii Fit. (2020). https://www.nintendo.co.uk/Games/Wii/Wii-Fit-283894.html

[37] Dario Pavllo, Christoph Feichtenhofer, David Grangier, and Michael Auli. 2019. 3D human pose estimation in video with temporal convolutions and semi-supervised training. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.

[38] Ken Pfeuffer, Mélodie Vidal, Jayson Turner, Andreas Bulling, and Hans Gellersen. 2013. Pursuit Calibration: Making Gaze Calibration Less Tedious and More Flexible. In *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology (UIST '13)*. ACM, New York, NY, USA, 261–270. DOI:http://dx.doi.org/10.1145/2501988.2501998

[39] Simon Rogers, John Williamson, Craig Stewart, and Roderick Murray-Smith. 2010. FingerCloud: uncertainty and autonomy handover incapacitive sensing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 577–580.

[40] Takashi Satou, Haruhiko Kojima, Akihito Akutsu, and Yoshinobu Tonomura. 1999. CyberCoaster: Polygonal line shaped slider interface to spatio-temporal media. In *Proceedings of the seventh ACM international conference on Multimedia (Part 2)*. 202.

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

[42] Richard A Schmidt and Gabriele Wulf. 1997. Continuous concurrent feedback degrades skill learning: Implications for training and simulation. *Human factors* 39, 4 (1997), 509–525.

[43] Roland Sigrist, Georg Rauter, Robert Riener, and Peter Wolf. 2013. Augmented visual, auditory, haptic, and multimodal feedback in motor learning: a review. *Psychonomic bulletin & review* 20, 1 (2013), 21–53.

[44] Adrian Smith. 2013. *Sequential Monte Carlo methods in practice*. Springer Science & Business Media.

[45] Maurício Sousa, João Vieira, Daniel Medeiros, Artur Arsenio, and Joaquim Jorge. 2016. SleeveAR: Augmented reality for rehabilitation using realtime feedback. In *Proceedings of the 21st international conference on intelligent user interfaces*. 175–185.

[46] Richard Tang, Xing-Dong Yang, Scott Bateman, Joaquim Jorge, and Anthony Tang. 2015. Physio@ Home: Exploring visual guidance and feedback techniques for physiotherapy exercises. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. 4123–4132.

[47] Bronwyn Tarr, Mel Slater, and Emma Cohen. 2018. Synchrony and social connection in immersive Virtual Reality. *Scientific reports* 8, 1 (2018), 1–8.

[48] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. 2005. Probabilistic robotics. 2005. *Massachusetts Institute of Technology, USA* (2005).

[49] Milka Trajkova and Mexhid Ferati. 2015. Usability evaluation of kinect-based system for ballet movements. In *International Conference of Design, User Experience, and Usability*. Springer, 464–472.

[50] Carnegie Mellon University. 2020. CMUSphinx Open Source Speech Recognition. (2020). https://cmusphinx.github.io/

[51] Jeroen JG Van Merrienboer and John Sweller. 2005. Cognitive load theory and complex learning: Recent developments and future directions. *Educational psychology review* 17, 2 (2005), 147–177.

[52] Neil Vaughan, Bodgan Gabrys, and Venketesh N Dubey. 2016. An overview of self-adaptive technologies within virtual reality training. *Computer Science Review* 22 (2016), 65–87.

[53] Eduardo Velloso, Andreas Bulling, and Hans Gellersen. 2013. MotionMA: motion modelling and analysis by demonstration. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 1309–1318.

[54] Eduardo Velloso, Marcus Carter, Joshua Newn, Augusto Esteves, Christopher Clarke, and Hans Gellersen. 2017. Motion Correlation: Selecting Objects by Matching Their Movement. *ACM Trans. Comput.-Hum. Interact.* 24, 3, Article 22 (April 2017), 35 pages. DOI: http://dx.doi.org/10.1145/3064937

[55] Robert Walter, Gilles Bailly, and Jörg Müller. 2013. StrikeAPose: revealing mid-air gestures on public displays. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 841–850.

[56] Yoshihiro Watanabe, Hiroaki Ohno, Takashi Komuro, and Masatoshi Ishikawa. Synchronized Video: An Interface for Harmonizing Video with Body Movements. In *22nd Symposium on User Interface Software and Technology (UIST2009)(Victoria, 2009.10. 5)/Adjunct Proceedings*. 75–76.

[57] WL Westcott, RA Winett, ES Anderson, JR Wojcik, and others. 2001. Effects of regular and slow speed resistance training on muscle strength. *Journal of sports medicine and physical fitness* 41, 2 (2001), 154.

[58] John Williamson. 2006. *Continuous uncertain interaction*. Ph.D. Dissertation. University of Glasgow.

[59] Carolee J Winstein and Richard A Schmidt. 1990. Reduced frequency of knowledge of results enhances motor skill learning. *Journal of Experimental Psychology: Learning, Memory, and Cognition* 16, 4 (1990), 677.

[60] Gabriele Wulf, Markus Raupach, and Felix Pfeiffer. 2005. Self-controlled observational practice enhances learning. *Research Quarterly for Exercise and Sport* 76, 1 (2005), 107–111.

[61] Ungyeon Yang and Gerard Jounghyun Kim. 2002. Implementation and evaluation of "just follow me": An immersive, VR-based, motion-training system. *Presence: Teleoperators & Virtual Environments* 11, 3 (2002), 304–323.