Trinity College Dublin
Coláiste na Tríonóide, Baile Átha Cliath
The University of Dublin

Understanding and Improving Physical Interactions in Virtual Reality

by

Goksu Yamac

Supervisor: Prof. Carol O'Sullivan

A thesis submitted in partial fulfillment
of the requirements for the degree of

Doctor of Philosophy

in the

School of Computer Science and Statistics

October 2023

Declaration

I declare that this thesis has not been submitted as an exercise for a degree at this or any other university and it is entirely my own work.

I agree to deposit this thesis in the University's open-access institutional repository or allow the Library to do so on my behalf, subject to Irish Copyright Legislation and Trinity College Library conditions of use and acknowledgment.

I consent to the examiner retaining a copy of the thesis beyond the examining period, should they so wish (EU GDPR May 2018).

Signed: Gyanee

October 2023

Dedicated to my late grandparents,
Ayhan and Remziye.

Abstract

An important challenge in AR/VR is to enable virtual interactions that look and feel natural. Our goal in this work was to identify certain failures of AR/VR interactions, understand them, and propose solutions for them so that these platforms can accommodate better and more diverse experiences. To this end, we conducted two case studies on dumbbell *Lifting* and ball *Throwing* interactions in VR, with a focus on *Human Perception*.

We first developed a pilot system for *Throwing* in VR, and conducted a perceptual study to determine the importance of visual trajectory cues on throwing performance. We found that limiting visual feedback detracts from virtual throwing performance. We also ran a study to develop a better understanding of how the Point of Release (PoR) of a ball affects the perception of animated throwing motions. In this study, the participants viewed animations of a virtual human throwing a ball, in which the point of release was modified to be early or late. We found that errors in overarm throws with a late PoR are detected more easily than an early PoR, while the opposite is true for underarm throws. The viewpoint and the distance the ball travels also have an effect on perceived realism. Finally, we hypothesized that the typical experience of throwing in VR, i.e., holding a controller and using a button to release the projectile, may feel unnatural. We therefore developed a novel real-time physical interaction system, called *ReTro*, that allows users to throw a virtual ball without using an intermediary device such as a controller. For the implementation of the ReTro system, we developed a detection algorithm to predict the PoR of a throwing motion in real-time. This was achieved by training a PoR prediction model using motion features extracted from arm joints. The evaluation of ReTro using pre-recorded throwing motion data resulted in detection errors of less than 50 milliseconds. Another output of this analysis is a presentation of the relative importance of different joints and motion features for the PoR prediction task. Finally, qualitative results from users of ReTro in VR indicate that, although it performed better for Underarm than Overarm throws, the task of throwing without a controller felt very natural.

In our second case study of *Lifting*, we investigated people's sensitivity to physicality errors in order to understand when they are likely to be noticeable and need to be mitigated. As a user lifts virtual objects in AR/VR, there may be dynamic inconsistencies in the motion of the virtual avatar due to a mismatch between the shapes of the user's body and their virtual avatar. There could also be a mismatch between the real and virtual objects being interacted with, such as a real controller vs. a virtual boulder. We use the term "physicality errors" to distinguish them from simple physical errors, such as footskate. Physicality errors involve plausible motions, but with dynamic inconsistencies. We used the exercise of a dumbbell lift to explore the impact of motion kinematics and varied sources of visual information, which included changing the sizes of the body and manipulated objects, and displaying muscular strain. Our results suggest that kinematic (motion) information has a dominant impact on the perception of effort, but that visual information, particularly the visual size of the lifted object, has a strong impact on perceived weight. This can lead to perceptual mismatches which reduce perceived naturalness. Small errors may not be noticeable, but large errors reduce naturalness. These results can be used to inform the development of animation algorithms.

Acknowledgements

First and foremost, I would like to heartfully thank my supervisor, Prof. Carol O'Sullivan, for her generous and continuous support, illuminating guidance, and true mentorship. Her open-mindedness in research has made this journey much more exciting. I want to express my gratitude to Prof. Michael Neff, who has been an important collaborator, for the time and effort he has put into helping me learn and explore.

I would like to thank my examiners, Prof. Rachel McDonnell and Prof. Anne-Hélène Olivier, for their time and effort in evaluating my work and giving very valuable feedback. Particularly, Prof. Anne-Hélène Olivier's expertise and willingness to help have improved the quality of this work.

I am thankful to be part of the research group, T-Motion, with all the insightful discussions and fond memories. The people of TCD Graphics and Visualization group have been a good source of knowledge and fun.

I would like to acknowledge the research centres, ADAPT and CONNECT, for funding my research and making it possible for me to get this degree. I would also like to thank Prof. Niloy Mitra, who graciously hosted me for two weeks in UCL at the early stages of my studies.

To all the participants who have participated in my experiments, I extend my thanks. I hope I have been accurate with my duration estimations.

My family has been the bedrock of my success in getting this degree. My partner, Cansu, has offered her unwavering love and support, and has been the lifting hand many times during this journey. My parents, Deniz and Kadri, who have instilled in me hard work and dedication, along with my brother Gökhan, have been always there for me. My dog, Kuki, has been the most joyful distraction and a daily reminder to stay active.

Relevant Publications

- Yamac, G., O'Sullivan, C. & Neff, M. (2023). Understanding the Impact of Visual and Kinematic Information on the Perception of Physicality Errors. ACM Transactions on Applied Perception (TAP) (Conditionally Accepted).
- Yamac, G., Chang, J. J., & O'Sullivan, C. (2023). Let it go! Point of release prediction for virtual throwing. Computers & Graphics, 110, 11-18.
- Yamac, G., & O'Sullivan, C. (2022, September). FauxThrow: Exploring the Effects of Incorrect Point of Release in Throwing Motions. In ACM Symposium on Applied Perception 2022 (pp. 1-5).
- Yamac, G., & O'Sullivan, C. (2022). Eye on the Ball: The effect of visual cue on virtual throwing. In SIGGRAPH Asia 2022 Posters (pp. 1-2).
- Yamac, G., Mitra, N. J., & O'Sullivan, C. (2021, March). Detecting the point of release of virtual projectiles in AR/VR. In 2021 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW) (pp. 563-564). IEEE.

Contents

List of Figures

List of Tables

1 Introduction

Starting in the last decade, a new wave of multimedia technologies has started reaching consumers with the potential of changing the way we interact with computers and with each other digitally as a society. Augmented and Virtual Reality (AR/VR), or Mixed Reality (MR) as an umbrella term, offer more immersive experiences than non-immersive technologies, such as PCs and smartphones, by seamlessly blending real and digital worlds to varying degrees. Modern non-immersive technologies can target our visual and auditory senses with high-fidelity graphics and audio. However, as content is presented within a visually constrained space they cannot offer fully immersive experiences. On the other hand, a seamless augmentation (AR) or replacement (VR) of reality through a head-mounted display can better create the illusion of an alternate reality.

AR and VR are far from reaching their full potential, yet they are already transforming fields such as entertainment, health, and education, and changing how designing, prototyping, training, and collaborations are done. Market forecasts on both AR and VR technologies show that both will have a substantial impact on the future of society. Almost all Big Tech companies entered the market by either developing their own headsets or acquiring companies that work on these technologies. In 2022, Facebook rebranded itself as Meta to announce a strategic plan of building the "metaverse", i.e., a computer-generated and globally connected 3D world that people enter wearing their AR/VR headsets to carry out daily activities such as meetings, sports, shopping and concerts. Commenting on how 5G and VR will combine to shape the future, Intel has proclaimed that:

> *"A new dawn of VR-driven experiences will emerge as early as 2025 [6]."*

Many large organizations are already utilizing AR/VR as an integral part of their work, e.g., NASA for space-walking simulations; Boeing and Airbus for prototyping and design; Audi and IKEA for virtual/augmented showrooms; and law enforcement agencies for training. In research, new directions of research are being pursued to explore the effects of virtual environments (VE) and their content, such as virtual avatars, crowds and interactions. VR is especially useful for enabling fully controllable experimental environments, which can be difficult or impossible to create and maintain in the physical world.

Along with the investments in AR/VR coming to fruition, advancements in computational technologies are enabling the generation and display of sophisticated audiovisual content that is indistinguishable from what we perceive in physical reality. In the field of Computer Graphics, there are very high-definition renderings of content such as real and synthetic humans, objects, and highly realistic scenes. Notably, MetaHuman by Unreal Engine in 2021 was a big leap forward in terms of access to, and usability of, photo-realistic humanoid avatars. Through the provision of such affordable high-end content creation capabilities, now both practitioners and researchers can utilize and investigate this next iteration of humanoid avatars in their work. In audio, there are advancements in speech and sound synthesizers, voice assistants like Siri and Alexa, and binaural sound. For simulating the sense of touch, there is a range of devices from gloves to full-body suits that are equipped with exoskeletons, sensors, and electrodes that generate tactile stimulation. For the senses of taste and smell, there are specialized displays that can be programmed to release stimulants through tubes or capsules, although the advancements have mainly been experimental for now.

It may seem that a congruently combined stimulation of the senses using different modalities in an AR/VR experience can offer full immersion that completely and satisfactorily swaps our experience of reality. However, not only does AR/VR currently lack the technical sophistication to fully replicate real-life multi-sensory experiences, but these different modalities also sum up to a much larger whole that forms our subjective, perceived reality. In a human's experience of reality, perception organizes sensory input and forms a coherent understanding and an identification of the environment. Beyond our sensory modalities, many other subtle concepts are vital components of our perception, such as the senses of agency [7], ownership [8], and embodiment [9]. These subtle senses are generally unchallenged in our daily lives thanks to the consistent nature of our physical reality. Yet, these components are critical for sustaining a user's sense of presence in a VE, which is defined as "the perceptual illusion of non-mediation" by Lombard and Ditton [10].

More formally, Slater and Wilbur [11] define immersion as "a description of a technology, and describes the extent to which the computer displays are capable of delivering an inclusive, extensive, surrounding and vivid illusion of reality to the senses of a human participant", and the sense of presence as "a state of consciousness, the (psychological) sense of being in the virtual environment". In other words, immersion is an objective quality of the displayed multi-modal content, and presence is a subjective measure of how present a user feels in the content (See Chapter 2.1 for a discussion). Ultimately, AR/VR experiences aim to provide and maintain a high sense of presence that is not deteriorated by elements of that experience, such that the experience can fulfill its purpose, be it enjoyment, involvement, learning, and more. However, it has been demonstrated in real-life examples that immersion does not directly lead to a sense of presence [12]. Rather than on better hardware, as noted by Oculus former CTO, J. Carmack:

> *"...the focus needs to be on improving the user experience [13]."*

In the physical world, interactions with objects around us happen naturally, as various forces are exerted between objects and limbs without much conscious mental effort. The details of this force exchange are calculated and flawlessly actuated according to the various properties of interacting surfaces such as friction coefficient, mass, and velocity. However, the problem of simulating natural physical interactions with virtual entities within a dynamically changing environment (real or virtual) remains an open challenge [14]. However, in AR/VR, such physical laws need to be programmed into simulated environments so that the users are presented with an alternate reality that is in some way consistent with their expectations. Due to limitations in hardware and computational power, implementing the laws of physics usually requires approximations to be made, thus risking unwanted perceptual consequences that may diminish the sense of presence. As a result, physics-based animation is often not effective when complex geometries are involved, due to concerns over performance or implausible outcomes.

This becomes an even more challenging problem when one of the involved geometries is of a user's hand. In this case, as a user moves their hand, the complex virtual geometry following the user's physical hand comes in contact with another virtual object. Assuming that the virtual object is a non-deformable and unmovable rigid body, this object applies a limitation on the motion of the virtual hand, but there is nothing that limits the motion of the physical hand in physical reality. Therefore, the physical hand can just move through that space, leading to complexities for both tracking and force calculations. Typically, such implementations include two pairs of virtual hands, with one pair following the physical hands all the time without having any colliders, and the second pair being subject to the physics and having active colliders.

It is therefore clear that many problems need to be addressed to achieve natural and plausible physical interactions in AR/VR. The goal of our research is to identify and evaluate factors that affect the perception of anomalous interactions for two examples (throwing and lifting), and to use this information to inform the development of methods to enhance the user experience. In the next section, we outline the specific problems that we address in AR/VR and share the motivation for studying them.

1.1 Motivation

Research Questions: In this thesis, we address the challenge of simulating natural physical interactions for two interaction types: ball ***throwing*** and dumbbell ***lifting***. We explore the following research questions:

- How sensitive are people to anomalies in the simulation of throwing and lifting in Virtual Reality (for both first-person and third-person views)?
- Can we use knowledge of human perception, together with modern Computer Science techniques, to help increase the plausibility of these interactions?

Early-stage exploration of computers and phones has produced an abundance of knowledge, which is unfortunately only partially transferable to AR/VR. User interaction is one of the most important aspects that has the power to "make or break" a technology. Interactions in computers and phones are limited to two dimensions, mainly using a keyboard/mouse or touch as input, with a display screen as the interface. In AR/VR, user interactions can cover a three-dimensional space, typically carried out with a hand-held controller or one's own hand as tracked by cameras on the device. Several platforms have provided such 3D interfaces in the past, e.g., Nintendo Wii and Sony Move, but their development has been discontinued.

Whereas traditional user interfaces focused on efficiency and convenience by devising easy-to-use interactions, AR/VR interfaces can accommodate natural actions such as pushing, pulling, grasping, lifting, and throwing. An important challenge for AR/VR is to ensure that virtual actions/interactions feel natural to the user in a way that is similar to experiencing the sensations and the outcome of doing them in physical reality. This is particularly important if AR/VR is to become a platform for acquiring and practicing real-life skills. We, therefore, investigate this challenge in our work by exploring *the interaction of throwing a ball in VR*.

AR/VR technologies are considered to have the potential to break the barriers of physical reality by facilitating immersive social interactions and activities, including games, gatherings, events, and more. In such immersive social experiences, a user's experience is shaped not only by their own capabilities and interactions as a user, but also by how plausible other users' representations, i.e., avatars, and interactions are. Therefore, we consider a second important challenge for AR/VR to be that virtual interactions performed by others in the same VE should look plausible to the observer to maintain a sense of presence. To tackle this challenge, we investigate *the interaction of dumbbell lifting*.

1.1.1 Throwing Interactions

To simulate the throwing of a virtual projectile, a physics-based implementation needs to: continuously calculate the exchange of forces between the virtual hand and the virtual projectile; precisely release the projectile when the user opens their hand; and accurately simulate the trajectory as it is released. There are two major issues with this: i) the real hand will not have a sense of the virtual projectile unless an advanced haptic device is employed, and the user will likely hold a hand pose that is too tight or too loose, leading to inaccurate force calculations; ii) the hand has a complex geometry, and precise frame-by-frame calculation of the force exchange will use too much or all of the computational capacity. Therefore, most AR/VR throwing games employ a controller, using it as an attachment point for the object to be thrown and its buttons to indicate the release of the ball. In this case, there is no longer any force calculation, and the problem is simplified to releasing the projectile upon a button press. However, although it works well functionally, throwing with a controller can feel unnatural.

Studies in HCI and games have shown that, when a computer assists a user in completing a task, the feeling of control is often lost [15, 16]. Throwing interactions in VR to date have mostly been implemented using either a controller with a virtual ball [17, 18], or a constrained real ball [19]. A user throwing a virtual ball needs to keep holding a controller during and after the throw, contrary to what the action naturally requires, i.e., releasing the projectile from the hand. Furthermore, the user has to precisely time the release of the projectile by using the buttons on the controller, which also differs from how a throw is performed in physical reality. The effects of such issues would amplify in the context of sports training, where motions are faster and high precision is crucial. Therefore, it makes sense to offer unconstrained interactivity to the user for a more natural throwing experience. We explore the challenge of throwing a virtual ball in VR without the use of a controller.

1.1.2 Lifting Interactions

As humans interact with objects in the physical world, they orchestrate their limbs to carry out that interaction in a specific way, e.g., to minimize the effort of lifting a heavy object, to maximize precision in throwing. When this interaction is observed, observers can typically infer many details about the interaction by perceiving attributes such as posture, facial expressions, interacted objects, and sound. More specifically, when a person is observed lifting an object, the observer can estimate the weight of the interacted object and the amount of effort displayed by the person [20, 21] (See Chapter 2 for a thorough literature review). This is a problematic situation for AR/VR since virtual objects fundamentally do not possess any mass, virtual interactions thus contain the risk of looking unrealistic and deteriorating the presence, e.g., observing the lifting of a virtual boulder with no effort. We use the term "physicality errors" to refer to errors that arise due to a mismatch in the dynamics of a person’s motion and the visualized movements of their avatar in VR. Physicality errors involve plausible motions, but with dynamic inconsistencies. Even with perfect tracking and ideal virtual worlds, such errors are inevitable in virtual reality whenever a person adopts an avatar that does not match their own proportions or lifts a virtual object that appears heavier than the movement of their hand.

Overall, such errors require a series of investigations. The ultimate research question for AR/VR is "How much is the influence of dynamical inconsistencies on the sense of presence for users?". To answer this, it is first necessary to investigate people's sensitivity to these inconsistencies. If they are easily detectable, what could be some artificial modifications, e.g., modifying animation speed, or adding sound effects, that would reduce people's ability to detect them? In this work, we perform a thorough investigation to understand people's sensitivity to dynamical inconsistencies under various conditions.

1.2 List of Contributions

1. A preliminary VR throwing implementation and perception study (Exp. T1) that explores the effect of visual trajectory cues on throwing performance in VR; we found that limited visual feedback detracts from virtual throwing performance and that throwing performance decreased with throw distance (see Chapter 3);
2. A second perception study (Exp. T2) to explore how timing errors in the Point of Release (PoR) of a ball can affect the perceived realism of throwing motions; we found that people are asymmetrically sensitive to early and late delays in overarm and underarm throws, with early release not as noticeable for underarm throws as it was for overarm throws, and late releases were less frequently noticed for overarm throws (see Sec. 4.1);
3. a regression model that predicts the remaining time until the PoR of a throwing motion in real-time (see Sec. 4.2);
4. a real-time VR system, called *ReTro*, that uses this model for throwing interactions using only the user's arm motion (see Sec. 4.3);
5. a ranking of 15 joint and motion feature combinations based on their effectiveness for the PoR prediction task, which can help to guide the selection and placement of sensors (see Sec. 4.3);

6. a corpus of 1679 full-body throwing motions from six actors including PoR ground truth approximations that will be made publicly available (see Sec. 4.2.1).
7. a series of experiments in which we investigated the perceptual impact of both the kinematic signal (i.e. the motion) and varied visual signals (the size of the avatar, the size of the lifted object, and the presence of muscle deformations that convey strain), where participants watched animated models perform dumbbell lifting and estimated the effort of avatars and the weight of the lifted dumbbells (see Chapter 5). We found that:
   - effort judgement is influenced by all channels (motion kinematics, the avatar's body, the size of manipulated objects, and muscle strain);
   - while effort judgement is mainly influenced by motion kinematics, weight judgement is more influenced by a lifted object;
   - muscle strain can be used as a tool to make incorrect motion kinematics less noticeable;
   - large inconsistencies between motion kinematics and lifted dumbbell appear unnatural.

1.3 Scope

In throwing, we investigate the importance of throwing trajectory feedback and PoR timing errors in two separate perceptual studies. The study on projectile trajectory feedback was conducted in VR with participants performing throws. The second study on PoR timing errors was conducted on a computer display in which participants watched an animated virtual character performing throws with different release timing errors. Because of the restriction period during COVID, we were unable to conduct the second experiment in VR. However, for the final evaluation, we built a real-time virtual throwing system that uses a machine learning-based model to detect PoR and ran a short, in-person, user study.

In the examination of lifting motion, we focus on understanding people's ability to estimate effort and weight when presented with visual stimuli of animated models lifting

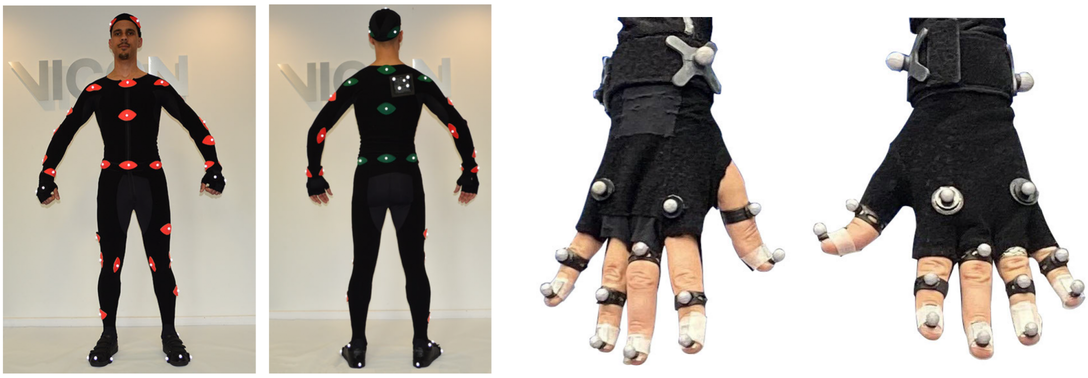


Figure 1.1: L: Body marker placements. R: Hand marker placements [1].

dumbbells under various conditions. We also explore whether people's estimations can be tweaked by adding skin deformations. Our scope does not include the experience of a person performing a dumbbell lift in VR. We also do not assess concepts such as the sense of presence or the sense of co-presence, which are left for future work.

1.4 Methodology

1.4.1 Optical Motion Capture

Research on human motion relies heavily on high-end capture systems as a tool for capturing high-quality motion data. We utilize an optical motion capture system by Vicon with 21 cameras, capturing at 120 Hz. Except for the data capture in Chapter 5, this is the setup used in the rest of the work. The actors wear a lycra suit and 53 reflective body markers and 20 reflective hand markers (2 per finger) (see Figure 1.1). Through these reflective markers, the infrared signals transmitted by the optical cameras are reflected back and tracked, and the position and orientation of the markers are calculated. In real-time, the cluster of body markers is solved to form the actor's skeleton, and the actor's motion is tracked.

1.4.2 Psychophysical Methods

In designing our user studies, we utilized multiple psychophysical methods. For discrimination tasks, we used Yes/No (sometimes also identified as two-alternative forced choice (2AFC), [22]) and two-interval forced-choice (2IFC) designs to learn the psychometric function between stimuli and response and to discover the threshold at which an effect tested by the stimuli is detected. In a Yes/No task, a stimulus and a question are presented to a participant with two possible choices to choose from, typically mutually exclusive answers such as a "yes" and a "no". In a forced-choice design, the participant is provided with a set of stimuli (simultaneously unless otherwise stated) and a question and asked to choose one stimulus from the set. 2AFC is considered the most basic form of a forced-choice design, in which the participant is asked to choose an answer from two stimuli. In other versions such as 2IFC, two stimuli are presented sequentially, and the participant makes a selection between the stimulus based on the question.

We also implemented several rating-scale tasks, asking participants to choose a numerical value, continuous or discrete, for the stimuli according to the underlying research question being investigated. The Likert scale, being the most widely used rating-scale measure, provides a symmetric agree-disagree scale, which we employed to measure the naturalness of stimulus, and to self-report on the sense of agency, and the sense of presence. Furthermore, we implemented effort and weight rating tasks to explore people's ability to estimate the weight of a lifted dumbbell and the percentage effort displayed by the lifter.

In the analysis of the data collected from these tasks, we use two different statistical models, ANOVA and Linear Mixed-Effect models.

1.4.3 Machine Learning

This work utilizes machine learning to model the release of a ball during a throwing motion. We focus on feedforward neural networks and long short-term memory (LSTM) networks.

Feedforward Neural Networks

Artificial neural networks are known for modeling complex functions by learning a large set of parameters that are optimized with respect to a loss function on large datasets. Operating on the principles of forward pass and backpropagation, a very large number of parameters can be learned iteratively and efficiently to achieve successful detection

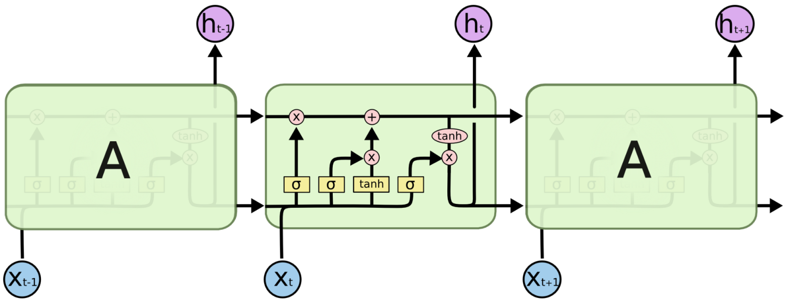


Figure 1.2: Visualization of an LSTM layer (Image taken from Christopher Olah's blog [2])

performance on unseen data. Also, the selection of a suitable loss function plays a crucial role in the outcome of the training. We use supervised learning by providing the network with both the output and the input to the modeled function during training.

Feedforward neural networks only flow information forward except for backpropagation during training. Therefore, these networks include multilayer perceptron (MLP) architectures, convolutional neural networks (CNN), and radial basis function neural networks. In this work, we only use MLP architecture in the feedforward neural network category. MLP architectures consist of stacked layers of fully connected artificial neurons. These neurons linearly combine the incoming signals and add a learnable bias term, which is then passed through a differentiable nonlinear activation function to become input to the next layer.

Long Short-Term Memory Networks

Long short-term memory networks are a type of recurrent neural network (RNN) that is used for modeling temporal data such as speech, text, and motion. Different from feedforward neural networks, these architectures maintain memory units that control the impact of past information (or even future information in the case of bidirectional architectures) on inferences about the current state. These networks operate on the same principles as feedforward neural networks, with the only difference being that backpropagation also has to be performed over time, hence it is named backpropagation through time (BPTT).

While a traditional RNN only combines the hidden state which carries information about the previous time step with the current input, LSTM networks include several subcomponents that facilitate the processing and passing of information from previous inputs. These subcomponents apply nonlinear operations on cell state, as visualized in Figure 1.2. The subcomponents are called forget gate, input gate, and output gate. The forget gate combines the hidden state of the previous time step with the input of the current time step, which is then passed through a fully connected layer with sigmoid activation to be scaled between 0-1, deciding which information of the cell state should be kept or discarded. Secondly, the input gate is responsible for calculating new cell state values that will be added to the previous cell state value. Lastly, the output gate sets the hidden state for the cell by applying *tanh* function to the cell state to map it between $[-1, 1]$ and filtering that through multiplication with a sigmoid, similar to the forget gate.

1.5 Overview

The structure of the thesis is as follows:

- Chapter 2 covers the relevant literature and discusses how our study fits in the available literature.
- Chapter 3 discusses a perceptual study on the importance of ball trajectory in VR throwing and a preliminary model for PoR detection.
- In Chapter 4, we share a perceptual study on release timing. We revise our Point of Release detection models and present a fully implemented system that detects virtual throws in real-time. A case study of the system with six participants is included.
- Chapter 5 delivers the set of perceptual studies conducted on dumbbell lifting motion and contains a discussion of the results.
- Chapter 6 summarizes the work and delivers a general conclusion. Limitations and possible future work are discussed.

2 Background

2.1 AR/VR

2.1.1 Taxonomy

In the past, there have been many attempts to introduce a taxonomy for all types of Human-Computer Interaction (HCI) devices. A very famous one is the 'Virtuality-Reality continuum' by Milgram et al. [23], which represents the real environment on one end and the virtual environment on the other end. In between, there are two segments, Augmented Reality (AR) and Augmented Virtuality (AV). All HCI technologies with a display are classified as one of those four options based on how many actual or virtual elements they use. AV represents the technologies that augment virtual environments with information from the actual environment, but there is not a clearly defined boundary between AR and AV. To go beyond the oversimplification of one dimension, Mann [24] proposed a two-dimensional representation with the emphasis that VR is not only about mediating a Virtual Environment to the user's eyes, but at the same time is about blocking the Real Environment from being perceived by the user. In this framework, he places the 'Mediality Continuum' on one axis and the 'Virtuality Continuum' on the other.

Speicher et al. [25] address the shortcoming of 'Virtuality-Reality continuum' in that it is only based on visual content, covering a limited portion of our reality. Furthermore, they interviewed AR/VR experts and reviewed studies to survey how MR is defined, finding that there is no universally agreed definition despite the popularity of the term. Skarbez et al. [26] argue that 'Virtuality-Reality continuum' is in fact discontinuous when only exteroceptive senses (sight, hearing, smell, taste, touch) are addressed by AR/VR devices. This is because even when using an ultimate display with all convincing exteroceptive signals, interoceptive senses such as the vestibular and proprioceptive senses will be controlled by the physical reality. Therefore, a fully virtual experience would require the immersion of the entire consciousness like in the movie *Matrix*.

Although having many conceptual similarities, AR and VR have been positioned uniquely in different fields. The emergence of both technologies has resulted in many new fora for scientists to explore them. Mel Slater, in his seminal survey on VR applications, thoroughly discusses the ways in which VR enhances a wide variety of application domains [27]. In a related survey paper on mobile AR, Chatzopoulos et al. [28] lay out the relevant applications and core technical components. While VR and mobile AR have reached a mature stage, AR headsets still need a lot of improvement before they become a common tool for everyday use. This longer lead-in time is mainly due to the technological challenges in stand-alone see-through optical devices and is the main reason we are focusing on VR for now. In Chapter 6, we discuss on a fundamental level how the knowledge gathered in this work transfers to AR.

VR offers a convenient platform for cognition experiments through its fully controllable immersive environments. This enables, for example, a fruitful ground for psychology experiments where people are exposed to various stimuli to better understand human nature and behavior in a reproducible and controlled manner [29]. In the field of medicine, VR interventions have been shown to be an effective tool in acute pain reduction [30] and surgical training [31]. Many other perceptual studies can be carried out thanks to the low development cost, maximum experimental control, high reproducibility, and effortless manipulation of stimuli.

2.1.2 Key Concepts

Immersion and presence are the two main concepts to describe the quality of a VR experience. Multiple definitions for immersion and presence have been formed over the early years of immersive technologies. Mel Slater defines immersion as an objective quality of a system's ability to immerse a user in an experience, and presence as the subjective feeling being present in the mediated reality [11]. In contrast, Witmer and

Singer define immersion as "a psychological state characterized by perceiving oneself to be enveloped by, included in, and interacting with an environment that provides a continuous stream of stimuli and experiences" [32]. VR research community is currently using Mel Slater's terminology. AR/VR has raised attention to further concepts such as presence, ownership, embodiment, agency, and co-presence that have been studied in philosophy, psychology, sociology, and cognitive science, which have always been part of the human experience but became a point of interest for computer scientists as MR advanced as a technology.

Sense of Presence

The sense of presence has been defined and operationalized in various ways over the past few decades [33]. Mel Slater defines it as "the sense of being there", and breaks it into subcomponents, place illusion and plausibility illusion [34], with the former referring to the illusion of being in a place, while the latter describes the illusion of events taking place in a VE. In [10], Lombard and Ditton use the words "perceptual illusion of non-mediation" to define presence. Witmer and Singer [35] describe it as "the subjective experience of being in one place or environment, even when one is physically situated in another". Following these definitions, many measurement methods have been developed [33, 36].

Presence is considered a defining characteristic of a VE, and it was assumed early on that presence could be the vital sense in unfolding the full potential of VR [37]. Therefore, the exploration of the factors that affect presence has been an important research topic. Usoh and Slater [38] have investigated these factors in two categories, internal and external, by separating the subjective elements that are about to a user and the objective features of a VR setup, respectively. Accordingly, the listed external factors are display resolution and overall immersion of the VR setup, interactivity, interaction devices and techniques, motion mapping, and responsiveness—most of which are the fundamental properties of our reality. Others have looked at how presence relates to various types of VR experiences including teaching and learning [39, 40, 41, 42], exposure therapy [43, 44], marketing [45, 46], and gaming [47]. Overall, these studies have reported a positive correlation between the sense of presence and positive outcomes.

Despite the empirical evidence showing a relationship between the sense of presence and user experience, there is skepticism towards presence measurement tools and a search for better tools [33, 36, 48]. Souza et al. [36] analyzed 239 user studies that measure the presence, finding 85.8% used subjective measures (self-report surveys), only 2.5% used objective measures (physiological signals), and 11.7% used a combination of them. The main skepticism toward self-report questionnaires is that they fail to capture the phenomenon as it happens since the users have to report it after, and it is ambiguous whether people have a shared understanding of the sense of presence. Regarding objective measures such as heart rate and arousal, it is considered problematic to attribute presence as the causal factor or the main contributor where the content itself can induce these states as well.

The sense of presence does not inhabit any social components. Social presence and co-presence are the two concepts that are brought in from fields of sociology and social psychology to describe the sense of being present with other people in a VE [49]. For a taxonomy, readers are referred to [50]. Although social presence and co-presence have been defined differently, they are being used interchangeably in VR research [39, 51]. Bailenson et al. [51] argue that the increase in collaborative virtual experiences requires a detailed understanding of virtual interactions, with the sense of co-presence as one of the constituents. The impact of avatar and behavior realism on co-presence has been studied multiple times in the past with contradictory findings, with some studies reporting a positive correlation [52], and others reporting no relationship [53, 54, 55]. In similarity with presence, co-presence is measured through subjective (self-report survey) and objective (behavioral, cognitive) measures.

Sense of Agency

A sense of agency has been defined in the fields of philosophy and cognitive science as the "experience of oneself as the agent of one's own actions—and not of others' actions" [56]. It is the sense that a person has of changing something in the external world: e.g., causing an object to move by picking it up or throwing it; or internally: e.g., thinking about a particular topic, or making a decision. There have been many psychological and neurophysiological studies that explore the sense of agency [57]. For example, the sense of agency for participants in one experiment was reduced when an unexpected sound was played when pressing a button, and when that sound was delayed [58]. In another study, both angular and temporal errors caused a lower sense of agency when using a joystick [59]. It has also been shown that priming, i.e., giving a participant a direct or subliminal hint about what the outcome should be before actually performing the action, increased the sense of agency when deciding whether their actions or a computer's caused a moving square to stop at a particular position [60].

In VR, the sense of agency has been considered to be closely linked with the senses of presence [61] and embodiment [9]. Recently, however, the sense of agency has been conceptually recast as being a distinct percept, based on the three key principles of priority (intention-action-result), consistency (outcome matches expectation), and exclusivity (no other cause except one's own actions) [62]. This allows the sense of agency to be explored as a percept in its own right in VR, distinct from others, and allows for manipulations based on these principles to study their effects in VR.

Studies exploring the sense of agency in Human-Computer Interaction (HCI) and games have found that when a computer assists a user in completing a task, e.g., controlling an on-screen object using a mouse [15], or controlling a character in a game [16], at some point the sense of agency is lost, even when the computer completes the task correctly. These provide the motivation to deliver a strong sense of agency in AR/VR by allowing a real person to throw a virtual projectile using their full body motion and to perceive that the physical simulation of the ball's motion meets their expectations, given the forces that they have applied.

Computer animation has also been used to explore the perception of a virtual human throwing a ball, where observers' sensitivity to manipulations of overarm and underarm biological throwing animations were explored [63]. Participants perceived shortened underarm throws to be particularly unnatural, and simultaneously modifying the thrower's motion and the release velocity of the ball does not significantly improve the perceptual plausibility of edited throwing animations. However, editing the angle of release of the ball while leaving the magnitude of release velocity and the motion of the thrower unchanged was found to improve the perceptual plausibility of shortened underarm throws. Clearly, the relationship between the thrower's motion and the resulting trajectory of the ball has a strong impact on the perceived plausibility of the interaction, but what will be the effect of the visibility of the projectile's trajectory when the person themselves is throwing the ball?

2.1.3 Interaction Systems

Interaction system plays an important role in shaping the performance and the experience of a user in any type of HCI. With AR/VR, HCI has taken a more natural and immersive form, amplifying the importance of the interaction system. Different from computers and mobile devices, user interaction in AR/VR stimulate a wider range of senses including immersion, agency, ownership, realism, and presence. Among the many components of an interaction system, we will address the ones that are specifically relevant to our study, which are input devices/modalities, interaction techniques and mapping, interaction visualization—or user representation. Readers are referred to [64] for thorough coverage of 3D user interfaces.

Input Modalities

Input modality, or input device, specifies how a user sends information to the system with the aim of interacting with it. Currently, the established modality of interaction in AR/VR is handheld controllers or physical hands. Although both modalities provide a degree of robustness, naturalness, and control, each has limitations. Researchers have been studying these two input modalities together and separately in various settings to understand how they shape user experience [65, 66, 67]. All of these studies commonly report that users achieve better performance in tasks using handheld controllers, but hand interactions induce more realism, presence, ownership, and enjoyment.

Moehring and Froehlich [67] investigated these two input modalities in more detail for interactions within the reach of the user's arms in a CAVE system and a head-mounted display, finding that even though controllers are faster and more robust, finger-based interactions are generally preferred. Seinfeld et al. [65] also included a custom keyboard in their comparison of interaction devices next to handheld controllers and hands, finding that controllers and hands outperform keyboards in both performance and embodiment. However, the performance aspect is highly related to the task performed, and keyboards typically outperform other input devices in text entry tasks [68]. Lin et al. [66] explored the same comparison in VR while also studying the effect of virtual hand sizes. They reported that despite worse performance in comparison with using handheld controllers, subjects preferred using real hands for increased realism and ownership. No main effect of hand size on user experience was reported. Lin and Jörg [69] further looked into the effect of visual hand appearance on the VHI using six different appearances of hand, including highly realistic, cartoony, and zombie hands, as well as a non-anthropomorphic hand: a wooden block. Leap Motion was used for hand tracking and interactions. They reported that the created illusion of ownership is the strongest with the highly realistic hand, and weakest with the non-anthropomorphic hand. Moreover, the appearance of the hand did not have an effect on the agency.

Argelaguet et al. [70] investigated the same problem as Lin and Jörg [69], again with Leap Motion, using a different set of virtual hand representations, i.e., a sphere, a simplified robotic hand, and a high-realism virtual hand. Contrary to their initial hypothesis and Lin and Jörg [69], they found that the abstract representation provided a higher sense of agency despite that the sphere only allows translational tracking. Furthermore, the sense of ownership depended on the virtual representation, and the virtual representations that are more morphologically similar induced a higher sense of ownership. Lougiakis et al. [71] compared a different set of virtual representations, i.e., a sphere, a handheld controller, and a hand, and using handheld controllers for tracking. The virtual representation had no effect on agency, and the virtual hand provided the highest sense of ownership. This suggests that the previous finding of Argelaguet et al. [70] about the better agency of abstract representations may be related to poor tracking performance by Leap Motion in tracking the detailed motion of fingers, leading to a more accurate tracking in the representation with less DoF.

More recently, Adkins et al. [72] conducted a between-subjects study on interaction devices (controller-based or hand-based interactions) and grasping visualizations (show or hide virtual hand while grasping) together in a more involving VR game rather than a short task, aiming to divert the focus of subjects from interaction to gameplay. Their findings are in support of what was found by other studies: ownership, realism, enjoyment, and presence are higher when hand-based interactions are used.

Interaction Techniques

In interaction design, interaction metaphors are commonly used to explain one concept in terms of another. An interaction metaphor controls the complexity of a system by providing the users with a framework that aids in transferring prior knowledge in an unfamiliar situation [73]. According to Erickson [74], "the purpose of an interface metaphor is to provide users with a useful model of the system". The two common interaction metaphors in VR are virtual pointer [75, 76, 77, 78] and virtual hand [79, 80, 81] metaphors, which are also categorized as egocentric metaphors [82]. Other less common interaction metaphors include Worlds in Miniature (WIM) [83] and automatic scaling [84].

In the ray-cast based interaction technique or the virtual pointer metaphor, a user uses a ray that is typically cast from their virtual hand or controller to the virtual space to select and manipulate objects in the environment. This interaction technique does not inherit the fundamental challenges of virtual hand interaction techniques. However, there are distinct challenges relating to both hardware and software, such as target occlusion, ambiguity, density, input device accuracy, and latency. Currently, most of the hardware-related challenges have been overcome, but there is no widely adopted interaction technique that solves all the software-related challenges. Numerous notable solutions have been proposed for target assistance [75, 79, 80, 85, 86, 87, 88, 89, 90, 91, 92, 93], which is the grand challenge in ray-casting based interaction.

With the advancements in tracking technologies, gaze tracking is on the way to becoming a common feature in head-mounted displays. In the first iteration of consumer VR devices, i.e., mobile VR, the head direction was used as an approximation of gaze, and it was used as the ray-casting tool since handheld controllers were not yet introduced. More recently, eye tracking has become a built-in feature in the new MR headset by Meta, i.e., Meta Quest Pro. Eye movement is a heavily researched topic that is informative about a person's attention [94], decision-making [95], cognitive state [96], and memory [94, 97]. To make use of gaze information to assist with the object selection process, Sidenmark et al. [98] proposed *Outline Pursuits*, a two-step selection process that involves the selection of a region and a gaze confirmation of the target object. In the initially selected region, a moving stimulus starts outlining each object, and the user confirms the object by gaze-tracking the stimulus that outlines the target object.

As AR/VR platforms became able to accommodate a more advanced and wider variety of applications that demand increased productivity, efficiency, and speed, task-specific interaction techniques are being introduced, e.g., 3D-tracked multi-touch tablets for design [99], pen-based interactions for spreadsheet manipulation [100], a combination of hand and pointer metaphors for immersive analytics [101]. Notably, Pham and Stuerzlinger [102] reported that their custom interaction pen outperformed typical handheld controllers in pointing tasks both quantitatively and qualitatively. These results are backed by the study conducted by Li et al. [103], in which tripod grip pen interactions outperformed controller interactions in two separate experiments of long-range and short-range interactions, suggesting that precision and dexterity of fingers should be utilized further in AR/VR interactions.

Interaction techniques that adopt the virtual hand metaphor use a one-to-one mapping of real-world actions, implementing fundamental interactions such as touching, grasping, pushing, and pulling, that operate within the reach of a user's arms. In virtual interactions, perceptual plausibility is cumbersome to achieve, requiring convincing physics-based simulations. As discussed before, real-time physics-based VR interactions are difficult: first, haptic feedback is what regulates the amount of force a user applies in real-life interactions, and the lack of it makes it difficult to estimate the amount of force to apply in virtual interactions; secondly, a detailed calculation of the exchange of forces is computationally heavy.

Holl et al. [104] have suggested using the Coulomb friction model as an approximation of the friction force created between surfaces to implement plausible physics-based interactions, including pushing, pulling, grasping, and dexterous manipulations. Kim and Park [105] focused on modeling the deformation of skin globally, by a weighted sum of the transforms, and locally, through sparse physics particles on the hand mesh. With this contact heuristic, they can reduce the computational load to a reasonable level and iteratively calculate the deformations. Liu [106] formulated this problem as an optimization task and synthesized hand motion using pose and desired trajectory priors.

More recently, Yang et al. [107] introduced Contact Potential Field, a learnable framework that models hand-object interactions using springs that attracts and repulses vertices on hand and object based on affinity. Hand-object interaction modeling is not only of interest to the AR/VR community, and it is also fundamental for HAR, robotic, and teleoperation, in which a detailed understanding of the physical scene is vital [108]. We will cover this subject in more detail in Section 2.2.2.

As the physics-based simulations of hand-object interactions are both costly and challenging, simple kinematic gestures are utilized heavily to support certain interactions. For example, in high-end AR/VR devices such as Microsoft Hololens or Meta Quest 2 (using the hand tracking features), users perform specific hand gestures, including tapping and pinching, to guide the manipulation of objects within their reach. In such interactions, typically, the user still makes virtual contact with the object, but the object is kinematic and the manipulation is triggered through the user's gestures. Despite being a natural and intuitive form of interaction, these interaction techniques can trigger fatigue and hinder accessibility and utilization of the virtual space [109]. Certain modifications that tackle these problems have been suggested, e.g., Go-go technique [79]. Accurate hand-tracking with off-the-shelf cameras has not been possible until recently, which is a factor that highly impacts the quality of experience in what is supposed to be a natural form of interaction. In the next section, we delve more deeply into gesture-based interactions.

Gesture-based Interactions

A gesture is an intentional or unintentional pose or motion of a person's upper limbs for non-verbal communication. In daily communication with one another, gestures are used habitually to express intention and add expressiveness. Because of this, gesture-based interactions appear as the most natural form HCI can take. In terms of knowledge transfer, the absence of gestures does not necessarily decrease the quality of the communication under normal conditions, e.g., phone conversations [110], but gestures are relied on when speech is not enough to follow the conveyed meaning, i.e., high noise [111]. A gestural-based interaction is therefore quite different since gestures play a primary rather than a secondary role, requiring users to be very conscious of what and how they perform. This highlights the difficulty of coming up with a natural and intuitive set of gestures that are easily memorable and performable by everyone [112, 113, 114].

Nielsen et al. [114] drew attention to this problem by designing the gestures with consideration to the system rather than to the user by coming up with a set of gestures that a system can easily decode, but this led to a bad interface, creating a set of gestures that are difficult to memorize, rationalize, and use effectively. This highlights the importance of user-centered interface design. Further studies have explored using psycho-physiological measures such as intuitiveness and comfort in the design of gesture vocabularies [115].

Despite great advances, gesture-based interactions are not yet able to match mouse and touch-based interactions on 2D displays according to the two-dimensional Fitts' law [116], as shown by Sambrooks and Wilkinson [117]. The well-known Fitts' law [118] postulates that the time to move to a target depends on the size of the target and the distance to the target. Initially developed for one-dimensional movements, many modifications have adapted it to two-dimensional and three-dimensional interfaces since then [119]. In another study, Jones et al. conducted a comparison of Leap Motion and mouse interactions on user experience and performance, finding that Leap Motion leads to worse productivity and user experience overall. It is important to note that none of these studies have performed the comparison in a three-dimensional environment, which is a difficult domain for the adaptation of Fitts' law, according to Triantafyllidis and Li [119]. As accurate real-time tracking of human motion and gesture recognition is a prerequisite for gesture-based interactions, we discuss these aspects in Section 2.2.2, in particular for the hands.

Interaction mapping

Interaction mapping specifies how a user's physical motion is translated into a digital environment and utilized as an input signal in a digital system. Steuer [120] defines it as, "the manner in which actions performed by users of interactive media are connected to corresponding changes in the mediated environment". The impact of interaction mapping is two-fold; first, it restricts the user to performing interactions in a certain way, causing certain physical involvement; second, as a consequence of the employed mapping, the user's altered input affects performance, in return affects the user experience, e.g., enjoyment, presence. The effect of interaction devices on user experience and performance has been conveyed in Section 2.1.3, which showed that controller-based interactions, albeit less natural, achieve better performance compared to the use of physical hands. Similarly, it is important to assess how and whether more natural interaction mappings can enable better AR/VR experiences.

Video games have been significant in the study of interaction mappings, as they can simulate complex alternative realities with varying degrees of abstraction, e.g., a virtual car controlled with a keyboard/mouse or a steering wheel simulator in a car-racing simulation game. Furthermore, video games are known to induce spatial presence by offering a partially immersive medium in which users can interactively influence the flow of events [120, 121]. Skalski et al. [122] proposed a typology of natural mapping consisting of four types: directional (e.g., keyboard arrows to provide directional control), kinesic (i.e., involvement of body without a realistic controller), incomplete tangible (e.g., Nintendo Wii controller), and realistic tangible (e.g., arcade games with gun controllers). They conducted two studies evaluating users' gaming experience in two types of games (driving and golf simulation) with different sets of controllers belonging to different categories and found that the controllers with higher perceived naturalness enhanced the spatial presence of users.

McGloin et al. [123] conducted a similar comparison between a handheld controller and a natural Wii controller in a tennis simulation video game, finding that natural mapping was a predictor for game enjoyment. Notably, they also reported that natural mapping was a predictor for the realism of graphics and sound, pointing to an underlying relationship between interaction naturalness and perception of realism. Shafer et al. [124] also investigated motion-based controllers (Wii, Kinect, and Sony Move), and looked at their impact on spatial presence, perceived reality, and enjoyment. The researchers found that the players' perceived reality influenced their spatial presence, and their spatial presence impacted their enjoyment. Differently, Shafer and Popova [125] studied three levels of naturally mapped interfaces, each from a category of Sklaski's typology. They found that in control mappings without an object in hand, i.e., the body as the controller, perceived interactivity had a significant impact on perceived realism and spatial presence, while in mappings including an object or a controller in hand, the perceived reality was the major predictor of spatial presence. Successively, spatial presence affected enjoyment.

As AR/VR came to fruition, researchers have started examining natural mappings in these platforms. Seibert and Shafer [126] compared the experience of gaming in VR versus on a computer. In the VR case, they utilized Razer Hydra, a dual-wielded motion-sensing game controller, which is an incomplete tangible mapping with respect to Steuer's typology [120]. The computer condition utilized a keyboard and a mouse, representing directional mapping. Simultaneous to the assessment of interaction methods, they compared the significance of display types for spatial presence and perceived controller naturalness. Researchers reported that the spatial presence was affected more by the controller type, i.e., Razer Hydra or mouse and keyboard, than the display type, i.e., HMD or computer display. Interestingly, users perceived the mouse and keyboard as more natural compared to Razer Hydra, emphasizing that a natural interface needs to be intuitive. Recently, Reer et al. [127] investigated natural mapping (VR versus computer) in the context of Self-determination theory (SDT), a theory for human motivation and psychological functioning, to understand game enjoyment. They hypothesized and affirmed that a more natural mapping like VR is positively related to two postulates of SDT, autonomy and competence needs, which in turn positively affects game enjoyment. Similar results were reported in other studies [128, 129].

The literature shows that more natural mappings lead to greater enjoyment and user satisfaction. However, as shown in the case of Razer Hydra, certain devices can reduce the naturalness of a mapping if users are unfamiliar with them. Most research on natural mappings has been conducted in games for user enjoyment, but there are many other aspects that are interesting and essential for VR. For example, the effect of natural mappings in the context of sports has not been studied in detail. Furthermore, most studies so far compare VR and computer gaming, but it would be more useful in the future to focus on comparing AR/VR specific mappings, such that human perception of three-dimensional virtual interactions can be explored more deeply.

Interaction Fidelity

There is no established objective measure to compare different interaction methods. Developed by McMahan [130], the "Framework for Interaction Fidelity Analysis" (FIFA) is a notable framework for evaluating different interaction techniques. In the most recent version of this framework [3], provided in Figure 2.1, an interaction is evaluated on three categories—biomechanical symmetry, input veracity, and control symmetry.

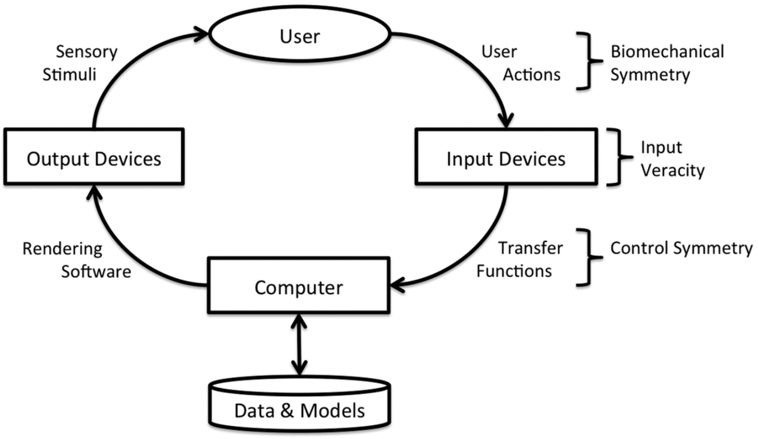


Figure 2.1: The User-System Loop and the three categories of revised FIFA, from [3].

Biomechanical symmetry is a measure of how much the virtual implementation of an interaction resembles the real interaction, e.g., throwing a ball using a 2-DoF mouse (Low biomechanical symmetry) versus throwing a ball using your body (High biomechanical symmetry). The input veracity component looks at how precisely an input device can capture a user's motion. For example, a wearable IMU sensor has lower tracking accuracy and higher latency compared to a motion capture system, and hence lower input veracity. Lastly, control symmetry refers to how much control an interaction provides to the user in a real-world task. As an example, they compare the selection and manipulation of a virtual object using two different techniques—a handheld controller that directly parents the virtual object with no positional offset and a ray-casting technique that does the same with an offset. While the former has high control symmetry, the offset of the latter is a limitation on the control, providing low control symmetry.

Following FIFA, McMahan et al. [3] pursued a case study of past research and categorized the adopted interaction techniques. They found that high and low-fidelity interactions lead to comparable performances, but medium-fidelity interactions result in poor performances. They argue that a low interaction fidelity (IF) resembles mainstream interfaces, and it is therefore easier to adapt to. High IF, on the other hand, is closer to how we interact with the physical world, making it natural and intuitive. This relationship between interaction fidelity and performance creates a U-shaped curve, which they interpret as the "uncanny valley of interaction fidelity".

Specifically for VR, Rogers et al. [131] investigated the effect of IF on player experience for object manipulation and whole-body movements. They found that, while high IF offers better object manipulation, whole-body movement interactions are not preferred at high IF, where people are more in favor of abstractions. These results seem to contradict the hypothesis of an uncanny valley for interaction fidelity.

Haptics Rendering

Haptics rendering refers to the simulation of the sense of touch over kinesthetic and tactile modalities. Kinesthetic feedback involves the sensation of force and torque by muscles, tendons, and joints. Tactile sensations by mechanoreceptors embedded in the skin include pressure, vibration, and shear [132]. These modalities have different utilities: kinesthetic feedback supports the estimation of force-related measurements such as weight, whereas tactile feedback aids in the identification of interaction material type. As haptic feedback is a factor in throwing, we include a brief review here, and refer the reader to a survey by Basdogan et al. [133] for more detailed coverage.

Haptic displays can be classified as *active* [134, 135, 136], *passive* [137, 138], or *hybrid* [139, 140]. In active displays, simulation is carried out by the actuators on the device. In stationary active displays (ie., the device is grounded), generated forces can easily be counterbalanced, allowing a wide range of forces to be simulated. PHANToM [134] is an early example that tracks and exerts forces on the fingertips, as is HapticMaster [135], which provides a high-performance interface using a grounded robotic arm. More recently, Siu et al. [141] used pin arrays to develop a tabletop haptic display. The pin of arrays can be programmed to take different shapes and act as both active and passive feedback. Furthermore, the display can be mounted on a robot that navigates next to the user. While stationary displays can accommodate more DoF and feedback sensitivity, they heavily restrict the motion of the user and are expensive and complex to develop. Handheld haptic displays are a low-cost, high-mobility alternative, albeit with limited capability, and include exoskeleton [139, 142, 143, 144], shape-changing [136, 145] and torque-generating [146, 147] displays.

Passive haptics is the technique of including physical objects or devices in virtual interactions to represent the haptic feedback of virtual objects. Passive haptic feedback can be implemented in the form of passive proxy objects around the user [137, 138], or in the form of wearable equipment [148]. It is shown in multiple studies that the inclusion of passive haptics in VE can improve user presence and performance [137, 149, 150]. Passive haptics is particularly powerful in the simulation of tactile feedback where actuator-based generation is too complex, e.g., hair. Carlin et al. [149] have simulated the haptics of a virtual spider using a toy spider during arachnophobia. In another study, the railing on a raised platform was utilized as passive haptics for exposure therapy on the fear of heights [151].

In the third category of hybrid displays, the device again contains actuators, but the actuators do not exert a force directly on the user. Zenner and Kruger [140] introduced the concept of Dynamic Passive Haptic Feedback (DPHF), where the actuators on their novel handheld display, *Shifty*, operate to manipulate the weight distribution of the device. In their evaluation of *Shifty*, they compared it against passive proxy objects and found that their device increases perceived realism and fun. In a different work, Zenner and Kruger [152] introduced *Drag:on*, another handheld DPHF device with haptic feedback based on drag and weight shift. Each of the five states refers to a specific shape, creating different weight distributions and different drag forces applied during motion. An exoskeleton hybrid haptic display is *Dexmo* [139]. As a user moves their hand in a VE wearing *Dexmo*, the exoskeleton activates force feedback when it detects a fingertip collides with a virtual object. The force feedback is limited to the locking of the finger cap, which simulates the feeling of touch at the fingertip.

Both active and passive haptics have their shortcomings: active haptic approaches generally involve complex hardware and software and are expensive; passive haptic approaches are typically application specific, so lack generalization and reusability. Hybrid solutions, e.g., DPHF, try to bridge the gap between active and passive haptics, but still cannot be generalized for a variety of actions. Despite many research efforts [132, 153, 154], the incorporation of haptic displays into mainstream AR/VR has been slower than for visual and auditory technologies. Common input devices, e.g., handheld controllers, cannot render kinesthetic effects such as virtual weight or inertia, and can only deliver limited vibrotactile stimuli. Consumer AR/VR requires cost- and energy-efficient, portable, non-invasive, and socially acceptable haptic displays, such as the bracelet- and sleeve-type wearables, such as those proposed by Pezent et al. [155] and Zhu et al. [156].

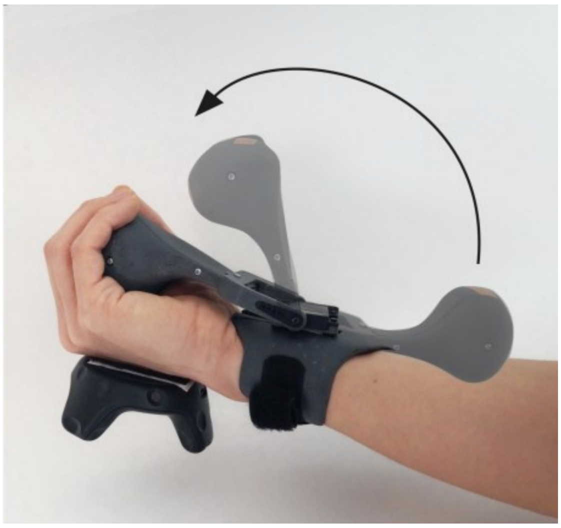


Figure 2.2: *PIVOT*, an actuated wrist-worn haptic device (from Kovacs et al. [4]).
.

Redirected walking is a well-known topic of research in VR locomotion where a user's perception is hacked in a way that their physical locomotion is not the same as their virtually displayed locomotion such that limited physical spaces can be used more effectively for moving around [157]. Several adaptations of this technique to haptics research have been made to improve the use of passive haptics [158, 159, 160]. These techniques are based on the evidence that vision dominates proprioception [161]. The potential improvements for passive haptics are two-fold: first, redirected walking can manipulate a user to think that a passive proxy object represents multiple virtual objects in the VE [160]; second, these exploits can loosen the spatial constraints on the design and the implementation of passive haptics. Zenner et al. [162] proposed a combination of DPHF with haptic retargeting to show how the use of haptic retargeting can be extended beyond passive haptics. The utilization of haptic retargeting has enhanced the capabilities of their DPHF device. Lastly, encountered-type haptics is a novel category that explores using drones [163, 164, 165], robots [166, 167, 168], and custom setups [169], utilizing the high mobility of these devices to simulate dynamic feedback to a user in VR.

Most relevant to throwing, Kovacs et al. [4] developed *PIVOT*, a wrist-worn haptic device that has an actuated pivoting elliptic handle for haptic feedback, shown in Figure 2.2. Allowing grasping, catching, and throwing, the device is also capable of rendering dynamic forces acting on virtual objects to the user, such as gravity and inertia. For example, when a user grabs an apple hanging from a tree in a VE, the device is able to apply a counter-directional force to render the resistance of the branch. When reaching for a virtual object in a VE, the pivot handle automatically approaches the user's palm for synchronized contact. When throwing, an object typically requires the person to open their hand to let go of the object, so the device uses touch sensors to release the virtual object from the hand. A user study on throwing shows that their device performs quite well, although participants who wore *PIVOT* for extended periods of time reported feeling numbness in their hands. However, the advantages of providing haptic feedback may outweigh these drawbacks. By providing a realistic sense of touch, users can gain the ability to exert realistic forces on virtual objects, which can enable fully physics-based interactions. Such a system would require a precise, real-time way of estimating or measuring the force exerted by the user, which is an ongoing research problem involving AR/VR, computer vision, and robotics [170, 171, 172, 173].

In a perceptual study, Villegas et al. [14] developed a training scenario to investigate the user experience of using real haptics vs. controller-based interactions. The users trained using their real hands and interacting with real objects. For the hand-held controller, an alternative training was provided, with no augmented real objects but only virtual elements. Their results showed that the feeling of presence and embodiment, as well as the performance, has improved with real haptics. With respect to lifting in VR, Gomez et al. [174] demonstrated that manipulating the animation of a self-avatar can generate different haptic perceptions.

Biophysiological Signals

The interest in improving the life quality of people with physical impairments has drawn research efforts into exploring the use of neural activities to simulate motions through actuating robotic extensions [175, 176, 177]. The goal of this type of research is to develop a mapping between neural activity and performed motion and to simulate this motion through actuated robotic extensions such that an impaired person can carry out an action cognitively without having to physically perform it. The studies mainly employ the tracking of brain signals through Electroencephalography (EEG) [175, 176, 177, 178], but they also explore tracking the activity of skeletal muscles through Electromyography (EMG) [179, 180], and hybrid solutions that include both [181, 182].

With the emergence of AR/VR, these methods are being investigated as alternative interfaces for more natural, non-obtrusive interactions [183, 184, 185]. In particular, EMG readings from forearm muscles are studied for hand pose and orientation estimation [186], gesture recognition [187], and force estimation [173]. Notably, Zhang et al. [173] introduced an interface that estimates the applied force per finger using forearm EMG. They simultaneously capture fingertip forces and skeletal muscle activity in the forearm and then train a supervised model to detect the fingertip forces from decoded EMG readings. They demonstrate their results on different actions such as pinching and pushing.

There are major challenges to these technologies that prevent them from becoming the mainstream input signals for AR/VR. First, the readings of electrical signals are highly susceptible to noise and interference, and therefore not easily decodable. Secondly, there is a large variation between humans, which makes generalization difficult. In EEG, the activity in the brain can change during the day for the same action, which makes data collection a challenging process [178].

2.1.4 Sports Training and Assessment

Professional sports is a highly competitive industry with high utilization of technology in every aspect, including performance assessment, coaching, simulation, injury rehabilitation, and many more. Computer vision has been one of the critical research domains, used for automated tracking of movement and action detection for performance assessment and broadcasting enhancements [188]. While earlier works relied on human supervision [189], recent methods can automatically extract player trajectories using a single camera [190]. Huge progress has also been made in action detection [191], which we cover in detail in Section 2.2.2. However, affordable vision-based solutions fail to provide detailed tracking of all players and may suffer from issues such as player occlu-

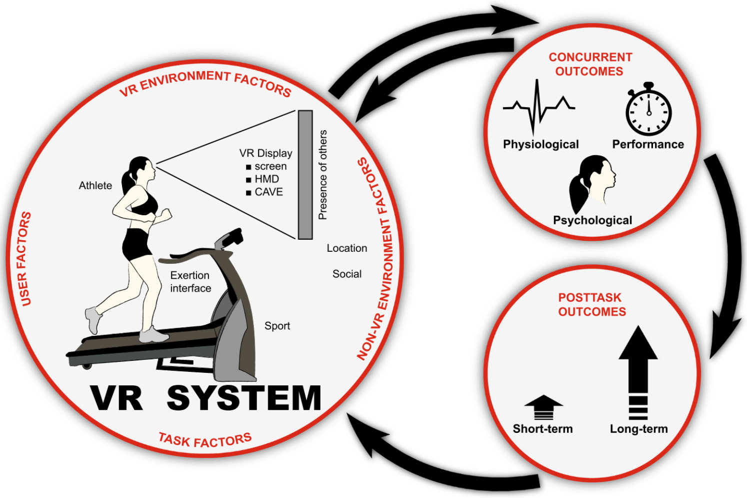


Figure 2.3: A conceptual model that shows the components of a VR sports task (from [5]).

sion, and field of view.

On the other hand, sensor-based tracking offers accessible, lightweight, affordable solutions for kinematics analysis in sports [192]. In these approaches, raw sensor readings from sports equipment [193] or players [194] are wirelessly communicated to a computer for offline or real-time processing using various methods such as CNNs [195], SVMs [196], HMMs [197], and DTW [198]. Please see [188, 192, 199] for a thorough review.

Sports training is another domain where technology plays an indispensable role through affordable and fully controllable digital training methods that facilitate skill acquisition. These methods range from video playbacks for decision-making training [200] to immersive VEs for motor skills training [201]. Regarding motor skills training, a systematic review conducted by Michalski et al. [201] showed the scarcity in the number of studies published that suitably evaluate the type of motor skills training performed. They excluded 34 out of 38 articles for not assessing the transfer of virtually acquired skills to the real world, which is reasonable given the fact that any virtual training is meaningless unless the skills can be transferred to the real world. This requires longitudinal, carefully planned studies such as [202], in which intermediate baseball players are adaptively trained for batting in a projection-based display, and then their statistics were analyzed in the following season, as well as the highest level of competition they reach in the following five years. The reviewed studies targeted beginner- and intermediate-level players, which raises the question of whether VR in the current fidelity that it can provide is good enough for expert sportspeople to benefit from. The scarcity of studies reported in the review by Michalski et al. [201] also highlights the complexity of setting up motor skills training studies, in terms of technology, design, and recruitment. Wood et al. [203] tackles one of the technological challenges by evaluating the construct validity of their training VE, which is "the degree to which the simulation provides an accurate representation of core features of the task".

In the category of perceptual and cognitive training, video-based methods have been repeatedly reported to be successful in skills development, and a detailed review can be found in the survey paper of Larkin et al. [204]. More immersive display technologies such as CAVE and VR offer more variety in the types of training facilitated. For example, Argelaguet et al. [205] explored sports psychology by developing a VE that seeks to induce competitive anxiety and pressure on trainees in a pistol shooting simulation, which may be impossible to achieve in a non-immersive technology. Lynch et al. [206] investigated the ability of rugby players to detect deceptive motions such as a side-step by displaying stimuli in a CAVE system. Other perceptual training formats for athletes include sports vision training for visual skills such as depth perception, and perceptual-cognitive training for perceptual-cognitive skills such as decision-making, and anticipation. Many survey papers successfully convey the current state of the field: Fadde and Zaichkowsky [207] draw attention to the importance of *deliberate practice*— a term introduced by Ericsson et al. [208] to formulate the type of specialized practice needed to achieve "expert" skills— and reviewed different video-based and VR-based approaches for perceptual-cognitive skills; Hadlow et al. [209] introduced a framework to collectively described different training methods (vision training and cognitive-perceptual training) and reviewed the emerging technologies with regards to this collective framework; Neumann et al. [5] reviewed interactive display technologies for sports using a broad conceptual model (see Figure 2.3); Appelbaum and Erickson [210] focused on sports vision training and surveyed the emerging technologies; Faure et al. [211] reviewed how VR has been used for sports training with a focus on team ball sports.

In relevance to our work on virtual throwing, Covaci et al. [19] developed a VR free-throw training system with feedback guidance. The proposed system utilizes a large immersive display showing a basketball court, in front of which a user stands and performs a virtual free throw using a physical basketball that has a restricted motion range. The motion of the basketball is tracked by a motion capture system and its trajectory is simulated in real-time and projected onto the screen. In this system, three different visual conditions are compared: first-person perspective, third-person perspective, and third-person perspective with trajectory guidance feedback modes, providing insights on the potential use of guidance feedback for training.

Regarding event detection in sports, Schuldhaus et al. [212] performed an offline analysis of shot/pass classification in soccer using inertial sensors attached to the legs of players. Their classification pipeline involved the extraction of peak detection from accelerometer data to locate events, segmentation, and feature extraction (mean, variance, skewness, and kurtosis) as data preprocessing, and model training (SVM, Classification and Regression Tree (CART), and Naive Bayes (NB)). A similar setup was used in tennis by Connaghan et al. [213], consisting of an inertial measuring unit (IMU) on a player's forearm for offline stroke detection. Different from these studies, our virtual throwing system performs real-time point-of-release detection.

2.2 Human Motion

The interest in a better understanding of human motion relates to most fields including but not limited to sports, entertainment, medicine, and business. Our work focuses on two specific case studies: ball throwing and dumbbell lifting.

2.2.1 Throwing

In this section, we cover the literature on throwing with a focus on overarm/underarm throwing in VR and PoR. The biomechanics of throwing is the assessment of the motion as throwing is performed [214] through various tracking equipment such as motion capture, video cameras, and wearable sensors. Research in this field mainly interests professionals that seek to improve and assess sports performance [215], avoid injuries [216, 217], understand phenomenons [218, 219, 220], and design equipment [221].

Proximal-to-distal sequencing (P-D sequence) is a widely known phenomenon in activities involving overarm throwing, which characterizes the generation of muscle activation and motion gradually from the more central to distal joints, e.g., starting with the shoulder, then the elbow, and finally the hand. Initially discovered by Herring and Chapman [222] in an investigation of simulating overarm throws, P-D sequencing is regarded to be developed as the most efficient way of throwing [223]. It is extensively studied in sports, including in baseball [224], javelin [225], American football [226], handball [227, 228]. There are several studies not conforming to this phenomenon [227, 228].

Critical to our work is the works of Hore et al. [218, 219, 220, 229] on the timing of finger opening in overarm throws. Following the work of Calvin [230] and Becker et al. [231] that suggest different timing windows for precise throwing, Hore et al. [219] investigated the capability of the central nervous system (CNS) in the control of release timing, and found that accurate and fast long throws require a release window precision of less than one millisecond. In their next study, Hore et al. [229] tested the hypothesis that proprioceptive feedback from wrist and elbow motion helps with the timing of the release, which they found was not the case. Hore et al. [218] followed this up with the proposal of an internal model for back forces applied by the ball to the fingers and gathered evidence that is consistent with the proposed model. Lastly, Hore et al. [220] investigated the precision of finger opening without the existence of a ball, finding that throwers were less rapid in performing it. This is a critical finding for our work on virtual throwing which is performed without the involvement of a physical object. However, all of the discussed studies on release timing have looked at fast, long-ranged throws, which falls outside of our scope for now. This is discussed further in future work (see Chapter 6).

Maselli et al. [232] investigated the predictability of target hit region in non-expert real throwing using kinematic cues. Using Principal Component Analysis (PCA) for dimensionality reduction and Linear Discriminant Analysis (LDA) for classification, they analyzed motion kinematics over various sizes of time intervals. They reported that while between-thrower style variations were present and heavily influenced the results, the outgoing direction prediction can be accurately performed as early as 400-500ms before the point of release. Interestingly, they found that the contralateral arm and lower limbs contained the most important cues for such early predictions. In another study, Maselli et al. [233] used the data from their former work [232] to examine the ability of non-experts to predict the future trajectory of a ball throw by intercepting it with a racket in VR, with one hypothesis being that the motion of throwing informs the observer and improves their interception score. Assessing three different conditions displaying ball only, thrower only, and both thrower and ball, they provided evidence of intrinsic knowledge of extracting action outcomes even in non-experts.

Zindulka et al. [17] conducted an experiment to evaluate how real throwing performance compares to throwing in VR. For virtual throwing, they employed the HTC Vive Controllers, through which the users have pressed a button to release the virtual ball. In the experiment, they evaluated the subjects in three conditions: overarm throw (vertical target), underarm throw (horizontal target), and overarm throw (far horizontal target). For each condition, a virtual replica of the real set-up was created. They reported that throwing in VR is half as accurate and one-third as precise as real throwing. They argued that the inability of the users to precisely time the release due to the VR controllers could be the reason for this. This is one of our contributions to the virtual throwing system that we suggest, which does not require any intermediary device such as a controller. In a similar work, Butkus and Čeponis [18] investigated virtual throwing with a focus on distance perception. Different from Zindulka et al. [17], a VR controller was not employed for virtual throwing, instead, a throw was initiated once the velocity of the hand, estimated through a Vive tracker, started decreasing during a swing motion. They concluded that people apply 3-5% more force when throwing in VR, but also communicated significant limitations such as a limited pool of participants.

2.2.2 Gesture and Action Detection

The advent of ubiquitous computing has amplified the importance of gesture, action, and activity recognition. A gesture is a specific type of physical movement (e.g., pinching motion), an action is a specific type of behavior (e.g., grasping), and an activity is a broader term that can encompass both gestures and actions (e.g., playing throw and catch). Our focus in this discussion will be on gesture and action, and readers are referred to [234, 235] for surveys on activity recognition. While gesture recognition is studied more in the context of HCI [236], action recognition scopes numerous fields, such as healthcare monitoring [237, 238, 239], sports performance assessment [212, 213, 240], smart environments [241, 242] and security surveillance [243].

Vision-based and wearable-based devices are the two main types of devices used for gesture and action recognition. Vision-based solutions are more accessible and already implemented in consumer VR, such as real hand tracking in Meta's Quest 2 headset [244], but they suffer from occlusion, poor lighting conditions, and privacy concerns. Wearable-based solutions, e.g., smart wristbands, do not have these restrictions since the tracking sensors are located directly on the body, however, they risk being invasive and motion-restricting. With wristbands and gloves being the more common wearable-based devices, other alternatives include surface electromyography (sEMG), motion capture systems, and various sensors [245]. sEMG is a promising technology that is being heavily researched, involving a set of sensors to be placed on the body to measure muscular activity. Some of the main challenges of sEMG devices, such as high noise-to-signal ratio and sensor crosstalk, selection of a suitable sensor setup, and user comfort, prevent the technology from fully taking off, currently limiting the use to prosthesis and rehabilitation [246]. Yet, Meta has publicly shared their interest and investment in this technology, and it is likely that wristband sEMG devices will complement, or replace, handheld controllers in the near future [247]. Last but not least, high-end MoCap technologies such as optical motion capture, e.g., Vicon, and multi-sensor data fusion, e.g., XSens, are very costly, and mainly used for capturing large amounts of data in film studios and research facilities. A real-time utilization of this data requires real-time streaming of the data to a separate computer.

In vision-based approaches, one or multiple cameras capture the upper body limbs of the user. Typically, the captured images are preprocessed to segment and extract informative features of the hands, e.g., a region of interest (ROI), and optical flow histograms, which are processed for gesture/action detection. Before the advent of deep learning, many different methods were explored, including HMMs [248, 249], SVMs [250, 251], DTW [252, 253, 254, 255], and many more [256, 257]. For example, Cabral et al. [258] designed an interface that the users can interact with using their heads and hands. In their processing pipeline, a face detection algorithm finds the head position and skin color, followed by the detection of hand positions using color segmentation. The introduction of commodity depth cameras, e.g., Kinect, was another leap forward in this field, accelerating research in gesture recognition [259, 260, 261, 262], action recognition [263], and hand tracking [264, 265]. Similarly, Leap Motion Controller (LMC), a sensor device released in 2013, enabled effortless hand tracking using infrared cameras and LEDs, paving the way further for the translation of natural gesture motion into computers [266]. This controller accommodated many gesture recognition studies: In [267], Lu et al. developed a dynamic hand gesture recognition system using a Hidden Conditional Neural Field (HCNF) classifier; Martin et al. [268] proposed a pipeline that implements multi-class SVM operating on data from both LMC and Kinect devices simultaneously; Chuan et al. [269] and Mohandes et al. [270] explored the use of LMC for an American Sign Language recognition and Arabic Sign Language recognition, respectively.

Deep learning has become the primary method for recognition tasks in both vision-based and wearable-based setups due to its exceptional performance in detection and estimation problems. Deep learning models are able to accurately approximate very complex nonlinear functions by hierarchically extracting low-level and high-level features from input data. The main limitation of deep learning is the requirement for vast amounts of data and computational power. However, for learning tasks that utilize image and video data, the prevalence of such datasets negates this limitation. Convolutional neural networks (CNNs) are prevalent in learning from image [271] and time series data [272]. Tran et al. [262] extracted hand contour and fingertip information using a Kinect camera to train a CNN to recognize seven complex hand gestures. Wu [273] proposed a double-channel CNN (DC-CNN) network that processes hand images and edge-detected hand images separately over several layers, connecting them at a fully-connected layer before classification. In [274], Mujahid et al. trained YOLOv3, a highly efficient and effective object detection model [275], for gesture recognition, and also evaluated it against several other popular object detection models, VGG16, SGD, and SSD. The sequential modeling capabilities of LSTMs, RNNs, and three-dimensional CNNs have also been explored. Hakim et al. [276] investigated spatiotemporal feature learning by combining LSTMs and CNNs in an architecture for dynamic hand gesture recognition. To enhance the utilization of temporal dynamics both ways, Pigou et al. [277] suggested an architecture with bidirectional RNNs. A detailed survey on vision-based approaches for gesture recognition can be found in [256].

RNN and LSTMs can struggle due to vanishing gradients in learning very long-term, which is required in fields such as natural language processing (NLP) and human motion learning tasks. Vaswani et al. [278] introduced Transformer, a parallelizable neural network architecture based on attention mechanisms that excel in learning long-term dependencies. Transformer architectures have been shown to perform better than LSTM and RNN architectures in comparative studies [279, 280]. Transformers have since been explored greatly, as discussed in detail in [281]. Also recently, Graph neural networks (GNNs) are receiving a lot of attention from researchers due to their ability to model complex forms of data that are structured as a graph [282]. GNNs are powerful in utilizing the node connectivity in data that is naturally structured as a graph, e.g., social networks and chemical compounds, but other data such as human motion can also be represented as a graph, with joints as nodes [283, 284]. In [283], Yan et al. developed spatial-temporal graph convolutional networks (ST-GCN) for skeleton-based action recognition, in which joints and joint connections represent nodes and *spatial* edges, respectively, and same-joint connections over consecutive frames represent *temporal* edges. Adopting this approach, Li et al. [285] developed a hand gesture graph convolutional network (HG-GCN) that can learn from a small dataset of hand motions and achieve fast, high-accuracy detection on two hand gesture datasets.

Although vision-based approaches receive more attention due to being low-cost, ubiquitous, and outside-in, wearable devices are generally more robust since they are not affected by external conditions, such as lighting and occlusion. Many alternative approaches are being explored in the research on wearables for gesture-based interactions. For example, triboelectric sensors utilize the triboelectric effect, a phenomenon in which certain materials become electrically charged when they are separated from another material with which they were previously in contact. These sensors are very low-power, as demonstrated by the study of Tan et al. [286], in which a self-powered, i.e., without needing an external energy supply, full keyboard is implemented on a novel gesture recognition wristband with triboelectric sensors, achieving a maximum of 92.6% accuracy. Similarly, Wu et al. [287] developed a self-powered smart glove for gesture recognition using triboelectric effects. Moreover, Moin et al. [288] introduced a small wearable biosensing system with low energy consumption that performs in-sensor adaptive machine learning of gestures, achieving a classification accuracy of 97%. Their method utilizes hyperdimensional computing (HD), a computational approach that is inspired by the way the brain processes information. It is based on the idea that the brain uses patterns of neural activity, i.e., high dimensional vectors, rather than scalar numbers, to perform computations. This approach has been studied in the literature as a way to improve computational efficiency and performance in areas such as text classification [289], speech recognition [290], EEG classification [291], and many more. For a thorough review of HD and triboelectric sensors literature, readers are referred to [292] and [293], respectively.

2.2.3 Change Point Detection

The examination of time series data to detect the occurrence of an abrupt event is called change point detection (CPD), with important applications in fields such as medical monitoring [294, 295], climate change detection [296, 297], seismology [298], smart homes [299], human activity recognition [300, 301], image analysis [302], and speech recognition [303]. Time series data analysis is a heavily researched topic that is foundational to the current state of technology [304], providing CPD research with numerous tools, including many supervised and unsupervised methods [305]. Aminikhanghahi and Cook [305] thoroughly reviewed the terminology, methods, and evaluation metrics, laying out several of the important challenges in the field such as algorithm robustness and handling non-stationary time series. In terms of using supervised methods, an additional challenge is annotating large amounts of data containing many different transitions, especially in fields that contain many class labels.

A fundamental difference between CPD techniques and our technique is the assumption that the two states before and after the transition follow different distributions, which is at the core of many introduced techniques in CPD research. In [295], Aminikhanghahi and Cook explore transition-aware activity segmentation from continuous readings of smart home sensors. They employ a non-parametric density ratio technique called SEP [299], which models the density ratio of two consecutive windows of a fixed size to decide whether a change point occurs, and use it to enhance activity recognition. Their proposed system requires readings from times (t+1) and (t+2) to make a decision for time t, referred to as 2-real time. In situations where sensor readings are not frequent, this may mean long time intervals. In our approach to release detection, we formulate it as a prediction problem and build up our knowledge from the start of the throwing motion. Bermejo et al. [306] tackled the same problem using an embedding-based approach, achieving better accuracy with a computationally low-cost algorithm.

2.3 Perception of Human Motion

2.3.1 Effort and weight estimation

There have been many studies on the perception of character motion. For example, Harrison et al. [307] investigated sensitivity to changes in limb length during animation; Reitsma and Pollard [308] explored sensitivity to errors in human jumping animations; Jain et al. [309] studied the ability to identify adult vs. child motion; McDonnell et al. investigated the perception of sex from walking motion [310]; Hodgins et al. [311] showed that anomalies in facial motion were more disturbing than in body motion; Hoyet et al. [312] demonstrated that people are sensitive to errors in pushing interactions, and earlier explored the perception of human motion when lifting different weights [313]. Other work explores the attractiveness of human motion [314] and how motion attractiveness impacts people's comfortable proximity to avatars [315].

Work examining interactions in VR includes Canales et al. [316], who compared different visualizations of hands for grasping tasks and found that if the hand was accurately tracked and allowed to penetrate objects being picked up, performance was better, but if the motion was adjusted to avoid interpenetration, users preferred it. Other work showed that when people were asked to adjust an avatar to match their body proportions, they underestimated weight by 10-20%, but other parameters were generally within +/-6% [317].

In a very relevant study, Kenny et al. [318, 319] explored people's sensitivity to mismatches between avatar body size and motion for pushing, lifting and throwing actions. Data was collected for two weight groups of male actors performing each action. Participants viewed the animations on avatars that matched or mismatched the size of the performer. They were not able to detect these mismatches, and naturalness ratings were not degraded, but participants did change their interpretation of the physical activity.

For lifting, heavier avatars were perceived to lift heavier objects, but motion had no significant effect, with similar results for throwing. For pushing, there was a significant interaction with motion, where light avatars animated with the motion of heavy actors were perceived as pushing lighter sleds than heavy avatars animated with light actors' motions. Notably, the objects the avatars were interacting with were not visible, which allowed observers to interpret the change in stimuli as reflecting a change in object properties (e.g., a heavier imagined sled). Such freedom is unrealistic in most practical scenarios, so in our work, we include visualization of the manipulated object.

Beyond computer graphics, there has been a strong interest in understanding people's ability to perceive dynamics from motion. Runeson and Frykholm [320] proposed the theory of kinematic specification of dynamics (KSD), "which states that movements specify the causal factors of events," citing varied evidence of people perceiving varied dynamic quantities (e.g., the mass of colliding objects [321]). Gilden and Proffitt [322] counter-argue that people use heuristics to make these judgments, and they are only accurate for a "single dimension of information" (particle motions). Blake and Shiffrar [323] suggest both motion and human form are important for judging actions.

Starting with the pioneering work of Runeson and Frykholm [324], much of the research on the perception of dynamics from human motion has focused on a box-lifting task and employed point-light displays (video that shows only points, normally at joint centers), with one study substituting a dumbbell lift [325]. Studies consistently show that people are able to estimate the weight of the lifted box [20, 21, 324, 326, 327] or dumbbell [325]. The accuracy of their estimates varies widely, from near perfect correlation between actual and perceived weight (e.g. the female actor in [324]), or correlations of .9 in ideal conditions [327], to much lower accuracy in other studies ([327, Exp. 5], correlations below .5 in [21]). One paper found people were less accurate below 30lbs [20] and another that they overestimated light weights, and underestimated heavy [21]. The lifting phase of a lift and carry motion is sufficient to make the judgment [326].

Variations of the study design that improve estimates include: showing a reference lift of a specified weight [324, 325, 327], although this acts as an attractor [20], not telling the actors the weight of the box [324], having people perform their own max lift to gain haptic experience [325, 327], knowing the size of the lifter [20], rating a single lifter at a time [21], and using average strength actors [327]. Video outperformed point-light displays in [328], suggesting there is important information beyond kinematics, something we explore by adding muscle strain cues.

Studies also observed kinematic changes that correlate with weight, chiefly object velocity decreases with weight [21, 325, 326, 328], although not with all actors [21]. Dwell time at the start of lift, hip angle [326] and max trunk velocity [21] also vary. Manipulating kinematic patterns can change weight perceptions [326, 327]. However, it was insufficient to show only the motion of the box [327, 328] or one degree of freedom movement of the elbow for dumbbell lifts [325]. Neck strain in a filmed pilot study was mentioned as a cue by participants [324], something we simulate. Finally, there is a response in the motor cortex that is consistent with force cues from kinematics and from hand contraction state, which manifests visually in the color and deformation of the hand [329]. Other work also suggests the motor cortex may be active in motion perception [330].

Grierson et al. [328] found that box size information does not confound the perception of lifted weight for point-light displays that indicate box size, but people thought a small box weighed less with a video presentation. Gordon et al. [331] found people scale motor programs for a lighter weight when lifting a smaller box, but after the task, estimate that the smaller box is heavier. Visual cues appear to be integrated into the programming of manipulative forces during precision grip.

Some studies asked viewers to estimate effort as well as weight. Shim et al. [21] found participants made fewer errors estimating lifters' effort, but if people knew a lifter's size and weight before making judgments, their estimates of weight and effort became comparable. An unexpected result showed participants judged heavier lifts of a much stronger and larger lifter as being lighter to those of a normal, weaker lifter moving less weight. Since ordinal judgments were correct, they suggest observers may be more attuned to effort than weight [327]. Other work explores how physical interactions can imply the presence of an unseen object [332].

2.3.2 Perception of Physical Interactions

O'Sullivan et al. [333] explored the factors that affect a user's perception of a collision in a real-time simulation. In such systems, an approximately accurate result is usually more acceptable than the delay caused by calculating a physically accurate response. They determined that delayed collision responses and angular and momentum distortions of an object's trajectory affect the perceived plausibility of such simulations, and that viewpoint also plays a role. Hoyet et al. [312] also found that errors in timing, forces, and incorrect angles can reduce the plausibility of physical interactions between virtual people.

Vicovaro et al. [63] tested user sensitivity to manipulations of overarm and underarm biological throwing animations. Participants perceived shortened underarm throws to be particularly unnatural, and simultaneously modifying the thrower's motion and the release velocity of the ball did not significantly improve the perceptual plausibility of edited throwing animations. However, editing the angle of release of the ball while leaving the magnitude of release velocity and the motion of the thrower unchanged was found to improve the perceptual plausibility of shortened underarm throws. These studies have motivated the selection of the factors we wish to explore in our PoR experiment.

Nusseck et al. [334] investigated people's ability to rate a bouncing ball's elasticity and predict its future position. They found that people employ different heuristics in solving these tasks, which suggests the provided information is critical in how people approach such tasks and come up with heuristics.

3 Virtual Throwing: Initial Exploration

This chapter describes the studies conducted in the early stages of the work that have been insightful in making important design decisions for the later work. This includes an experiment investigating the effect of visual trajectory cues on the accuracy of virtual throwing, and a first attempt at PoR detection using feedforward neural networks in a supervised manner.

Throwing is one of the rudimentary actions that humans learn to perform as infants. It is also integral to many sports games and it has even been suggested by some researchers to have a key role in human evolution [230]. We consider throwing to be an important inclusion in AR/VR as a virtual interaction for fields such as entertainment, sports training, and even rehabilitation in certain situations.

3.1 Exp. T1: The Effect of Visual Cue on Virtual Throwing

Despite rapid developments in AR devices, their field of view (FOV) is still much lower than for VR headsets (e.g., the diagonal FOV of 52° for the Microsoft Hololens™ vs. 113° for the HTC Vive™ Pro 2). This reduction in visual feedback can be problematic for certain tasks, such as ball throwing. We present an experiment in VR, where participants threw a virtual ball at virtual targets, with different levels of visual feedback. The objective of the experiment was to investigate how visual cues affect the way that participants perform virtual throws, which may be useful for the design of AR/VR systems. Eighteen participants used their own body motion to throw a virtual ball at virtual targets and we simultaneously captured their full body motion (MoCap) using an optical motion capture system for offline analysis.

Previously, as we discussed in detail in Section 2.2.1, Zindulka et al. [17] found that people are less accurate when throwing in VR than for real throwing, while Butkus and Ceponis [18] found that throwing accuracy in VR increased with distance and that throwing velocity was higher in VR than in reality. These studies used a device to control the ball, whereas, in our study, we use VR gloves to emulate more closely the experience of a real throw. Nusseck et al. [334] demonstrated the difficulty of predicting the properties of a bouncing ball during manipulations of the trajectory's visibility. We also vary the visibility of a thrown ball's trajectory in VR.

As a first study, we developed a VR experiment where participants used their full bodies to throw a virtual ball at virtual targets. We varied the amount of visual information about the ball's trajectory and found that this visibility was an important factor in their performance.

3.1.1 VR Implementation

The Virtual Environment (VE) was displayed using an HTC Vive™ headset and created in Unity. The VE contained a block with the virtual ball on top and a 50cm diameter target on the ground (see Figure 3.1). Participants entered the VE standing next to the block, facing the target region. The block was on the same side as the participant's handedness. We aimed to generate interactions that felt natural, created a sense of agency, and maintained focus on the task. To vary the target distances, we divided the floor surface into three regions 2m wide and 75cm deep, starting at 1.25m from the participant, and separated from each other by 1.25m. During development and testing, we observed that it was difficult to throw the ball very far, so we reduced the furthest point of the *Far* region by 50cm. A floor plan of the virtual environment is provided in Figure 3.2. Overall, the minimum and maximum intervals of the distances a region can contain between a thrower and a spawned target are as follows: Near region [1.25m, 2.24m], Medium region [2.50m,3.40,m], and Far region [3.75m, 4.12m]. All target positions were generated randomly within each region in real-time for each participant.

We implemented two tracking types: i) an HTC Vive controller was attached to the throwing forearm for *real-time* arm tracking. Manus VR gloves provided real-time fin-

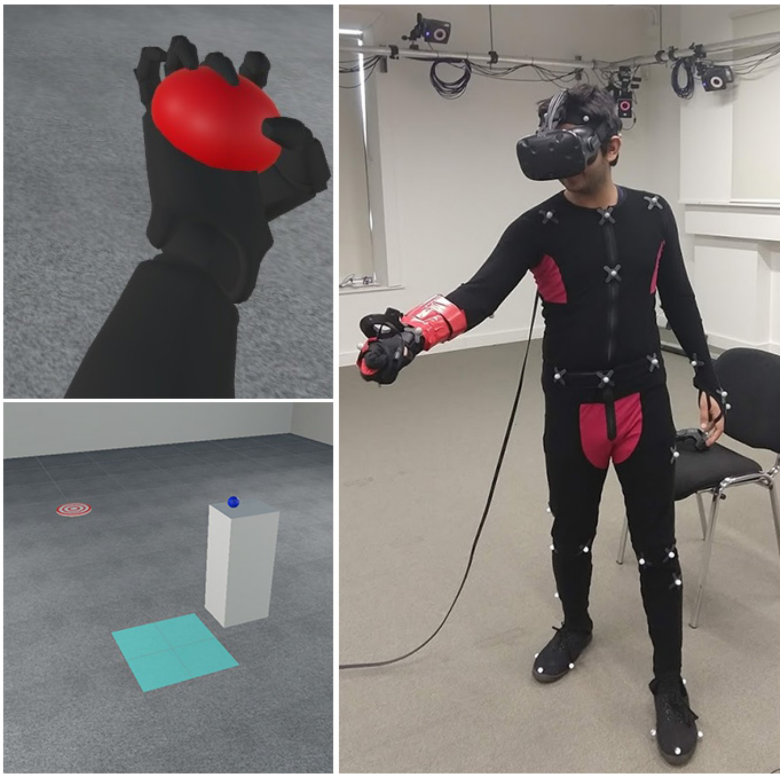


Figure 3.1: L: VR scene where users performed virtual ball throws during data capture; R: Equipment used in the VR capture. Motion capture suit, a Manus VR glove, and an HTC Vive controller

ger tracking through 10 sensors that measure the fingers' proximal and intermediate phalanges and this information is wirelessly transmitted to Unity to control VE interactions; and ii) we also captured participants' full-body motion for later *offline analysis* with a 21-camera Vicon optical motion capture system and 53 body markers at 120Hz (See Fig. 3.1). To synchronize, we established a connection between the VR computer and the motion capture system by sending a timecode via the Vicon Datastream Unity SDK when the virtual ball is released, allowing the PoR in the motion capture data to be located.

We implemented two algorithms for grabbing and releasing (throwing) the ball. The grab mechanism was initiated when two conditions were met: i) index and middle finger phalanges made contact with the ball, which was tracked using the primitive colliders on the hand and ball objects, and ii) index and middle fingers were flexed to the point of

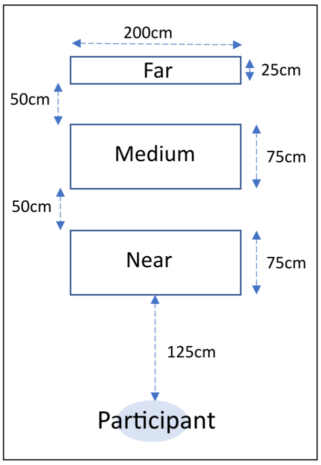


Figure 3.2: Floor plan of the virtual experiment space displaying the sizes of target spawn regions (Near, Medium, Far) and the participant's standing position.

a ball grab. The flexion amount was set before the capture for each participant by asking them to hold a real tennis ball with the VR gloves as we recorded the finger flexion data. Once the virtual ball was grabbed during the capture, its position was interpolated to the predefined center of the hand and it followed the hand during the throw motion up until the PoR. Between the grab and release, a velocity estimation algorithm continuously calculated the average velocity over a window of the nine previous frames, which was applied to the ball at the frame it was released.

A realistic implementation of the release mechanism is very critical in shaping the expectations of the user and satisfying their intentions. In reality, a ball release happens naturally, and the forces exerted between the ball and the palm occur without much conscious mental effort. The details of this force exchange are calculated according to the various properties of interacting surfaces such as friction coefficient, mass, and velocity. For a virtual ball release, not only do we not have access to all this information, but we also do not have the hardware to instantly measure, compute and simulate the throw with all these details. Therefore, in order to implement a natural and simplified release algorithm, we decided after many trials to use a rotational rate of change in index and middle finger phalanges of 3°/sec as a threshold, which provides a good indication of an opening hand. Once this threshold is passed, the ball is released from the hand. In the frame that the ball is released, we fetch a timecode from the MoCap computer to later pinpoint the release in the motion data. We treat this timecode as the ground truth PoR for our motion analysis (although it should be noted that this is not the true ground truth, but rather a heuristic).

3.1.2 Method

Eighteen participants (10F, 8M, aged 18-38) were recruited from University students and staff. Self-reported familiarity with VR was low on average. Each participant wore the gloves, tracker, and motion capture suit, and following calibration, they each performed a total of 63 virtual ball throws (out of which only the first 21 were used for the hypothesis testing) at targets of random distances, giving an overall total of 1134 throws for all participants. There were no restrictions or guidance on how to perform the throws other than asking the participants to stand inside a square displayed at their feet so that they maintained their distance from the targets. They each used their preferred hand for all throws. Participants were allowed to repeat the throw if the ball dropped unintentionally, which was a common situation in the early throws of the capture. Furthermore, some participants found the task easier than others, which led to some noise in the data. Visualizations of the data can be found in the Appendix.

When a target was hit or missed, the target turned respectively green or red. Two levels of visual feedback *Mode* were presented: (i) in Full mode, the ball was visible throughout the full trajectory of the throw; and (ii) in Minimal mode, the ball was only visible until it was grabbed, after which it would gradually fade in the participant's hand and remained invisible for the duration of the throw. Therefore, they only knew whether they hit the target or not when it changed color. To vary the *Distance* of the targets, we divided the floor surface into three regions with a width of 2m, referred to as *Near* (1.25-2m), *Mid* (2.5-3.25m) and *Far* (3.75-4m). The depth of the Far region was reduced in size during testing, as it proved to be very difficult to hit any targets beyond that distance. Targets were spawned randomly within these regions at run-time.

Wearing the headset, gloves, tracker, and mocap suit, 18 participants (10F, 8M, aged 18-38) each performed a total of 21 virtual ball throws to targets at random Distances in one of the visual Modes. They were instructed to stand inside a square at their feet in order to maintain their distance from the targets. Each participant performed 21 throws:

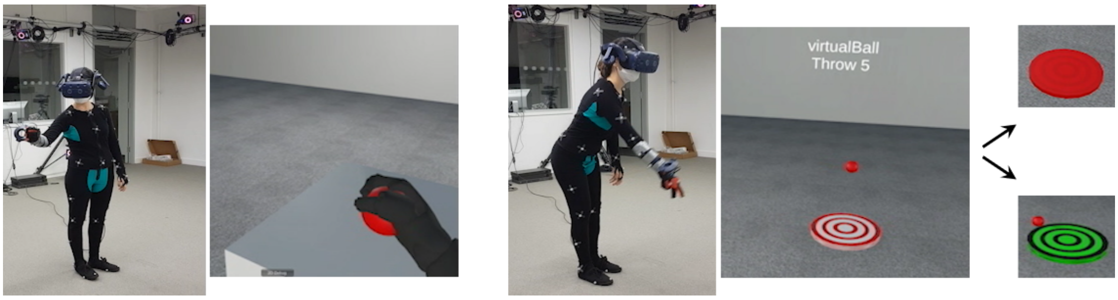


Figure 3.3: VR experiment: a participant grabbing (l), throwing (m), and receiving visual feedback (r)

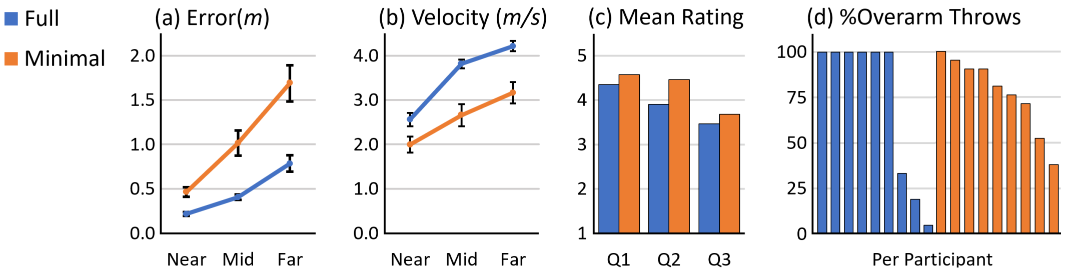


Figure 3.4: Results from the VR experiment (see Table 4.1 for all significant effects).

1 Mode (Full or Minimal, in counterbalanced order) at 3 Distances (Far, Mid, Near), with 7 repetitions of each condition. All conditions were presented in randomized order

3.1.3 Results and Analysis

We conducted a mixed repeated measures Analysis of Variance (ANOVA), with between-groups categorical predictor Mode(2) and within-groups independent variable Distance(3). The dependent variables were throw *Velocity* (meters/second) and throw *Error* (distance in meters of the ball's first-floor contact from the center of the target). Post-hoc analysis was performed using Bonferroni tests. The results are presented in Table 4.1 and Figure 3.4(a,b).

Post-hoc significance testing of the Mode effects revealed that Error was higher and Velocity was lower in the Minimal mode (i.e., Minimal $<$ Full); and the Distance effects show that both Error and Velocity increased with distance (i.e., Near $<$ Mid $<$ Far). However, the interaction effects (Mode X Distance) show that there is no significant difference between Full and Minimal at the near distance. In Full mode, Error is largest

Table 3.1: Significant effects ($p < 0.05$) for ANOVA with effect sizes (partial η^2)

Effect	**F**	p	η^2
***T*HROW *E*RROR**			
Mode	$F_{1,16} = 20.4$	$< .0005$	.56
Distance	$F_{2,32} = 67.9$	$< .00005$	.81
Mode × Distance	$F_{2,32} = 9.1$	$< .005$	.36
***T*HROW *V*ELOCITY**			
Mode	$F_{1,16} = 17.0$	$< .005$	.52
Distance	$F_{2,32} = 89.0$	$< .00005$	.85
Mode × Distance	$F_{2,32} = 4.3$	$< .05$	.21

at the Far distance, but is not significantly different for Near and Mid, i.e., (Near $\approx$ Mid) $<$ Far), whereas Velocity is lowest at the Near distance, but does not differ for Mid and Far (i.e., Near $<$ Mid) $\approx$ Far). Nevertheless, our results confirm that reduced visual feedback results in increased errors and decreased velocity. Our finding that error increased with target distance is different from previous results in virtual throwing [18] with a controller, but is more consistent with the real world and suggests that using one's own body and finger motions may be important when trying to emulate real throwing.

At the end of the experiment, participants were asked to rate their agreement or disagreement with the following statements on a Likert scale from 1-6: Q1: *I felt like the arm belonged to me;* Q2: *I felt like I was in control of the ball;* and Q3: *I felt like the ball behaved as I expected.* We present the average ratings in Figure 3.4(c). The low sample size for the between-groups factor Mode did not show any significant differences between Full and Minimal modes. Participants agreed the most with Q1 and the least with Q3. As ratings were recorded only once at the end, a better future option would be to increase participant numbers or to collect ratings more frequently during the experiment.

Finally, we processed the high-quality full-body motion capture data for follow-on analysis. We fitted a skeleton to the cluster of markers to locate body joints. Both joint positions and rotations were recorded and the correlation between the velocity calculated from this data and that calculated at run-time was high (>0.8, p<.05, N=1134), indicating that acceptable tracking accuracy was achieved during the real-time experiment. We also counted the percentage of overarm and underarm throws for each combination of factors and found that people use overarm throws more often than underarm, but that they use them less often in the absence of visual feedback. (See Figure 3.4)(d).

Our results demonstrate that limited visual feedback detracts from virtual throwing performance, as the throwing error increased and the velocity decreased when minimal visual cues were present. However, the results of self-report questionnaires did not demonstrate improved plausibility or immersion, which needs further exploration. We also found that error increased with throw distance, unlike previous results with a controller, which suggests that emulating natural hand/ball interactions may result in a more realistic experience. Our results may provide insights for the design of new experiments in AR/VR, and motivate further studies to explore factors that contribute to plausible and immersive physical interactions, such as accurate/plausible physical simulation of interacting objects, object and surface deformations, haptic feedback, and visual appearance.

3.2 Detection of Point of Release

Following the experiment, we used the captured data in our first attempt at PoR detection by training a binary classification model. Our approach is summarized in Figure 3.5. In a typical immersive VR gaming setup, using a VR headset for display, a data glove, and a single tracking sensor to trigger the throw of the virtual ball, 18 participants used their full-body motion to throw a virtual ball at virtual targets while fully immersed in a VE. We synchronize the mocap data with the real-time PoR using timecodes sent from the VR engine. We trained multiple frame-level binary classifiers to detect the Point of Release (PoR) of the ball based on different combinations of the Vicon motion features (rotation, position, linear and rotational velocity) of the main arm joints (wrist, elbow, and shoulder).

From our results, we are able to identify which are the most promising joint-feature combinations for identifying the PoR in real-time (Section V). This can then provide guidance on where to place a small number of low-cost wearable sensors or markers that can be tracked in real-time and provide plausible results. For example, we found that the wrist is the most informative joint and rotation is the most informative motion

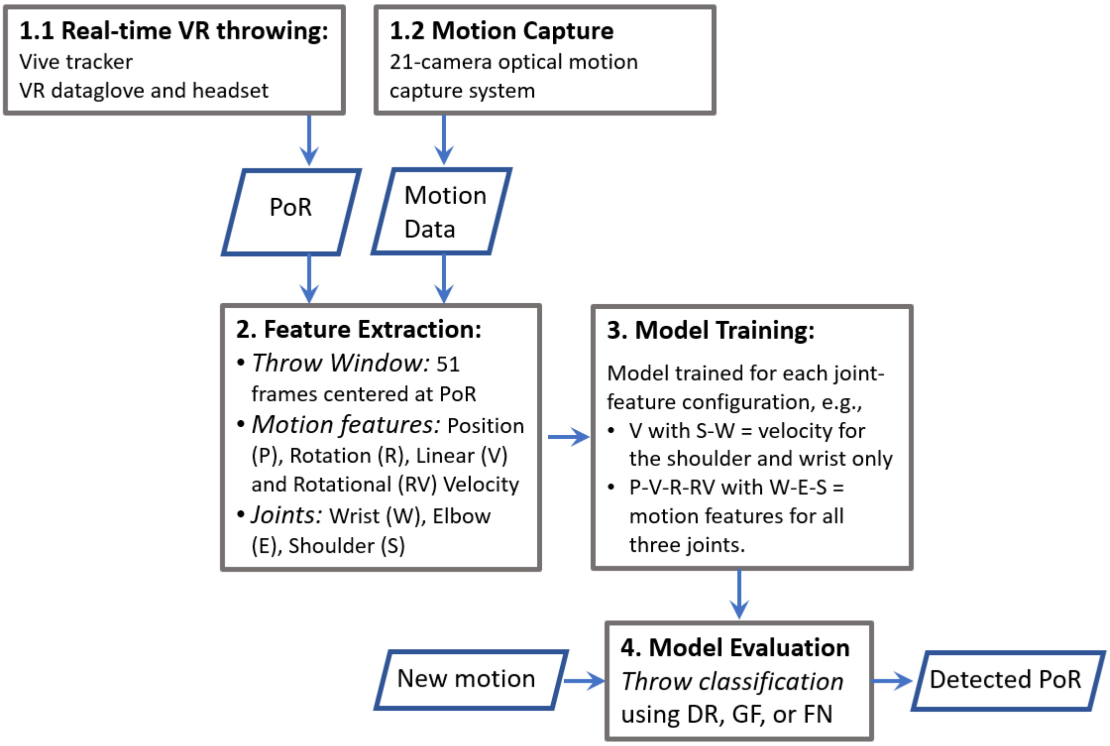

1.1 Real-time VR throwing:
Vive tracker
VR dataglove and headset
1.2 Motion Capture
21-camera optical motion capture system
PoR
Motion Data
2. Feature Extraction:
• Throw Window: 51 frames centered at PoR
• Motion features: Position (P), Rotation (R), Linear (V) and Rotational (RV) Velocity
• Joints: Wrist (W), Elbow (E), Shoulder (S)
3. Model Training:
Model trained for each joint-feature configuration, e.g.,
• V with S-W = velocity for the shoulder and wrist only
• P-V-R-RV with W-E-S = motion features for all three joints.
New motion
4. Model Evaluation
Throw classification using DR, GF, or FN
Detected PoR

Figure 3.5: Overview of our approach. We extract joints and motion features from high-quality motion data and use them to train binary classifiers to detect the Point of Release (PoR) of a thrown projectile from limited real-time data.

feature for detecting the point of release of a throw.

3.2.1 Throw Detection Model

In this section, we describe the steps performed to develop our proposed throw detection model, including data preprocessing, motion feature extraction, and model training. The dataset consists of high-quality motion data and release timecodes of 1134 virtual throws. We train a frame-level binary classifier that uses sliding window analysis to detect the PoR frame based on the estimated probabilities inferred from the motion feature input. The four motion features we use are Position (P), Velocity (V), Rotation (R), and Rotational Velocity (RV). The classifier is supervised based on PoR timecodes. In this study, we limited the scope to exploring the impacts of joints and motion features, although many other features could be explored as a follow-on.

Data Preprocessing

The motion capture system automatically fits a skeleton in real-time to the point cluster of marker positions. The skeleton consists of a hierarchy of 30 joints with the hip joint as the root node. The skeleton is unique for every participant as their body and limb sizes can vary. We read joint positions and rotations from the output motion files [335]. To locate and extract throwing motion sequences, we center a window of W frames, called the *throw window* (see Figure 3.6), at the PoR frame of each of the 1134 throws. We set W to 51 frames (425 milliseconds) to include the most important phases of a throwing motion and remove the redundant data where the participant was idle. The cleaned dataset included $W * 1134$ frames of human motion data.

Next, we transform the joint rotation representations of the throw data into quaternions since they are provided as Euler angles. Euler angle parametrization is not ideal for machine learning due to discontinuities and singularities. Despite having one additional parameter than the Euler representation to encode the rotation, their advantages of yielding numerically stable systems outweigh this disadvantage (see [336] for a thorough review of rotation parameterizations).

Position P is originally measured with respect to the center of the motion capture space, so we convert it to the relative position by changing the origin of the coordinate system to be the hip center of the participant. We also calculate the position of the arm joints projected onto the direction the thrower is facing. Thus, we remove any variability introduced by the orientation of the thrower and improve the generalizability of our ML model. To project the position locally, we first identified local axes and then projected global position to each local axes, i.e., dot product. In our identification of the local axes, we assumed that local y-axis is the up direction, i.e., (0,1,0); x-axis is the vector from the left shoulder to the right shoulder; and z-axis is the cross product between x- and y-axis, i.e. cross(x,y).

In the VR capture, participants were instructed to move their lower body as little as possible to maintain a distance from the virtual targets, so only the upper body contained relevant information about the throw. To further lower the number of dimensions in the data, we used only the three joints along the throwing arm, i.e., wrist (W), elbow (E), and shoulder (S).

Velocity V is estimated using the finite-difference method on positions, i.e., $V_f = (P_f - P_{f-1})/\Delta_f$ where f is the frame index, Δ_f is the time between frames and is a constant $= 1/120$ seconds (8.3 ms). We calculate Rotational Velocity RV using the angle-axis representation of rotation data, where rotation in 3D space is a unit vector representing the direction and a scalar θ represents the rotation magnitude. RV calculation using this representation is identical to linear velocity calculation, so $RV = \Delta\theta/\Delta_f$, where Δ_f is the time between frames and is a constant $= 1/120$ seconds (8.3 milliseconds). Applying the finite-difference method, $RV_f = (\theta_f - \theta_{f-1})/\Delta_f$.

For every frame in each *throw window*, the joint motion features of a number of previous frames up to and including that frame are extracted and sequenced in an array to form a *sliding window* of size w. Note that this process ensures that the model can be used in real-time as it does not require information about future frames. We extract the four motion features P, V, R, and RV from the motion data. Besides being intuitive, these features are central to human biomechanics studies [337] and are calculated from the position and rotation data described above.

Lastly, for all throws, a binary vector (the *target* vector) of size W is also defined, where the element corresponding to the ground truth PoR frame (i.e., the center frame) is set to $True$ and all others are set to $False$. This, however, creates a highly skewed dataset with zeros significantly outnumbering ones by 1:$(W - 1)$. We handle this in our model by using a weighted loss function or downsampling the data.

Model Training

All of these data are used to define a joint-feature configuration, or *design matrix* X, which is used as input to our model for training. The size of any X is specified by the number of joints used times their motion features (e.g., {P, V} for wrist, {R, RV} for shoulder) and the number of sliding window frames. Let $M = \{M_1, M_2, ..., M_m\}$ denote the set of motion features where each $M_i \in \mathbb{R}^{D_i}$. Let $J = \{J_1, J_2, ..., J_n\}$ be the set of joints and each J_i is linked to a subset of motion features M_k, $M_k \subset M$, i.e., $M_k \to J_i$. Every motion feature is included as many times as the sliding window size, w. Accordingly, the total number of columns in the design matrix can be calculated as:

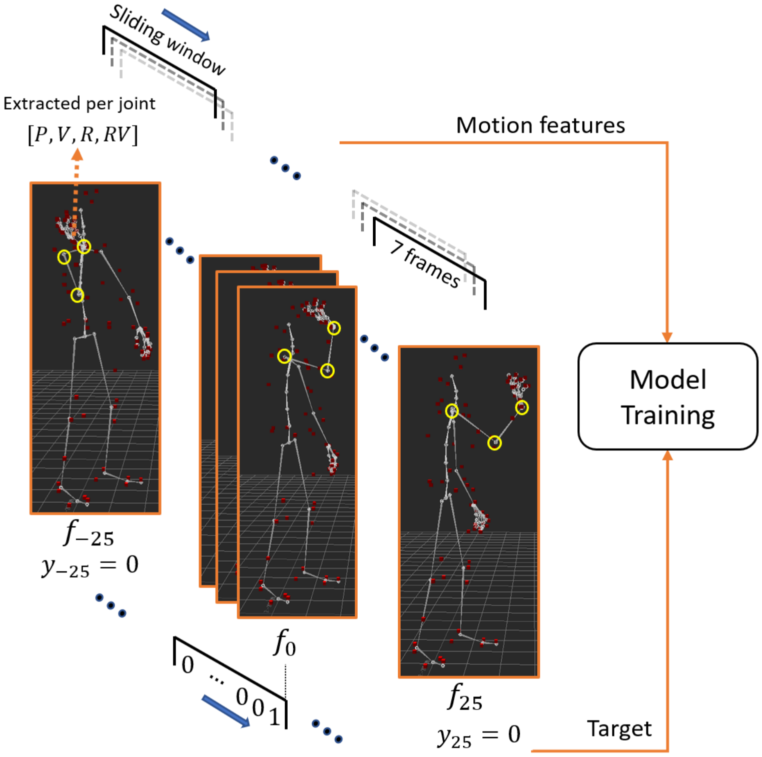


Figure 3.6: Data preprocessing: A sliding window sequences motion features from the 6 previous frames and the current frame to form the model input of a single frame. This is repeated for each joint used by the model. Motion features are P=Position, V=Velocity, R=Rotation, RV=Rotational Velocity; the Target vector contains a single True label at the PoR, f_t.

$$\#columns = w \sum_{i=0}^{n} \sum_{\forall M_k \to J_i} D_k \tag{1}$$

Each of the motion features contributes to the size of the input by their dimensions. Positions are three-dimensional, rotation quaternions are four-dimensional, and both velocity and angular velocity (magnitudes) are one-dimensional. Equation 1 is useful because we are applying a search over multiple combinations of joints and motion features. The number of rows in X is specified by the *throw window* size, W, times 1134 throws. The final size of X is $(1134 \times W)$ by $\#columns$. To create the target array, we stack 1134 target vectors, so the final size is $1134 \times W$. Stochastic gradient descent was used for optimization. All of the features were scaled and transformed where necessary, e.g. skewness. Input data is formed by merging data from different joints with any subset of motion features. In total, with three joints and four features, this creates a set of twelve joint-feature selections. A full configuration, therefore, means that all twelve joint-feature selections are included. Following Equation 1, this creates $7 \times 3 \times 9 = 189$ columns in X.

The models consist of stacked, fully connected feed-forward neural network layers with a sigmoid layer at the output and binary cross-entropy as the loss function. The input dimensions and the number of neurons in hidden layers varies based on the type of joint-feature combination being used, but networks are shallow due to the dataset being small. ReLU activation was used, and each fully connected layer was followed by batch normalization and dropout.

The models are developed using PyTorch [338]. Parameters of the network are initialized using uniform Xavier initialization [339] and we used a batch size of 128. The learning rate was initially set to 0.05 and it was decayed by a factor of 0.75 when there were no improvements. Each model was run for a maximum of 50 epochs with early stopping.

Algorithm 1: Delayed Response (DR) algorithm

```
Input  : Moving-averaged window of probabilities, $\{p_0, ...p_T\}$
Output: $frame_{max}$, $max$
$p_{threshold} = 0.5$, $max = 0$
while $i < T$ and $max\_reached = False$ do
    if $p_i > p_{threshold}$ then
        if $p_i > p_{i-1}$ then
            $max = p_i$; $frame_{max} = i$;
        end
    else
        $max\_reached = True$;
    end
end
```

Detection Methods: We use three methods to estimate the PoR frame in a *throw window* from the model's output probabilities (see Figure 3.7). The first approach is inspired by the nature of a real ball throw, i.e., a release cannot be reversed and it can happen only once during a *throw window*. In our first method, *First Nonzero* (FN), we start classifying each frame in the *throw window* starting with f_{t-25} and detect a release at the first estimated PoR frame. The classification is done based on the specified threshold value p. In our analysis, we used probabilities ($p = 0.3$) (FN30) and ($p = 0.5$) (FN50). For each classified throw, the distance between the detected PoR frame and the real PoR frame is stored.

The FN method offers a very basic solution but it only applies a threshold p, and it does not assess the rate of change at the model output as the motion unfolds. We propose a second method, *Delayed Response* (DR), that introduces a small delay in order to check the rate of change at the output probabilities of the model. The delay is introduced by a moving-average filter centered around the target frame in order to reduce the noise at the model's output. After searching over different filter sizes, we found a filter size of 5 frames to be optimal, causing a delay of 2 frames.

Similar to FN, we also apply a threshold to avoid the algorithm from being prone to noise. A pseudocode is provided in Algorithm 1. We have used threshold values of $p = 0.3$ (DR30) and $p = 0.5$ (DR50) in our analysis.

When the network is passed an entire *throw window*, it outputs a pseudo probability distribution over the window. Our third method, *Gaussian Fit* (GF), fits a Gaussian distribution to this sequence of probabilities and sets its mode to be the detected PoR frame. The use of all frames in a window means that this method is not applicable for real-time use, but is useful for comparison and we expect an increase in PoR detection probability around the ground truth. It acts as a gold standard that displays the optimal performance of the network and avoids any performance drops introduced by the detection methods.

Model Evaluation: We evaluate a trained model using multiple metrics such as detection rate and average detection error. We use a novel metric, *within*, which uses the size of the time mismatch between the actual and estimated PoR frames to calculate accuracy. In Change Point Detection research, it is common for algorithms to be evaluated on *within*, e.g., within-x seconds change-point detection [299]. We prefer this metric because common binary classification metrics do not provide enough insights into the capabilities of a time series classification model. To clarify this point with an example, let us assume that a trained model has a classification accuracy of 90 percent, meaning for a throw motion window of 100 frames, a significant number of frames would be classified correctly. However, if one of the falsely classified frames is near the beginning of the motion, the model would evaluate it as a release and initiate the throw at an early point in the motion. Therefore, the model needs to be evaluated by its ability to cluster the estimated PoR frames around the real PoR frame, whether they are correctly classified or not. We formulate $within(d)$ as follows:

$$within(d) = \frac{1}{100} \sum_{i=1}^{100} \mathbb{1}(|arg(\hat{Y}_i = 1) - arg(Y_i = 1)| \leq d) \tag{2}$$

where Y_i represents the target vector for one throw, $\hat{Y}_i$ represents the predicted target vector, $\mathbb{1}(.)$ is the indicator function and 100 is the size of the test set. Since $within(x)$ sums up estimations that are less than x frames apart from the ground truth, $within(y) > within(x)$ for any $y > x$. It is thus a cumulative and monotonically increasing function (see Fig. 3.9).

In words, $within(d)$ metric displays the percentage of throws in the test set that are detected within d frames of distance from the actual PoR. For our data captured at 120 Hz, a frame mismatch of size 10 results in a shift of 83 milliseconds (<0.1 seconds). We provide the results using the *within* metric as it appears to be effective in the current study. Clustering the estimated PoR frames around the actual PoR frame has highlighted the importance of the correct classification of non-PoR frames rather than PoR ones. We observed that the ideal classification of a throw, i.e., the whole *throw window* is classified as zero except the throwing frame, almost never occurs. Hence, we modified the weighted loss further to bias it towards the correct classification of non-PoR frames.

3.2.2 Results

The trained models were tested by randomly subsampling the full set of throws into 10 subsets of size 100 each. Each subset was separated from the rest of the data and was not exposed to the model at any step until the very end for evaluation. As a baseline classifier, we have used Scikit-learn [340] to train multiple k-Nearest Neighbor (kNN) models with different parameters (n). The best-performing one is picked and used as the baseline classifier. The kNN classifier with the best results was achieved with $n = 2$.

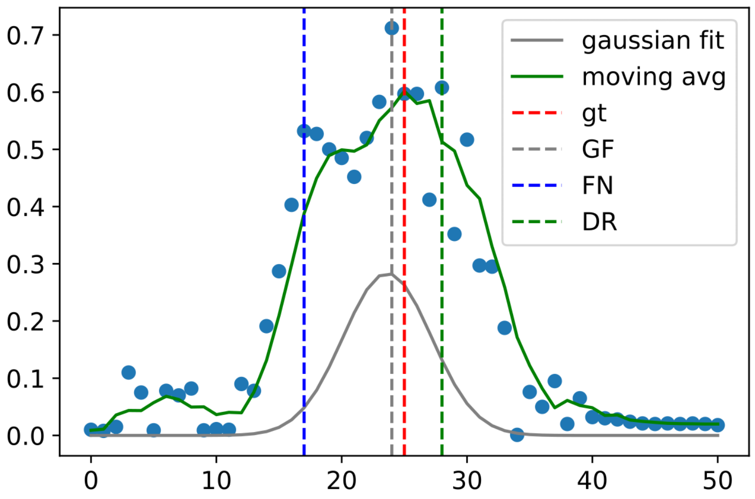
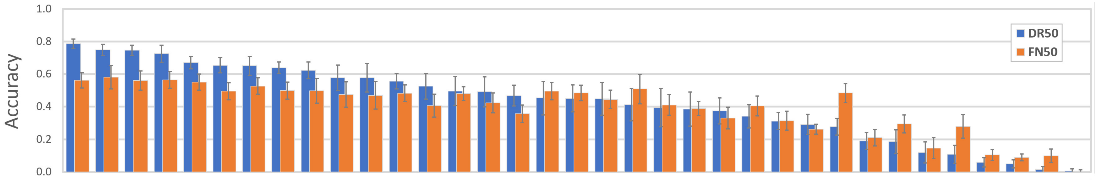


Figure 3.7: Illustration of FN, DR, and GF methods. Ground truth lies at the center of the window and scattered points represent the estimated probabilities of a frame representing a PoR. The FN method classifies the first frame that has a probability above 0.5 to be the PoR frame. The DR method runs a moving average filter and searches for the maxima after probability 0.5 is reached (see Algorithm 1). GF method fits a Gaussian distribution to the window of probabilities and assigns the *argmax* frame as the PoR of the *throw window*.

In phase 1 testing, we first performed a semi-exhaustive analysis over a group of joint-feature configurations to try to rank motion features and joints by their performance for frame-level throw classification. We used both real-time methods, FN and DR. The configurations are color-coded and sorted based on their $\mathit{within}(5)$ accuracy using DR method in Figure 3.8. We interpret the results of the semi-exhaustive analysis and offer general findings. Based on these phase 1 results, in phase 2 we select several key joint-feature configurations and a more thorough analysis of multiple metrics is presented, shown in Figure 3.9. We choose the configurations with the goal of optimizing data size and model performance, in order to create a relative importance ranking of features and joints. Both phases involved training a unique model for each configuration.

Semi-exhaustive Analysis: The phase 1 results are obtained by averaging the $\mathit{within}(5)$ scores of ten test sets (see Figure 3.8). For the semi-exhaustive analysis, the configurations of the sets of joints $\{W, E, S\}$ and motion features $\{P, V, R, RV\}$ (where



Column #		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35
P	W	🟨		🟨		🟨	🟨			🟨				🟨		🟨											🟨									
	E	🟨		🟨		🟨		🟨		🟨						🟨								🟨							🟨					
	S	🟨		🟨		🟨				🟨	🟨			🟨										🟨												🟨
V	W	🟩				🟩	🟩											🟩			🟩							🟩				🟩				
	E	🟩				🟩		🟩										🟩			🟩								🟩					🟩		
	S	🟩				🟩					🟩										🟩							🟩	🟩						🟩	
R	W	🟦	🟦	🟦	🟦		🟦					🟦			🟦							🟦														
	E	🟦	🟦	🟦	🟦			🟦	🟦			🟦					🟦																			
	S	🟦	🟦	🟦	🟦				🟦		🟦				🟦											🟦										
RV	W	🟧			🟧		🟧						🟧						🟧	🟧					🟧											
	E	🟧			🟧			🟧											🟧	🟧			🟧							🟧						
	S	🟧			🟧						🟧		🟧						🟧				🟧										🟧			
#Joints:		3	3	3	3	3	1	1	2	33	1	2	2	2	2	2	1	2	3	2	3	1	2	2	1	1	1	2	2	1	1	1	1	1	1	1
#Neurons L1:		250	150	150	200	100	80	80	100	80	80	80	40	60	100	60	40	40	60	40	60	40	40	60	20	40	30	40	40	20	30	20	20	20	20	30
#Neurons L2:		150	100	100	150	50	40	40	60	40	40	40	20	30	50	30	20	20	30	20	30	20	20	30	10	20	10	20	20	10	10	10	10	10	10	10

Figure 3.8: Table of joint-feature combinations and within-5 results of the *First Nonzero* (FN50) and *Delayed Response* (DR50) methods, averaged over ten runs and sorted by *Delayed Response* values. Each column in the table represents a joint-feature combination. Each row represents a joint-feature selection. The color indicates that data from the specific feature-joint selection is used in the model, e.g., column 1 uses position and rotation for all three joints, but does not use any other motion features. W,E,S=wrist, elbow, shoulder; P, V, R, RV=position, velocity, rotation, rotational velocity; L1, L2 = number of neurons in each layer.

W=wrist, E=elbow, S=shoulder, P=position, V=velocity, R=rotation, RV=rotational velocity) are as follows:

- Every single motion feature M_i is linked with all possible combinations of 1, 2, and 3 joints (i.e., $\{W, E, S, WE, WS, ES, WES\}$; e.g., in Fig. 3.8, col. 2 shows the combination of R with all three joints.

- If device A measure features P and V, and device B measures R and RV, we choose all combinations that they can measure. Thus, we always pair P with V (P-V) and R with RV (R-RV). All combinations of motion feature pairs are linked with all three joints, i.e., col. 1 is combination P-V, R-RV, col. 4 is R-RV, column 5 is P-V.

- Position and rotation (P-R) together are linked with all joints (col. 3).

- If a device C measures all features P,V,R,RV, we apply this device to each single joint W,E,S (col. 6, 7, 10).

Looking at the DR ranking, first 4 configurations have very similar results, reaching above 70% accuracy. The only configuration that achieved above 70% using a single motion feature is rotation (column 2), which suggests that rotational information is very critical. The comparison of configurations 2 and 8 also show that wrist rotation is not very critical when elbow and shoulder rotations are used. Moreover, all configurations using only R in two-joint combinations (columns 8, 11, 14) perform relatively well, indicating that rotation is informative for all three joints.

Another important observation is the poor performance of configurations involving only velocity (columns 31, 33, and 34). This is intuitive for the elbow and shoulder, as they do not move a lot in an overarm throw unless the motion is very intense, which was rarely the case in this dataset. The motion along the arm relies heavily on the type of throwing performed, e.g., motion occurs along the whole arm in an underarm throw, but only in the forearm for a light overarm throw. An analysis that categorized the dataset into overarm and underarm throws showed that there was in fact a higher percentage of overarm throws (64%). This partly explains the poor performance of a position and/or velocity features in all combinations of $\{E, S\}$. The surprising result of wrist velocity with FN exceeding DR in $within(5)$ performance has two interpretations. One is that the network's output hardly reaches the $p = 0.5$ threshold, and the smoothing introduced by DR lowers the probability below 0.5. Secondly, the 2-frame delay pushes the predicted PoR out of the $within(5)$ range and thus harms the performance. However, it is not clear why wrist velocity exhibits poor detection performance.

The slightly worse performance of rotational velocity versus rotation can be explained by rotation having multiple dimensions and thereby containing more information about the motion. Rotational velocity, on the other hand, is a scalar magnitude and therefore the model finds it difficult to learn to distinguish motion patterns. The model trained on the linear velocity of the wrist (column 31) does not suffer from the same issue, probably because it exhibits a more distinctive pattern around the PoR frame.

The comparison of DR and FN performances provides some insights on the capabilities of our system. While both methods require the network to be 50% convinced to make a decision, the use of the rate of change in DR causes an advantage in informative joint-feature configurations such that the probability curve guides DR to a more accurate

Configuration	Wrist	Elbow	Shoulder
C1	P,V,R,RV	P,V,R,RV	P,V,R,RV
C2	P,V	-	R,RV
C3	V	-	R
C4	P,V	-	-
C5	-	-	R,RV

Table 3.2: Selected joint-feature configurations for phase 2, where P = Position, V = Velocity, R = Rotation, RV = Rotational Velocity. Results are provided in Figure 3.9.

detection. This advantage decays as the configurations become less informative of the PoR, and even results in worse performance in some cases, caused by the moving-average filter. FN, on the other hand, displays less variation in performance among all of the configurations due to its more basic nature of it.

In the best 16 configurations, a clear advantage of DR is observed. Starting from configuration 17, the performances of both methods are fairly similar with FN being superior in a few cases (columns 27 and 31). Thus, we are evaluating those configurations by their FN results. Looking at configurations 20 and 28, it is seen that the exclusion of linear velocity in the wrist joint causes a big drop in performance. This idea is supported by the comparison of configurations 31, 33, and 34 where models are trained using linear velocity on a single joint.

Overall, the findings are:

- R is the most informative single motion feature when all joints are used (column 2).
- P, V on their own are significantly more informative for the wrist joint (columns 12, 22) than for the elbow (columns 19, 27) or shoulder (columns 31, 32).
- R-RV and P-V perform similarly when all joints are used.

Complete Analysis: As a follow-up on the semi-exhaustive analysis, we chose a selection of candidate joint-feature configurations, shown in Table 3.2, and evaluated them using a selection of detection methods and multiple metrics. The configurations are cho-

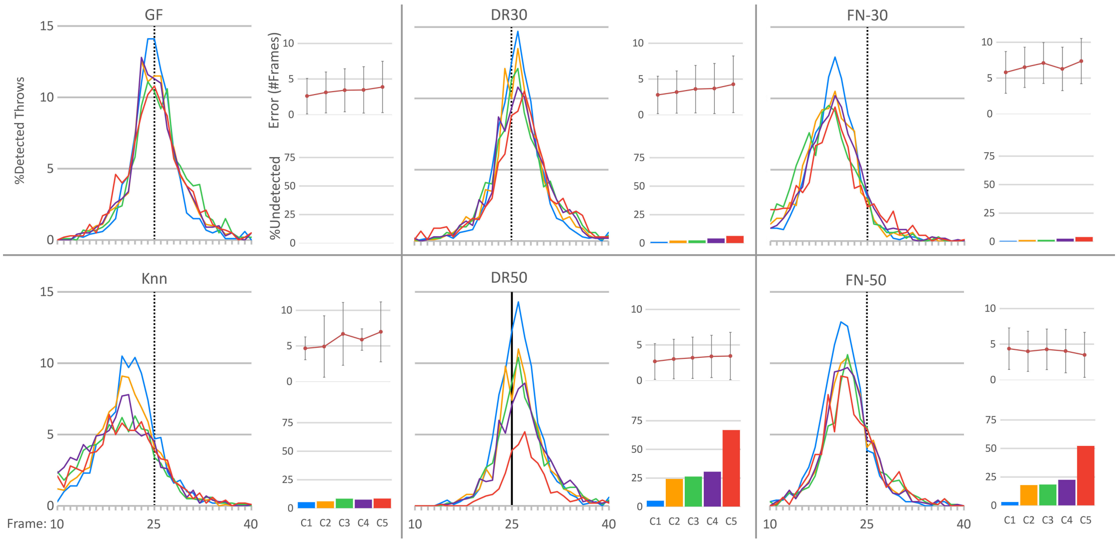


Figure 3.9: Point of release detection results provided as detection index histograms, detection ratios (%*throws*), and mean absolute detection errors (#*frames*). The detection methods Delayed Response (DR30, DR50), First Nonzero (FN30, FN50), and Gaussian Fit (GF) are applied to the output of the trained neural network models of selected joint-feature configurations (C1-C5). K-Nearest Neighbours (kNN, $n = 2$) with the FN50 detection method is provided as a baseline classifier. See Table 3.2 for configurations.

sen based on small but important differences between them. For example, C2 and C3 examine the impact of P and RV in classification accuracy. Comparison of C2 and C4 provides insights on the importance of rotational information, R and RV, of the shoulder. Similarly, C2 versus C5 conveys how necessary it is to have the position of the wrist. For the detection methods, we have employed GF, DR, and FN methods. We used two different threshold values, 0.3 and 0.5, to analyze their effect on DR and FN performances. As a baseline, we have trained a kNN classifier ($n = 2$) for each selected configuration. See Figure 3.9 for the different detection methods applied.

The first metric used is the detection index histogram, which shows the number of PoR detections at each frame out of the 100 throws in the test set. It shows the bias in detection performance with respect to early or late detection. The second metric used is the average absolute error. The absolute values were used to avoid early and late detection errors to cancel out each other. This metric provides a more certain measure on how far the average prediction lies with respect to the ground truth. Lastly, we present the undetected throw percentage, which displays the capacity of the model to detect a throw. Together, these metrics provide a full overview of the performance of the model and the detection method. The results are shown in Figure 3.9.

A comparison of configurations over all detection methods shows that C1 has a clear advantage in the throw detection ratios with around 5% of undetected throws in each detection method, and it also has the best average error. This is significantly important for a setting where the system is tested on a wider throwing motion. The model trained with C1 is capable of detecting almost all of the throws within a reasonable distance from the ground truth.

Configurations C2, C3, and C4 rank similarly in all detection methods with respect to both average error and detection ratio. Thus, we state that the inclusion of P and RV in the comparison of C2 and C3 does not improve the results. Moreover, the inclusion of R and RV on top of P and V (of configuration C4) does not improve C2. The low detection ratio of C5 in almost all graphs shows that shoulder motion information alone is not enough to detect PoR. The comparison of C2 and C5 emphasizes the importance of the wrist joint. The type of throw performed has a large impact on limiting the motion of the elbow and shoulder, while wrist motion is present in both overarm and underarm throws. The undetected throw ratio for C5 emphasizes this because the models are trained on a dataset with 64% overarm throws. Similarly, the detection ratio for configurations except C1 relies heavily on the type of detection method applied, indicating that the models trained with these configurations have relatively high uncertainty about the PoR, and therefore their performances are affected by different detection methods.

The FN method for both thresholds is skewed to the left, resulting in early detection. This is intuitive since this method flags a PoR at the first frame the model's output reaches the threshold. The effect of changing the threshold is observed in the undetected throws ratio and the average error. With a lower threshold, FN30 is able to detect above 95% of the throws for all configurations C1-C5. On the other hand, FN50 is only able to detect half of the throws for configuration C5. This increase in detection ratio causes a decrease in average error in all five configurations. An interesting observation at this point is that while FN30 performs worse for all configurations in terms of average error compared to FN50, DR30 performs similarly compared to DR50 with a much better detection ratio. This is the result of using the rate of change of the model's output in the

DR method.

Using the GF method, all the selected configurations (C1-C5) achieve similar performance, showing that each motion feature contributes to PoR detection. There are no undetected throws in GF since there is a detection no matter how small the output probabilities are. When methods FN and DR are compared with GF, significantly better performance is observed in GF. This has two explanations: i) due to the probability thresholds applied in FN and DR, there is a number of undetected throws, i.e. undetected ratio; and ii) the threshold is reached too early in the throwing window for the FN method, or the algorithm is stuck at a local maximum for the DR method, which causes worse performance compared to the GF method.

We combine the results of the two analyses as follows:

- All of the chosen motion features are informative about the detection PoR.
- R is the most informative motion feature.
- Wrist is the most informative joint.

3.3 Discussion

In this Chapter, we first presented a preliminary implementation of virtual throwing in VR, and an experiment that used this implementation to study the effect of visual cues on virtual throwing performance. We then described a high-level semi-exhaustive approach to determine which motion features and body joints are critical for PoR frame classification when throwing virtual projectiles. Our analysis shows that the extracted motion features of the arm provide a good estimate of the PoR within a reasonable time error. The semi-exhaustive search over joint-feature configurations showed that some motion features are more critical than others, e.g., rotation provides the most information. Moreover, an importance ranking among joints can be observed, e.g., the motion of the wrist is more informative than the elbow and the shoulder joints. The complete analysis of several selected joint-feature configurations (see Table 3.2 and Figure 3.9) extended our findings further by confirming the importance of the wrist joint, and showing that all motion features provide useful information.

By introducing the *within* metric, we presented an approach that seeks to leverage the limitations of human perception. As a follow-up on this research, we decided to conduct a user study to explore how throws with inaccurate release timing are perceived (i.e., too early or too late) (See Chapter 4). Such a user study makes the results of this study more valuable and easy to interpret. For example, one hypothesis is that humans are more accepting of early releases than late releases, as delays would reduce the perception of causality, which is an important factor in achieving a sense of agency. If this proves to be true, we would need to modify the *within* metric which is symmetric, as we evaluate our models with the assumption that perception of early and late throws would have similar effects.

There are important limitations to the proposed throw detection system in this chapter. First of all, we did not perform cross-validation, which is a crucial aspect in real-life machine learning tasks, which makes sure a model succeeds in unseen data. We did not perform cross-validation because our dataset contained participants of unique body sizes. In future work, we plan to transform motion data such that the impact of these differences in limb sizes and actor height is diminished. Another limitation is that the model was not evaluated on a dataset containing multiple actions or longer motion sequences. Here, the analysis is done over the assumption that every fixed motion window is a throw and contains a PoR, which is the main reason why the GF method acts most successfully. A complete throw detection system needs to be evaluated on longer motion sequences with more variety. We consider the dataset to be another limitation. The data was captured during a VR experiment and the virtual throwing algorithm was a set of heuristics that operated plausibly for some participants but not so much for others, with some participants reporting that the PoR felt delayed. In return, the delayed PoR experience could have had an effect on how subjects carried out their next throws. Therefore, we will improve the dataset for our next study.

4 Predicting the Point of Release of a Ball

In this chapter, we present the studies performed following the preliminary work, containing a perceptual study on PoR timing errors and a revised investigation of PoR detection, including a proof-of-concept real-time PoR detection system called ReTro.

4.1 Exp. T2: Perception of Point of Release

From infancy, humans can recognize when something unexpected happens in the physical world [341]. When expectations are not met in a virtual setting, such as a game or Virtual Reality (VR) experience, the user experience will be impacted. As technologies develop rapidly to support interactive experiences that encompass both the virtual and the real, new problems arise when virtual physical events do not result in the expected outcomes. Several factors may affect the perception of a thrown virtual ball, e.g., active throwing in VR or controlling a game avatar. The most challenging aspect of simulating virtual throws is accurately detecting the moment when the thrower releases the ball, known as PoR.

Errors in the timing of the PoR can lead to visually disturbing results if the ball's trajectory does not match the viewer's or thrower's expectation (Figure 4.1). In this section, we examine the effect of release timing on the perception of throwing. Participants watched videos of virtual throwing motions with different levels of PoR timing errors. Their task was to judge whether the motion had been modified or not. The questions we wished to answer from these studies were as follows: Does the *View* from which you observe increase or decrease your ability to detect an anomalous throw, e.g., watching the throw from a *Side View*, as when attending a baseball game or similar; or watching it from a *Front View*, as if throwing the ball oneself? Is it easier or more difficult to detect an error based on the *Distance* of the throw? Are different types of throwing motions more or less robust to errors in timing, e.g., *Underarm* vs. *Overarm* throws? What is the PoR timing *Error* beyond which an anomalous throwing motion can be detected?

With respect to active throwing, it was found that people are less accurate in VR than in reality [17], mainly due to lower accuracy in distance and height dimensions (see Section 2.2.1 for more details). This suggests that there were errors in release timing along throw trajectories. Furthermore, Butkus and Ceponis [18] conducted a VR throwing experiment to study distance perception and found that throwing accuracy improved as the distance increased in VR, and the throwing velocity was higher in VR compared to reality. Covaci et al. [19] evaluated how effective VR would be for training beginner players to throw a basketball. By tracking a real ball and simulating its continued trajectory within a Virtual Environment, they ensured that the PoR detection and ball trajectory were very accurate. They also explored different viewpoints and found that participants estimated the distance to the basket more accurately from a third-person view. Finally, Faure et al. [342] examined other factors that can affect the perception of ball-throwing, such as expertise.

Based on the above-mentioned studies, we chose several variables for an experiment exploring viewer sensitivity to errors in the Point of Release (PoR) of a ball during throwing motions. We hypothesized that the *View* from which participants observe the throwing motions will affect their ability to detect an error, e.g., perhaps certain anomalies, such as angular distortion, would be more visible from the front than from the side. Due to the different velocity properties of long and short throws, our hypothesis was that *Distance* will also play a role, e.g., it could be that errors in slower throws are easier to detect and that the motion of the *Arm* will also change the dynamics of the throw, as it did in [63]. We also hypothesized that there will be an asymmetry in early and late PoR timing *Error*, which will vary depending on the other factors.

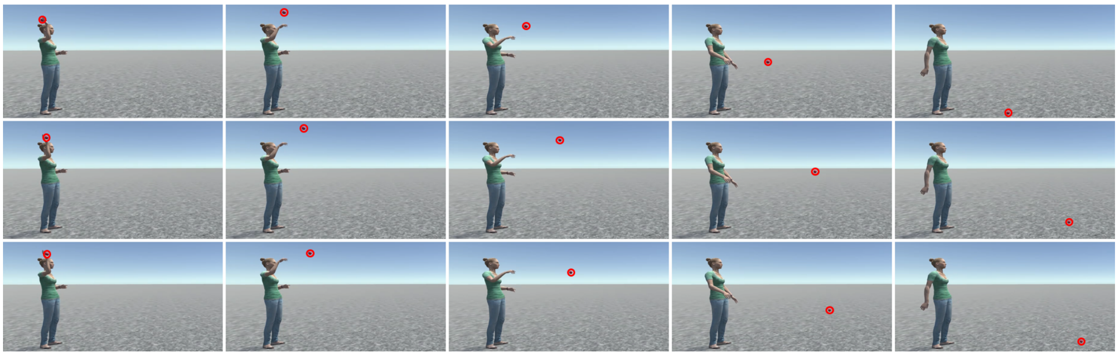


Figure 4.1: Changes in ball trajectory caused by modifying the Point of Release (PoR) of a throwing motion: early release (t); original trajectory (m); late release (b); for the same five animation frames

4.1.1 Dataset

The dataset used in the preparation of the stimuli involves the simultaneous capture of only two actors (1M, 1F) due to the circumstances of the COVID-19 pandemic. This capture only involved the optical motion capture system as we decided to switch to Vicon's full five-finger tracking from Manus VR gloves. We captured around 1000 throws using a tennis ball with the distance between the two actors incrementally changing from 2 meters to 6 meters. Both the catcher and the thrower were captured for efficiency reasons and to identify the Point of Catch (PoC) which was later used to calculate the ground truth PoR. Before the capture, we investigated the size of the capture area which the MoCap system was able to track accurately, and found that a simultaneous capture allows the actors to stand only up to four meters apart. To overcome this, when capturing far throws, we captured the full body motion of only one actor and tracked only the hand motion of the second actor through a marker attached to the back of the hand.

4.1.2 PoR Ground Truth Approximation

Extracting the PoR from motion data is a crucial component of the work as it forms the ground truth used in the perceptual study of PoR and supervised machine learning for PoR prediction. Over the course of this work, our algorithms and approach for this problem have evolved. Our modified approach can be found in Section 4.2.2.

The question of "What constitutes the release of a throw?" does not have a straightforward answer. It is not clear whether the release happens when the ball loses contact with the hand or when the fingers start opening to release the ball. Past studies on throwing have relied on measurements that approximate the point of release such as pressure-sensitive microswitches [219] and joint angle calculations using LEDs [231], which are heuristics. As we have full-body motion data for the thrower and the catcher, we decided to approach this as a trajectory-fitting problem, in which we simulate the trajectory of the ball and iteratively find the best-fitting release frame. There are two assumptions to simplify the problem. The first assumption is that the release of a throw is instantaneous rather than occurring over a period of time; and second, the velocity of the hand is perfectly transferred to the ball at the point of release without any loss of energy.

The motion data from the MoCap system only provides joint motions, making it necessary to specify a ball release position inside the palm. We process the entire dataset by adding a ball release node to the motion data of every actor. Figure 4.3 shows several examples. During a throw, the positioning of the ball in the palm can significantly affect its trajectory depending on the speed of the throw. Trajectories of the ball are simulated from this node with the node's velocity calculated from its position. In Figure 4.2, we demonstrate how our approach can yield different results when the ball is assumed to be released from different positions in the hand. The three different positions not only lead to considerable differences in trajectory but also suggest different PoR frames. An alternative approach is to also perform a search over different release positions together with the time dimension, however, this results in multiple possible solutions since we use a very simple physical model to simulate ball trajectory. Eventually, we employed A3 (shown in Figure 4.3) as the attachment point and used that to find the PoR frame in all throws.

The PoR and PoC frames for both actors were calculated offline in two steps using multiple heuristics: (i) candidate PoR frames were located, (ii) a trajectory was fitted between the thrower and the catcher. In the first step, we chose hand velocity, middle and index finger rotations, and hip velocity as our indicators in candidate PoR frame selection. Middle and index finger rotations provide a clear indication of a releasing motion, however, they do not suffice on their own to decide it is a throw, hence we make use of hand velocity and hip velocity. On both hands, we apply a constant velocity mag-

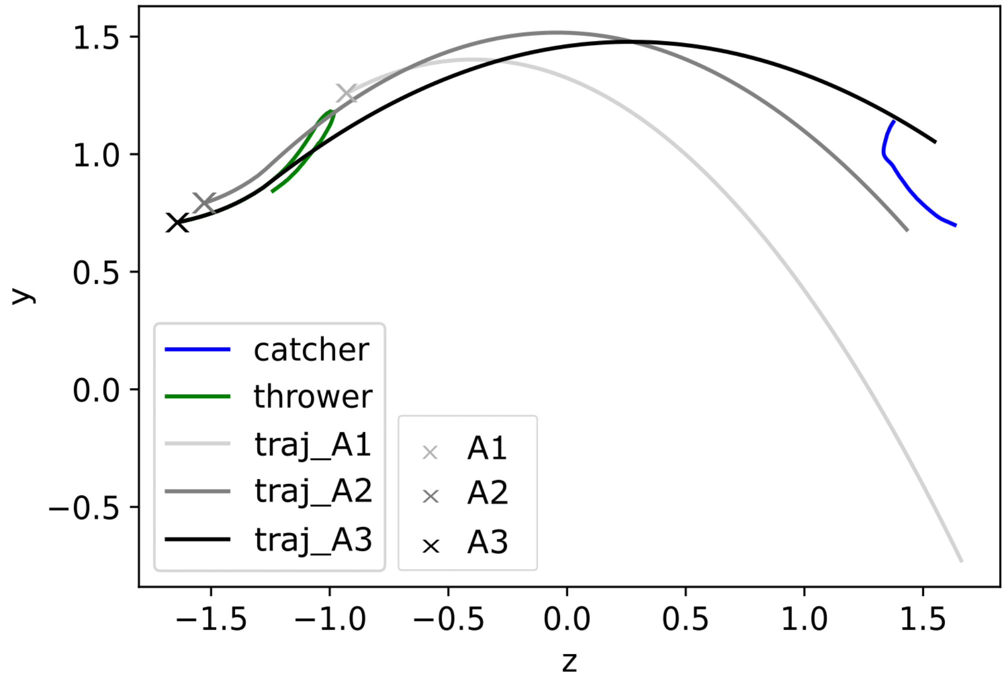


Figure 4.2: The impact of ball attachment position in the fitted trajectory, demonstrated on a single throw. Each trajectory is the best-fitting trajectory using a different attachment point (A1,A2,A3), as shown in Figure 4.3. The catcher's and thrower's right hand positions are also included. In the illustrated throw, the catcher's hand moves in the positive y direction during the motion.

nitude threshold and only consider the intervals that have higher velocity magnitude, in order to locate throw swings. Similarly, a large hip velocity magnitude implies that the motion is walking or running, and not throwing. This is because the actors were instructed to be stationary during a throw. Therefore, if the hip velocity magnitude is large, we filter out those regions from our set. In terms of finger rotations, we calculate an average of the variance of the index and middle finger flexion over a window of 21 frames located around the candidate PoR frame and compared it against a threshold of 10. Finally, an interval of frames that satisfies all three conditions is obtained, in which we choose the frame with maximum hand velocity and perform trajectory fitting around that candidate PoR frame. A flow diagram is provided in Figure 4.4.

In step 2, we improved on these candidate frames and fitted a ball trajectory between the

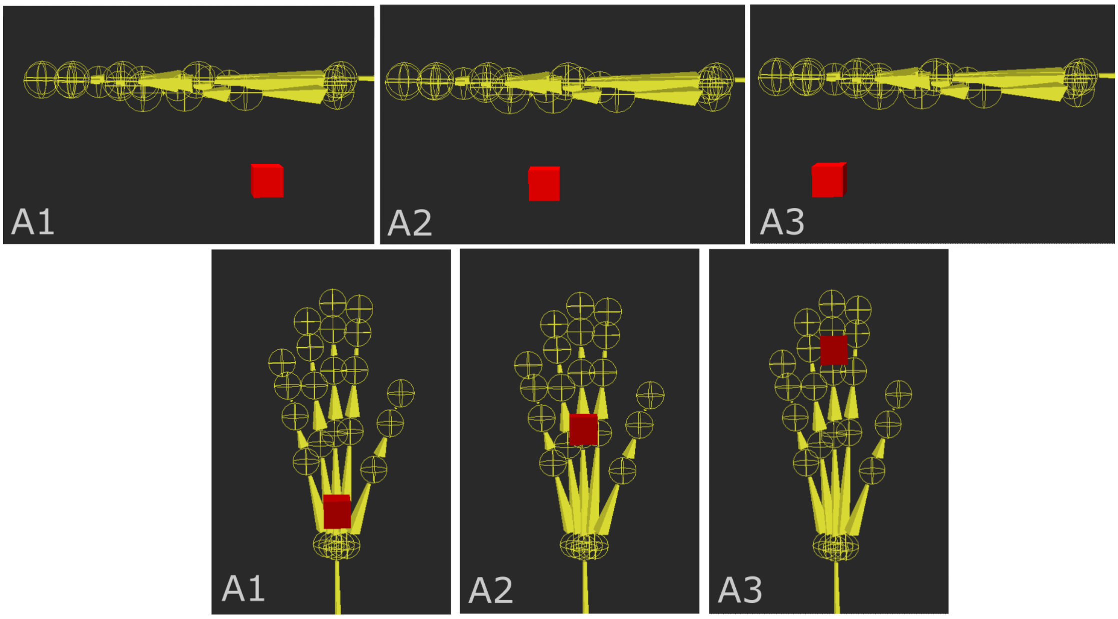


Figure 4.3: Top- and side-view visualizations of three different attachment points (S1,S2,S3) used in the assessment of attachment position impact on trajectory fitting, as shown in Figure 4.2.

thrower and the catcher. First, the distance between the actors, and the thrower's hand velocity was used to estimate the airtime for the ball and find a candidate PoC frame. The horizontal velocity of the ball in the air was assumed to be constant. We located search windows of 20 frames around the candidate PoC and candidate PoR frames to calculate the best-performing trajectory. In an exhaustive manner, trajectories for the ball were simulated for all potential PoC and PoR frames (ignoring rotational forces and drag). The error is defined as the point on the trajectory with the minimum distance between the ball and the hands of the catcher. The candidate PoR and PoC frames with the trajectory that reached the lowest distance were then selected.

4.1.3 Stimuli

We selected a total of eight throws (two Overarm-Near, two Underarm-Near, two Overarm-Far, two Underarm-Far) from our dataset to create our stimuli. A visual representation of the throwing motion can be found in Appendix A1.1 for all of the eight throws. We sorted all throws based on landing distance and release velocity. Two examples of over-

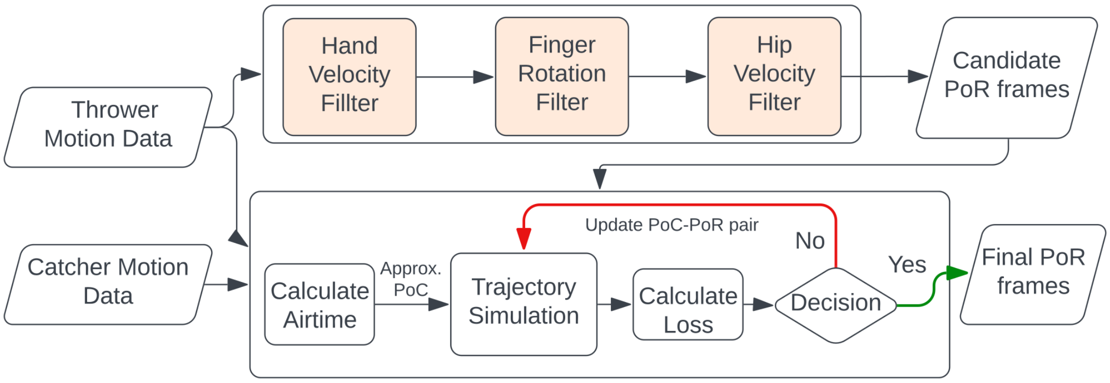


Figure 4.4: Flow diagram of PoR ground truth approximation.

arm and underarm throws at two landing distances of similar release velocities were picked based on observed motion neutrality (i.e., no noticeable features such as a lifted left arm or a bent knee) and motion quality (i.e., no retargeting artefacts). Near and Far distances were selected as 1.9 meters and 5 meters, respectively. 1.9 meters is close to the upper limit of the Near region in Exp. T1 and 5 meters slightly above the upper limit of the Far region in Exp. T1 before it was modified to 4 meters from 4.5 meters. The PoR and PoC frames were also visualized to ensure that the PoR timing was accurate.

To create the throwing animations, we retargeted the motions of one female actor to a 3D avatar in Unity. We created a ball with a 6cm diameter and attached it to the throwing hand of the avatar. As very high timing precision and control were required, we generated lookup tables to read the position of the ball rather than using Unity's internal event system, which can be unreliable because of changing frame rates. To go beyond the 120 fps precision of the mocap data, quadratic interpolation between positional data points was used to acquire 1000 fps hand position data. The lookup tables contained time-position pairs of the hand. In Unity, the time of an animation clip is represented by normalized time, which takes a value between 0 (start) and 1 (end). We converted this normalized time into the original mocap time of the lookup table.

At the start of each throw, a new trajectory was generated for the ball based on the delay specified for the current PoR, and a second lookup table was created for the new trajectory. Using these high-precision trajectory lookup tables, the ball could be released with any desired delay and the trajectories were not affected by Unity's frame rate. After the ball was positioned using the pre-calculated values in each frame, Unity's physics system took over to animate the ball when it was close to hitting the ground, in order to simulate a natural bounce (as the catch is not simulated). The clips were then recorded from two different views: the Side view reflects the case where participants observe another person throwing a ball, and the Front view emulates the experience of observing one's own avatar throwing a ball (e.g., as in a VR environment).

4.1.4 Method

A total of 34 participants (19M, 15F, ages 18-60+), with a variety of backgrounds and expertise, were recruited via email lists and social media. The experiment was run online using Qualtrics. After viewing the instructions and informed consent form, followed by entering some demographical details, participants pressed a key to begin the experiment. No personally identifying information was recorded (i.e., the fully anonymous mode was used), and the system ensured that each individual participated only once. The instructions also stated that a display should be used that will allow the motion details to be clearly seen, i.e., a monitor or laptop screen.

We used a within-subjects design, with independent variables View (Side, Front), Distance/Dist (Far, Near), Arm (Overarm, Underarm), and 11 different levels of timing Error: +/- 10, 20, 30, 40 and 50 milliseconds, plus the original motion with error 0. This resulted in 2*View X 2*Dist X 2*Arm X 11*Error = 88 combinations, with two repetitions of each using different motion clips, giving 176 video stimuli to be viewed by participants.

Participants viewed either a block of Side view videos first, followed by the Front view block; or vice versa, counterbalanced across all participants. For each block, a set of 8 stimuli was displayed at the start for training purposes, using videos that were not in the set of stimuli. These stimuli consisted of four Original throws (i.e., no PoR delay), from Near and Far distances, and for Overarm and Underarm. Each of the originals was followed by the corresponding Modified throw with an early or late Error. For the modified training examples, we used the most extreme delay of +/- 50 milliseconds. Participants then viewed all 88 videos in that block in random order and selected a

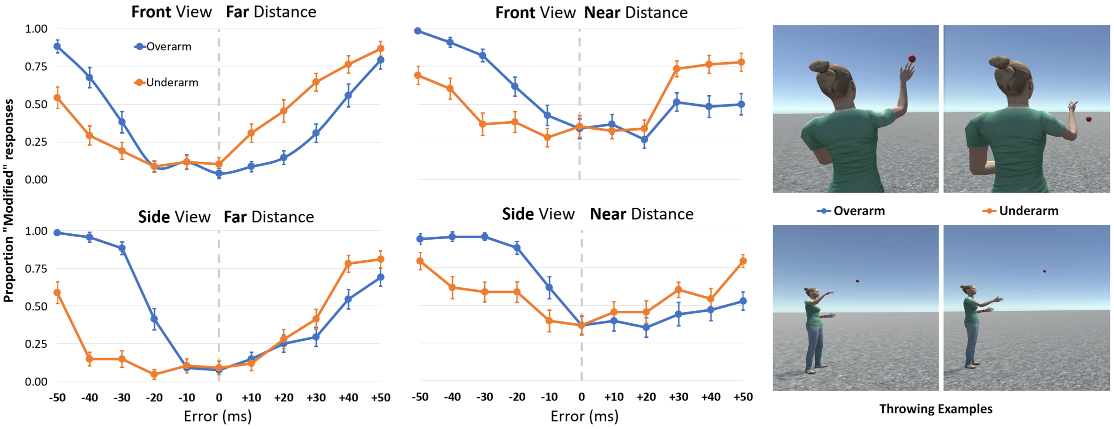


Figure 4.5: Four-way interaction effect VIEW*DIST*ARM*ERR with means calculated across all participants. Examples of frames of throwing motions are shown on the right.

button to indicate whether each throw was Modified or Original.

4.1.5 Results

We performed a repeated measures ANalysis Of VAriance with multivariate analysis (ANOVA/ MANOVA) to test for main and interaction effects of independent variables VIEW, Distance (DIST), ARM (over or under), and Error (ERR). Newman-Keuls and Bonferroni post-hoc tests were used to check for significant differences. All significant effects are reported in Table 4.1.

The main effects show that participants responded Modified more often for Near throws than for Far throws, and for Overarm than for Underarm. The errors were all statistically significantly larger than for the original, except for an early or late release of +/-10ms. However, all the factors interacted with each other, as shown in Figure 4.5. The interaction effects show that, when an Overarm throw had an early PoR error (-50 to -10), it was more often reported as Modified than when the PoR error was late (10 to 50), whereas the opposite was true for Underarm throws. We can also see that participants were much less accurate with their responses for Near throws, where over 30% of the Original throws (PoR error = 0) were mistakenly reported to be Modified. Furthermore, the larger number of Modified responses for Overarm throws than for Underarm is only true for the Side view (VIEW*ARM).

Table 4.1: Significant effects ($p < 0.05$) for Perception of PoR ANOVA/MANOVA with effect sizes (partial η^2)

Effect	**F-Test**	η^2
Main Effects		
DIST	$F_{1,33} = 36.69, p < 0.005$	0.53
ARM	$F_{1,33} = 18.64, p < 0.005$	0.36
ERR	$F_{10.330} = 97.73, p < 0.005$	0.75
Two-way Interaction Effects		
VIEW × ARM	$F_{1,33} = 7.19, p < 0.05$	0.18
VIEW × ERR	$F_{10,330} = 7.44, p < 0.005$	0.18
DIST × ERR	$F_{10,330} = 21.41, p < 0.005$	0.39
ARM × ERR	$F_{10,330} = 42.82, p < 0.005$	0.56
Three-way Interaction Effects		
VIEW × DIST × ARM	$F_{1,33} = 16.16, p < 0.005$	0.33
VIEW × DIST × ERR	$F_{10,330} = 2.23, p < 0.05$	0.06
VIEW × ARM × ERR	$F_{4,80} = 2.86, p < 0.05$	0.10
DIST × ARM × ERR	$F_{10,330} = 3.77, p < 0.005$	0.10
Four-way Interaction Effects (see Figure 4.5)		
VIEW × DIST × ARM × ERR	$F_{10,330} = 3.57, p < 0.005$	0.10

To explore possible reasons for the perceived accuracy or inaccuracy of these throws, we extracted some error metrics of the ball's motion trajectory for each type of throw and performed a correlation analysis (Table 4.2). The Angle metric measures the horizontal deviation of the angle of the ball's trajectory for each throw with a PoR Error from that of the original throw; the Landing metric gives the deviation in the length of the trajectory from that of the original throw; and the Velocity metric measures the deviation of the ball's velocity from that of the original throw.

We first tested the correlation of each error metric with the PoR error size. We see, for the Far view only, that significant positive correlations were found for the Angle and Landing errors with the PoR timing error, but not for the Velocity error. Next, we tested for correlations with the participants' error detection ratio, i.e., the proportion of Modified responses. The Front View and Side View entries refer to the average user

Table 4.2: Correlations of ball motion errors with participants' error detection ratio. Significant results ($p < 0.05$) are highlighted in red

		Angle	Landing	Velocity
3**FAR OVER*	PoR Error	0.81	0.98	0.07
	Front View	0.88	0.96	0.19
	Side View	0.91	0.91	0.30
3**FAR UNDER*	PoR Error	0.68	0.76	0.11
	Front View	0.93	0.87	0.09
	Side View	0.91	0.79	-0.02
3**NEAR OVER*	PoR Error	0.25	0.35	0.57
3**NEAR UNDER*	PoR Error	0.54	0.41	-0.32
	Front View	0.73	0.70	-0.41
	Side View	0.43	0.46	-0.18

responses to those same animations.

We can see that angular deviations are highly correlated with the participant responses, consistent with previous work on collisions [312, 333]. This is also evident from simply looking at the stimuli, as a slight twist of the thrower's hand before or after the real PoR can cause very significant angular distortions. The landing position of the ball also had a significant impact on viewers' judgments and, as almost all landing errors involved a shortening of the distance the ball traveled, this is consistent with the results of Hoyet et al. [63].

This experiment on the perception of PoR delays in throwing motion has demonstrated how manipulating the PoR of motion-captured throwing motions can affect the perception of those motions. We have assessed how noticeable a range of delays are for different distances (Near, Far), different throwing types (Overarm, Underarm), and different views (Front, Side). The results suggest that people are asymmetrically sensitive to early and late delays in overarm and underarm throws. Early release was not as noticeable for underarm throws as it was for overarm throws. Similarly, late releases were less frequently noticed for overarm throws.

However, this was just the first study to explore this problem and we cannot yet generalize from results with this limited number of throws and actors to the wide variety of different throwing types and styles that exist. It is important for future work on this topic to incorporate the participants' level of expertise, as being an expert is known to be linked with improved anticipatory skills to guide responsive motor actions [343, 344, 345]. Nevertheless, considering the prevalence of throwing motions in games and VR, we believe that these findings offer valuable insights to guide further studies and the development of interactive applications. Next, we present a system that performs real-time detection of the PoR virtual throws, which is informed by the results of this experiment.

4.2 Prediction of Point of Release

In this section, we describe the steps performed to develop *ReTro*, a real-time PoR detection system, including a new dataset, data preprocessing, motion feature extraction, model training, and a thorough evaluation. This section uses the words "subject" and "actor" interchangeably.

4.2.1 Dataset

The motions of six actors (3F, 3M) were captured in three pairs while throwing and catching a cricket ball between them (1679 throws in total), using the default MoCap system with 21 optical cameras at 120 fps. Both a catcher and a thrower were captured for efficiency reasons, and to identify the PoC which was later used to calculate the ground truth PoR. Each actor wore 53 body and 20 finger markers (two per finger) as before. The cricket ball, which was not tracked due to interference between the ball and finger markers, was thrown back and forth. The two actors stood between 3.5m and 4.5m meters apart and varied the force with which they threw the ball at each other. The ball's landing distances were later calculated by removing the catcher and extending the simulation until contact with the ground, resulting in a set of simulated trajectories with horizontal landing distances distributed between ca. 3.5m and 6.5m (Figure 4.6). Compared with the data captured in Exp. T1, this dataset contains a wider interval and more continuous distribution of samples. In Exp. T1, the gap between target spawn regions created a discontinuity in the samples collected (See Section 3.1.1 for details). Although target spawn positions are not equal to landing distances since people introduce motor skill errors, we assume a high correlation between them. Therefore, the better homogeneity of this dataset makes it a better candidate for generalizability. We also observed by looking at some of the motion data from Exp. T1 that throws to targets that are very close to the thrower, e.g., 1.25m-2m in the Near condition of Exp. T1, are generally very effortless and not very rich in terms of motion kinematics (See the Appendix for a visual representation of Near throws).

Different from the data capture carried out for the PoR experiment in Section 4.1.2, this data capture involves recording the audio of the capture using a mobile phone in order to capture ball catch sounds. We have decided to include audio as an additional modality because the sound of a catch provides us with a certain ground truth. The inclusion of audio capture is why we moved from a tennis ball to a cricket ball in the capture since a cricket ball is heavier and makes more sound on impact with the catcher's palms. To amplify the sound of a catch, we asked the actors to expose their palms during the catch as much as possible, and we also taped a piece of aluminum foil in an effort to amplify the sound. The foil was used in the off-hand which was only involved in catching. We used a clapperboard with markers to synchronize audio and motion data in post-processing (Figure 4.9). We further utilized the clapperboard to signal the actors to perform a throw in order to make sure that both actors are back in a neutral pose before throwing. If the throw was unsuccessful, i.e. the ball ended up on the ground, the clapperboard was clapped twice as a flag for later removal. A single clap was executed when the previous throw was successful.

The impact sound of the ball with the catcher's hand was later detected from the peaks in the audio data during post-processing and used to locate the PoC frame (Figure **??**), which was used to find the best-fitting trajectories and thereby identify the PoR audio frames.

4.2.2 Improved PoR Ground Truth Approximation

In Section 4.1.2, we described our initial approach to ground truth PoR approximation. With the inclusion of audio, we have an updated pipeline (see Figure 4.7). Same as before, in two main steps, candidate PoR frame selection and trajectory optimization, we identify the pair of best-performing PoR and PoC frames. The step of selecting candidate PoR frames is the same as Section 4.1.2 and the readers can find the details in

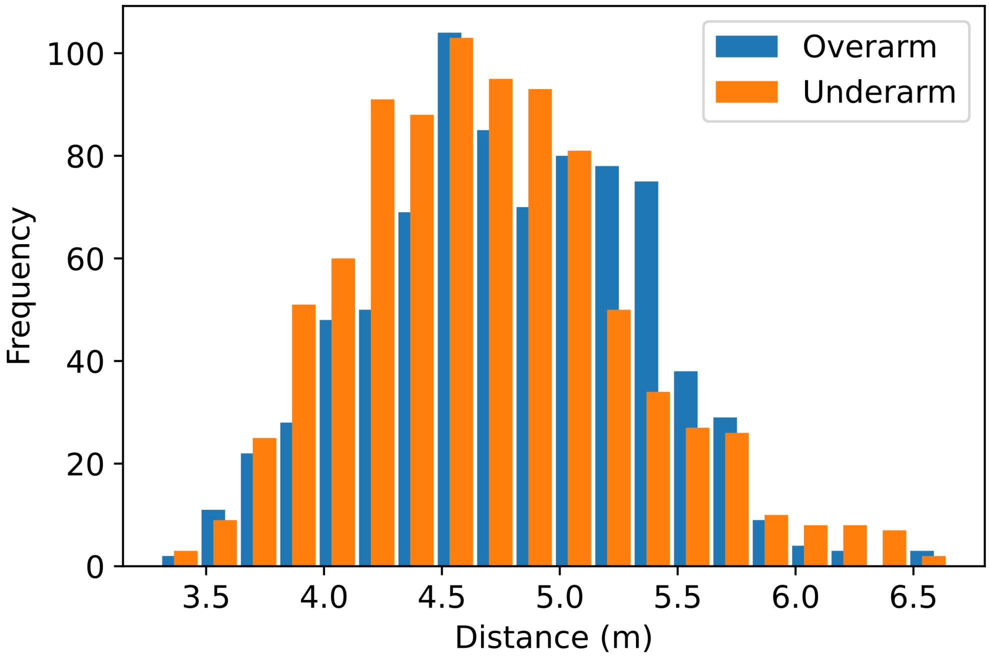


Figure 4.6: Horizontal landing distances of simulated ball trajectories for the captured throws.

that section. Here, we discuss the modified components of the pipeline.

Trajectory fitting is performed in a window of frames around each candidate PoR frame to select the best pair of PoR and PoC frames according to a distance error, which is the closest distance between the simulated ball trajectory and the catcher's hands. First, we calculate an approximate time for the ball to arrive at the catcher, i.e., airtime, if it was released at the candidate frame, giving us a candidate PoC frame. The calculation of airtime assumes the horizontal velocity of the ball stays constant. Using our prior knowledge of the PoC frames extracted from audio, we can measure how the distance between the hands is distributed at audio PoC frames. As it resembles a Gaussian distribution (μ, σ^2), we've chosen a threshold of $(\mu + 2 * \sigma)$ (see Figure 4.10). Next, we look for the earliest frame around the candidate PoC in which the distance between hands reaches this threshold. Assuming that the hands would close up further or maintain their distance after reaching the threshold, we perform a reverse trajectory simulation for the current and the next eight frames to find the nine best PoR frames for each PoC.

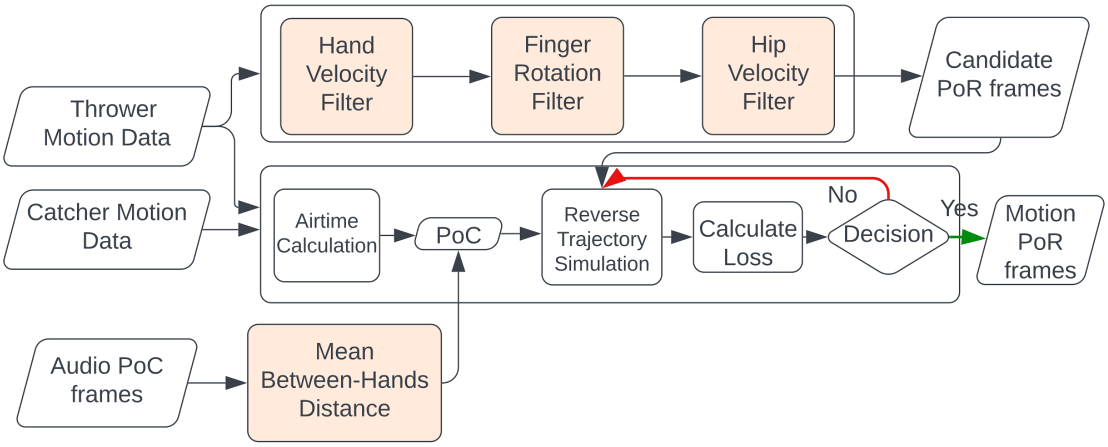


Figure 4.7: Flow diagram of improved PoR ground truth approximation.

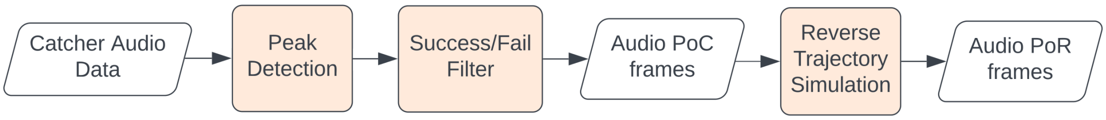


Figure 4.8: Flow diagram of audio PoC and PoR extraction.

Reverse trajectory simulation uses an approximate airtime of 0.8 seconds and searches an interval of $(PoC - 1, PoC - 0.6)$ seconds to find the best-fitting PoR trajectory. Next, we calculate the most frequent PoR frame, $PoR_{Popular}$ and the PoR frame with the minimum error, PoR_{Best} and test for the following to choose a final PoR frame from the set of nine best performing PoR frames:

- If $PoR_{Popular}$ and PoR_{Best} are the same, choose that as the final PoR frame.
- If $PoR_{Popular}$ and PoR_{Best} are different, choose $PoR_{Popular}$ as the final PoR frame.
- If there is no $PoR_{Popular}$, choose PoR_{Best} as the final PoR frame.
- If the distance error of PoR_{Best} is too large (>1), do not select a final PoR frame.

This procedure is repeated for all throws and a set of MoCap PoR frames are extracted. As a post-processing step, we check all instances where a MoCap PoR is close to, but different from, an audio PoR, in which case we use the latter since the audio signal is more reliable. Drag and rotational forces in all trajectory calculations are ignored and the acceleration of gravity is set to $9.81 m/s^2$.

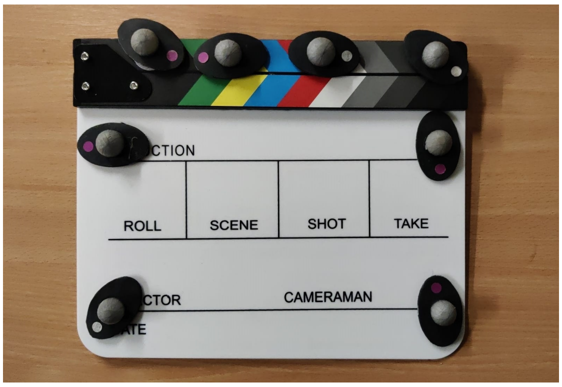


Figure 4.9: The clapperboard used for instructing the actors and audio-MoCap synchronization.

As a clear catch sound was not identified in every throw, we used an alternative method using multiple heuristics to extract the PoR frames using only motion data, following a similar approach to that described in Section 4.1.2. First, we located a set of candidate PoR frames by finding the frames where the velocity of the hand exceeds a threshold of 1.8 meters per second. To verify a candidate PoR, we looked at the mean-variance of the index and middle finger flexion over a window of 21 frames located around the candidate PoR frame. We compared it against a threshold of 10. We also checked the hip velocity to ensure that a walking motion did not cause the motion of the arms. A candidate PoR was picked when these two conditions were satisfied.

The final dataset consists of high-quality motion data of 1679 virtual throws from six different actors. The average height of the actors was 1.68m (SD = 11cm). Out of the 1679 throws, 810 (%48) are Overarm and 869 (%52) are Underarm. The throw type (Overarm, Underarm) is annotated automatically by comparing the height of the hand with the height of the shoulder five frames before the PoR for every throw.

We propose a regression model that predicts the time remaining until the PoR occurs. We use the same motion features, i.e., Position (P), Velocity (V), Rotation (R), and Rotational Velocity (RV), and the same three joints, i.e., wrist (W), elbow (E), and shoulder (S) as Section 3.2. The model is supervised on the time to the PoR frame.

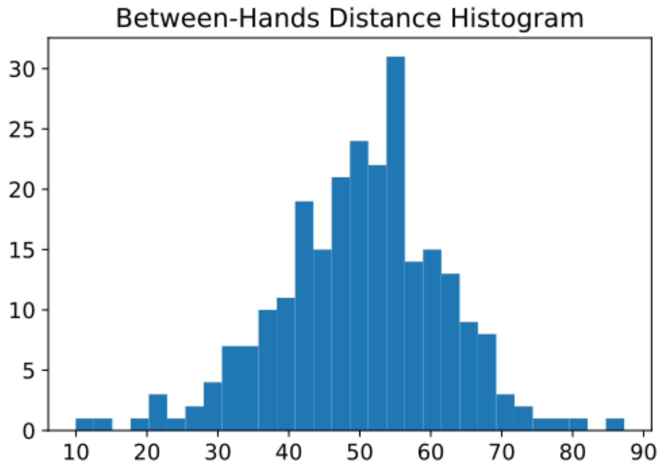


Figure 4.10: Histogram of the distance between the catcher's hand at the frame of catch, as located from the audio data.

4.2.3 Data Preprocessing

The motion capture system automatically fits a skeleton in real-time to the point cluster of marker positions. The skeleton is unique for every participant as their body and limb sizes can vary. As a normalization step for training, we scale all of the subject skeletons to the same height of 170cm. We do not scale individual joints and allow variations in bone length. This way, the model can generalize to new subjects of different heights and different bone lengths.

The motion data is originally captured at 120 fps, which is typically too high for VR and for ubiquitous tracking devices such as wearable sensors. We perform quadratic interpolation on the data to resample at different frame rates. We use 60 fps and 72 fps data depending on the evaluation mode.

We extract a window of W frames, called the *throw window* (see Figure 4.11), from each of the 1679 throws. We set W to 30 frames (≈500ms for 60fps data) and the final frame in a window is five frames after the PoR. This means the first frame of the motion is 400ms ahead of the PoR. Compared with Section 3.2 where the extracted window is symmetric around the PoR with a total size of ≈408ms, the first frame is 200ms earlier than the PoR frame. This change in the window of interest by excluding most of the

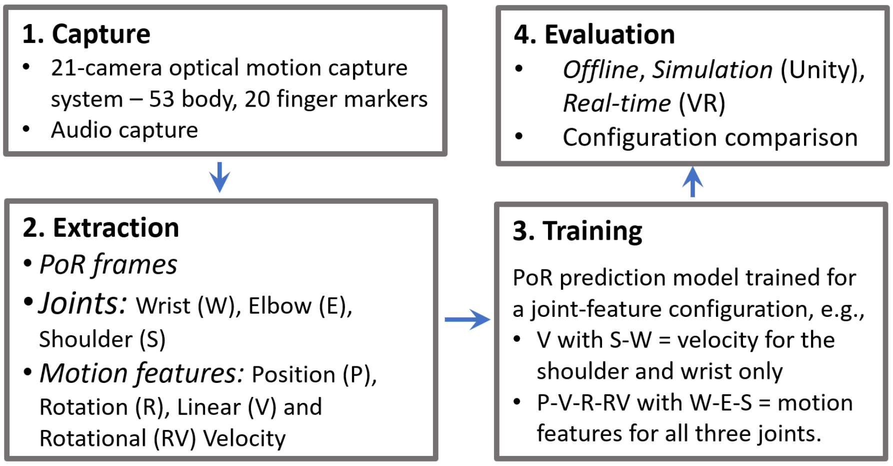

1. Capture
21-camera optical motion capture system – 53 body, 20 finger markers
Audio capture
2. Extraction
PoR frames
Joints: Wrist (W), Elbow (E), Shoulder (S)
Motion features: Position (P), Rotation (R), Linear (V) and Rotational (RV) Velocity
3. Training
PoR prediction model trained for a joint-feature configuration, e.g.,
V with S-W = velocity for the shoulder and wrist only
P-V-R-RV with W-E-S = motion features for all three joints.
4. Evaluation
Offline, Simulation (Unity), Real-time (VR)
Configuration comparison

Figure 4.11: Overview of our approach. Different from the preliminary work (see Figure 3.5), we evaluate the models in multiple ways including a real-time simulation system and a VR application.

post-release frames comes as a result of Exp T2 which showed that even delays that we initially considered as small, e.g., 50 ms, are distinguished quite easily in the context of PoR perception. As we removed the motion that is more than 50ms later than the release, we increased the starting frame from 200ms to 400ms earlier than the PoR, as a result of our increased confidence in our improved system. The final dataset includes $W * 1679$ frames of human motion data.

Next, we extract the same motion features (Position, Velocity, Rotation, Rotational Velocity) as described in Section 3.2. Finally, we generate the *target* vector for all throws. Different from the preliminary PoR classification, the target vector is the calculated duration to the ground truth PoR frame of size W, where the duration between frames is fixed to the sampling rate of motion data.

4.2.4 Model Training

In the same way as our earlier work on per-frame PoR classification (Section 3.2), all of this data is used to define a joint-feature configuration, or *design matrix* X, which is used as input to our model for training. The size of any X is specified by the number of motion features used by the joints.

Each of the motion features increases the size of the input by their dimensions. The projected positions and velocities are three-dimensional, rotation quaternions are four-dimensional, and angular velocity (magnitude) is one-dimensional. In most of our discussions, we use the model with *Full* configuration, i.e., using all four motion features from all three joints. The discussion on other configurations is to rank them based on their performance of PoR prediction. The number of rows in X is specified by the throw window size, W, times the number of throws, 1679. X has a final size of $(1679 \times W)$ by $\#columns$. The target array consists of 1679 stacked target vectors, with a final size of $1679 \times W$. For a window size W of 30 frames, the design matrix for the *Full* configuration has the final size of 50,370*33.

Before training, we normalize the features using a min-max scaler. The models are developed using Tensorflow [346]. Parameters of the network are initialized using uniform Xavier initialization [339] and we used a batch size of 256. The learning rate was initially set to 0.03 and it was decayed by a factor of 0.8 when there were no improvements. Each model was trained for a maximum of 100 epochs with early stopping. The models consist of stacked LSTM layers with dropout layers (p=0.4) in between. As a sequential model, LSTM is a suitable candidate for modeling time series data such as ours [347]. We use two timesteps as the input, which was found as the best-performing hyperparameter through trial and error. The models contain five LSTM layers with 80 units each. We use a weighted loss function to make the predictions more accurate towards the actual PoR, where y_{true} is the time to PoR and y_{pred} is the estimated time to PoR:

$$Loss = \begin{cases} \text{if } |y_{true} - 100| \leq 50 \\ \text{otherwise} \end{cases}$$

We train and evaluate models using cross-validation. Out of the six actors in the dataset, we separate one of the actors each time as the test set and train the model with the rest of the data. We additionally train each cross-validation fold five times and average the results to remove the effect of randomness.

```
Algorithm 2: Prediction algorithm - Part 1
  Input       : Prediction at time t, $Y_t$
  Output      : Release decision (True, False), $R$
                Release time (ms), $Y_{final}$
  Parameters: Detection range (ms), $[D_s, D_e]$
                Queue of size m, $Q_m$
                Fixed time between frames (ms), $\Delta$
                Prediction frequency (frame), $f_p$
                Allowed between frame difference (ms), $\delta_{btw}$
  $[D_s, D_e] = [-30, 150]$, $\Delta = 1000/fps$, $\delta_{btw} = 40ms$
  $Y_{prev} = Y_{t-1}$, $\Delta_t = 0$
  while $R$ is False do
      if $Y_t > D_s$ and $Y_t < D_e$ then
          $\Delta_t = Y_{prev} - Y_t$
          $lower = \mathrm{f}_p * \Delta - \sqrt{f_p} * 2 * \delta_{btw}$
          $upper = \mathrm{f}_p * \Delta + \sqrt{f_p} * 2 * \delta_{btw}$
          if $\Delta_t \geq lower$ and $\Delta_t \leq upper$ then
              if $Q_m$ is full then
                  Dequeue $Q_m$ and Enqueue $Y_t$ to $Q_m$
                  $R = True$
                  forall $Y \in Q_m$ do
                      $Y_{final}$ += $Y/m$
                  end
              end
              else
                  Enqueue $Y_t$ to $Q_m$
              end
          end
          else
              Empty $Q_m$
          end
      end
      $Y_{prev} = Y_t$
  end
```

Algorithm 3: Prediction algorithm - Part 2

```
Input      : Release time (ms), Y$_{final}$
             System time at release decision, $T_{rel}$
Output     : Release ball (True, False), $R_b$
             Release time (ms), $Y_{final}$
Parameters: Fixed time between frames (ms), $\Delta$
             Current system time, $T_{curr}$
             Network delay (ms), $\Delta_N$
$T_{curr}$ = 0, $Y_{final}$ -= $\Delta_N$
while $R_b isFalse$ do
    if $T_{curr} - T_{rel} \geq Y_{final} - \Delta/2$ then
        $R_b = True$
    end
    $T_{curr} + = \Delta$
end
```

4.2.5 Prediction Method

The prediction model outputs a single prediction on the remaining time until a PoR, given the motion data. As seen in Figure 4.12, the model has varying precision and bias in different regions of the curve. For a real-time evaluation of the model, we pick a specific range where it operates best, $[-30, 150]$ ms, and consider predictions that fall in that range. We choose a value for how much variation is allowed between predictions by looking at the standard deviation of the curve in the range $[-30, 150]$. We chose 40 ms as the allowed variation after testing with multiple values. We assume that the prediction follows a Gaussian distribution in the range $[-30, 150]$, and the difference between two consecutive predictions is therefore also a Gaussian distribution. By following the statistical rule that around 95% of samples from a Gaussian distribution fall within two standard deviations, we specify *lower* and *upper* thresholds of Algorithm 2.

As the model outputs new predictions, we maintain a queue of size three to make a release decision. Unless a new prediction violates the allowed variation or falls outside the allowed range $[-30, 150]$ ms, a release decision is made when the queue is filled. This phase of the prediction method is provided in Algorithm 2. Once a release decision is made, a final release time, Y_{final}, is calculated. With each update cycle, we decrement the duration of a cycle from the final release time, and initiate the release when the predicted time arrives (See Algorithm 3). We employ these algorithms for *Simulation* and *Real-time* evaluation modes.

The average inference time of the prediction model is calculated to be 30ms. Therefore, a Unity application that operates at 60 fps ($\Delta_f = 16.67ms$) can request a prediction in approximately every two frames. To account for possible performance drops, we set the prediction frequency to once in every three frames. In addition to the inference time bottleneck, the system is limited to making decisions during a frame. Therefore, if the suggested decision time is reached between frames, e.g., Y_{pred} = 8 ms when f_{t_1} = 0 ms and f_{t_2} = 16 ms, the system gives a delayed response. The amount of maximum delay introduced is half of the time between frames, i.e. 8.33 ms when the system runs at 60 fps ($\Delta_f = 16.67ms$). This is depicted in Algorithm 3.

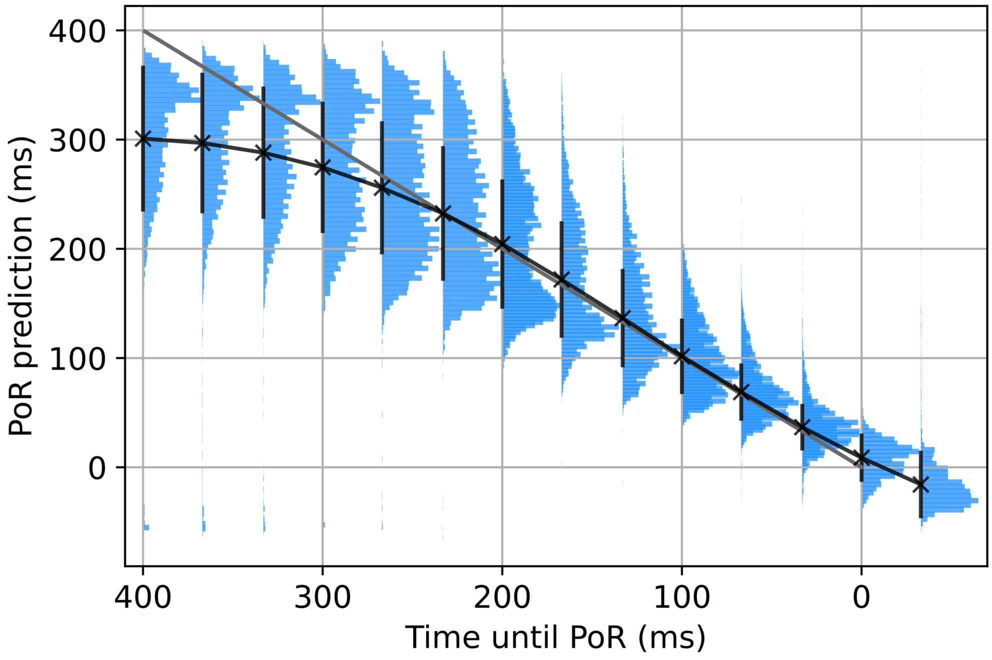


Figure 4.12: *Offline* performance of the *Full* prediction model in a window of 450ms, averaged over all the throws in the dataset. The PoR lies at $x = 0$. Error bars show standard deviation. Blue shows the distribution of all the predictions for a particular instance of x-axis. The diagonal line ($x = y$) is provided as the ideal predictor.

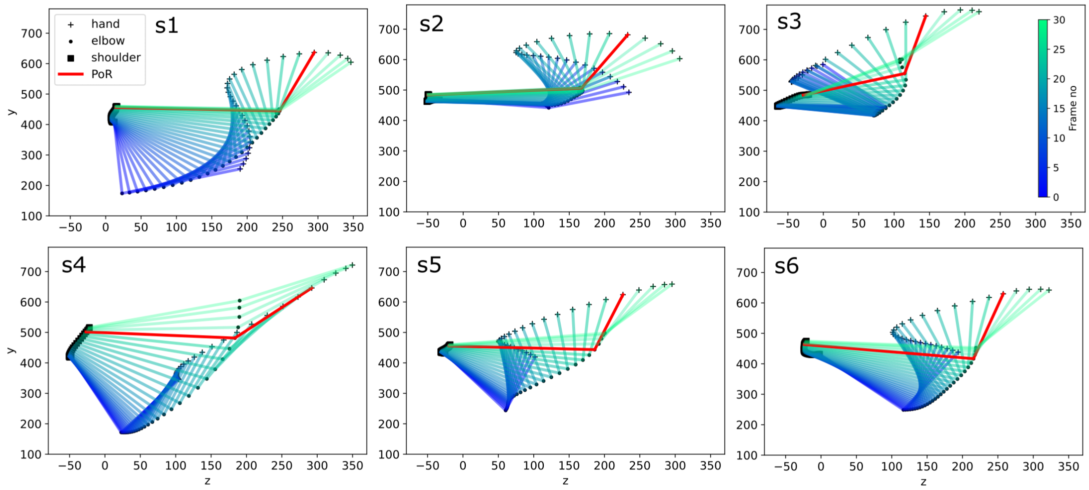


Figure 4.13: Averaged projected local positions of the three arm joints for Overarm throws in side view (y and z axes). Each plot represents one subject. The red line represents the ground truth PoR (t=25).

4.3 Evaluation of PoR Model

In this section, we present our evaluation of the prediction model. *Offline* mode evaluations were first conducted in Python without considering the delays introduced by a real-time implementation such as inference time, frame rate, and network delay. We then implemented ReTro, a real-time throwing system, using the *Full* configuration model, i.e., using all features of all joints. We first performed a quantitative evaluation in *Simulation* mode, with pre-recorded motion capture data. This was followed by a qualitative user test in *Real-time* mode using an interactive VR application.

4.3.1 Offline Mode Evaluation

Overarm/Underarm Comparison: The prediction model performs differently for Overarm and Underarm throws (Figure 4.20). Although the performances are similar when the actual PoR is less than 100 milliseconds away, the distributions of the predictions converge sooner for Underarm than for Overarm. There may be multiple possible explanations for this: it might be that Underarm throws have a more predictable motion pattern than Overarm throws, enabling a more accurate early prediction, or, there might be more style variation in the early phase for Overarm throws compared to Underarm

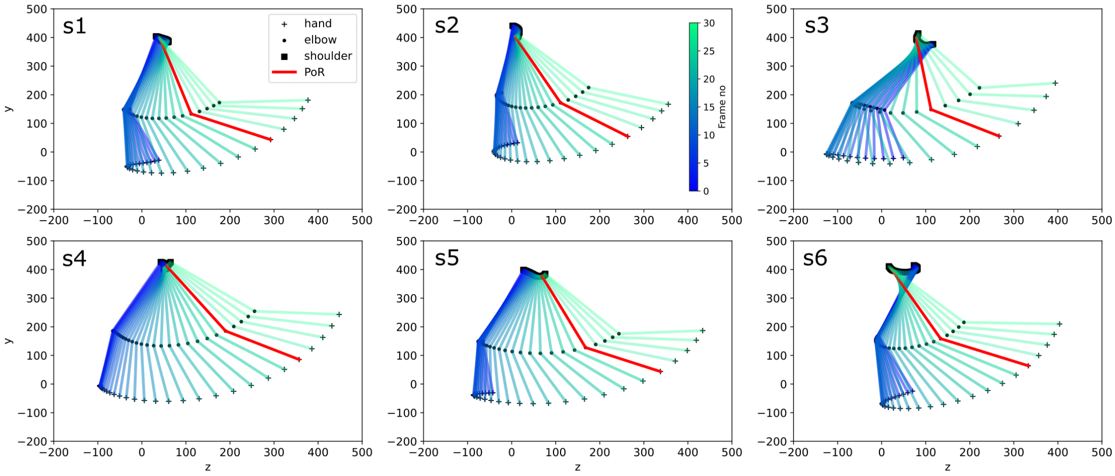


Figure 4.14: Averaged projected local positions of the three arm joints for Underarm throws in side view (y and z axes). Each plot represents one subject, e.g., s1 for subject 1. The red line represents the ground truth PoR (t=25).

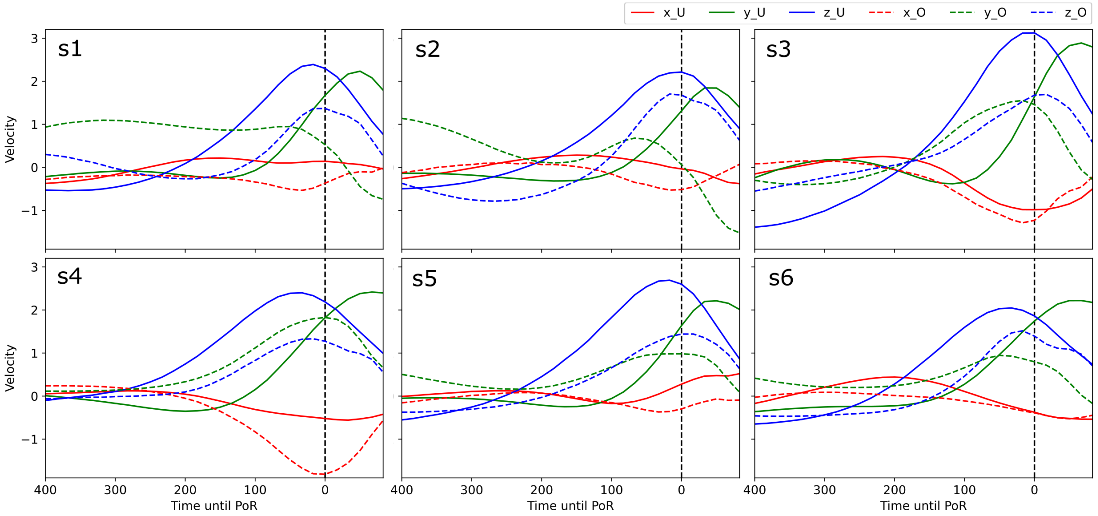


Figure 4.15: Averaged velocity (m/s) of hand joint calculated from the projected local positions, separated for each axis (X, Y, Z) and throwing type (Underarm and Overarm, denoted by U and O, respectively). Each plot represents one subject, e.g., s1 for subject 1. The window size is 30 frames and the vertical line represents the PoR (t=25).

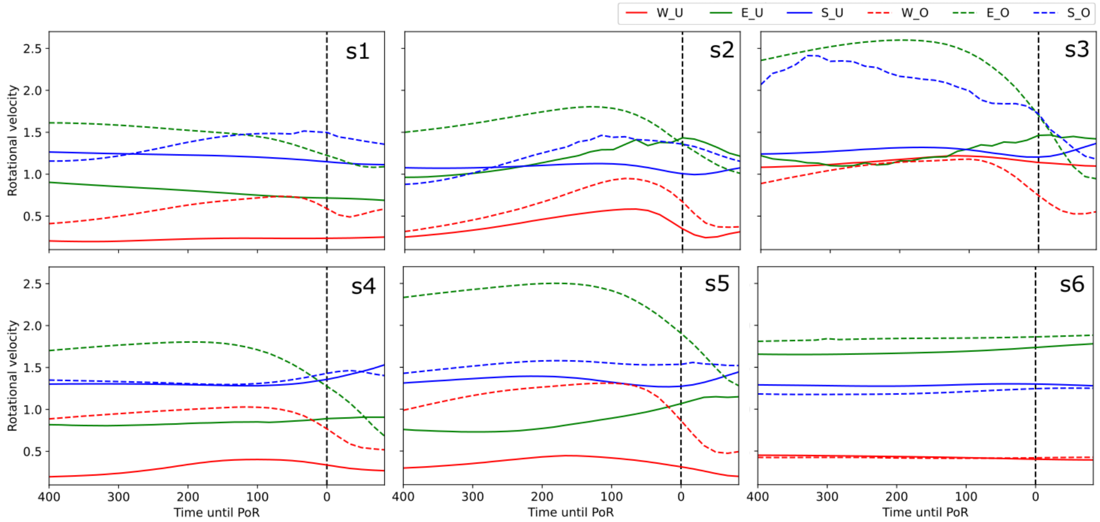


Figure 4.16: Averaged rotational velocities (°/s) calculated from joint rotations, separated for each joint (W, E, S) and throwing type (Underarm and Overarm, denoted by U and O, respectively).

throws, requiring more actors in the dataset. In Figure 4.17, a difference between subject performances is observed based on throwing type. In Underarm, two groups (S1,S2,S3) and (S4,S5,S6) perform similarly, which could be due to a similarity in throwing styles or joint length ratios for the subjects in each group. For Overarm, there is more variation between subjects in the early region, possibly due to differences in style.

To understand the underlying differences between Overarm and Underarm throws, we visualized the average throwing motion in projected position space, separately for Overarm and Underarm throws (Figures 4.13 and 4.14, respectively). Although these figures do not display the x-axis of the data, the motion mainly takes place on the y- and z-axes in the projected position space. In these figures, we see that almost every actor has their unique style of performing an Overarm throw, with certain actors starting the motion with their elbow lower and others higher. Yet, despite having many different starting poses, the motions converge to a more similar pose, which can be seen by looking at the PoR lines. On the other hand, Underarm throws contain many similar patterns throughout, which shows why the model is able to start making accurate predictions early on.

To further compare actor motions, we look at averaged velocity and averaged rotational velocity patterns shown in Figures 4.15 and 4.16. The average velocity for an actor is calculated by averaging the velocity that is calculated from the projected local position over the *throw window*. The average rotational velocity for an actor is calculated by averaging the rotational velocity values over the *throw window*. It is important to note that both of these calculations are carried out on retargeted motions. This has different implications for linear versus rotational velocity: In linear velocity, rescaling of the limb sizes directly affects the velocity of the hand since the distance covered by the hand is decreased or increased depending on whether the actor is taller or shorter than the retargeted body, respectively. On the other hand, angular velocity is not affected by this transformation, as a joint's rotational motion remains unchanged.

As the values do not reflect the actual measurements for linear velocity, we mainly compare their patterns. In Figure 4.15, we see that while early phases of the motion display high interpersonal differences in mainly the y-component, the z-component of the motion follows a similar pattern in both Overarm and Underarm throws. Furthermore, the pattern displayed in the Underarm throws for forward and upward motion is much more similar between actors compared to Overarm throws, which also supports our claim that there is more style variation in Overarm motion. Looking at rotational velocities in Figure 4.16, we conceivably see that the wrist accommodates the lowest angular velocity among the arm joints except for s4. Moreover, we can observe the heavy involvement of the elbow joint in Overarm throws compared to Underarm throws. Overall, rotational velocity patterns are less interpretable than velocity patterns.

In Figure 4.20, there are more outliers in the prediction distributions for Overarm throws in the early region (>200ms). A negative prediction at an early stage of a throw means that the model is not fully capable of differentiating the "follow-through" phase of a throwing motion from the "preparation" phase it. In *Simulation* and *Real-time* modes, we work around this by incorporating a velocity threshold of 1.8 m/s similar to how it was used for PoR extraction in Section 4.2.3. The model starts making predictions once the velocity threshold is reached during a throw.

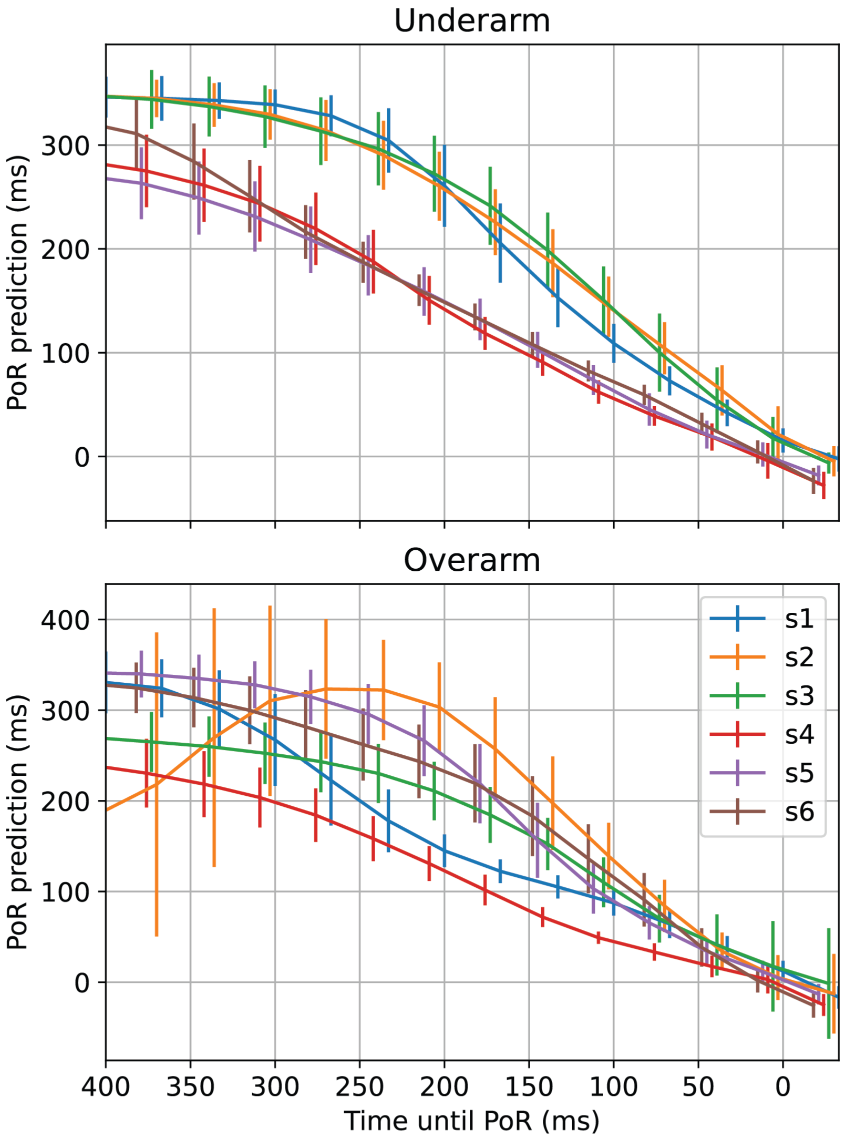


Figure 4.17: Per subject performance of *Full* configuration in *Offline* mode separately for Overarm and Underarm. Error bars show standard deviation.

Configuration Comparisons: For a thorough evaluation, we train models with different joint-feature configurations and evaluate them in *Offline* mode, i.e., by comparing only the curves without a prediction algorithm or any consideration of the delays introduced by the system. The aim of this offline evaluation is to explore the feasibility of building the system with other equipment and to determine the relative importance of joints and motion features for this task. We also aim to compare the new results with the prelimi-

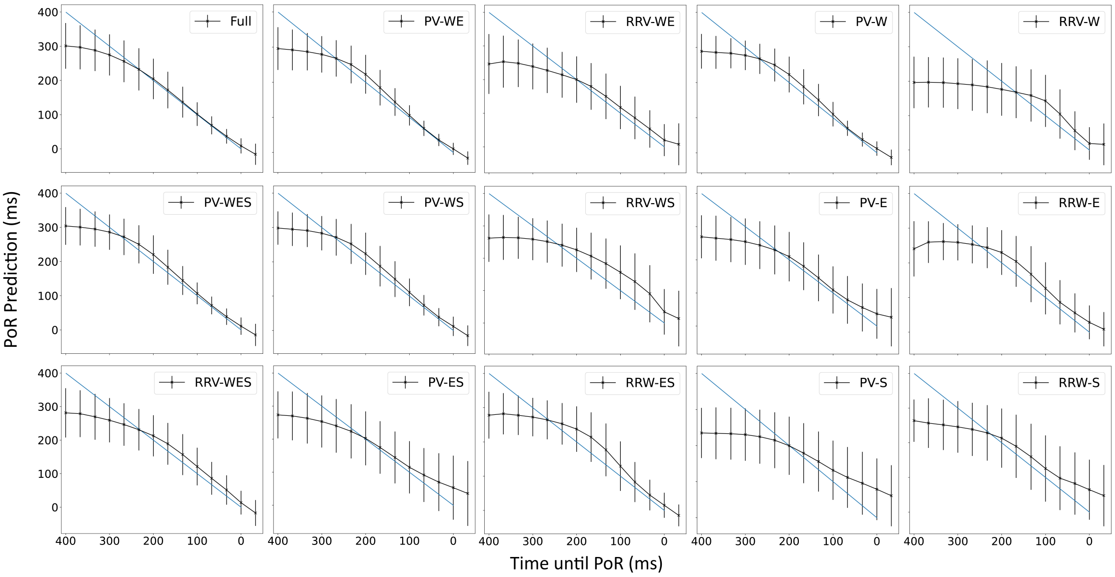


Figure 4.18: Each plot shows the *Offline* results of a model trained with a different joint-feature configuration. The first and the second parts of a label represent motion features and joints used, respectively, e.g., PV-WE is Position and Velocity extracted from Wrist and Elbow joints. W,E,S=wrist, elbow, shoulder; P, V, R, RV=position, velocity, rotation, rotational velocity. The diagonal $x = y$ line is provided as the ideal predictor.

nary work of Section 3.2 (see Figure 3.8).

For comparison purposes, we pick the joint-feature configurations with the assumption that P/V and R/RV features are available simultaneously in pairs, same as Chapter 3. We train the same neural architecture that was described in Section 4.2.4. It is possible to further improve the performance of each configuration by tuning the network architecture and the hyperparameters individually, but we do not pursue that here.

We compare the results of 15 key configurations in *Offline* mode trained with 60 fps data. For convenience, we refer to a configuration by writing motion features followed by joints, e.g. PV-WE configuration is Position and Velocity extracted from Wrist and Elbow joints. In Figure 4.18, configurations on the first column show that configuration RRV-WES, i.e., rotational features from all joints, performs significantly worse compared to configuration PV-WES, i.e., positional features from all joints, which perform almost similarly as *Full* configuration.

The inclusion of Wrist is the most critical for the performance of a model that uses Position and Velocity (Columns 2 and 4). Shoulder on its own contains the least information about the PoR when positional information is considered. A comparison between configurations PV-WES and PV-WE shows that the inclusion of Shoulder does not improve the performance of the model. This is also valid for a comparison of PV-WS and PV-W.

When rotational motion features are considered (Columns 3 and 5), Elbow stands out as the most important joint, followed by Shoulder and lastly Wrist. A comparison of joints that use positional features versus rotational features (Columns 2 versus 3, Columns 4 versus 5) shows that none of the rotational configurations reach the same level of performance as the positional configurations except RRW-ES versus PV-ES, which is quite intuitive since a shoulder joint typically rotates without moving too much in a throwing motion. Yet, in a Shoulder only configuration, RRW performs almost as poorly as PV.

4.3.2 Simulation Mode Evaluation

In this mode, we evaluate the *Full* configuration model by simulating a real-time throwing system using pre-recorded animation clips exported from the dataset. To create animations from the recorded motion, we sample the data at 60 fps and retarget the

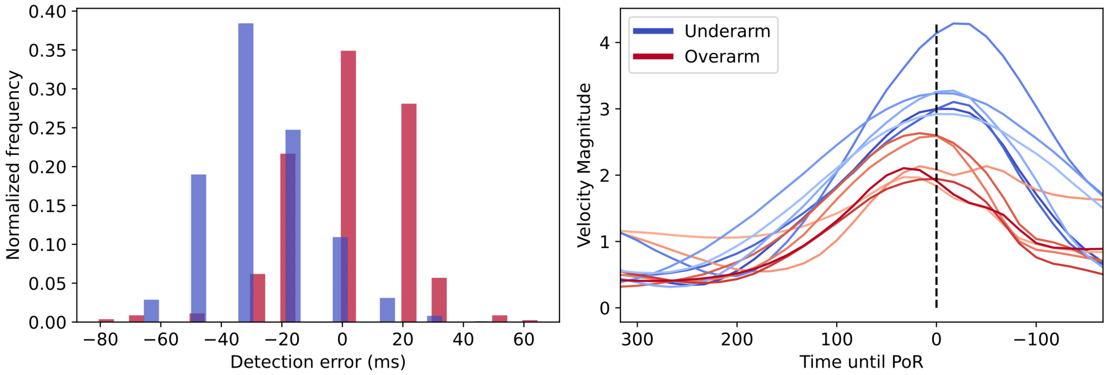


Figure 4.19: Detection error histogram (Left) of a heuristic-based detection algorithm that uses velocity magnitude (m/s), calculated on all throws in the dataset. Overarm: mean=0.08, stdev=19.96, Underarm: mean=-27.46, stdev=18.99. Velocity magnitudes (Right) are separated for each subject.

entire dataset onto a template skinned model provided by Vicon. 60 fps is generally considered the ideal frame rate for a game [348]. We rescaled the template model to a height of 1.70m, which is what was used for training. After importing the dataset as animation files into Unity, a window of one second is extracted from each throwing motion, with the PoR lying at t=0.45s.

For each actor in the dataset, we create an animation controller with their throwing motion clips and run the entire sequence in Unity. As the clips are played, motion features from arm joints are calculated, normalized, stacked, and sent to the model as input. Unity uses a left-handed coordinate system, so we perform necessary transformations on the motion features to make them match with the motion features of a right-handed coordinate system.

The model makes real-time predictions and a throw is registered based on a simple prediction algorithm (See Algorithms 2 and 3). To access the PoR prediction model, we use NetMQ, which creates a lightweight connection between Unity and Python. The model runs in the background and processes an input upon request by Unity. We use the same evaluation method used for training: when motion from a specific actor is being evaluated, the model that separates that particular actor as the test set is used. The target frame rate of Unity is set to 60 fps to match the data rate that was used to train the model.

Using the *Simulation* predictions of Figure 4.20, a distribution of detection errors is measured in Unity using the prediction algorithm (see Figure 4.21). The average detection error is 10.85ms for Overarm and 2.72ms for Underarm, with most of the detections lying within a 50ms delay. The number of undetected throws is one out of 1679 throws. We refer to the *Real-time* mode to explore how these delays are experienced by users in VR.

4.3.3 Real-time Mode Evaluation

We test ReTro in real-time in VR by streaming live motion data of new users from the motion capture system into Unity. Throws are performed by interacting with a virtual ball using the hands only. A separate motion capture computer processes the motion data in real-time and sends relevant information to Unity using Vicon Datastream SDK. A VR application built in Unity is displayed to the users on an Oculus Quest 2 headset tethered with Oculus Link. We align the motion capture and VR coordinate systems using an alignment tool provided by Vicon.

Since this evaluation method is tested with new users that are not in the dataset, we re-train the *Full* configuration model without separating any actors as the test set. The Oculus Quest 2, which enforces a minimum frame rate of 72, is used. As the frame rate of the model, i.e., 60 fps, no longer matches the system's frame rate, i.e., 72 fps, we resample the motion data and train the model with 72 fps data.

Initially, we attempted to stream the entire skeleton of the user and render the full body, but we eventually only rendered a hand mesh to decrease the amount of computation. We stream motion information of five joints into Unity: hips, upper spine, shoulder, elbow, and wrist. Shoulder, elbow, and wrist are the joints whose extracted motion features become inputs to the prediction model. The upper spine is the parent joint of the shoulder, and we use it to construct a hierarchy and read the local rotation of the shoulder. Hips are needed in order to rescale the user's body to a height of 1.70m which is what the model is trained with. Since the global position is streamed continuously, rescaling has to be repeated every frame.

The frame rate fluctuation amount is much larger than what it is for *Simulation* mode

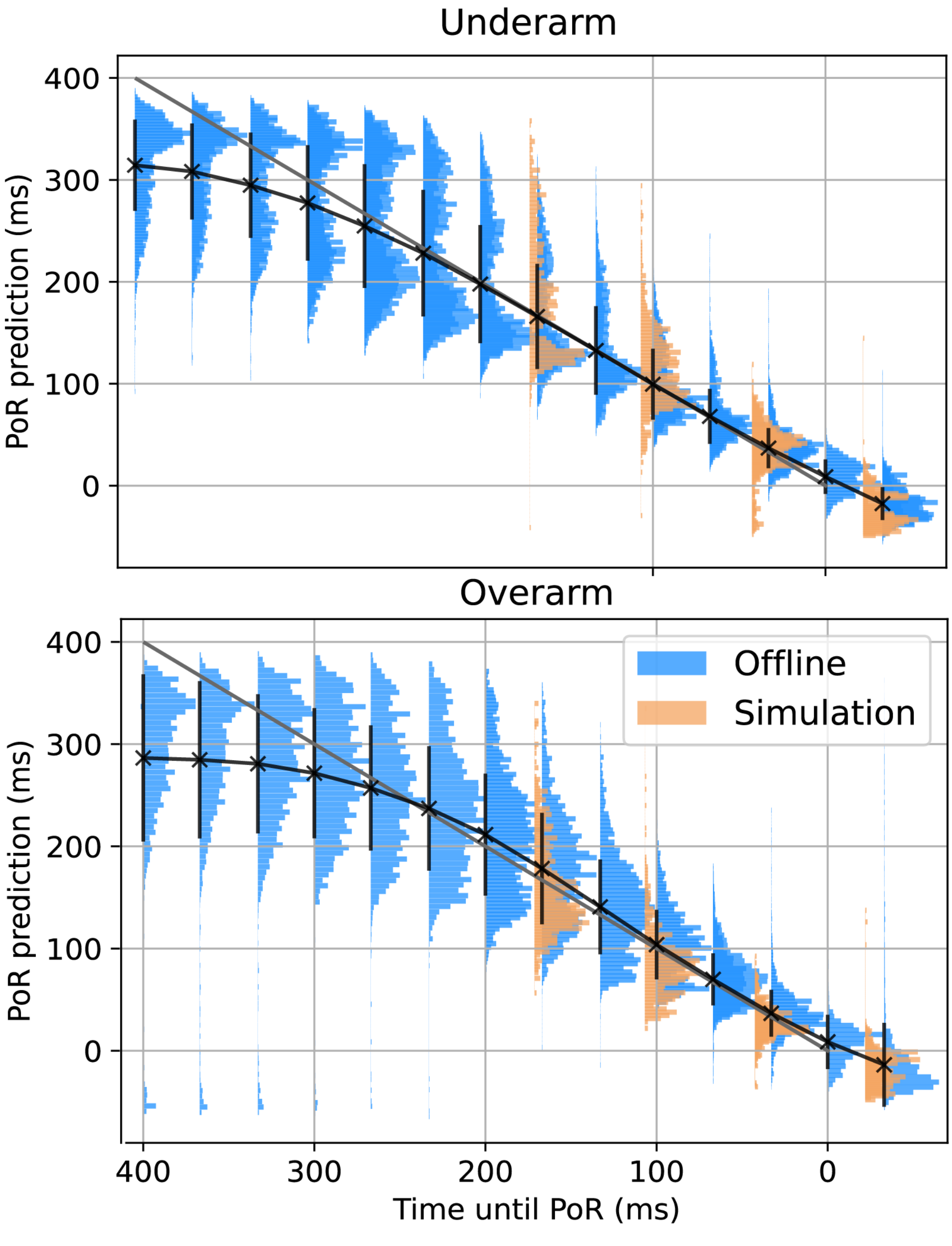


Figure 4.20: *Offline* and *Simulation* evaluation of regression model for the Full configuration for Overarm and Underarm.

due to the introduction of VR and motion capture data streaming. Since the models are trained with data sampled at fixed timesteps, it is possible that these fluctuations have a negative impact on the performance of the model.

Six volunteers used ReTro and provided feedback (5M, 1F, ages 18-32). The VR scene included a ball 7cm in diameter, a stand for the ball, a target to throw at, and a floor plane. Two displays placed in the scene showed the type of throw to perform, i.e., overarm or underarm, and the instantaneous frame rate. The users were instructed to pause throwing if they noticed large fluctuations in the frame rate. They could see their right hand but their fingers were not tracked. In every throw, the user grabbed the ball from the stand in from of them before performing a throw. A grab was triggered when the hand collider made contact with the ball collider, upon which a pre-recorded grab animation was played.

The users first practiced both underarm and overarm throws freely as a warm-up without any restrictions. This was followed by two blocks of throws in counterbalanced order, one each for overarm and underarm motions. In each block, they aimed at a spawned target using either all overarm or all underarm motions for nine throws, followed by a block using the other type of throwing motion. The targets incrementally moved away from the user after every three throws, giving three horizontal distances of 3.25m, 4m, and 4.75m. After a thrown ball made contact with the ground, the target was colored green or red to indicate a hit or a miss, respectively.

After completing every nine throws in a block, the users were asked for feedback on the type of throw they performed. At the end of the session, we asked them to compare their experiences with underarm and overarm throws and to comment overall on the system. All six users reported that Overarm throws were much less realistic than Underarm throws because too much force was needed for the ball to reach the target and the release of the ball felt too late. Even though the performance of the model in Overarm and Underarm PoR detection does not differ too much, the negative experience of the participants in Overarm throws suggests that the ground truth for Overarm throws might be biased.

Only two reported that Underarm release times felt slightly late, but all six agreed that this task was easier and felt much more natural or fun. Two participants suggested that

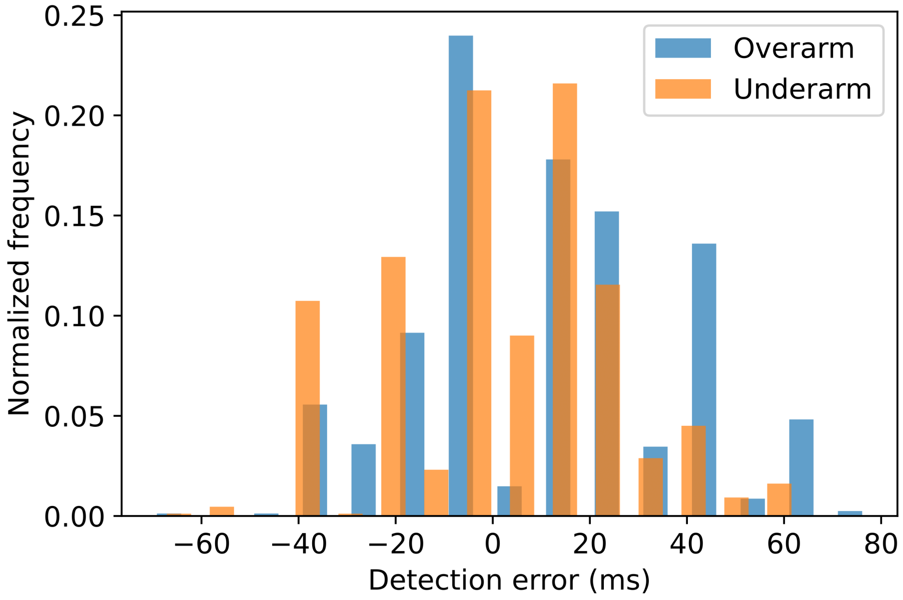


Figure 4.21: Detection error histogram in *Simulation* mode for Overarm and Underarm throws. Calculated on pre-recorded motion clips. Overarm: mean=10.85, stdev=25.87, Underarm: mean=2.72, stdev=23.04.

seeing a fully tracked hand would enhance the sense of embodiment, but did not think that this would affect their performance. Regarding the overall experience, these are some of the comments received:

- *"If the option was between, say, having a controller in your hand and not being able to throw it, and not having a controller and more naturally performing the throw, I would prefer this."*
- *"It feels more immersive compared to holding a controller. It feels more real. You don't really feel like it's VR, it's just your body moving."*
- *"Because you can't throw a controller, I think this is better, much more natural."*
- *"If I had a controller for this, I would be using a button to release. This would give you more control over the ball, but less natural. If you're aiming to get the ball at the correct area, control is more important than naturalness. So, in a result-oriented task, control would be better I think."*

4.3.4 Comparisons with other evaluations

Comparison with Preliminary Results: It is necessary to compare the results of this evaluation with the results of our preliminary work on PoR detection in Chapter 3.2, where a similar evaluation of 35 different configurations was carried out.

In the preliminary assessment (*Comp*-1), differently from the current assessment (*Comp*-2), we found that Rotation feature displayed better performance than other features in several ways. First, when used on a single joint, Rotation performed better than the other features alone. Second, the configuration using the Rotation feature on all joints was a top performer. Several factors contribute to this mismatch between the two findings, which are the differences in the datasets, the preprocessing steps, and the employed ML models. The dataset used in *Comp*-1 was captured with a heuristic-based virtual throwing algorithm operating on finger rotation, which may have failed to capture any underlying correlations that occur in real throws between our motion features and PoR. We stated in Chapter 3 that some participants of Exp. T1 have experienced difficulties controlling the virtual ball, affecting both their performance and motion. Furthermore, participants were not restricted to a throwing type, which led to the dataset being dominated by overarm throws (%64). Lastly and most importantly, in Chapter 3, we did not implement rescaling of the subjects' skeletons as a technique to better cluster motion features of all actors (See the Appendix for the projected position features of Exp. T1), and to compensate for this lack of size invariance, we did not perform cross-validation during training. In Figure 3.9 in Chapter 3, the performances of different detection methods for *Full* configuration are provided. DR50, the best-performing real-time method, has an average (non-absolute) error of ∼10ms with 16.66ms delay introduced by DR method, which sums up to a detection delay of 25ms with around 5% of the throws undetected. On the other hand, the average detection error demonstrated in *Simulation Mode Evaluation* (Sec. 4.3.2) is 6.62ms with only one undetected throw. It is also important to note that the preliminary model was trained on 120 fps data while the current model was trained on 60 fps data. Although our transition from a feedforward neural network architecture to LSTMs might have contributed to this improvement, we think that the dataset and the motion rescaling step have the biggest impact.

Comparison with heuristic-based solution: In the study by Butkus et al. [18], virtual throwing is implemented without a controller by using velocity as a heuristic for releasing the ball. In this approach, the ball is released when the hand velocity, tracked by a Vive tracker attached to the hand, starts decreasing. In our analysis of different configurations, we found that configurations using PV features on Wrist are noticeably more accurate, especially toward the PoR. Therefore, we decided to go back to evaluate the heuristic-based release algorithm to understand how it compares with our approach. As the manuscript [18] does not provide the details of their implementation, we implemented the heuristic using the magnitude of global velocity. In a controlled setting where subjects perform throws in a predetermined direction as in the case of [18], using the forward component of the global velocity can also be feasible.

In a typical overarm or underarm throwing motion, the arm performs a swinging motion by first moving backward and then forward with respect to the throwing direction. Such transitions of motion direction must be considered delicately when designing an algorithm based on velocity so that a release is not triggered in the early phases of a throwing motion where velocity changes are non-monotonous. Due to this, in our evaluation of the heuristic-based algorithms, we start our PoR search 5 frames before the ground truth PoR frame. On the other hand, by training a model, it is possible to not only recognize these motion segments as part of the throwing motion but also to generate accurate predictions about the PoR in the early phases.

In Figure 4.19, results of the velocity magnitude-based algorithm are presented as detection error histograms. The detections are centered around the origin for Overarm throws but slightly shifted for Underarm throws, i.e., 27.46ms late detection on average. This can also be inspected by looking at the averaged velocity magnitude curves per subject on the same figure. For Underarm throws, there is a clear offset between the peak average velocity magnitude and the curves closely resemble second-degree polynomials, but for Overarm throws, the curves do not follow very similar trends, with two local maxima in a few of them around the ground truth. Although the reasons for this are not evident, two possible explanations are the ground truth and style variation. It should also be noted that these calculations are carried out on the motion data that was scaled up/down to a skeleton height of 170cm, and therefore the visualized magnitudes are not equal to the velocity magnitudes of the actual throws.

4.4 Discussion

In this chapter, we first presented an experiment to explore the perceptual consequences of PoR timing delays. An improved PoR prediction model was also presented, along with a real-time PoR detection system, ReTro, that takes as input the motion features extracted per each arm joint and outputs a prediction of the PoR time in real-time, were presented. We thoroughly evaluated the model both offline and in real-time. We also compared 15 different joint-feature configurations to understand how motion features contribute to the task of prediction point of release. We found that the wrist is the most informative joint and P/V features are generally superior to R/RV features. These results differ from our analysis of different configurations in Chapter 3, which we think is mainly caused by the use of less reliable PoR ground truth in the earlier chapter. The current evaluation can be useful for optimizing the effectiveness of wearable motion sensors in a system like this.

After finding that P/V features are very effective on the wrist joint for PoR detection, we implemented a velocity-based heuristic release algorithm following the work of [18]. The results of this analysis showed that such a heuristic performs comparably well, which might give the impression that our system does not provide much gain in return for the complexity of the setup. However, given the simplicity of the throwing motions evaluated in this work, we argue that our approach will be able to perform far better on more complex and faster motions, e.g., chucking in cricket, where the capacity to learn and the value of early detection increases.

Our real-time VR implementation of the PoR detection model using a motion capture system enabled users to interact naturally with a virtual ball without any intermediary device, i.e. a controller. Six volunteer users reported that underarm throws generally felt very accurate and natural, while overarm throws felt delayed and less natural. Four users reported that using their hands rather than a controller was much more natural and realistic. This novel physical interaction method could be adapted to other physical interactions that could benefit from an early prediction model.

Throwing a real ball is highly influenced by the haptic backforce exerted, and it has been shown that when a throwing motion without a ball in the hand is made, fingers are opened much less rapidly [220]. The inclusion of a haptic device, such as PIVOT [4], could therefore be helpful in emulating this experience. Furthermore, it is possible that the use of a real ball during motion capture may have affected release detection in the virtual ball scenario, which needs further study. The model was less robust at predicting overarm throwing release points according to the users, so more improvements are needed in this case. As this qualitative feedback does not match with the quantitative results, i.e. overarm and underarm sets perform similarly in Simulation mode (see Figure 4.21), and the velocity magnitudes display an inconsistent pattern for Overarm throws (see Figure 4.19), it suggests that the set of ground truth for overarm throws may not be as accurately approximated as for underarm throws, and an improved ground truth extraction approach can help with this, e.g. pressure sensors in the palms.

In future work, we are interested in exploring TensorRT to reduce the inference time of our model, which is currently the main computational bottleneck of the system. It is also of interest to explore more advanced prediction algorithms and to train the model with data of varying timesteps rather than of fixed timesteps so that it will be more robust to fluctuating frame rates. Further ideas are suggested in Chapter 6.

5 Perception of Dumbbell Lifting

In this chapter, we study "physicality errors", i.e., the errors that arise due to a mismatch in the dynamics of a person's motion and the visualized movements of their avatar in VR, by employing the scenario of dumbbell lifts. In a series of experiments, we ask participants to watch humanoid models performing pre-recorded dumbbell lifts and respond to forced-choice and Likert scale questions.

We preferred using dumbbell lifts for this evaluation because it is a simple physical activity that is well understood by the general public and can easily be performed with different levels of resistance. We investigate the perceptual impact of both the kinematic signal (i.e. the motion) and varied visual signals (the size of the virtual character, the size of the lifted object, and the presence of muscle deformations that convey strain) through a series of experiments. In each experiment, people watch a series of animations in virtual reality (VR) of virtual characters lifting dumbbells and estimating the effort and weight of the action. In some experiments, they additionally rate naturalness and one experiment employs an interval forced-choice design to evaluate if fake lifts can be distinguished from real ones. The characters are driven by the recorded motion of two average-strength male actors and two strong male actors. Character bodies are matched to the actors, with the strong actors being taller and heavier. This allows multiple combinations of body type, motion kinematics, and displayed weight, along with a blendshape model to support realistic muscle deformations.

Actor	Max Dumbbell Weight	Age	Weight (lbs)	Height	Dominant Hand
Avg. Male 1 (AA1)	27 lbs	24	139	5ft. 9in.	R
Avg. Male 2 (AA2)	35 lbs	38	191	5ft. 9in.	R
Strong Male 1 (SA1)	60 lbs	30	237	6ft. 1in.	R
Strong Male 2 (SA2)	60 lbs	27	231	6ft. 2in.	R

Table 5.1: Performers in weight lifting task.

Table 5.2: Summary of the Experiments

#	Name	Deformations	Body	Dumbbell	Medium
1	Baseline	No	Actor Matched	Not Shown	VR
2	Body Shape	No	All Four Actors	Motion Matched	VR
3	Dumbbell Size	No	Actor Matched	All weights for actor	VR
4	Muscle Strain	Yes	Actor Matched	Not Shown	VR
5	Discrimination w/Strain	Yes	Actor Matched	Matched and mismatched	VR
6	Baseline	No	Actor Matched	Not Shown	Online

5.1 Methods

The study was organized into a sequence of experiments (Table 5.2) that explored various signals of physicality (kinematics, character representation, object representation, and muscle deformations). All experiments followed the same basic design, which will be described here.

5.1.1 Experimental Design

During each experiment, participants were recruited for a single session in which they completed a survey in VR. After an orientation on using a VR headset and the controls employed in the survey, they were given brief instructions on the experiment in the headset. They were then shown a range of clips that indicated the type of variation they might see in the experiment. This avoids learning the stimuli range taking place during the first few trials. After this, they saw a randomized sequence of short motion clips that showed a single person lifting a dumbbell in a simple room. After each clip, participants were asked to rate the clip in terms of the perceived Effort of the lifter, from 0 to 100% where 100% represented their maximum effort; their perception of the Weight that was lifted, from 0 to 100 lbs (they were told that weights may not span the entire scale) and, for Exps. 2-4, the Naturalness of the motion by rating the statement “The motion in

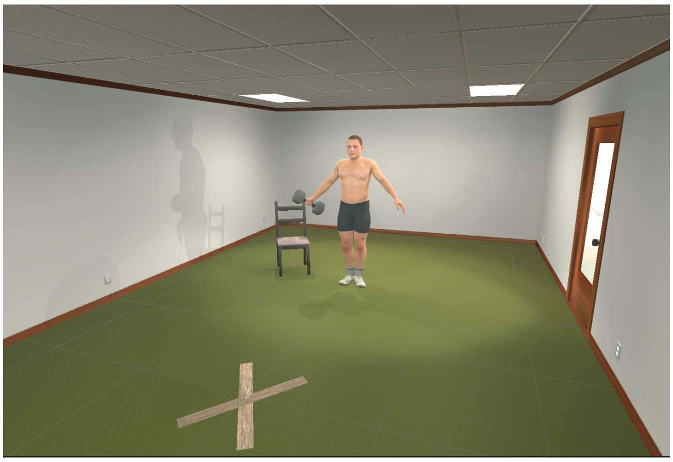


Figure 5.1: Participants stood on an X on the floor in front of the lifter throughout the experiment

this clip appears natural" on a 7-point Likert scale ranging from Strongly Disagree to Strongly Agree. At the end of the experiment, they participated in a brief exit interview and debrief.

5.1.2 Apparatus

The experiment environment was developed in Unity and presented on an Oculus Quest 2 headset. This headset has a resolution of 1832×1920 pixels per eye with around 90° horizontal and vertical Field of View. The virtual environment (VE) was a standard room with a door, several lights, and several power outlets on the walls. It contained a chair that acted as a scale marker. An X was placed on the floor to make sure that every participant observes the stimuli from a same distance of about 1.5m (Figure 5.1). The stimuli were presented fully facing the X mark, in order to maximize the amount of information conveyed to the participants. Participants interacted with the VE using an Oculus Touch controller. A visible ray coming out from the virtual representation of the controller was used to control interface widgets.

5.1.3 Stimuli

The lifters that provided the source motion for the study were recruited through online advertisements to a pool of actors. The actors submitted their maximum dumbbell

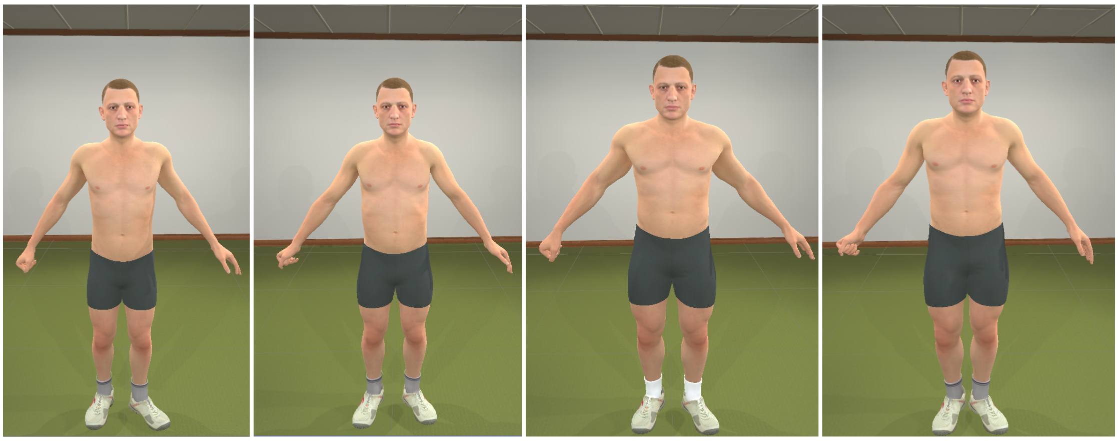


Figure 5.2: The models for each of the four performers, AA1, AA2, SA1, SA2.

curl and a short video of themselves performing a lift. The selection was based on establishing two sets of lifters, two “strong” actors and two “average” that had clearly differentiated maximum lifts. This allowed us to examine the impact of both body size and lifter strength on observations of dynamics. Details on the selected actors are summarized in Table 5.1.

To establish their actual maximum lift, all actors came to the studio at least a day before the actual motion recording. They started by lifting what they estimated would be their maximum lift. The weight was gradually increased as long as they could complete three repetitions of the lift. They were given time to rest between each set. The process stopped when they decided they had reached their maximum, which was recorded for use during the motion capture session.

The motion of each actor was recorded using a Vicon, marker-based optical motion capture system featuring forty 16MP cameras. A standard marker set was used consisting of 67 markers. Each actor performed lifts at 0 (holding nothing), 25, 50, 75, and 100% of their maximum lift. They performed a single set of three lifts at a given weight before resting. The order of weights was randomized for each actor to avoid any fatigue patterns. Two sets of lifts were recorded for each weight for each actor. The motion capture data were solved to a skeleton using a custom solver that minimizes the root mean square error over the marker set.

5.1.4 Model and Deformations

The virtual character model needed to support two research goals: allow variation in body shape and provide plausible muscle deformations. It was beyond the scope of the project to create and validate a full simulation model of human muscle. Instead, we employed an artist-driven approach whereby an artist with twenty years of experience in the visual effects industry generated a model and set of blend shapes to control deformations. The muscle deformations used in Exps. 4 and 5 were designed to show a high level of strain, rather than being tuned to each lift. This allowed us to investigate if strain cues are impactful, but further work would be required to tune deformations to arbitrary lifts. The effectiveness of the model deformations for conveying strain was validated (Section 5.2.4).

The base avatar model was generated with Human Generator V2, a tool for creating fairly realistic human models with varied body shapes in Blender [349]. Three types of blend shapes were used on the model (Table 5.3): *Body Type Shapes* - were used to create a variety of body shapes to account for various amounts of muscle mass and belly fat. These included a thin model, a very muscular model, and a "bellyOut" shape that indicated a large amount of fat around the midriff (used for AA2, who was somewhat stockier). The motion capture solver automatically scales the skeleton limb lengths based on a range of motion recording. The model was further fit to each actor by adjusting these blendshapes, using both the markers and actor footage as reference (Figure 5.2).

Corrective Shapes were created to preserve volume and improve anatomical detail as the model moved through a range of motion. These shapes were driven by the joint angle of a related skeletal joint with the one exception of the "latIn" shapes that were used to prevent penetration of the arms muscles with the latissimus dorsi. These 'latIn" shapes where driven by the distance between the elbow and the side of the body, and provided a simulation of interaction between these surfaces.

Tense Shapes were a collection of shapes built to emulate muscle strain and physical exertion (Figure 5.3). The limited range of motion in the study and the similarity of the animation cycles allowed these shapes to be grouped into regions. This reduced dimensionality made for easier retargeting to the animation clips.

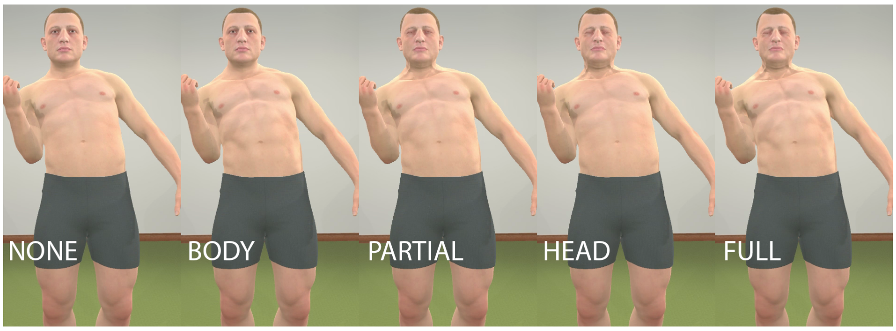


Figure 5.3: Muscle deformations are shown on virtual character SA1 for an intense moment in a lift with the five different deformation levels: A) no deformation, B) BODY, C) PARTIAL, D) HEAD and E) FULL. See the video for examples of the deformations animated.

The animation of these clips was done with shape activations offset from one another in time. For example, shapes indicating great effort (such as the clenching of the mouth or squinting of eyes) were dialed in during "mid curl" where the effort expended is greatest. The video of the mocap session proved uninformative about actual deformation as the actors were wearing black mocap suits. Instead, references of weight lifters and videos taken by the artist helped inform the creation and animation of these shapes. Wrap3D [350] was used as a basis for many of the facial shapes.

The muscle deformations were animated by hand for one reference clip and then retargeted to all clips. The retargeting process took as input the start and end frame of each up or downswing in both the reference clip and the target clip. It then scaled the timing of the keys from the reference clip to the target clip based on these landmarks in the timeline.

5.1.5 Demographics

The number of participants, mean age (SD), and gender data are summarized in Table 5.4. Other data were similar across the VR participant pools. The ethnicity of participants was: 81.3% White/Caucasian, 6.7% Black/African/African American, 6% Asian/Asian American, 4.4% Latin/Hispanic, and 3.1% preferred not to disclose. The largest group had some experience with VR (48.2%), 35.2% had no experience, and the remainder had more extensive experience. A majority had some experience exercising

Table 5.3: Blendshapes in the Avatar Model

Body Type:	
Slim	Reduce overall muscle mass
BellyOut	Increase torso fat
Corrective shapes:	
Leg	Leg lifting corrective
Forearm	Corrective for arm bending at elbow
WristDown	Flexion of the wrist
WristUp	Extension of the wrist
LatIN	Prevents arm from penetrating the side of the torso
Tense Shapes:	
Torso Tense	Abdominal, obliques and pectoralis strain (abs and oblique deltas reduced dramatically for BellyOut variation)
NasilFold	Nasolabial fold (crease at the inside edge of cheek)
PlatFront	Platysmal sheet, the broad sheet of muscle fibers extending from the collarbone to the edge of the jaw
Plat	Platysmal sheet lateral (indicating extra strain/effort)
SternoMastoid	Large muscle from the corner of jaw/head to the start of collarbone
JawClench	Clench jaw
FaceClose	A “wince” comprised of closing of eyes, cheek raiser and tightening, raising of lips
LegFlex	All major muscles around knee
ArmFlex	Flex the bicep/tricep muscles and also added some forearm definition
EyeClose	Used in sync with face close to offset timing

Exp.	N	Age	Gender
1	30	37.7 (12.7)	F 50%, M 46.7%, O 3.3%
2	35	35.7 (9.7)	F 48.6%, M 48.6%, O 2.9%
3 & 5	35	33.9 (9.5)	F 45.7%, M 45.7%, O 8.6%
4	35	34.5 (8.3)	F 45.7%, M 51.4%, O 2.9%
6	194	39.3 (11.1)	F 36.1%, M 63.9%

Table 5.4: Participant demographics. Gender category O is a composite of Non-binary/third gender and Other.

or weightlifting (62.0%) or did it regularly (31.3%). Similarly, a majority had some experience seeing others lift weights (62.6%) with some seeing it regularly (24.2%).

5.1.6 Analysis

Analysis was generally performed using linear mixed effect models [351, 352]. These offer a more general approach than ANOVA as they include fixed and random effects, but similarly predict the dependence of a response term (e.g., perceived effort) on one or more factors (e.g., the size of the virtual character's body). The participant ID was treated as a random effect since the participant pool is merely a sample of the more general population. Type II Wald chisquare tests were used to evaluate significance within the models. Post-hoc analysis was conducted by calculating pairwise comparisons using estimated marginal means with Tukey correction. Exceptions to this analysis scheme will be noted in the relevant sections. The error bars in all plots show standard error.

5.2 Experiments 1-6

5.2.1 Exp. 1, Baseline

The first experiment was designed to provide a baseline measurement of how well people can perceive weight and effort from motion kinematics. For this reason, the lifted dumbbells were not shown. This also makes the work more directly comparable with previous work on point light displays. The displayed body was approximately matched to that of the lifter, as described in Section 5.1.4. 40 clips were used in this experiment (4 actors x 5 different weights x 2 repetitions). Weights were evenly spaced between 0 and each person's maximum lift, so they were not the same for each actor. Since previous research [21] has shown that people are more accurate when making estimates for a single actor at a time, clips were grouped by actor and randomized within actor. Participants rated their perception of both effort and weight for each clip, but the order of these questions was counterbalanced, so only half rated effort first.

Figure 5.4 shows perceived effort as a function of the actual effort made by the lifter, considering their maximum lift to be 100% effort. Participants' estimates of the weights for the various lifts by actor are shown in Figure 5.6. For every lifter, there is a highly

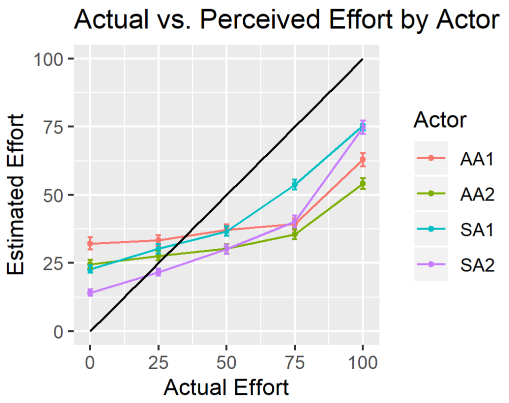


Figure 5.4: Perceived effort by lifter compared to actual effort for the Baseline experiment. The black line indicates perfect performance.

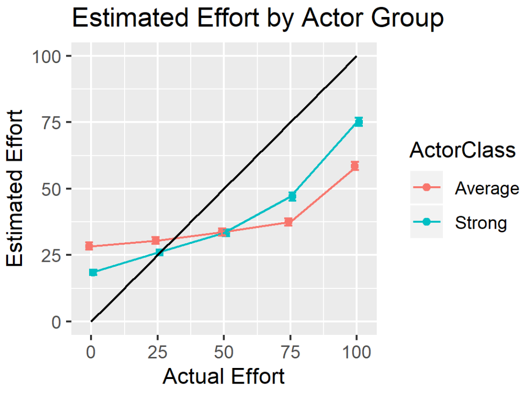


Figure 5.5: Perceived effort, grouped by lifter strength.

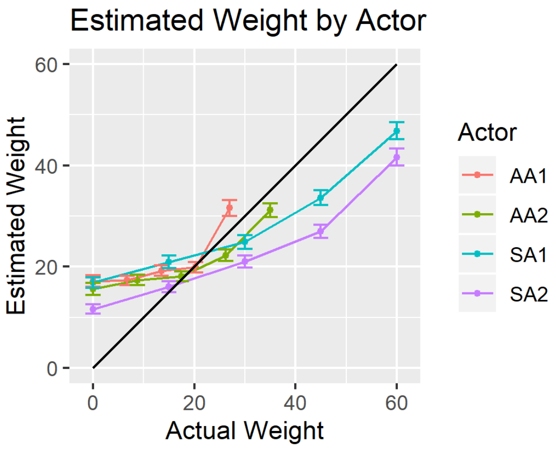


Figure 5.6: Perceived weight by actor in the Baseline experiment.

significant correlation between perceived effort/weight and actual effort/weight (Pearson's product-moment correlation with all p-values less than 1e-14). The correlations are strong for the stronger lifters and medium for the average lifters on both measures: (Effort Pearson's r: AA1=.41 , AA2=.48 , SA1=.73 , SA2=.73; Weight Pearson's r: AA1=.36 , AA2=.41 , SA1=.61 , SA2=.64). These findings suggest that at least to some degree, people are able to observe both weight and effort from lifters' kinematics.

It can also be observed that correlations are somewhat stronger for effort. Using Fisher's Z transform to compare correlations shows that these differences are not significant for the average strength lifters (AA1: Z=.72, p=.47; AA2: Z=1.06, p=.29), but are significant for the strong lifters (SA1: Z=2.68, p=.0074; SA2: Z=2.08, p=.038). This suggests that people may be better able to observe effort than weight for the strong lifters. Visual inspection of Figure 5.4 shows similar slopes within the Average Strength and Strong groups, but differences between them, a trend made more clear in Figure 5.5. Lifter group does significantly affect observations based on fitting a linear mixed effects model ($\chi^2(1) = 15.645$, $p < .001$). People were more accurate in their effort perceptions for the strong group.

Observation of Figures 5.4 to 5.6 shows a curvilinear relationship between the actual and perceived values such that the curve is flatter for lower effort/weight and more steep towards the maximum. Mirroring previous analyses (e.g., [21]), a linear mixed effects model was fit to each actor for each of Effort and Weight. Since the Weight levels differ across actors, we chose to fit an individual model to each actor, rather than treating Actor as an additional factor. In all cases, the independent variable (effort or weight) had a highly significant impact on the perceived estimates (p<.001, details in Appendix), reflecting that people are adjusting their judgments based on both the actual effort and actual weight. Pairwise comparisons were done for each model. These show that the 100% lifts differed significantly from the 75% lifts for both Effort and Weight for all actors, but lower levels were generally not significantly different for the average actors (details in Appendix). This reflects the more moderate slope in this part of the response curve and implies that observers may be less sensitive to these more subtle variations in kinematics. This suggests that the motion of an actor becomes more distinct as they approach their maximum lift. A possible explanation for this might be the involvement of additional kinematic cues, such as the movement of the trunk.

Relationship between Effort and Weight

In order to better understand the relationship between effort and weight estimates, we can normalize the estimated weight values (i.e. express them as a percentage of some max lift) so that effort and weight can be plotted on the same scale. Normalizing with the actual max lift of each performer produces the plots in Figure 5.7, which shows the curves have similar shapes, but are not aligned for the average actors. If instead, we optimize for a normalizing constant for each actor by minimizing the difference between the normalized weight and estimated effort, we produce the chart in Figure 5.8 with weights: AA1: 51.3lbs, AA2: 60.6lbs, SA1: 65.3lbs, SA2: 64.8lbs. There is clearly a strong correspondence between the curves for estimated effort and estimated weight (Pearson's correlation r = 0.74), so it may be that people were making a single judgment based on the motion and then scaling that to estimate the other quantity. We will revisit this in the discussion (Section 5.2.7) when there is more evidence that effort is the quantity estimated.

The weights are minimizing the error in $weight_{est} = effort_{est} * w/100$ where w is the normalizing constant, so provide an estimation of people estimate the max lift of each actor (100% effort). Alternatively, calculating the regression between estimated weight and estimated effort yields slightly different constants (AA1: 47.1, AA2: 55.6, SA1: 62.2, SA2: 58.7) because of the squared error term in regression, but the same pattern emerges. With either analysis, the max lift of AA1 is slightly lower and slightly higher for SA1 and SA2. It is worth noting that the range of these values (51.3 to 65.3 lbs or 47.1 to 62.2 lbs based on regression) is far less than the range of the actual max lifts of the actors (27 to 60 lbs). This may suggest that people do make strength predictions based on body size, but their accuracy in doing so is limited. Alternatively, it may be that the virtual characters used did not adequately capture the details necessary to make these estimates accurately.

5.2.2 Exp. 2, Body Shape

The goal of this study was to understand how the size of the virtual character's body impacted perceptions of weight and effort. This is relevant for situations where someone's avatar may not match their body proportions. To provide a more realistic use case for

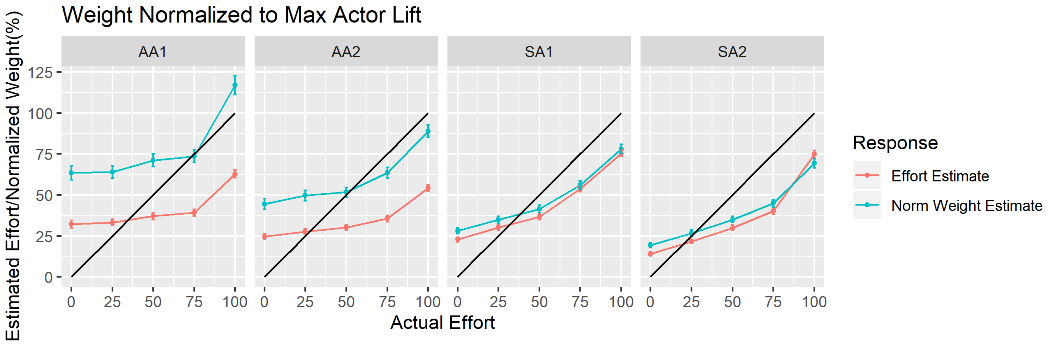


Figure 5.7: Perceived effort and weight for weights that are normalized based on the actual max lift for each lifter.

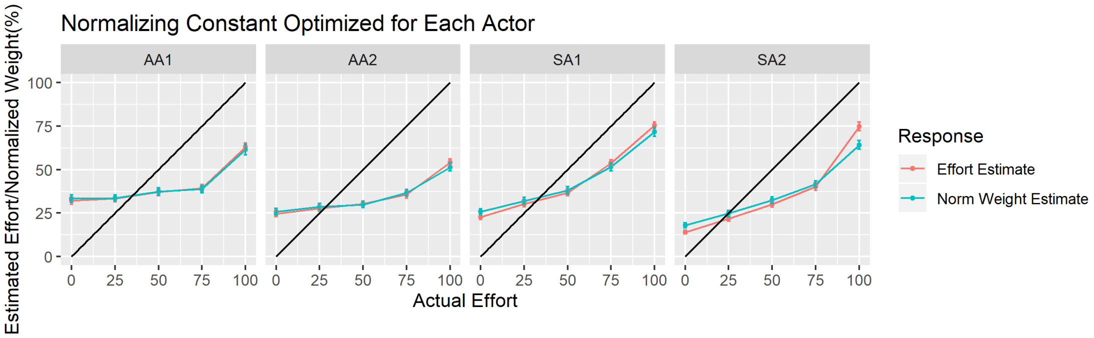


Figure 5.8: Perceived effort and normalized weight where the normalizing constant is optimized for each actor (AA1: 51.3lbs, AA2: 60.6lbs, SA1: 65.3lbs, SA2: 64.8lbs).

VR applications, the dumbbells were visualized in the virtual character animations. In all cases, the size of the dumbbell was matched to the lift motion used (e.g., if the lifter had lifted 30 lbs, a 30 lb dumbbell was displayed). Each motion was displayed on the virtual character models of each of the four actors (Body factor). Thus the motion and visualized dumbbell provided consistent signals, but the body shape was inconsistent (matched to the original lifter in some clips, and not in others). To maintain the balance between the matched and unmatched groups, one matched animation was included for each mismatched, so there were six clips for each actor-weight combination (three on the matched actor characters and one on each of the unmatched actor characters). In total there were 96 clips (6 body-motion combinations x 4 actors x 4 weights). Note that the zero-weight lifts were not used. The presentation was fully randomized. After each clip, participants were asked to rate weight and effort as before and also asked to rate the naturalness of the clip on a 7-point Likert scale to explore if mismatched clips were less believable. Since we are combining different actor body shapes and actor motions in this section, we add 'm' after an actor ID to specify motion and 'b' body shape, e.g., AA1m and AA1b.

A first question is: does body type impact the perception of effort? Effort ratings for the motions of each actor displayed on each virtual character model are shown in Figure 5.9. It is clear that effort ratings are largely consistent across these variations in body shape. Linear mixed effect models fit to each actor motion show that body shape did not have a significant impact on Effort ratings for AA1m, AA2m, or SA2m (AA1m: Effort $\chi^2(3) = 533.2$, $p < .0001$, Body $\chi^2(3) = 3.97$, $p = .26$, Effort:Body$\chi^2(9) = 14.55$, $p = .10$; AA2m: Effort $\chi^2(3) = 253.46$, $p < .0001$ Body $\chi^2(3) = 1.08$, $p = .78$, Effort:Body$\chi^2(9) = 12.64$, $p = .18$; SA2m: Effort $\chi^2(3) = 723.9$, $p < .0001$ Body $\chi^2(3) = 6.26$, $p = .10$ Effort:Body$\chi^2(9) = 13.27$, $p = .15$) . However, there was a significant impact of Body shape on Effort ratings for the motion of SA1m (SA1m: Effort $\chi^2(3) = 915.8$, $p < .0001$ Body $\chi^2(3) = 63.72$, $p < .0001$ Effort:Body$\chi^2(9) = 26.10$, $p = .0020$) . Post-hoc analysis shows that the only significant differences related to the AA1 body. Perceived effort on the AA1 character was significantly less than SA1 and SA2 bodies at 50 (p<.0001, p=.0005), 75 (p<.0001, p<.0001) and 100% (p=.0001, p=.0003) actual effort. It was also significantly less than AA2 at 25% (p=.005) and almost at 50% (p=.0504). It is not clear what is causing this difference.

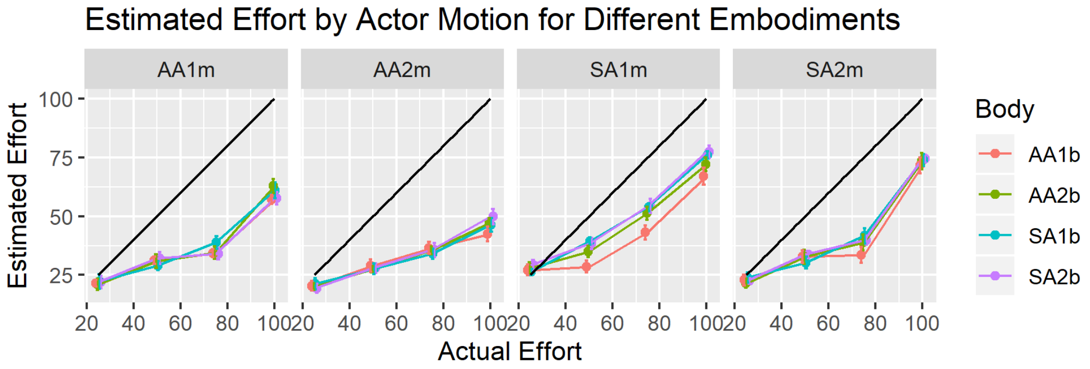


Figure 5.9: Perceived effort as body type is changed. Facets show motion from different actors, colors code different character bodies.

A next question is whether changing body shape impacts the perception of the lifted weight. Here the answer is yes. The impact of body shape on weight is shown in Figure 5.10. Linear mixed effect models fit to the data for each actor in all cases show significant main impacts of Body and of actual Weight on perceived weight, but no interaction (AA1: Weight $\chi^2(3) = 813.9$, $p < .0001$, Body $\chi^2(3) = 22.34$, $p < .0001$, Weight:Body$\chi^2(9) = 14.34$, $p = .11$; AA2: Weight $\chi^2(3) = 711.6$, $p < .0001$, Body $\chi^2(3) = 46.87$, $p < .0001$, Weight:Body$\chi^2(9) = 11.61$, $p = .24$; SA1: Weight $\chi^2(3) = 730.1$, $p < .0001$, Body $\chi^2(3) = 62.35$, $p < .0001$, Weight:Body$\chi^2(9) = 11.72$, $p = .23$; SA2: Weight $\chi^2(3) = 696.7$, $p < .0001$, Body $\chi^2(3) = 42.8$, $p < .0001$, Weight:Body$\chi^2(9) = 6.87$, $p = .65$) . Post-hoc analysis shows that the weight estimates for the average size virtual characters were always significantly lower than the weight estimates for the strong virtual characters (larger bodies) except the virtual characters of AA1 vs. SA1 fell slightly below significance for the motion of AA2 (t=-2.411, p=0.076). This suggests that given the same perception of effort, people assume the larger virtual characters are lifting more weight.

Given that body shape does impact the perception of weight, the next question is how much? Since the effect was significant at the class level, Average vs. Strong, differences were calculated by comparing the means of the ratings for these classes. In 15 of the 16 cases, the estimates were heavier for the Strong class. The one outlier is the lightest

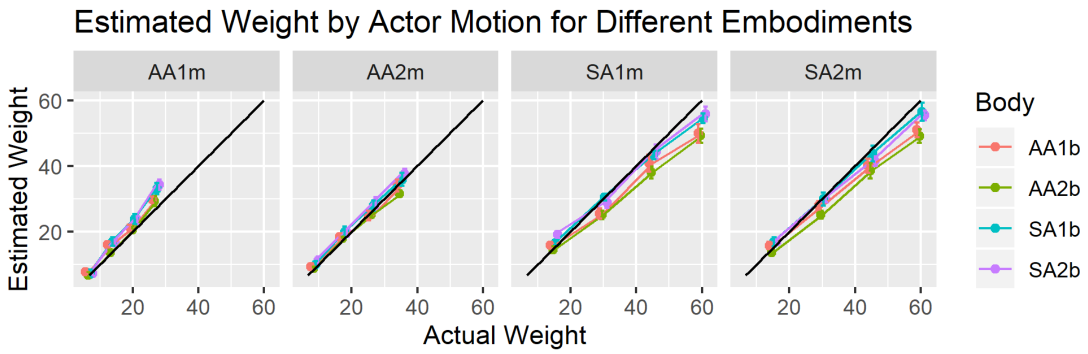


Figure 5.10: Perceived dumbbell weight as body type is changed. Facets show motion from different actors, colors code different virtual character bodies.

lift performed by AA1 (lift was 6.75lbs, mean Average estimate 7.56, Strong estimate 7.36lbs). For the remaining classes, the mean Strong estimate was between 1.5 and 5.7 pounds heavier and generally increased for larger weights. As a percentage of the actual lift, the Strong estimates averaged 11% higher.

Finally, we explore whether changing the body size impacts the naturalness of the motion. To analyze this, a linear mixed effects model was fit to the data with the 7-point Likert scale Naturalness ratings as response variable and factors Effort, actor Motion (performer of the lifts) and Body (character model used for display). There were significant main effects for Body shape and actor Motion and all two-way interactions were significant, but the three-way was not. The two interactions related to Body shape, Effort:Body shape and actor Motion:Body shape are shown in Figures 5.11 and 5.12, respectively. The significant differences from post-hoc analysis are marked. Most drops in Naturalness are related to the AA1 model, with some related to AA2. When AA1b is used on motion from larger actors (which would also feature larger dumbbells) and at higher efforts (heavier lifts), it looks less natural. Motion from larger actors also looks less natural on AA2, although the difference is less pronounced.

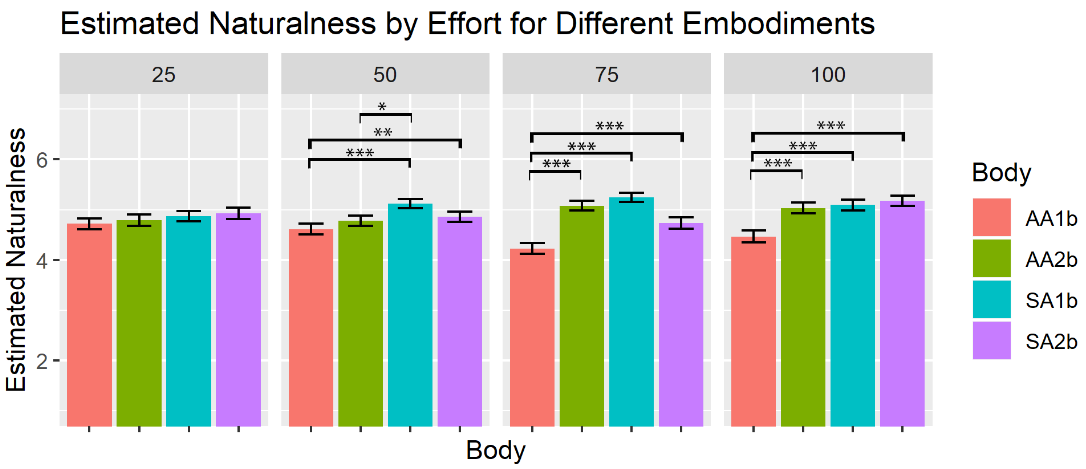


Figure 5.11: Naturalness for different body shapes. Facets show different effort lifts.

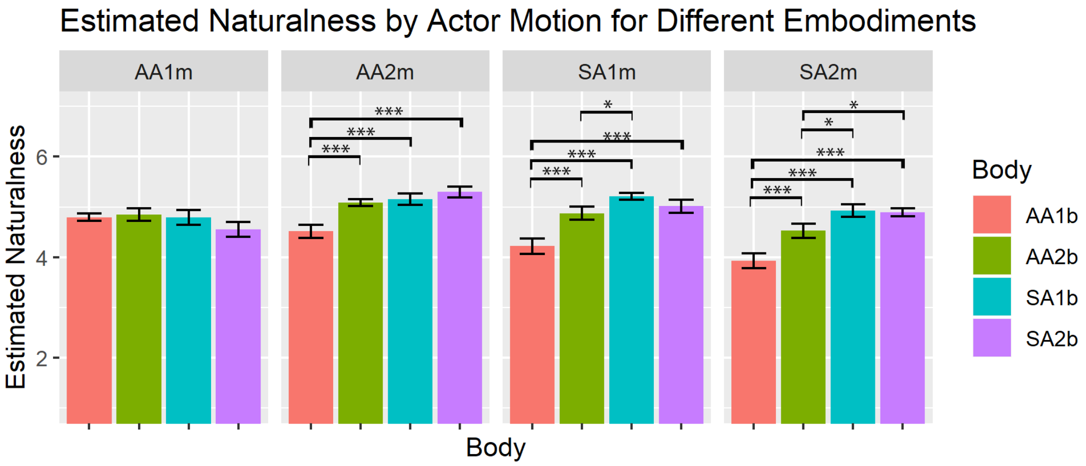


Figure 5.12: Naturalness for different body shapes. Facets show motion from different actors.

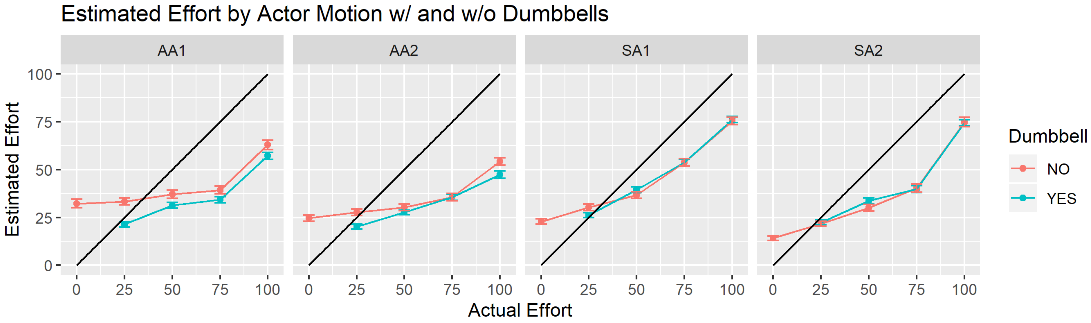


Figure 5.13: Perceived effort with and without displayed dumbbells. Facets show motion from different actors.

Impact of Showing Dumbbells

Between Exps. 1 and 2, we have data for the same motions with and without the display of dumbbells. This allows us to consider whether the visual appearance of dumbbells influences the perception of effort. Plots of effort with and without visualized dumbbells for the same motion and body model from Exp. 1 and Exp. 2 are shown in Figure 5.13. Results are mixed. There is no significant difference for SA1 and SA2 (SA1: Effort $\chi^2(3) = 1243.7$, $p < .0001$, Dumbbell $\chi^2(3) = .0020$, $p = .96$, Effort:Dumbbell$\chi^2(9) = 5.68$, $p = .13$; SA2: Effort $\chi^2(3) = 1052.9$, $p < .0001$, Dumbbell $\chi^2(3) = .083$, $p = .77$, Effort:Dumbbell$\chi^2(9) = 1.79$, $p = .62$) . For AA1, there is a significant main effect for Dumbbell, but no significant interaction between Dumbbell and Actual Effort (AA1: Effort $\chi^2(3) = 622.7$, $p < .0001$, Dumbbell $\chi^2(3) = 3.92$, $p = .047$, Effort:Dumbbell$\chi^2(9) = 7.63$, $p = .054$) . Effort estimates with dumbbells present are lower. For AA2, there is a significant interaction between Dumbbell and Actual Effort (AA2: Effort $\chi^2(3) = 421.2$, $p < .0001$, Dumbbell $\chi^2(3) = 1.92$, $p = .17$, Effort:Dumbbell$\chi^2(9) = 9.5$, $p = .023$) , with post-hoc analysis showing the clips with dumbbells rate significantly lower (t=2.10, p=0.038) at 25% effort and no significant differences at other effort levels (100% effort is tendential (t=1.906, p=0.059)).

For interpreting these results, we can consider evidence from Exp. 1 that it seems participants estimated the amount the average lifters could lift as being higher than it is, in line with strong lifters. Perhaps when they saw the dumbbells, participants adjusted down their estimates of effort for average lifters.

Next, we consider whether the visualization of dumbbells impacted the perception of weight. Data from Exps. 1 and 2 for the same lifts, with and without dumbbells, is shown in Figure 5.14. The patterns are clearly different, with much improved estimates when dumbbells are present. This difference was confirmed by again fitting a linear mixed effect model to the motion of each actor. In all cases, there is a significant interaction between the lifted weight and the visual presence of dumbbells (AA1: Weight $\chi^2(3) = 613.6$, $p < .0001$, Dumbbell $\chi^2(3) = 3.04$, $p = .081$, Weight:Dumbbell$\chi^2(9) = 52.62$, $p < .0001$; AA2: Weight $\chi^2(3) = 659.7$, $p < .0001$, Dumbbell $\chi^2(3) = .514$, $p = .47$, Weight:Dumbbell$\chi^2(9) = 63.88$, $p < .0001$; SA1: Weight $\chi^2(3) = 976.2$, $p < .0001$, Dumbbell $\chi^2(3) = 3.61$, $p = .057$, Weight:Dumbbell$\chi^2(9) = 48.60$, $p < .0001$; SA2: Weight $\chi^2(3) = 870.6$, $p < .0001$, Dumbbell $\chi^2(3) = 17.74$, $p < .0001$, Weight:Dumbbell$\chi^2(9) = 44.08$, $p < .0001$). The weights where the presence of the dumbbell lead to significant changes in ratings are the ones that appear visually different in Figure 5.14, 6.75 lbs for AA1; 8.75 lbs for AA2; 30, 45 and 60lbs for both SA1 and SA2. There does not appear to be an obvious pattern to these differences, other than that the presence of the weights largely corrected judgment errors made when they were not present. The correlations between real and perceived weight substantially improved when dumbbells were shown, and in all cases the improvement is statistically significant using Fisher's Z transform to compare correlations: AA1: r= .69 vs. .36, z = 6.2, p <.00001; AA2: r=.73 vs. .41, z = 6.49, p<.00001; AA2: r=.77 vs. .61, z = 4.1, p<.00001; SA2: r=.76 vs. .64, z=3.14, p<.00001). This suggests that the visual appearance of the dumbbells had a major impact on participants' ability to accurately estimate weight.

5.2.3 Exp. 3, Dumbbell Size

The goal of this experiment was to understand how changing the size of the dumbbell impacted perceptions of weight and effort. This corresponds to cases where a person's avatar is shown moving objects of different mass to what they are moving when they drive the avatar. In all clips, the virtual character's body was matched to the actor that provided the source motion. The clips were displayed with all possible weights for that actor, so here body and motion were consistent, but dumbbell size was varied. In total,

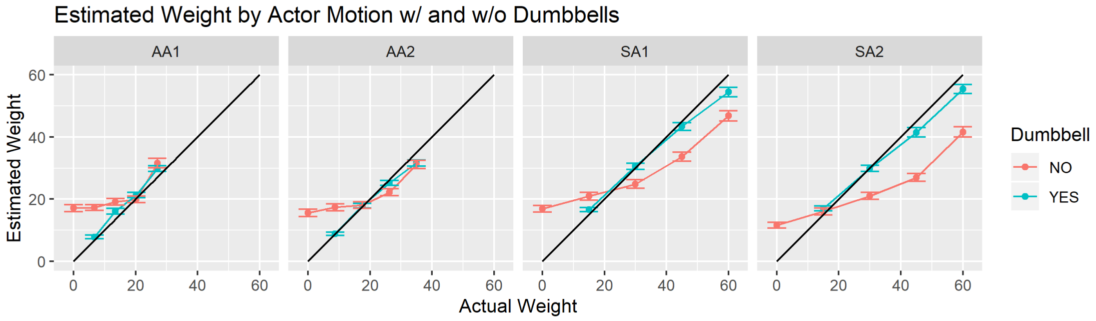


Figure 5.14: Perceived lifted weight with and without displayed dumbbells. Facets show motion from different actors.

80 clips were prepared (5 motions x 4 dumbbell sizes x 4 actors). The zero lift motion was used, but only dumbbells with a mass greater than zero were displayed. As with Exp. 1, actors were grouped and clips were randomized within each actor. Participants rated weight, effort, and naturalness.

Figure 5.15 shows the relation between actual effort and perceived effort for each actor when dumbbell size is changed across clips. Again, linear mixed effect models were fit to the data for each actor. Statistical analysis indicates that in all cases, there is a significant main effect for dumbbell size (size indicates a particular weight) and for AA2 there is also a significant interaction between dumbbell weight and effort (AA1: Effort $\chi^2(4) = 303.5$, $p < .0001$, Dumbbell $\chi^2(3) = 13.75$, $p = .0032$, Effort:Dumbbell $\chi^2(12) = 15.19$, $p = .23$; AA2: Effort $\chi^2(4) = 176.2$, $p < .0001$, Dumbbell $\chi^2(3) = 39.67$, $p < .0001$, Effort:Dumbbell $\chi^2(12) = 35.93$, $p = .0033$; SA1: Effort $\chi^2(4) = 649.9$, $p < .0001$, Dumbbell $\chi^2(3) = 12.78$, $p = .0052$, Effort:Dumbbell $\chi^2(12) = 11.48$, $p = .49$; SA2: Effort $\chi^2(4) = 603.3$, $p < .0001$, Dumbbell $\chi^2(3) = 10.66$, $p = .014$, Effort:Dumbbell $\chi^2(12) = 10.98$, $p = .53$). The majority of the variation is for Effort not dumbbell size, however, and post-hoc analysis shows that the impact of dumbbell weight is almost exclusively limited to the lightest dumbbell for each actor being perceived as less effort to lift than some of the heavier three. For AA1, 6.75lbs was perceived as less effort than 20.25lbs (p=0.0038) and 27lbs (p=.0280). For AA2, the interaction was also significant: 8.75 lbs was perceived as less effort than 26.25lbs at 50% Effort (p=.0003) and (25% Effort (p=.0002); 8.75lbs was perceived as less effort than 35lbs at 75% Effort (p=.0002), 50% Effort (p= 0.027) and 25% Effort (p=.0021);

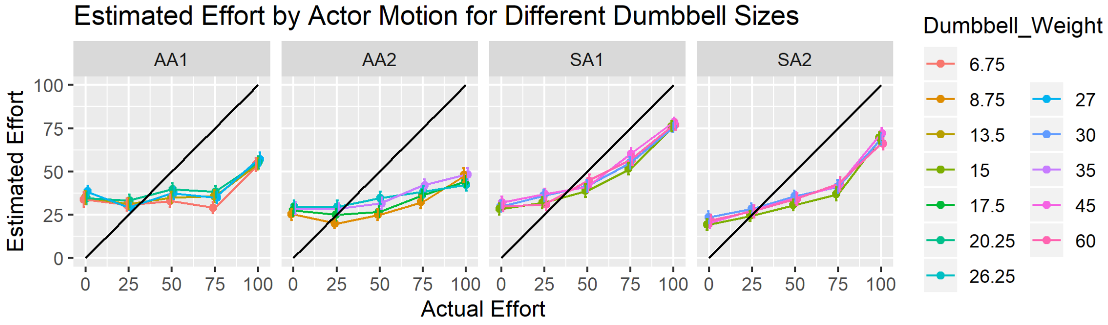


Figure 5.15: Perceived effort as a function of actual effort for different dumbbell sizes. Facets show motion from different actors.

and in the one case involving a dumbbell other than the lightest, 17.5lbs was seen as less effort than 26.25lbs at 50% Effort (p= .0040). For SA1, 15lbs was perceived as less effort than 45 (p= .0027) and for SA2, (15lbs was perceived as less effort than 30 (p=.018) and 45 (p=.046). This rather limited impact is consistent with the visually quite consistent ratings across dumbbell size shown in the figure.

The impact of dumbbell size on the perception of weight is shown in Figure 5.16. Statistical analysis confirms what is visually clear: the visual size of the dumbbell has a very strong impact on the perceived weight. Again, linear mixed effect models were fit to each actor. There was a significant main effect for both actual Weight and visualized Dumbbell Weight for all actors, and no significant interactions (AA1: Weight $\chi^2(3) = 38.3$, $p < .0001$, Dumbbell $\chi^2(3) = 427.1$, $p < .0001$, Weight:Dumbbell$\chi^2(9) = 6.24$, $p = .90$; AA2: Weight $\chi^2(3) = 25.23$, $p < .0001$, Dumbbell $\chi^2(3) = 628.5$, $p < .0001$, Weight:Dumbbell$\chi^2(9) = 15.6$, $p = .211$; SA1: Weight $\chi^2(3) = 65.30$, $p < .0001$, Dumbbell $\chi^2(3) = 794.3$, $p < .0001$, Weight:Dumbbell$\chi^2(9) = 14.36$, $p = .28$; SA2: Weight $\chi^2(3) = 59.66$, $p < .0001$, Dumbbell $\chi^2(3) = 592.7$, $p < .0001$, Weight:Dumbbell$\chi^2(9) = 10.92$, $p = .53$) . Unlike for effort, the majority of the variance was due to visualized Dumbbell Weight, not the actual lift performed. Post-hoc analysis showed that every dumbbell size led to significantly different perceived weight than every other dumbbell size for all actors (all p<.0001). The impact of motion kinematics is more modest. For all actors, the heaviest lift was still seen as heavier than all other, with the exception of the second heaviest lift for AA2. The additional significant differences were: AA2, {0 < 26.25lb lift}; SA2, {15 < 30, 45lb lifts}. Plot-

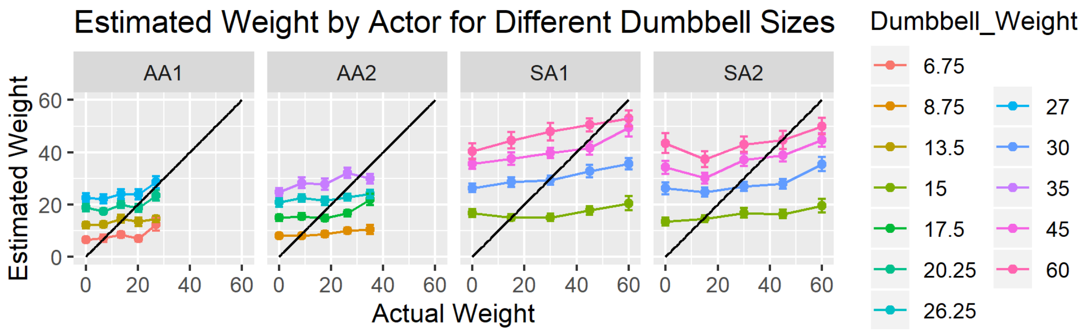


Figure 5.16: Perceived weight as a function of actual weight for different dumbbell sizes. Facets show motion from different actors.

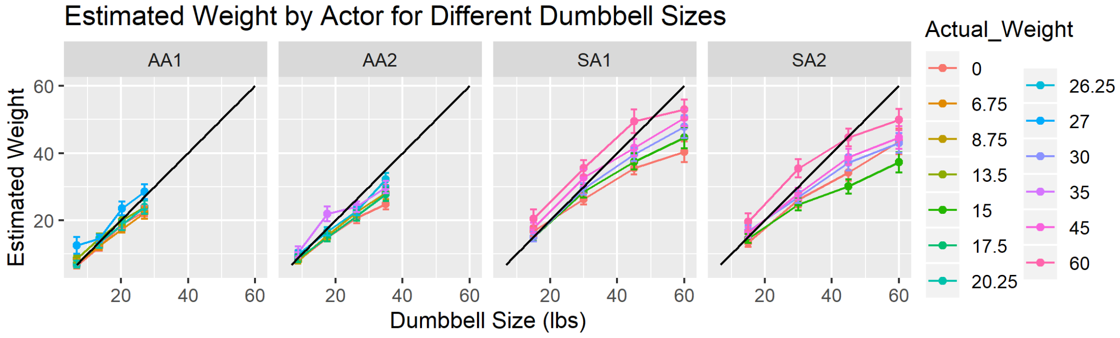


Figure 5.17: Perceived weight as a function of dumbbell size for different actual weights. Facets show motion from different actors, colors indicate different actual weights.

ting perceived weight as a function of dumbbell size (Figure 5.17) illustrates that much of the weight estimate is based on the dumbbell size, but some clear variation comes from the kinematics of the motion. It can be concluded that dumbbell size has a major impact on perceived weight, but only a minor impact on perceived effort.

Naturalness ratings were used to evaluate if people were sensitive to mismatches between the lift motion and the displayed dumbbell. A single linear mixed effect model was fit to the full data set with 7-point Naturalness Likert ratings as the response variable and factors lifter Effort, Dumbbell shape (as a percent of largest dumbbell used by lifter), and Motion Actor, along with all interactions. There were significant main effects for all three factors and significant interactions for Effort:Dumbbell Shape and Effort:Actor. Since we are most interested in the impact of dumbbell shape, we performed post-hoc analysis on the Effort:Dumbbell shape interaction, which is shown in

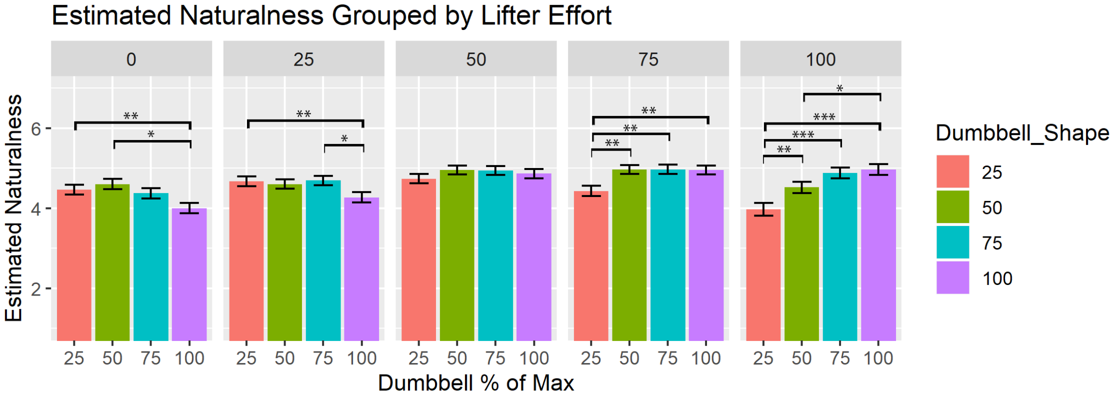


Figure 5.18: Naturalness ratings as a function of different dumbbell sizes. Facets show different effort levels.

Figure 5.18 with significant differences marked. It can be seen that Naturalness ratings decrease for the more extreme combinations. For 0 and 25% effort lifts, the largest dumbbell was perceived as less natural. This is the most likely problem case for practical VR, as people may be moving just their hands or a light controller, but be shown lifting a heavy object. There is a slightly more pronounced drop in Naturalness ratings at the other end of the spectrum, where the smallest dumbbell size was seen as unnatural at both 75 and 100% lifts. At the 100% lift, the 50% dumbbell was also seen as less natural.

5.2.4 Exp. 4, Strain Deformations

The goal of this experiment was to understand the impact of adding muscle deformations on the perception of weight and effort. In all cases, the motion clip was matched to the body model of the original performer. The dumbbells were not shown to allow a direct investigation of the relative impact of motion kinematics and visualized muscle strain. Each lift was shown with one of five deformation strain levels: NONE (the base clips used before), FULL (flexion of the body, face and neck in correspondence with the lifted motion), FACE (the face and neck flexion from FULL), BODY (the body and arm flexion from FULL) and PARTIAL (a reduced magnitude version of FULL), as shown in Figure 5.3 and the accompanying video. See Section 5.1.4 for details of how deformations were modeled. As this was a preliminary investigation into the impact of muscle strain, no attempt was made to tune the strain to the particular weight lifted. In total, there were 100 clips (5 strain levels x 5 motions x 4 actors). Clips were randomized within each actor group.

Manipulation Check

In order to confirm that the strain animations read as desired, we performed a manipulation check as an online experiment on Amazon Mechanical Turk using Mephisto library[1]. We have set qualifications such that only people who have already completed over a 1000 tasks with above 95% approval rating could participate in the experiment. The duration of the task was 30 minutes and the paid amount was $7.5. Fifty participants viewed a sequence of clips and after each clip rated the prompt: "How much strain do you think the person in the video is exhibiting? (0 - no strain, 100 - maximum strain)". The videos contained 5 strain levels x 4 actors x 2 samples for a total of 40 clips. All strain animations were done on a character in a static A-pose to avoid any impact from motion. A linear mixed effect model showed a significant main effect for Deformation (Actor $\chi^2(3) = 6.51$, $p = .089$, Deformation $\chi^2(4) = 1724.4$, $p < .0001$, Actor:Deformation $\chi^2(12) = 6.29$, $p = .90$). The data averaged across actors is plotted in Figure 5.19 and the pattern was similar for each actor. Post-hoc analysis shows that the difference between every strain deformation was highly significant (p<.001). This confirms that the strain deformations are read as intended.

Figure 5.20 shows the impact of deformation conditions on perceived effort. Linear mixed effect models were fit to the data for each actor. The general ordering in terms of increasing perceived Effort is: NONE, BODY, PARTIAL, HEAD, and FULL. The differences largely relate to which of these are statistically separated. For AA1 and SA1, there is a significant main effect of Deformation and no interaction (AA1: Effort $\chi^2(4) = 126.2$, $p < .0001$, Deformation $\chi^2(4) = 549.0$, $p < .0001$, Effort:Deformation $\chi^2(16) = 17.1$, $p = .38$; SA1: Effort $\chi^2(4) = 454.3$, $p < .0001$, Deformation $\chi^2(4) = 482.7$, $p < .0001$, Effort:Deformation $\chi^2(16) = 24.77$, $p = .074$). In both cases, HEAD and FULL are not significantly different, but each of the other levels is. For AA2 and SA2, there is a significant interaction between Effort and Deformation (AA2: Effort $\chi^2(4) = 136.5$, $p < .0001$, Deformation $\chi^2(4) = 809.1$, $p < .0001$, Effort:Deformation $\chi^2(16) = 26.48$, $p = .048$; SA2: Effort $\chi^2(4) = 328.9$, $p < .0001$, Deformation

[1] https://github.com/facebookresearch/Mephisto

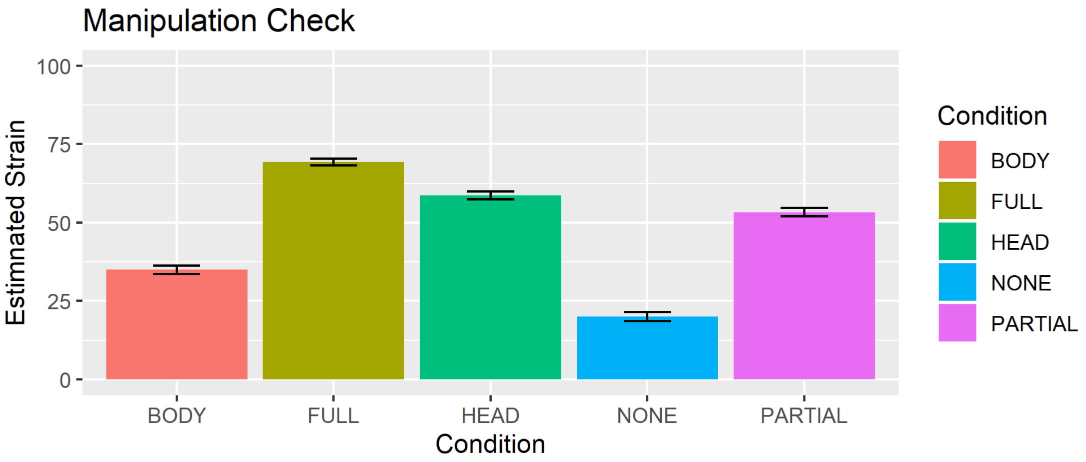


Figure 5.19: Manipulation check ratings for the level of strain conveyed by each deformation condition.

$\chi^2(4) = 385.7$, $p < .0001$, Effort:Deformation $\chi^2(16) = 30.10$, $p = .017$). For AA2, the differences that are not significant are: BODY - NONE and FULL - HEAD at all effort levels, HEAD - PARTIAL 0, 50, and 100% effort and FULL - PARTIAL 50 and 75% effort. For SA2, HEAD - PARTIAL are not significantly different at 0% effort and BODY - NONE, FULL - HEAD, FULL - PARTIAL and HEAD - PARTIAL are not significant at 50 or 75% effort. At 100% effort, there are two separated groups: FULL, HEAD, PARTIAL and NONE, BODY. Overall, the deformations have a clear impact on effort, especially those involving the head and neck (FULL, HEAD, PARTIAL). The impact of BODY on its own is more limited.

Figure 5.21 shows the impact of muscle deformations on the perception of weight. A linear mixed effects model was fit to the data for each actor. In each case, there was a main effect of Deformation, but not an interaction between Deformation and Weight (AA1: Weight $\chi^2(4) = 174.0$, $p < .0001$, Deformation $\chi^2(4) = 250.9$, $p < .0001$, Weight:Deformation $\chi^2(16) = 15.$, $p = .51$; AA2: Weight $\chi^2(4) = 128.7$, $p < .0001$, Deformation $\chi^2(4) = 347.7$, $p < .0001$, Weight:Deformation $\chi^2(16) = 25.51$, $p =$ $.061$; SA1: Weight $\chi^2(4) = 499.2$, $p < .0001$, Deformation $\chi^2(4) = 205.6$, $p < .0001$, Weight:Deformation $\chi^2(16) = 25.1$, $p = .068$; SA2: Weight $\chi^2(4) = 370.6$, $p < .0001$, Deformation $\chi^2(4) = 101.0$, $p < .0001$, Weight:Deformation $\chi^2(16) = 21.74$, $p = .15$) . The overall pattern is the same as with Effort, with the levels ordered NONE, BODY,

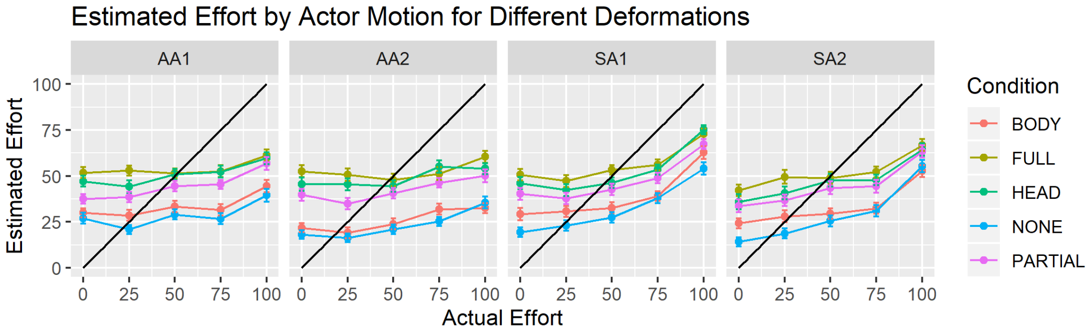


Figure 5.20: Perceived effort as a function of actual effort for different deformations. Facets show motion from different actors.

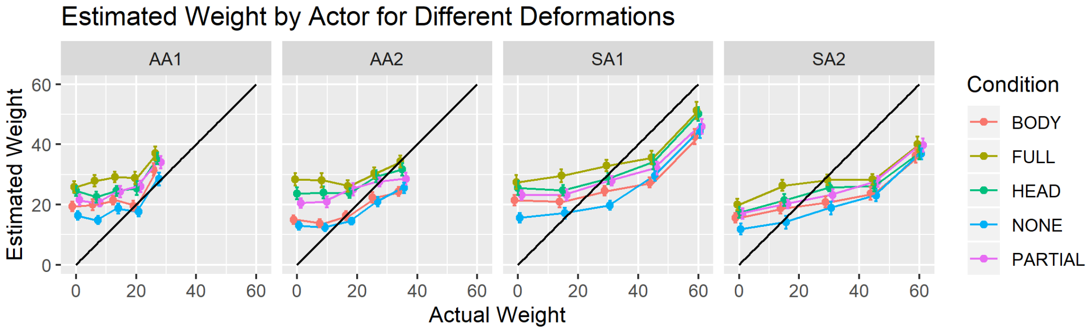


Figure 5.21: Perceived weight as a function of actual weight for different deformations. Facets show motion from different actors.

PARTIAL, HEAD and FULL and variation on whether they are statistically separated. For AA2, SA1 and SA2, there are three groupings: NONE, BODY, PARTIAL, HEAD and FULL. For AA1, all levels are significantly different except PARTIAL and HEAD. As with Effort, the deformations have a clear impact on the perception of lifted Weight.

Figures 5.22 and 5.23 quantify the impact of the strain deformations on effort and weight, respectively, by plotting the mean difference from the examples with no deformation. The general trends across deformation conditions reflect those of the previous analysis. Two additional points can be noted. First, the impact of the deformations can be substantial: up to a 30% increase of perceived effort for the FULL deformations and up to a 10-15lb increase in estimated weight. Second, the impact of deformations is

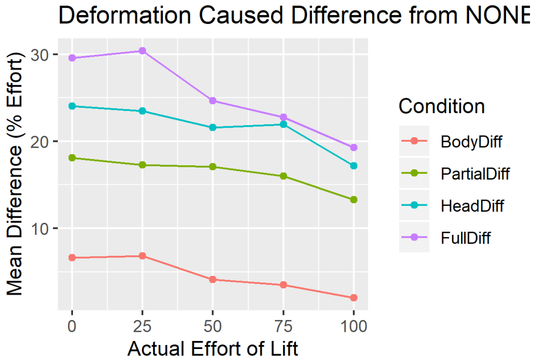


Figure 5.22: Perceived effort as a function of actual effort with various levels of muscle deformation.

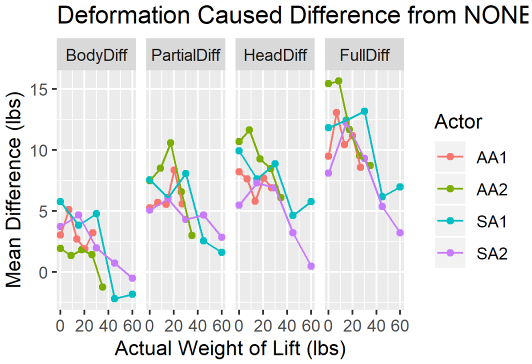


Figure 5.23: Perceived weight as a function of actual weight with various levels of muscle deformation.

largest on the lightest lifts and reduces as the lifts become heavier. This suggests that the deformation and kinematic signals are acting in concert and when there is limited evidence of effort on the kinematic channel, the deformation channel can have more impact.

To analyze any impact on naturalness, we fit a linear mixed effects model to the full set of data with Naturalness ratings as the response variable and factors lift Effort and Deformation, along with their interaction. We also fit a model with the additional factor Actor, but this did not improve the fit, so we analyze the first model. It showed significant main effects for Effort and Deformation, and a significant interaction (Effort $\chi^2(4) = 19.64$, $p < .0001$, Deformation $\chi^2(4) = 26.05$, $p < .0001$, Effort:Deformation $\chi^2(16) = 63.8$, $p < .0001$). The data for the interaction is plotted in Figure 5.24, with significant differences marked based on a post-hoc, pairwise comparison. It can be seen that Naturalness ratings drop at either end when the strain deformations do not match the motion. For zero effort lifts, the strain on the head and neck was seen as significantly less natural than no deformations or body-only deformations, which is consistent with the strain being a mismatch with the lift. Interestingly, it was also seen as less natural than FULL, so the combined body and facial strain cues were still plausible. At the maximum, 100% Effort, the no deformation condition was seen as less natural than PARTIAL, HEAD and FULL, and BODY only deformation was seen as less natural than HEAD and FULL.

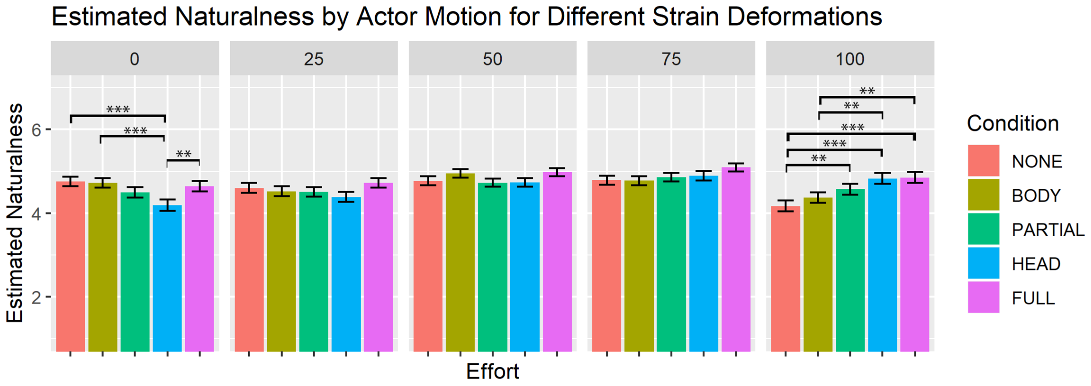


Figure 5.24: Naturalness ratings with changes in deformation. Facets correspond to different naturalness levels.

5.2.5 Exp. 5, Discrimination

This experiment had twin goals. The first was to understand when people could detect a zero-weight lift as being a fake lift and the second was to investigate if adding muscle deformations made it more difficult to detect zero-weight lifts. Zero-effort lifts were chosen as the comparison point because they correspond to the motion a person would perform if they were interacting in VR using hand tracking. Unlike the previous experiments, this experiment was run as an Interval Forced-Choice experiment in which participants were shown two clips in sequence and had to decide which clip was a correct visualization of the lift. In all cases, one lift was a zero-weight lift and the other lift was one of the weights greater than zero. They were told the weight of the target lift and a dumbbell of that size was used in both clips. The order was randomized. Two types of pairs were run. One had no muscle deformation on either clip. The second had FULL deformation on the zero-weight clip and no deformation on the actual lift.

The teal bars in Figure 5.25 shows the proportion of time people can detect the real lift compared to a zero lift, when both show the same dumbbell size. Chance level is 50%, so discrimination experiments often use 75% as a threshold for reliable detection [22]. While there are too few weight intervals to reasonably fit a psychometric function to the data, the overall patterns are clear. Light lifts of 25% effort are detected roughly at chance level, meaning people had difficulty distinguishing between these and zero lifts. For greater effort lifts, the 75% discrimination threshold is exceeded in all cases, but the

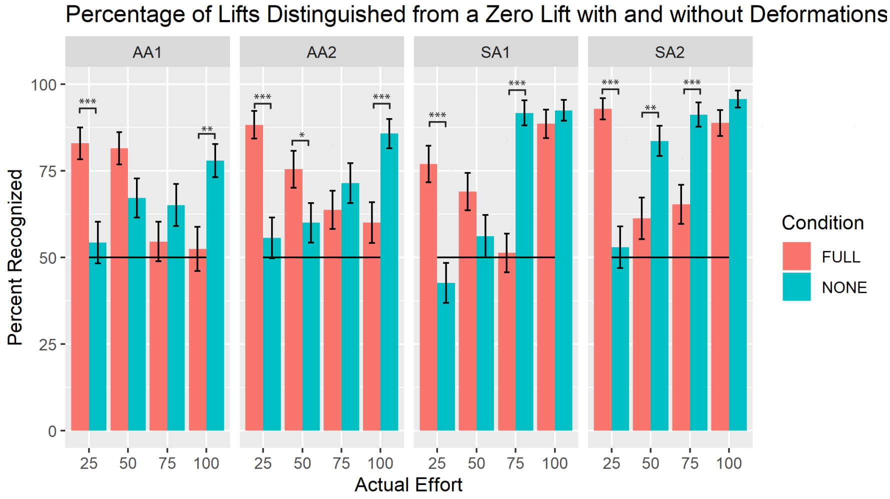


Figure 5.25: Detection rates for zero lifts with and without deformations, compared to actual weight lifts.

point at which this happens varies based on the lifter. For the average strength lifters, it occurred for the 100% lifts (27 and 35 lbs for AA1 and AA2, respectively). For the strong lifters, it occurred at lower effort, 75% and above for SA1 (45 lbs and up) and 50% effort for SA2 (30 lbs and up). Kinematic aspects of the lift such as weight shift will depend both on the amount lifted and the size of the actor. It may be that heavier lifts created more signal, even at lower effort, so were easier to distinguish.

The orange bars in Figure 5.25 show the proportion of people able to identify the correct lift when the opposing lift is a zero effort lift with the FULL strain deformation applied, and teal bars had no muscle deformations. Given that the response data, correct detection, was binary, we fit a generalized linear mixed effect model with binomial distribution and logit link function. The factors were Deformation condition (FULL or NONE), Effort level and Actor and all interactions were included in the model. Deformation was not significant, but all other main factors and interactions were (see Appendix for statistical details). Given that our main interest is the impact of Deformation, we report post-hoc analysis of the Deformation:Effort:Actor interaction in Figure 5.25. Detection rates near 50% (chance) indicate that people were not able to reliably detect the zero lift from the actual weight lift. Interestingly, in all cases for 25% effort lifts, adding muscle deformations to the zero-lift made it easier to detect the correct lift. This is likely because the FULL deformation simply showed too much strain for the displayed dumbbell. For each actor, there was at least one case where the added deformations made it significantly more difficult to detect the real lift compared to the zero lift. In these cases, the addition of muscle deformation could be a useful technique for obscuring the fact that users were unencumbered when their virtual characters were lifting objects. The general trend is for the detection of the "zero lift with strain" to be easy at light lifts and become more difficult at heavy lifts. This is reasonable as the strain level used in these clips was quite intense, so only appropriate for heavier lifts. It speaks to the need to tune the strain level to the desired exertion of the character. For the strong actors, the zero animation with deformations was less detectable at 100% effort than the zero animation with no deformations, but this difference was no longer significant. It may be that at these extreme lifts, the kinematic signal was so strong, muscle deformations alone did not provide enough counter information to override it.

5.2.6 Exp. 6, Comparison with 2D

We replicated Exp. 1 using Amazon Mechanical Turk as online surveys offer a potential mechanism to easily scale the research to larger participant pools. In past work, we had found an excellent match between the results of online studies and in-person VR studies, even though the online studies were only in 2D. However, in this case, results clearly differed.

Amazon Mechanical Turk experiment was set up the same way as described for Exp. 5 (Section 5.2.4). The duration of the experiment was 75 minutes and the pay was $18.75. The task included a video size setting step to make sure the entire video clip is visible to the viewer on their particular screen resolution. Due to the static nature of the experiment, we set a viewing height of 1.4m and 6° angle at 1.5m horizontal distance from the stimuli.

Figures 5.26 and 5.27 shows a comparison of VR and MTurk results for Effort and Weight respectively. We report here the raw data and a simple filtering data that drops all respondents that had a correlation with the mean below 0.15 [353]. While the general trend holds across media, it is clear that the results done in VR are systematically lower for both effort and weight than those done online using 2D videos of the stimuli.

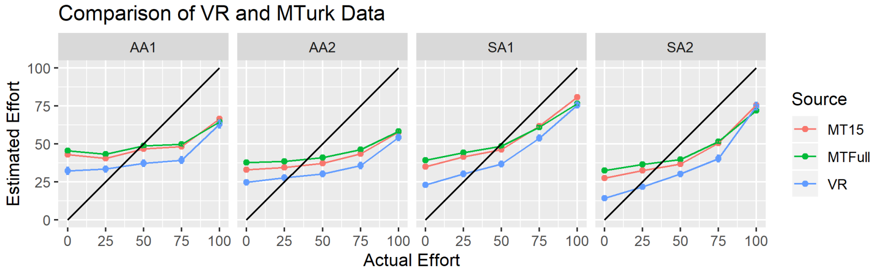


Figure 5.26: Perceived effort as a function of actual effort for online and VR studies. Facets show motion from different actors.

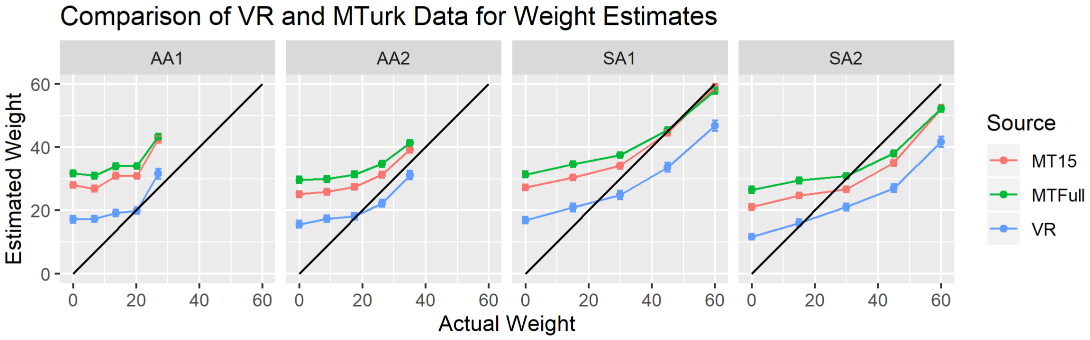


Figure 5.27: Perceived weight as a function of actual weight for online and VR studies. Facets show motion from different actors.

Statistical results confirm this difference.

The inter-rater coder correlation between the data was lower in MTurk. This might suggest more sloppy work or more difficulty producing the task. Even if we increase this threshold to 0.8, a threshold similar to the VR data, the data does not converge on the same responses as in VR. While participants consistently order the weight and effort levels as expected in each medium, it appears that the magnitude of their judgments varies. It, therefore, seems wisest to study the VR impact in VR, rather than online.

5.2.7 Overall Conclusions

While estimates are not perfect, Exp. 1 provides evidence that people can gauge both effort and weight from kinematic signals. This confirms earlier work with point-light displays and shows that this ability extends to characters in VR. The overall correlation between actual and perceived values is consistent with those reported in [21], but below the highest estimates in the literature (e.g., [324]). We did not provide a reference lift with a specified weight in any of our experiments. While this has been shown to improve the accuracy of predictions [324, 325, 327], it is not a likely scenario in real VR applications. This may account for why our correlations are somewhat lower than the highest values reported in the literature, although we did always include an indication of actor size, which is also a useful leveling cue [20].

Notably, people are less sensitive to differences at lower weight/effort levels. The Discrimination experiment (Exp. 5) showed a similar finding. People were not sensitive to the differences between 0 and 25% effort lifts, but they were sensitive to differences between higher effort lifts (somewhere between 50 and 100%, depending on the lifter), or lifts over about 30 lbs. Designers of VR experiences may be able to mitigate much of the potential negative impact of potential discrepancies by staying under these thresholds in the object manipulations they display.

We also found people tended to overestimate lower weight lifts and underestimate higher weight lifts [20, 21]. As with previous work [21], we found in Exp. 1 that people made less error estimating effort than weight. We found larger, stronger lifters were accurately perceived as lifting heavier weights when they did so, unlike previous findings where this was unexpectedly reversed [327].

Perception of effort appears to be largely driven by motion kinematics and, if present, displays of muscular strain. Visual size indicators of either the virtual character or lifted object look to have a limited impact. The virtual character size had no impact on effort for motion from three of our actors (Exp. 2). For the fourth, SA1's motion, the effort estimates were lower when shown on the smallest virtual character, AA1, for 25, 50, and 100% lifts, but did not otherwise significantly differ. When comparing the same motions with and without displaying dumbbells from Exps. 2 and 1, there was no significant impact on effort estimates for the strong actors, but effort estimates are lower for the average actors when dumbbells are present. A possible explanation is that people correct a faulty baseline assumption that the average actors were stronger than they are once dumbbells are shown. Varying dumbbell size had a limited impact on effort, largely constrained to the smallest of the four dumbbells being seen as lower effort than the remainder at some actual effort levels for each actor. When only kinematic information and these visual size indicators are present, it appears that the kinematic information dominates the perception of effort. However, when muscle deformations are added to the animation, these have a clear impact on effort (Exp. 4), particularly deformations involving the head and neck (FULL, HEAD, PARTIAL). The impact of BODY flexion on its own is more limited.

Visual size indicators, especially of the lifted object, appear to dominate the perception of weight, while kinematics still contributes. There was a consistent impact on the perception of weight for body size, where the smaller virtual characters were estimated to lift about 11% less on average. Comparing motions with and without dumbbells from Exps. 2 and 1 established that showing the dumbbells lead to a marked improvement in weight estimates with significantly higher correlations between actual and estimated weight. Exp. 3 further established that visualization of the dumbbells had a large, and likely dominant, impact on the perception of weight with every size dumbbell seen as a significantly different weight than every other dumbbell for all actors (as an example, the 60 lb dumbbell shown for SA1's 15 lb lift was estimated to be 44.6 lbs on average, whereas the 15 lb dumbbell shown with a 60 lb lift was only estimated to be about 20.5lbs). Adding muscle deformations also has a clear impact on weight perception, although the interaction of muscle deformation and dumbbell size remains to be explored. For AA1, every level of deformation was significantly different from the others except PARTIAL and HEAD. For the remaining actors, there were three groupings: {NONE, BODY}, {PARTIAL, HEAD}, and {FULL}, going from least to most impact.

Unlike Grierson et al.'s [328] findings for point-light displays, we found visualizing the size of the lifted object had a strong effect on weight perception. This may in part stem from dumbbells providing a clearer indication of weight than box size as boxes may be filled with vastly different density material.

Given the clear correspondence between effort and normalized weight estimates in Figure 5.8 when dumbbells were not shown (these are strongly correlated, with a Pearson's r=.74), it may be that people were making a single estimate of performance based largely on motion kinematics in this experiment and using this to estimate both weight and effort. When the visual dumbbell information was introduced, they relied heavily on that channel to estimate weight, but effort estimates were still largely based on kinematics. *This lead to divergence in the two estimates when information on the channels was not congruent (e.g., mismatched dumbbells).* The comparison between effort and weight estimates also provides evidence that people had fairly similar mental estimates of what a person could lift across the body variations shown and these were not consistent with the actors' actual strength range. When they were given additional information, they appeared to revise these estimates (e.g., when shown small dumbbells for the average lifters, they reduced effort estimates).

We saw that body shape changes do impact the judgment of weight, with the larger "strong" virtual characters being judged as lifting heavier weights. This is consistent with Kenny et al. [318, 319], although their animations did not include a visualization of the lifted object, so that may not be necessary for the difference. This study adds a potential causal mechanism to that finding: since the impression of effort is largely constant, the same effort combined with a larger body appears to produce a larger estimated weight. We also found degradations in naturalness not observed in Kenny et al. that may result from including a visualization of the lifted object. This grounds the task and no longer allows people to justify mismatches by imagining the weight of the object is different from what it is.

Muscle deformations add an additional signal that influenced the perception of both effort and weight. In Exp. 4, the impact of the strain deformations is greatest at lowest effort/weight levels and attenuates as these increase. This suggests that the deformation and kinematic signals are acting in concert and when there is limited evidence of effort on the kinematic channel, the deformation channel can have more impact. The discrimination experiment (Exp. 5) showed that adding deformations to zero lifts with heavier dumbbells can make it harder to notice the difference between these animations and the correct lifts without muscle deformation. For light dumbbells, adding FULL deformations to the zero lifts make it more noticeable that these are not correct, presumably because this is an unreasonable amount of strain for the visualized dumbbell. This points to the need for real systems to carefully tune the deformations to the desired weight/effort perception.

A potential application of adding strain animations is to mitigate the impact of mismatches between user kinematics and visualized motion. It is interesting that the HEAD deformation looks to carry much of the impact of FULL deformation for effort, and to a lesser degree for weight. For VR applications, this provides substantial freedom to use only face and neck deformation on characters that are clothed, wearing space suits, etc. and still achieve most of the impact. Clearly these deformations would need to be tuned to the desired effect. It may also vary highly on the application whether it is appropriate to add such strain animations. They could be potentially distracting or misleading. Such an approach would be introducing an artificial strain signal to replace a signal missing in the motion kinematics. This raises an interesting issue of how to balance verisimilitude with actual faithfulness to the person's behavior.

Drops in Naturalness ratings are driven by mismatches. For mismatched body types, naturalness ratings drop when the motion and dumbbell size of larger virtual characters is played on smaller virtual characters (motion and dumbbell sizes from all other actors played on AA1 and, to a reduced degree, SA1 and SA2 motion and dumbbells shown on AA2). The AA1 motion also looked less natural at higher effort/weights. It may be simply that the larger dumbbells looked less plausible for these smaller figures. When changing dumbbell size in Exp. 3, naturalness fell when the size of the dumbbell was a poor match for the actual lift (either large dumbbells with low-effort lifts or small dumbbells with high-effort lifts). The cases where dumbbell size looked less natural in Figure 5.18 correspond to cases where the effort was out of line with weight estimates. Effort perception was largely constant across the different displayed dumbbells for a particular weight lift, but weight varied heavily based on the displayed dumbbell. Finally, for the muscle deformations, the HEAD deformation was less natural at 0% effort lifts, which is reasonable as this shows a high level of strain that would mismatch with the kinematic signal. It is interesting that there is not a similar drop for FULL. NONE and BODY were seen as less natural at 100% effort. This is again a mismatch of a kinematic signal indicating high effort, but muscle deformations that do not reflect this effort. Taken together, these findings emphasize the importance of calibrating all signals - kinematics, visual size indicators, and muscle deformation - to avoid degrading the user experience.

There are of course limitations to the work. One significant limitation is that the study only involved male virtual characters and was not racially diverse. As a first study, this allowed us to easily have quite muscular virtual characters and display them shirtless, while also roughly matching the virtual character to the actor pool. It is important to explore if any of the findings here might change as the gender or race of the virtual character varies. There may be stereotype assumptions that come into play and this may also vary with the participant pool. A much larger study would be required to explore this. It would also be worthwhile to look at nonhuman virtual characters and the full range of beings people might want to inhabit in VR.

5.3 Discussion

This chapter described a series of experiments that explore how people understand the dynamic properties of actions based on motion kinematics, the virtual character's body, the size of manipulated objects, and muscle strain. By looking separately at people's perceptions of effort and weight, we were able to show that their judgments of these quantities are impacted by different signals. While effort is influenced by all channels, motion kinematics appears to have a dominant role, especially when muscle flexion is not shown. On the other hand, visual indicators of size, particularly of the lifted object, have a strong influence on the perception of weight. If kinematics and visualization are not matched, this can produce incongruent information, where people's estimates of effort and weight are inconsistent and can lead to degraded naturalness. It may even be that such mismatched signals are one of the contributors to the uncanny. While this set of studies indicates there is an operating range of moderate weights where such discrepancies are not likely noticed, going beyond that creates errors that must be avoided or mitigated, motivating the need for new animation algorithms.

6 Conclusions and Future Work

In this work, we investigated two specific case studies of interactions in VR systems, i.e., throwing and lifting interactions. In these cases, human motion plays a more vital role than for non-immersive platforms due to the heavy involvement of the human body in user interactions. The capability of rendering fully immersive 3D scenes with natural interactions is needed for a growing number of different tasks, activities, and training scenarios in AR and VR, with the aim of emulating real-life experiences and skills transfer. Our goal in this work was to identify some anomalies that occur in virtual interactions, to understand them, and to propose solutions to enable more plausible and diverse experiences. Our work raises important questions for future work. As the capabilities of AR/VR hardware improve, it is essential to explore the perceptual factors that come along with, or emerge from, these advancements, so that the full potential of AR/VR can be harvested.

Perception of PoR

To understand the consequences of PoR errors in human perception, we conducted a perceptual study to explore the sensitivity of viewers to PoR delays. By manipulating the point of release of captured throwing motions, we have assessed how noticeable different delays are for different distances, throwing types, and views. The results suggest that people are asymmetrically sensitive to early and late delays in overarm and underarm throws.

Future Work: To our knowledge, this was the first study to explore this problem. Therefore, it is difficult to draw general conclusions from the limited set of throws and actors, given that there are numerous other factors that might influence the results, such as the level of expertise of the observer and the ball type. It is known that professional athletes are able to read cues from other players that help them react faster and with more precision in ball-sports [343, 344, 345]. Such expertise may significantly improve the ability to notice whether a PoR timing delay is present or not, which makes this an interesting direction for future work. Furthermore, the perceptual study on PoR delays was conducted with the participants in the role of viewers. Despite including a third-person view in the experiment to simulate the experience of a thrower, the participants did not actively perform throwing interactions (due to Covid restrictions) but answered based on their observations. The response of a person to PoR delays might be different in the role of a thrower, so a valuable future direction would be to test whether and how much these results change in these circumstances.

Detection and Prediction of PoR

Focusing on the action of throwing projectiles, we first implemented a preliminary VR system and ran a study to investigate the impact of visual trajectory feedback on throwing performance. We found that limited visual feedback reduced throwing performance, but the users did not report any significant differences in their plausibility or immersion between these conditions. One limitation of this study was that the participants performed virtual throws using a pair of VR gloves that relied on a release algorithm we developed. Some participants reported that the release mechanism was not very accurate, which could have affected the results of the study. For that reason, we decided to investigate natural throwing without the restriction of using gloves or a controller. To this end, we proposed a novel real-time physical interaction system, called ReTro, to allow users to throw a virtual ball without using an intermediary device such as a controller.

To develop ReTro, we employed a data-driven approach and trained supervised neural network models that predict the PoR during a throwing motion, using motion data from the throwing arm as input. With two different sets of virtual (from our preliminary work) and real throwing data, we investigated how to best extract an accurate set of ground truth PoRs. Ultimately, we used the real throwing dataset of six actors in pairs performing 1679 total throws for ground truth PoR labeling.

Evaluation of the system with pre-recorded throwing motion data resulted in detection errors of less than 50 milliseconds. Qualitative results from six users of the real-time system in VR indicated that the task of throwing without a controller was very natural, but the system delivered a more accurate release of underarm throws than overarm throws. Our results are consistent with the literature on input modalities (Sec. 2.1.3), which reported that the employment of physical hands led to more natural but worse performance than the employment of controllers. We also reported the relative importance of different joints and motion features for the PoR prediction task, finding position and velocity to be the most informative.

Future Work: ReTro enables more natural throwing interactions than hand-held controllers, as confirmed by our participants. However, some reported a reduced sense of control in terms of the PoR, which is due to errors introduced by the prediction model. We suggest several ways in which this can be pursued in the future, e.g., implementing TensorRT to reduce the inference time of the model (currently $\sim$35ms) to solve the computation time bottleneck; improving the model by exploring more advanced neural network architectures such as Spatial-Temporal Graph Neural Networks (ST-GNN); capturing more data, especially overarm throws; improving ground truth approximation. Our perceptual study on PoR delays has shown that viewers perceive delays differently in overarm and underarm throws. Such findings can be incorporated into the throwing system by including a simultaneous throwing-type classifier to conceal PoR errors.

To evaluate ReTro, we tracked users in real-time via a high-end optical MoCap system as a proof of concept. However, it is important to pursue further testing with wearable sensors such as XSens Dot or other VR tracking devices such as Vive Trackers. Such tracking devices will introduce lower frame rates and additional transmission delays, which emphasizes the importance of improving ReTro's performance. Currently, although very promising, built-in hand tracking technologies, e.g., Oculus Quest 2, are not capable of tracking fast and occluded hand motions. Once tracking technologies advance further, detailed hand tracking will enable utilizing finger information for PoR prediction. This might allow inferring the position of the ball inside the hand during a throwing motion for a detailed estimation of the exerted forces on the ball that will lead to a more realistic trajectory simulation.

As this research was carried out in VR while addressing both AR and VR, it is important to note the differences and similarities that will arise for AR in the use of the knowledge explored in this work. We focus on the fundamental distinction that is the exposure of their external surrounding to a user in AR compared to VR. Regarding our work on throwing, everything we have learned for VR is valid for AR, including the perceptual studies and our throw detection system, ReTRo. In the short qualitative study of ReTRo, we did not utilize detailed hand tracking and only evaluated the release detection performance of our system. In AR, the visibility of the real hand might introduce additional challenges in terms of the plausibility of the ball placement inside the actual hand.

A motivating factor for finding more plausible ways of virtual throwing and developing ReTro has been to improve the potential of AR/VR in becoming tools for motor skills transfer. Therefore, as with the PoR perception experiment, the importance of the level of expertise comes up again, for both the actors in the dataset and the participants of the small case study. What feels like a reasonable or unrecognizable delay for a non-expert thrower, will very likely feel like a large amount of delay. Currently, the system has been trained on motion by non-experts, and it is an interesting direction to capture throwing data from expert throwers, including longer distance throws. The inclusion of expert throwers and more bodily-involving, longer-distance throws might enable the use of more and different motion cues in the development of the system, e.g., the importance of lower limbs and contralateral arm in outcome prediction by Maselli et al. [232].

Perception of Lifting

In our studies on the perception of lifting, we focused on the fundamental problem of dynamic inconsistencies between real and virtual worlds regarding user interaction and embodiment. We looked at the lifting motion by conducting a series of experiments that explore how people understand the dynamic properties of actions based on motion kinematics, the avatar's body, the size of manipulated objects, and muscle strain. By looking separately at people's perceptions of effort and weight, we were able to show that their judgments of these quantities are impacted by different signals. While effort is influenced by all channels, motion kinematics appears to have a dominant role, especially when muscle flexion is not shown. On the other hand, visual indicators of size, particularly of the lifted object, have a strong influence on the perception of weight. If kinematics and visualization are not matched, this can produce incongruent information, where people's estimates of effort and weight are inconsistent and can lead to degraded naturalness. It may even be that such mismatched signals are one of the contributors to the uncanny. While this paper indicates there is an operating range of moderate weights where such discrepancies are not likely noticed, going beyond that creates errors that must be avoided or mitigated, motivating the need for new animation algorithms.

Future Work: We used dumbbell lifting exercises to investigate the problem of dynamic inconsistencies by creating tasks of effort and weight estimations. Although many similar studies have been conducted in the past, this is one of the first investigations carried out and aimed at VR. In the future, more VR studies should be carried out on different interaction types and avatar shapes, using stimuli from different sets of actors including other races and genders. Developing universal solutions for physicality errors will only be possible once their effects are investigated for all types of interactions or a general understanding of the problem has been reached.

We have employed a novel approach by using muscle deformations to manipulate participants' perceptions of effort and weight estimation. However, we have used an artist-driven approach to create the deformations, so it would be useful to develop a more automated approach. Furthermore, we only experimented with muscle deformations as a way of imitating high effort, but a similar effect can be created by other signs such as gasping and changes of color in facial skin.

Our study on lifting was aimed more distinctly at VR, with one of the main motivations being a mismatch between the user and their avatar's body shapes, leading to unnatural or implausible motion kinematics. In an AR scenario with two people in the same physical space partaking in an activity involving interactions with virtual objects, such a transformation of body shapes is not feasible given the current state of AR technology. Nevertheless, dynamic inconsistencies can still arise in AR regardless of the body shape, when the interacted virtual object, e.g., a dumbbell, is different from the interacted actual object, e.g., a controller. In this type of situation, a core preliminary challenge would be to conceal the real interacted object, e.g., a controller, and overlay a different object, e.g., a dumbbell. This is a particularly difficult problem for optical see-through displays where the control over the view is highly limited compared to video pass-through displays [354, 355]. Similarly, applying muscle deformations to modulate the displayed effort would require body scanning and deformation fitting, such that muscle deformations can be overlaid realistically. Beyond such technical challenges, what we have learned from these studies is also valid for AR.

In response to our studies on both throwing and lifting, we identified problems and came up with solutions (i.e., ReTro and muscle deformations resp.) to improve user experiences in AR/VR. However, we did not attempt to explore the overall effect of our solutions on fundamental concepts such as the sense of presence, co-presence, and agency. Such an evaluation, in the form of a gamified experiment, would make our solutions more tangible. Finally, we refer back to answer the two main research questions addressed at the beginning.

- How sensitive are people to anomalies in the simulation of throwing and lifting in Virtual Reality (for both first-person and third-person views)?
    - People are highly sensitive to PoR delays in throwing in the role of an observer, with a high detection rate of delays as small as 50ms. Their sensitivity is asymmetrical and displays differences based on throwing type, throwing distance, and viewing mode.
    - In the simulation of lifting in VR, people are able to detect anomalies, i.e. dynamic inconsistency, when the visualized interaction mismatches the actual interaction by a certain amount. The sensitivity to the anomaly is affected by many factors, including the avatar's body shape, dumbbell shape, and motion kinematics.
- Can we use knowledge of human perception, together with modern Computer Science techniques, to help increase the plausibility of these interactions?
  In both of our studies on throwing and lifting, we have been able to use modern techniques to help improve the interactions positively:
    - By training a PoR detection model that uses arm motion, we sought to eliminate the need for a handheld controller for throwing. In ReTRo, this was implemented as part of a real-time throwing system and received positive feedback. Although the performance of ReTRo is not currently on par with controller-based throwing, it was found more natural and enjoyable. We

have not incorporated the results of the PoR perception study directly in the development of ReTRo, it provided a reference interval.

- We have employed blendshape-based muscle deformations to ameliorate the effect of dynamic inconsistencies in observing lifting motions. We found that it is in fact possible to change people's perception of weight and effort, as well as enhance perceived naturalness, by adding visual cues indicating muscle contractions.

Bibliography

[1] Award winning motion capture systems, Nov 2022. URL `https://www.vicon.com/`.

[2] Christopher Olah. Understanding lstm networks. 2015.

[3] Ryan P McMahan, Chengyuan Lai, and Swaroop K Pal. Interaction fidelity: the uncanny valley of virtual reality interactions. In *International conference on virtual, Augmented and Mixed Reality*, pages 59–70. Springer, 2016.

[4] Robert Kovacs, Eyal Ofek, Mar Gonzalez Franco, Alexa Fay Siu, Sebastian Marwecki, Christian Holz, and Mike Sinclair. Haptic pivot: On-demand handhelds in vr. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*, pages 1046–1059, 2020.

[5] David L Neumann, Robyn L Moffitt, Patrick R Thomas, Kylie Loveday, David P Watling, Chantal L Lombard, Simona Antonova, and Michael A Tremeer. A systematic review of the application of interactive virtual reality to sport. *Virtual Reality*, 22(3):183–198, 2018.

[6] Intel. 5g economics of entertainment report, 2019. URL `https://newsroom.intel.com/wp-content/uploads/sites/11/2018/10/intel-5g-economics-backgrounder.pdf`.

[7] J. Moore. What is the sense of agency and why does it matter? *Frontiers in psychology*, 7(Article 1272), 2016.

[8] Matthew Botvinick and Jonathan Cohen. Rubber hands 'feel'touch that eyes see. *Nature*, 391(6669):756–756, 1998.

[9] K. Kilteni, R. Groten, and M. Slater. The sense of embodiment in virtual reality. *Presence Teleoperators & Virtual Environments*, 21(4):373–387, 2012.

[10] Matthew Lombard and Theresa Ditton. At the heart of it all: The concept of presence. *Journal of computer-mediated communication*, 3(2):JCMC321, 1997.

[11] Mel Slater and Sylvia Wilbur. A framework for immersive virtual environments (five): Speculations on the role of presence in virtual environments. *Presence: Teleoperators & Virtual Environments*, 6(6):603–616, 1997.

[12] Mel Slater. A note on presence terminology. *Presence connect*, 3(3):1–5, 2003.

[13] Tomislav Bezmalinovic. For john carmack, vr doesn't need better hardware to succeed, Apr 2023. URL `https://mixed-news.com/en/john-carmack-what-vr-needs-to-succeed/`.

[14] Alvaro Villegas, Pablo Perez, et al. Realistic training in vr using physical manipulation. In *IEEE VR Workshop (VRW)*, pages 109–118, 2020.

[15] Coyle D., Moore J., Kristensson P.O., Blackwell A.F., and Fletcher P.C. I did that! measuring users' experience of agency in their own actions. In *ACM Conference on Human Factors in Computing Systems (CHI)*, pages 2025–2034, 2012.

[16] J. McCann and N.S. Pollard. Responsive characters from motion fragments. *ACM Transactions on Graphics*, 26(3):6, 2007.

[17] Tim Zindulka, Myroslav Bachynskyi, and Jörg Müller. Performance and experience of throwing in virtual reality. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, pages 1–8, 2020.

[18] Karolis Butkus and Tautvydas Ceponis. Accuracy of throwing distance perception in virtual reality. In *Proceedings of the International Conference on Information Technologies, IVUS 2019*, volume 2470 of *CEUR Workshop Proceedings*, pages 121–124, 2019.

[19] Alexandra Covaci, Anne-Hélène Olivier, and Franck Multon. Visual perspective and feedback guidance for vr free-throw training. *IEEE computer graphics and applications*, 35(5):55–65, 2015.

[20] Geoffrey P Bingham. Scaling judgments of lifted weight: Lifter size and the role of the standard. *Ecological Psychology*, 5(1):31–64, 1993.

[21] Jaeho Shim, Les G Carlton, and Jitae Kim. Estimation of lifted weight and produced effort through perception of point-light display. *Perception*, 33(3):277–291, 2004.

[22] Douglas W Cunningham and Christian Wallraven. *Experimental design: From user studies to psychophysics*. AK Peters/CRC Press, 2019.

[23] Paul Milgram, Haruo Takemura, Akira Utsumi, and Fumio Kishino. Augmented reality: A class of displays on the reality-virtuality continuum. In *Telemanipulator and telepresence technologies*, volume 2351, pages 282–292. International Society for Optics and Photonics, 1995.

[24] Steve Mann. Mediated reality with implementations for everyday life. *Presence Connect*, 1, 2002.

[25] Maximilian Speicher, Brian D Hall, and Michael Nebeling. What is mixed reality? In *Proceedings of the 2019 CHI conference on human factors in computing systems*, pages 1–15, 2019.

[26] Richard Skarbez, Missie Smith, and Mary C Whitton. Revisiting milgram and kishino's reality-virtuality continuum. *Frontiers in Virtual Reality*, 2:647997, 2021.

[27] Mel Slater and Maria V Sanchez-Vives. Enhancing our lives with immersive virtual reality. *Frontiers in Robotics and AI*, 3:74, 2016.

[28] Dimitris Chatzopoulos, Carlos Bermejo, Zhanpeng Huang, and Pan Hui. Mobile augmented reality survey: From where we are to where we go. *Ieee Access*, 5: 6917–6950, 2017.

[29] Xueni Pan and Antonia F de C Hamilton. Why and how to use virtual reality to study human social interaction: The challenges of exploring a new research landscape. *British Journal of Psychology*, 109(3):395–417, 2018.

[30] Brian Mallari, Emily K Spaeth, Henry Goh, and Benjamin S Boyd. Virtual reality as an analgesic for acute and chronic pain in adults: a systematic review and meta-analysis. *Journal of pain research*, 12:2053, 2019.

[31] Eugenia Yiannakopoulou, Nikolaos Nikiteas, Despina Perrea, and Christos Tsigris. Virtual reality simulators and training in laparoscopic surgery. *International Journal of Surgery*, 13:60–64, 2015.

[32] Bob G Witmer, Christian J Jerome, and Michael J Singer. The factor structure of the presence questionnaire. *Presence: Teleoperators & Virtual Environments*, 14 (3):298–312, 2005.

[33] Richard Skarbez, Frederick P Brooks, Jr, and Mary C Whitton. A survey of presence and related concepts. *ACM Computing Surveys (CSUR)*, 50(6):1–39, 2017.

[34] Mel Slater. Place illusion and plausibility can lead to realistic behaviour in immersive virtual environments. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 364(1535):3549–3557, 2009.

[35] Bob G Witmer and Michael J Singer. Measuring presence in virtual environments: A presence questionnaire. *Presence*, 7(3):225–240, 1998.

[36] Vinicius Souza, Anderson Maciel, Luciana Nedel, and Regis Kopper. Measuring presence in virtual environments: A survey. *ACM Computing Surveys (CSUR)*, 54(8):1–37, 2021.

[37] Christine Youngblut. Experience of presence in virtual environments. Technical report, INSTITUTE FOR DEFENSE ANALYSES ALEXANDRIA VA, 2003.

[38] Mel Slater and Martin Usoh. Presence in immersive virtual environments. In *Proceedings of IEEE virtual reality annual international symposium*, pages 90–96. IEEE, 1993.

[39] Jeremy N Bailenson, Nick Yee, Jim Blascovich, Andrew C Beall, Nicole Lundblad, and Michael Jin. The use of immersive virtual reality in the learning sciences: Digital transformations of teachers, students, and social context. *The Journal of the Learning Sciences*, 17(1):102–141, 2008.

[40] Fabio Buttussi and Luca Chittaro. Effects of different types of virtual reality display on presence and learning in a safety training scenario. *IEEE transactions on visualization and computer graphics*, 24(2):1063–1076, 2017.

[41] Matias N Selzer, Nicolas F Gazcon, and Martin L Larrea. Effects of virtual presence and learning outcome using low-end virtual reality systems. *Displays*, 59:9–15, 2019.

[42] Elinda Ai-Lim Lee, Kok Wai Wong, and Chun Che Fung. How does desktop virtual reality enhance learning outcomes? a structural equation modeling approach. *Computers & Education*, 55(4):1424–1442, 2010.

[43] Merel Krijn, Paul MG Emmelkamp, Roeline Biemond, Claudius de Wilde de Ligny, Martijn J Schuemie, and Charles APG van der Mast. Treatment of acrophobia in virtual reality: The role of immersion and presence. *Behaviour research and therapy*, 42(2):229–239, 2004.

[44] Giuseppe Riva, Fabrizia Mantovani, and Andrea Gaggioli. Presence and rehabilitation: toward second-generation virtual reality applications in neuropsychology. *Journal of neuroengineering and rehabilitation*, 1(1):1–11, 2004.

[45] Matthew Lombard and Jennifer Snyder-Duch. Interactive advertising and presence: A framework. *Journal of interactive Advertising*, 1(2):56–65, 2001.

[46] Iis P Tussyadiah, Dan Wang, Timothy H Jung, and M Claudia Tom Dieck. Virtual reality, presence, and attitude change: Empirical evidence from tourism. *Tourism management*, 66:140–154, 2018.

[47] David Weibel, Bartholomäus Wissmath, Stephan Habegger, Yves Steiner, and Rudolf Groner. Playing online games against computer-vs. human-controlled opponents: Effects on presence, flow, and enjoyment. *Computers in human behavior*, 24(5):2274–2291, 2008.

[48] Mel Slater. How colorful was your day? why questionnaires cannot assess presence in virtual environments. *Presence*, 13(4):484–493, 2004.

[49] Ralph Schroeder. Copresence and interaction in virtual environments: An overview of the range of issues. In *Presence 2002: Fifth international workshop*, pages 274–295. Citeseer, 2002.

[50] Shanyang Zhao. Toward a taxonomy of copresence. *Presence*, 12(5):445–455, 2003.

[51] Jeremy N Bailenson, Eyal Aharoni, Andrew C Beall, Rosanna E Guadagno, Aleksandar Dimov, and Jim Blascovich. Comparing behavioral and self-report measures of embodied agents' social presence in immersive virtual environments. In *Proceedings of the 7th Annual International Workshop on PRESENCE*, pages 1864–1105. IEEE, 2004.

[52] Kristine L Nowak. The influence of anthropomorphism and agency on social judgment in virtual environments. *Journal of Computer-Mediated Communication*, 9(2):JCMC925, 2004.

[53] Jeremy N Bailenson, Jim Blascovich, Andrew C Beall, and Jack M Loomis. Equilibrium theory revisited: Mutual gaze and personal space in virtual environments. *Presence: Teleoperators & Virtual Environments*, 10(6):583–598, 2001.

[54] Maia Garau, Mel Slater, Vinoba Vinayagamoorthy, Andrea Brogni, Anthony Steed, and M Angela Sasse. The impact of avatar realism and eye gaze control on perceived quality of communication in a shared immersive virtual environment. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 529–536, 2003.

[55] Jeremy N Bailenson, Kim Swinth, Crystal Hoyt, Susan Persky, Alex Dimov, and Jim Blascovich. The independent and interactive effects of embodied-agent appearance and behavior on self-report, cognitive, and behavioral markers of copresence in immersive virtual environments. *Presence*, 14(4):379–393, 2005.

[56] S Gallagher. Philosophical conceptions of the self: Implications for cognitive science. *Trends in Cognitive Sciences*, 4(1):14–21, 2000.

[57] N. David, A. Newen, and K. Vogeley. The ‘sense of agency’ and its underlying cognitive and neural mechanisms. *Consciousness and Cognition*, 17(2):523–534, 2008.

[58] A. Sato and A. Yasuda. Illusion of sense of self-agency: Discrepancy between the predicted and actual sensory consequences of actions modulates the sense of self-agency, but not the sense of self-ownership. *Cognition*, 94:241–255, 2005.

[59] N. Franck, C. Farrer, N. Georgieff, M. Marie-Cardine, J. Dalery, T. d’Amato, and M. Jeannerod. Defective recognition of one’s own actions in patients with schizophrenia. *American Journal of Psychiatry*, 158:454–459, 2001.

[60] H. Aarts, R. Custers, and D. M. Wegner. On the inference of personal authorship: Enhancing experienced agency by priming effect information. *Consciousness and Cognition*, 14(3):439–458, 2005.

[61] M.V. Sanchez-Vives and M. Slater. From presence to consciousness through virtual reality. *Nature Reviews Neuroscience*, 6:332–339, 2005.

[62] C. Jeunet, L. Albert, F. Argelaguet, and A. Lécuyer. ‘do you feel in control?’: Towards novel approaches to characterise, manipulate and measure the sense of agency in virtual environments. *IEEE Transactions on Visualization and Computer Graphics*, 24(4):1486–1495, 2018.

[63] M. Vicovaro, L. Hoyet, L. Burigana, and C. O’Sullivan. Perceptual evaluation of motion editing for realistic throwing animations. *ACM Transactions on Applied Perception*, 11(2):10, 2014.

[64] Joseph J LaViola Jr, Ernst Kruijff, Ryan P McMahan, Doug Bowman, and Ivan P Poupyrev. *3D user interfaces: theory and practice*. Addison-Wesley Professional, 2017.

[65] Sofia Seinfeld, Tiare Feuchtner, Johannes Pinzek, and Jorg Muller. Impact of information placement and user representations in vr on performance and embodiment. *IEEE transactions on visualization and computer graphics*, 2020.

[66] Lorraine Lin, Aline Normoyle, Alexandra Adkins, Yu Sun, Andrew Robb, Yuting Ye, Massimiliano Di Luca, and Sophie Jörg. The effect of hand size and interaction modality on the virtual hand illusion. In *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pages 510–518. IEEE, 2019.

[67] Mathias Moehring and Bernd Froehlich. Effective manipulation of virtual objects within arm's reach. In *2011 IEEE Virtual Reality Conference*, pages 131–138. IEEE, 2011.

[68] Tafadzwa Joseph Dube and Ahmed Sabbir Arif. Text entry in virtual reality: A comprehensive review of the literature. In *International Conference on Human-Computer Interaction*, pages 419–437. Springer, 2019.

[69] Lorraine Lin and Sophie Jörg. Need a hand? how appearance affects the virtual hand illusion. In *Proceedings of the ACM symposium on applied perception*, pages 69–76, 2016.

[70] Ferran Argelaguet, Ludovic Hoyet, Michaël Trico, and Anatole Lécuyer. The role of interaction in virtual embodiment: Effects of the virtual hand representation. In *2016 IEEE virtual reality (VR)*, pages 3–10. IEEE, 2016.

[71] Christos Lougiakis, Akrivi Katifori, Maria Roussou, and Ioannis-Panagiotis Ioannidis. Effects of virtual hand representation on interaction and embodiment in hmd-based virtual environments using controllers. In *2020 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pages 510–518. IEEE, 2020.

[72] Alex Adkins, Lorraine Lin, Aline Normoyle, Ryan Canales, Yuting Ye, and Sophie Jörg. Evaluating grasping visualizations and control modes in a vr game. *ACM Transactions on Applied Perception (TAP)*, 18(4):1–14, 2021.

[73] Dennis C Neale and John M Carroll. The role of metaphors in user interface design. In *Handbook of human-computer interaction*, pages 441–462. Elsevier, 1997.

[74] Thomas D Erickson. Working with interface metaphors. In *Readings in Human–Computer Interaction*, pages 147–151. Elsevier, 1995.

[75] Frank Steinicke, Timo Ropinski, and Klaus Hinrichs. Object selection in virtual environments using an improved virtual pointer metaphor. In *Computer vision and graphics*, pages 320–326. Springer, 2006.

[76] Jiandong Liang and Mark Green. Jdcad: A highly interactive 3d modeling system. *Computers & graphics*, 18(4):499–506, 1994.

[77] Mark R Mine. Virtual environment interaction techniques. *UNC Chapel Hill CS Dept*, 1995.

[78] Marc Baloup, Thomas Pietrzak, and Géry Casiez. Raycursor: A 3d pointing facilitation technique based on raycasting. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pages 1–12, 2019.

[79] Ivan Poupyrev, Mark Billinghurst, Suzanne Weghorst, and Tadao Ichikawa. The go-go interaction technique: non-linear mapping for direct manipulation in vr. In *Proceedings of the 9th annual ACM symposium on User interface software and technology*, pages 79–80, 1996.

[80] Doug A Bowman and Larry F Hodges. An evaluation of techniques for grabbing and manipulating remote objects in immersive virtual environments. In *Proceedings of the 1997 symposium on Interactive 3D graphics*, pages 35–ff, 1997.

[81] Richard H Jacoby, Mark Ferneau, and Jim Humphries. Gestural interaction in a virtual environment. In *Stereoscopic Displays and Virtual Reality Systems*, volume 2177, pages 355–364. SPIE, 1994.

[82] Ivan Poupyrev, Tadao Ichikawa, Suzanne Weghorst, and Mark Billinghurst. Egocentric object manipulation in virtual environments: empirical evaluation of interaction techniques. In *Computer graphics forum*, volume 17, pages 41–52. Wiley Online Library, 1998.

[83] Richard Stoakley, Matthew J Conway, and Randy Pausch. Virtual reality on a wim: interactive worlds in miniature. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 265–272, 1995.

[84] Mark R Mine, Frederick P Brooks Jr, and Carlo H Sequin. Moving objects in space: exploiting proprioception in virtual-environment interaction. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pages 19–26, 1997.

[85] Patrick Baudisch, Alexander Zotov, Edward Cutrell, and Ken Hinckley. Starburst: a target expansion algorithm for non-uniform target distributions. In *Proceedings of the working conference on Advanced visual interfaces*, pages 129–137, 2008.

[86] Renaud Blanch and Michael Ortega. Benchmarking pointing techniques with distractors: adding a density factor to fitts' pointing paradigm. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1629–1638, 2011.

[87] Ferran Argelaguet and Carlos Andujar. Efficient 3d pointing selection in cluttered virtual environments. *IEEE Computer Graphics and Applications*, 29(6):34–43, 2009.

[88] Olivier Chapuis, Jean-Baptiste Labrune, and Emmanuel Pietriga. Dynaspot: speed-dependent area cursor. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1391–1400, 2009.

[89] Tovi Grossman and Ravin Balakrishnan. The bubble cursor: enhancing target acquisition by dynamic resizing of the cursor's activation area. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 281–290, 2005.

[90] Joona Laukkanen, Poika Isokoski, and Kari-Jouko Räihä. The cone and the lazy bubble: two efficient alternatives between the point cursor and the bubble cursor. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 309–312, 2008.

[91] Martez E Mott and Jacob O Wobbrock. Beating the bubble: using kinematic triggering in the bubble lens for acquiring small, dense targets. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 733–742, 2014.

[92] Lode Vanacken, Tovi Grossman, and Karin Coninx. Multimodal selection techniques for dense and occluded 3d virtual environments. *International Journal of Human-Computer Studies*, 67(3):237–255, 2009.

[93] Maxime Guillon, François Leitner, and Laurence Nigay. Investigating visual feedforward for target expansion techniques. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pages 2777–2786, 2015.

[94] Jan Theeuwes, Artem Belopolsky, and Christian NL Olivers. Interactions between working memory, attention and eye movements. *Acta psychologica*, 132 (2):106–114, 2009.

[95] Jacob L Orquin and Simone Mueller Loose. Attention and choice: A review on eye movements in decision making. *Acta psychologica*, 144(1):190–206, 2013.

[96] Jia Zheng Lim, James Mountstephens, and Jason Teo. Emotion recognition using eye-tracking: taxonomy, review and current challenges. *Sensors*, 20(8):2384, 2020.

[97] Deborah E Hannula, Robert R Althoff, David E Warren, Lily Riggs, Neal J Cohen, and Jennifer D Ryan. Worth a glance: using eye movements to investigate the cognitive neuroscience of memory. *Frontiers in human neuroscience*, 4:166, 2010.

[98] Ludwig Sidenmark, Christopher Clarke, Xuesong Zhang, Jenny Phu, and Hans Gellersen. Outline pursuits: Gaze-assisted selection of occluded objects in virtual reality. In *Proceedings of the 2020 chi conference on human factors in computing systems*, pages 1–13, 2020.

[99] Hemant Bhaskar Surale, Aakar Gupta, Mark Hancock, and Daniel Vogel. Tabletinvr: Exploring the design space for using a multi-touch tablet in virtual reality. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pages 1–13, 2019.

[100] Travis Gesslein, Verena Biener, Philipp Gagel, Daniel Schneider, Per Ola Kristensson, Eyal Ofek, Michel Pahud, and Jens Grubert. Pen-based interaction with spreadsheets in mobile virtual reality. In *2020 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 361–373. IEEE, 2020.

[101] Jorge Wagner, Wolfgang Stuerzlinger, and Luciana Nedel. Comparing and combining virtual hand and virtual ray pointer interactions for data manipulation in immersive analytics. *IEEE Transactions on Visualization and Computer Graphics*, 27(5):2513–2523, 2021.

[102] Duc-Minh Pham and Wolfgang Stuerzlinger. Is the pen mightier than the controller? a comparison of input devices for selection in virtual and augmented reality. In *25th ACM Symposium on Virtual Reality Software and Technology*, pages 1–11, 2019.

[103] Nianlong Li, Teng Han, Feng Tian, Jin Huang, Minghui Sun, Pourang Irani, and Jason Alexander. Get a grip: Evaluating grip gestures for vr input using a lightweight pen. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, pages 1–13, 2020.

[104] Markus Höll, Markus Oberweger, Clemens Arth, and Vincent Lepetit. Efficient physics-based implementation for realistic hand-object interaction in virtual reality. In *2018 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pages 175–182. IEEE, 2018.

[105] Jun-Sik Kim and Jung-Min Park. Physics-based hand interaction with virtual objects. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3814–3819. IEEE, 2015.

[106] C Karen Liu. Dextrous manipulation from a grasping pose. In *ACM SIGGRAPH 2009 papers*, pages 1–6. 2009.

[107] Lixin Yang, Xinyu Zhan, Kailin Li, Wenqiang Xu, Jiefeng Li, and Cewu Lu. Cpf: Learning a contact potential field to model the hand-object interaction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11097–11106, 2021.

[108] Ammar Ahmad, Cyrille Migniot, and Albert Dipanda. Hand pose estimation and tracking in real and virtual interaction: A review. *Image and Vision Computing*, 89:35–49, 2019.

[109] Juan David Hincapié-Ramos, Xiang Guo, Paymahn Moghadasian, and Pourang Irani. Consumed endurance: a metric to quantify arm fatigue of mid-air interactions. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1063–1072, 2014.

[110] Ederyn Williams. Experimental comparisons of face-to-face and mediated communication: A review. *Psychological Bulletin*, 84(5):963, 1977.

[111] William T Rogers. The contribution of kinesic illustrators toward the comprehension of verbal behavior within utterances. *Human communication research*, 5 (1):54–62, 1978.

[112] Sukeshini A Grandhi, Gina Joue, and Irene Mittelberg. Understanding naturalness and intuitiveness in gesture production: insights for touchless gestural interfaces. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 821–824, 2011.

[113] Mark Billinghurst, Tham Piumsomboon, and Huidong Bai. Hands in space: Gesture interaction with augmented-reality interfaces. *IEEE computer graphics and applications*, 34(1):77–80, 2014.

[114] Michael Nielsen, Moritz Störring, Thomas B Moeslund, and Erik Granum. A procedure for developing intuitive and ergonomic gesture interfaces for hci. In *International gesture workshop*, pages 409–420. Springer, 2003.

[115] Helman I Stern, Juan P Wachs, and Yael Edan. Designing hand gesture vocabularies for natural interaction by combining psycho-physiological and recognition factors. *International Journal of Semantic Computing*, 2(01):137–160, 2008.

[116] I Scott MacKenzie and William Buxton. Extending fitts' law to two-dimensional tasks. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 219–226, 1992.

[117] Lawrence Sambrooks and Brett Wilkinson. Comparison of gestural, touch, and mouse interaction with fitts' law. In *Proceedings of the 25th Australian Computer-Human Interaction Conference: Augmentation, Application, Innovation, Collaboration*, pages 119–122, 2013.

[118] Paul M Fitts. The information capacity of the human motor system in controlling the amplitude of movement. *Journal of experimental psychology*, 47(6):381, 1954.

[119] Eleftherios Triantafyllidis and Zhibin Li. The challenges in modeling human performance in 3d space with fitts' law. In *Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems*, pages 1–9, 2021.

[120] Jonathan Steuer. Defining virtual reality: Dimensions determining telepresence. *Journal of communication*, 42(4):73–93, 1992.

[121] Christoph Klimmt and Peter Vorderer. Media psychology "is not yet there": Introducing theories on media entertainment to the presence debate. *Presence*, 12 (4):346–359, 2003.

[122] Paul Skalski, Ron Tamborini, Ashleigh Shelton, Michael Buncher, and Pete Lindmark. Mapping the road to fun: Natural video game controllers, presence, and game enjoyment. *New Media & Society*, 13(2):224–242, 2011.

[123] Rory McGloin, Kirstie M Farrar, and Marina Krcmar. The impact of controller naturalness on spatial presence, gamer enjoyment, and perceived realism in a tennis simulation video game. *Presence: Teleoperators and Virtual Environments*, 20(4):309–324, 2011.

[124] Daniel M Shafer, Corey P Carbonara, and Lucy Popova. Spatial presence and perceived reality as predictors of motion-based video game enjoyment. *Presence: Teleoperators and Virtual Environments*, 20(6):591–619, 2011.

[125] Daniel M Shafer, Corey P Carbonara, and Lucy Popova. Controller required? the impact of natural mapping on interactivity, realism, presence, and enjoyment in motion-based video games. *Presence: Teleoperators and Virtual Environments*, 23(3):267–286, 2014.

[126] Jonmichael Seibert and Daniel M Shafer. Control mapping in virtual reality: effects on spatial presence and controller naturalness. *Virtual Reality*, 22(1):79–88, 2018.

[127] Felix Reer, Lars-Ole Wehden, Robin Janzik, Wai Yen Tang, and Thorsten Quandt. Virtual reality technology and game enjoyment: The contributions of natural

mapping and need satisfaction. *Computers in Human Behavior*, 132:107242, 2022.

[128] Federica Pallavicini, Alessandro Pepe, and Maria Eleonora Minissi. Gaming in virtual reality: What changes in terms of usability, emotional response and sense of presence compared to non-immersive video games? *Simulation & Gaming*, 50(2):136–159, 2019.

[129] William J Shelstad, Dustin C Smith, and Barbara S Chaparro. Gaming on the rift: How virtual reality affects game user satisfaction. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, volume 61, pages 2072–2076. SAGE Publications Sage CA: Los Angeles, CA, 2017.

[130] Ryan Patrick McMahan. *Exploring the effects of higher-fidelity display and interaction for virtual reality games*. PhD thesis, Virginia Tech, 2011.

[131] Katja Rogers, Jana Funke, Julian Frommel, Sven Stamm, and Michael Weber. Exploring interaction fidelity in virtual reality: Object manipulation and whole-body movements. In *Proceedings of the 2019 CHI conference on human factors in computing systems*, pages 1–14, 2019.

[132] Heather Culbertson, Samuel B Schorr, and Allison M Okamura. Haptics: The present and future of artificial touch sensation. *Annual Review of Control, Robotics, and Autonomous Systems*, 1(1):385–409, 2018.

[133] Cagatay Basdogan, Frederic Giraud, Vincent Levesque, and Seungmoon Choi. A review of surface haptics: Enabling tactile effects on touch surfaces. *IEEE transactions on haptics*, 13(3):450–470, 2020.

[134] Thomas H Massie, J Kenneth Salisbury, et al. The phantom haptic interface: A device for probing virtual objects. In *Proceedings of the ASME winter annual meeting, symposium on haptic interfaces for virtual environment and teleoperator systems*, volume 55, pages 295–300. Chicago, IL, 1994.

[135] Richard Q Van der Linde, Piet Lammertse, Erwin Frederiksen, and B Ruiter. The hapticmaster, a new high-performance haptic interface. In *Proc. Eurohaptics*, pages 1–5. Edinburgh University, 2002.

[136] Eric Whitmire, Hrvoje Benko, Christian Holz, Eyal Ofek, and Mike Sinclair. Haptic revolver: Touch, shear, texture, and shape rendering on a reconfigurable virtual reality controller. In *Proceedings of the 2018 CHI conference on human factors in computing systems*, pages 1–12, 2018.

[137] Brent Edward Insko. *Passive haptics significantly enhances virtual environments*. The University of North Carolina at Chapel Hill, 2001.

[138] Adalberto L Simeone, Eduardo Velloso, and Hans Gellersen. Substitutional reality: Using the physical environment to design virtual reality experiences. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pages 3307–3316, 2015.

[139] Xiaochi Gu, Yifei Zhang, Weize Sun, Yuanzhe Bian, Dao Zhou, and Per Ola Kristensson. Dexmo: An inexpensive and lightweight mechanical exoskeleton for motion capture and force feedback in vr. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, pages 1991–1995, 2016.

[140] Andre Zenner and Antonio Krüger. Shifty: A weight-shifting dynamic passive haptic proxy to enhance object perception in virtual reality. *IEEE transactions on visualization and computer graphics*, 23(4):1285–1294, 2017.

[141] Alexa F Siu, Eric J Gonzalez, Shenli Yuan, Jason B Ginsberg, and Sean Follmer. Shapeshift: 2d spatial manipulation and self-actuation of tabletop shape displays for tangible and haptic interaction. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, pages 1–13, 2018.

[142] Mourad Bouzit, George Popescu, Grigore Burdea, and Rares Boian. The rutgers master ii-nd force feedback glove. In *Proceedings 10th Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems. HAPTICS 2002*, pages 145–152. IEEE, 2002.

[143] Cybertouch. URL `http://www.cyberglovesystems.com/cybertouch/`.

[144] Inrak Choi and Sean Follmer. Wolverine: A wearable haptic interface for grasping in vr. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, pages 117–119, 2016.

[145] Hrvoje Benko, Christian Holz, Mike Sinclair, and Eyal Ofek. Normaltouch and texturetouch: High-fidelity 3d haptic shape rendering on handheld virtual reality controllers. In *Proceedings of the 29th annual symposium on user interface software and technology*, pages 717–728, 2016.

[146] Colin Swindells, Alex Unden, and Tao Sang. Torquebar: an ungrounded haptic feedback device. In *Proceedings of the 5th international conference on Multimodal interfaces*, pages 52–59, 2003.

[147] Akash Badshah, Sidhant Gupta, Daniel Morris, Shwetak Patel, and Desney Tan. Gyrotab: A handheld device that provides reactive torque feedback. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 3153–3156, 2012.

[148] Merwan Achibet, Adrien Girard, Anthony Talvas, Maud Marchal, and Anatole Lécuyer. Elastic-arm: Human-scale passive haptic feedback for augmenting interaction and perception in virtual environments. In *2015 IEEE Virtual Reality (VR)*, pages 63–68. IEEE, 2015.

[149] Albert S Carlin, Hunter G Hoffman, and Suzanne Weghorst. Virtual reality and tactile augmentation in the treatment of spider phobia: a case report. *Behaviour research and therapy*, 35(2):153–158, 1997.

[150] Hunter G Hoffman. Physically touching virtual objects using tactile augmentation enhances the realism of virtual environments. In *Proceedings. IEEE 1998 Virtual Reality Annual International Symposium (Cat. No. 98CB36180)*, pages 59–63. IEEE, 1998.

[151] Barbara Olasov Rothbaum, Larry F Hodges, Rob Kooper, Dan Opdyke, James S Williford, and Max North. Virtual reality graded exposure in the treatment of acrophobia: A case report. *Behavior therapy*, 26(3):547–554, 1995.

[152] André Zenner and Antonio Krüger. Drag: on: A virtual reality controller providing haptic feedback based on drag and weight shift. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pages 1–12, 2019.

[153] Pingjun Xia. New advances for haptic rendering: state of the art. *The Visual Computer*, 34(2):271–287, 2018.

[154] Wang Dangxiao, GUO Yuan, LIU Shiyi, Yuru Zhang, Xu Weiliang, and Xiao Jing. Haptic display for virtual reality: progress and challenges. *Virtual Reality & Intelligent Hardware*, 1(2):136–162, 2019.

[155] Evan Pezent, Ali Israr, Majed Samad, Shea Robinson, Priyanshu Agarwal, Hrvoje Benko, and Nick Colonnese. Tasbi: Multisensory squeeze and vibrotactile wrist haptics for augmented and virtual reality. In *2019 IEEE World Haptics Conference (WHC)*, pages 1–6. IEEE, 2019.

[156] Mengjia Zhu, Amirhossein H Memar, Aakar Gupta, Majed Samad, Priyanshu Agarwal, Yon Visell, Sean J Keller, and Nicholas Colonnese. Pneusleeve: In-fabric multimodal actuation and sensing in a soft, compact, and expressive haptic sleeve. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, pages 1–12, 2020.

[157] Sharif Razzaque. *Redirected walking*. The University of North Carolina at Chapel Hill, 2005.

[158] Luv Kohli, Eric Burns, Dorian Miller, and Henry Fuchs. Combining passive haptics with redirected walking. In *Proceedings of the 2005 international conference on Augmented tele-existence*, pages 253–254, 2005.

[159] Luv Kohli. Redirected touching: Warping space to remap passive haptics. In *2010 IEEE Symposium on 3D User Interfaces (3DUI)*, pages 129–130. IEEE, 2010.

[160] Mahdi Azmandian, Mark Hancock, Hrvoje Benko, Eyal Ofek, and Andrew D Wilson. Haptic retargeting: Dynamic repurposing of passive haptics for enhanced virtual reality experiences. In *Proceedings of the 2016 chi conference on human factors in computing systems*, pages 1968–1979, 2016.

[161] Eric Burns, Sharif Razzaque, Mary C Whitton, and Frederick P Brooks. Macbeth: The avatar which i see before me and its movement toward my hand. In *2007 IEEE Virtual Reality Conference*, pages 295–296. IEEE, 2007.

[162] André Zenner, Kristin Ullmann, and Antonio Krüger. Combining dynamic passive haptics and haptic retargeting for enhanced haptic feedback in virtual reality. *IEEE Transactions on Visualization and Computer Graphics*, 27(5):2627–2637, 2021.

[163] Parastoo Abtahi, Benoit Landry, Jackie Yang, Marco Pavone, Sean Follmer, and James A Landay. Beyond the force: Using quadcopters to appropriate objects and the environment for haptics in virtual reality. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pages 1–13, 2019.

[164] Kotaro Yamaguchi, Ginga Kato, Yoshihiro Kuroda, Kiyoshi Kiyokawa, and Haruo Takemura. A non-grounded and encountered-type haptic display using a drone. In *Proceedings of the 2016 Symposium on Spatial User Interaction*, pages 43–46, 2016.

[165] Matthias Hoppe, Pascal Knierim, Thomas Kosch, Markus Funk, Lauren Futami, Stefan Schneegass, Niels Henze, Albrecht Schmidt, and Tonja Machulla. Vrhapticdrones: Providing haptics in virtual reality through quadcopters. In *Proceedings of the 17th International Conference on Mobile and Ubiquitous Multimedia*, pages 7–18, 2018.

[166] Ryo Suzuki, Hooman Hedayati, Clement Zheng, James L Bohn, Daniel Szafir, Ellen Yi-Luen Do, Mark D Gross, and Daniel Leithinger. Roomshift: Room-scale dynamic haptics for vr with furniture-moving swarm robots. In *Proceedings of*

the 2020 CHI conference on human factors in computing systems, pages 1–11, 2020.

[167] Ryo Suzuki, Eyal Ofek, Mike Sinclair, Daniel Leithinger, and Mar Gonzalez-Franco. Hapticbots: Distributed encountered-type haptics for vr with multiple shape-changing mobile robots. In *The 34th Annual ACM Symposium on User Interface Software and Technology*, pages 1269–1281, 2021.

[168] Eric J Gonzalez, Parastoo Abtahi, and Sean Follmer. Reach+ extending the reachability of encountered-type haptics devices through dynamic redirection in vr. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*, pages 236–248, 2020.

[169] Hsin-Yu Huang, Chih-Wei Ning, Po-Yao Wang, Jen-Hao Cheng, and Lung-Pan Cheng. Haptic-go-round: A surrounding platform for encounter-type haptics in virtual reality experiences. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, pages 1–10, 2020.

[170] Tao Liu, Yoshio Inoue, and Kyoko Shibata. A small and low-cost 3-d tactile sensor for a wearable force plate. *IEEE Sensors Journal*, 9(9):1103–1110, 2009.

[171] Kiana Ehsani, Shubham Tulsiani, Saurabh Gupta, Ali Farhadi, and Abhinav Gupta. Use the force, luke! learning to predict physical forces by simulating effects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 224–233, 2020.

[172] Yiyue Luo, Yunzhu Li, Pratyusha Sharma, Wan Shou, Kui Wu, Michael Foshey, Beichen Li, Tomás Palacios, Antonio Torralba, and Wojciech Matusik. Learning human–environment interactions using conformal tactile textiles. *Nature Electronics*, 4(3):193–201, 2021.

[173] Yunxiang Zhang, Benjamin Liang, Boyuan Chen, Paul Torrens, S Farokh Atashzar, Dahua Lin, and Qi Sun. Force-aware interface via electromyography for natural vr/ar interaction. *arXiv preprint arXiv:2210.01225*, 2022.

[174] David Antonio Gómez Jáuregui, Ferran Argelaguet Sanz, Anne-Hélène Olivier, Maud Marchal, and FranckMulton. Toward ”pseudo-haptic avatars”: Modifying the visual animation of self-avatar can simulate the perception of weight lifting. *IEEE Transactions on Visualization and Computer Graphics*, 20(4):654–661, 2014.

[175] Leigh R Hochberg, Mijail D Serruya, Gerhard M Friehs, Jon A Mukand, Maryam Saleh, Abraham H Caplan, Almut Branner, David Chen, Richard D Penn, and

John P Donoghue. Neuronal ensemble control of prosthetic devices by a human with tetraplegia. *Nature*, 442(7099):164–171, 2006.

[176] Leigh R Hochberg, Daniel Bacher, Beata Jarosiewicz, Nicolas Y Masse, John D Simeral, Joern Vogel, Sami Haddadin, Jie Liu, Sydney S Cash, Patrick Van Der Smagt, et al. Reach and grasp by people with tetraplegia using a neurally controlled robotic arm. *Nature*, 485(7398):372–375, 2012.

[177] Sharlene N Flesher, John E Downey, Jeffrey M Weiss, Christopher L Hughes, Angelica J Herrera, Elizabeth C Tyler-Kabara, Michael L Boninger, Jennifer L Collinger, and Robert A Gaunt. A brain-computer interface that evokes tactile sensations improves robotic arm control. *Science*, 372(6544):831–836, 2021.

[178] Yannick Roy, Hubert Banville, Isabela Albuquerque, Alexandre Gramfort, Tiago H Falk, and Jocelyn Faubert. Deep learning-based electroencephalography analysis: a systematic review. *Journal of neural engineering*, 16(5):051001, 2019.

[179] Chuanjiang Li, Jian Ren, Huaiqi Huang, Bin Wang, Yanfei Zhu, and Huosheng Hu. Pca and deep learning based myoelectric grasping control of a prosthetic hand. *Biomedical engineering online*, 17(1):1–18, 2018.

[180] Dezhen Xiong, Daohui Zhang, Xingang Zhao, and Yiwen Zhao. Deep learning for emg-based human-machine interaction: A review. *IEEE/CAA Journal of Automatica Sinica*, 8(3):512–533, 2021.

[181] Jinhua Zhang, Baozeng Wang, Cheng Zhang, Yanqing Xiao, and Michael Yu Wang. An eeg/emg/eog-based multimodal human-machine interface to real-time control of a soft robot hand. *Frontiers in neurorobotics*, 13:7, 2019.

[182] Thilina Dulantha Lalitharatne, Kenbu Teramoto, Yoshiaki Hayashi, and Kazuo Kiguchi. Towards hybrid eeg-emg-based control approaches to be used in biorobotics applications: Current status, challenges and future directions. *Paladyn, Journal of Behavioral Robotics*, 4(2):147–154, 2013.

[183] Babis Koniaris, Ivan Huerta, Maggie Kosek, Karen Darragh, Charles Malleson, Joanna Jamrozy, Nick Swafford, Jose Guitian, Bochang Moon, Ali Israr, et al. Iridium: immersive rendered interactive deep media. In *ACM SIGGRAPH 2016 VR Village*, pages 1–2. 2016.

[184] Mamoru Hirota, Ayumu Tsuboi, Masayuki Yokoyama, and Masao Yanagisawa. Gesture recognition of air-tapping and its application to character input in vr space. In *SIGGRAPH Asia 2018 Posters*, pages 1–2. 2018.

[185] Anany Dwivedi, Yongje Kwon, and Minas Liarokapis. Emg-based decoding of manipulation motions in virtual reality: Towards immersive interfaces. In *2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 3296–3303. IEEE, 2020.

[186] Yilin Liu, Shijia Zhang, and Mahanth Gowda. Neuropose: 3d hand pose tracking using emg wearables. In *Proceedings of the Web Conference 2021*, pages 1471–1482, 2021.

[187] Paras Gulati, Qin Hu, and S Farokh Atashzar. Toward deep generalization of peripheral emg-based human-robot interfacing: A hybrid explainable solution for neurorobotic systems. *IEEE Robotics and Automation Letters*, 6(2):2650–2657, 2021.

[188] Graham Thomas, Rikke Gade, Thomas B Moeslund, Peter Carr, and Adrian Hilton. Computer vision for sports: Current applications and research topics. *Computer Vision and Image Understanding*, 159:3–18, 2017.

[189] Janex Pers and Stanislav Kovacic. Computer vision system for tracking players in sports games. In *IWISPA 2000. Proceedings of the First International Workshop on Image and Signal Processing and Analysis. in conjunction with 22nd International Conference on Information Technology Interfaces.(IEEE*, pages 177–182. IEEE, 2000.

[190] Manuel Stein, Halldor Janetzko, Andreas Lamprecht, Thorsten Breitkreutz, Philipp Zimmermann, Bastian Goldlücke, Tobias Schreck, Gennady Andrienko, Michael Grossniklaus, and Daniel A Keim. Bring it to the pitch: Combining video and movement data to enhance team sport analysis. *IEEE transactions on visualization and computer graphics*, 24(1):13–22, 2017.

[191] James Hong, Matthew Fisher, Michaël Gharbi, and Kayvon Fatahalian. Video pose distillation for few-shot, fine-grained sports action recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9254–9263, 2021.

[192] Manju Rana and Vikas Mittal. Wearable sensors for real-time kinematics analysis in sports: a review. *IEEE Sensors Journal*, 21(2):1187–1207, 2020.

[193] StanceBeam. Smart cricket bat sensor: Cricket bat technology. URL `https://www.stancebeam.com/`.

[194] Teaching the world how to move. URL `https://www.turingsense.com/#pivot`.

[195] Libin Jiao, Hao Wu, Rongfang Bie, Anton Umek, and Anton Kos. Multi-sensor golf swing classification using deep cnn. *Procedia Computer Science*, 129:59–65, 2018.

[196] Juri Taborri, Eduardo Palermo, and Stefano Rossi. Automatic detection of faults in race walking: A comparative analysis of machine-learning algorithms fed with inertial sensor data. *Sensors*, 19(6):1461, 2019.

[197] Farzin Dadashi, Arash Arami, Florent Crettenand, Gregoire P Millet, John Komar, Ludovic Seifert, and Kamiar Aminian. A hidden markov model of the breaststroke swimming temporal phases using wearable inertial measurement units. In *2013 IEEE international conference on body sensor networks*, pages 1–6. Ieee, 2013.

[198] Rupika Srivastava, Ayush Patwari, Sunil Kumar, Gaurav Mishra, Laksmi Kaligounder, and Purnendu Sinha. Efficient characterization of tennis shots and game analysis using wearable sensors data. In *2015 IEEE sensors*, pages 1–4. IEEE, 2015.

[199] Emily E Cust, Alice J Sweeting, Kevin Ball, and Sam Robertson. Machine and deep learning for sport-specific movement recognition: A systematic review of model development and performance. *Journal of sports sciences*, 37(5):568–600, 2019.

[200] Peter Jae Fadde. *Interactive video training of perceptual decision making in the sport of baseball*. PhD thesis, Purdue University, 2002.

[201] Stefan C Michalski, Ancret Szpak, and Tobias Loetscher. Using virtual environments to improve real-world motor skills in sports: a systematic review. *Frontiers in psychology*, 10:2159, 2019.

[202] Rob Gray. Transfer of training from virtual to real baseball batting. *Frontiers in psychology*, 8:2183, 2017.

[203] Greg Wood, David J Wright, David Harris, A Pal, Zoë Claire Franklin, and Samuel J Vine. Testing the construct validity of a soccer-specific virtual reality simulator using novice, academy, and professional soccer players. *Virtual Reality*, 25(1):43–51, 2021.

[204] Paul Larkin, Christopher Mesagno, Michael Spittle, Jason Berry, et al. An evaluation of video-based training programs for perceptual-cognitive skill development. a systematic review of current sport-based knowledge. *International Journal of Sport Psychology*, 46(6):555–586, 2015.

[205] Ferran Argelaguet Sanz, Franck Multon, and Anatole Lécuyer. A methodology for introducing competitive anxiety and pressure in vr sports training. *Frontiers in Robotics and AI*, 2:10, 2015.

[206] Sean Dean Lynch, Anne-Hélène Olivier, Benoit Bideau, and Richard Kulpa. Detection of deceptive motions in rugby from visual motion cues. *Plos one*, 14(9): e0220878, 2019.

[207] Peter J Fadde and Leonard Zaichkowsky. Training perceptual-cognitive skills in sports using technology. *Journal of Sport Psychology in Action*, 9(4):239–248, 2018.

[208] K Anders Ericsson, Ralf T Krampe, and Clemens Tesch-Römer. The role of deliberate practice in the acquisition of expert performance. *Psychological review*, 100(3):363, 1993.

[209] Stephen Mark Hadlow, Derek Panchuk, David Lindsay Mann, Marc Ronald Portus, and Bruce Abernethy. Modified perceptual training in sport: a new classification framework. *Journal of Science and Medicine in Sport*, 21(9):950–958, 2018.

[210] L Gregory Appelbaum and Graham Erickson. Sports vision training: A review of the state-of-the-art in digital training techniques. *International Review of Sport and Exercise Psychology*, 11(1):160–189, 2018.

[211] Charles Faure, Annabelle Limballe, Benoit Bideau, and Richard Kulpa. Virtual reality to assess and train team ball sports performance: A scoping review. *Journal of sports Sciences*, 38(2):192–205, 2020.

[212] Dominik Schuldhaus, Constantin Zwick, et al. Inertial sensor-based approach for shot/pass classification during a soccer match. In *KDD Workshop on Large-Scale Sports Analytics*, pages 1–4, 2015.

[213] Damien Connaghan, Phillip Kelly, Noel E O'Connor, Mark Gaffney, Michael Walsh, and Cian O'Mathuna. Multi-sensor classification of tennis strokes. In *SENSORS, 2011 IEEE*, pages 1437–1440. IEEE, 2011.

[214] David A Winter. *Biomechanics and motor control of human movement*. John Wiley & Sons, 2009.

[215] Glenn Fleisig. The biomechanics of throwing. In *ISBS-Conference Proceedings Archive*, 2001.

[216] Glenn S Fleisig, Steven W Barrentine, Rafael F Escamilla, and James R Andrews. Biomechanics of overhand throwing with implications for injuries. *Sports medicine*, 21(6):421–437, 1996.

[217] Anne E Atwater. Biomechanics of overarm throwing movements and of throwing injuries. *Exercise and sport sciences reviews*, 7(1):43–86, 1979.

[218] J Hore, S Watts, and D Tweed. Prediction and compensation by an internal model for back forces during finger opening in an overarm throw. *Journal of Neurophysiology*, 82(3):1187–1197, 1999.

[219] J Hore, S Watts, J Martin, and B Miller. Timing of finger opening and ball release in fast and accurate overarm throws. *Experimental Brain Research*, 103(2):277–286, 1995.

[220] Jon Hore and Sherry Watts. Skilled throwers use physics to time ball release to the nearest millisecond. *Journal of Neurophysiology*, 106(4):2024–2033, 2011.

[221] Darren J Stefanyshyn and John W Wannop. Biomechanics research and sport equipment development. *Sports Engineering*, 18(4):191–202, 2015.

[222] RM Herring and AE Chapman. Effects of changes in segmental values and timing of both torque and torque reversal in simulated throws. *Journal of Biomechanics*, 25(10):1173–1184, 1992.

[223] Carol A Putnam. Sequential motions of body segments in striking and throwing skills: descriptions and explanations. *Journal of biomechanics*, 26:125–135, 1993.

[224] Bruce Elliott, J Robert Grove, Barry Gibson, and B Thurston. A three-dimensional cinematographic analysis of the fastball and curveball pitches in baseball. *Journal of Applied Biomechanics*, 2(1):20–28, 1986.

[225] RJ Best, RM Bartlett, and CJ Morriss. A three-dimensional analysis of javelin throwing technique. *Journal of sports sciences*, 11(4):315–328, 1993.

[226] Glenn S Fleisig, Rafael F Escamilla, James R Andrews, Tomoyuki Matsuo, Yvonne Satterwhite, and Steve W Barrentine. Kinematic and kinetic comparison between baseball pitching and football passing. *Journal of Applied Biomechanics*, 12(2):207–224, 1996.

[227] Roland van den Tillaar and Gertjan Ettema. Is there a proximal-to-distal sequence in overarm throwing in team handball? *Journal of sports sciences*, 27(9):949–955, 2009.

[228] Laetitia Fradet, Maïtel Botcazou, Carole Durocher, Armel Cretual, Franck Multon, Jacques Prioux, and Paul Delamarche. Do handball throws always exhibit a proximal-to-distal segmental sequence? *Journal of sports sciences*, 22(5):439–447, 2004.

[229] J Hore, R Ritchie, and S Watts. Finger opening in an overarm throw is not triggered by proprioceptive feedback from elbow extension or wrist flexion. *Experimental Brain Research*, 125(3):302–312, 1999.

[230] William H Calvin. A stone's throw and its launch window: Timing precision and its implications for language and hominid brains. *Journal of theoretical Biology*, 104(1):121–135, 1983.

[231] WJ Becker, E Kunesch, and H-J Freund. Coordination of a multi-joint movement in normal humans and in patients with cerebellar dysfunction. *Canadian journal of neurological sciences*, 17(3):264–274, 1990.

[232] Antonella Maselli, Aishwar Dhawan, Benedetta Cesqui, Marta Russo, Francesco Lacquaniti, and Andrea d'Avella. Where are you throwing the ball? i better watch your body, not just your arm! *Frontiers in Human Neuroscience*, 11:505, 2017.

[233] Antonella Maselli, Paolo De Pasquale, Francesco Lacquaniti, and Andrea d'Avella. Interception of virtual throws reveals predictive skills based on the visual processing of throwing kinematics. *Iscience*, 25(10), 2022.

[234] Charmi Jobanputra, Jatna Bavishi, and Nishant Doshi. Human activity recognition: A survey. *Procedia Computer Science*, 155:698–703, 2019.

[235] Jindong Wang, Yiqiang Chen, Shuji Hao, Xiaohui Peng, and Lisha Hu. Deep learning for sensor-based activity recognition: A survey. *Pattern recognition letters*, 119:3–11, 2019.

[236] Mais Yasen and Shaidah Jusoh. A systematic review on hand gesture recognition techniques, challenges and applications. *PeerJ Computer Science*, 5:e218, 2019.

[237] Jesse Hoey, Thomas Plötz, et al. Rapid specification and automated generation of prompting systems to assist people with dementia. *Pervasive and Mobile Computing*, 7(3):299–318, 2011.

[238] Wei-Yi Cheng, Alf Scotland, Florian Lipsmeier, et al. Human activity recognition from sensor-based large-scale continuous monitoring of parkinson's disease patients. In *2017 IEEE/ACM International Conference on Connected Health: Applications, Systems and Engineering Technologies (CHASE)*, pages 249–250. IEEE, 2017.

[239] Akin Avci, Stephan Bosch, Marin-Perianu, et al. Activity recognition using inertial sensing for healthcare, wellbeing and sports applications: A survey. In *23th Int. conf. on architecture of computing systems*, pages 1–10. VDE, 2010.

[240] Fasih Haider, Fahim A Salim, Dees BW Postma, et al. A super-bagging method for volleyball action recognition using wearable sensors. *Multimodal Technologies and Interaction*, 4(2):33, 2020.

[241] Konstantinos Avgerinakis, Alexia Briassouli, and Ioannis Kompatsiaris. Recognition of activities of daily living for smart home environments. In *Intelligent Environments*, pages 173–180, 2013.

[242] Aiguo Wang, Guilin Chen, Cuijuan Shang, Miaofei Zhang, and Li Liu. Human activity recognition in a smart home environment with stacked denoising autoencoders. In *Int. conf. on web-age information management*, pages 29–40, 2016.

[243] Dinesh Singh and C Krishna Mohan. Graph formulation of video activities for abnormal activity recognition. *Pattern Recognition*, 65:265–272, 2017.

[244] Shangchen Han, Beibei Liu, Randi Cabezas, Christopher D Twigg, Peizhao Zhang, Jeff Petkau, Tsz-Ho Yu, Chun-Jung Tai, Muzaffer Akbay, Zheng Wang, et al. Megatrack: monochrome egocentric articulated hand-tracking for virtual reality. *ACM Transactions on Graphics (ToG)*, 39(4):87–1, 2020.

[245] Lin Guo, Zongxing Lu, and Ligang Yao. Human-machine interaction sensing technology based on hand gesture recognition: A review. *IEEE Transactions on Human-Machine Systems*, 2021.

[246] Alberto Ranavolo, Mariano Serrao, and Francesco Draicchio. Critical issues and imminent challenges in the use of semg in return-to-work rehabilitation of patients affected by neurological disorders in the epoch of human–robot collaborative technologies. *Frontiers in Neurology*, 11:572069, 2020.

[247] Inside facebook reality labs: Wrist-based interaction for the next computing platform, Mar 2021. URL `https://tech.facebook.com/reality-labs/2021/3/inside-facebook-reality-labs-wrist-based-interaction-for-the-next-computi`

[248] Yeh-Kuang Wu, Hui-Chun Wang, Liung-Chun Chang, and Ke-Chun Li. Using hmms and depth information for signer-independent sign language recognition. In *International Workshop on Multi-disciplinary Trends in Artificial Intelligence*, pages 79–86. Springer, 2013.

[249] Jakub Gałka, Mariusz Mąsior, Mateusz Zaborski, and Katarzyna Barczewska. Inertial motion sensing glove for sign language gesture acquisition and recognition. *IEEE Sensors Journal*, 16(16):6310–6316, 2016.

[250] Poonam Suryanarayan, Anbumani Subramanian, and Dinesh Mandalapu. Dynamic hand pose recognition using depth data. In *2010 20th International Conference on Pattern Recognition*, pages 3105–3108. IEEE, 2010.

[251] Liwei Liu, Junliang Xing, Haizhou Ai, and Xiang Ruan. Hand posture recognition using finger geometric feature. In *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)*, pages 565–568. IEEE, 2012.

[252] Guillaume Plouffe and Ana-Maria Cretu. Static and dynamic hand gesture recognition in depth data using dynamic time warping. *IEEE transactions on instrumentation and measurement*, 65(2):305–316, 2015.

[253] Jeroen F Lichtenauer, Emile A Hendriks, and Marcel JT Reinders. Sign language recognition by combining statistical dtw and independent classification. *IEEE transactions on pattern analysis and machine intelligence*, 30(11):2040–2046, 2008.

[254] Tarik Arici, Sait Celebi, Ali S Aydin, and Talha T Temiz. Robust gesture recognition using feature pre-processing and weighted dynamic time warping. *Multimedia Tools and Applications*, 72(3):3045–3062, 2014.

[255] Antonio Hernández-Vela, Miguel Angel Bautista, Xavier Perez-Sala, Víctor Ponce-López, Sergio Escalera, Xavier Baró, Oriol Pujol, and Cecilio Angulo. Probability-based dynamic time warping and bag-of-visual-and-depth-words for human gesture recognition in rgb-d. *Pattern Recognition Letters*, 50:112–121, 2014.

[256] Munir Oudah, Ali Al-Naji, and Javaan Chahl. Hand gesture recognition based on computer vision: a review of techniques. *journal of Imaging*, 6(8):73, 2020.

[257] Hong Cheng, Lu Yang, and Zicheng Liu. Survey on 3d hand gesture recognition. *IEEE transactions on circuits and systems for video technology*, 26(9): 1659–1673, 2015.

[258] Marcio C Cabral, Carlos H Morimoto, and Marcelo K Zuffo. On the usability of gesture interfaces in virtual reality environments. In *Proceedings of the 2005 Latin American conference on Human-computer interaction*, pages 100–108, 2005.

[259] Kanad K Biswas and Saurav Kumar Basu. Gesture recognition using microsoft kinect®. In *The 5th international conference on automation, robotics and applications*, pages 100–103. IEEE, 2011.

[260] Yi Li. Hand gesture recognition using kinect. In *2012 IEEE International Conference on Computer Science and Automation Engineering*, pages 196–199. IEEE, 2012.

[261] Orasa Patsadu, Chakarida Nukoolkit, and Bunthit Watanapa. Human gesture recognition using kinect camera. In *2012 ninth international conference on computer science and software engineering (JCSSE)*, pages 28–32. IEEE, 2012.

[262] Dinh-Son Tran, Ngoc-Huynh Ho, Hyung-Jeong Yang, Eu-Tteum Baek, Soo-Hyung Kim, and Gueesang Lee. Real-time hand gesture spotting and recognition using rgb-d camera and 3d convolutional neural network. *Applied Sciences*, 10 (2):722, 2020.

[263] Lei Wang, Du Q Huynh, and Piotr Koniusz. A comparative review of recent kinect-based action recognition algorithms. *IEEE Transactions on Image Processing*, 29:15–28, 2019.

[264] Chenxuan Xi, Jianxin Chen, Chenxue Zhao, Qicheng Pei, and Lizheng Liu. Real-time hand tracking using kinect. In *Proceedings of the 2nd International Conference on Digital Signal Processing*, pages 37–42, 2018.

[265] Guanglong Du, Ping Zhang, Jianhua Mai, and Zeling Li. Markerless kinect-based hand tracking for robot teleoperation. *International Journal of Advanced Robotic Systems*, 9(2):36, 2012.

[266] Frank Weichert, Daniel Bachmann, Bartholomäus Rudak, and Denis Fisseler. Analysis of the accuracy and robustness of the leap motion controller. *Sensors*, 13(5):6380–6393, 2013.

[267] Wei Lu, Zheng Tong, and Jinghui Chu. Dynamic hand gesture recognition with leap motion controller. *IEEE Signal Processing Letters*, 23(9):1188–1192, 2016.

[268] Giulio Marin, Fabio Dominio, and Pietro Zanuttigh. Hand gesture recognition with leap motion and kinect devices. In *2014 IEEE International conference on image processing (ICIP)*, pages 1565–1569. IEEE, 2014.

[269] Ching-Hua Chuan, Eric Regina, and Caroline Guardino. American sign language recognition using leap motion sensor. In *2014 13th International Conference on Machine Learning and Applications*, pages 541–544. IEEE, 2014.

[270] Mohamed Mohandes, S Aliyu, and M Deriche. Arabic sign language recognition using the leap motion controller. In *2014 IEEE 23rd International Symposium on Industrial Electronics (ISIE)*, pages 960–965. IEEE, 2014.

[271] Li Liu, Wanli Ouyang, Xiaogang Wang, Paul Fieguth, Jie Chen, Xinwang Liu, and Matti Pietikäinen. Deep learning for generic object detection: A survey. *International journal of computer vision*, 128:261–318, 2020.

[272] Raghavendra Chalapathy and Sanjay Chawla. Deep learning for anomaly detection: A survey. *arXiv preprint arXiv:1901.03407*, 2019.

[273] Xiao Yan Wu. A hand gesture recognition algorithm based on dc-cnn. *Multimedia Tools and Applications*, 79(13):9193–9205, 2020.

[274] Abdullah Mujahid, Mazhar Javed Awan, Awais Yasin, Mazin Abed Mohammed, Robertas Damaševičius, Rytis Maskeliūnas, and Karrar Hameed Abdulkareem. Real-time hand gesture recognition based on deep learning yolov3 model. *Applied Sciences*, 11(9):4164, 2021.

[275] Liquan Zhao and Shuaiyang Li. Object detection algorithm based on improved yolov3. *Electronics*, 9(3):537, 2020.

[276] Noorkholis Luthfil Hakim, Timothy K Shih, Sandeli Priyanwada Kasthuri Arachchi, Wisnu Aditya, Yi-Cheng Chen, and Chih-Yang Lin. Dynamic hand gesture recognition using 3dcnn and lstm with fsm context-aware model. *Sensors*, 19(24):5429, 2019.

[277] Lionel Pigou, Aäron Van Den Oord, Sander Dieleman, Mieke Van Herreweghe, and Joni Dambre. Beyond temporal pooling: Recurrence and temporal convolutions for gesture recognition in video. *International Journal of Computer Vision*, 126(2):430–439, 2018.

[278] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[279] Shigeki Karita, Nanxin Chen, Tomoki Hayashi, Takaaki Hori, Hirofumi Inaguma, Ziyan Jiang, Masao Someki, Nelson Enrique Yalta Soplin, Ryuichi Yamamoto, Xiaofei Wang, et al. A comparative study on transformer vs rnn in speech applications. In *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 449–456. IEEE, 2019.

[280] Albert Zeyer, Parnia Bahar, Kazuki Irie, Ralf Schlüter, and Hermann Ney. A comparison of transformer and lstm encoder decoder models for asr. In *2019 IEEE*

Automatic Speech Recognition and Understanding Workshop (ASRU), pages 8–15. IEEE, 2019.

[281] Tianyang Lin, Yuxin Wang, Xiangyang Liu, and Xipeng Qiu. A survey of transformers. *AI Open*, 2022.

[282] Jie Zhou, Ganqu Cui, Shengding Hu, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. Graph neural networks: A review of methods and applications. *AI Open*, 1:57–81, 2020.

[283] Sijie Yan, Yuanjun Xiong, and Dahua Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *Thirty-second AAAI conference on artificial intelligence*, 2018.

[284] Maosen Li, Siheng Chen, Yangheng Zhao, Ya Zhang, Yanfeng Wang, and Qi Tian. Dynamic multiscale graph neural networks for 3d skeleton based human motion prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 214–223, 2020.

[285] Yong Li, Zihang He, Xiang Ye, Zuguo He, and Kangrong Han. Spatial temporal graph convolutional networks for skeleton-based dynamic hand gesture recognition. *EURASIP Journal on Image and Video Processing*, 2019(1):1–7, 2019.

[286] Puchuan Tan, Xi Han, Yang Zou, Xuecheng Qu, Jiangtao Xue, Tong Li, Yiqian Wang, Ruizeng Luo, Xi Cui, Yuan Xi, et al. Self-powered gesture recognition wristband enabled by machine learning for full keyboard and multicommand input. *Advanced Materials*, page 2200793, 2022.

[287] Hanxiang Wu, Hang Guo, Zongming Su, Mayue Shi, Xuexian Chen, Xiaoliang Cheng, Mengdi Han, and Haixia Zhang. Fabric-based self-powered noncontact smart gloves for gesture recognition. *Journal of materials chemistry A*, 6(41): 20277–20288, 2018.

[288] Ali Moin, Andy Zhou, Abbas Rahimi, Alisha Menon, Simone Benatti, George Alexandrov, Senam Tamakloe, Jonathan Ting, Natasha Yamamoto, Yasser Khan, et al. A wearable biosensing system with in-sensor adaptive machine learning for hand gesture recognition. *Nature Electronics*, 4(1):54–63, 2021.

[289] Fateme Rasti Najafabadi, Abbas Rahimi, Pentti Kanerva, and Jan M Rabaey. Hyperdimensional computing for text classification. In *Design, automation test in Europe conference exhibition (DATE), University Booth*, pages 1–1, 2016.

[290] Mohsen Imani, Deqian Kong, Abbas Rahimi, and Tajana Rosing. Voicehd: Hyperdimensional computing for efficient speech recognition. In *2017 IEEE international conference on rebooting computing (ICRC)*, pages 1–8. IEEE, 2017.

[291] Abbas Rahimi, Pentti Kanerva, José del R Millán, and Jan M Rabaey. Hyperdimensional computing for noninvasive brain-computer interfaces: Blind and one-shot classification of eeg error-related potentials. In *10th EAI Int. Conf. on Bio-inspired Information and Communications Technologies*, number CONF, 2017.

[292] Lulu Ge and Keshab K Parhi. Classification using hyperdimensional computing: A review. *IEEE Circuits and Systems Magazine*, 20(2):30–47, 2020.

[293] Xianjie Pu, Shanshan An, Qian Tang, Hengyu Guo, and Chenguo Hu. Wearable triboelectric sensors for biomedical monitoring and human-machine interface. *Iscience*, 24(1):102027, 2021.

[294] Oliver Faust, U Rajendra Acharya, Hojjat Adeli, and Amir Adeli. Wavelet-based eeg processing for computer-aided seizure detection and epilepsy diagnosis. *Seizure*, 26:56–64, 2015.

[295] Samaneh Aminikhanghahi and Diane J Cook. Enhancing activity recognition using cpd-based activity segmentation. *Pervasive and Mobile Computing*, 53: 75–89, 2019.

[296] Jaxk Reeves, Jien Chen, Xiaolan L Wang, Robert Lund, and Qi Qi Lu. A review and comparison of changepoint detection techniques for climate data. *Journal of applied meteorology and climatology*, 46(6):900–915, 2007.

[297] Victor Brovkin, Edward Brook, John W Williams, Sebastian Bathiany, Timothy M Lenton, Michael Barton, Robert M DeConto, Jonathan F Donges, Andrey Ganopolski, Jerry McManus, et al. Past abrupt changes, tipping points and cascading impacts in the earth system. *Nature Geoscience*, 14(8):550–558, 2021.

[298] Daniel Amorese. Applying a change-point detection method on frequency-magnitude distributions. *Bulletin of the Seismological Society of America*, 97 (5):1742–1749, 2007.

[299] Samaneh Aminikhanghahi, Tinghui Wang, and Diane J Cook. Real-time change point detection with application to smart home time series data. *IEEE Transactions on Knowledge and Data Engineering*, 31(5):1010–1023, 2018.

[300] Diane J Cook and Narayanan C Krishnan. *Activity learning: discovering, recognizing, and predicting human behavior from sensor data*. John Wiley & Sons, 2015.

[301] Kyle D Feuz, Diane J Cook, Cody Rosasco, Kayela Robertson, and Maureen Schmitter-Edgecombe. Automated detection of activity transitions for prompting. *IEEE transactions on human-machine systems*, 45(5):575–585, 2014.

[302] Richard J Radke, Srinivas Andra, Omar Al-Kofahi, and Badrinath Roysam. Image change detection algorithms: a systematic survey. *IEEE transactions on image processing*, 14(3):294–307, 2005.

[303] Md Foezur Rahman Chowdhury, S-A Selouani, and D O'Shaughnessy. Bayesian on-line spectral change point detection: a soft computing approach for on-line asr. *International Journal of Speech Technology*, 15(1):5–23, 2012.

[304] James Douglas Hamilton. *Time series analysis*. Princeton university press, 2020.

[305] Samaneh Aminikhanghahi and Diane J Cook. A survey of methods for time series change point detection. *Knowledge and information systems*, 51(2):339–367, 2017.

[306] Unai Bermejo, Aitor Almeida, Aritz Bilbao-Jayo, and Gorka Azkune. Embedding-based real-time change point detection with application to activity segmentation in smart home time series data. *Expert Systems with Applications*, 185:115641, 2021.

[307] Jason Harrison, Ronald A Rensink, and Michiel Van De Panne. Obscuring length changes during animated motion. *ACM Transactions on Graphics (TOG)*, 23(3): 569–573, 2004.

[308] Paul SA Reitsma and Nancy S Pollard. Perceptual metrics for character animation: sensitivity to errors in ballistic motion. In *ACM SIGGRAPH 2003 Papers*, pages 537–542. 2003.

[309] Eakta Jain, Lisa Anthony, Aishat Aloba, Amanda Castonguay, Isabella Cuba, Alex Shaw, and Julia Woodward. Is the motion of a child perceivably different from the motion of an adult? *ACM Transactions on Applied Perception*, 13(4): 1–17, 2016.

[310] Rachel McDonnell, Sophie Jörg, Jessica K Hodgins, Fiona Newell, and Carol O'sullivan. Evaluating the effect of motion and body shape on the perceived sex of virtual characters. *ACM Transactions on Applied Perception*, 5(4):1–14, 2009.

[311] Jessica Hodgins, Sophie Jörg, Carol O'Sullivan, Sang Il Park, and Moshe Mahler. The saliency of anomalies in animated human characters. *ACM Transactions on Applied Perception (TAP)*, 7(4):1–14, 2010.

[312] Ludovic Hoyet, Rachel McDonnell, and Carol O'Sullivan. Push it real: Perceiving causality in virtual interactions. *ACM Transactions on Graphics (TOG)*, 31 (4):1–9, 2012.

[313] L. Hoyet, F. Multon, T. Komura, and A. Lecuyer. Can we distinguish biological motions of virtual humans? perceptual study with captured motions of weight lifting. In *Proceedings of ACM Virtual Reality and Technology (VRST)*, pages 87–90, 2010.

[314] Ludovic Hoyet, Kenneth Ryall, Katja Zibrek, Hwangpil Park, Jehee Lee, Jessica Hodgins, and Carol O'sullivan. Evaluating the distinctiveness and attractiveness of human motions on realistic virtual bodies. *ACM Transactions on Graphics (TOG)*, 32(6):1–11, 2013.

[315] Katja Zibrek, Benjamin Niay, Anne-Hélène Olivier, Ludovic Hoyet, Julien Pettré, and Rachel Mcdonnell. The effect of gender and attractiveness of motion on proximity in virtual reality. *ACM Transactions on Applied Perception*, 17(4):1–15, 2020.

[316] Ryan Canales, Aline Normoyle, Yu Sun, Yuting Ye, Massimiliano Di Luca, and Sophie Jörg. Virtual grasping feedback and virtual hand ownership. In *ACM Symposium on Applied Perception 2019*, pages 1–9, 2019.

[317] Anne Thaler, Sergi Pujades, Jeanine K Stefanucci, Sarah H Creem-Regehr, Joachim Tesch, Michael J Black, and Betty J Mohler. The influence of visual perspective on body size estimation in immersive virtual reality. In *ACM Symposium on Applied Perception 2019*, pages 1–12, 2019.

[318] Sophie Kenny, Naureen Mahmood, Claire Honda, Michael J Black, and Nikolaus F Troje. Effects of animation retargeting on perceived action outcomes. In *Proceedings of the ACM Symposium on Applied Perception*, pages 1–7, 2017.

[319] Sophie Kenny, Naureen Mahmood, Claire Honda, Michael J Black, and Nikolaus F Troje. Perceptual effects of inconsistency in human animations. *ACM Transactions on Applied Perception*, 16(1):1–18, 2019.

[320] Sverker Runeson and Gunilla Frykholm. Kinematic specification of dynamics as an informational basis for person-and-action perception: expectation, gender recognition, and deceptive intention. *Journal of experimental psychology: general*, 112(4):585, 1983.

[321] James T Todd and William H Warren Jr. Visual perception of relative mass in dynamic events. *Perception*, 11(3):325–335, 1982.

[322] David L Gilden and Dennis R Proffitt. Understanding collision dynamics. *Journal of Experimental Psychology: Human Perception and Performance*, 15(2): 372, 1989.

[323] Randolph Blake and Maggie Shiffrar. Perception of human motion. *Annual review of psychology*, 58, 2007.

[324] Sverker Runeson and Gunilla Frykholm. Visual perception of lifted weight. *Journal of Experimental Psychology: Human Perception and Performance*, 7(4):733, 1981.

[325] Geoffrey P Bingham. Kinematic form and scaling: Further investigations on the visual perception of lifted weight. *Journal of Experimental Psychology: Human Perception and Performance*, 13(2):155, 1987.

[326] Jaeho Shim and Les G Carlton. Perception of kinematic characteristics in the motion of lifted weight. *Journal of motor behavior*, 29(2):131–146, 1997.

[327] Jaeho Shim, Heiko Hecht, Jung-Eun Lee, Dong-Won Yook, and Ji-Tae Kim. The limits of visual mass perception. *Quarterly Journal of Experimental Psychology*, 62(11):2210–2221, 2009.

[328] Lawrence EM Grierson, Simran Ohson, and James Lyons. The relative influences of movement kinematics and extrinsic object characteristics on the perception of lifted weight. *Attention, Perception, & Psychophysics*, 75(8):1906–1913, 2013.

[329] Kaat Alaerts, Stephan P Swinnen, and Nicole Wenderoth. Observing how others lift light or heavy objects: which visual cues mediate the encoding of muscular force in the primary motor cortex? *Neuropsychologia*, 48(7):2082–2090, 2010.

[330] Antonia Hamilton, Daniel Wolpert, and Uta Frith. Your own action influences how you perceive another person's action. *Current biology*, 14(6):493–498, 2004.

[331] AM Gordon, H Forssberg, RS Johansson, and G Westling. Visual size cues in the programming of manipulative forces during precision grip. *Experimental brain research*, 83(3):477–482, 1991.

[332] Patrick C Little and Chaz Firestone. Physically implied surfaces. *Psychological Science*, page 0956797620939942, 2021.

[333] Carol O'Sullivan, John Dingliana, Thanh Giang, and Mary K. Kaiser. Evaluating the visual fidelity of physically based animations. *ACM Transactions on Graphics*, 22(3):527–536, 2003.

[334] M. Nusseck, J. Lagarde, B. Bardy, R. Fleming, and H.H. Bülthoff. Perception and prediction of simple object interactions. In *ACM Symposium on Applied Perception in Graphics and Visualization (APGV)*, pages 27–34, 2007.

[335] Maddock Meredith, Steve Maddock, et al. Motion capture file formats explained. *Department of Computer Science, University of Sheffield*, 211:241–244, 2001.

[336] F Sebastian Grassia. Practical parameterization of rotations using the exponential map. *Journal of graphics tools*, 3(3):29–48, 1998.

[337] Joseph Hamill and Kathleen M Knutzen. *Biomechanical basis of human movement*. Lippincott Williams & Wilkins, 2006.

[338] Adam Paszke et al. PyTorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. 2019.

[339] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256, 2010.

[340] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[341] R. Baillargeon. Infants' understanding of the physical world. In *Advances in psychological science. Biological and cognitive aspects*, volume 2, pages 503–529. 1998.

[342] C. Faure, A. Limballe, B. Bideau, and R. Kulpa. Virtual reality to assess and train team ball sports performance: A scoping review. *Journal of sports Sciences*, 38 (2):192–205, 2020.

[343] Sean Müller and Bruce Abernethy. Expert anticipatory skill in striking sports: A review and a model. *Research quarterly for exercise and sport*, 83(2):175–187, 2012.

[344] Sean Müller, Bruce Abernethy, and Damian Farrow. How do world-class cricket batsmen anticipate a bowler's intention? *Quarterly journal of experimental psychology*, 59(12):2162–2186, 2006.

[345] Kielan Yarrow, Peter Brown, and John W Krakauer. Inside the brain of an elite athlete: the neural processes that support high achievement in sports. *Nature Reviews Neuroscience*, 10(8):585–596, 2009.

[346] Martín Abadi, Ashish Agarwal, Paul Barham, et al. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. URL `https://www.tensorflow.org/`. Software available from tensorflow.org.

[347] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[348] Samuel Stewart. What is the best fps for gaming? [2022 guide] - gamingscan, 2022. URL `https://www.gamingscan.com/best-fps-gaming/`. Last accessed on August 4$^{\text{th}}$, 2022.

[349] Human generator v2. `https://www.humgen3d.com/` or `https://blendermarket.com/products/humgen3d`. [Tool for character modeling in blender; Online; Accessed April 2021].

[350] Wrap3d. `https://www.russian3dscanner.com/`. [Tool for working with textures; Accessed December 2021].

[351] Douglas Bates et al. Fitting linear mixed models in r. *R news*, 5(1):27–30, 2005.

[352] Douglas Bates, Martin Mächler, Ben Bolker, and Steve Walker. Fitting linear mixed-effects models using lme4. *arXiv preprint arXiv:1406.5823*, 2014.

[353] Jon Sprouse. A validation of amazon mechanical turk for the collection of acceptability judgments in linguistic theory. *Behavior research methods*, 43(1): 155–167, 2011.

[354] Atsushi Mori and Yuta Itoh. Dronecamo: Modifying human-drone comfort via augmented reality. In *2019 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct)*, pages 167–168. IEEE, 2019.

[355] Yuta Itoh, Tobias Langlotz, Jonathan Sutton, and Alexander Plopski. Towards indistinguishable augmented reality: A survey on optical see-through head-mounted displays. *ACM Computing Surveys (CSUR)*, 54(6):1–36, 2021.

A1 Appendix

A1.1 Supplementary figures for Throwing

Here, we provide visualizations of the throws performed in Exp. T1 and Exp. T2. As the motion is both spatially high-dimensional and temporal, we provide it in a transformed space for readability, as in Figure 4.13. To transform the data, the global position is locally projected through a procedure described in detail in Section 3.2.1. We further average this locally projected space to get an average motion performed by each actor. In Figures A1.1 and A1.2, only Overarm throws are presented. Each row represents a participant (1-18), and each column represents a different distance (Near-Medium-Far) depending on the target's spawn position. Figures A1.3 and A1.4 visualize only Underarm throws. If a certain row is missing or an entry is empty, it means the participant has not performed any such throws (Overarm or Underarm) in that region.

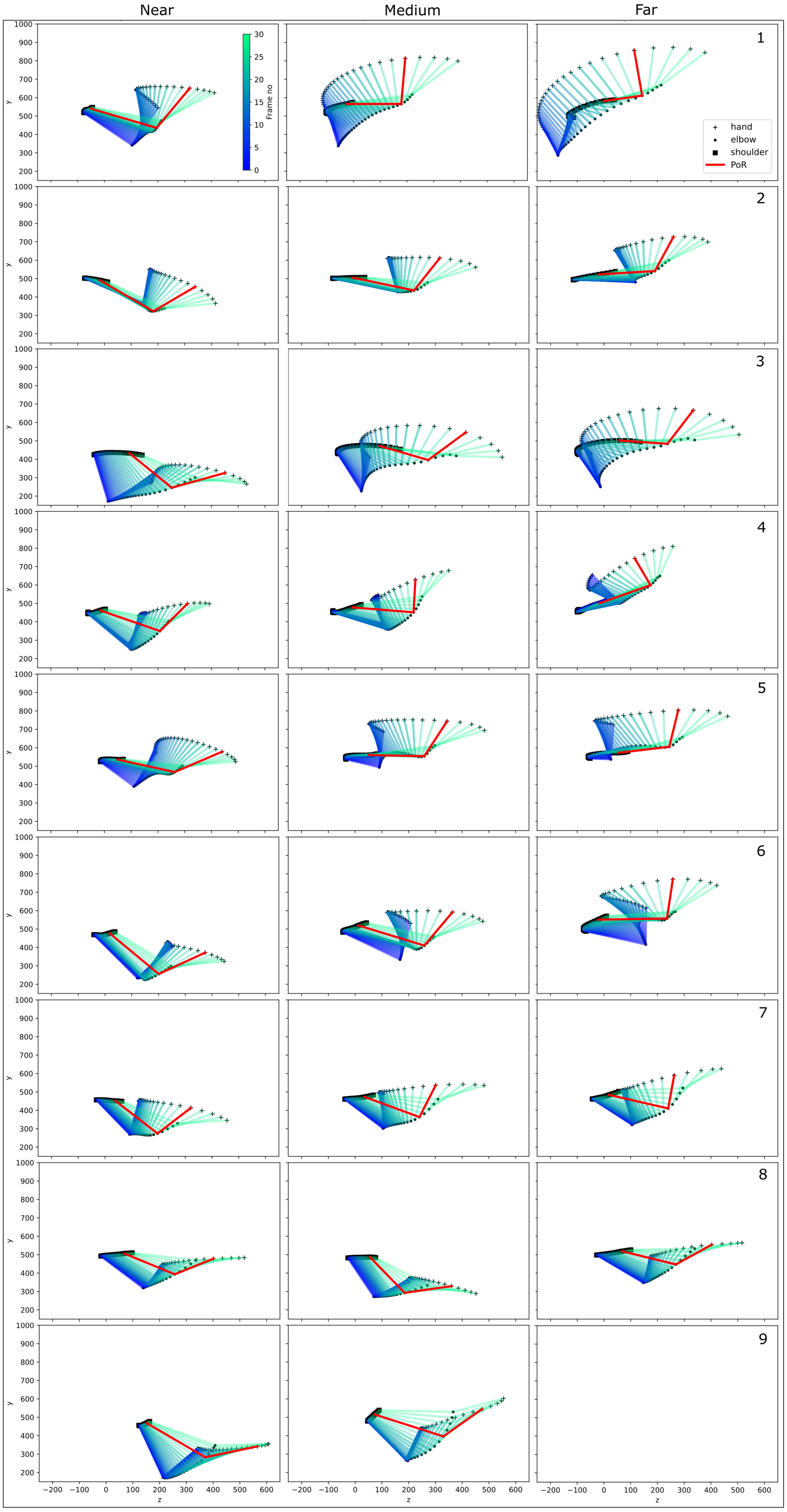


Figure A1.1: Averaged projected local positions of the three arm joints for Overarm throws in side view (y and z axes) for participants 1-10 of Exp. T1. Each row represents one subject, e.g., '1' for subject 1. Each column represents a different distance. The red line represents the ground truth PoR (t=25).

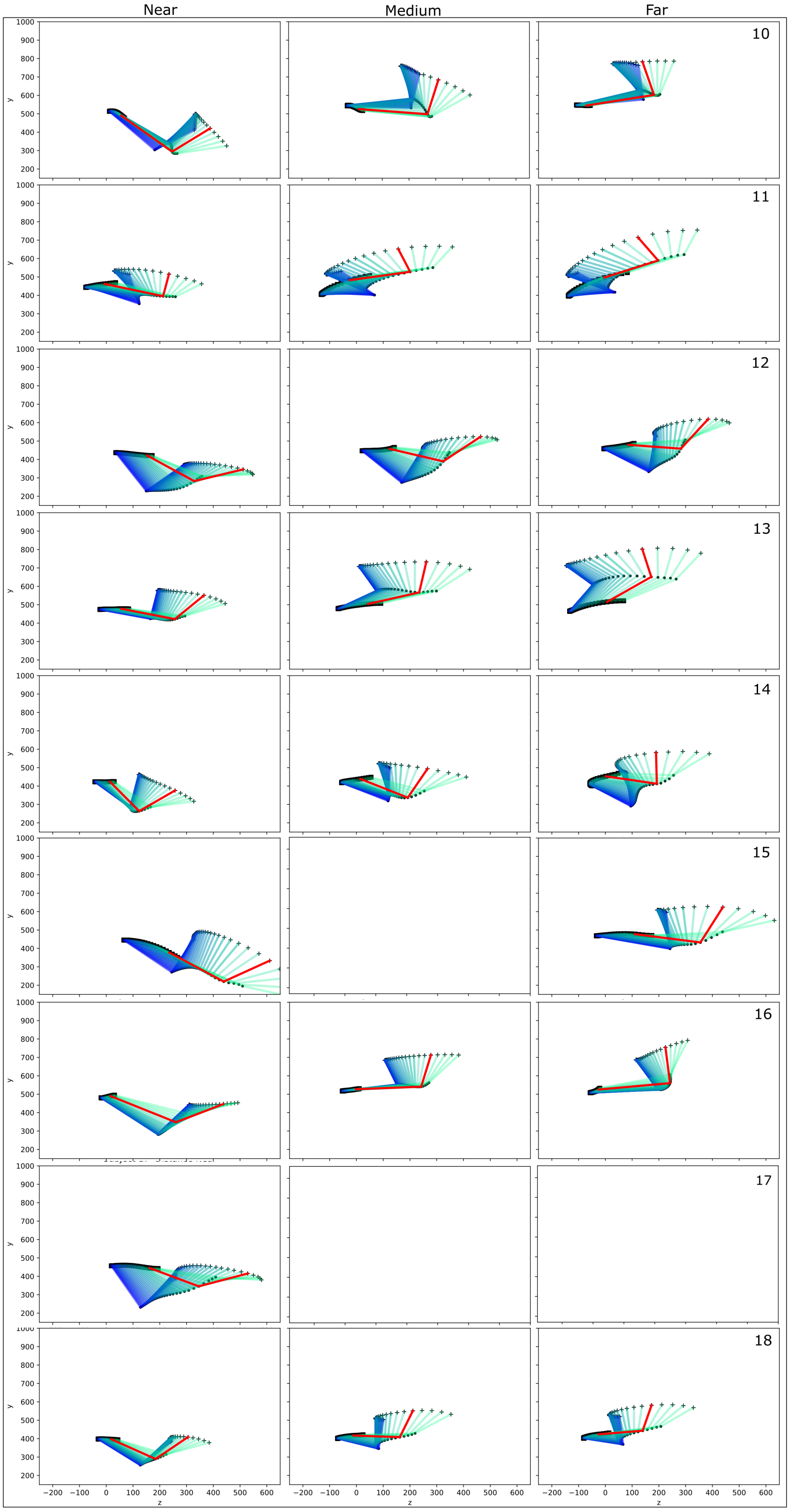


Figure A1.2: Averaged projected local positions of the three arm joints for Overarm throws in side view (y and z axes) for participants 11-18 of Exp. T1.

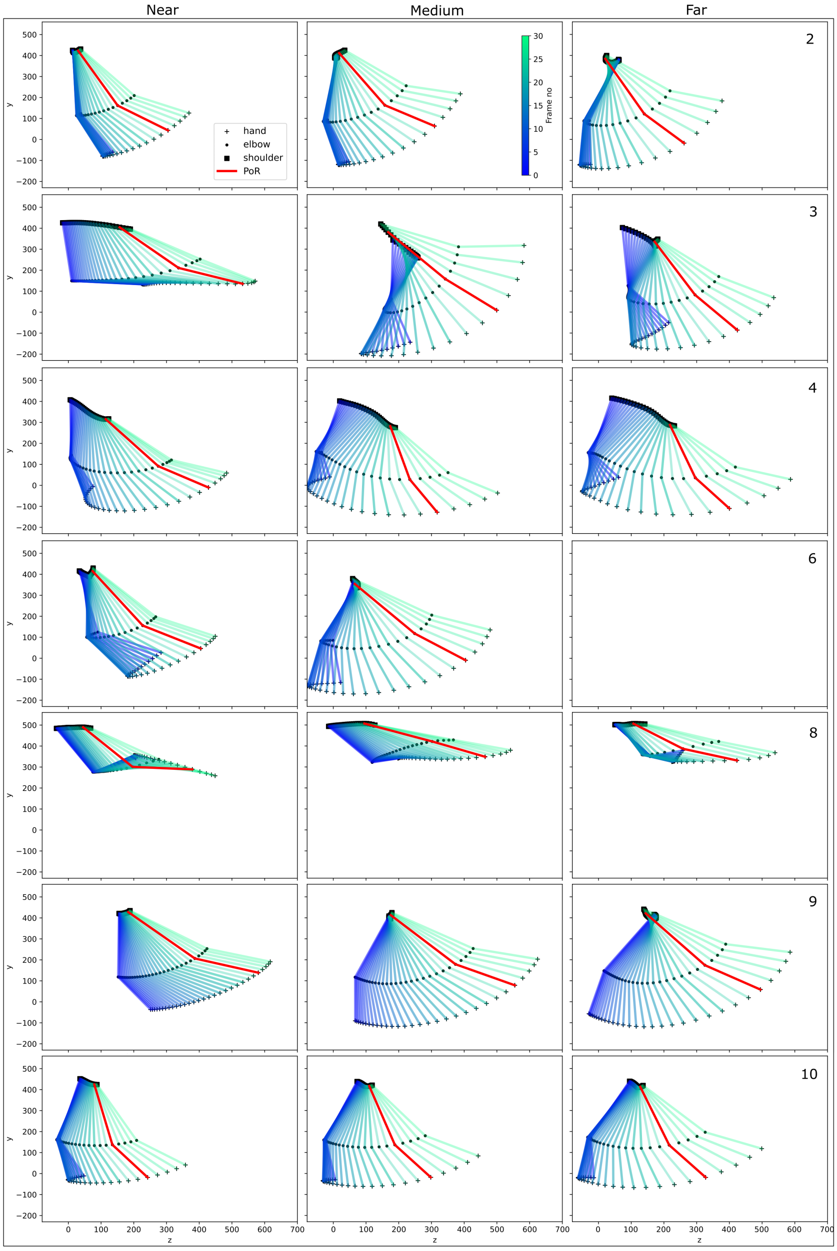


Figure A1.3: Averaged projected local positions of the three arm joints for Underarm throws in side view (y and z axes) for participants 1-10 of Exp. T1.

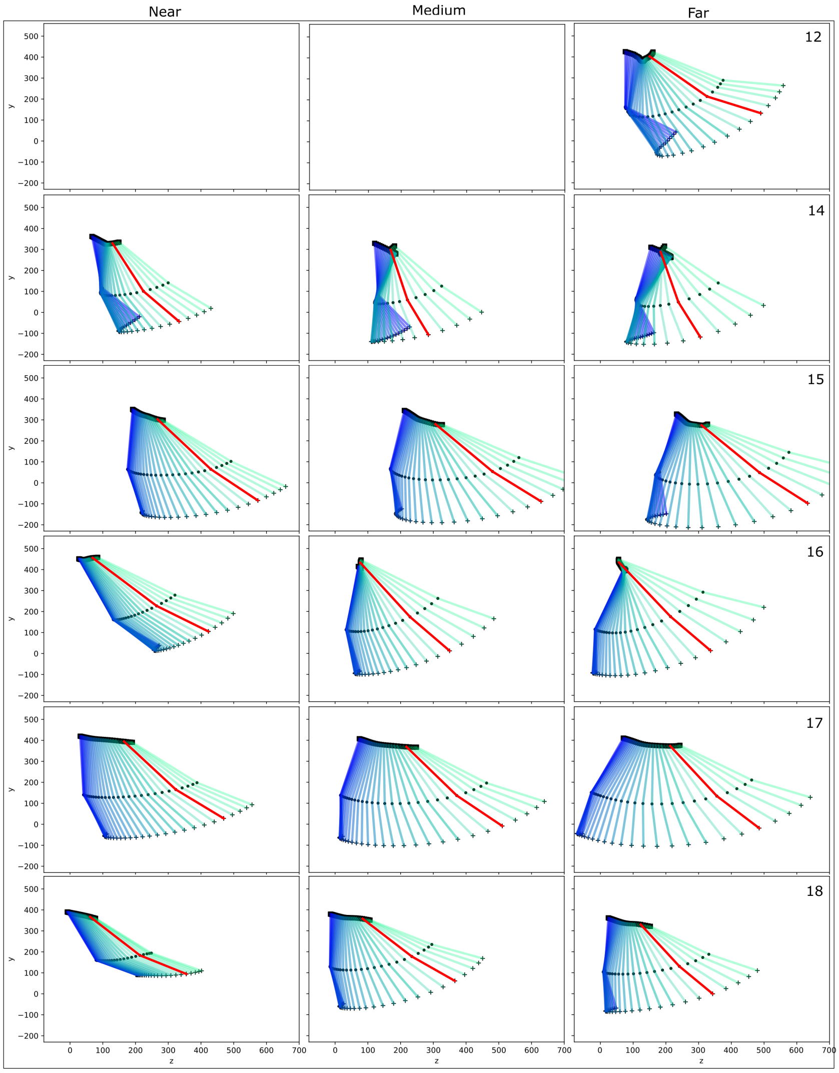


Figure A1.4: Averaged projected local positions of the three arm joints for Underarm throws in side view (y and z axes) for participants 12-18 of Exp. T1.

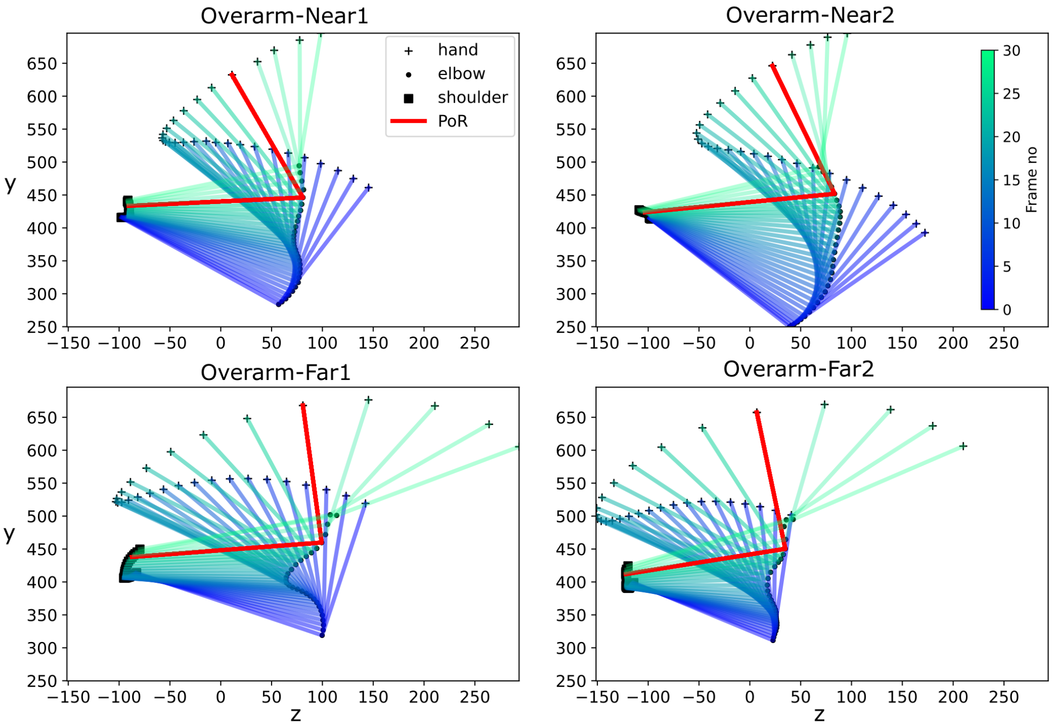


Figure A1.5: Averaged projected local positions of the three arm joints for Overarm throws in Exp. T2, displayed in the side view (y and z axes).

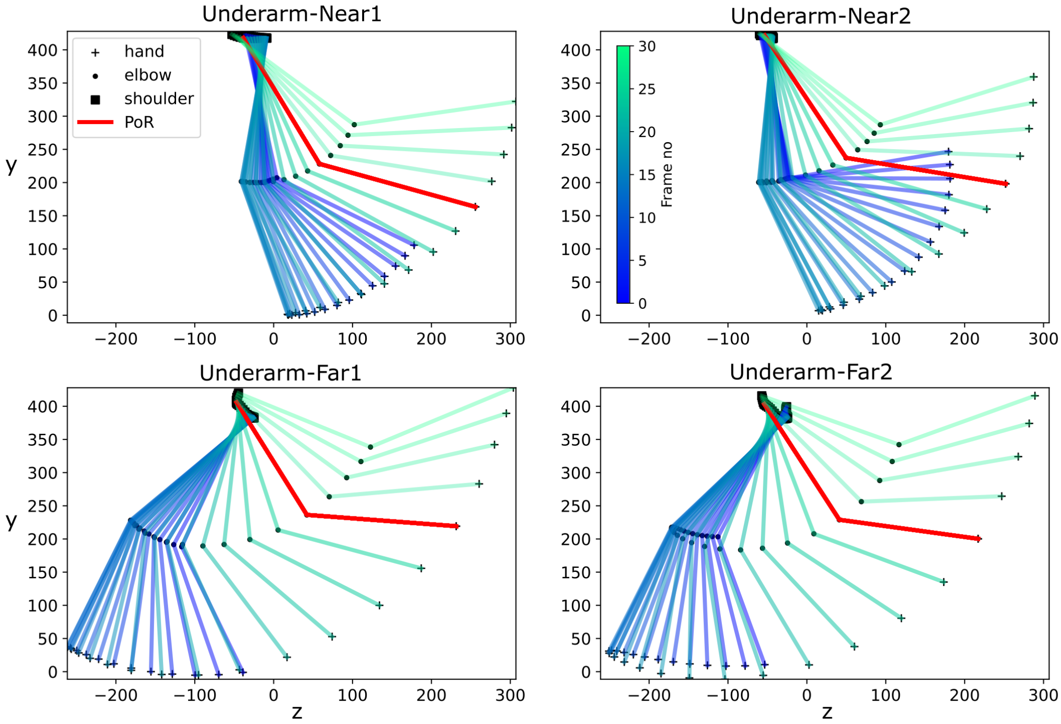


Figure A1.6: Averaged projected local positions of the three arm joints for Underarm throws in Exp. T2, displayed in the side view (y and z axes).

A1.2 Supplementary data for Lifting

Table A1.1: Significance of variation in effort and weight estimates from models for each actor for Experiment 1 (Baseline) using Analysis of Deviance: Type II Wald chisquare tests *(Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ')*

	EFFORT			WEIGHT		
	χ^2	*dof*	$p(>\chi^2)$	χ^2	*dof*	$p(>\chi^2)$
AA1	338.05	4	< 2.2e-16	257.87	4	< 2.2e-16
AA2	295.13	4	< 2.2e-16	229.91	4	< 2.2e-16
SA1	740.88	4	< 2.2e-16	550.71	4	< 2.2e-16
SA2	685.03	4	< 2.2e-16	625.54	4	< 2.2e-16

Pairwise comparisons were performed for each level of effort and weight. In the ideal case, each would be significantly separated from its predecessor. Instead of just looking at the significance cutoff of $\alpha = .05$, which can overly simplify relationships, Table A1.2 shows the distribution of p-values for the ten pairwise comparisons for each model. In every case, the largest effort/weight are clearly separated from all others (p<.001). However, for the average strength actors, both effort and weight are generally not significantly different at the lower levels, given the sample size. This reflects the more moderate slope in this part of the response curve and a more moderate slope for the average lifters than the strong.

Table A1.2: Tukey-adjusted P-values for pairwise comparisons of the different stimuli levels. There are ten comparisons for each actor. Pairs with significance lower than .001 are indicated and the remainder are marked "rest". Grey cells are above the alpha = 0.05 test line (not significant).

p-value	[1, .5)	[.5, .1)	[.1, 0.05)	[.05, .01)	[.01, .001)	< .001
EFFORT						
AA1	0-25 50-75	25-50	0-50	25-75	0-75	rest
AA2	0-25 25-50		50-75	0-50		rest
SA1				25-50	0-25	rest
SA2				0-25 25-50	50-75	rest
WEIGHT						
AA1	0-6.75 13.5-20.25	0-13.5 6.75-13.5 6.75-20.25	0-22.5			rest
AA2	0-8.75 8.75-17.5	0-17.5			17.5-26.25	rest
SA1			15-30	0-15		rest
SA2				0-15	15-30	rest

Table A1.3: Naturalness ratings for Experiment 2 using Analysis of Deviance: Type II Wald chisquare tests *(Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ')*

BODY SHAPE			
	χ^2	*dof*	$p(> \chi^2)$
Effort	1.39	3	0.7086
Body	88.09	3	< 2.2e-16 ***
Motion	38.54	3	2.168e-08 ***
Effort:Body	58.01	9	3.230e-09 ***
Effort:Motion	106.42	9	< 2.2e-16 ***
Body:Motion	75.42	9	1.304e-12 ***
Effort:Body:Motion	25.23	27	0.5615

Table A1.4: Naturalness ratings for Experiment 3 using Analysis of Deviance: Type II Wald chisquare tests *(Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ')*

DUMBBELL SIZE			
	χ^2	*dof*	$p(>\chi^2)$
Effort	20.08	4	< 0.0005 ***
Dumbbell	27.32	3	5.033e-06 ***
Actor	13.53	3	< 0.005 **
Effort:Dumbbell	78.50	12	7.975e-12 ***
Effort:Actor	21.97	12	< 0.05 *
Dumbbell:Actor	13.47	9	0.142
Effort:Dumbbell:Actor	38.12	36	0.373

Table A1.5: Discrimination scores (i.e., correct responses) for Experiment 5

ZERO LIFT DETECTION			
	χ^2	*dof*	$p(>\chi^2)$
Deformation	0.08	1	0.78
Effort	21.73	3	7.4e-05 ***
Actor	19.26	3	< 0.0005 ***
Deformation:Effort	120.30	3	< 2.2e-16 ***
Deformation:Actor	8.00	3	< 0.05 .
Effort:Actor	51.13	9	6.6e-08 ***
Deformation:Effort:Actor	25.59	9	0.37

Table A1.6: FULL-NONE Contrasts (dof are Inf in all cases)

ZERO LIFT DETECTION					
Actor	Effort	Estimate	SE	z-ratio	*p*
AA1	25	1.561	0.408	3.826	0.0001
	50	0.774	0.409	1.895	0.0581
	75	-0.556	0.361	-1.540	0.1235
	100	-1.212	0.384	-3.158	0.0016
AA2	25	1.945	0.455	4.275	<.0001
	50	0.827	0.384	2.154	0.0313
	75	-0.430	0.377	-1.142	0.2536
	100	-1.547	0.431	-3.591	0.0003
SA1	25	1.619	0.387	4.182	<.0001
	50	0.470	0.365	1.288	0.1976
	75	-2.366	0.525	-4.510	<.0001
	100	-0.485	0.591	-0.822	0.4113
SA2	25	2.552	0.530	4.814	<.0001
	50	-1.202	0.414	-2.906	0.0037
	75	-1.659	0.501	-3.308	0.0009
	100	-1.189	0.705	-1.688	0.0915