Mathias Parger

Inverse Kinematics for Virtual Reality

Master Thesis

Graz University of Technology

Institute of Computer Graphics and Vision
Head: Univ.-Prof. Dipl.-Ing. Dr.techn. Dieter Schmalstieg

Supervisor: Ass.Prof. Dipl.-Ing. Dr.techn. Markus Steinberger
Co-Supervisor: Dipl.-Ing. BSc. Jörg Müller

Graz, July 2018

This document is set in Palatino, compiled with pdfLaTeX2e and `Biber`.

The LaTeX template from Karl Voit is based on KOMA script and can be found online: https://github.com/novoid/LaTeX-KOMA-template

Statutory Declaration

I declare that I have authored this thesis independently, that I have not used other than the declared sources/resources, and that I have explicitly marked all material which has been quoted either literally or by content from the used sources.

Graz, ____________________________ ____________________________________

Date Signature

Eidesstattliche Erklärung[1]

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt, und die den benutzten Quellen wörtlich und inhaltlich entnommenen Stellen als solche kenntlich gemacht habe.

Graz, am ____________________________ ____________________________________

Datum Unterschrift

[1]Beschluss der Curricula-Kommission für Bachelor-, Master- und Diplomstudien vom 10.11.2008; Genehmigung des Senates am 1.12.2008

Abstract

The introduction of motion controllers for consumer Virtual Reality (VR) devices shifted the focus from classical gamepad-based applications to intuitive controls with one's own hands. There are many examples of research in bringing further body parts into the virtual world which use motion capture, Kinect or Inverse Kinematics (IK) for different applications. However, to our knowledge, there does not yet exist a study on the ability to display additional body parts in VR with current consumer devices without additional hardware. In order to find that out, we create a study that uses arms in addition to hands. We compare motion capture with an IK system in different settings to see if IK can be used for animating arms, how it performs to a very accurate pose tracking system, and if having arms improves the user experience. The study shows that IK can be used for games with strong usage of arms and that it is at least on par with and in most cases better than motion capture. The participants strongly prefer having arms over having only hands when they are asked which methods feels most like one's own body. Yet, it also shows that having arms does not necessarily improve the feeling of embodiment when they are not needed for gameplay. Arms animated with IK always achieved at least as good results as having only hands and in some cases the results are clearly better. This indicates that the accuracy of the elbow and shoulder estimation does not have to be perfect for an improved experience as long as the delay and accuracy of the hand stay the same. With that knowledge, applications that use arms for gameplay can be created with current consumer hardware and without additional devices. Some examples could be a sword game where the player uses its arm to hold the shield, a boxing game or some gadgets which are placed on the arms and wrists.

Kurzfassung

Die Markteinführung von Bewegungs-Controller für virtuelle Realität (VR) hat den Fokus in Anwendungen von klassischer, Gamepad-Steuerung auf intuitive Steuerung mittels Hände verlagert. Es wurde bereits viel Forschung betrieben, um weitere Körperteile in VR darzustellen, etwa durch die Nutzung von Motion Capture, Microsoft Kinect oder Inverse Kinematik (IK). Dennoch existiert unseres Wissens nach noch keine Studie über die Möglichkeit, weitere Körperteile mit aktueller Konsumentenhardware und ohne Verwendung zusätzlicher Hardware in VR darzustellen. Um das herauszufinden, erstellen wir eine Studie in welcher Arme zusätzlich zu den Händen verwendet werden. Wir vergleichen ein Motion Capture-System mit einem IK-System in unterschiedlichen Anwendungsfällen, um herauszufinden, ob IK für die Animation von Armen benutzt werden kann, wie gut es im Vergleich zu Motion Capture funktioniert und ob die Darstellung von Armen das Erlebnis verbessert. Die Studie hat ergeben, dass IK für Spiele mit starker Arm-Nutzung verwendet werden kann und dass es von den Studienteilnehmern mindestens gleich gut und in den meisten Fällen besser als Motion Capture bewertet wurde. Die Nutzer bevorzugen Arme gegenüber nur Händen, wenn sie danach gefragt werden, welche Methode das stärkste Gefühl von Verkörperung erreicht. Dennoch hat die Studie gezeigt, dass Arme zu haben nicht notwendigerweise das Gefühl von Verkörperung verbessert solange sie nicht zur Interaktion benötigt werden. IK-animierte Arme sind jedoch in allen Fällen mindestens gleich gut und in manchen Fällen sogar klar besser als nur Hände. Dadurch lässt sich ableiten, dass Genauigkeit von Schulter und Ellbogen nicht perfekt sein müssen, um das Ergebnis zu verbessern, solange die Verzögerung und Genauigkeit der Hand gleichbleibt. Mit dieser Erkenntnis lassen sich Anwendungen erstellen, die Arme zur Interaktion verwenden und mit aktueller Konsumentenhardware und ohne zusätzlicher Hardware funktionieren. Anwendungsfälle wären zum Beispiel Schwertkampf-Spiele, in welchen der Spieler mit seinem Arm einen Schild hält, Boxen-Spiele oder Anwendungen mit Gadgets, die an Arm und Handgelenk befestigt sind.

Contents

1. Introduction

Recent improvements in VR hardware led to a revival of the research in immersion and presence in VR. With the introduction of motion controllers for HTC Vive and Oculus Rift, interaction in VR applications changed completely from conventional controller usage to intuitive interaction with the virtual environment using the player's hands. Combined with room-scale tracking, which allows for movement within a few square-meters, the feeling of presence is considerably stronger than in applications running on a 2D monitor. Yet, we are still far away from one of the main goals in VR: the achievement of an experience like in the movie *The Matrix*. While there is a long list of potential improvements for VR like full-body tracking, cordless head mounted displays (HMDs) and haptic feedback, etc., most of these improvements require additional hardware.

In this thesis, we are trying to improve the feeling of presence with the standard consumer equipment consisting of an HMD, 2 motion controllers and tracking hardware. Studies have shown that one of the key components for the feeling of presence is embodiment, i.e. the feeling of the virtual body being your own (Schultze, 2010). Due to missing tracking information, most VR applications only show a representation of the user's hands in the virtual environment. Arms, shoulders and the rest of the body remain invisible and are not used for interaction. While the representation of the user's legs is very difficult and inaccurate without additional tracking devices, arms and shoulders can be estimated quite well with existing tracking information using an IK solver.

IK is the process of calculating joint rotations which connect the start anchor with the end anchor using multiple joints and bones. Constraints can be used to avoid unrealistic angles, e.g. clamping elbow angles avoids turning the lower arm the other way around. In our case, the start point is the HMD position and the two end points are the hands. With this information, we must estimate plausible positions and rotations for the shoulders, upper arms, and lower arms. The main problem of IK solvers for estimating the arm pose is that due to missing information about the user's pose, it is impossible to create correct poses for the avatar. Elbows and shoulders can be moved freely without moving or rotating the hands or head which are the only body parts that are tracked by VR devices. Thus, the goal of the IK solver is not to create a copy of the user's pose, but to create poses that have a low average error for all users in the most frequently used poses. Those poses vary strongly per application. One could optimize an IK system only for one use case with very predictable poses like for example archery, darts, boxing or driving cars. Other applications like yoga, simulations or virtual meetings are more difficult to estimate and are likely to produce higher pose errors. We tried two different IK systems in Unity: SAFullBodyIK (StereoArts, 2018), a free and open-source solution, and Final IK (RootMotion, 2018), a paid asset from the Unity Asset Store. Unfortunately, we were not convinced by neither of them and thus decided to create our own IK solver which is optimized for VR arm movement.

IK is neither a new invention, nor are we the first who use it for arm movement in VR games. Still, to our knowledge, there does not yet exist a study on how it changes the user experience and how it performs compared to the industry standard for human motion tracking, an optical motion capture system. For this study, we create an IK solver which is optimized for arm movement in VR and compare it to an industrial motion capture system in different test settings. From this experiment we hope to find answers to the question if virtual arms improve the feeling of presence, if they can be used for interaction in VR application and if IK solvers are precise enough when only the tracking data of controllers and HMD is available.

2. Related Work

Embodiment is one of the most important attributes for increasing the feeling of presence. It can be defined as "the degree to which an avatar affords the user equal or greater functionality expected of our natural bodies" (Costa, Kim, and Biocca, 2013). In other words, it describes the feeling of an virtual avatar being part of your own body. A well-known example for embodiment is the rubber hand illusion (Botvinick and Cohen, 1998). The rubber hand illusion is the illusion of a rubber hand being your own hand. In this experiment, the real hands are invisible to the participants. Instead, rubber hands are placed in front of the participants. When both hands get the same stimulation at the same time, the participants see the stimulation being applied to the rubber hands and feel it with their real hands. After a few seconds or minutes, the participants start to embody the rubber hands. To demonstrate this effect, the study conductors often use hammers with which they suddenly hit the rubber hands. When the participants see a dangerous attempt on their rubber hands, they pull away their real hands, because they think that the rubber hands belong to their real body. This experiment shows how fast the human mind can forget that the virtual avatar is not a part of one's own body although it is aware of it. Embodiment consists of three main components: sensory input, physical manipulation and self-identity. The rubber hand illusion is based on sensory input and does not provide the ability of physical manipulation. In this thesis, however, we concentrate only on physical manipulation which is expanded by adding arms to the avatar which are be controlled by natural movement of the user.

Most consumer VR applications with motion controllers only show hands in first-person perspective. In multiplayer games or when looking into a mirror in the virtual world, full bodies are displayed using an IK pose estimation for animation of avatars viewed from third person perspective.

These pose estimations work well for viewing other players, but are often too inaccurate for seeing one's own avatar in first person perspective and even less when using them for interaction. Steed et al. (2016) showed in a study that incorrect pose controls can even decrease the feeling of presence and embodiment compared to a static avatar. They created an experiment which used a mobile HMD without body tracking or tracked controllers. Yet, they instructed the player to tip their leg with their hands along the music and used an estimated, predefined animation for the avatar. When the users did not tip their legs at the same time as the avatar did, their feeling of embodiment was decreased strongly and they felt less present than when they and their avatar did not move at all. We circumvent this issue by only displaying body parts of which we know the pose or of which we can at least create a good estimation. Chest, hips and legs are not visible to the user and therefore do not create unrealistic and disturbing movement.

Despite the importance of embodiment for presence and the fact that VR is capable of creating experiences with much higher immersion and presence than monitor-based games, little research on embodiment in VR was done in the last years. Tiator et al. (2018) developed a VR trampoline jumping game which used IK for full-body pose estimation. The IK system used five motion capture tracked rigid bodies placed on head, hands and feet for pose estimation. The study participants could choose between an avatar with predefined animations or an avatar animated using IK. While Tiator et al. (2018) assume that the IK animated avatar provides more safety while jumping on the trampoline, they did not evaluate that issue. They also received negative feedback about the avatar like "The virtual character doesn't quite correspond with the real person.". Unfortunately, they did not mention the IK system they used or the pose error it produced. Nevertheless, we expect that our avatar achieves a stronger feeling of embodiment since we only have to estimate the arms instead of the whole body. Also, trampoline jumping leads to many unnatural poses for human bodies and therefore increases the difficulty of correct pose predictions.

Many more applications with virtual avatars for VR were created using different animation systems. Chang et al. (2017) created a telepresence system which uses full-body IK with head and hands tracking information only. A real time full-body motion capture system was used by Spanlang, Normand, Giannopoulos, et al. (2010) and Spanlang, Normand, Borland, et al. (2014) combined with a full-body haptic feedback system. Another alternative is to use Microsoft Kinect for body tracking (Lee and Lim, 2015). Unfortunately, none of these projects was evaluated in terms of embodiment, presence or pose error. Thus, we cannot use their work to improve our study or to compare the pose error when using our IK solver or the motion capture system. At the same time this shows the importance of our study for exploring how having arms changes the user experience and if arms can already be used in games with current consumer VR devices without additional hardware.

3. Avatar animation

The first goal of this thesis is to examine the influence of having a representation of the user's arms in VR and being able to use them for interaction. The second objective is to create a meaningful comparison between the usage of IK versus motion capture for arm and shoulder movement. On this account, we create an IK solver for arms and shoulders [3.1], a real-time motion capture interface [3.2] and an application with different tasks to execute [4]. While the motion capture system is aware of the VR tracking devices and uses their tracking data for better results, the IK system is completely unaware of the motion capture data. This independence is important since the IK system must be able to run on its own when used at home.

The hardware and software used for implementation of this project consist of:

- VR headset: Oculus Rift CV1, HTC Vive
- Tracked controllers: Oculus Touch, HTC Vive
- Game engine: Unity 2017.3.1f1
- Motion Capture System: Optitrack 1.7.2

3.1. Inverse Kinematics

Current consumer VR devices track the position and orientation of the HMD and the two motion controllers. Using this data and knowledge about the human body, the IK solver has to estimate the user's hand, lower arm and upper arm position and rotation. While the hand pose can be estimated easily by a fixed offset rotation and position depending on the used controller, the upper arm positioning needs a complex shoulder estimation. Even more difficult proves to be the estimation of the elbow position which influences the lower arm position and rotation, a body part which is visible to the user most of the time in VR applications. In this section, we first describe the shoulder IK solver, then the basic arm IK solver and finally our solution for elbow position estimation. All calculations are done using scaled vectors for distances. Thus, the Euclidean distances between hand and shoulder as well as the players height are divided by the users arm length and body size. The scaling assures that the IK solver creates equal results for users of different size.

3.1.1. Shoulder Pose Estimation

The first step of the IK solving iteration is the shoulder estimation which is necessary for providing good anchor points for the upper arms. The shoulder estimation consists of three parts: shoulder center positioning, shoulder center rotation and the distinct rotation of left and right shoulder depending on the shoulder-to-hand distance. In this section, the *Tait-Bryan angles* are used for explicitness of the axis directions using yaw, pitch and roll around the up, right and forward axes.

Shoulder center position estimation

The center position of the shoulder only depends on the HMD position and orientation. First, we add a local offset position to the HMD position to compensate the distance between HMD, which sits in front of the user's eyes and shoulders. Therefore, distances between HMD and the head pivot as well as from the head pivot to the neck pivot need to be corrected. As these distances differ for each user and require lots of calibration, this step is simplified by using a single vector from the HMD to the shoulders with fixed distance values for all users. This offset must be calculated in local coordinates of the HMD to ensure that the shoulder position does not change when the user is looking left and right or up and down.

Shoulder center rotation estimation

Shoulder center rotation estimation uses the pose of the HMD for pitch estimation and the position of the motion controllers relative to the HMD for yaw calculation. Due to the complexity of roll estimation and the fact that it is by far not as important as yaw and pitch, the shoulder's roll is assumed to be constant.

For pitch estimation, the ratio between the user's height y_0 and the current HMD distance to the ground y_h is combined with the HMD's pitch r_h. The smaller the distance to the ground and the more the HMD is looking down, the more the user's shoulder is looking down. If only one of those values is high, i.e. when the user is standing upright and looking on the ground or when the user is kneeling, but the chest is upright, the influence on the shoulder's pitch α is reduced. The parameters a and b are used as weights. Figure 3.1 shows some example poses of the user and the corresponding shoulder pitch estimation.

$$r = \frac{y_h}{y_0} \tag{3.1}$$

$$\alpha = r \times (a + b \times r_h) \tag{3.2}$$

Yaw calculation is the most difficult part of the shoulder estimator and it has a large impact on the final accuracy of the IK solver. The yaw estimation should be as robust as possible against head movements when the rest of the body is moving as well as when only the hands and arms are moved. The naive approach would be to use the HMD yaw for the shoulder, but large problems occur when the head is turned to the side. Humans are able to turn their head to the left and right approximately 90° in each direction.

Figure 3.1.: Shoulder position and rotation in different head poses. Note that the shoulder rotation does not change from the left first to the second head pose. However, the shoulder does not stay beneath the world center of the head, but in the local center.

Thus, while this approach would work well for forward-facing situations, the error would be very large when looking to the side. Because of that, we are not using the HMD rotation for yaw estimation. Instead, we derive the yaw from the sum of the normalized directions from the HMD to the motion controllers. This way, the shoulders remain stable while rotating the head at the expense of comparably small yaw errors when moving the hands. We use the yaw of the HMD only to prevent the shoulders from looking backwards when both hands are behind the player and to clamp the yaw difference between head and shoulders.

$$Y = \frac{X_l - X_h}{||X_l - X_h||} + \frac{X_r - X_h}{||X_r - X_h||} \tag{3.3}$$

$$\gamma = atan_2\left(\frac{Y_1}{||Y||}, \frac{Y_0}{||Y||}\right) \tag{3.4}$$

where

- Y is the 2D mean direction vector
- X_l is the 2D position of the left controller in top view
- X_h is the 2D position of the HMD in top view
- X_r is the 2D position of the right controller in top view
- γ is the resulting yaw

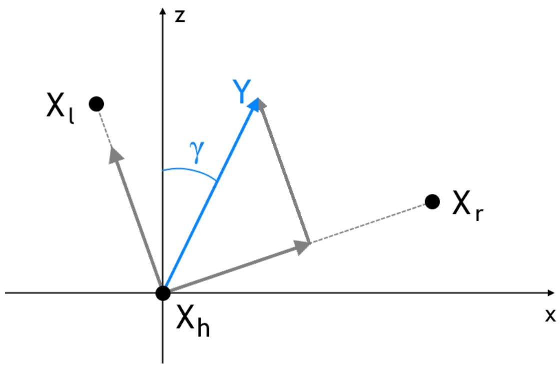


Figure 3.2.: The shoulder center yaw γ is calculated using the sum of the normalized directions from head to hands.

$atan_2$ refers to the angle corrected arctan which ensures that the angle lies within the correct Cartesian quadrant.

Finally, γ is clamped to be within a $\pm 90°$ range to the yaw of the HMD. The resulting rotation is applied to the local shoulder offset position to ensure that the shoulder is placed behind the head instead of beneath it when the shoulder is facing downwards. This process is illustrated in figure 3.1.

Distinct shoulder rotation estimation

The last step of the shoulder pose estimation is the distinct shoulder rotation estimation. This step allows the left and right shoulder to move independently within a small range and by that, enables the hands to reach distances which would otherwise be clamped by the arm length. While the real distinct shoulder position is very difficult to calculate, we assume that the shoulder follows the hand position.

In our simplified model, the left and right shoulders are attached to the shoulder center and therefore, their rotation pivot also lies in the center. If the distance between shoulder and hand exceeds a threshold, it is rotated towards a direction which reduces the shoulder-hand distance. The higher the distance, the higher the rotation. This enables the upper arm anchor to move forwards, backwards and upwards. Equation 3.6 shows the yaw

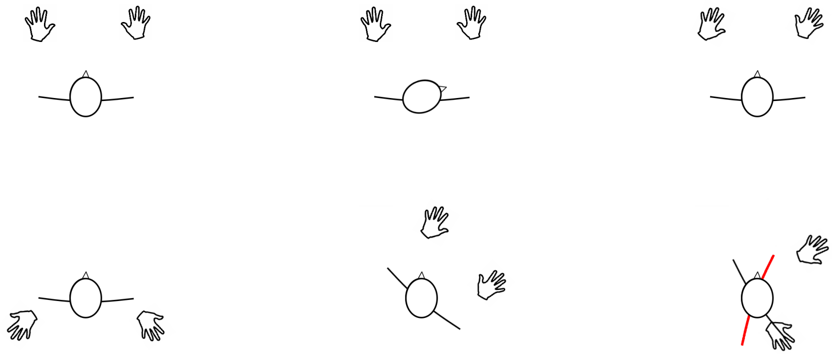


Figure 3.3.: Different hand and head poses and their corresponding shoulder orientation. The red line in the bottom right picture illustrates the clamping process relative to the head orientation.

calculation in case the hand is in front of the shoulder. If the hand is behind the shoulder, the resulting angle is negated.

$$r = (X_h - X_d)^T \cdot \frac{Y}{||Y||} \frac{1}{l_a} \tag{3.5}$$

$$\gamma_d = max(0, min((r - r_o) \times c, \gamma_{d,max})) \tag{3.6}$$

where

- r is the shoulder-hand distance ratio
- X_h is the 3D position of the hand
- X_d is the 3D position of the distinct shoulder
- Y is the 3D forward direction vector of the shoulder center which can be derived from Y in equation 3.3
- l_a is the arm length
- γ_d is the resulting distinct shoulder yaw
- r_0 is the distance ratio threshold
- c is a scaling constant

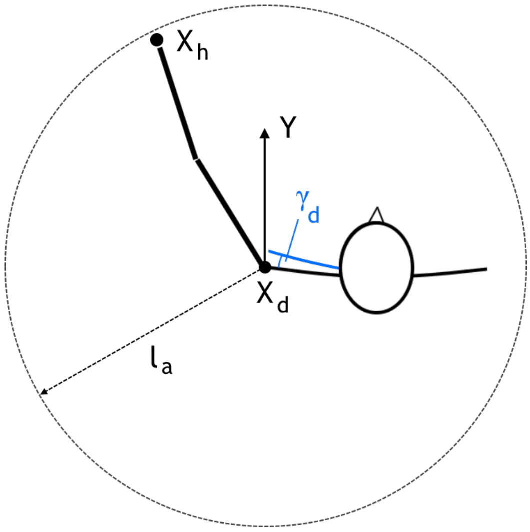


Figure 3.4.: The distinct shoulder yaw γ_d is calculated using the ratio between the shoulder to hand distance and the arm length.

3.1.2. Arm IK part I: Trigonometry

At this point, the shoulders are positioned and rotated and we can start to connect the upper arms to the left and right shoulders and create a good pose estimation. The arm IK solver connects the shoulder with the hand target position and finds a plausible elbow position. The step by step progress is displayed in figure 3.8.

In part I of the arm IK solver, the inner angle of the elbow and the shoulder rotation is determined. This part is the easier part as it can be solved by simple trigonometry. The purpose of this step is to find angles for the upper arm and the inner elbow rotation which can be used in Forward Kinematics (FK) so that the hand reaches its target position. The positions and angles are illustrated in figure 3.5

The upper arm pivot X_u, the target hand position X_t and the lower and upper arm lengths l_l and l_u are used to calculated the inner angle α of the elbow by using the cosine rule.

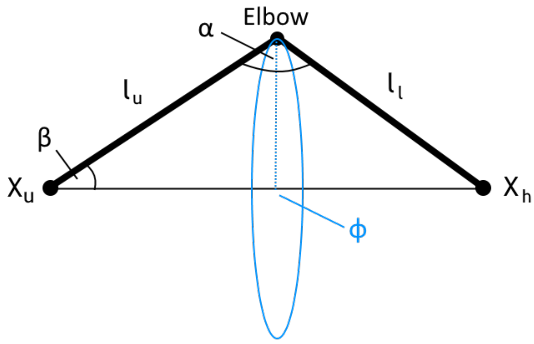


Figure 3.5.: Simplified illustration of angles which have to be determined with the arm IK solver. The blue line illustrates all possible elbow positions in the third dimension.

$$\alpha = \arccos\left(\frac{l_u^2 + l_l^2 - ||X_t - X_u||^2}{2 l_u l_l}\right) \tag{3.7}$$

After applying α to the elbow, the distance between shoulder and hand pivot is now equal to the target distance. In the next step, the upper arm will be rotated towards the target direction, which can be accomplished by using the method *Transform.LookAt()* provided by Unity. This method calculates the rotation between the forward vector and the vector from source to target. The delta rotation is then multiplied with the source rotation to make the forward vector point towards the target. Now the upper arm is looking towards the target hand position, but the hand is still at a wrong place. In the last step, the upper arm is moved away from the target position along the lower arm's longitudinal axis. Therefore, the angle β between the target and the hand pivot X_h is calculated by using the cosine rule.

$$\beta = \arccos\left(\frac{||X_t - X_u||^2 + ||X_h - X_u||^2 - ||X_t - X_h||}{2||X_h - X_u|| \times ||X_t - X_u||}\right) \tag{3.8}$$

The shoulder is now rotated using the angle β around the elbow's up axis.

3.1.3. Arm IK part II: Elbow Positioning

After applying the IK steps as described in section 3.1.1 and 3.1.2, shoulder and hand are now on their target positions. However, the elbow is at a random position around the shoulder-hand axis ω depending on where it has been before the current IK iteration. In this part of the arm IK solver, the goal is to place the elbow on a plausible angle around ω. The elbow positioning is split into four steps: rotate the elbow to a base position, calculate an angle based on the hand position relative to the shoulder, apply corrections for close positions and apply corrections to clamp the relative hand rotation.

Base rotation

Since the current elbow position depends on the position before the ongoing IK iteration, it has to be pointing towards a fixed direction as a basis for the upcoming steps. In the base position, the elbow is always on the topmost position it can reach, thus it is always pointing upwards. Using U_s, the cross product of the shoulder's local up vector with ω, and the upper arm's local up vector U_u, the angle ϕ between them is calculated using the dot-product of their normalized vectors. The offset rotation is corrected by converting the rotation from the axis-angle representation into a quaternion and adding it to the upper arm's current rotation.

Elbow rotation from relative hand position

While some assumptions about the elbow angle are easy to realize, e.g. that it should always point away from the body center and that it should be pointing backwards when the hand is in front of the shoulder, there is still a range of around 180° on which it can move. The strongest indicator for the elbow angle is the hand position in local coordinates of the shoulder, $X_{h,s}$. When the hand is lifted from its highest to the lowest position, the elbow starts pointing downwards, then outwards and finally slightly upwards (see figure 3.6). Similar movements can be observed on all 3 axes. We use different thresholds t^i, weights w^i and minima ϕ^i_{min} for each axis and then

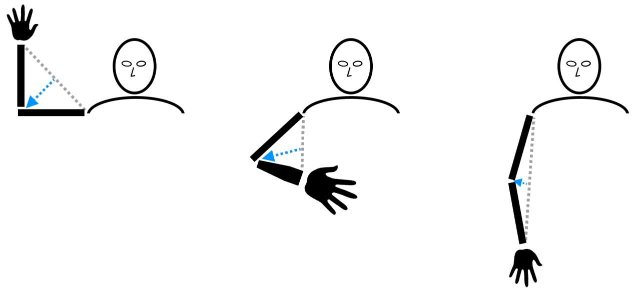


Figure 3.6.: Elbow orientations are mainly estimated using the hand to shoulder distance. The illustrations show plausible orientations when the hand is above, in front or beneath the shoulder.

combine them all by calculating the sum of the three values. Afterwards, a fixed offset angle is added and the angle ϕ is clamped to stay within a given range.

$$\phi^i = max(\phi^i_{min}, X^i_{h,s} - t^i \times w^i) \tag{3.9}$$

$$\phi = min\left(\phi_{max}, max\left(\phi_{min}, \phi_0 + \sum_i \phi^i\right)\right) \tag{3.10}$$

Elbow rotation correction

When the elbow is always set to face upwards as a first step in the elbow rotation estimation, problems occur when the hand is beneath or above the shoulder. With the constraint that the elbow is always pointing upwards, it is also always pointing into the same direction as $X_{h,s}$ when looked at in top view. Therefore, when the hand is very close to the shoulder's up vector, small movements around it can result in a 360° rotation of the arm around the shoulder. This error cannot be corrected with better parameters for the equations in section 3.1.3, because it is non-linear. Thus, in a small area around the shoulder's up vector, the elbow is corrected to point towards a predefined direction using the same procedure as described in section 3.1.3. The influence of this correction increases with decreasing distance from the hand to the shoulder's up vector. When the hand is behind the shoulder,

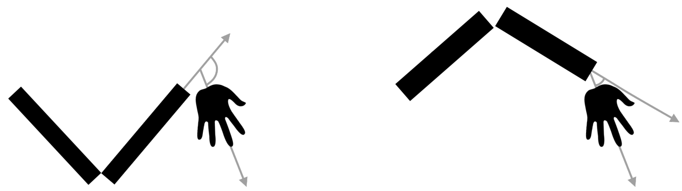
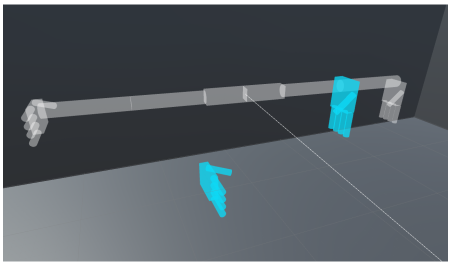


Figure 3.7.: If the angle between hand and lower arm exceeds a threshold, the elbow is rotated to correct the unrealistic pose.

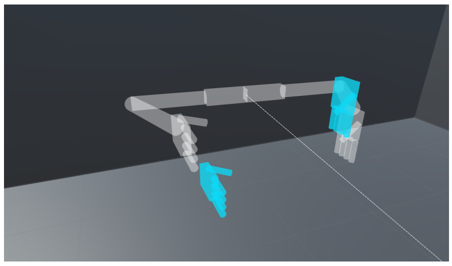
this step completely overrides the elbow pose calculated in the previous steps.

Elbow rotation from hand rotation

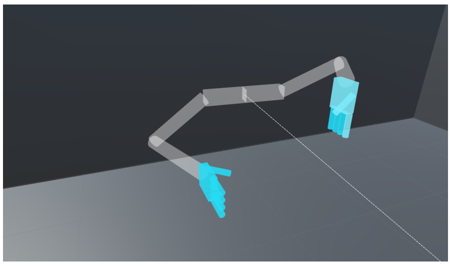
After completing the IK solving steps prior to this, shoulders and arms are placed, the hands are posed correctly and the elbow is in a plausible angle around the shoulder-hand axis. Starting from this point, some last corrections can be applied by considering the hand orientation. Although hand rotations are not used for elbow positioning, they can be added as a last step for local rotation clamping. By that, unrealistically high hand rotations can be corrected by rotating the elbow in a direction which reduces the local hand rotation. . We defined an offset angle and threshold in which no rotation correction takes place. The higher the angle is above the thresholds, the stronger is the correction. The threshold should be sufficiently high so that not much movement is visible when the user is only rotating his hand on the same place. Yet, it should be sensitive enough to disallow unrealistic rotations. We chose a very high threshold of $\pm 54°$ around the offset position to prevent unwanted rotations.



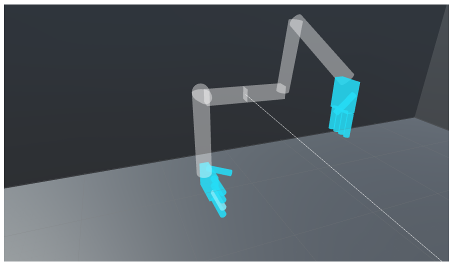
(a) Blue target hand poses and starting pose of the avatar.



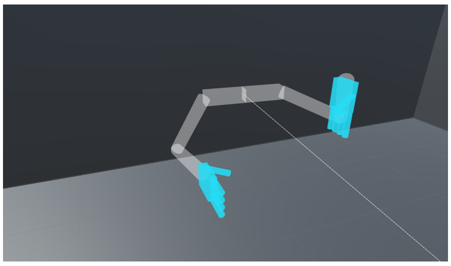
(b) Step 1: Calculate inner elbow angle.



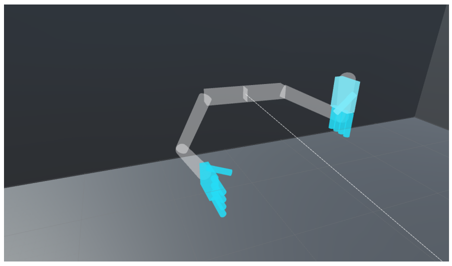
(c) Step 2: Align hand with target by rotating the upper arm.



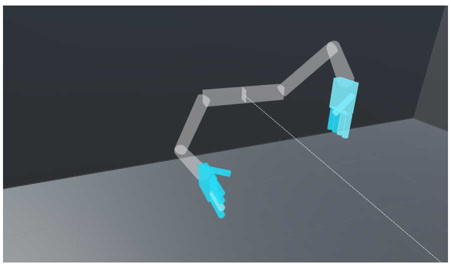
(d) Step 3: Get the elbow in the base rotation.



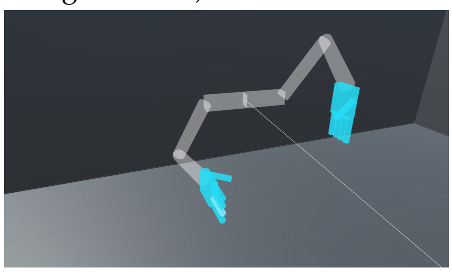
(e) Step 4: Calculate the elbow angle by shoulder to hand distance.



(f) Step 5: Correct elbow for hands close to shoulder. (Small shift on right elbow.)



(g) Step 6: Correct elbow rotation using the forward vector of the hand.



(h) Step 7: Correct elbow rotation using the right vector of the hand.

Figure 3.8.: Step by step progression of the arm IK solver.

3.2. Motion Capture and Calibration

For real time motion capture we use Optitrack Motive 1.7.2, a commercial application which includes a skeleton tracking system. The skeleton definitions and the current pose are transmitted live via local network from Motive to Unity using the official Unity plugin provided by Optitrack (2018). This data is used for upper and lower arm as well as for shoulder rotation. However, shoulder positioning is done using the IK system described in 3.1.1, because the motion capture suit cannot be worn correctly when wearing a VR HMD. Thus, Optitrack measures incorrect head orientations which lead to incorrect head to shoulder distances. For shoulder orientation, on the other hand, motion capture data is used. Furthermore, the hand rotations are also used from the VR controllers, because they are of higher precision and have a lower delay and because the user has to wear them during the study for controlling user interfaces and the bow anyhow.

Using motion capture, the final position of the avatar's body parts is highly dependent on the calibration accuracy. With a good calibration, the position and rotation of each joint can be very close to it's real pose. With a bad calibration, e.g. if the markers on the motion capture suit are not well placed or when our shoulder estimation does not work well for the users, the position of the hand can be offset a few centimeters. When starting the application, the user has to run a calibration before continuing. In this step, the user's height and wrist to wrist distance is measured using the VR HMD and motion controllers. With this information about the user, the avatar's shoulder width and arm length is set. As a last step, rotational offset between motion capture and VR is corrected by using the vector between the hands measured in both systems and calculating the offset rotation. From that moment, the avatar's shoulder rotation is set using the chest and shoulder orientations from the motion capture system.

4. Application and Study Design

With our IK solver and the motion capture interface in place, the next step is to create a study in order to find out how much having arms changes the user experience and how IK performs compared to motion capture. We created two different tasks for the study participants. In the first task, the users must use their arms for interaction. We created a simple goal-keeper game *goalie* where the players have to hit incoming balls with the correct arm or hand. The second task does not require arms for playing the game. In this game, we want to find out if the presence of arms does improve the experience even though they are not needed for interaction. Therefore, we created an archery game in which the bow holding arm is always visible to the player while aiming. In both tasks, the order of pose modes is randomized, i.e. it randomly starts in either hand only, IK or motion capture mode. The participants can practice each task once in a short tutorial round in the same mode in which they start the actual tasks. The parameters of the IK system remain constant during the whole study. Thus, they are not optimized for different use cases in order to create meaningful results for more complex applications where the pose of the user is not easily predictable.

The motion capture suits and VR motion controllers are worn by the study participants throughout the whole study out of simplicity and to prevent them from knowing which mode is active at any time.

4.1. Goalie Game

In the goalie game, IK and motion capture are compared to each other for arm posing. In this game, good arm movement is as important as good

Figure 4.1.: Left: The goalie game from the perspective of the user. Right: a user playing the goalie game with fast arm movement.

hand movement. The players' arms and hands are surrounded by colliders in four different colors. There exist four different targets in the same four colors. The objective is to hit as many of the incoming flying targets with the body part of matching color. The game is split into three 20 seconds lasting rounds with increasing speed. The amount of spawned and correctly hit targets is tracked separately for each body part for later evaluation. We chose 4 different random seeds for this game which provide a similar distribution between the targets per body part.

4.2. Archery Game

The archery game does not make use of the arms for gameplay. Nevertheless, at least the bow holding arm is always visible to the player and the line pulling arm is too when reaching for the line. In archery, the arm could help as a reference for aiming and to feel more like being in the virtual environment. Yet, these effects have to be evaluated through the study. The players have to hit as many targets as they can within 60 seconds. There is always one target in the room and when it is hit, a new target spawns at a different location inside a given space. The users have to hold the bow with one hand and pull the line with the other hand to draw an arrow. The

Figure 4.2.: Left: The archery game from the perspective of the user. Right: a user playing the archery game.

arrow is shot by releasing the line in the direction it was aiming at that time. The strength can be controlled by pulling the line farther.

4.3. Questions

The study contains 6 different questionnaires. Information about the user is asked in the pre-questionnaire before the user starts with the calibration of his avatar. After both iterations of the goalie game, there is a questionnaire about how realistic the arms moved and the difficulty of this task with these controls. After playing through both modes, the players are asked which method they preferred. The same procedure is used for the archery game. Yet, in this game the participants are asked how much the avatar felt like a part of their real body, if they feel like being in the virtual environment and if seeing one's own avatar helped in completing the task. Afterwards, they are asked which method they liked best. In the last questionnaire, the players can switch between IK, motion capture and hand only mode and move them freely without having to complete a task. After trying through all modes, they select the mode which achieves the strongest feeling of being one's own arm. The full list of questions is listed in tables 4.1 to 4.6.

All questions which are answered in a range between two extremes use a

Nr.	Question
1	What is your age?
2	What is your sex?
3	Which is your dominant hand?
4	How experienced are you in Virtual Reality? (1..6)
5	Do you study computer science or anything comparable?
6	Do you suffer colorblindness?
7	Do you have an impaired vision?

Table 4.1.: Pre-Questionnaire.

Nr.	Question
1	I felt like the virtual arm is a part of my real body. (1..6)
2	I was able to fully control the arm movement. (1..6)
3	When moving my real arm, the virtual arm moved the same. (1..6)
4	With a more accurate or less delayed arm movement I would have achieved higher scores. (1..6)
5	How easy was this task to complete with this controls? (1..6)
6	How high was your mental load? (1..6)
7	How do you rate your performance? (1..6)

Table 4.2.: Questionnaire which is answered after each iteration of the goalie game.

6-point Likert scale from 1 to 6, where 6 is the positive, agreeing answer and 1 is the negative, disagreeing answer. In questions where the users select a mode they like best, the mode names are always replaced by the order in which the modes occurred. Thus, instead of *IK*, *Motion Capture* or *Hand Only*, the users choose between *First*, *Second* and *Third* mode. All questionnaires are answered by the participants directly in VR.

Nr.	Question
1	Which method leads to the highest feeling of embodiment?
2	Which method do you prefer for playing fast games?
3	Which method do you prefer overall?

Table 4.3.: Post-Goalie questionnaire.

Nr.	Question
1	I felt like my avatar is a part of my real body. (1..6)
2	I felt like being in the virtual environment.(1..6)
3	I was able to aim quickly. (1..6)
4	Seeing my avatar helped in estimation of scale and distance in the virtual environment. (1..6)
5	Seeing my avatar helped controlling the bow. (1..6)
6	How easy was it to complete this task with this controls? (1..6)
7	How high was your mental load? (1..6)
8	How do you rate your performance? (1..6)

Table 4.4.: Questionnaire which is answered after every iteration of the archery game.

Nr.	Question
1	Which method leads to the highest feeling of embodiment?
2	Which method do you prefer for playing fast games?
3	Which method do you prefer overall?

Table 4.5.: Post-Archery questionnaire.

Nr.	Question
1	Select the mode which achieves the strongest feeling of having your own arm in VR.

Table 4.6.: Post-Questionnaire.

5. Results

In this chapter, we point out the most important questions and try to find good correlations. The raw questionnaire results are attached in the appendix in chapter A. All p-values were calculated using Welch's t-test (WELCH, 1947) for significance testing in the difference of two results and Spearman's rank (Spearman, 1904) for find linear correlations between two characteristics. We use a p-value borderline of 0.05 for accepting or rejecting the null hypothesis.

The study was completed by 76 participants of which 55 could be completed without issues. The results of the 21 participants, who suffered severe motion capture tracking issues during the study, are not considered in this evaluation. 25% of the participants were female, 93% were right-handed and 82% studied computer science. 52% had an impaired vision using the VR HMD and one participant suffered colorblindness. On a scale from 1 (not at all) to 6 (very experienced), the mean experience in VR of the participants was 2.47 with a standard deviation (SD) of 1.39.

In the goalie and archery games, the order in which the methods were played was randomized to ensure that the latter methods do not result in better scores due to increased practice of the participant. 31 participants started the goalie game with IK and 24 with motion capture. The archery game was started in hand only mode by 20 participants, 18 started with IK and 17 with motion capture. This uneven amount happened due to the usage of randomly chosen modes. However, we did not see a round by round increase of the scores in this game and the distribution is not strongly unbalanced. Thus, we expect that this distribution of amounts a mode is played first did not influence the results significantly.

5.1. Goalie Game

The questionnaire after each iteration of the goalie game (table 5.1) showed quite similar results for motion capture and IK. The difference between the mean values was never larger than 0.6 on a range from 1 to 6. Both methods achieved a strong feeling of the virtual arm being part of one's own body with values of around 4.8. The participants reported with an average of 5.11 that the virtual arms moved the same as their real arms in IK mode and 4.53 in motion capture mode. Question 4 was answered with roughly 3 in both modes. Thus, the players do not expect their scores to be greatly higher with better controls. However, this also states that they did not consider the arm movement as perfect in neither mode. Q5 and Q6 state that the task was not easy and that it was mentally demanding. Motion capture seems to be more difficult than IK with a p-value of 0.008. In total, IK achieved significantly better results than motion capture in 4 of 7 questions. The p-values of the remaining 3 questions is above 0.05, i.e. it is insignificant.

We also analyzed the influence of pose error to Q1, the feeling of embodiment. As ground truth, we used the position of the motion controllers for the hand position as well as the elbow angle obtained from motion capture for comparison. Using IK, the mean and root mean square (RMS) elbow offset angles of -0.95° and 32.98° did not correlate with the feeling of embodiment. The hand position error only shows a small correlation with the feeling of embodiment in motion capture mode with a p-value of 0.044 on the RMS elbow angle error.

After playing the game in both modes, the participants were asked to select which modes they preferred. IK was chosen by 67% of the participants for leading to the highest feeling of embodiment. 69% stated that they prefer IK for playing fast games and that they also prefer this method overall. See table A.3 for raw results.

On average, the participants hit 139 of 165 targets correctly using IK and 126 using motion capture. With a Welch's test p-value of 0.008 on the score of each player in both modes, the score does depend on the mode. There was also a slow correlation in the consistency of the scores per player with a p-value of 0.049. Thus, a player who achieved a high score in one mode was slightly more likely to achieve a high score in the second mode.

Question	mode	1	2	3	4	5	6	mean	SD	t	p
I felt like the virtual arm is a part of my real body.	IK	0	3	2	12	21	17	4.85	1.07	2.14	0.034
	MC	3	3	5	12	24	8	4.36	1.30		
I was able to fully control the arm movement.	IK	1	0	4	10	22	18	4.93	1.04	2.06	0.042
	MC	4	3	7	7	20	14	4.42	1.49		
When moving my real arm, the virtual arm moved the same.	IK	0	1	4	8	17	25	5.11	1.02	2.73	0.007
	MC	2	1	7	11	24	10	4.53	1.19		
With a more accurate or less delayed arm movement I would have achieved higher scores.	IK	9	18	11	5	7	5	2.96	1.55	-0.62	0.539
	MC	7	16	11	10	5	6	3.15	1.52		
How easy was this task to complete with this controls?	IK	0	1	10	22	16	6	4.29	0.95	2.71	0.008
	MC	4	4	18	13	11	5	3.69	1.32		
How high was your mental load?	IK	1	4	10	20	17	3	4.04	1.08	-0.77	0.444
	MC	1	2	11	20	13	8	4.20	1.13		
How do you rate your performance?	IK	0	3	19	24	8	1	3.73	0.84	1.46	0.148
	MC	3	9	16	18	6	3	3.44	1.20		

Table 5.1.: Questionnaire after each iteration of the goalie game. t is the calculated t statistics and p is the two-tailed p-value calculated using the Welch's test. IK stands for inverse kinematics and MC for motion capture.

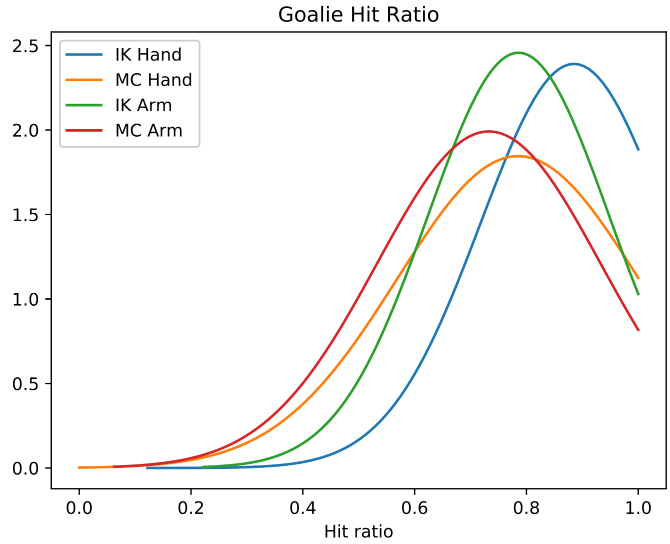


Figure 5.1.: Normal distributions of hit ratios of hand and arms in the goalie game. IK stands for inverse kinematics and MC for motion capture.

5.2. Archery Game

In the archery game, the questionnaire after IK, motion capture and hand only mode shows a much higher difference between the methods than in the goalie game. While the mean value for questions 1, 2, 3, 4, 6 and 8 were similar between hand only and IK, they were always significantly higher than the motion capture results. The only exception was question 7, where the participants stated to have an equally high mental load throughout all methods. While, with a p-value of 0.656, IK did not achieve a different feeling of embodiment than hand only mode, motion capture, with a p-value of 0.000, reduces this feeling strongly. The differences in questions 4 and 5, although present, were not large enough to be of statistical relevance. Thus, seeing an arm did not help in the estimation of scale and distance, nor for controlling the bow. With mean values of 4.89 and 4.62, IK and hand only achieved nearly the same feeling of embodiment. On average, the results for IK are very high and slightly better than in the goalie game. On the other hand, motion capture achieved an even stronger decrease in embodiment.

Question	mode	1	2	3	4	5	6	mean	SD
I felt like my avatar is a part of my real body.	IK	2	2	2	6	25	18	4.89	1.22
	MC	5	9	15	17	8	1	3.31	1.22
	HA	0	2	7	13	21	12	4.62	1.07
I felt like being in the virtual environment.	IK	0	2	1	4	32	16	5.07	0.87
	MC	2	7	6	23	13	4	3.91	1.21
	HA	1	0	5	11	26	12	4.76	1.01
I was able to aim quickly.	IK	2	1	5	11	21	15	4.69	1.22
	MC	7	14	16	12	5	1	2.95	1.23
	HA	1	2	7	13	16	16	4.62	1.23
Seeing my avatar helped in estimation of scale and distance in the virtual environment.	IK	0	4	3	10	19	19	4.84	1.17
	MC	2	9	10	18	13	3	3.73	1.24
	HA	0	5	10	14	19	7	4.24	1.16
Seeing my avatar helped controlling the bow.	IK	0	2	4	6	20	23	5.05	1.07
	MC	4	4	15	12	16	4	3.80	1.33
	HA	0	2	7	11	22	13	4.67	1.08
How easy was it to complete this task with this controls?	IK	0	3	6	12	17	17	4.71	1.17
	MC	10	13	11	14	7	0	2.91	1.31
	HA	1	4	6	15	20	9	4.38	1.20
How high was your mental load?	IK	2	12	19	12	9	1	3.31	1.14
	MC	1	8	16	18	10	2	3.62	1.10
	HA	3	12	18	12	9	1	3.27	1.18
How do you rate your performance?	IK	0	5	4	23	18	5	4.25	1.03
	MC	9	15	17	13	1	0	2.67	1.06
	HA	1	6	11	18	16	3	3.93	1.14

Table 5.2.: Questionnaire which is answered after every iteration of the archery game.

See table 5.3 for the results of Welch's test between all modes and table 5.2 for raw answers. We did not find a correlation between the elbow angle error in IK mode and the hand position offset in motion capture mode with the feeling of embodiment. Furthermore, the score did not correlate with the feeling of embodiment.

In the post-questionnaire of the archery game 71% of the participants stated that IK led to the highest feeling of embodiment, 26% selected hand only and 2 persons, or 3%, selected motion capture. For playing fast games and overall in VR, 69% and 71% prefer IK, 27% and 23% hand only and 3% and 6% motion capture. The raw results are listed in table A.5.

The hits per participant varied strongly between the modes. In IK mode,

Modes	value	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
inverse kinematics, motion capture, hand only	H	46.02	33.71	50.05	22.13	27.89	45.70	3.02	47.15
	p	0.000	0.000	0.000	0.000	0.000	0.000	0.221	0.000
inverse kinematics vs motion capture	t	6.75	5.74	7.41	4.77	5.41	7.53	-1.43	7.85
	p	0.000	0.000	0.000	0.000	0.000	0.000	0.467	0.000
inverse kinematics vs hand only	t	1.24	1.70	0.31	2.67	1.85	1.44	0.16	1.56
	p	0.656	0.274	2.274	0.026	0.203	0.462	2.614	0.363
motion capture vs hand only	t	-5.93	-3.99	-7.08	-2.20	-3.75	-6.09	1.57	-5.91
	p	0.000	0.000	0.000	0.090	0.001	0.000	0.358	0.000

Table 5.3.: Statistical analytics of the questionnaire after each iteration of the archery game. Kruskal-Wallis H-test is used in the first line to see if there are significant differences between the three modes. Lines 2-4 show the Bonferroni-corrected Welch's test results between different modes.

players hit 15.98 targets on average, 14.91 in hand only mode and 8.35 with motion capture. The difference between IK and hand only mode was insignificant with a Welch's p-value of 0.50. The difference between IK and motion capture, on the other hand, was very strong with a p-value of 1×10^{-6}. The archery scores were consistent between different modes. Thus, a player who achieved a high score in one mode, was more likely to achieve high scores in other modes as well with p-values ranging from 1.95×10^{-5} between hand only and motion capture to 2.63×10^{-9} between IK and hand only.

5.3. Post-Questionnaire and Additional Results

In the post-questionnaire, participants were able to test the modes IK, motion capture and hand only without being distracted by playing a fast game. Thus, they could compare the actual movement of the arms calmly. In this questionnaire, 15% selected that hand only mode achieved the strongest feeling of having your own arm in VR. 54% opted for IK and 31% for motion capture. The post-questionnaire results are listed in table A.6.

Although we tried to find random seeds of similar distribution and difficulty for the goalie game, the random seeds correlated with the player scores. For

Game	0	1	2	3	r	p
Goalie	139.739	136.61	128.25	127.63	-0.197	0.0389
Archery	13.40	13.04	12.8	-	-0.055	0.479

Table 5.4.: Mean score per random seed ordered by score. Spearman's rank r and p-value show correlation between seed index and score in the goalie game, but none in archery.

this test, we calculated the average score per random seed and ordered them by decreasing score. All random seeds were then replaced by their index in the list from 0, for the seed with the highest score, to n-1, for the seed with the lowest score. In the goalie game, a small correlation between seed index and score can be seen. In the archery game, the scores are independent of the seeds. See table 5.4.

We did not find a connection between VR experience and scores. Also, the participants' performances were not consistent between the goalie and the archery games.

We decided to create our own IK solver for this study since we were not satisfied with available IK solvers. A comparison between different IK systems would have blown up the scope of the study. Hence, we cannot say which method felt the most natural for the users. Yet, we can compare the elbow angle differences between the different IK systems. After three test runs of the study, we improved the IK solver parameters of our solution and SAFullBodyIK to minimize the elbow angle error we measured through the motion capture elbow. The Final IK solver does not provide parameters which we could use to reduce the error. Through parameter optimizations, we were able to reduce the RMS angle error of the test run to less than 20 degrees in our solution, to less than 30 with SAFullyBodyIK while Final IK was nearly at 40 degrees. After using these parameters in the study on 55 participants, it is clear that there is no single solution that fits all users. We ran a replay of all study participants' recordings with 10x original speed and compared the elbow angles of all IK systems with the elbow angle tracked by the motion capture system. The mean and RMS errors of our IK solver, the Final IK module *Limb IK* and the SAFullBodyIK are displayed in table 5.5. In the goalie game, our solution achieved only slightly better results than Final IK and SAFullBodyIK. However, we achieved clearly the best

Game	System	Left Mean	Right Mean	Left RMS	Right RMS
Goalie	Ours	0.41	0.92	40.82	40.89
	Final IK	3.77	-0.19	42.45	41.07
	SAFullBodyIK	-11.21	13.43	43.15	43.16
Archery	Ours	-7.28	-3.36	47.14	41.76
	Final IK	-9,59	-5.03	51.02	45.19
	SAFullBodyIK	-16.93	-17.73	50.85	46.60

Table 5.5.: Comparison of the angle error in degrees using different IK systems. The elbow angle from motion capture is used as a reference for all error calculations.

Game	Left	Right
Goalie	24.56	24.19
Archery	52.66	23.62

Table 5.6.: RMS of the elbow angle errors in degrees to the mean elbow rotation within the hand's surrounding cell. The normalized hand position (in a -1 to 1 range) is sampled in a 20 x 20 x 20 entries large space to calculate mean elbow angles within each cell.

results in the archery game. Final IK and SAFullBodyIK achieved nearly the same accuracy in archery with a RMS error of roughly 51° and 46° for left and right arm. Table 5.6 shows how strong elbow angles vary within a small positional range. The mean elbow angle was calculated for a single cell in a 20 x 20 x 20 range, which is a size of 7 x 7 x 7cm for an average arm of 70cm length.

Since the users wore the hood of the motion capture suit on top of the HMD, Optitrack had difficulties to track the head correctly. To circumvent this issue, we decided to use the shoulder IK solver not only in IK mode, but also in motion capture mode. However, this issue also makes it more difficult to evaluate the shoulder estimation since we could not track correct head and shoulder poses using motion capture. Instead, we used publicly available motion capture data as a reference for evaluation (table 5.7). We used four datasets available on the Unity Asset Store which provide a large variety in animations:

Unity Raw Mocap standing, slow walking, little interaction

Basic Motion fighting, crouching, drinking, pulling, pushing, sitting

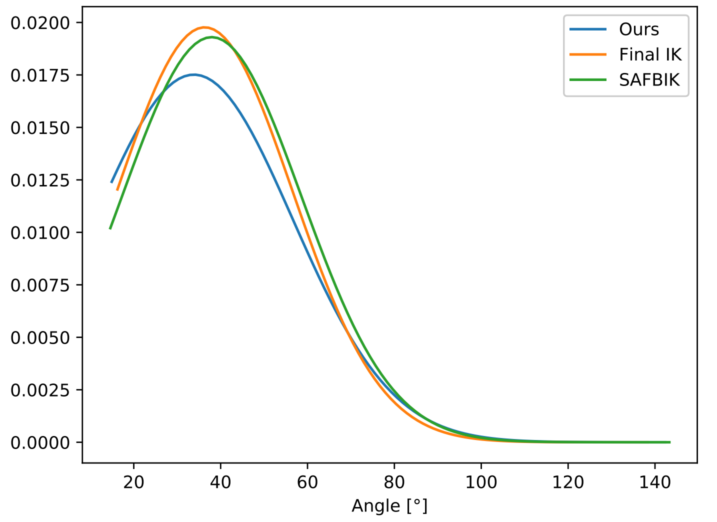


Figure 5.2.: Normal distributions of the RMS elbow angle error using different IK systems in the goalie game.

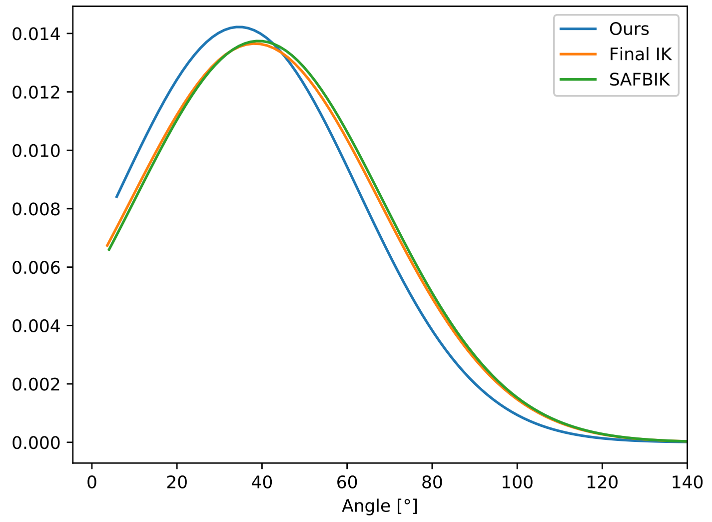


Figure 5.3.: Normal distributions of the RMS elbow angle error using different IK systems in the archery game.

Dataset	Center Angle	Center Pos.	Left Pos.	Right Pos.
Unity Raw Mocap	9.4°	3.4 cm	4.0 cm	3.7 cm
Basic Motion	35.3°	8.9 cm	10.2 cm	10.6 cm
Mixed Motion	29.8°	9.0 cm	13.2 cm	13.4 cm
Mixed Motion 2	33.6°	9.4 cm	14.0 cm	11.5 cm

Table 5.7.: RMS shoulder angle and position errors using our shoulder IK solver on motion capture datasets. Our estimation for shoulder center, left shoulder and right shoulder was compared to the motion capture poses.

Mixed Motion baseball, ninja poses, crawling

Mixed Motion 2 bodybuilding, golf, American football.

Since we optimized our IK solver for VR games, we only used the animations which are likely in VR games and removed animations which include lying, sprinting, jumping, doing backflips and similar activities.

Unity Raw Mocap is the easiest of all datasets since only standing, slow walking and simple interactions are left after removing all sprinting and jumping animations. With a RMS position error of 3.4 to 4.0 cm, the shoulder estimation was very accurate. The other motion capture animations are more difficult and provide a good reference for accuracy in very interactive games. In these animations, the error is approximately three times as high. The shoulder center angle error is between 29.8° and 35.3° compared to 9.4° using Unity Raw Mocap. The center, left and right RMS position errors are between 8.9 to 14.0 cm.

Using the same animations, we also created a comparison between different arm IK solvers using the correct shoulder position of the motion capture records (table 5.8). Our IK solver achieved equal or better results than Final IK and SAFullBodyIK with RMS elbow angle errors between 11.1° and 42.0°. The error of Final IK ranges from 31.2° to 52.6° and is very consistent across the datasets. SAFullBodyIK on the other hand delivers inconsistent accuracy with errors ranging from 29.5° to 70.9°.

The delay between the motion capture pose and the real pose is difficult to measure precisely. However, we were able to narrow down the delay to approximately two frames of the HMD, between 11ms and 22ms, on top of the delay of the Oculus Touch controllers. For the user, this means a delay

Dataset	Our IK w/ shoulder		Our IK		Final IK		SAFullBodyIK	
	Left	Right	Left	Right	Left	Right	Left	Right
Unity Raw Mocap	11.8	23.5	11.1	27.5	44.6	31.2	40.7	30.0
Basic Motion	22.8	33.0	24.2	32.2	42.4	34.8	61.5	57.7
Mixed Motion	17.6	26.4	16.2	29.7	52.6	33.5	47.0	29.5
Mixed Motion 2	30.1	27.8	42.0	30.8	37.0	42.3	38.5	70.9

Table 5.8.: Comparison of different IK solvers using the RMS elbow angle error on four motion capture datasets. All IK systems except for 'Our IK with shoulder' use the shoulder pose from the motion capture input. 'Our IK with shoulder' uses our own shoulder IK to see the impact of the subsequent errors.

of 3 frames for motion capture controls and 1 frame using the Oculus Touch controller from movement to screen. In some cases when Optitrack had difficulties to track the markers, the delay was increased by 2-3 frames.

6. Discussion

The main goal of this thesis was to find out how having arms in VR changes the user experience and how an IK system which uses only head and hand poses from VR tracking performs compared to an industrial motion capture system.

In the archery game, where hand only mode is compared with both motion capture and IK, motion capture always achieved the worst scores and answers in the questionnaire of all modes. This can be explained by looking at the technical limitations of the motion capture setup. The additional delay of the motion capture system and network transmission result in unresponsive controls of the bow - especially when compared with the positionally tracked controllers which are used in the other modes. Also, this game requires much more precision in hand position for aiming than the goalie game where $\pm 5cm$ do not make a big difference. A technical limitation of motion capture is the difficulty to distinguish between two markers when they are very close to each other or when they are concealed. This happened very often in the archery game when the participants grabbed the line or when they pulled the line closely to their chest. In these cases, the skeleton tracking started shaking, i.e. the joints made very fast movements which made it difficult to aim and decreased the feeling of the virtual arm being one's own.

IK and hand only mode use the same high precision and low delay hand tracking and are therefore better suited for comparison between hand only avatars and avatars with arms in this game. In the questionnaire after each round of the game, both modes were rated very similarly throughout all questions. Even though the difference between IK and hand only is not large enough to be of importance, the high mean value of 4.89 out of 6 in question 1 should be mentioned. Our IK solver created greatly larger RMS angle errors for the left elbow than in the goalie game. Yet, because this arm is usually used as bow holding arm and therefore rarely moved and not important for interaction, it still lead to a very high feeling of embodiment and the angle difference did not seem to bother the participants much. Seeing one's avatar did help in controlling the bow. However, having only hands achieved the same effect as having fully modeled arms with mean values of 4.67 and 5.05. Nevertheless, when asked which method they liked best overall, 71% chose IK, 6% opted for motion capture and only 23% preferred hand only mode. This nearly 4:1 ratio between arms and hand only is a clear indicator that arms can improve the user experience. Even more, it shows that with nearly a 4:1 ratio between IK and hand only and a 4:1 ratio between hand only and motion capture, the ability to see one's arms in VR is not sufficient for all use cases. When precise controls are required by a game, users prefer an avatar with only the body parts needed to play a game over an imprecise avatar with additional body parts. However, with identical precision, having arms is preferred even when they are not needed for playing the game.

In the post-questionnaire, the participants were able to try all three modes (IK, motion capture, hand only) calmly. They had to select the mode which achieved the strongest feeling of having one's own arm in VR. They were not told what the differences between the modes were or which aspects they should focus on, but they were free to choose what is important to them. 54% opted for IK and 34% for motion capture. Only 15% preferred hand only. These are clear results that having arms does improve the feeling of embodiment compared to only having hands when users actively focus on their avatars. It is also a surprising indicator that IK animated arms can feel more realistic than arms animated by motion capture.

With an average score of 139 hits with IK and 126 hits with motion capture, IK is not even inferior to motion capture in games with heavy elbow movement. The questionnaire after each round showed very promising results for both modes in all aspects. With an approximate mean of 5 points out of 6, the arms moved realistically, were fully controllable and hence felt like they were part of one's real body. The players believed that they could achieve slightly higher scores with better controls, yet they did not believe that the controls are their main issue. It is also noticeable how similar both modes were rated. 2 questions showed clearly that IK achieved better results, 2 questions only achieved a small correlation with p-values of 0.034 and 0.042 and the remaining 3 questions achieved equal results. Some participants even asked if there was a difference between both modes. Thus, the questionnaire after the goalie game, where 69% selected IK and 31% selected motion capture as best method, should be considered with caution.

Looking at side effects which arms could potentially invoke; no significant improvements could be observed in this study. Having arms in VR did neither increase the feeling of being in the virtual environment nor in controlling the bow. The only positive side effect of having arms compared to only having hands is that it helps in estimation of scale and distance in the virtual environment. It proved to be more important to have fast and precise hands than having fully animated arms for achieving the feeling of being in the virtual environment during the archery game.

We were confident that we could create IK solvers that are much better for elbow positioning than other available systems. The comparison in table 5.5 does support this assumption. Our solution worked slightly better in the goalie game and it was significantly better in the archery game than Final IK and SAFullBodyIK. Table 5.6 can be used as a reference for what is achievable with IK using the hand positions only. It shows how high the RMS angle error still is when using the mean elbow angle in a small cube of roughly 7x7x7cm in a single scenario. Decreasing the error down to zero is impossible when using only the hand and head poses and it is even more difficult to create an IK solver that works equally precise for different users. However, parameters can be optimized for different use cases. For example, we could use different parameters for the bow holding arm than for the line pulling arm, because both poses are very predictable. We decided to use the same parameters for both games and for the post-questionnaire where players were able to move their arms freely and observe their poses. Nevertheless, our IK solver showed the highest accuracy in both games and worked significantly better in the comparison with the motion capture datasets. Thus, we do not believe that other IK solvers would have achieved better results towards IK in the study.

Our shoulder IK solver achieved a high accuracy for simple animations when compared to different motion capture datasets with an RMS error of 4cm. When applied to more difficult animations like golfing or leaning back in a chair, the error increases to roughly 10cm. This error influences the upper arm position and thereby also the arm IK solvers. When the shoulder position is incorrect, the arm IK estimates the elbow position for the incorrect elbow. Thus, it leads to a subsequent offset of the elbow angles. Nevertheless, table 5.8 shows that our arm IK achieved comparable accuracy with our shoulder estimation to the IK estimation using the correct shoulder position.

Considering how much just a small error in the shoulder rotation can change the hand position using motion capture, 8cm to 10cm errors in the hand position are not bad. Although it could be possible to reduce it down even further using better motion capture systems. Optitrack provides solutions for VR HMDs to circumvent the head tracking issues which occur when wearing an infrared light emitting HMD. With a good head tracking, the head to shoulder distance and rotation could be improved strongly, resulting in more accurate hand positions. An additional reduction of the delay could enhance the experience even more.

7. Conclusion

In this thesis we found out that IK can be used for animating arms and is at least on par with and in most cases better than motion capture considering the different benefits and drawbacks of these methods. Having arms does not necessarily improve the experience if the arms are not needed and the users are concentrating on a demanding task. Even more, if side-effects of having arms influence the controls negatively, players prefer not to have arms, but precise controls. When the users are in a calm situation and have time to observe their hands and arms, they clearly prefer having arms over having hands only. In these use cases, IK and motion capture were chosen by the same amount of participants as the method which achieves the strongest feeling of embodiment for them. Also in very demanding games, where arms are needed for interaction, IK shows very promising results and is not inferior to motion capture. In some cases, the users do not even notice a difference between those methods.

With this knowledge, developers can create experiences which use arms for interaction without the need of additional hardware. Some examples are boxing games or a sword fighting game where the player holds a shield with one arm. Yet, there is no necessity to add arms to all VR applications when they are not needed for interaction.

We achieved meaningful results that IK works equally good as motion capture in the goalie game and that motion capture and IK create a stronger feeling of embodiment than hand only mode when the users have time to focus on their body. The differences between the answers in the archery game and the post-questionnaire also indicate that less demanding games or short breaks during the game could have influenced the feeling of embodiment between different methods when the users then have more time to focus on their body. In this case, having very demanding, high score-based games at first and the ability to watch their body calmly afterwards proved to be a good decision for providing significant results in different use cases. Yet, it would be interesting to investigate other, less demanding applications like social meetings or simulations which could create different outcomes in terms of embodiment. A questionnaire about different IK solvers would be a better method to compare IK solvers than by the RMS elbow angle error. However, this topic remains open for future research. Furthermore, adding arms to one's avatar is just the beginning. While feet and legs might not be easy to animate without additional tracking devices, chest and belly could possibly be animated with IK as well. Especially since the shoulder is already estimated for the animation of the arms.

Appendix

Bibliography

Botvinick, Matthew and Jonathan Cohen (1998). "Rubber hands 'feel' touch that eyes see." In: *Nature* 391.6669, pp. 756–756. DOI: 10.1038/35784 (cit. on p. 3).

Chang, Jian et al., eds. (2017). *Next Generation Computer Animation Techniques*. Springer International Publishing. DOI: 10.1007/978-3-319-69487-0 (cit. on p. 4).

Costa, Mark R., Sung Yeun Kim, and Frank Biocca (2013). "Embodiment and Embodied Cognition." In: *Virtual Augmented and Mixed Reality. Designing and Developing Augmented and Virtual Environments*. Springer Berlin Heidelberg, pp. 333–342. DOI: 10.1007/978-3-642-39405-8_37 (cit. on p. 3).

Lee, Dongik and Hankyu Lim (2015). "Virtual Reality Contents using the OculusLift and Kinect." In: *Proceedings of the MCSI*, pp. 102–105 (cit. on p. 5).

Optitrack (2018). *Optitrack Plugin for Unity*. http://optitrack.com/downloads/plugins.html. (Visited on 06/05/2018) (cit. on p. 19).

RootMotion (2018). *Final IK*. https://assetstore.unity.com/packages/tools/animation/final-ik-14290. (Visited on 06/05/2018) (cit. on p. 2).

Schultze, Ulrike (2010). "Embodiment and presence in virtual worlds: a review." In: *Journal of Information Technology* 25.4, pp. 434–449. DOI: 10.1057/jit.2010.25 (cit. on p. 1).

Spanlang, Bernhard, Jean-Marie Normand, David Borland, et al. (2014). "How to Build an Embodiment Lab: Achieving Body Representation Illusions in Virtual Reality." In: *Frontiers in Robotics and AI* 1. DOI: 10.3389/frobt.2014.00009 (cit. on p. 4).

Spanlang, Bernhard, Jean-Marie Normand, Elias Giannopoulos, et al. (2010). "A first person avatar system with haptic feedback." In: *Proceedings of the 17th ACM Symposium on Virtual Reality Software and Technology - VRST '10*. ACM Press. DOI: 10.1145/1889863.1889870 (cit. on p. 4).

Spearman, C. (1904). "The Proof and Measurement of Association between Two Things." In: *The American Journal of Psychology* 15.1, p. 72. DOI: 10.2307/1412159 (cit. on p. 27).

Steed, Anthony et al. (2016). "An 'In the Wild' Experiment on Presence and Embodiment using Consumer Virtual Reality Equipment." In: *IEEE Transactions on Visualization and Computer Graphics* 22.4. search: virtual reality embodiment motion capture, pp. 1406–1414. DOI: 10.1109/tvcg.2016.2518135 (cit. on p. 4).

StereoArts (2018). *SAFullBodyIK*. https://github.com/Stereoarts/SAFullBodyIK. (Visited on 06/05/2018) (cit. on p. 2).

Tiator, Marcel et al. (2018). "Trampoline Jumping with a Head-Mounted Display in Virtual Reality Entertainment." In: *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*. Springer International Publishing, pp. 105–119. DOI: 10.1007/978-3-319-73062-2_8 (cit. on p. 4).

WELCH, B. L. (1947). "THE GENERALIZATION OF 'STUDENT'S' PROBLEM WHEN SEVERAL DIFFERENT POPULATION VARLANCES ARE INVOLVED." In: *Biometrika* 34.1-2, pp. 28–35. DOI: 10.1093/biomet/34.1-2.28 (cit. on p. 27).

Abbreviations

FK Forward Kinematics. 13

HMD head mounted display. 1, 2, 4, 8–11, 19, 27, 34, 36, 42

IK Inverse Kinematics. v, 1–5, 7–9, 13–15, 17–19, 21, 23, 24, 27, 28, 30–37, 39–44

RMS root mean square. 28, 33–37, 39, 41, 44

SD standard deviation. 27

VR Virtual Reality. v, 1–5, 7, 8, 19, 21, 24, 27, 31–33, 36, 39–43

Appendix A.

Raw Study Results

Question	Answer							
What is your age?	Mean 26.18	SD 4.93	Min 15.00	Max 38.00				
What is your sex?	Male 41 (75%)	Female 14 (25%)	Other 0 (0%)					
Which is your dominant hand?	Left 4 (7%)	Right 51 (93%)						
How experienced are you in Virtual Reality	1 16	2 17	3 9	4 9	5 1	6 3	Mean 2.47	SD 1.39
Do you study computer science or anything comparable?	Yes 45 (82%)	No 10 (18%)						
Do you suffer colorblindness?	Yes 1 (2%)	No 54 (98%)						
Do you have an impaired vision?	Yes 29 (52%)	No 26 (48%)						

Table A.1.: Pre-Questionnaire.

Question	mode	1	2	3	4	5	6	mean	SD
I felt like the virtual arm is a part of my real body.	IK	0	3	2	12	21	17	4.85	1.07
	MC	3	3	5	12	24	8	4.36	1.30
I was able to fully control the arm movement.	IK	1	0	4	10	22	18	4.93	1.04
	MC	4	3	7	7	20	14	4.42	1.49
When moving my real arm, the virtual arm moved the same.	IK	0	1	4	8	17	25	5.11	1.02
	MC	2	1	7	11	24	10	4.53	1.19
With a more accurate or less delayed arm movement I would have achieved higher scores.	IK	9	18	11	5	7	5	2.96	1.55
	MC	7	16	11	10	5	6	3.15	1.52
How easy was this task to complete with this controls?	IK	0	1	10	22	16	6	4.29	0.95
	MC	4	4	18	13	11	5	3.69	1.32
How high was your mental load?	IK	1	4	10	20	17	3	4.04	1.08
	MC	1	2	11	20	13	8	4.20	1.13
How do you rate your performance?	IK	0	3	19	24	8	1	3.73	0.84
	MC	3	9	16	18	6	3	3.44	1.20

Table A.2.: Questionnaire which is answered after every iteration of the goalie game.

Question	Answer	
Which method leads to the highest feeling of embodiment?	Inverse Kinematics 37 (67%)	Motion Capture 18 (33%)
Which method do you prefer for playing fast games?	Inverse Kinematics 38 (69%)	Motion Capture 17 (31%)
Which method do you prefer overall?	Inverse Kinematics 38 (69%)	Motion Capture 17 (31%)

Table A.3.: Post-Goalie questionnaire. Note that mode names were hidden and the order was randomized.

Question	mode	1	2	3	4	5	6	mean	SD
I felt like my avatar is a part of my real body.	IK	2	2	2	6	25	18	4.89	1.22
	MC	5	9	15	17	8	1	3.31	1.22
	HA	0	2	7	13	21	12	4.62	1.07
I felt like being in the virtual environment.	IK	0	2	1	4	32	16	5.07	0.87
	MC	2	7	6	23	13	4	3.91	1.21
	HA	1	0	5	11	26	12	4.76	1.01
I was able to aim quickly.	IK	2	1	5	11	21	15	4.69	1.22
	MC	7	14	16	12	5	1	2.95	1.23
	HA	1	2	7	13	16	16	4.62	1.23
Seeing my avatar helped in estimation of scale and distance in the virtual environment.	IK	0	4	3	10	19	19	4.84	1.17
	MC	2	9	10	18	13	3	3.73	1.24
	HA	0	5	10	14	19	7	4.24	1.16
Seeing my avatar helped controlling the bow.	IK	0	2	4	6	20	23	5.05	1.07
	MC	4	4	15	12	16	4	3.80	1.33
	HA	0	2	7	11	22	13	4.67	1.08
How easy was it to complete this task with this controls?	IK	0	3	6	12	17	17	4.71	1.17
	MC	10	13	11	14	7	0	2.91	1.31
	HA	1	4	6	15	20	9	4.38	1.20
How high was your mental load?	IK	2	12	19	12	9	1	3.31	1.14
	MC	1	8	16	18	10	2	3.62	1.10
	HA	3	12	18	12	9	1	3.27	1.18
How do you rate your performance?	IK	0	5	4	23	18	5	4.25	1.03
	MC	9	15	17	13	1	0	2.67	1.06
	HA	1	6	11	18	16	3	3.93	1.14

Table A.4.: Questionnaire which is answered after every iteration of the archery game.

Question	Answer		
Which method leads to the highest feeling of embodiment?	Hand Only 14 (26%)	Inverse Kinematics 39 (71%)	Motion Capture 2 (3%)
Which method do you prefer for playing fast games?	Hand Only 15 (27%)	Inverse Kinematics 38 (69%)	Motion Capture 2 (3%)
Which method do you prefer overall?	Hand Only 13 (23%)	Inverse Kinematics 39 (71%)	Motion Capture 3 (6%)

Table A.5.: Post-Archery questionnaire. Note that mode names were hidden and the order was randomized.

Question	Answer		
Select the mode which achieves the strongest feeling of having your own arm in VR.	Hand Only 8 (15%)	Inverse Kinematics 30 (54%)	Motion Capture 17 (31%)

Table A.6.: Post-Questionnaire. Note that mode names were hidden and the order was randomized.