Human Upper-Body Inverse Kinematics for Increased Embodiment in Consumer-Grade Virtual Reality

Mathias Parger
Graz University of Technology
Graz, Austria
mathias.parger@icg.tugraz.at

Joerg H. Mueller
Graz University of Technology
Graz, Austria
joerg.mueller@tugraz.at

Dieter Schmalstieg
Graz University of Technology
Graz, Austria
schmalstieg@tugraz.at

Markus Steinberger
Graz University of Technology
Graz, Austria
steinberger@icg.tugraz.at

ABSTRACT

Having a virtual body can increase embodiment in virtual reality (VR) applications. However, comsumer-grade VR falls short of delivering sufficient sensory information for full-body motion capture. Consequently, most current VR applications do not even show arms, although they are often in the field of view. We address this shortcoming with a novel human upper-body inverse kinematics algorithm specifically targeted at tracking from head and hand sensors only. We present heuristics for elbow positioning depending on the shoulder-to-hand distance and for avoiding reaching unnatural joint limits. Our results show that our method increases the accuracy compared to general inverse kinematics applied to human arms with the same tracking input. In a user study, participants preferred our method over displaying disembodied hands without arms, but also over a more expensive motion capture system. In particular, our study shows that virtual arms animated with our inverse kinematics system can be used for applications involving heavy arm movement. We demonstrate that our method can not only be used to increase embodiment, but can also support interaction involving arms or shoulders, such as holding up a shield.

CCS CONCEPTS

• **Human-centered computing** → **Virtual reality**; • **Computing methodologies** → *Motion processing*; Motion capture;

KEYWORDS

Virtual Reality, Inverse Kinematics, Motion Capture, Animation, Embodiment, Presence

ACM Reference Format:
Mathias Parger, Joerg H. Mueller, Dieter Schmalstieg, and Markus Steinberger. 2018. Human Upper-Body Inverse Kinematics for Increased Embodiment in Consumer-Grade Virtual Reality. In *VRST 2018: 24th ACM Symposium on Virtual Reality Software and Technology (VRST '18), November 28-December 1, 2018, Tokyo, Japan.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3281505.3281529

VRST '18, November 28-December 1, 2018, Tokyo, Japan

ACM ISBN 978-1-4503-6086-9/18/11…$15.00
https://doi.org/10.1145/3281505.3281529

1 INTRODUCTION

In Virtual Reality (VR), one of the key components of creating presence is embodiment, the feeling of owning the virtual body [Schultze 2010]. While it is commonly assumed that bringing additional body parts into VR improves embodiment, this is only true if the body parts can be sufficiently controlled. Displaying body parts which do not follow the movement of the user can even reduce the feeling of embodiment, if the mismatches between real and virtual body are too high [Steed et al. 2016]. If the user's physical movements are not properly replicated or if the virtual body moves on its own, the user does not identify with the body. Application developers therefore prefer to only display body parts which can be animated precisely.

A straight-forward implementation of an animated body in VR requires full-body motion capturing. Unfortunately, such motion capture systems occupy a large space, require the user to wear a special suit, and are not affordable for consumers. Inexpensive depth sensors, such as the Microsoft Kinect, do not have a high enough resolution and frame rate to compete with professional motion capture systems.

Instead of using motion capturing or depth sensors, we propose to infer a human upper-body pose from the standard tracking sensors that users of consumer-grade VR already own: one tracking sensor for the headset and two tracking sensors for controllers held in the left and right hand. Since no direct pose measurements are available for arms and shoulders, we use inverse kinematics (IK) to calculate the missing joint angles between the end-effectors of a kinematic chain (in our case, the human arms).

In general, IK problems are underdetermined, as multiple solutions exist to arrange a kinematic chain to match a given end-effector configuration. IK in robotics usually optimizes for minimal path length, which produces unnatural human poses. We find that IK creates best results for mimicking human motion if biologically motivated constraints for the joints are used and only a few joints have to be estimated. Therefore, we restrict the IK to arms, since no tracking information on torso and legs is available. While humans rarely glance at their own legs, adding arms has high potential of improving embodiment, since the arms are often visible when the user interacts with the environment. Interactions can be extended from just using hands to include wrists or arms, for example, when boxing, using a shield or wearing gadgets on the lower arm.

In this paper, we introduce a novel IK approach that exploits the constraints imposed by the human body and typical human behavior to compensate for the frugal tracking abilities of consumer-grade VR. We define kinematic chains starting at the head (rather than the spine, which is more common) and introduce heuristics for the shoulder and elbow poses targeting usual VR scenarios. We compare our approach to conventional IK solutions and to motion capturing. We report on a user study with 55 participants, which compares our approach to full motion capturing and to a base condition with hands only. Thus, we make the following contributions:

- Our inverse kinematics algorithm targets first-person VR experiences with head and hand tracking. Since it is specifically tailored for the application in VR, it results in more realistic poses compared to other state of the art human or general IK solutions.
- We report on a study investigating the effect of having arms on embodiment.
- Moreover, we investigate whether our IK solution can compete with a professional motion capture system.

2 RELATED WORK

Embodiment in VR [Biocca 1997; Kilteni et al. 2012], can be related to experiments on evoking body ownership through visual-tactile correlation [Botvinick and Cohen 1998; Petkova and Ehrsson 2008], some of which were later reproduced in VR [Slater 2008; Slater et al. 2010]. Kokkinara and Slater [Kokkinara and Slater 2014] found that visual-motor correlation also increases embodiment. However, Steed et al. [2016] showed that incorrect poses can also decrease embodiment, suggesting that it might be better to not show limbs if their pose is not accurate enough.

Most of the research on embodiment in VR uses expensive motion capture systems [Spanlang et al. 2014, 2010]. Consumer-grade depth sensors can be used for body tracking [Lee and Lim 2015], but do not provide the needed accuracy. Some VR systems present a full body using IK [Jiang et al. 2016; Roth et al. 2016; Tan et al. 2017], but fail to evaluate their system in terms of pose error or run a user study to evaluate embodiment, making it difficult to assess any effect on embodiment. In this work, we aim to fill this gap.

While the inverse kinematics problem originally arose in robotics, where first solutions were found [Paul 1981], they were quickly adopted in computer graphics for animation. Aristidou et al. [2018] give a survey with an extensive overview of IK methods used in computer graphics. In the survey they categorize IK solvers into four main categories. We want to discuss the practicability of each category for our specific application scenario and present a few more examples specialized on human IK.

- **Analytic solvers** are simple, fast to compute and do not have convergence problems that numerical solvers have, but it might be difficult to implement constraints or multiple tasks to influence which of all possible solutions is computed. Human arm IK solvers often focus on mapping the hand position to the elbow position using various parameters [Gielen et al. 1997; Gielen 2009; Kondo 1994]. For example, Kondo [1994] proposed an IK solver for arms based on the sensorimotor transformation model [Soechting and Flanders 1989] that approximates the arm posture by linearly mapping the spherical coordinates of the hand relative to the shoulder.
- **Numerical solvers** either use a first (Jacobian) or second order (Newton) approximation of the forward kinematics or some heuristics to iteratively solve the IK problem. These methods create slow and smooth movements at the cost of an iterative process that requires more computational effort than an analytic solution. Due to their iterative nature, they may also run into problems of singularities and non-convergence. One advantage is that further targets can easily be added to the iterative optimization. Examples include work minimization [Admiraal et al. 2004; Kang et al. 2005], angular velocity minimization [Wang 1999] or joint limit distance maximization [Faria et al. 2018; Kim and Rosen 2015]. Especially joint limit avoidance leads to natural relaxed poses that are useful for simple human activities, but may not work well for more complex activities like sports, where strained poses close to joint limits are taken more frequently.
- **Data-driven solvers** are based on collected data, usually motion capture data that is used to find a similar solution to the current pose [Artemiadis et al. 2010; Asfour and Dillmann [n. d.]; Liang and Liu [n. d.]]. Machine learning and recently especially deep learning techniques are very popular at the moment as they can provide high quality, specialized solutions depending on the quality and amount of data they were trained with. The disadvantages of data-driven methods are the expensive data acquisition, wrong solutions for poses that are not covered well in the training data and errors caused by low quality data.

The method proposed in this paper uses an analytic, parametric model with joint limit avoidance. Compared to iterative methods, analytic solutions have lower computational cost and induce very little latency, which is crucial in VR applications. Since we did not want to limit ourselves to a specific VR applications, we ruled out data-driven methods, as data collection would be too expensive. The specific target on the human upper body allows us to use a simple kinematics chain that can easily be solved analytically and allows us to avoid the problems that can arise in iterative and data-driven methods. We solve remaining degrees of freedom within the analytic solution space using observation based heuristics, including thresholds for rotations and interpolation between different parameters that influence the specific joints' motion. Our solution is easy to implement and does not need an algorithm that is able to solve generic kinematics chains.

During our research, we only found a single upper-body IK solution which only depends on head and hand poses. Jiang et al. [2016] created a full-body avatar for VR applications which uses IK for upper body animation. Full-body avatars must accommodate a large variety of body poses and therefore need an extensive set of constraints. For example, they use a state memory of the previous frames to detect if the user is standing or crouching. Upper body bending is restricted while standing, and waist pose updates are restricted while crouching. A further limitation of their solution is that the shoulder depends on the head forward vector and is only updated if the head had a low average velocity during the last frames. Our method represents the kinematic chain from shoulder

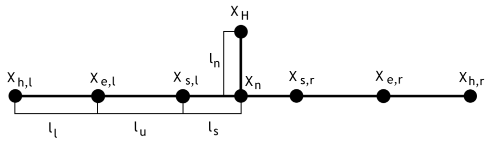

Figure 1: The kinematics chain consists of the head X_H, neck X_n and, for each arm, a shoulder X_s, elbow X_e and hand X_h, which are at a calibrated distance l_u and l_l, the upper and lower arm lengths.

to hand and requires fewer constraints. This allows us to use two stateless IK system, one per arm, with per-frame updates.

3 INVERSE KINEMATICS SOLVER

We are using a simple kinematics chain between head and hands, as shown in Figure 1. Our parametric solution starts at the head and follows the chain towards the hands. The IK solver relies on the knowledge of arm length and body size (height), which needs to be calibrated. Such calibration can for example follow the work of Han et al. [2016]. This method requires relatively accurate movements of the user. Instead we simplify the calibration with a single standing T-pose as this pose is also used for the calibration of the motion capture system which we compare against later. The distance between neck and head is fixed to $l_n = 0.13$ m and between left and right shoulder fixed to $2l_s = 0.31$ m. The upper and lower arm length l_u and l_l are assumed to be equal and calculated using the distance between the hands as

$$l_l = l_u = \frac{||\overrightarrow{X_{h,l}X_{h,r}}|| - 2l_s}{4}. \tag{1}$$

The vertical distance of the HMD to the ground plane is denoted by h_0. We utilize intrinsic *Tait-Bryan angles* with yaw α, pitch β and roll γ for joint rotations, since they allow axis-specific rotation limitation.

3.1 Neck joint

The neck joint is the approximate center of rotation of the head. Based on this assumption, we use a fixed offset in local HMD coordinates to compute the neck's position. Therefore, the position only depends on the position and orientation of the HMD. In order to avoid complex calibration steps for the user, predefined offset vectors are used to connect the HMD with the neck and the neck with the shoulders.

We estimate the neck's world space orientation based on the pose of the HMD for pitch β_n and the position of the motion controllers relative to the HMD for yaw α_n. Due to the complexity of roll estimation and the fact that it is not as important as yaw and pitch in VR applications, the neck's roll is assumed to be zero. The pitch is assumed to mostly depend on the distance h of the HMD to the ground, i.e., whether the user is standing upright ($h = h_0$), and the pitch of the HMD β_H. The smaller the distance to the ground and the more the HMD is looking down, the more the user's chest is assumed to be bent forward. Since these two factors are dependent,

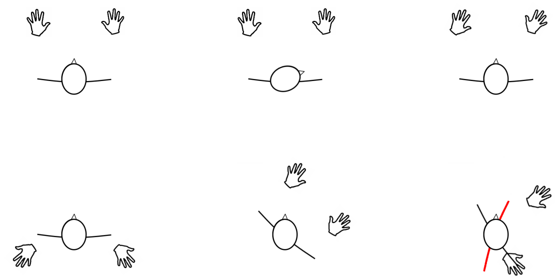

Figure 2: Different hand and head poses and their corresponding neck orientation. The hand position gives the most reliable hint for the neck forward vector. The red line in the bottom right picture illustrates the clamping process relative to the head orientation.

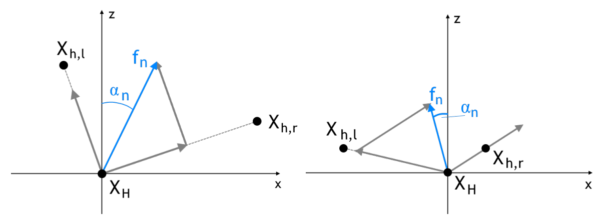

Figure 3: The neck's yaw α_n is calculated using the sum of the normalized directions from head to hands.

we use a multiplicative function to compute β_n as

$$\beta_n = \frac{h_0 - h}{h_0} \cdot (\beta_{n,0} + b \cdot \beta_H)\,, \tag{2}$$

where $\beta_{n,0} = 135.3°$ and $b = 0.333$ are used as weights.

The yaw estimation requires special attention, since it has a large impact on the final accuracy of the IK solver. Deriving the yaw from the orientation of the HMD would entail heavy shoulder movement when the user is looking left or right (Figure 2). However, most VR applications are not designed such that user are frequently required to change the yaw of the body. Thus, we derive the yaw from the sum of the normalized directions from the HMD to the motion controllers, as shown in Figure 3. This heuristic ensures that the shoulders remain relatively stable during head rotations, at the expense of small yaw errors when moving the hands. The measured yaw of the HMD is only used for disambiguation when both hands are placed behind the torso and to prevent the yaw difference between head and neck from exceeding $\pm 90°$.

3.2 Shoulder joint

The movement of the shoulder relative to the neck is typically small and mostly happens when the arm is fully extended already to reach a bit further. We define the shoulders' neutral positions $X_{s,n}$ as a simple translation by l_s along the neck's side directions. If the distance between neutral shoulder $X_{s,n}$ and hand X_h exceeds a threshold, it is rotated towards the hand by changing yaw α_s and roll γ_s of the shoulder. Figure 4 shows this for the yaw, which is

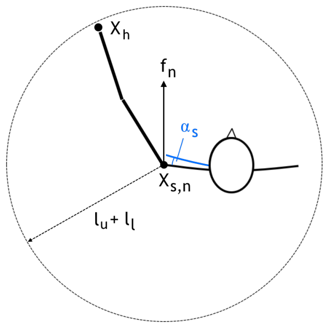

Figure 4: The shoulder rotation α_s is calculated using the ratio between the shoulder-to-hand distance and the arm length.

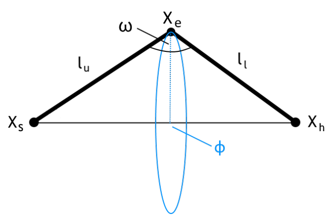

Figure 5: Simplified illustration of angles which have to be determined with the arm IK solver. The blue circle illustrates all possible elbow positions.

determined based on the neck's forward unit vector $\hat{f}_n$ as

$$\alpha_s = c\left(\frac{\overrightarrow{X_{s,n}X_h}^T \cdot \hat{f}_n}{l_u + l_l} - d\right), \tag{3}$$

where $c = 30°$ is a scaling constant and $d = 0.5$ a threshold before which no rotation occurs. The resulting value is clamped between 0 and 33°. The same equation is used for the roll γ_s with the difference of using the neck's up unit vector $\hat{u}_n$ instead of $\hat{f}_n$. This enables the upper arm anchor to move forwards and upwards.

3.3 Elbow joint

Given the position of shoulder and hand, the elbow can easily be positioned by computing the inner angle ω of the elbow using the cosine rule, as can be seen in Figure 5. The general solution of the elbow joint is a circle on the plane normal to the shoulder-hand axis and the center on that axis. Therefore, the difficult part of solving the elbow joint is finding the direction in which the elbow should be oriented. Previous work follows different strategies, typically with the target to simply produce a natural looking pose, while we target to determine the orientation as accurately as possible. For example, Yonemoto et al. [2000] determined that a fixed elbow orientation based on evaluating a motion capture dataset is sufficient for a natural looking pose. Kallmann [2008] uses an iterative method to avoid angle limits and collisions. Our computation of the elbow direction does not require iteration and is based on three heuristics that we apply in three consecutive steps:

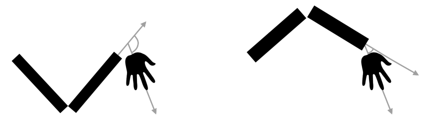

Figure 6: If the angle between hand and lower arm exceeds a threshold, the elbow is rotated to correct for the resulting unrealistic pose.

(1) We compute a model based on the hand position relative to the shoulder, since we found this to be the most important influence on elbow positioning when the hand is in front of the body.
(2) We apply corrections for positions close to the neck's up vector and behind the shoulder, so that unnaturally fast rotations of the arm are avoided.
(3) We apply corrections when the joint limit at the wrist is exceeded, so that unnatural rotations of the wrist are avoided.

Elbow rotation from relative hand position. Basic heuristics concerning elbow pose are easy to define and implement: The elbow should always point away from the body center, and it should be pointing backwards when the hand is in front of the shoulder. Unfortunately, these heuristics still leave a range of around 180° on which the elbow can move. One of the strongest indicators for the elbow angle is the hand position in local coordinates of the shoulder, X_h^s. To choose a solution on the circle, we define the angle ϕ to be zero if the elbow points in the direction of the up-vector of the neck. As ϕ increases, the elbow first points outwards and then downwards. We use a function loosely inspired by neural networks to compute ϕ as

$$\phi = \phi_0 + \sum_i \max\left(0, X_{h,i}^s \cdot w_i + b_i\right), \tag{4}$$

with biases $b = [30\ 120\ 65]$, weights $w = [-50\ -60\ 260]$ and a fixed offset angle $\phi_0 = 15°$. Afterwards, ϕ is clamped to stay within a given range of 13° to 175°. The parameters are retrieved by observation and error minimization on a set of sample arm poses.

Elbow rotation singularity correction. When calculating a swivel angle of the elbow instead of a target direction vector, problems occur when the hand is vertically aligned with (i.e., beneath or above) the shoulder. As the hand is very close to the shoulder's up-down-axis, small movements around it can result in a 360° rotation of the arm around the shoulder. Starting from a threshold distance of 0.5 m of the hand to the shoulder's vertical axis, we linearly blend the elbow direction vector $\vec{v}_\phi$ resulting from ϕ of the last step with a fixed one $\vec{v}_f = [0.133\ -0.443\ -0.886]$ and update ϕ with the result. We found that applying the same blending as the hand moves behind the shoulder also increases IK quality, blending within the range 0 to 0.1 m on the forward axis.

Elbow rotation from wrist rotation. The last step is to bring the arm into a relaxed position that avoids exceeding joint limits at the wrist. Unrealistically large wrist rotations can be corrected by rotating the elbow in a direction which reduces the wrist rotation as shown in Figure 6. Again, Tait-Bryan angles come in handy here, since they allow us to easily apply different constraints on the different axes. For example, if the yaw of the hand in the elbow's coordinate system α_h^e exceeds an upper threshold α_u^e then the elbow is corrected depending on how much the angle exceeds the threshold by

$$\Delta\phi = c_{\alpha,u}\left(\alpha_h^e - \alpha_u^e\right)^2 . \tag{5}$$

We use $c_{\alpha,u}$ as a scaling constant and a quadratic function to accelerate the correction. The same formula is used with different angles, scaling constants and thresholds and results are applied to ϕ additively. The threshold must be chosen high enough so that the correction is not noticeable when the user is only rotating a hand in-place, yet, it should be sensitive enough to prevent unrealistic rotations. We chose thresholds for yaw $\alpha_{u/l}^e$ as $\pm 45°$ with scaling constants $c_{\alpha,u/l} = \pm(135°)^{-1}$. For roll we chose a lower threshold $\gamma_l^e = 0°$ with a scaling constant of $c_{\gamma,l} = -(600°)^{-1}$ and an upper threshold $\gamma_u^e = 90°$ with $c_{\gamma,u} = (300°)^{-1}$.

4 COMPARISON WITH OTHER IK SOLVERS

To validate the quality of our IK solver, we compare it to approaches from different solution categories. As a baseline we use motion capture datasets with different types of motions which provide varying levels of difficulty to the solvers. As comparison methods we chose a general purpose Forward And Backward Reaching Inverse Kinematics (FABRIK) [Aristidou and Lasenby 2011] iterative solver, the solution by Jiang et al. [2016] which is specifically made for our application scenario, and one commercially available solution (Final IK [RootMotion [n. d.]]) as well as one open source solution (SAFullBodyIK [StereoArts [n. d.]]) that are both specifically designed for humans.

4.1 Methods

FABRIK is an iterative IK solver which is able to track multiple targets at once. Thus, opposite to other iterative solvers, like cyclic coordinate descent, which are designed for serial chains, both hands can be solved for at once in FABRIK. In comparison to other solutions which use rotation angles and matrices for solving, FABRIK tries to find joint positions via locating points on a line. This leads to visually realistic poses in few iterations and low computational cost. We use FABRIK on the full skeleton from Figure 1 with corresponding joint angle limits for all joints.

Jiang et al. [2016] created a full body IK system for VR that only requires head and hand poses as input. Their upper body IK solution combines the head and hand position on the horizontal plane for neck forward direction calculation. The waist is always on the same horizontal position as the head if the user is in the standing state. If the user is looking and moving downwards, the system switches into the crouch state in which the waist position is fixed on the horizontal plane and the upper body starts to bend. As their solution is not publicly available and they do not provide specific parameters, we reimplemented their solution and tuned parameters to give best results. However, we were unable to achieve reasonable results for neck or shoulder placement. It is not clear whether this fact stems from the approach itself or our implementation. As a remedy, we provide their solver with ground truth shoulder location and only test their arm IK. It uses a fixed pole vector as elbow target direction.

Similarly, Final IK and SAFullBodyIK lack an IK solver for the chain from head to shoulder. Thus, we also provide them with the ground truth shoulder location, again greatly reducing the difficulty. Both solvers use their parametric models based on the hand position given in local shoulder coordinates X_h^s to determine the elbow positioning. Final IK uses the unit vector $\hat{v}_h^s$ pointing from the shoulder to X_h^s in a spherical linear interpolation to combine a table of list elbow directions for different stored shoulder-hand-unit-vectors using the dot product as interpolation weight. SAFullBodyIK does not only use the direction, but also the distance to the hand in a set of heuristic linear interpolations to find the elbow rotation angle ϕ. Neither method makes use of the hand orientation to avoid unnatural bends at the wrist.

4.2 Results and Discussion

As test sets for comparison, we use publicly available motion capture data. We select four datasets which cover a variety of motions likely to appear in VR games:

- Unity Raw Mocap (URM) [Technologies [n. d.]]: standing, slow walking, little interaction
- Basic Motion (BM) [3D-Brothers [n. d.]a]: fighting, crouching, drinking, pulling, pushing, sitting
- Mixed Motion (MM) [3D-Brothers [n. d.]b]: baseball, ninja poses, crawling
- Mixed Motion 2 (MM2) [3D-Brothers [n. d.]c]: bodybuilding, golf, American football

Note that since our target are consumer-grade VR games, these datasets do not contain more extreme motions, such as lying down, sprinting, jumping or back flips.

For each joint, we obtain the root means squared error (RMSE) between each joint's ground truth position and the IK solution. The results of the comparison are given in Table 1.

Compared to the FABRIK solver, our specialized solver results in lower errors except for the neck joint in the MM and MM2 datasets. Since the neck joint is not part of the displayed arm in VR, this is negligible. The error of the shoulder joints is consistently at least 20 % lower than the error of FABRIK For the elbow this percentage increases to at least 36 % or 10 cm, which is probably caused by FABRIK just avoiding joint limits and using no further knowledge of typical elbow directions.

Since the solution by Jiang et al., Final IK and SAFullBodyIK use the ground truth shoulder position, we expect better results for the elbow than for the approaches that compute the complete chain. This is the case for the datasets containing more diverse motions (MM and MM2), but notably the elbow error is just about two centimeters higher in our method even though the shoulder error is above ten centimeters, indicating that our methods works very well overall. At lower shoulder errors, we achieve similar (BM) or even lower (URM) elbow errors than these methods although we estimate the complete chain. Between these methods the errors are very similar and each method has the lowest elbow error in one of the three more difficult datasets, suggesting that the methods all have similar quality.

Table 1: RMSE in cm of neck, shoulder and elbow positions for four motion capture data sets. Our solution and FABRIK can compute the entire IK chain, while Jiang et. al, Final IK and SAFullBodyIK, use the ground truth shoulders and compute the arm IK only.

Anim	Joint	Ours	FABRIK	Jiang	Final IK	SAFBIK
URM	neck	3.4	4.9	-	-	-
	shoulder	3.9	4.9	-	-	-
	elbow	4.6	15.7	5.4	6.2	5.9
BM	neck	8.9	10.0	-	-	-
	shoulder	10.4	14.1	-	-	-
	elbow	15.0	27.4	15.6	15.6	13.9
MM	neck	9.0	8.7	-	-	-
	shoulder	13.3	19.7	-	-	-
	elbow	21.7	34.3	19.1	17.9	20.0
MM2	neck	9.4	8.1	-	-	-
	shoulder	12.8	16.3	-	-	-
	elbow	17.8	29.6	15.8	16.0	16.2

Unity Raw Mocap is the easiest of all datasets containing only standing, slow walking and simple interactions. 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.

5 USER STUDY

The results in section 4.2 are promising and show that an optimized upper body IK system for VR can generate reasonable accuracy for a wide set of motions which are likely encountered in VR applications. However, to determine how an IK system performs in real consumer-grade VR environments, we conduct a user study. The goal of our user study is twofold. First, we want to determine whether the proposed IK system performs well enough to allow users to complete tasks where the arms are required. Second, we want to determine whether adding arms to the virtual avatar improves the feeling of embodiment. To this end, we formulate the following hypotheses

- **H1** Using our upper body IK system achieves equally good results as a full motion capturing system.
- **H2** Displaying well-behaved arms in VR increases the user's feeling of embodiment.
- **H3** Having the choice between well-behaved arms and no-arms solutions, users prefer solutions with arms.

5.1 Study Design

To test our hypotheses, we use a within-subject design and split the study into two tasks. In the first task, *goalie*, users must use their hands and arms for interaction. Colored balls, originating from a distance, are moving towards the participants, who can block the balls only with the matching body part (Figure 7). The second task, *archery*, does not require arms for playing the game. The participants need to use a virtual bow to shoot at randomly placed targets. Every hit advances the game. This task places arms prominently and thus they might effect the experience, even though they are not needed for interaction (Figure 8).

Figure 7: Left: The goalie game from the perspective of the user. Right: a user playing the goalie game with fast arm movement.

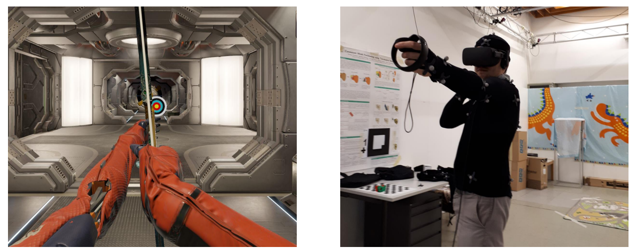

Figure 8: Left: The archery game from the perspective of the user. Right: a user playing the archery game.

We test three conditions: *hand-only*, *motion capture arms*, and *IK arms*. *hand-only* uses the controllers of the VR setup for hand positions and displays only hands. *motion capture arms* uses a full motion capture suit to track the entire body. We use the motions capture results for hand, elbow and shoulder placement to render the entire arm. *IK arms* receives the tracked HMD and VR motion controller locations as input and applies our IK solution to compute shoulder and elbow locations for arm rendering. To prevent users from guessing different modes, the motion capture suit and VR motion controllers are worn throughout the whole study.

We use the following hardware and software for the user study:

- Game engine: Unity 2017.3.1f1
- VR system: Oculus Rift CV1
- Motion Capture: Optitrack Motive 2.0 with six cameras

In both tasks, the order of conditions is randomized. As *goalie* requires arms, the hand-only condition does not make sense and thus the first task is performed only with the other two conditions. Before each task and condition, the participants are asked to practice in a short tutorial round. To setup our IK system and fix all parameters as outlined alongside the description of our IK solution, we conducted a short pre-study.

5.2 Measurements and Questionnaires

The main study starts with a simple demographic questionnaire. During both tasks, we track the performance of each participant by recording successful hits of balls in the goalie game and shots on the target in the archery game. After each condition in both tasks, we ask participants to answer a questionnaire with seven (goalie) and

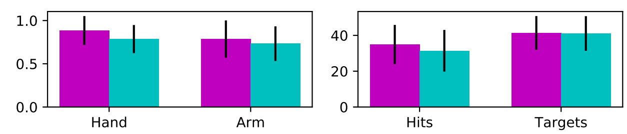

Figure 9: Left: Hit ratio of hands and arm in the goalie game. Right: Number of hits and targets in the goalie game in both modes. ■ IK, ■ Motion Capture

eight (archery) questions that are answered on a six-point Likert scale.

The questions for both games address embodiment, controllability, accuracy, confidence, difficulty, mental load, and subjective performance. The archery game questions additionally cover world scale and presence.

After completing a task in all conditions, participants are asked to indicate their preference between all conditions concerning *embodiment* ("Which method leads to the highest feeling of embodiment"), *fast games* ("Which method do you prefer for playing fast games"), and *overall* ("Which method do you prefer overall").

After completing both tasks, participants are placed in an environment where they can move freely without completing any task and switch through the three conditions. After trying all conditions, participants are asked to select the mode which achieves the strongest feeling of *embodiment* ("Select the mode which achieves the strongest feeling of having your own arm in VR").

6 RESULTS

We initially recruited 76 participants for the user study, of those 21 participants were excluded, resulting in 55 overall participants. Reasons for exclusion were impaired vision that could not be corrected with the HMD, colorblindness, and inability to adjust the motion capturing suit to their body size. 25 % of the participants were female, 93 % were right-handed, and 82 % studied computer science. Among the chosen subjects, 52 % had impaired vision, but either used contact lenses or could adjust the VR HMD to achieve good vision. On a scale from 1 (not at all) to 6 (very experienced), participants rated their experience in VR with an average of 2.47 with a standard deviation (STD) of 1.39.

All tests for statistical significants were calculated using Welch's t-test when comparing result pairs and Spearman's rank to find linear correlations between two characteristics. Kruskal-Wallis H-test and Chi-Squared "Goodness of Fit" test were used for significance testing if more than two groups are compared and to evaluate the significance of single choice questions. For post-hoc testing we apply Bonferroni adjustment. We use a p-value borderline of 0.05 for accepting or rejecting the null hypothesis. In the goalie and archery games, the order in which the methods were played was randomized to counteract learning effects.

6.1 Goalie Game

On average, participants hit 139 out of 165 targets correctly using *IK arms*, and, 126 using motion *motion capture arms*, which is a significant difference ($t(85.9) = 2.67$, $p < .01$). Figure 9 shows the distribution of the hand and arm hit ratios. It can be observed that the hit ratio of the hands and arms is higher and more consistent with *IK arms* than with *motion capture arms* (hands: $t(177.7) = 3.83$, $p < .001$; arms: $t(188.3) = 2.15$, $p < .05$). The results also show a correlation in the consistency of the scores per player with a p-value of < .05. Thus, a player who achieved a high score in one mode was slightly more likely to achieve a high score in the second mode.

The questionnaire results are summarized in Figure 10. We observe a statistically significant difference in embodiment ($t(95.0) = 2.14$, $p < .05$), controllability ($t(78.7) = 2.06$, $p < .05$), accuracy ($t(99.4) = 2.73$, $p < .01$), and difficulty ($t(80.8) = 2.71$, $p < .01$). There was no difference in confidence ($t(107.9) = -0.62$, $p > .5$), mental load ($t(106.9) = -0.77$, $p > .4$), and subjective performance ($t(78.5) = 1.46$, $p > .1$).

The post questionnaire showed that participants preferred *IK arms* over *motion captured arms* concerning embodiment (67 % $\chi^2(1) = 5.9$, $p < .05$), for *fast games* (69 % $\chi^2(1) = 7.28$, $p < .01$), and *overall* (69 % $\chi^2(1) = 7.28$, $p < .01$), see Figure 11.

The sum of results in the goalie game give a clear picture. While both *IK arms* and *motion capture arms* received strongly positive feedback, participants still rate *IK arms* significantly better and also achieve better results with this approach. These facts not only confirm hypothesis **H1**—that our IK system achieves equally good results as a motion capture system—but even surpasses the motion capture results, especially in terms of embodiment, controllability, and accuracy.

This result seems surprising at first, as we considered motion capturing as the ground truth for our study. Although many participants thought that both conditions are indistinguishable "Were the modes the same?", some participants commented that motion capturing felt a bit "sluggish" or "slightly slower". An in-depth analysis of the motion capture data shows that the captured data is accurate and there seem to be no tracking issues. However, due to the technical setup, which ran the motion capturing on a dedicated server, *motion capture arms* introduced an additional delay due to network transmission. Precisely measuring the network delay turned out to be difficult, but we were able to narrow it down to approximately three frames of the HMD (22 ms to 33 ms). In contrast, the delay introduced by the motion controllers is a single frame. In some cases, when Optitrack has difficulties to track the markers, the delay can increase by another two to three frames. We believe this delay to be the reason for the slightly better performance of IK over motion capturing, especially in a game that requires very fast motion.

It should be noted that this situation has its roots purely in a technical limitation and a motion capturing system with less delay would likely improve the results. Nevertheless, the results show that our IK system leads to very plausible and believable results, even when accurate and very responsive arm positions are required in the VR application. This is also underlined by the high questionnaire results for controllability (4.93/6) and accuracy (5.11/6).

After playing the game in both modes, the participants were asked to select which modes they preferred (see Figure 11). 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.

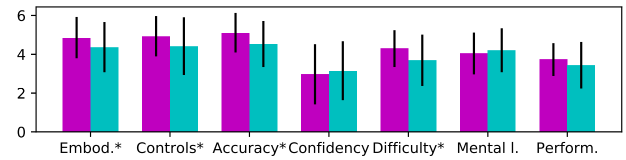

Figure 10: Questionnaire results in the goalie game. * marks questions with significant differences (p < 0.05). ■ IK arms, ■ Motion Capture arms

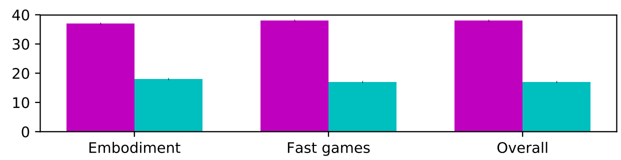

Figure 11: Questionnaire after playing the goalie game in both modes. The results of all questions are of statistical significance (p < 0.05). ■ IK, ■ Motion Capture

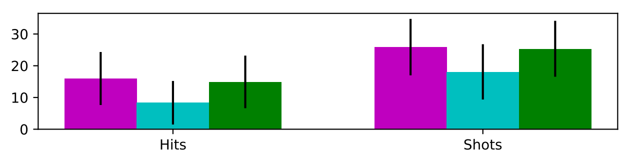

Figure 12: Number of hits and shots in the archery game in the three modes. ■ IK, ■ Motion Capture, ■ Hand only

6.2 Archery Game

In the archery game, clear differences of the conditions can be observed. The hits per participant varied strongly, as shown in Figure 12. With *IK arms*, players hit 15.98 targets on average, 14.91 with *hand-only* and 8.35 with *motion capture arms*. The difference between *IK arms* and *hand-only* is insignificant ($t(108.0) = .67, p > .5$). The difference between *IK arms* and *motion capture arms* ($t(94.7) = 5.2$, $p < .001$), and between *hand-only* and *motion capture arms* ($t(95.6) = -4.5$, $p < .001$) are both very strong. 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 (*IK arms* and *hand-only* $correlation(53) = .70$ $p < .001$, *IK arms* and *motion capture arms* $correlation(53) = .50$ $p < .001$, *hand-only motion capture arms* $correlation(53) = .54$ $p < .001$).

The questionnaire results are summarized in Figure 13 with the measured statistical values given in Table 2. The results show that mental load was perceived the same in all conditions. For embodiment, presence, quick arms, world scale, controllability, difficulty, and performance there were differences. The differences among those questions all indicated that both *IK arms* and *hand-only* were always rated significantly better than *motion capture arms*. While *IK arms* was on average rated higher than *hand-only*, none of these differences was significant after Bonferroni adjustment.

The post-questionnaire of the archery game is summarized in Figure 14. 71 % of the participants stated that *IK arms* led to the highest feeling of embodiment, 26 % selected *hand-only* and 2 persons, or 3 %, selected *motion capture arms* ($\chi^2(2) = 38.88$, $p < .001$). For playing fast games in VR, 69 % prefer *IK arms*, 27 % *hand-only* and 3 % *motion capture arms* ($\chi^2(2) = 36.26$, $p < .001$). Overall, 71 % prefer *IK arms*, 23 % *hand-only* and 6 % *motion capture arms* ($\chi^2(2) = 37.68$, $p < .001$).

Before analyzing the individual results, it should be noted that *IK arms* and *hand-only* achieved similar results, while *motion capture arms* is significantly worse. While the delay of the motion capture system can explain some degeneration in performance, the archery game reveals another issue with motion capture. Archery requires much more precision in hand position for aiming than the goalie game, where ±5 cm do not make a big difference. A limitation of our motion capture setup is the difficulty to track the arms when the participants grabbed the line of the bow and pulled it closely to their chest. In these cases, our six camera setup was not sufficient to accurately track the string-pulling arm due to occlusion from the torso, and the skeleton tracking started shaking. These artifacts made it difficult to aim and decreased the quality of the experience. In contrast, the visio-inertial technology of the consumer-grade VR system is not noticeable affected in this situation.

The questionnaires administered between conditions indicate that *IK arms* and *hand-only* (and thus the display of arms) do not lead to differences in the feeling of embodiment for a task which does not require arms. However, when arms are displayed that do not match the body the feeling of embodiment can be ruined, leading to *motion capture arms* being rated significantly worse. The same is true for the feeling of presence. Similarly, seeing an arm did not help in the estimation of scale and distance, for controlling the bow, and it did not make the task easier. Again, a non-accurate arm reduces these abilities.

However, when looking at the post-experience questionnaire, where participants had to choose one mode that achieves the highest feeling of embodiment, is best for fast games, and gives them the best overall feeling, they chose *IK arms* over the other modes. These overall results partially support **H2**, showing that there are situations where displaying well-behaved arms can increase the feeling of embodiment. However, when users are not focused on arms, their added benefit might not be apparent. With the preference for choosing *IK arms*, **H3** is clearly supported; giving users the choice between well-behaved arms or none, they choose having arms, even if they are not needed for the task. At the same time, ill-behaved arms are considered a strong disturbance, and users prefer not to display arms at all if they cannot reliably be estimated.

6.3 Post-Questionnaire and Additional Results

The results of the final test, which allowed participants to test all three conditions freely, is shown in Figure 14. Statistically analysis shows that these results are significant ($\chi^2(2) = 13.35, p < .005$), with a significant difference between *IK arms* and *hand-only* ($\chi^2(1) = 11.6$, $p < .001$) and non-significant differences between *hand-only* and *motion capture arms* ($\chi^2(1) = 2.56$, $p > .1$), *IK*

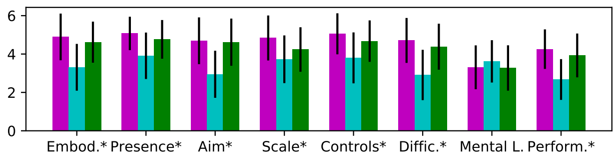

Figure 13: Questionnaire results in the archery game. * marks questions with significant differences (p < 0.05). ■ IK, ■ Motion Capture, ■ Hand only

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

Question	IK - MC - Hand		IK - MC		IK - Hand		MC - Hand	
	H	p	t	p	t	p	t	p
embodiment	46.02	0.000	6.75	0.000	1.24	0.656	-5.93	0.000
presence	33.71	0.000	5.74	0.000	1.70	0.274	-3.99	0.000
quick arms	50.05	0.000	7.41	0.000	0.31	2.274	-7.08	0.000
wolrd scale	22.13	0.000	4.77	0.000	2.67	0.026	-2.20	0.090
controllabilty	27.89	0.000	5.41	0.000	1.85	0.203	-3.75	0.001
difficullty	45.70	0.000	1.44	0.000	1.44	0.462	-6.09	0.000
mental load	3.02	0.221	0.16	0.467	0.16	2.614	1.57	0.358
performance	47.15	0.000	1.56	0.000	1.56	0.363	-5.91	0.000

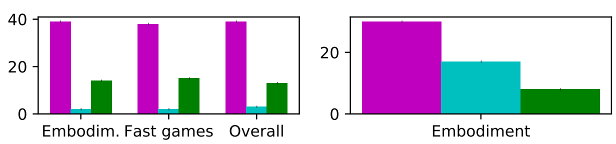

Figure 14: Left: Questionnaire after playing the archery game in both modes. Right: Post-Questionnaire. The results of all questions are of statistical significance (p < 0.05). ■ IK, ■ Motion Capture, ■ Hand only

arms and *motion capture arms* $\chi^2(1) = 3.06, p > .05$). This result contrasts the results from the archery game mode. First it can be observed that when users calmly evaluate their virtual body and are presented with working arms they choose arms over *hand-only*, again underlining **H3** and strengthening our considerations for **H2**. The different result between *IK arms* and *motion capture arms* compared to archery game can be explained by some participants trying similar movements to the ones they just performed during the archery game while others did not. The ones who reenacted archery experienced tracking issues and chose *IK arms*. Those who performed more general movements did not experience differences between both conditions and thus chose either.

It should also be noted that we did not find a connection between VR experience and scores in any task. Also, the participants' performances were not consistent between the goalie and the archery games.

6.4 Discussion and Summary

Our IK solution achieved accurate and believable arm motion for both tasks, which is underlined by the general high feedback for embodiment, controllability and accuracy. Due to the lower delay of our IK solution compared to motion capturing, our IK solution achieved even better objective and subjective results. Furthermore, even a motion capture setup may not be sufficient when movements lead to too many occlusions in the camera setup. An IK solution avoids most of these issues at no additional costs, achieving the best overall performance. At least, we deem **H1** as fully supported.

Compared to hands-only, there is indication that arms increase the feeling of embodiment. However, when they are secondary to the interaction, they may not increase that feeling. Thus, we deem **H2** only partially supported. We also confirmed that, when body parts are not consistent with the users movement, they will reduce embodiment and all other experience-related measures. Finally, when given the choice, participants clearly chose arm support over hands only, and thus **H3** is also supported. Overall, upper body IK seems an obvious choice for consumer-grade VR systems.

7 CONCLUSION

In this work we created an IK solver for human arms optimized for consumer-grade VR setups which only use motion controller and headset positions. We have shown that our IK solution performs better than solvers which are not optimized for this specific use case. Our study has shown that our approach can be used for generating realistic and responsive arms in VR. The IK solution also proved to be indistinguishable from a ground truth motion capturing considering joint positioning. In fact, our IK solution shows less delay and never suffers from tracking errors due to occlusion and thus even performed better than motion capturing.

Our IK solution allows to integrate arms into VR applications and thus enhance the interaction capabilities with the environment. However, when arms are not required for a task, they may not necessarily improve the experience, in particular, when users are concentrating on a demanding task. When user are in calm environment or actively compare a well-behaved arms solution to a hands-only mode, they clearly prefer having arms. During the user study, we could also confirm that displaying arms that do not match ones real body strongly deteriorate the experience.

While our solution is general and works surprisingly well for simple VR interactions, there is certainly potential to consider more involved motions and additional priors. Having knowledge about the performed action or pose, e.g., sitting, walking, or playing sports, IK parameters could be altered to give a more likely solution. Integrating background knowledge about a person could also allow better prediction of their motion without adding additional sensors. For example, age, weight, or sex might be parameters that allow for a better prediction. Of course, adding additional sensors would allow adding further body parts, like feet or legs. We believe it might even be possible to add torso animations without adding sensors, especially since the shoulder is already estimated for the animation of the arms.

Considering the outstanding quantitative results of our IK solution compared to other available approaches as well as the qualitative feedback, we suggest our approach universally for consumer-grade VR applications. It is publicly available at https://github.com/dabeschte/VRArmIK.

ACKNOWLEDGMENTS

We would like to thank Eva Haring for her help with the user study. This work was partially funded by the German Research Foundation (DFG) grant STE 2565/1-1, the Austrian Science Fund (FWF) grant I 3007, the Austrian Science Fund (FWF) grant KLI 678-B31, and FFG grant 859208.

REFERENCES

3D-Brothers. [n. d.]a. Basic Motion. https://assetstore.unity.com/packages/3d/animations/basic-motion-25834. Accessed on Aug 15, 2018.

3D-Brothers. [n. d.]b. Mixed Motion. https://assetstore.unity.com/packages/3d/animations/mixed-motion-25837. Accessed on Aug 15, 2018.

3D-Brothers. [n. d.]c. Mixed Motion 2. https://assetstore.unity.com/packages/3d/animations/mixed-motion-2-25844. Accessed on Aug 15, 2018.

Marjan A Admiraal, Martijn J M A M Kusters, and Stan C. A. M. Gielen. 2004. Modeling kinematics and dynamics of human arm movements. *Motor control* 8 3 (2004), 312–38.

Andreas Aristidou and Joan Lasenby. 2011. FABRIK: A fast, iterative solver for the Inverse Kinematics problem. *Graphical Models* 73, 5 (sep 2011), 243–260. https://doi.org/10.1016/j.gmod.2011.05.003

Andreas Aristidou, Joan Lasenby, Yiorgos Chrysanthou, and Ariel Shamir. 2018. Inverse Kinematics Techniques in Computer Graphics: A Survey. In *Computer Graphics Forum*, Vol. 37. Wiley Online Library, 35–58.

Panagiotis K. Artemiadis, Pantelis T. Katsiaris, and Kostas J. Kyriakopoulos. 2010. A biomimetic approach to inverse kinematics for a redundant robot arm. *Autonomous Robots* 29, 3-4 (jun 2010), 293–308. https://doi.org/10.1007/s10514-010-9196-x

T. Asfour and R. Dillmann. [n. d.]. Human-like motion of a humanoid robot arm based on a closed-form solution of the inverse kinematics problem. In *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003) (Cat. No.03CH37453)*. IEEE. https://doi.org/10.1109/iros.2003.1248841

Frank Biocca. 1997. The cyborg's dilemma: Progressive embodiment in virtual environments. *Journal of computer-mediated communication* 3, 2 (1997), JCMC324.

Matthew Botvinick and Jonathan Cohen. 1998. Rubber hands âĂŸfeelâĂŹtouch that eyes see. *Nature* 391, 6669 (1998), 756.

Carlos Faria, Flora Ferreira, Wolfram Erlhagen, SĂľrgio Monteiro, and Estela Bicho. 2018. Position-based kinematics for 7-DoF serial manipulators with global configuration control, joint limit and singularity avoidance. *Mechanism and Machine Theory* 121 (mar 2018), 317–334. https://doi.org/10.1016/j.mechmachtheory.2017.10.025

C.C.A.M. Gielen, E. J. Vrijenhoek, T. Flash, and S.F.W. Neggers. 1997. Arm Position Constraints During Pointing and Reaching in 3-D Space. *Journal of Neurophysiology* 78, 2 (aug 1997), 660–673. https://doi.org/10.1152/jn.1997.78.2.660

Stan Gielen. 2009. Review of Models for the Generation of Multi-Joint Movements in 3-D. In *Advances in Experimental Medicine and Biology*. Springer US, 523–550. https://doi.org/10.1007/978-0-387-77064-2_28

Ping-Hsuan Han, Kuan-Wen Chen, Chen-Hsin Hsieh, Yu-Jie Huang, and Yi-Ping Hung. 2016. Ar-arm: Augmented visualization for guiding arm movement in the first-person perspective. In *Proceedings of the 7th Augmented Human International Conference 2016*. ACM, 31.

Fan Jiang, Xubo Yang, and Lele Feng. 2016. Real-time full-body motion reconstruction and recognition for off-the-shelf VR devices. In *Proceedings of the 15th ACM SIGGRAPH Conference on Virtual-Reality Continuum and Its Applications in Industry - VRCAI '16*. Association for Computing Machinery (ACM). https://doi.org/10.1145/3013971.3013987 search: inverse kinematics head mounted display neural networks only for action determination, not for IK.

Marcelo Kallmann. 2008. Analytical inverse kinematics with body posture control. *Computer animation and virtual worlds* 19, 2 (2008), 79–91.

Tao Kang, Jiping He, and Stephen I. Helms Tillery. 2005. Determining natural arm configuration along a reaching trajectory. *Experimental Brain Research* 167, 3 (oct 2005), 352–361. https://doi.org/10.1007/s00221-005-0039-5

Konstantina Kilteni, Raphaela Groten, and Mel Slater. 2012. The sense of embodiment in virtual reality. *Presence: Teleoperators and Virtual Environments* 21, 4 (2012), 373–387.

Hyunchul Kim and Jacob Rosen. 2015. Predicting Redundancy of a 7 DOF Upper Limb Exoskeleton Toward Improved Transparency between Human and Robot. *Journal of Intelligent & Robotic Systems* 80, S1 (feb 2015), 99–119. https://doi.org/10.1007/s10846-015-0212-4

Elena Kokkinara and Mel Slater. 2014. Measuring the effects through time of the influence of visuomotor and visuotactile synchronous stimulation on a virtual body ownership illusion. *Perception* 43, 1 (2014), 43–58.

Koichi Kondo. 1994. *Inverse Kinematics of a Human Arm.* Technical Report. Stanford, CA, USA.

Dongik Lee and Hankyu Lim. 2015. Virtual Reality Contents using the OculusLift and Kinect. In *Proceedings of the MCSI*. 102–105.

Tzu-Chen Liang and Jing-Sin Liu. [n. d.]. An improved trajectory planner for redundant manipulators in constrained workspace. In *Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No.99CH36288C)*. IEEE. https://doi.org/10.1109/robot.1999.774078

Richard P Paul. 1981. *Robot manipulators: mathematics, programming, and control: the computer control of robot manipulators*. Richard Paul.

Valeria I. Petkova and H. Henrik Ehrsson. 2008. If I Were You: Perceptual Illusion of Body Swapping. *PLoS ONE* 3, 12 (dec 2008), e3832. https://doi.org/10.1371/journal.pone.0003832

RootMotion. [n. d.]. Final IK. https://assetstore.unity.com/packages/tools/animation/final-ik-14290. Accessed on Aug 15, 2018.

Daniel Roth, Jean-Luc Lugrin, Julia BĂijser, Gary Bente, Arnulph Fuhrmann, and Marc Erich Latoschik. 2016. A simplified inverse kinematic approach for embodied vr applications. In *Virtual Reality (VR), 2016 IEEE*. IEEE, 275–276.

Ulrike Schultze. 2010. Embodiment and presence in virtual worlds: a review. *Journal of Information Technology* 25, 4 (dec 2010), 434–449. https://doi.org/10.1057/jit.2010.25

Mel Slater. 2008. Towards a digital body: The virtual arm illusion. *Frontiers in Human Neuroscience* 2 (2008). https://doi.org/10.3389/neuro.09.006.2008

Mel Slater, Bernhard Spanlang, Maria V Sanchez-Vives, and Olaf Blanke. 2010. First person experience of body transfer in virtual reality. *PloS one* 5, 5 (2010), e10564.

J. F. Soechting and M. Flanders. 1989. Errors in pointing are due to approximations in sensorimotor transformations. *Journal of Neurophysiology* 62, 2 (aug 1989), 595–608. https://doi.org/10.1152/jn.1989.62.2.595

Bernhard Spanlang, Jean-Marie Normand, David Borland, Konstantina Kilteni, Elias Giannopoulos, AusiĂăs PomĂľs, Mar GonzĂąlez-Franco, Daniel Perez-Marcos, Jorge Arroyo-Palacios, Xavi Navarro Muncunill, and Mel Slater. 2014. How to Build an Embodiment Lab: Achieving Body Representation Illusions in Virtual Reality. *Frontiers in Robotics and AI* 1 (nov 2014). https://doi.org/10.3389/frobt.2014.00009

Bernhard Spanlang, Jean-Marie Normand, Elias Giannopoulos, and Mel Slater. 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. https://doi.org/10.1145/1889863.1889870

Anthony Steed, Sebastian Frlston, Maria Murcia Lopez, Jason Drummond, Ye Pan, and David Swapp. 2016. An 'In the Wild' Experiment on Presence and Embodiment using Consumer Virtual Reality Equipment. *IEEE Transactions on Visualization and Computer Graphics* 22, 4 (apr 2016), 1406–1414. https://doi.org/10.1109/tvcg.2016.2518135 search: virtual reality embodiment motion capture.

StereoArts. [n. d.]. SAFullBodyIK. https://github.com/Stereoarts/SAFullBodyIK. Accessed on Aug 15, 2018.

Zhipeng Tan, Yuning Hu, and Kun Xu. 2017. Virtual Reality Based Immersive Telepresence System for Remote Conversation and Collaboration. In *Next Generation Computer Animation Techniques*, Jian Chang, Jian Jun Zhang, Nadia Magnenat Thalmann, Shi-Min Hu, Ruofeng Tong, and Wencheng Wang (Eds.). Springer International Publishing, Cham, 234–247.

Unity Technologies. [n. d.]. Raw Mocap Data for Mecanim. https://assetstore.unity.com/packages/3d/animations/raw-mocap-data-for-mecanim-5330. Accessed on Aug 15, 2018.

Xuguang Wang. 1999. A behavior-based inverse kinematics algorithm to predict arm prehension postures for computer-aided ergonomic evaluation. *Journal of Biomechanics* 32, 5 (may 1999), 453–460. https://doi.org/10.1016/s0021-9290(99)00023-8

Satoshi Yonemoto, Daisaku Arita, and Rin-ichiro Taniguchi. 2000. Real-time human motion analysis and IK-based human figure control. In *humo*. IEEE, 149.