P. Caserman, P. Achenbach, and S. Göbel, “Analysis of Inverse Kinemat- ics Solutions for Full-Body Reconstruction in Virtual Reality,” in 2019 IEEE 7th International Conference on Serious Games and Applications for Health (SeGAH), Aug 2019, pp. 1–8.

Analysis of Inverse Kinematics Solutions for Full-Body Reconstruction in Virtual Reality

Polona Caserman
Multimedia Communications Lab
Technische Universität Darmstadt
Darmstadt, Germany
polona.caserman@kom.tu-darmstadt.de

Philipp Achenbach
Multimedia Communications Lab
Technische Universität Darmstadt
Darmstadt, Germany
philipp.achenbach@kom.tu-darmstadt.de

Stefan Göbel
Multimedia Communications Lab
Technische Universität Darmstadt
Darmstadt, Germany
stefan.goebel@kom.tu-darmstadt.de

***Abstract*—Many Virtual Reality (VR) applications usually visualize only the VR controllers or floating hands. However, to create an immersive experience, a full-body avatar is essential. We reconstruct a full-body avatar by tracking the position and orientation of the head, hands, feet, and hip. To track the arm movements, the user has to hold two HTC Vive controllers. Additionally, the user has to bind Vive trackers to both ankles and the hip. We apply some of the most popular Inverse Kinematics (IK) methods to estimate the full-body pose. We perform parameter optimization to analyze the damping constant, the maximum number of iterations, and error value for the position as well as rotation. We made several tests between the IK methods in terms of the accuracy and the time to solve the IK problem. The results show that Damped Least Squares (DLS) outperforms the other methods. We furthermore conducted a user study to evaluate the subjective quality of the DLS method. Evaluation results show that the motion reconstruction for lower-body is very accurate; however, for the upper-body, some inaccuracies can occur. Such a motion reconstruction approach can be used in VR exergames, e.g., users can learn different poses while observing the movements in a virtual mirror and by looking down towards their own body.**

***Index Terms*—Full-Body Motion Reconstruction, Immersive Virtual Reality, Inverse Kinematics, Joint Configuration**

I. Introduction

Virtual Reality (VR) is becoming increasingly important in many areas, e.g., exergames [1], exposure therapy [2], and for training/simulation [3]. Most VR applications aim to create an immersive virtual environment to give the user a sensation of “being there.” As Slater and Wilbur [4] point out, immersion requires a self-representation in the virtual environment, thus a virtual body. Moreover, to improve presence, the user in VR must identify herself/himself with this virtual body. The focus on presence and immersion requires to create a connection between a user and a virtual environment as well as a user and a virtual body. Therefore, it is crucial to track the movements of the user and to visualize a full-body avatar in VR.

Body pose estimation can be done using different devices. Several studies focus on using a Microsoft Kinect to reconstruct a body pose [1]. The main advantage of this device is that the user does not need to bind any sensors to the body. However, a single camera often fails to capture non-frontal poses, suffers from occlusion, and provides insufficient data. Therefore, some researchers combine a depth camera with inertial measurement units [5] or even use inertial sensors without external cameras [6]. A far more effective motion capture technology is based on small retro-reflective markers, which are usually attached to the motion capture suit. Commercial solutions such as OptiTrack or Vicon are very accurate and precise, but also very expensive. Previous works have shown that such a system is very suitable to visualize a full-body avatar in multiplayer VR applications [7].

All motion tracking systems have their advantages and disadvantages. The Kinect is limited either by its latency of insufficient accuracy. Expensive and accurate motion capture suits often depend on many sensors and can create discomfort. We aim to minimize the number of sensors attached to the user’s body while still accurately and reliably reconstructing movements. We use a HTC Vive HMD to track head movements. To track the arms, the user has to hold one Vive controller in each hand. Additionally, the user has to bind one Vive tracker to each foot and to the hip. Since the position and orientation of these six joints are known, we can solve the Inverse Kinematics (IK) problem to estimate the full-body pose. The major contributions of our work include the following:

- Reducing the number of sensors to enable full-body motion reconstruction
- Analyzing different IK solutions regarding computation time and accuracy
- Applying parameter optimization to optimize and validate the IK parameters, such as damping constant, maximum error (for position and rotation), and a maximum number of iterations
- Evaluating the subjective quality of the IK method

II. Related Work

A. Inverse Kinematics Methods

The main problem in IK is to choose appropriate joint values (e.g., the angle of a rotational joint) so that the end-effector reaches the desired target position and orientation [8]. To solve the IK problem, we can use different methods, e.g., analytic or iterative methods. Systems with simple structure and low number of Degree of Freedom (DoF) can be solved

TABLE I: Angular restrictions for each joint.

Joint	x-axis		y-axis		z-axis	
	Min	Max	Min	Max	Min	Max
Wrist	$-55°$	$45°$	/	/	$-85°$	$75°$
Elbow	$0°$	$155°$	/	/	/	/
Shoulder	$-65°$	$195°$	$-105°$	$105°$	$-145°$	$105°$
Knee	$0°$	$155°$	/	/	/	/
Hip	$-145°$	$45°$	$-75°$	$55°$	$-65°$	$65°$

analytically. However, due to complex geometry for long kinetic chains, the analytical methods are not suitable for computer animations [9].

Systems with a complex structure are usually solved numerically. The most popular methods are based on the Jacobian matrix [8], [10]. Jacobian matrix methods aim to find a linear approximation of the problem to move the end-effector to the desired target gradually. Many different methods have been proposed to calculate the Jacobian Inverse, such as *Jacobian Transpose* (JT), *Jacobian Pseudo-Inverse* (JPI), *Damped Least Squares* (DLS), *Singular Value Decomposition* (SVD), *Damped Least Squares with Singular Value Decomposition* (SVD-DLS), and *Selectively Damped Least Squares* (SDLS). Jacobi methods usually provide very accurate results. However, complex mathematical calculations lead to higher computational costs, especially when the matrix is large [11]. Furthermore, methods such as JPI and SVD suffer from singularity problems [8], [12].

Full-Body Motion Reconstruction using Inverse Kinematics

Previous works have shown promising results using different motion capture systems for full-body motion reconstruction, e.g., using only a HTC Vive HMD and two controllers [13]. Likewise, recent commercial approaches, such as *Orion IKinema*[1] or *DeepMotion*[2] use Vive devices strapped to the limbs to reconstruct full-body avatar in real-time.

Aristidou and Lasenby [14] propose a novel method, called FABRIK. Even though the method converges in a few iterations and yields visually realistic poses, their approach does not take the desired rotation into account. Bentrah et al. [15] suggest an extension of the FABRIK and minimize the number of moving joints to reach targets and can handle environmental obstacles. Similar to FABRIK, Meredith et al. [9] reduce computational costs by using only half Jacobian and thus ignore the orientation of the end-effector.

Furthermore, Kenwright et al. [10] propose a realistic, robust, and computationally fast method of solving the IK problem using the *Gauss-Seidel* iterative method. Moreover, Unzueta et al. [16] describe a Sequential IK solver. The researchers solve the IK problem sequentially, using a simple analytic-iterative IK algorithm in different parts of the body in a specific order. Their approach can also prevent self-collisions

[1]https://ikinema.com/orion, retrieved March 4th, 2019

[2]https://www.deepmotion.com, retrieved March 4th, 2019

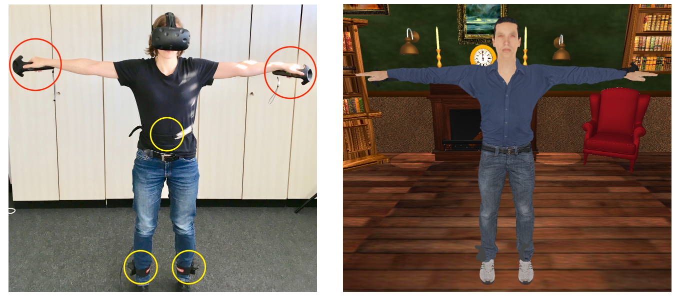


Fig. 1: Calibration process: A student is standing in a T-pose and holding two Vive controllers (red), while three trackers are bound to the feet and hip (yellow).

in posture reconstruction, e.g., the penetration of the elbows in the torso.

III. Body Model

We design a realistic virtual avatar with 32 bones using the *MakeHuman*[3] tool to visualize full-body motions in VR. To obtain a natural pose, we define for each joint in the IK chain the DoF. We specify the upper and lower limits for these joints, as suggested by Hamilton et al. [17] and furthermore modify the joint limits by adding a tolerance of $\pm 15°$ (see Table I). The skeleton consists of four kinematic chains: wrist, elbow, and shoulder for both arms (2×6 DoF) as well as knee and hip for both legs (2×4 DoF).

Skeleton Calibration

We apply skeleton calibration so that motion reconstruction works independently of the body height and the position/rotation of the attached VR devices. For the calibration, the user has to stand in a T-pose as depicted in Figure 1.

First, while the user is standing in the T-pose, we obtain the raw position and rotation (quaternion) of the Vive trackers and controllers and transform them into the local coordinate system of the avatar. Then, we determine the offset position and rotation for each end-effector. We calculate the distance between the desired position/quaternion ($\mathbf{p}_{\text{des}}$, $\mathbf{q}_{\text{des}} \in \mathbb{R}^{4\times 1}$) and the actual position/quaternion ($\mathbf{p}_{\text{act}}$, $\mathbf{q}_{\text{act}} \in \mathbb{R}^{4\times 1}$):

$$\mathbf{p}_{\text{offset}} = (\mathbf{T}(\mathbf{p}_{\text{des}}[t']) \cdot \mathbf{R}(\mathbf{q}_{\text{act}}[t']))^{-1} \cdot \mathbf{p}_{\text{act}}[t'], \tag{1}$$

$$\mathbf{q}_{\text{offset}} = (\mathbf{q}_{\text{des}}[t'])^{-1} \cdot \mathbf{q}_{\text{act}}[t'], \tag{2}$$

where $\mathbf{T} \in \mathbb{R}^{4\times 4}$ is a transformation and $\mathbf{R} \in \mathbb{R}^{4\times 4}$ a rotation matrix at time step t' (at the moment of the calibration). Thus, offset values ($\mathbf{p}_{\text{offset}}$, $\mathbf{q}_{\text{offset}} \in \mathbb{R}^{4\times 1}$) are calculated only once.

Finally, in each further frame, we calculate the final position $\mathbf{p}_{\text{final}}[t] \in \mathbb{R}^{4\times 1}$ and quaternion $\mathbf{q}_{\text{final}}[t] \in \mathbb{R}^{4\times 1}$:

$$\mathbf{q}_{\text{final}}[t] = \mathbf{q}_{\text{des}}[t] \cdot \mathbf{q}_{\text{offset}}, \tag{3}$$

$$\mathbf{p}_{\text{final}}[t] = \mathbf{T}(\mathbf{p}_{\text{des}}[t]) \cdot \mathbf{R}(\mathbf{q}_{\text{final}}) \cdot \mathbf{p}_{\text{offset}}, \tag{4}$$

where $\mathbf{p}_{\text{des}}[t]$ and $\mathbf{q}_{\text{des}}[t]$ are the position and quaternion values at each time step t with respect to the avatar local coordinate system.

[3]http://www.makehumancommunity.org, retrieved March 4th, 2019

TABLE II: The value range of the λ.

	Value range	Step size
JT	$[0.05, 1]$	0.05
JPI	$[0.05, 1]$	0.05
DLS	$[0, 1.5]$	0.05
SVD	$[0.01, 0.25]$	0.01
SVD-DLS	$[0, 1.5]$	0.05
SDLS	$[0.002, 0.052]$	0.002

IV. Applied Methods

We apply the most popular IK solutions, such as JT, JPI, DLS, SVD, SVD-DLS, and SDLS. To achieve maximum performance, we use the low-level game API and hardware abstraction library *Kore*.[4] In addition, we provide (c++) source code to researchers.[5]

For motion reconstruction in VR, the IK method has to be fast, so that the user does not perceive delays. The IK approach must solve the problem for multiple end-effectors in real-time and also has to represent realistic movements. There are mainly three challenges:

1) The pose must look natural.
2) The IK method must reconstruct the pose in real-time.
3) The IK method must solve the problem for multiple end-effectors.

Firstly, to ensure natural movements, we specify the upper and lower limit for each joint, as already mentioned in Section III. However, even with the constraints, several solutions can still exist. We choose the pose that is close to the previous pose. For example, some IK methods, such as SDLS, define the maximum angle change to keep the difference between the two poses small.

Secondly, we have to ensure low latency so that the user in VR can interact through the avatar in real-time. If the latency is too high, the user perceives the movement of the virtual avatar delayed, which decreases immersion. However, sometimes the IK solver cannot determine a pose, e.g., when the end-effector cannot reach the target position or when the algorithm is stuck in a local extremum. In this case, we define conditions at which the IK method terminates, i.e., a maximum number of iterations and maximum position as well as rotation error. Thus, the algorithm terminates when the end-effectors are sufficiently close to the targets or when the algorithm reaches the maximum number of iterations.

Finally, the IK method must reconstruct the full-body pose of multiple end-effectors, i.e., hands and feet. To reduce the size of the Jacobian matrix, we use an approach presented in our previous work [18]. For each end-effector, we consider at most two predecessor joints. For example, for the hand to reach the target, we rotate the wrist, elbow, and shoulder. For the leg, we rotate only the knee and hip. Thus, in our approach, each joint is influenced by at most one end-effector. The end-effectors are therefore independent of each other and can be considered separately. With this reduced Jacobian matrices, our approach can easily handle multiple end-effectors to reconstruct a full-body avatar in real-time.

[4]https://github.com/Kode/Kore, retrieved retrieved 4th, 2019

[5]https://github.com/CatCuddler/BodyTracking, retrieved March 4th, 2019

TABLE III: An overview of the optimized parameter values.

	JT	JPI	DLS	SVD	SVD-DLS	SDLS
λ	0.35	0.05	0.2	0.03	0.2	0.018
e_{maxPos} [m]	0.01	0.1	0.001	0.01	0.001	0.01
e_{maxRot} [rad]	0.01	0.1	0.01	0.01	0.01	0.01
it_{max}	10	100	20	10	20	60

Analysis tool for parameter optimization

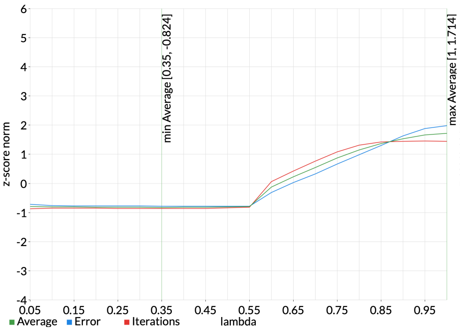
The accuracy is defined as the distance between the target and actual end-effector. We achieve good accuracy when the error in position and rotation is small. The speed is defined as the average time which is required to reach the desired position as accurately as possible. Thus, time is dependent on the number of iterations and time per iteration.

We collected raw data (position vector and quaternion) for each VR device while five participants walked in a circle, performed squats, and ran in place. For the parameter optimization, we use this dataset and accordingly change the parameters, e.g., λ, maximum error for position e_{maxPos} and rotation e_{maxRot}, and maximum number of iterations it_{max}. The λ value has different functionality for different IK methods, e.g., a damping constant (JT, JPI, DLS, and SVD-DLS), a threshold (SVD) or it specifies maximum angle change (SDLS). We define the interval of the λ value for each IK method separately (see Table II). Furthermore, as proposed by Buss and Kim [8], we search for the optimal e_{maxPos} and e_{maxRot} in the range of 0.0001 and 0.1 and increase values exponentially in each step. For optimal it_{max}, we search in the range of 10 and 250 and increase the value by 10 in each step.

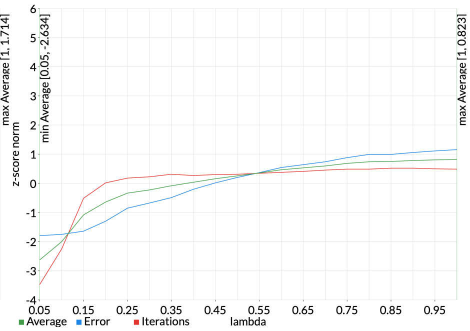
After we obtain the values for all parameters, we standardize the results. We want to optimize the values regarding the accuracy (minimum average error) and the time needed to reach the target (minimum average iteration number). However, the parameter that minimizes the time does not necessarily also minimizes the error. To this end, we use the *z-score* to transform normal variates to standard score form [19]. The *z-score transformation* allows us to compare the parameter values along each other. To get the optimal value for each parameter, we cross-validate the values between different motions (walking, performing squats, running). We search for the smallest error with the shortest calculation time and therefore, the minimum average. Thus, we choose the parameter value that gives us the lowest cross-validation average error.

V. Parameter optimization results

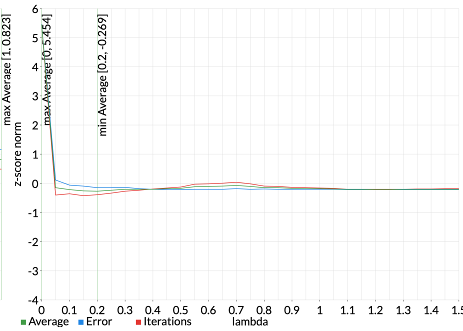
The parameter optimization aims to achieve an optimal parameter configuration for each IK method, which ensures



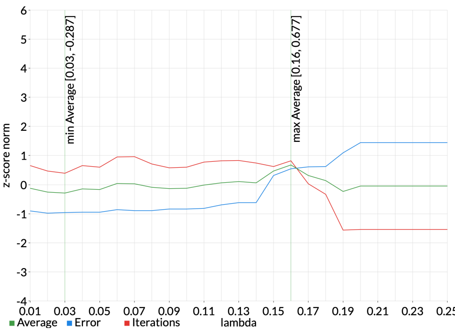
(a) JT



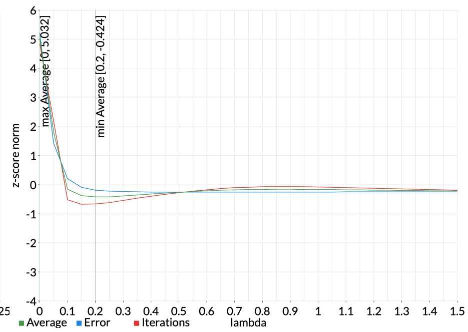
(b) JPI



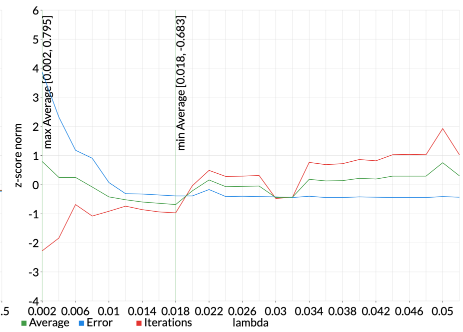
(c) DLS



(d) SVD



(e) SVD-DLS



(f) SDLS

Fig. 2: Optimizing the λ value regarding the error and number of iterations.

the most accurate possible result with the shortest possible computing time. The optimal parameter values for each IK method after cross-validation are specified in Table III.

A. Optimal λ value

Figure 2 shows the optimal λ value (*minAverage*) for each IK method. The λ depends on the error (blue) and the number of iterations (red). The JT and the JPI methods both use the λ value as damping constant to keep the calculated change in joints small. For JT, the error and the number of iterations increase rapidly with $\lambda > 0.55$ (see Figure 2a). For JPI, with increasing λ value, the error continually increases while the number of iterations converges (see Figure 2b). Both methods get unstable with a larger damping constant. JT achieves the optimal result with $\lambda = 0.35$ and JPI with $\lambda = 0.05$.

The parameter optimization shows that $\lambda = 0.2$ is the optimal value for the DLS and SVD-DLS. In Figure 2c and 2e, we can observe that for very small values $\lambda < 0.1$, the methods become unstable and both, the error and the number of iterations, increase. The SVD uses the λ value as a threshold. The parameter optimization reveals that the method performs best for $\lambda = 0.03$. With $\lambda = 0.16$, the error increases significantly and the number of iterations decreases (see Figure 2d). Furthermore, with $\lambda > 0.2$, the number of iterations and the error remain constant.

The advantage of SDLS over other IK methods is that we do not need to specify the damping constant. The only parameter to set is the λ_{max}. It defines the maximum possible change in any joint angle. Our results show that SDLS performs best with $\lambda_{max} = 0.018$.

B. Optimal e_{maxPos} and e_{maxRot}

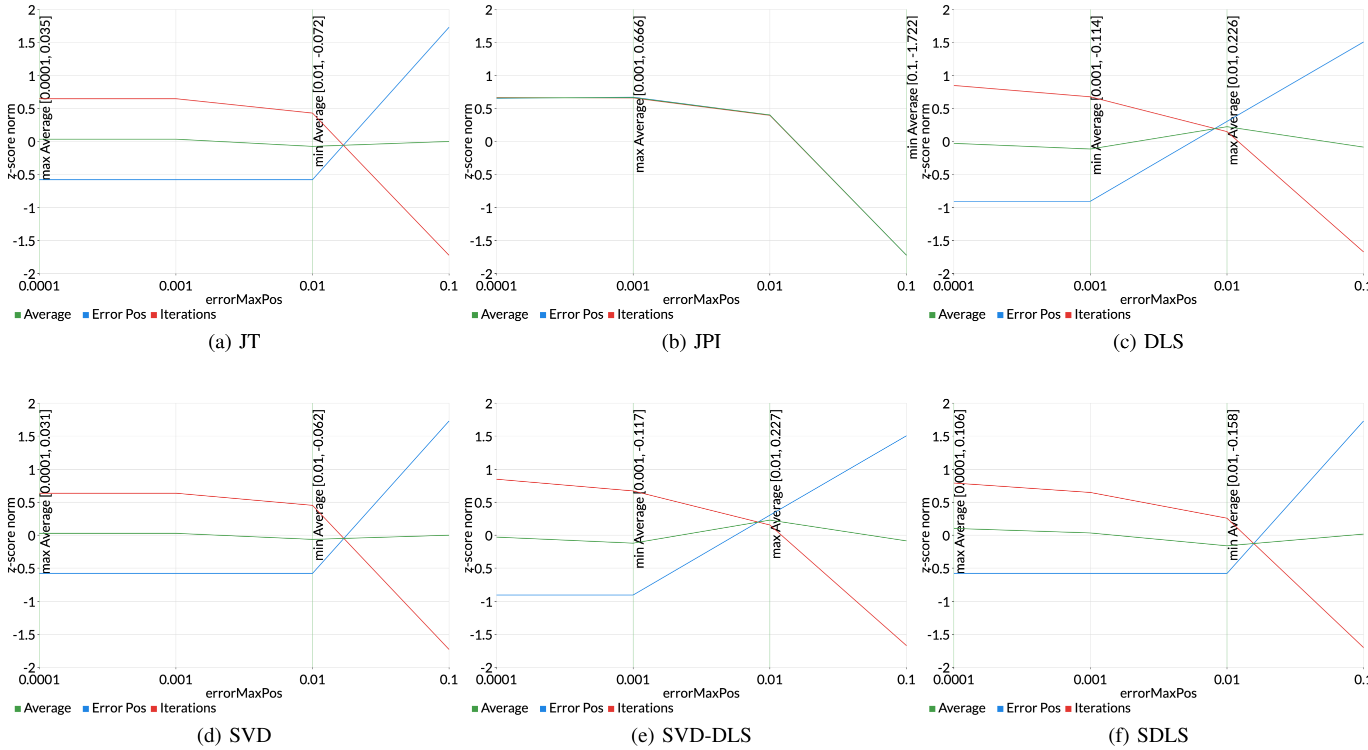
Maximum position e_{maxPos} and rotation e_{maxRot} error determine the accuracy. On the one hand, a small error contributes to more accurate results. However, as the accuracy increases, so does the number of iterations and thus the computation time. On the other hand, a large error leads to a faster result, but with less accuracy.

Figure 3 shows that for all IK methods (except for the JPI) with the increased position error $e_{maxPos} \to 0.1$ (blue), the number of iterations (red) decreases. The results for the rotation error are analogous to the position error. As expected, with increasing accuracy, the number of iterations will also increase. Especially in the range of 0.01 and 0.1, the number of iterations drops rapidly while the error increases.

The optimal position value for the JT, the SVD, and the SDLS methods is $e_{maxPos} = 0.01$ and for DLS as well as SVD-DLS $e_{maxPos} = 0.001$. For all methods (except for JPI) the optimal value for e_{maxRot} is 0.01. The results for the JPI are contradictory. With increased maximum error, the number of iterations decreases. We will discuss this instability problem in Section VI.

C. Optimal it_{max} value

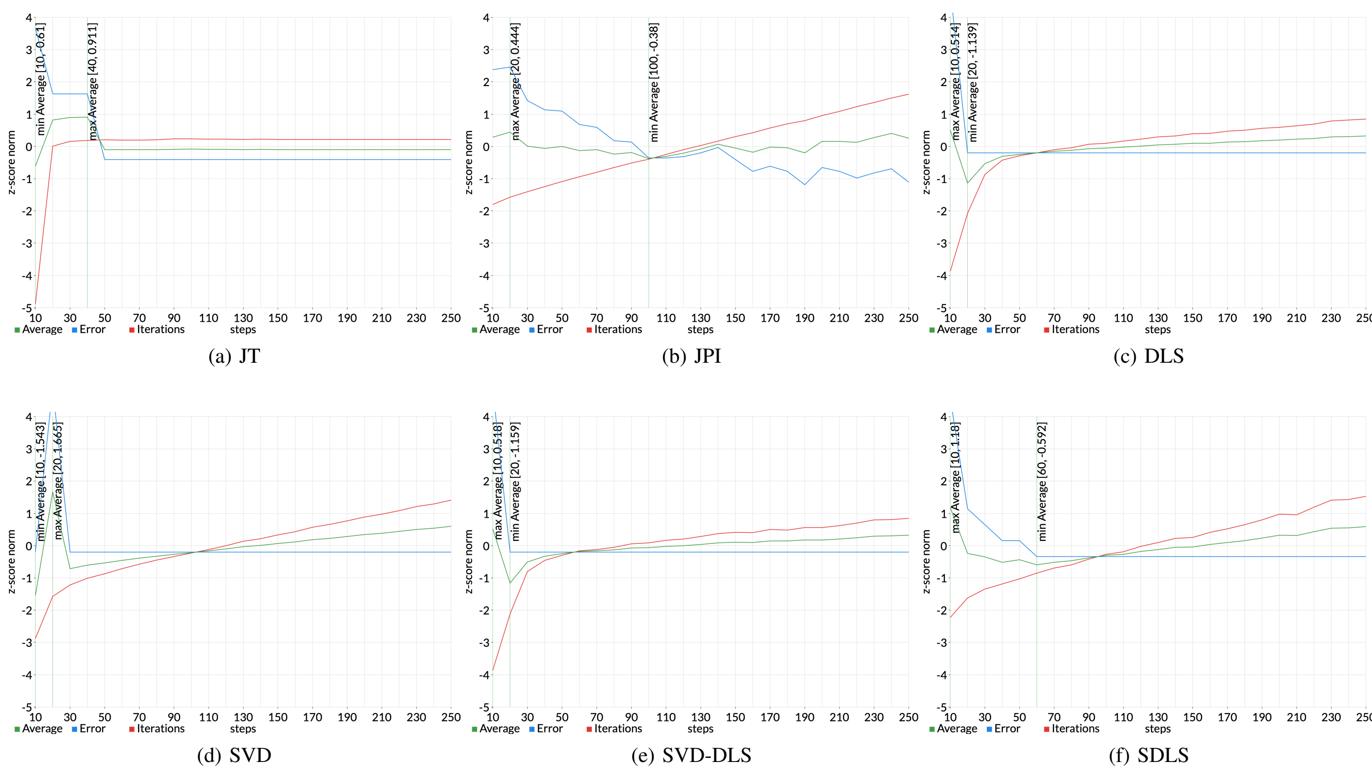
As shown in Figure 4, for most IK methods, the number of iterations (red) increases while the error (blue) decreases. Only the JPI and the SVD method show some unstable results.



(a) JT (b) JPI (c) DLS

(d) SVD (e) SVD-DLS (f) SDLS

Fig. 3: Optimizing the maximum position error regarding the number of iterations.



(a) JT (b) JPI (c) DLS

(d) SVD (e) SVD-DLS (f) SDLS

Fig. 4: Optimizing the number of iterations regarding the position and rotation error.

TABLE IV: Reconstruction comparison in terms of cost.

	it_{num}	t_{it} [μs]	t_{total} [μs]
JT	7.917	14.465	114.519
JPI	32.75	17.17	562.318
DLS	6.511	17.824	116.052
SVD	5.689	29.083	165.336
SVD-DLS	6.505	28.829	187.533
SDLS	7.717	27.122	209.3

TABLE V: Reconstruction comparison in terms of error.

IK	e_{pos} [mm]	e_{rot} [deg]
JT	86	1.318
JPI	190	25.898
DLS	76	3.724
SVD	81	3.037
SVD-DLS	76	3.724
SDLS	98	1.776

Observing the results for the JT (see Figure 4a), we can see that for $it_{max} > 50$ both, the number of iterations and the error, remain constant. We can obtain the optimal value regarding the error with less than ten iterations. As already mentioned before, JPI suffers from singularity problems and therefore, the total error shows unstable results (see Figure 4b). We achieve the best results with $it_{max} = 100$. Furthermore, there is a similarity between the DLS and the SVD-DLS method. For both methods, we need at least 20 iterations to achieve the best results. The SDLS converges much slower and requires at least 60 iterations to minimize the error.

D. Motion Reconstruction Costs

Motion reconstruction costs depend on the number of iterations and the time per iteration. Table IV summarizes the average number of iterations it_{num}, the time per iterations t_{it}, and the total time t_{total} required by each IK method.

JT is the most time efficient ($t_{total} = 114.52\ \mu s$); however, it needs many iterations ($it_{num} = 7.917$) and, thus, converges only slowly. Although DLS performs similarly and requires fewer iterations than JT ($it_{num} = 6.511$), it needs more time per iteration. Hence, DLS is the second fastest method ($t_{total} = 116.05\ \mu s$).

Even though the SVD needs the lowest number of iterations ($it_{num} = 5.689$), the method requires, due to the computationally intensive singular value decomposition, the longest time per iteration ($t_{it} = 29.083\ \mu s$). DLS and SVD-DLS need a similar number of iterations; however, SVD-DLS needs significantly more time per iteration. Likewise, the results suggest that SDLS is very computationally expensive. The SDLS method is, apart from JPI, with $t_{total} = 209.3\ \mu s$ the slowest.

E. Motion Reconstruction Accuracy

The position and rotation error define the accuracy of the motion reconstruction. We measure the accuracy by comparing the position and rotation of the end-effector with the actual position and rotation of the Vive devices. Thus, for each IK method, we calculate the difference between the desired and actual end-effector position as well as rotation.

In general, a small error will provide more accurate results; however, the number of iterations will increase. We use the optimal parameter values to calculate the average position and rotation error. The results are shown in Table V.

Observing the error, we can see that all IK methods, except for the JPI, have a similar error in position. In terms of position error, the DLS and the SVD-DLS methods perform best with $e_{pos} = 76\ mm$. However, in terms of rotation error, both methods perform badly with $e_{rot} = 3.724°$. Only the JPI method performs even worst and suffers from singularity problems.

For SDLS, we can observe exactly the opposite results. While the position error is very high ($e_{pos} = 98\ mm$), the rotation error remains very low ($e_{rot} = 1.776°$). Likewise, the JT also results in a large position error ($e_{pos} = 86\ mm$) but shows the smallest rotation error ($e_{rot} = 1.31°$).

VI. Parameter Optimization Discussion

From the results presented in Table IV and V, we can see that the JT performs best regarding speed ($t_{total} = 114.519\ \mu s$) and gives very accurate results in terms of orientation ($e_{rot} = 1.318°$). However, DLS method achieves a smaller position error ($e_{pos} = 76\ mm$) and needs only $1.5 \mu s$ more time to solve the IK problem. Even though the DLS method has a bigger rotation error, the user can hardly perceive a small difference of only a few degrees. Both approaches represent smooth and accurate movements. Nevertheless, we believe that the user recognizes the difference between the actual and the desired position more clearly than the change between the actual and the desired rotation. Therefore, we prefer the DLS over the JT method.

Figure 5 shows the results of the full-body reconstruction based on the DLS. We can reconstruct complex poses such as kneeling and kicking in real-time. This motion reconstruction can be beneficial for VR exergames, e.g., to learn a particular movement such as yoga poses. The users can observe the movements in a virtual mirror and by looking down towards their body. Another possible application scenario could be a serious game for training purposes or rehabilitation.

The supplementary video[6] for this paper shows the user performing some tasks in VR. We reconstruct the motions using the DLS with the optimal parameters. As one can see in the video, the results sometimes suffer in terms of smoothness, e.g., the elbow can snap to a new location. These inaccuracies can happen when the difference between the new joint orientation, compared to the joint orientation in the previous frame is too large. We could solve this problem by using the Slerp method and interpolating between the quaternions. Furthermore, to improve the results, we could

[6]https://youtu.be/x4SS8_-XY38, retrieved May 10th, 2019

Fig. 5: Examples of our real-time full-body motion reconstruction based on HTC Vive trackers and controllers.

use weights when optimizing error. In other words, we should focus on minimizing the position error rather than minimizing the rotation error. The experiments with subjects have also shown that position accuracy is more important than rotation accuracy (see Section VII).

Compared to Aristidou and Lasenby [14] with the median position error of 58.68 mm, we obtain for DLS and SVD-DLS a similar result (76 mm). However, they did not consider the rotation error. Our solution outperforms [20] (mean rotation error of $7.8°$) and [16] (average rotation error of $16.9°$) across the rotation error. We obtain a minimal rotation error of less than $4°$ for all IK methods, except for the JPI. Furthermore, for the JT, the SDLS, and the DLS methods, we need significantly fewer iterations than Buss and Kim [8] to reach the desired target.

Limitations: In this paper, we evaluated the IK methods in terms of error, i.e., distance to end-effector. We did not measure the position error for the elbow or knee. To calculate the accuracy of each joint, we would need to compare the estimated pose against a baseline. One possibility would be to bind further Vive trackers to the body and then calculate the error of the individual joints. In future work, we want to compare our results with ground truth. We believe that this will help us to optimize the parameters even better, which will lead to better results.

VII. User Study Results

We conducted a user study to measure the presence, agency, sense of body ownership, and end-to-end latency. The questions were adapted from Kilteni et al. [21] and can be seen in Table VI. We calculate the median (MED) and Inter-Quartile Range (IQR) for each question.

We recruited ten participants (six males and four females), with an age range between 23 and 31 (average age was 27.1 years). First, we presented the HTC Vive headset and bound the trackers to the body. Then, we explained the calibration process. After the avatar was calibrated, the participants could see an avatar while looking down towards their real body and in a virtual mirror. The results of the questionnaire assess the subjective quality of the optimized DLS.

The level of reported *presence* was very high, with a median score of 4.5 (IQR = 1.25). The results show that the participants felt as if they were really "being there" in the virtual reality. Furthermore, the median score of the *sense of body ownership* question was 3.75. The large IQR score of 3.125 suggests that the participants tend to hold strong opinions either for or against it. To understand the wide range of possible answers, we have to take a closer look at *arm* and *leg ownership*. The results for the *arm ownership* show a median score of 3.5 (IQR = 2) and for the *leg ownership*, a median score of 4.5 (IQR = 2.25). Thus, the participants scored the ownership for legs significantly higher than for the arms ($p = 0.002$). We believe that the results of large IQR for the overall *sense of body ownership* arise from the fact that the reconstruction of the upper-body sometimes fails while the reconstruction of the lower-body is very accurate.

Most participants agreed that the avatar size matched the real one (*estimation of body parts*) with a median score of 5 (IQR = 1). Furthermore, the participants rated both (similar) questions about the *sense of agency* very high (MED = 4). The results show that the participants were in control of the avatar and they could control the movements of the virtual body. Furthermore, the participants disagreed that the movements of the avatar seem to be another person's movement (MED = 2, IQR = 2.25). Thus, the low score for *full-body disconnection* supports the high score for the *sense of agency*. Moreover, the results show that the participants perceived only a small delay of the virtual avatar (MED = 2, IQR = 2.25).

VIII. Conclusion

In this paper, we analyzed many IK methods to find a suitable approach for full-body motion reconstruction in VR in terms of low latency and high accuracy. We optimized the damping constant, maximum error in position and rotation as well as the maximum number of iterations. The results show that both the JT and the DLS methods are very fast and

TABLE VI: Questionnaire responses.

Condition	**Questions** Each question was assessed on a 5-point Likert scale: 1 *Strongly Disagree* and 5 *Strongly Agree*.	MED	IQR
Presence	I felt like I was present in the virtual world.	4.5	1.25
Sense of body ownership	I felt as if the virtual body was my body.	4	3
	I had the feeling to look at my body when I looked down at myself.	3.5	3.25
Arm ownership	It seemed like the virtual arms belonged to me.	3.5	2
Leg ownership	It seemed like the virtual legs belonged to me.	4.5	2.25
Full-body disconnection	The movement of the hands and feet seemed to be another person's movement.	2	2.25
Estimation of body parts	It seemed like the size of the avatar matched the real one.	5	1
Sense of agency	It seemed like I was in the control of the virtual avatar.	4	2.25
	It felt as if I was controlling the movements of the virtual avatar.	4	1.25
Latency	It seemed like the movements of the virtual avatar were delayed.	2	2.25

provide only a small error. On the one hand, the results for the JT are more accurate in terms of orientation than position. On the other hand, the DLS can achieve a smaller position error and requires only slightly more time than the JT. We prefer the DLS over the JT because the user in VR perceives a large position error rather than rotation error. After the parameter optimization, we conducted a user study to evaluate the subjective quality of the DLS. The results show that the ownership for legs is significantly higher than for the arms. Moreover, the participants agreed that they could control the movements of the avatar. Thus, the results indicate that the DLS is suitable for full-body reconstruction in VR.

Future work will focus on improving accuracy. We want to minimize the position error rather than the rotation error. Furthermore, since the rotation for the elbow is sometimes inaccurate, we want to develop a new device with integrated IR sensors which can be attached to the user's forearm. Such a device would contribute to better results, as it would allow us to follow the rotation of the elbow more accurately.

Acknowledgment

This work has been co-funded by the German Federal Ministry of Education and Research (BMBF) within the framework of the Software Campus project "TargetVR" $[01IS17050]$.

References

[1] J. Bolton, M. Lambert, D. Lirette, and B. Unsworth, "PaperDude: A Virtual Reality Cycling Exergame," in *CHI '14 Extended Abstracts on Human Factors in Computing Systems*. ACM, 2014, pp. 475–478.

[2] P. Schäfer, M. Koller, J. Diemer, and G. Meixner, "Development and Evaluation of a Virtual Reality-System with Integrated Tracking of Extremities Under the Aspect of Acrophobia," in *2015 SAI Intelligent Systems Conference (IntelliSys)*. IEEE, 2015, pp. 408–417.

[3] C. Li, W. Liang, C. Quigley, Y. Zhao, and L. Yu, "Earthquake Safety Training through Virtual Drills," *IEEE Transactions on Visualization and Computer Graphics*, vol. 23, no. 4, pp. 1275–1284, 2017.

[4] M. Slater and S. Wilbur, "A Framework for Immersive Virtual Environments (FIVE): Speculations on the Role of Presence in Virtual Environments," *Presence: Teleoperators and Virtual Environments*, vol. 6, no. 6, pp. 603–616, 1997.

[5] T. Helten, M. Muller, H.-P. Seidel, and C. Theobalt, "Real-Time Body Tracking with One Depth Camera and Inertial Sensors," in *The IEEE International Conference on Computer Vision (ICCV)*, 2013.

[6] D. Roetenberg, H. Luinge, and P. Slycke, "Xsens MVN: Full 6DOF Human Motion Tracking Using Miniature Inertial Sensors," *Xsens Motion Technologies BV, Tech. Rep*, vol. 1, 2009.

[7] P. Bourdin, J. M. T. Sanahuja, C. C. Moya, P. Haggard, and M. Slater, "Persuading People in a Remote Destination to Sing by Beaming There," in *Proceedings of the 19th ACM Symposium on Virtual Reality Software and Technology*. ACM, 2013, pp. 123–132.

[8] S. R. Buss and J.-S. Kim, "Selectively Damped Least Squares for Inverse Kinematics," *Journal of Graphics Tools*, vol. 10, no. 3, pp. 37–49, 2005.

[9] M. Meredith and S. Maddock, "Real-time Inverse Kinematics: The return of the Jacobian," Technical Report No. CS-04-06, Department of Computer Science, University of Sheffield, Tech. Rep., 2004.

[10] B. Kenwright, "Real-time Character Inverse Kinematics Using the Gauss-Seidel Iterative Approximation Method," in *International Conference on Creative Content Technologies*, vol. 4, 2012, pp. 63–68.

[11] M. Fêdor, "Application of Inverse Kinematics for Skeleton Manipulation in Real-time," in *Proceedings of the 19th Spring Conference on Computer Graphics*. ACM, 2003, pp. 203–212.

[12] A. Aristidou, J. Lasenby, Y. Chrysanthou, and A. Shamir, "Inverse Kinematics Techniques in Computer Graphics: A Survey," *Computer Graphics Forum*, vol. 37, no. 6, pp. 35–58, 2018.

[13] F. Jiang, X. Yang, and L. Feng, "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 - Volume 1*. ACM, 2016, pp. 309–318.

[14] A. Aristidou and J. Lasenby, "FABRIK: A Fast, Iterative Solver for the Inverse Kinematics Problem," *Graphical Models*, vol. 73, no. 5, pp. 243 – 260, 2011.

[15] A. Bentrah, A. Djeffal, M. Babahenini, C. Gillet, P. Pudlo, and A. Taleb-Ahmed, "Full Body Adjustment Using Iterative Inverse Kinematic and Body Parts Correlation," in *Computational Science and Its Applications – ICCSA 2014*. Springer International Publishing, 2014, pp. 681–694.

[16] L. Unzueta, M. Peinado, R. Boulic, and A. Suescun, "Full-body Performance Animation with Sequential Inverse Kinematics," *Graphical Models*, vol. 70, no. 5, pp. 87 – 104, 2008.

[17] N. P. Hamilton, *Kinesiology: Scientific Basis of Human Motion*. Brown & Benchmark, 2011.

[18] P. Caserman, A. Garcia-Agundez, R. Konrad, S. Göbel, and R. Steinmetz, "Real-time Body Tracking in Virtual Reality Using a Vive Tracker," *Virtual Reality*, pp. 1–14, 2018.

[19] G. W. Milligan and M. C. Cooper, "A Study of Standardization of Variables in Cluster Analysis," *Journal of Classification*, vol. 5, no. 2, pp. 181–204, 1988.

[20] C. Malleson, A. Gilbert, M. Trumble, J. Collomosse, A. Hilton, and M. Volino, "Real-Time Full-Body Motion Capture from Video and IMUs," in *2017 International Conference on 3D Vision (3DV)*. IEEE, 2017, pp. 449–457.

[21] K. Kilteni, R. Groten, and M. Slater, "The Sense of Embodiment in Virtual Reality," *Presence: Teleoperators and Virtual Environments*, vol. 21, no. 4, pp. 373–387, 2012.