2018 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct)

VIRTOOAIR: VIrtual Reality TOOlbox for Avatar Intelligent Reconstruction

System for VR motion reconstruction based on a VR tracking system and a single RGB camera

Armin Becher*
Electrical Engineering and Computer Science
Technische Hochschule Ingolstadt

Cristian Axenie†
Audi Konfuzius Institute Ingolstadt
Technische Hochschule Ingolstadt

Thomas Grauschopf ‡
Electrical Engineering and Computer Science
Technische Hochschule Ingolstadt

Abstract

Realistic full-body avatar representation inside Virtual Reality is a big shortcoming of state-of-the-art VR systems. It remains a technically challenging task to capture human motion precisely without marker-based full-body tracking systems, which are expensive and impractical. Trying to tackle this challenge, we propose a simple yet efficient approach for avatar motion reconstruction. *VIRTOOAIR* (VIrtual Reality TOOlbox for Avatar Intelligent Reconstruction) combines Deep Learning for upper body reconstruction and most recent methods for single camera based pose recovery for the lower body parts. Our preliminary results demonstrate the advantages of our system's avatar pose reconstruction. This is mainly determined by the use of a powerful learning system, which offers significantly better results than existing heuristic solutions for inverse kinematics. Our system supports the paradigm shift towards learning systems capable to track full-body avatars inside Virtual Reality without the need of expensive external tracking hardware.

Index Terms: Machine Learning—Supervised learning by regression—;——Virtual Reality—Motion capture—

1 Introduction

With state-of-the-art Head Mounted Displays (HMDs), Virtual Reality (VR) is becoming affordable and impacts the general public. Although VR systems are getting cheaper and better, they are still limited in many ways. A critical shortcoming of these systems is the unrealistic representation of user avatars inside virtual worlds. Often, only the hand controllers are rendered but not the arms holding them. The rendering of lower body and legs is limited or lacking. This is a discomforting experience because the user perceives his or her own body only partially. Additionally, it has been shown in previous studies that users feel less present in virtual environments if they do not see their own body [18]. It also turned out that distance estimations inside virtual worlds improve if an animated virtual representation of the user's body exists [14]. These initial studies advocate for high-fidelity user's avatar motion estimation and reconstruction.

From a more practical perspective, a more significant shortcoming is the inadequate display of the remote user's avatar in distributed VR systems, or Immersive Collaborative Virtual Environments (ICVEs) [15]. In such cases, only a simple static head model and controllers are displayed. Yet, full body representations can increase the feeling of co-presence inside the virtual world, raise efficiency and reduce misunderstandings when working on collaborative tasks [6, 11].

Currently there is an increasing interest from both academic and industry to develop full body tracking or inverse kinematics (IK) solutions that could enable a more realistic avatar representation. Such existing techniques either require expensive marker-based tracking hardware or present only a limited number of user movements. Another drawback of marker-based tracking solutions is that users have to wear invasive motion-capturing suits, which can be detrimental for natural human motion.

To overcome these limitations, we introduce *VIRTOOAIR*, a system and methods for real-time motion capturing, estimation and reconstruction in virtual environments. The basic idea is to use the potential of modern Deep Learning algorithms, the 18 DOF VR tracking system, and an inexpensive RGB camera to predict and capture users' motions for high-fidelity reconstruction.

2 Related Work

A large body of literature has been published concerning realistic avatar representations without marker-based motion capturing systems. Some researchers use special depth cameras like the *Microsoft Kinect* or the *Intel RealSense* [17, 19]. As *Mehta et al.* outlined, Red Green Blue Depth (RGBD) camera based methods are a viable approach, but have to face challenges such as light sensitivity, size, power consumption, low resolution, and limited range while not being as widely available and reasonably priced as traditional RGB cameras [13]. Another problem with RGBD cameras like the *Microsoft Kinect* is that the infrared signal causes interference with the *HTC Vive* tracking system[1].

To overcome such limitations, other authors made use of inexpensive RGB cameras for full-body reconstruction [1, 13, 20, 23]. While they do not face the constraints of depth cameras, there are other challenges when using only RGB images for avatar reconstruction. For example, certain poses shown on monocular images are geometrically under-defined, and body part occlusions can make full-body position estimation a very difficult task [1].

From another perspective, existing systems make heavy use of computing power in order to regress the human body pose (i.e., using complex kinematic chain models) from sequences of RGB images. While these approaches show promising results, such systems are not capable of real-time tracking and only achieve frame-rates of less than two fps [4, 16, 23].

*e-mail: armin.becher@thi.de
†e-mail: cristian.axenie@audi-konfuzius-institut-ingolstadt.de
‡e-mail: thomas.grauschopf@thi.de

[1] https://steamcommunity.com/app/358720/discussions/0/485624149155072103/

978-1-5386-7592-2/18/$31.00 ©2018 IEEE
DOI 10.1109/ISMAR-Adjunct.2018.00085

Another drawback of most RGB camera based frameworks, such as [16, 20], and [23], is that they only capture the user's skeleton joint positions and no rotations. However, since rotations are essential to animate and visualize avatars correctly, these methods are only of limited use in VR avatar reconstruction.

Trying to circumvent restrictions of state-of-the-art camera reconstruction frameworks, *Kanazawa et al.* introduced a system which only relies on a single RGB camera to reconstruct a full 3D mesh of a human body. The system utilizes pre-trained Deep Learning networks for fast image processing to obtain camera parameters, pose and shape of a human actor in real-time. Since *Kanazawa et al.* directly regress avatar parameters from image pixels, they achieved end-to-end recovery of human shape and pose [10].

Such a single RGB input source is not sufficient in VR for high-fidelity reconstruction. A multimodal input approach is preferred to a purely visual approach. Hand and head controllers provide an augmented sensory input to disambiguate motion and contribute to a more precise avatar reconstruction. Previous work has proven that learning in a multimodal sensory space can unveil underlying correlations for improved position estimation [3, 7]. *VIRTOOAIR* follows a multimodal approach and introduces preliminary experiments focused on fusing VR controller data with camera data in order to extract precise motion data for high-fidelity avatar reconstruction.

3 Proposed System

VIRTOOAIR employs the success recipes of latest research in RGB camera-based tracking techniques and augments it with other sensory data (i.e., controllers) towards a multimodal VR tracking system capable of high-fidelity full-body avatar pose reconstruction. Figure 1 shows sample poses captured within the proposed system.

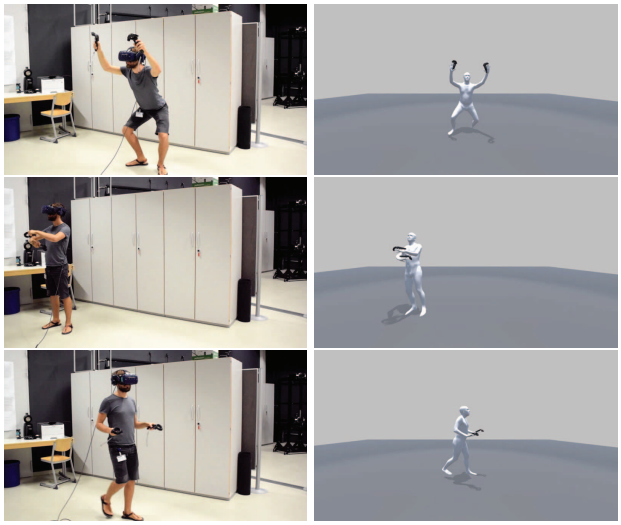

Figure 1: *VIRTOOAIR* reconstruction capabilities

VIRTOOAIR is composed of several modules. The first and second modules use the data of the VR tracking system to learn the inverse kinematics of the upper-body and the global rotation and position of a user inside VR. The third and fourth modules focus on the RGB video data to reconstruct the lower body pose of a user's avatar using a model similar to [10]. The last module is responsible for rendering the avatar inside the virtual world. Figure 2 gives an overview of the presented system's pipeline.

While the first four modules of *VIRTOOAIR* are written in *python*, the visualizer component uses the popular game engine *Unity 3D*.

3.1 End-to-End Reconstruction

For the lower body reconstruction, a modified version of the end-to-end recovery framework described by [10] is used. With their system, it is possible to directly regress human body shape and pose from pixels of an RGB image. They use the *Skinned Multi-Person Linear model* (*SMPL*) [12] as human body avatar. Therefore, the proposed method also works with the *SMPL* body model and uses the same skeleton structure.

3.1.1 Augmenting the Visual Input Features

One of the input requirements for the end-to-end recovery algorithm is a tight bounding box around the user's image for every frame. Because the framework described by *Kanazawa et al.* does not provide such functionality, an additional image preprocessing step was added. We augment the RGB input with a background subtracted image which was recorded before the user steps inside the tracking area. Such a feature extraction is applying a pixel mask to every frame to extract the user's body and then calculate a tight bounding box around that user.

3.1.2 Global Rotation, Position and Upper Body

The end-to-end framework can reconstruct human body poses based on only one RGB camera. For every frame, the joint rotations and camera parameters will be obtained. The actual position and global rotation of a user in space and a reasonable approximation of the arm position cannot be given due to the rather inaccurate and ambiguous camera-based tracking system. For this reason, only the lower body will be reconstructed with the end-to-end framework.

3.2 Neural Network Inverse Kinematic Learning

The global position and rotation, as well as the local joint angles of both arms, will be retrieved via a deep neural network, the second module in figure 2. Instead of using pixels of the RGB camera, the semantically superior VR tracking data is used as an input. The *HTC Vive* HMD comes with two hand controllers. Orientation and position of these devices are tracked via an external infrared tracking setup. Altogether, 18 degrees of freedom (DOF) are captured by *HTC's* external tracking system. As shown in figure 2, the global rotation of the headset and two vectors from the user's head to the hand positions (first module) are used as input for the inverse kinematic regression component. The output of the network is the global rotation of the *spine3* joint, the local rotations for the *neck*, *elbows*, *shoulders*, and both *collar* joints of the *SMPL* skeleton model.

As already outlined in [22], regressing joint rotations is a challenging task when training neural networks. One of the challenges with Euler rotation angles in $\mathbb{R}^3$ space is that they are not unique and suffer from singularities. In order to overcome this limitation, we use quaternions [9]. A quaternion can be used to represent 3D-rotations as a point on a hypersphere in $\mathbb{R}^4$ space.

Every rotation in 3D-space can be expressed by one quaternion and its negative (q = -q). To make the representation unambiguous, a preprocessing step is introduced which forces the quaternions to one side of the hypersphere. If the real part w is smaller than zero, the quaternion will be negated.

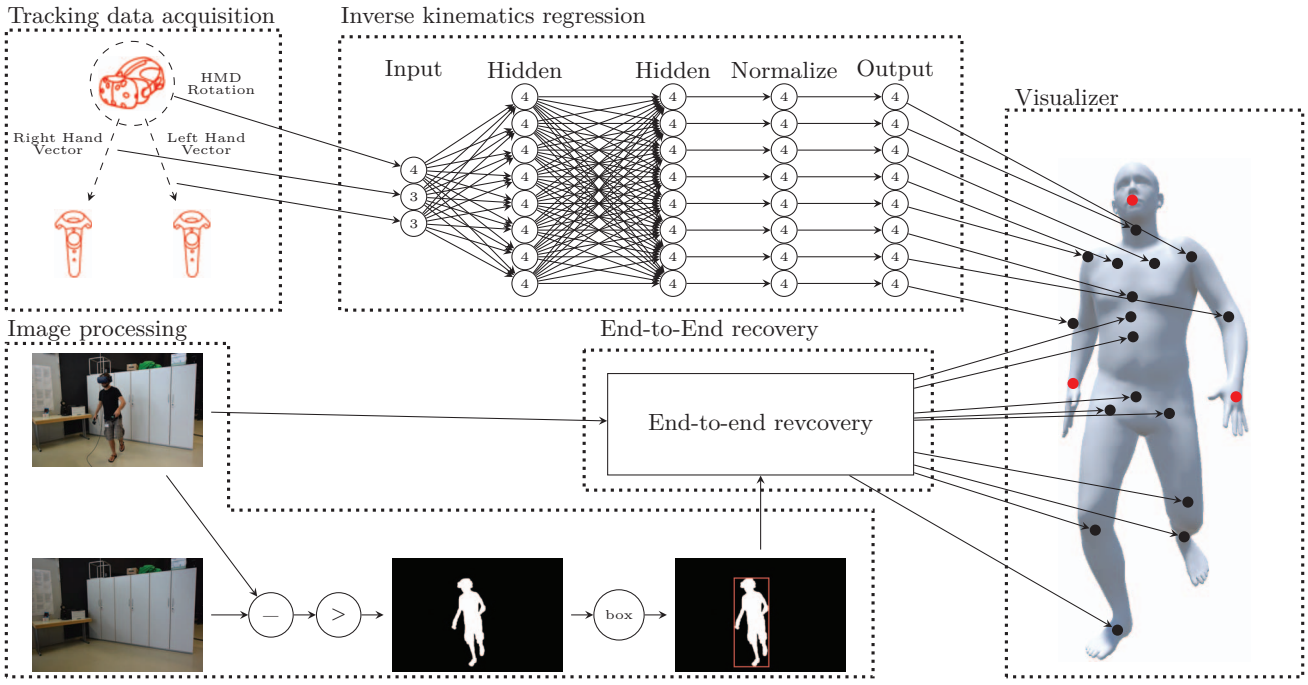

Figure 2: *VIRTOOAIR* processing pipeline. The first system component is the VR tracking data acquisition module. The VR tracking data (i.e., HMD and hands motion) are fed to a deep neural network to learn the upper body joint configurations. The deep neural network for inverse kinematics regression is the second module. The third module is responsible for the RGB image handling and contains a pre-processing routine (i.e., extract bounding box). Both, the bounding box and the camera image are constantly fed into the end-to-end recovery framework (i.e., fourth module) to reconstruct the lower body joint rotations. The last module (visualizer) is responsible for rendering the virtual body inside VR correctly.

Quaternions only represent a rotation in space if they lie on a hypersphere. To meet this condition, an additional normalization layer is added right before the output layer of the neural network inverse kinematics learner (see figure 2).

The network is trained using the rotational difference between the predicted and the ground-truth rotations of a large motion database. For quaternions different metrics exist [8]. The loss function of the neural network therefore measures the rotational offset between predicted and ground-truth poses:

$$\phi = 2 \cdot \arccos\left(|\langle q_1, q_2 \rangle|\right) \tag{1}$$

Because the system uses four-dimensional quaternion rotations and three-dimensional vectors, 10 input neurons (2 vectors and 1 quaternion) and 32 output neurons (8 quaternions) exist. The two hidden layers consist of 32 neurons. Rectified linear unit (ReLU) activation function is used for all layers except for the output layer which has a linear activation. The implementation uses *Keras* and *Tensorflow*. The source code will be available at `https://gitlab.com/akii-microlab/virtooair`.

3.2.1 Training the Inverse Kinematic Learner

To train the neural network, the Carnegie Mellon University (CMU) motion capture database [5] is used. It contains over 2,500 motion capturing sequences. The free *Huge FBX Mocap Library*[2] was used within *Unity 3D*. It is available in the *Unity 3D* asset store and contains the whole CMU Dataset converted into *Unity's* internal animation format. Within *Unity 3D*, all motions were mapped to the *SMPL* skeleton with a fixed default skeleton size. The whole dataset is split into a test set (activities 1 to 30) and a training set (activities 31 to 144). In order to regress the global rotation of a user (*spine3* joint of the *SMPL* model), the *SMPL* skeleton is rotated to a random direction around the up-axis for every frame generated for the training dataset. With this technique, it can be assured that the trained model will generalize better because it will not learn joint configurations depending on the user's current viewing direction.

[2] `https://assetstore.unity.com/packages/3d/animations/huge-fbx-mocap-library-part-1-19991`

3.2.2 Skeleton Size and Inverse Kinematics

As mentioned earlier, the kinematics learner is trained with the default skeleton size of the *SMPL* body model [12]. Because the system should also predict joint rotations of users with different body proportions, the length of both input vectors (the vectors from the head to the hands) is scaled relative to the user's arm span. During a short calibration process inside VR, the user stands in a T-pose and presses the trigger button of one of the two VR controllers. Afterwards, the skeleton is scaled to match the user's arm span. Because the neural network was trained on a vector length relative to the distance between both hands, the normalized head-to-hand vectors are sent to the trained inverse kinematics reconstruction system. This way we ensure that the local joint rotations are regressed regardless of the size of a user.

The proposed regression network only estimates the joint rotations. Therefore, it is possible that the end-effectors (both hands of the *SMPL* model) differ from the tracked controller positions. To ensure that the hand positions align with the

controllers, we used *cyclic coordinate descent* (*CCD*) [21] to iteratively make small adjustments to each joint until the end-effector and the target align.

4 Experimental Results

Our preliminary experiments prove that our inverse kinematics deep learner reconstructs the upper-body better than conventional IK solutions. In the following, we describe the methods and evaluate them against other approaches. For all tests a PC with the following hardware specifications is used: Intel i7-7700K CPU, 16 GB RAM, and a GeForce GTX 1070 graphics card.

As already mentioned, the CMU dataset was used to train the pose regression neural network. The test set (activities 1 to 30) is used to evaluate performance. The Mean Per Joint Position Error (MPJPE) is calculated using the Euclidean distance between the ground-truth and the predicted joint positions. For the Mean Per Joint Rotation Error (MPJRE), the quaternion metric shown in equation 1 is used.

The upper body kinematics learning algorithm is compared against three popular and widely used inverse kinematics solvers. *FABRIK* is an IK solving algorithm which avoids the use of direct joint rotations [2]. It iteratively finds joint positions via the location of points on a line. The *CCD* algorithm [21] is similar to *FABRIK*, yet instead of finding a point on a line, every single joint in the IK chain gets bent towards the target. Like *FABRIK*, *CCD* is an iterative IK solver which terminates when the last joint in the IK chain aligns with the target position.

The last IK algorithm the proposed method is compared to is called *Limb*. It comes with the commercially available *Unity 3D* plug-in *Final IK*[3]. It is based on a trigonometric IK solving strategy which tries to heuristically keep the joint configuration of both arms in a natural and relaxed configuration.

To compare the proposed IK solving algorithm with other IK solvers, the root joint (*spine3*) gets aligned with the ground-truth rotation and position. Table 1 shows how the upper body reconstruction differs from conventional IK solutions. As one can see from the results, *VIRTOOAIR* outperforms existing inverse kinematic solutions in angle and positional error due to its built-in soft constraints of human motion which are learned from a manifold of real-world human poses.

Table 1: Comparison between different IK algorithms

Method	MPJPE	MPJRE	Time
Limb	29.5 mm	67.9°	**0.1 ms**
CCD [21]	54.7 mm	105.8°	0.8 ms
FABRIK [2]	43.7 mm	88.4°	0.2 ms
VIRTOOAIR	**25.7 mm**	**13.5°**	2.2 ms

Currently, a limiting factor is the computation time (2.2 ms). This is due to the additional run of the *CCD* solver for end-effector target alignment and the communication latency among *Unity 3D* and the *pyhton* program implementing the inverse kinematics learner. This limitation will be tackled in the next stages, where a monolithic solution will be developed. The average processing of the neural network without the global pipeline and IK solver overhead takes only 0.2 ms. Therefore, more than one millisecond is spent on the Inter-Process Communication (IPC) which can presumably be optimized with other IPCs such as shared memory or the like.

We also evaluated the performance of our algorithm for the root joint (*Spine3*) position and rotation estimation on the same test data. The MPJPE is 54.1 mm and the MPJRE 14.2°. This shows that the proposed method is also capable of regressing global joint rotations. A simple heuristic solution such as using head rotations for the global orientation of the root joint leads to much worse results (MPJRE = 25.3°). This shows the superiority of our method which uses deep learning to make better pose estimations based on real-world tracking data.

With another test setup, we performed preliminary experiments for assessing lower body reconstruction performance. As ground-truth, three additional *HTC Vive* tracker got attached to different lower body joints of the tracked user (e.g., left hip, left knee and left foot). The *VIRTOOAIR* pipeline was used to reconstruct the avatar inside the virtual world. For about half a minute a user was walking back and forth in front of the camera while the Euclidean distances between the tracking targets and the reconstructed avatar joints were measured.

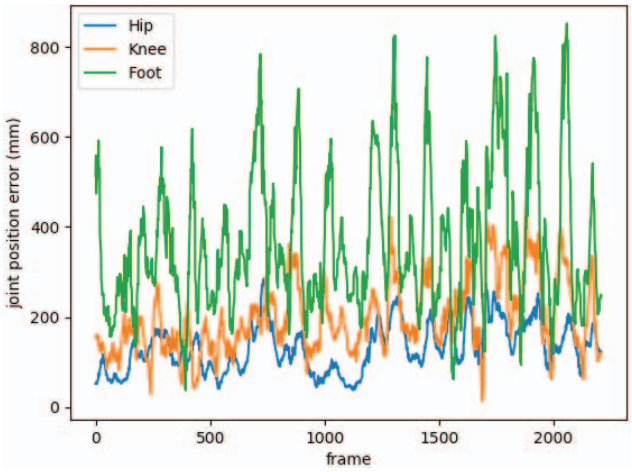

Figure 3: Lower body reconstruction results

Figure 3 shows the position error of each joint over time. The mean position error for all three joints are: hip 132 mm, knee 196 mm, and foot 366 mm. The position error increases the further the body parts are away from the root joint (in our case *Spine3*). This happens because all local rotation errors accumulate to an even more significant position error of the last joint. While evaluating the system we also noticed a delay in the camera reconstruction system which also contributes to a worse joint position result. On average the end-to-end recovery method only needed 8.2 ms for pose recovery. Therefore we conclude that an additional and much larger latency is induced by the USB-camera[4].

5 Limitations

As shown in the previous section, one of the limitations is that the lower body reconstruction has a high positional offset which also fluctuates strongly over time. This also leads to situations where the user's legs are not grounded to the virtual floor. The latency of the lower body reconstruction is also noticeable.

[3] `root-motion.com/finalikdox/html/page7.html`

[4] *Logitech C922 Pro*

Another challenge with the stated system is that a tight bounding box of the user is required in order to use the reconstruction framework. The background subtraction technique is limited because it will not work if objects other than the user's body are moving in front of the camera. It also requires the lighting condition of a scene to remain the same.

6 Conclusion and Future Work

VIRTOOAIR introduces a novel, inexpensive approach to achieve high-fidelity multimodal motion capturing and avatar representation in VR. By fusing an inverse kinematics learning module for precise upper-body motion reconstruction with single RGB camera input for lower-body estimation the system obtains a rich representation of human motion inside VR. The learning capabilities allow natural pose regression with cheap and affordable marker-less motion capturing hardware.

VIRTOOAIR is work in progress, but targets significant contributions for feasible distributed VR systems, by also tackling shape and texture estimation using similar deep learning tools in a unifying framework. We believe that such a framework will genuinely enable a robust, multimodal, high-fidelity representation of avatars in Virtual Reality. We will further improve the system's pose recovery performance by directly fusing VR tracking data and the lower semantic input of RGB images into one holistic end-to-end reconstruction process.

References

[1] A. Agarwal. *Machine learning for image based motion capture*. PhD thesis, Institut National Polytechnique de Grenoble-INPG, 2006.

[2] A. Aristidou and J. Lasenby. Fabrik: A fast, iterative solver for the inverse kinematics problem. *Graphical Models*, 73(5):243–260, 2011. doi: 10.1016/j.gmod.2011.05.003

[3] C. Axenie, C. Richter, and J. Conradt. A self-synthesis approach to perceptual learning for multisensory fusion in robotics. *Sensors*, 16(10), 2016.

[4] F. Bogo, A. Kanazawa, C. Lassner, P. Gehler, J. Romero, and M. J. Black. Keep it smpl: Automatic estimation of 3d human pose and shape from a single image. In *European Conference on Computer Vision*, pp. 561–578, 2016.

[5] Carnegie Mellon University. Cmu graphics lab motion capture database. doi: 10.17616/R3N35M

[6] J. S. Casanueva and E. H. Blake. The effects of avatars on co-presence in a collaborative virtual environment. In *Annual Conference of the South African Institute of Computer Scientists and Information Technologists (SAICSIT2001). Pretoria, South Africa*, 2001.

[7] Y. Du, Y. Wong, Y. Liu, F. Han, Y. Gui, Z. Wang, M. Kankanhalli, and W. Geng. Marker-less 3d human motion capture with monocular image sequence and height-maps. In B. Leibe, J. Matas, N. Sebe, and M. Welling, eds., *Computer Vision - ECCV 2016*, vol. 9908 of *Lecture Notes in Computer Science*, pp. 20–36. Springer International Publishing and Imprint: Springer, 2016.

[8] Q. Du Huynh. Metrics for 3d rotations: Comparison and analysis. *Journal of Mathematical Imaging and Vision*, 35(2):155–164, 2009. doi: 10.1007/s10851-009-0161-2

[9] W. R. Hamilton. Ii. on quaternions; or on a new system of imaginaries in algebra. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 25(163):10–13, 1844.

[10] A. Kanazawa, M. Black, D. W. Jacobs, and J. Malik. End-to-end recovery of human shape and pose. 2017.

[11] J. M. Linebarger and G. D. Kessler. The effect of avatar connectedness on task performance. *Lehigh Univ TR*, 2002.

[12] M. Loper, N. Mahmood, J. Romero, G. Pons-Moll, and M. J. Black. Smpl: a skinned multi-person linear model. *ACM Transactions on Graphics*, 34(6):1–16, 2015. doi: 10.1145/2816795.2818013

[13] D. Mehta, S. Sridhar, O. Sotnychenko, H. Rhodin, M. Shafiei, H.-P. Seidel, W. Xu, D. Casas, and C. Theobalt. Vnect. *ACM Transactions on Graphics*, 36(4):1–14, 2017. doi: 10.1145/3072959.3073596

[14] B. J. Mohler, S. H. Creem-Regehr, W. B. Thompson, and H. H. Bülthoff. The effect of viewing a self-avatar on distance judgments in an hmd-based virtual environment. *Presence: Teleoperators and Virtual Environments*, 19(3):230–242, 2010. doi: 10.1162/pres.19.3.230

[15] O. Otto, D. Roberts, and R. Wolff. A review on effective closely-coupled collaboration using immersive cve's. In H. Sun, ed., *Proceedings of the 2006 ACM international conference on Virtual reality continuum and its applications*, p. 145. ACM, New York, NY, 2006. doi: 10.1145/1128923.1128947

[16] A.-I. Popa, M. Zanfir, and C. Sminchisescu. Deep multitask architecture for integrated 2d and 3d human sensing. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[17] A. Qammaz, D. Michel, and A. Argyros. A hybrid method for 3d pose estimation of personalized human body models.

[18] M. V. Sanchez-Vives and M. Slater. From presence to consciousness through virtual reality. *Nature reviews. Neuroscience*, 6(4):332–339, 2005. doi: 10.1038/nrn1651

[19] M. Sra and C. Schmandt. Metaspace. In C. Latulipe, B. Hartmann, and T. Grossman, eds., *Proceedings of the 28th Annual ACM Symposium on User Interface Software and Technology*, pp. 47–48. Association for Computing Machinery, New York, NY, 2015. doi: 10.1145/2815585.2817802

[20] B. Tekin, I. Katircioglu, M. Salzmann, V. Lepetit, and P. Fua. Structured prediction of 3d human pose with deep neural networks. *arXiv preprint arXiv:1605.05180*, 2016.

[21] L.-C. Wang and C. C. Chen. A combined optimization method for solving the inverse kinematics problems of mechanical manipulators. *IEEE Transactions on Robotics and Automation*, 7(4):489–499, 1991. doi: 10.1109/70.86079

[22] X. Zhou, X. Sun, W. Zhang, S. Liang, and Y. Wei. Deep kinematic pose regression.

[23] X. Zhou, M. Zhu, S. Leonardos, K. G. Derpanis, and K. Daniilidis. Sparseness meets deepness: 3d human pose estimation from monocular video. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4966–4975, 2016.