Computer Animation and Virtual Worlds

WILEY

SPECIAL ISSUE PAPER OPEN ACCESS

Interaction With Virtual Objects Using Human Pose and Shape Estimation

Hong Son Nguyen[1] | DaEun Cheong[1] | Andrew Chalmers[2] | Myoung Gon Kim[1] | Taehyun Rhee[3] | JungHyun Han[1]

[1]Korea University, Seoul, South Korea | [2]Victoria University of Wellington, Wellington, New Zealand | [3]The University of Melbourne, Parkville, Australia

Correspondence: JungHyun Han (jhan@korea.ac.kr)

Received: 3 May 2025 | **Accepted:** 11 May 2025

Funding: This work was partly supported by the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. RS-2020-II200861), through the ICT Creative Consilience Program (IITP-2025-RS-2020-II201819) and the ITRC (Information Technology Research Center) Support Program (IITP-2025-RS-2020-II201460), and by the Commercializations Promotion Agency for R&D Outcomes (COMPA) grant funded by the Korea government (MSIT) (RS-2024-00459242).

Keywords: augmented reality | full-body interaction | pose and shape estimation

ABSTRACT

In this article, we propose an AR system that facilitates a user's natural interaction with virtual objects in an augmented reality environment. The system consists of three modules: human pose and shape estimation, camera-space calibration, and physics simulation. The first module estimates a user's 3D pose and shape from a single RGB video stream, thereby reducing the system setup cost and broadening potential applications. The camera-space calibration module estimates the user's camera-space position to align the user with the input RGB image. The physics simulation enables seamless and physically natural interaction with virtual objects. Two prototyping applications built upon the system prove an enhancement in the quality of interaction, fostering a more immersive and intuitive user experience.

1 | Introduction

For various augmented reality (AR) applications, such as virtual try-on and virtual training, it is essential to provide natural seamless interactions between real humans and virtual objects. Enabling such interactions usually necessitates capturing the humans' motions at real time. In an AR volleyball game, for example, actions such as spiking or blocking a virtual ball must be convincingly mirrored in the AR context.

For capturing the full-body joints of a human, various motion sensors and motion capture suits have been used [1–4]. However, the intrusive nature of marker-based systems and the encumbrance of motion capture suits pose significant barriers to widespread adoption in everyday AR use. An alternative approach involves using RGB-D sensors to compute 3D poses and approximate 3D shapes based on color and depth information [5–8]. Recently, deep learning-based approaches further the field by estimating 3D pose information using only RGB sensors [9–11], leading to the development of various virtual interaction scenarios [12–14]. The deep learning-based approaches have been extended to provide 3D shape information, which is crucial for handling collision and occlusion in AR environments [15–18].

In this article, we propose an AR system that takes an RGB video as input, estimates a user's 3D pose and shape, and facilitates the user's interaction with 3D virtual objects. As shown in Figure 1, the system comprises three modules for (1) human pose and shape estimation, (2) camera-space calibration, and (3) physics simulation.

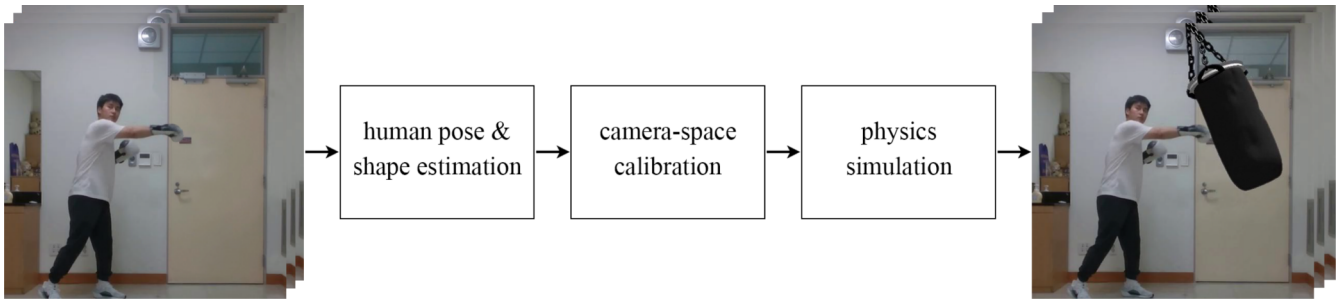


FIGURE 1 | Our AR system is composed of three modules that facilitate interactions between a real human and a virtual objects. In this AR boxing game, the user is interacting with the virtual punching bag.

The human pose and shape (HPS) estimated by the first module are represented in the parametric human model, SMPL-X [18]. As the SMPL-X model is a geometric entity, handling its interactions with virtual objects may appear to be straightforward. Unfortunately, the estimated model is *not* defined in the world or camera spaces. To enable the human model to move around and interact with dynamic virtual objects in the AR environment, we need to *calibrate* the human model every frame so that it is correctly transformed into the world or camera spaces. The focus of this article is on *camera-space calibration*. Finally, the third module for physics simulation is implemented using a game engine. Our contributions are summarized as follows:

- We present a working AR solution that enables interactions of a real human with 3D virtual objects. Taking only a single RGB video feed as input, it supports 3D shape-based natural interactions.
- Built upon deep learning techniques that are well balanced between accuracy and efficiency, the solution reconstructs and calibrates the human model in a robust manner.
- The solution works in an easy-to-setup AR environment, and therefore it can be used across various AR applications.

2 | Related Work

2.1 | Human Pose and Shape Estimation

A significant challenge in full-body interaction lies in the accurate tracking of human motion. Traditional methods in VR, AR, and graphics often rely on motion capture suits and sensors [19, 20], while others explore marker-based setups [1, 3]. Beyond full-body suits, alternative systems use sparse markers or wearable sensors combined with inverse kinematics or SLAM. Roth et al. [21] demonstrated that a reduced rigid-body marker set with inverse kinematics lowers latency and task load in VR without sacrificing body ownership. Yi et al. [4] proposed EgoLocate, which integrates inertial mocap and monocular SLAM to achieve real-time human motion capture and localization using six IMUs and a monocular camera. Ghorbani et al. [22] deployed deep learning models to estimate human motion and shape from the raw input of marker-based motion capture systems. Chen et al. [2] employed a Bi-LSTM model to reconstruct human motion from only four markers. Kim et al. [23] proposed DAMO, a deep solver that generalizes across marker configurations via end-to-end optimization. Despite these advances, marker- and sensor-based systems still suffer from restricted mobility [24] and high setup costs due to specialized hardware and calibration.

Facing the limitations of marker and sensor-based methods, researchers have leveraged RGB-D sensors, such as Kinect [25], for full-body motion estimation [5, 7, 8, 26]. Shum et al. [8] devised a framework for real-time motion tracking using Kinect. Cui et al. [5] employed two Kinect sensors, positioned at the front and back, to comprehensively capture human motion. Kim et al. [26] combined high-speed RGB and ToF sensors to capture dynamic human poses. Ren et al. [7] utilized a convolutional neural network (CNN) along with a depth sensor to capture human motions in large-scale scenarios.

RGB-based deep learning solutions [9–11] have emerged as effective alternatives for 3D pose estimation without specialized hardware. Existing methods for 3D pose estimation using monocular RGB camera fall into two categories: single-stage vs. two-stage. The single-stage techniques [27, 28] directly localize 3D body keypoints from input images. Sun et al. [28] used the soft-argmax operation to obtain joint locations from 3D heatmaps, whereas Moon et al. [27] proposed a top-down approach for multiperson pose estimation. Additionally, Apple introduced ARKit [29], a framework that estimates 3D human pose from RGB images for AR applications. In contrast, the two-stage methods [30–33] leverage accurate 2D pose estimation, lifting keypoints to 3D space. Pavllo et al. [32] introduced dilated temporal convolutions, and Zheng et al. [33] used a Vision Transformer to capture spatial and temporal dependencies.

The skinned multiperson linear (SMPL) model [34] has catalyzed a series of studies that aim at estimating 3D HPS using deep learning models. By representing 3D human body as differentiable functions of the HPS parameters, the model is well-suited for end-to-end training of neural networks. Notable contributions in this domain include SMPLify [15], Human Mesh Recovery (HMR) [16], SMPLify-X [18], and VIBE [17]. HPS estimation methods can be performed through optimization or regression techniques. Optimization-based methods [15, 35–37] align a body model with image cues such as joints, dense vertex correspondences or 2D segmentation masks. In the context of optimization, for example, Luvizon et al. [38] extended 3D HPS estimation to multi-human, scene-aware settings from a single

RGB video. On the other hand, regression-based methods [16, 17, 39–42] train a network using a loss akin to the optimization objective function, enabling it to predict body model parameters.

2.2 | Full-Body Interaction

In the past, in order to study full-body interaction in virtual environments, marker-based motion capture systems were commonly used [43–46]. Debarba et al. [44] enhanced interaction with virtual objects by utilizing a setup involving 4 LED markers and 14 cameras to capture human motions. Young et al. [46] implemented high five in virtual environments by employing a motion capture system.

The advent of RGB-D sensors has spurred extensive research into full-body interaction applications. As emphasized by Caserman et al. [47], VR applications employing full-body motion data from RGB-D cameras span a broad spectrum of activities. These include virtual climbing of ladders [48], cycling-based exergames [49–51], collaborative manufacturing tasks involving human–robot interaction [52], and the creation of role-playing games [53].

Beyond VR applications, commercial products such as Kinect and RealSense have enabled full-body interaction in AR. Despite the availability of these commercial solutions, there has been relatively little scholarly exploration into full-body interaction in AR [54]. Moreover, although these systems provide more flexibility compared with marker-based systems, they often necessitate the use of multiple RGB-D sensors, leading to intricate setups and higher costs.

Recent advancements in deep learning have enabled a number of approaches to tackle the challenge of human–virtual interaction. Hwang et al. [55] proposed a lightweight 3D human pose estimation model applied to virtual avatar reconstruction, with potential extensions to human–virtual interaction. Nguyen et al. [13] developed an AR fitness application that supports full-body interaction by integrating 3D pose estimation into an exercise interface. Wu et al. [14] created an AR-based martial arts training system enabling real-time full-body interaction through pose-guided motion evaluation. Recent studies [56–58] have expanded avatar-based interactions into immersive environments.

Alongside skeleton-based methods, human body shape has garnered significant attention for full-body interaction, particularly the SMPL [34]-based approaches. De et al. [59] used HMR to estimate HPS and created a virtual agent for interactive VR. Additionally, Šarić et al. [60] introduced an extended reality-based telemedicine collaboration system that presents a 3D avatar of the patient to a remote clinician, leveraging 3D HPS estimation derived from a deep learning model. Recent studies have expanded full-body interaction into immersive environments [61] and AR environments [62, 63].

Inspired by the learning-based methods, this article proposes an HPS estimation network, which utilizes the shape information to interact with virtual objects in AR environments. Our system is easy to set up, requiring no additional devices, and is demonstrated to be cost-effective.

3 | Camera-Space Calibration

Our AR setup is shown in Figure 2: A user moves on the floor and the large screen in front of the user displays the mirrored view of the environment using an RGB webcam, which is mounted on top of the screen and captures the *full body* of the user. The captured environment is augmented with virtual objects so that the user can interact with them.

Among the three modules presented in Figure 1, this article focuses on the second module, *camera-space calibration*, while briefly sketching the first (HPS estimation) and third (physics simulation) modules, for which there exist many off-the-shelf methods and commercial/open-source programs.

In general, existing methods for estimating a human's motion compute the HPS parameters for SMPL-X [18], which are denoted as θ and β, respectively. Using θ and β, the human's 3D triangular mesh is reconstructed, which we denote as $\mathcal{V}$. Then, we derive the joints' 3D positions from $\mathcal{V}$. What we call the *human model* consists of the 3D mesh $\mathcal{V}$ and the 3D joints denoted as $\mathcal{J}$. In order to estimate the human model, we use the Human Mesh Recovery (HMR) framework [16]. For more details, readers are referred to Appendix A.

In our AR setup, the human model is invisible to the user, but its mesh is used, for example, to detect *collisions* with the virtual objects. In the current implementation, the physics simulation module is implemented using Unity engine. However, it can be seamlessly replaced by other commercial or open-source engines.

3.1 | Calibration Network

HMR employs the *weak-perspective camera model*, which is implemented as an *orthographic projection* plus a *scaling*. Consequently, the reconstructed human model is not correctly located in the 3D coordinate system of the real camera, for example, its camera-space depth may be over- or under-estimated. Then, not only would we miss the *collision* between the human model and

FIGURE 2 | In this AR environment, the user can see his full body and the virtual objects in the large screen.

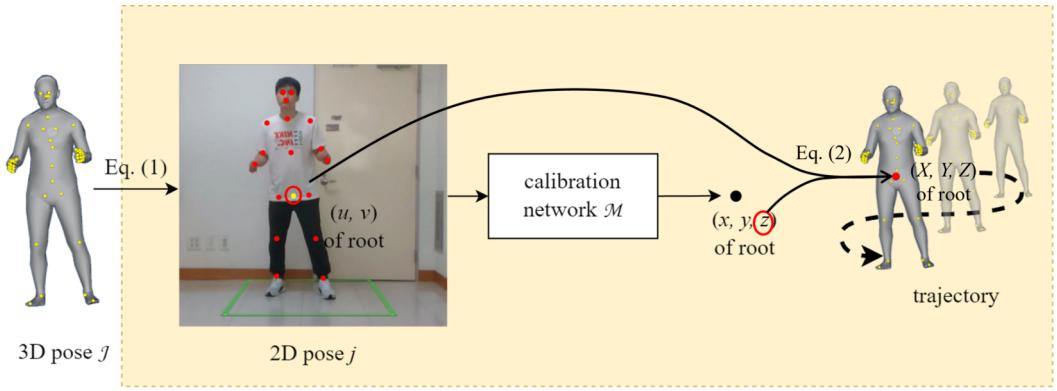


FIGURE 3 | For calibration, we use the 3D pose $\mathcal{J}$ computed by the HPS estimation module. The sequence of the calibrated root nodes makes up the trajectory.

virtual objects but also we would handle the *occlusion* between them incorrectly. Therefore, the human model must be *calibrated* so that it is transformed into the *full-perspective* camera space.

Figure 3 illustrates the calibration process. The essential step for calibration is to estimate the camera-space coordinates of the *root joint*. Once they are estimated correctly, all the other joints' coordinates can be determined immediately using their positions relative to the root, which are stored in the 3D pose $\mathcal{J}$.

For this, we employed a deep learning network proposed by Pavllo et al. [32], which processes "a sequence of 2D poses" to output the camera-space positions of the root. It was modified and retrained to take "a single frame of 2D pose" as input and estimate the root's camera-space position. Let $\mathcal{M}$ denote the network and j denote the 2D pose input to $\mathcal{M}$. By projecting the 3D pose $\mathcal{J}$ onto the image plane, we can obtain j. It is simple to implement because HMR returns not only θ and β but also the parameters of the weak-perspective camera model, that is, the scale factor $s \in \mathbb{R}$ and the 2D translation vector $t \in \mathbb{R}^2$. The 2D pose j is obtained using the 3D pose $\mathcal{J}$ as follows:

$$j = s\Pi(\mathcal{J}) + t \tag{1}$$

where $\Pi(\cdot)$ denotes orthographic projection. $\mathcal{M}(j)$ is the root's coordinates, (x, y, z), in the camera space.

The calibration network, $\mathcal{M}$, is trained using the Human3.6M dataset [64]. Note that the Human3.6M dataset contains only "square" images with aspect ratio one. In contrast, our system presented in Figure 1 is designed to take general "rectangular" images. Consequently, the x- and y-coordinates estimated by $\mathcal{M}$ are not reliable.

Therefore, we take only the z-coordinate, which is reliable. Let (X, Y, Z) denote the *calibrated* coordinates of the root, whereas $Z = z$, X, and Y are computed using the root's pixel coordinates, (u, v), in the image plane. See Figure 4; (u, v) are transformed by the inverse of the camera intrinsic matrix $\mathbf{K}$ and then are scaled by z:

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = z\mathbf{K}^{-1}\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} \tag{2}$$

For more details, readers are referred to Appendix B.

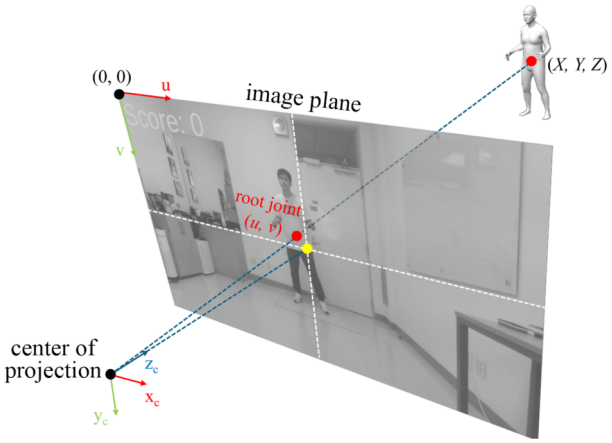


FIGURE 4 | The root's coordinates, (u, v), in the image plane are transformed by $\mathbf{K}^{-1}$, and then are scaled by z, which is estimated by $\mathcal{M}$, to define the camera-space coordinates, (X, Y, Z).

3.2 | Discussion

In general, the HPS methods based on the weak-perspective model are inaccurate at predicting 3D pose, due to its near-orthographic projection. For example, high levels of perspective distortion in selfies often lead to unacceptable 3D pose. In contrast, the predicted pose would be acceptable if the user is sufficiently far away from the camera. It is the case in our AR setup shown in Figure 2, where a fairly long distance between the camera and the user should be maintained so as to capture the user's full-body motion.

Recently, the problem of the weak-perspective model has been directly tackled by a few efforts that estimate the camera's intrinsic parameters [65, 66] or the user's depth [67]. While these methods improve the 3D accuracy, they introduce additional processing and training complexities. Our strategy of adopting HMR and integrating it with the calibration network can be considered as well-balanced between accuracy and efficiency. (We train HMR with the BEDLAM dataset [68], which provides diverse and realistic synthetic data).

The root joint's camera-space position keeps changing over the frames, and it is generally called *trajectory*. Existing methods

TABLE 1 | Calibration model evaluation.

# Input frames	Parameters	FLOPs	Trajectory errors
1 frame	**4.24M**	**8.46M**	**147.0 mm**
27 frames	8.51M	16.99M	147.7 mm
81 frames	12.70M	25.38M	181.3 mm

for calculating the trajectory involve complex processes, ranging from optimization-based frameworks [9, 69, 70] to estimating relative scene depths by exploiting scene constraints [71, 72]. Although these methods do not estimate the 3D shape, they estimate not only the trajectory but also the 3D pose, which is already computed by our "human pose and shape estimation" module. Instead of adopting such methods, which consume significant amount of computing resources to produce redundant information, that is, the 3D pose $\mathcal{J}$, we compute the 2D pose j from $\mathcal{J}$ "with little cost" and provide it for the modification of the network proposed by Pavllo et al. [32] to estimate the trajectory.

The original network proposed by Pavllo et al. [32] is based on dilated temporal convolutions, predicting the trajectory using both past and future data. On the other hand, they proposed *causal convolutions* for real-time applications, which have access only to past frames and predict the trajectory of the current frame. Readers are referred to Appendix C for visualization of the two types of convolutions.

To evaluate accuracy across the input sequences of different lengths, we made an experiment, where the causal convolutions were trained respectively with sequence lengths of 1, 27 and 81 frames. The case with a single-frame sequence corresponds to our calibration network. Training was made with Human3.6M dataset [64] in a *semi-supervised manner*, where Subjects 1, 5, and 6 were labeled, that is, both (u, v) and (x, y, z) coordinates were utilized, whereas Subjects 7 and 8 were treated as unlabeled, that is, only (u, v) coordinates were used.

We evaluated the three trajectory estimation models with Subjects 9 and 11 of the Human3.6M dataset. Obviously, our calibration network requires fewer parameters and FLOPs than the others, as shown in Table 1, making it more suitable for real-time applications. It is interesting to find that with respect to the *trajectory error*, which is defined as the average difference of the ground-truth and predicted positions of the root joint, our network is the most accurate, and the model with 27 frames is more accurate than that with 81 frames, that is, the longer input sequence leads to the larger error. With these findings, we speculate that in the causal convolutions, the current frame holds the most critical information for predicting the root joint position, and less relevant information of the past frames is accumulated over the frame sequence to hamper precise prediction.

4 | Applications

The equipment of our AR setup shown in Figure 2 consists of a webcam, a PC and a large screen that can be replaced by a TV. These are easily found in everyday environments, providing an easy-to-setup gaming platform. The setup can also be used for many other AR applications such as virtual try-on.

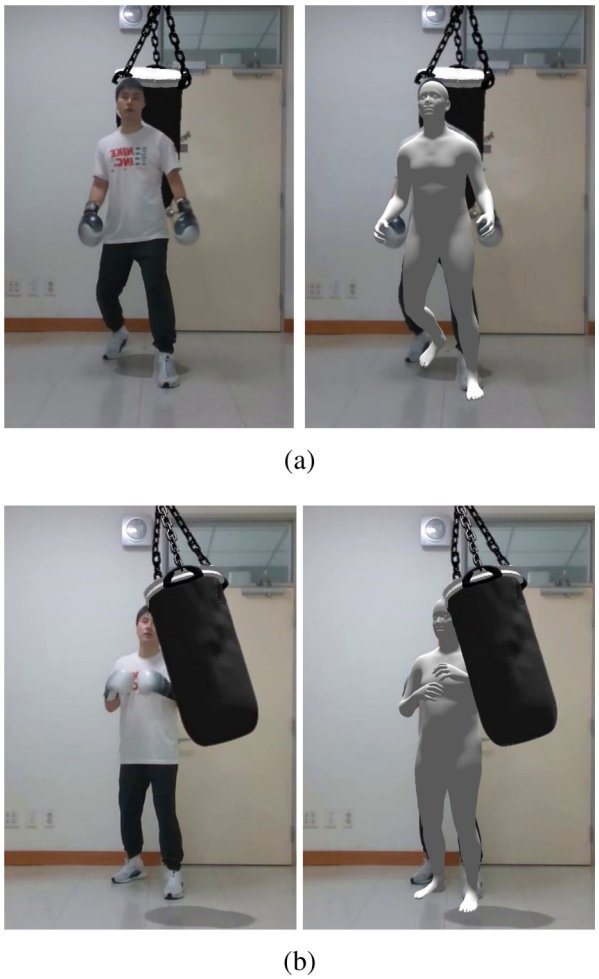


FIGURE 5 | Occlusion: (a) The real human occludes the virtual object. (b) The virtual Object occludes the real human.

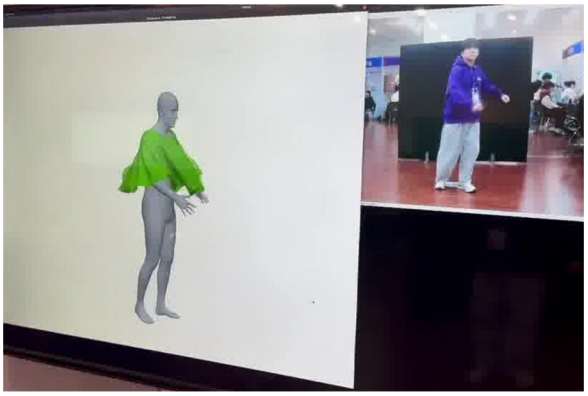
4.1 | Application 1: Boxing Game

Human shape estimation opens the door to a wide range of AR applications. We have developed a boxing game prototype, where the user punches a virtual object. In Figure 5, the user moves around a punching bag and keeps punching it. To convincingly composite the virtual object into the real-world environment, we use the estimated shape of the user. Figure 5 clearly shows that our system is able to handle *occlusion* between the real and virtual objects. For enhancing the realism, the shadow of the virtual punching bag is generated and composited into the scene using a virtual floor. The virtual floor can also be used, for example, to make a virtual ball bouncing in the real scene.

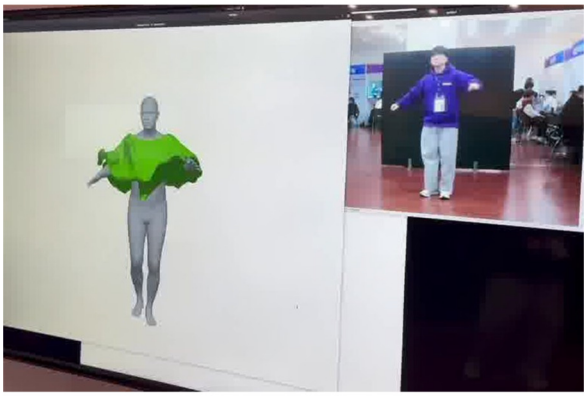
4.2 | Application 2: Virtual Try-On

Our AR setup presented in Figure 2 can be used for many other AR applications such as virtual try-on. Figure 6 shows a sequence of snapshots that visualize complex real-time interactions between a virtual garment and the human model

(a)



(b)



(c)

FIGURE 6 | In this virtual try-on demo, a real human physically interacts with a virtual garment in real time. (a), (b), (c).

reconstructed by the method presented in this article. In this proof-of-concept implementation, the garment is composed of 8K vertices. It is simulated with XPBD (extended position-based dynamics) [73] implemented in Taichi [74], a high-level language for GPU programming, and the proximity query for colliding triangles is accelerated via spatial hashing [75].

4.3 | Evaluation and Discussion

Table 2 shows the times consumed by the first module for HPS estimation (i.e., by Yolo v8n + HMR) and the second one for camera calibration (i.e., by the calibration network). The total time is 5.5 ms. When they are used for the boxing game, the time is increased to 6.6 ms, as shown in Table 3. It implies that Unity consumes 1.1 ms for rigid-body simulation and rendering. In contrast, the time is increased to 36.2 ms for the virtual try-on application, which implies that our XPBD implementation consumes approximately 30 ms.

TABLE 2 | Runtime performances of HPS estimation and calibration.

Model	Time (ms)
Yolo v8n	3.04
HMR	2.22
Calibration network	0.24
Total	5.5

TABLE 3 | Runtime performances of applications.

Application	Time (ms)	FPS
Boxing game	6.6	151
Virtual try-on	36.2	27

Using a simplistic cloth model, Figure 6 shows a proof-of-concept implementation. More realistic garments can be simulated with XPBD, but the design of realistic garments introduces a separate challenge that must be carefully managed in practical systems. For example, they have to be designed manually so that the clothes do not interpenetrate with the human mesh at initialization (i.e., at T pose). We believe that such an effort is orthogonal to the current contribution of this article. We envision that the matter can be addressed in a future work targeted at serious clothing applications.

Our AR setup has its own disadvantages. Unlike AR headsets such as HoloLens [76], Meta Quest 3 [77] and Apple Vision Pro [78], the large screen cannot provide binocular depth perception, often failing to create immersive experiences for users. Despite the drawbacks of lacking depth perception and immersion, however, we believe that the applications that can be built upon our AR setup are different from those on AR headsets, and the virtual try-on is a good example for the first category of applications.

5 | Conclusion and Future Work

This article presents a practical solution that enables natural interactions of a human with 3D virtual objects in AR environments using only a conventional RGB video camera. Built upon deep learning techniques that are well balanced between accuracy and efficiency, the solution reconstructs and calibrates the human model in a robust manner so as to handle collision and occlusion successfully.

While our approach may be able to facilitate significantly full-body interaction in AR, there is potential for further refinement. Our AR system relies on the HPS estimation technique, yet it does not offer precise 3D localization due to the limitations of the SMPL model. Our solution addresses this limitation through a

trajectory model. However, the trajectory model is limited within the cases relevant to training dataset. This can fail with challenging poses, for example, when the user is lying down. This would limit the potential applications of the current method to games and entertainment content. To extend the application range, an advanced technique needs to be developed, thus improving the quality of interactions.

Conflicts of Interest

The authors declare no conflicts of interest.

Data Availability Statement

Data sharing is not applicable to this article as no new data were created or analyzed in this study.

References

1. A. Cannavò, F. G. Pratticò, A. Bruno, and F. Lamberti, "Ar-Mocap: Using Augmented Reality to Support Motion Capture Acting," in *Proceedings of the 2023 IEEE Conference Virtual Reality and 3D User Interfaces (VR)* (IEEE, 2023), 318–327.

2. X. Chen, X. Jiang, L. Zhan, et al., "Full-Body Human Motion Reconstruction With Sparse Joint Tracking Using Flexible Sensors," *ACM Transactions on Multimedia Computing, Communications, and Applications* 20, no. 2 (2023): 1–19.

3. Y. Li, D. Weng, D. Li, and Y. Wang, "A Low-Cost Drift-Free Optical-Inertial Hybrid Motion Capture System for High-Precision Human Pose Detection," in *Proceedings of the 2019 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct)* (IEEE, 2019), 75–80.

4. X. Yi, Y. Zhou, M. Habermann, et al., "Egolocate: Real-Time Motion Capture, Localization, and Mapping With Sparse Body-Mounted Sensors," *ACM Transactions on Graphics* 42, no. 4 (2023): 1–17.

5. Y. Cui, W. Chang, and D. Stricker, "Fully Automatic Body Scanning and Motion Capture Using Two Kinects," in *SIGGRAPH Asia 2013 Posters* (Association for Computing Machinery, 2013), 1.

6. Z. Liu, L. Zhou, H. Leung, and H. P. Shum, "Kinect Posture Reconstruction Based on a Local Mixture of Gaussian Process Models," *IEEE Transactions on Visualization and Computer Graphics* 22, no. 11 (2015): 2437–2450.

7. Y. Ren, C. Zhao, Y. He, et al., "Lidar-Aid Inertial Poser: Large-Scale Human Motion Capture by Sparse Inertial and Lidar Sensors," *IEEE Transactions on Visualization and Computer Graphics* 29, no. 5 (2023): 2337–2347.

8. H. Shum and E. S. Ho, "Real-Time Physical Modelling of Character Movements With Microsoft Kinect," in *Proceedings of the 18th ACM Symposium on Virtual Reality Software and Technology* (Association for Computing Machinery, 2012), 17–24.

9. D. Mehta, O. Sotnychenko, F. Mueller, et al., "Xnect: Real-Time Multi-Person 3d Motion Capture With a Single RGB Camera," *ACM Transactions on Graphics (TOG)* 39, no. 4 (2020): 81–82.

10. D. Mehta, S. Sridhar, O. Sotnychenko, et al., "Vnect: Real-Time 3d Human Pose Estimation With a Single RGB Camera," *ACM Transactions on Graphics (TOG)* 36, no. 4 (2017): 1–14.

11. S. Shimada, V. Golyanik, W. Xu, P. Pérez, and C. Theobalt, "Neural Monocular 3d Human Motion Capture With Physical Awareness," *ACM Transactions on Graphics* 40, no. 4 (2021): 1–15.

12. M. Kari, T. Grosse-Puppendahl, L. F. Coelho, et al., "Transformr: Pose-Aware Object Substitution for Composing Alternate Mixed Realities," in *Proceedings of the 2021 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)* (IEEE, 2021), 69–79.

13. H. S. Nguyen, M. Kim, C. Im, S. Han, and J. Han, "Convnextpose: A Fast Accurate Method for 3d Human Pose Estimation and Its Ar Fitness Application in Mobile Devices," *IEEE Access* 11 (2023): 117393–117402, https://doi.org/10.1109/ACCESS.2023.3326343.

14. E. Wu and H. Koike, "Futurepose-Mixed Reality Martial Arts Training Using Real-Time 3d Human Pose Forecasting With a RGB Camera," in *Proceedings of the 2019 IEEE Winter Conference on Applications of Computer Vision (WACV)* (IEEE, 2019), 1384–1392.

15. 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 *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14 Proceedings, Part V 14* (Springer, 2016), 561–578.

16. A. Kanazawa, M. J. Black, D. W. Jacobs, and J. Malik, "End-To-End Recovery of Human Shape and Pose," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (IEEE, 2018), 7122–7131.

17. M. Kocabas, N. Athanasiou, and M. J. Black, "Vibe: Video Inference for Human Body Pose and Shape Estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (IEEE, 2020), 5253–5263.

18. G. Pavlakos, V. Choutas, N. Ghorbani, et al., "Expressive Body Capture: 3d Hands, Face, and Body From a Single Image," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (IEEE, 2019), 10975–10985.

19. C. Canton-Ferrer, J. R. Casas, and M. Pardas, "Marker-Based Human Motion Capture in Multiview Sequences," *EURASIP Journal on Advances in Signal Processing* 2010 (2010): 1–11.

20. A. C. Sementille, L. E. Lourenço, J. R. F. Brega, and I. Rodello, "A Motion Capture System Using Passive Markers," in *Proceedings of the 2004 ACM SIGGRAPH International Conference on Virtual Reality Continuum and Its Applications in Industry* (Association for Computing Machinery, 2004), 440–447.

21. D. Roth, J.-L. Lugrin, J. Büser, G. Bente, A. Fuhrmann, and M. E. Latoschik, "A Simplified Inverse Kinematic Approach for Embodied Vr Applications," in *2016 IEEE Virtual Reality (VR)* (IEEE, 2016), 275–276.

22. N. Ghorbani and M. J. Black, "Soma: Solving Optical Marker-Based Mocap Automatically," in *Proceedings of the IEEE/CVF International Conference on Computer Vision* (IEEE, 2021), 11117–11126.

23. K. Kim, S. Seo, D. Han, and H. Kang, "Damo: A Deep Solver for Arbitrary Marker Configuration in Optical Motion Capture," *ACM Transactions on Graphics* 44, no. 1 (2024): 1–14, https://doi.org/10.1145/3695865.

24. S. Chagué and C. Charbonnier, "Real Virtuality: A Multi-User Immersive Platform Connecting Real and Virtual Worlds," in *Proceedings of the 2016 Virtual Reality International Conference* (Association for Computing Machinery, 2016), 1–3.

25. R. A. Newcombe, S. Izadi, O. Hilliges, et al., "Kinectfusion: Real-Time Dense Surface Mapping and Tracking," in *Proceedings of the 2011 10th IEEE International Symposium on Mixed and Augmented Reality* (IEEE, 2011), 127–136.

26. J. Kim and M. Kim, "Motion Capture With High-Speed Rgb-d Cameras," in *Proceedings of the 2014 International Conference on Information and Communication Technology Convergence (ICTC)* (IEEE, 2014), 394–395.

27. G. Moon, J. Y. Chang, and K. M. Lee, "Camera Distance-Aware Top-Down Approach for 3d Multi-Person Pose Estimation From a Single Rgb Image," in *Proceedings of the IEEE/CVF International Conference on Computer Vision* (IEEE, 2019), 10133–10142.

28. X. Sun, B. Xiao, F. Wei, S. Liang, and Y. Wei, "Integral Human Pose Regression," in *Proceedings of the European Conference on Computer Vision (ECCV)* (Springer International Publishing, 2018), 529–545.

29. Apple, "Arkit," https://developer.apple.com/documentation/arkit/.

30. C.-H. Chen and D. Ramanan, "3D Human Pose Estimation= 2D Pose Estimation+ Matching," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (IEEE, 2017), 7035–7043.

31. J. Martinez, R. Hossain, J. Romero, and J. J. Little, "A Simple Yet Effective Baseline for 3d Human Pose Estimation," in *Proceedings of the IEEE International Conference on Computer Vision* (IEEE, 2017), 2640–2649.

32. D. Pavllo, C. Feichtenhofer, D. Grangier, and M. Auli, "3d Human Pose Estimation in Video With Temporal Convolutions and Semi-Supervised Training," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (IEEE, 2019), 7753–7762.

33. C. Zheng, S. Zhu, M. Mendieta, T. Yang, C. Chen, and Z. Ding, "3d Human Pose Estimation With Spatial and Temporal Transformers," in *Proceedings of the IEEE/CVF International Conference on Computer Vision* (IEEE, 2021), 11656–11665.

34. 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, no. 6 (2015): 1–16, https://doi.org/10.1145/2816795.2818013.

35. T. Fan, K. V. Alwala, D. Xiang, W. Xu, T. Murphey, and M. Mukadam, "Revitalizing Optimization for 3D Human Pose and Shape Estimation: A Sparse Constrained Formulation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision* (IEEE, 2021), 11457–11466.

36. D. Xiang, H. Joo, and Y. Sheikh, "Monocular Total Capture: Posing Face, Body, and Hands in the Wild," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (IEEE, 2019), 10965–10974.

37. D. Xiang, F. Prada, C. Wu, and J. Hodgins, "Monoclothcap: Towards Temporally Coherent Clothing Capture From Monocular RGB Video," in *Proceedings of the 2020 International Conference on 3D Vision (3DV)* (IEEE, 2020), 322–332.

38. D. C. Luvizon, M. Habermann, V. Golyanik, A. Kortylewski, and C. Theobalt, "Scene-Aware 3d Multi-Human Motion Capture From a Single Camera," in *Computer Graphics Forum*, vol. 42 (Wiley Online Library, 2023), 371–383.

39. Z. Qiu, Q. Yang, J. Wang, et al., "Psvt: End-To-End Multi-Person 3d Pose and Shape Estimation With Progressive Video Transformers," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (IEEE, 2023), 21254–21263.

40. Y. Sun, Q. Bao, W. Liu, Y. Fu, M. J. Black, and T. Mei, "Monocular, One-Stage, Regression of Multiple 3d People," in *Proceedings of the IEEE/CVF International Conference on Computer Vision* (IEEE, 2021), 11179–11188.

41. Y. Sun, Q. Bao, W. Liu, T. Mei, and M. J. Black, "Trace: 5d Temporal Regression of Avatars With Dynamic Cameras in 3d Environments," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (IEEE, 2023), 8856–8866.

42. Y. Sun, W. Liu, Q. Bao, Y. Fu, T. Mei, and M. J. Black, "Putting People in Their Place: Monocular Regression of 3d People in Depth," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (IEEE, 2022), 13243–13252.

43. 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* (Association for Computing Machinery, 2013), 123–132.

44. H. G. Debarba, S. Perrin, B. Herbelin, and R. Boulic, "Embodied Interaction Using Non-Planar Projections in Immersive Virtual Reality," in *Proceedings of the 21st ACM Symposium on Virtual Reality Software and Technology* (Association for Computing Machinery, 2015), 125–128.

45. T. C. Peck, S. Seinfeld, S. M. Aglioti, and M. Slater, "Putting Yourself in the Skin of a Black Avatar Reduces Implicit Racial Bias," *Consciousness and Cognition* 22, no. 3 (2013): 779–787.

46. M. K. Young, J. J. Rieser, and B. Bodenheimer, "Dyadic Interactions With Avatars in Immersive Virtual Environments: High Fiving," in *Proceedings of the ACM SIGGRAPH Symposium on Applied Perception* (Association for Computing Machinery, 2015), 119–126.

47. P. Caserman, A. Garcia-Agundez, and S. Göbel, "A Survey of Full-Body Motion Reconstruction in Immersive Virtual Reality Applications," *IEEE Transactions on Visualization and Computer Graphics* 26, no. 10 (2019): 3089–3108.

48. T. M. Takala and M. Matveinen, "Full Body Interaction in Virtual Reality With Affordable Hardware," in *IEEE Virtual Reality (VR)* (IEEE, 2014), 157.

49. 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* (Association for Computing Machinery, 2014), 475–478.

50. L. A. Shaw, B. C. Wünsche, C. Lutteroth, S. Marks, and R. Callies, Challenges in virtual reality exergame design (2015).

51. E. Tuveri, L. Macis, F. Sorrentino, L. D. Spano, and R. Scateni, "Fitmersive Games: Fitness Gamification Through Immersive Vr," in *Proceedings of the International Working Conference on Advanced Visual Interfaces* (Association for Computing Machinery, 2016), 212–215.

52. E. Matsas, G.-C. Vosniakos, and D. Batras, "Modelling Simple Human-Robot Collaborative Manufacturing Tasks in Interactive Virtual Environments," in *Proceedings of the 2016 Virtual Reality International Conference* (Association for Computing Machinery, 2016), 1–4.

53. K.-T. Chou, M.-C. Hsiu, and C. Wang, "Fighting Gulliver: An Experiment With Cross-Platform Players Fighting a Body-Controlled Giant," in *Proceedings of the 33rd Annual ACM Conference Extended Abstracts on Human Factors in Computing Systems* (Association for Computing Machinery, 2015), 65–68.

54. A. Genay, A. Lécuyer, and M. Hachet, "Being an Avatar for Real: A Survey on Virtual Embodiment in Augmented Reality," *IEEE Transactions on Visualization and Computer Graphics* 28, no. 12 (2021): 5071–5090.

55. D.-H. Hwang, S. Kim, N. Monet, H. Koike, and S. Bae, "Lightweight 3d Human Pose Estimation Network Training Using Teacher-Student Learning," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision* (IEEE, 2020), 479–488.

56. A. Chalmers, F. Zaman, and T. Rhee, "Avatar360: Emulating 6-Dof Perception in 360degpanoramasthroughavatar-Assistednavigation," in *Proceedings of the 2024 IEEE Conference Virtual Reality and 3D User Interfaces(V R)* (IEEE, 2024), 630–638.

57. T. Rhee, L. Petikam, B. Allen, and A. Chalmers, "Mr360: Mixed Reality Rendering for 360 Panoramic Videos," *IEEE Transactions on Visualization and Computer Graphics* 23, no. 4 (2017): 1379–1388.

58. T. Rhee, S. Thompson, D. Medeiros, R. Dos Anjos, and A. Chalmers, "Augmented Virtual Teleportation for High-Fidelity Telecollaboration," *IEEE Transactions on Visualization and Computer Graphics* 26, no. 5 (2020): 1923–1933.

59. G. D. de Dinechin and A. Paljic, "Virtual Agents From 360 Video for Interactive Virtual Reality," in *Proceedings of the 32nd International Conference on Computer Animation and Social Agents* (Association for Computing Machinery, 2019), 75–78.

60. M. Šarić, M. Russo, L. Kraljević, and D. Meter, "Extended Reality Telemedicine Collaboration System Using Patient Avatar Based on 3d Body Pose Estimation," *Sensors* 24, no. 1 (2023): 27.

61. T. Rhee, A. Chalmers, F. Zaman, A. Stangnes, and V. Roberts, "Real-Time Stage Modelling and Visual Effects for Live Performances," in *ACM SIGGRAPH 2023 Real-Time Live!* (Association for Computing Machinery, 2023), 1–2.

62. H. Nguyen, A. Chalmers, D. Cheong, M. Kim, T. Rhee, and J. Han, "A Simple but Effective Ar Framework for Human-Object Interaction," in *EuroXR 2024: Proceedings of the 21st EuroXR International Conference* (VTT Technical Research Centre of Finland, 2024), 67–71.

63. H. S. Nguyen, A. Chalmers, D. Cheong, M. G. Kim, T. Rhee, and J. Han, "Full-Body Interaction in Mixed Reality Using 3D Pose and Shape Estimation," in *2025 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)* (IEEE Computer Society, 2025), 1306–1307.

64. C. Ionescu, D. Papava, V. Olaru, and C. Sminchisescu, "Human3.6m: Large Scale Datasets and Predictive Methods for 3d Human Sensing in Natural Environments," *IEEE Transactions on Pattern Analysis and Machine Intelligence* 36, no. 7 (2013): 1325–1339.

65. M. Kocabas, C.-H. P. Huang, J. Tesch, L. Müller, O. Hilliges, and M. J. Black, "Spec: Seeing People in the Wild With an Estimated Camera," in *Proceedings of the IEEE/CVF International Conference on Computer Vision* (IEEE, 2021), 11035–11045.

66. P. Patel and M. J. Black, "Camerahmr: Aligning People With Perspective,"*arXiv preprint arXiv:2411.08128* (2024).

67. S. Wang, J. Li, T. Li, et al., "Blade: Single-View Body Mesh Learning Through Accurate Depth Estimation. arXiv preprint arXiv:2412.08640," (2024).

68. M. J. Black, P. Patel, J. Tesch, and J. Yang, "Bedlam: A Synthetic Dataset of Bodies Exhibiting Detailed Lifelike Animated Motion," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (IEEE, 2023), 8726–8737.

69. D. Mehta, H. Rhodin, D. Casas, et al., "Monocular 3d Human Pose Estimation in the Wild Using Improved Cnn Supervision," in *Proceedings of the 2017 International Conference on 3D Vision (3DV)* (IEEE, 2017), 506–516.

70. A. Zanfir, E. Marinoiu, M. Zanfir, A.-I. Popa, and C. Sminchisescu, "Deep Network for the Integrated 3d Sensing of Multiple People in Natural Images," *Advances in Neural Information Processing Systems* 31 (Curran Associates, Inc, 2018).

71. Z. Weng and S. Yeung, "Holistic 3d Human and Scene Mesh Estimation From Single View Images," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (IEEE, 2021), 334–343.

72. A. Zanfir, E. Marinoiu, and C. Sminchisescu, "Monocular 3d Pose and Shape Estimation of Multiple People in Natural Scenes-The Importance of Multiple Scene Constraints," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (IEEE, 2018), 2148–2157.

73. M. Macklin, M. Müller, and N. Chentanez, "Xpbd: Position-Based Simulation of Compliant Constrained Dynamics," in *Proceedings of the 9th International Conference on Motion in Games* (Association for Computing Machinery, 2016), 49–54.

74. Y. Hu, T.-M. Li, L. Anderson, J. Ragan-Kelley, and F. Durand, "Taichi: A Language for High-Performance Computation on Spatially Sparse Data Structures," *ACM Transactions on Graphics* 38, no. 6 (2019): 1–16.

75. M. Teschner, B. Heidelberger, M. Müller, D. Pomerantes, and M. H. Gross, "Optimized Spatial Hashing for Collision Detection of Deformable Objects," in *VMV*, vol. 3 (Aka GmbH, 2003), 47–54.

76. Microsoft, "Hololens 2," (2019), https://www.microsoft.com/en-us/hololens/.

77. Meta. Quest 3, https://www.meta.com/kr/quest/quest-3/, (2023).

78. Apple, "Vision Pro," (2024), https://www.apple.com/apple-vision-pro/.

79. J. Bai, F. Lu, K. Zhang, et al., "Onnx: Open Neural Network Exchange," (2019), https://github.com/onnx/onnx.

80. Unity, "Sentis," https://unity.com/products/sentis.

Supporting Information

Additional supporting information can be found online in the Supporting Information section. **Data S1.** Supporting Information.

Appendix A

Reconstructing Full-Body Pose and Shape

In our system, the bounding-box image returned by an object detector is scaled to 224×224 and then fed into the CNN encoder. The encoder-generated features are iteratively processed by a regression module to produce (i) the pose parameters, $\theta \in \mathbb{R}^{3K}$, where K denotes the number of skeletal joints ($K = 22$ in the current implementation), and (ii) the shape parameters, $\beta \in \mathbb{R}^{10}$.

The pose parameters, θ, which represent the relative rotations of $K = 22$ joints, are converted into the axis-angle representations for integration with SMPL-X model. The shape parameters, β, are characterized by the first 10 principal components of the PCA shape space. SMPL-X is a differentiable function, W, which takes θ and β as input and produces a triangular mesh, $\mathcal{V}$, of 10,475 vertices:

$$\mathcal{V} = W(\theta, \beta) \tag{A1}$$

The 3D joint positions, denoted by $\mathcal{J}$, are derived from $\mathcal{V}$ via a linear regression function R:

$$\mathcal{J} = R(\mathcal{V}) \tag{A2}$$

The 3D human model consists of the 3D mesh $\mathcal{V}$ and the 3D joints $\mathcal{J}$.

Appendix B

Calibration

Given the focal lengths, f_x and f_y, and the principal point, (c_x, c_y), the camera intrinsic matrix $\mathbf{K}$ is defined as follows:

$$\mathbf{K} = \begin{pmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix}$$

The root's homogeneous coordinates, $(u, v, 1)$, are transformed by the inverse of the intrinsic matrix $\mathbf{K}$ and scaled by z estimated by $\mathcal{M}$:

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = z\mathbf{K}^{-1}\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = z\begin{pmatrix} \frac{1}{f_x} & 0 & -\frac{c_x}{f_x} \\ 0 & \frac{1}{f_y} & -\frac{c_y}{f_y} \\ 0 & 0 & 1 \end{pmatrix}\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{pmatrix} z\left(\frac{u-c_x}{f_x}\right) \\ z\left(\frac{v-c_y}{f_y}\right) \\ z \end{pmatrix} \tag{B1}$$

Note that Z is identical to z estimated by $\mathcal{M}$.

Appendix C

Symmetric Convolutions Vs. Causal Convolutions

Figure C1 is a copy of Figure 6 from Pavllo et al. [32] and compares symmetric convolutions and casual ones.

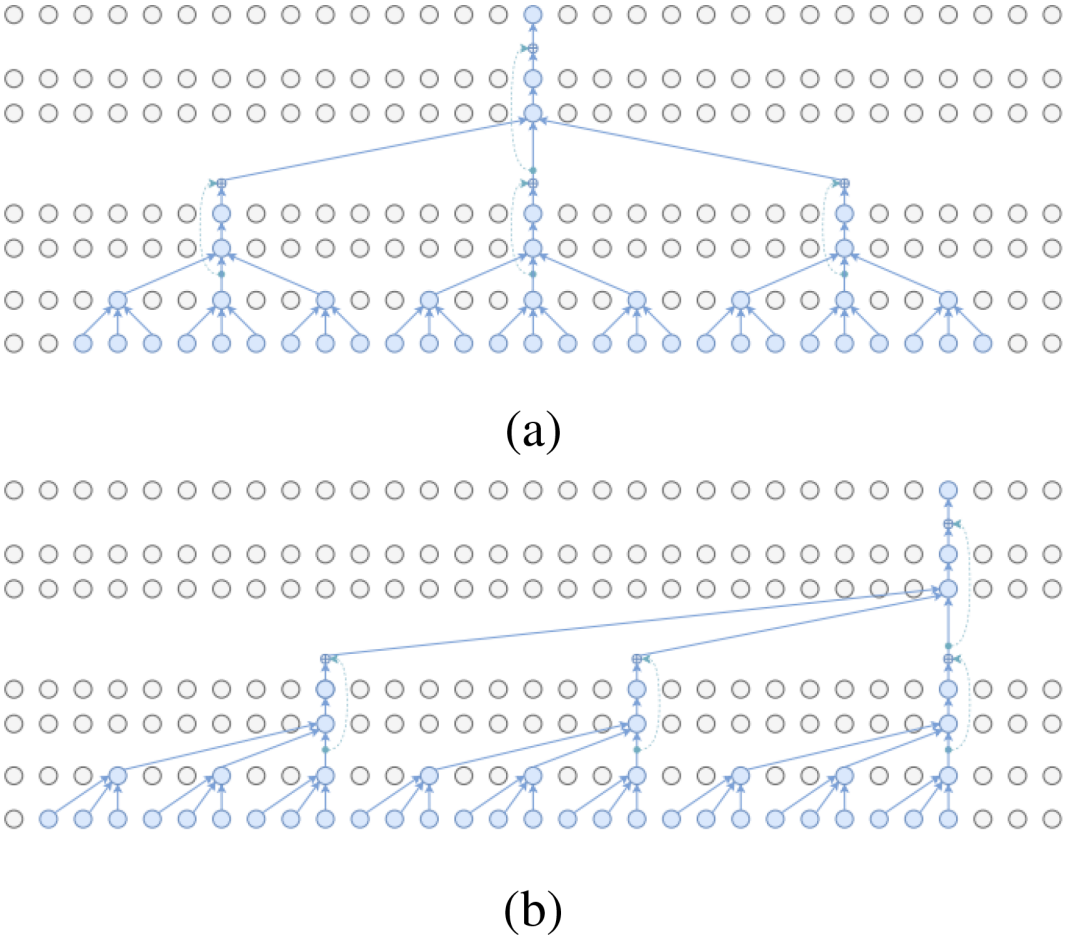


FIGURE C1 | Two types of convolutions: (a) Symmetric convolutions, (b) Causal convolutions.

Appendix D

Integration With Unity

After the calibration network (presented in Section 3.1) is trained, it is converted into the Open Neural Network Exchange (ONNX) [79] format. This conversion ensures compatibility with Unity via a lightweight cross-platform neural network inference library designed for Unity, Sentis [80], which supports deep learning network execution on both GPU and CPU.