ControllerPose: Inside-Out Body Capture with VR Controller Cameras

Karan Ahuja
Carnegie Mellon University
Pittsburgh, PA, USA
kahuja@cs.cmu.edu

Vivian Shen
Carnegie Mellon University
Pittsburgh, PA, USA
vhshen@andrew.cmu.edu

Cathy Mengying Fang
Carnegie Mellon University
Pittsburgh, PA, USA
catfang@mit.edu

Nathan Riopelle
Carnegie Mellon University
Pittsburgh, PA, USA
nriopell@andrew.cmu.edu

Andy Kong
Carnegie Mellon University
Pittsburgh, PA, USA
akong@andrew.cmu.edu

Chris Harrison
Carnegie Mellon University
Pittsburgh, PA, USA
chris.harrison@cs.cmu.edu

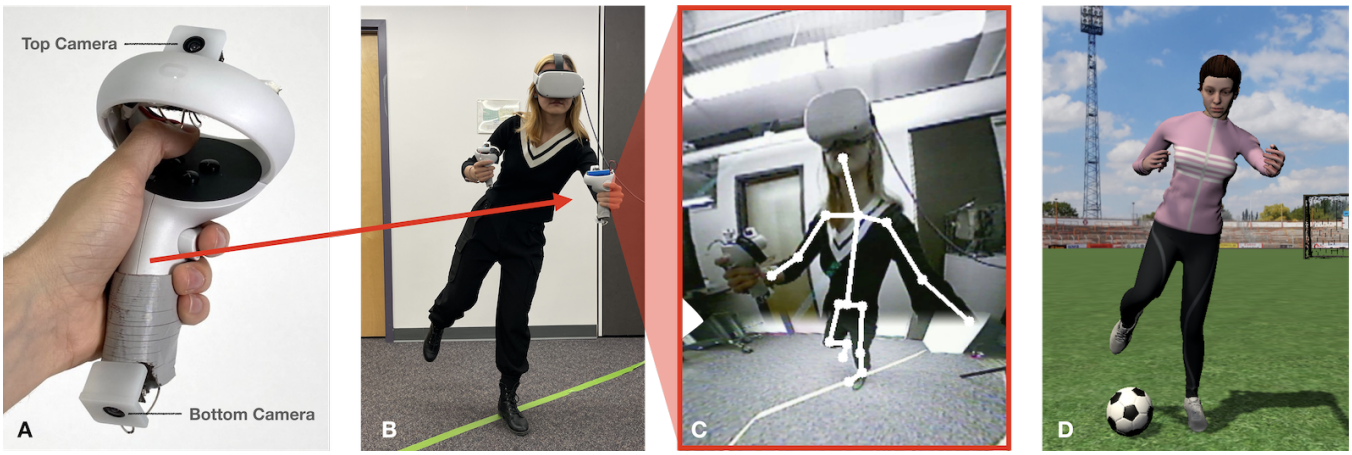

Figure 1: ControllerPose uses two fisheye cameras (A) on each VR hand controller (B) to composite a superior view for full-body 3D pose tracking (C), which is then used to control an avatar (D) offering a more embodied and immersive VR experience.

ABSTRACT

We present a new and practical method for capturing user body pose in virtual reality experiences: integrating cameras into handheld controllers, where batteries, computation and wireless communication already exist. By virtue of the hands operating in front of the user during many VR interactions, our controller-borne cameras can capture a superior view of the body for digitization. Our pipeline composites multiple camera views together, performs 3D body pose estimation, uses this data to control a rigged human model with inverse kinematics, and exposes the resulting user avatar to end user applications. We developed a series of demo applications illustrating the potential of our approach and more leg-centric interactions, such as balancing games and kicking soccer balls. We describe our proof-of-concept hardware and software, as well as results from our user study, which point to imminent feasibility.

CHI '22, April 29-May 5, 2022, New Orleans, LA, USA

ACM ISBN 978-1-4503-9157-3/22/04.
https://doi.org/10.1145/3491102.3502105

CCS CONCEPTS

• **Human-centered computing** → **Virtual reality**; *Interaction techniques.*

KEYWORDS

Virtual Reality; Pose Tracking; Motion Capture

ACM Reference Format:
Karan Ahuja, Vivian Shen, Cathy Mengying Fang, Nathan Riopelle, Andy Kong, and Chris Harrison. 2022. ControllerPose: Inside-Out Body Capture with VR Controller Cameras. In *CHI Conference on Human Factors in Computing Systems (CHI '22), April 29-May 5, 2022, New Orleans, LA, USA.* ACM, New York, NY, USA, 13 pages. https://doi.org/10.1145/3491102.3502105

1 INTRODUCTION

Today's virtual reality (VR) systems generally do not capture lower body pose – looking down in most VR experiences, one does not see their virtual legs, but rather an empty space with sometimes a diffuse shadow, reducing immersion and embodiment [34, 38]. This is because most contemporary VR systems only capture the motion of a user's head and hands, using IMUs in the headset and controllers, respectively. Some systems are able to capture other joints, such as the legs, by using accessory sensors, either worn [24, 25, 60] or placed in the environment [9, 12, 84]. The

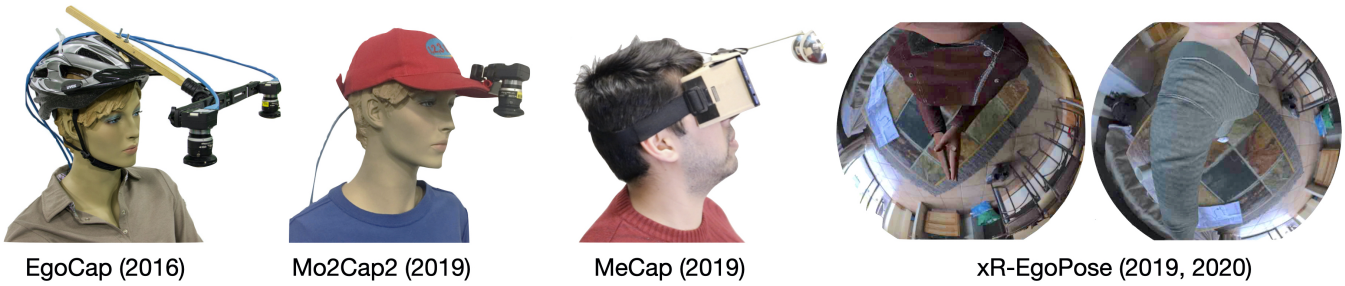

Figure 2: Four systems closely related to ControllerPose, though all instrument the head/headset as opposed to the hands/controllers. In order to see the lower body, the cameras operate quite far from the head, ranging from 7 to 25 cm from the user's forehead. xR-EgoPose [71, 72] does not provide a photo of their physical apparatus, but example views from the system are illustrative of occlusion issues faced by such approaches.

latter systems are comparably rare as the additional hardware cost and extra inconvenience of setup has been a major deterrent for consumer adoption.

A more practical approach is to use cameras in the headset [2, 72, 79] (Figure 2), though these innately have a very oblique view of the user, and line-of-sight to the lower body can be blocked by a user's clothing or body (e.g., stomach, breasts, hands - examples in Figure 2, far right). To increase user comfort, headset designs are becoming thinner, which reduces rotational inertia and torque on the head from the weight of the headset. Unfortunately, this will slowly preclude the ability to perform robust pose tracking with headset-borne cameras.

In this paper, we consider an alternative and practical method for capturing user body pose: integrating cameras into VR controllers (Figure 1A), where batteries, computation and wireless communication already exist. In a small motion capture study we ran, we found that the hands operate in front of the body a majority of the time (68.3%). Thus, we can opportunistically capture superior views of the body for digitization. In other cases, the hands are either too close or resting beside the body – in these cases, pose tracking will fail or would have to rely on e.g., inverse kinematics using available IMU data. However, the full-body tracking offered in many instances opens new and interesting leg-centric interactive experiences. For example, users can now stomp, lean, squat, lunge, balance and perform many other leg-driven interactions in VR (e.g., kicking a soccer ball in Figure 1D). We built a series of functional demo applications incorporating these types of motions, illustrating the potential and feasibility of our approach. In the following sections, we describe our implementation and user study, the results of which show our system can track full-body pose with a mean 3D joint error of 6.98 cm.

2 RELATED WORK

We now review related work in body capture and digitization, sensed both externally and by worn hardware. We more specifically discuss approaches that digitize the body for applications in virtual reality. Refer to [35] for a comprehensive survey.

2.1 External Body Pose Capture

There have been significant strides in capturing the user's pose with external sensors. Commercial systems such as Vicon [73] and OptiTrack [46] make use of retroreflective markers tracked by high-speed external infrared cameras. These systems are considered the gold standard and have been extensively used for character animation, games and movies as they provide low-latency and accurate motion capture ("MoCap").

In recent years, with the advent of deep learning, computer vision based pose estimation systems have become increasingly popular. These include systems that make use of RGB cameras [9, 12, 39, 51], depth sensors such as Kinect, OpenNI [50] and Intel RealSense [26], and systems that combine RGB and depth [43, 86]. Researchers have also considered non-optical tracking systems, utilizing modalities such as RF [85], capacitive sensing [83], magnetic fields [48, 54], mechanical linkages [67] and acoustics [5]. These systems have limited range and precision when compared to their visual counterparts.

In VR settings, the use of outside-in tracking (i.e., an external sensor looking at the user) has been explored in many systems such as the HTC Vive [22], Oculus Rift [42] and PlayStation VR [61]. These systems typically track the headset and two-handheld controllers and can roughly estimate the remainder of the joints with inverse kinematics [52, 59]. More faithful full-body tracking can be achieved by placing extra sensors on the limbs, such as HTC VIVE trackers [23], or bolstering capture with external cameras such as the Kinect.

2.2 Worn Body Pose Capture

Worn body capture systems generally offer greater ease-of-use and mobility to users, allowing for body digitization on-the-go and with less setup. While there are innumerable specialized systems that focus on digitizing the hands [17, 31, 82] and face [3, 28, 70], more related to our work are worn, self-contained approaches that attempt whole-body pose estimation. A wide variety of sensing techniques have been explored for whole-body capture, including exoskeletons [40], magnetic trackers [14], and ultrasonic beacons [74]. Distributed sensor-based approaches that instrument multiple points of the body with the same sensor are also popular; IMU's [7, 24, 68, 78], cameras [1, 47, 60], pressure sensors [81] and RFID's

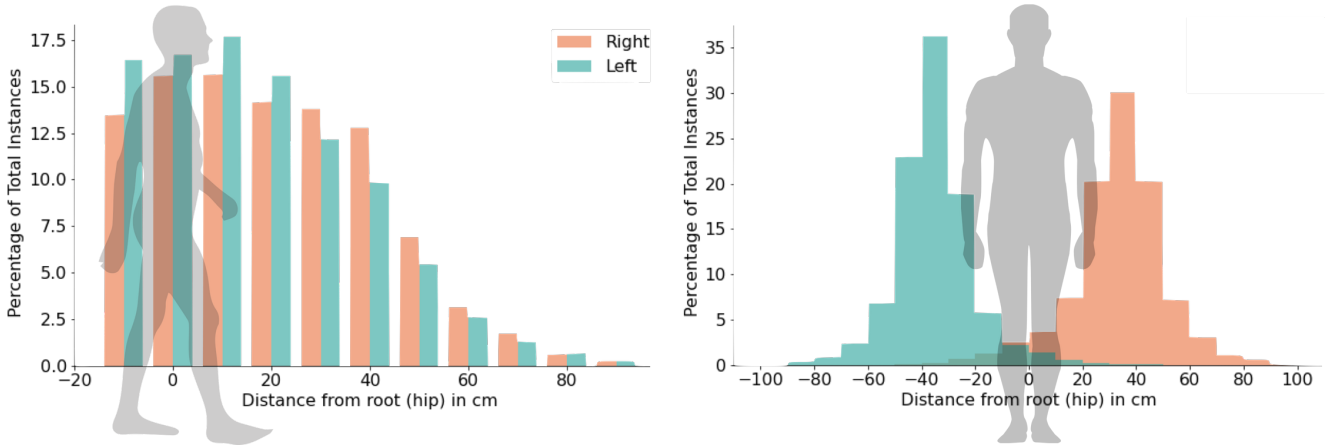

Figure 3: Left: Histogram showing the distribution of hand locations in front of the body (antero-posterior axis) during typical VR experiences. Right: Histogram showing the distribution of left and right hand locations along the mediolateral axis.

[30] have been explored for these purposes. PoseOnTheGo [6] demonstrated coarse estimation (as opposed to true tracking) of whole-body pose using a handheld smartphone by fusing camera, IMU and touchscreen data.

Most related to our work are computer vision approaches that track the wearer with one or more cameras. Researchers have explored instrumenting various body parts with cameras, for instance, Back-Hand-Pose [77] makes use of a wrist-worn fisheye camera for hand pose estimation. [10] instruments the shoe with a depth sensor for hand gesture recognition. Similar camera arrangements have also been explored in a pendant [63] and ring [13, 45, 80] form factor for hand and finger tracking. In terms of body tracking, OddEyeCam [32] makes use of a depth and wide angle RGB camera attached to a smartphone for tracking the shoulders of a user. Similarly, [25] and [29] make use of an ultra-wide fisheye camera mounted on the user's chest for 3D pose estimation. Shiratori et al. [60] placed 16 cameras on the body to capture the movement of joints as an inside-out tracking approach. Importantly, all of the previous approaches require additional cameras placed at locations that are not part of current consumer-grade VR hardware (headset and handheld controllers), whereas ControllerPose does not.

Most closely related to ControllerPose are body capture systems that augment existing VR hardware (i.e., no new accessories, no additional setup time). These include EgoCap [57], MeCap [2], Mo2Cap2 [79] and xR-EgoPose [71, 72], all shown in Figure 2. In these systems, one or more cameras operate roughly 7 to 25 cm from the face in order to get a reasonable view of the body. However, this design increases rotational inertia and applies a torque to the head due to gravity. Even at this distance, these system have very oblique views of the body that are prone to self-occlusion. As can be seen in the xR-EgoPose examples (Figure 2, far right), hands operating in front of the user (common in VR experiences) often block the view of the legs. Moreover, a user's stomach or breasts can also block the view (Figure 2, far right), as can items of clothing. We also note that the current industry trend is to make headsets thinner to reduce e.g., rotational inertia, which would increase body occlusion, making cameras on the headset less and less suitable for body pose tracking. Of course ControllerPose also suffers from occlusion, though from different sources, and so the two approaches seem complementary as we note in Future Work.

3 HAND LOCATION BACKGROUND STUDY

While there has been considerable research on the ergonomics of VR [55, 58, 75], we were unable to find useful statistics on where user's hands are located during typical VR experiences. To ground our assumptions, we ran a small motion capture study, in which five participants (mean age 21.0) played three popular titles for the Oculus Quest 2 [41] (one level each of Beat Saber [15], Superhot [66] and Pistol Whip [16]). When an app is running on the Quest 2, it is not possible to record the VR controller positions. Instead, we used a Kinect V2 placed 2.5 m in front of participants, and used the official SDK to create a basic program to record the 3D head, torso, hand and elbow positions at 30 FPS. Prior studies (benchmarking against a Vicon tracking system) have shown Kinect hand joint tracking error to be around 2.5 cm [8], more than sufficient for the purposes of this study. In total, we collected 55.2 minutes of playtime containing 99,353 body tracking instances for analysis.

The distribution of how far the hands operate in front of the body is shown in Figure 3. We found that 31.7% of the time, hands could be found held to the chest or resting by the side of the body. The other 68.3% of the time, the hands were operating in front of the user engaging with VR content, with a mean distance of 24.4 cm (SD=17.5). The horizontal distribution of the left and right hands is shown in Figure 3. On average, the left hand operates 35.2 cm (SD=14.5) from the body center, while the right arm operates 33.3 cm (SD=17.7) away. Interestingly, at least for the three VR apps we tried, the hands almost never crossed, and only rarely moved into the other hand's "air space".

More relevant to our problem domain is the mean Euclidean distance of the hand to the body center line (i.e., assuming the user is roughly centered in the controller camera's field of view, how far away would the camera be on average?). Our results show a mean Euclidean distance of 43.0 cm (SD=15.6) from camera to body. This

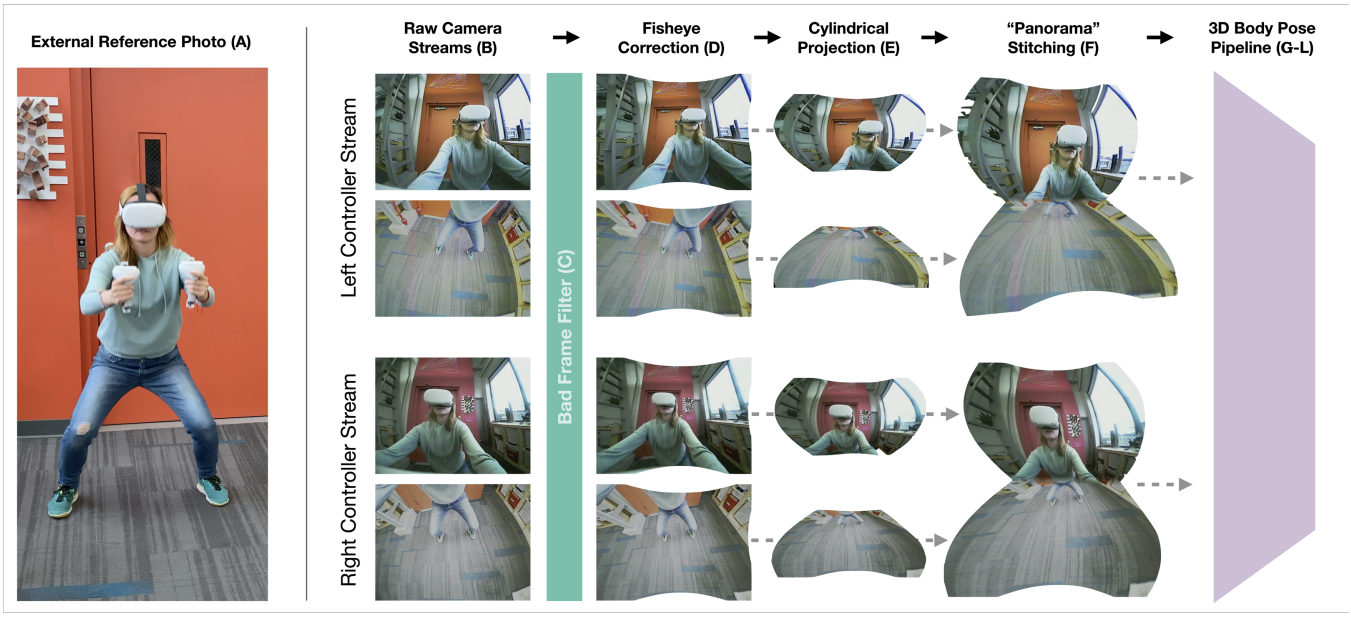

Figure 4: Camera compositing and unwarping pipeline: Our system ingests two live camera feeds, streamed from each of the two handheld controllers (B). Bad frames are detected and filtered (C). Good frames are corrected for fisheye distortion (D), followed by a cylindrical projection (E) and finally stitched together into a single "panorama" (F). We include an external photo for reference (A).

operational envelope was important in selecting cameras with a suitable field-of-view to capture the whole body.

4 IMPLEMENTATION

We now describe the various hardware and software components that work together to enable full-body pose tracking.

4.1 Hardware

As a test platform, we used a stock Oculus Quest 2 VR headset connected to a laptop (Intel i7 2.6 GHz, 16 GB, Nvidia RTX 2060 GPU) running Unity Apps via Oculus Link. We instrumented each hand controller with two wireless cameras – one placed on the upper ring and the other mounted to the bottom of the grip (Figure 1A). These locations are minimally occluded by the arm holding the controller. Both cameras [76] are wide-angle, providing a 120° vertical and 150° horizontal field of view at a resolution of 640×480. When combined in our later software pipeline, the composited field of view expands to 185° vertically (horizontal field of view remains the same). Both cameras are powered by a single 8.3 Wh LiPo battery providing around 2.5 hours of runtime. The four analog wireless cameras are read by a desktop computer (Intel Core i7 3.6 GHz, 16 GB, Nvidia 1080 Ti GPU) using USB receivers – the end-to-end (i.e., photon-to-motion) video latency is roughly 75 ms. We run our processing pipeline on the desktop, and the laptop tethered to the VR headset is only responsible for running Unity.

We emphasize that this apparatus is a proof of concept that facilitated rapid prototyping. We envision a commercial grade prototype with integrated cameras (perhaps even more than two). A bill of materials teardown of the Oculus Quest 1 [69] suggests all four Aptina camera modules in the headset, plus the Etron eSP770 camera controller IC, cost around $14.75 in commercial volumes, and it is likely that including cameras in the controllers would carry a similar cost.

Body pose processing could occur in one of two places. First, the low-resolution video could be streamed back to the headset where a more powerful CPU and GPU exist. The Oculus Quest 2 already has real-time hand pose tracking (21-joints × two hands); the body

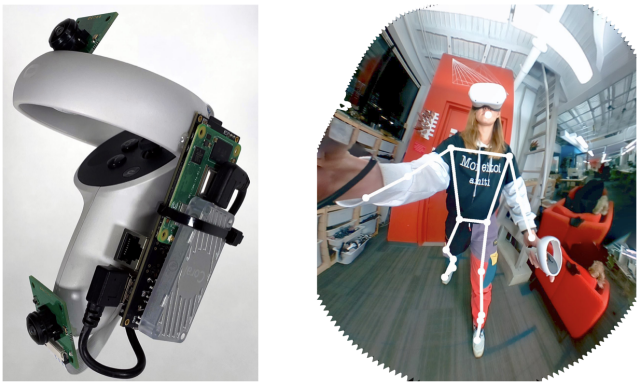

Figure 5: Left: Our second proof-of-concept controller that performs *on-board* body pose tracking at camera framerate. Right: An example stitched image from the two fisheye cameras with extracted body pose overlaid.

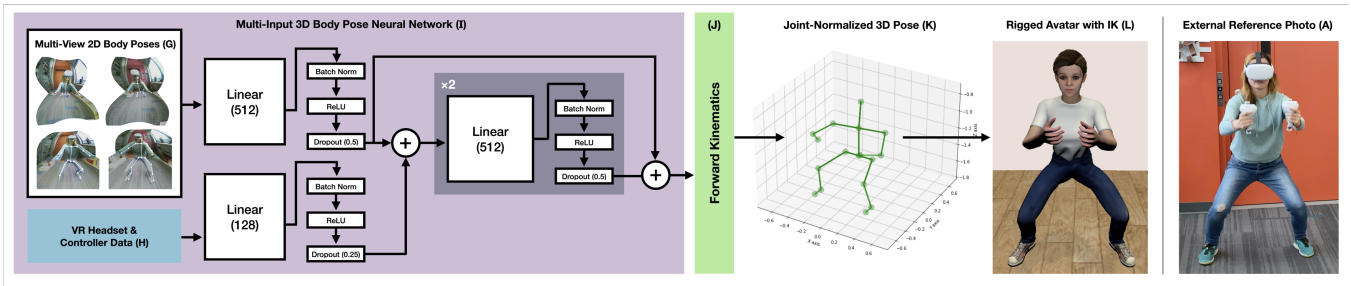

Figure 6: 3D pose estimation pipeline: Our system estimates 2D body pose for the left and right streams, as well as torso-normal-aligned views (G). This multi-view pose data is featured and combined with VR headset and controller position/rotation (H) as reported by the Oculus Quest 2's software. This multi-modal data is then fed into a multi-input neural network (I) with the depicted architecture, which outputs 3D angles for 17 joints. A forward kinematic pass (J) assembles the bone linkages into a joint-normalized 3D pose (K), which is then passed to Unity where an inverse kinematic solver poses an avatar (L) for interactive use. An external photo is provided for comparison (A).

is less complex in terms of keypoints (17 in our case) and could almost certainly run in realtime on the present headset hardware.

Alternatively, body pose processing could occur on-controller, with the processed skeletal data streamed back to the headset over the existing low-bandwidth wireless connection that is already streaming IMU, hand grip, joystick and button data. As a proof-of-concept, local-compute controller, we instrumented an Oculus Quest 2 controller with a StereoPi V2 board [64], two Arducam fisheye cameras (OV5647/LS-40180; 194°HFOV), Raspberry Pi Compute Module 4 [53] and Google Coral [18]. This hardware and an example stitched image can be seen in Figure 5. In the future, these components could be tightly integrated onto a single SOC. Our prototype controller performs on-board 2D pose extraction using Edge TPU PoseNet MobileNet V1 [51] at 29.6 FPS. The headset would then perform final computation using skeletal data from both controllers, which is computationally lightweight. For the rest of the paper, we only use and discuss our analog wireless camera controller implementation, as this was more compact and robust for user studies.

4.2 Compositing and Unwarping Pipeline

Figure 4 provides an overview of our compositing and unwarping pipeline. A reference external photo of a user is seen in Figure 4A. Our pipeline starts with the four raw camera feeds coming from the left and right controllers (Figure 4B). These analog transmissions can suffer from interference and occasional synchronization loss, resulting in unusable frames (roughly 1 in 50) that we filter (Figure 4C). Next, we correct each camera for fisheye distortion (Figure 4D). For this, we use OpenCV's fisheye camera model API [11], which requires a one-time checkerboard calibration procedure. On the corrected image, we then perform a cylindrical projection [36] (Fig 4E). Different projections preserve different aspects of an image (e.g., area, shape, distance, parallel lines). We found that cylindrical projection best preserved the relative proportions of users' bodies, especially at the region of stitching (but we note that many projections are applicable). We then proceed to stitch the two images together into a "panorama" with a gradient blend (Figure 4F). As the geometry between the cameras is fixed, a one time calibration is required to determine the stitch parameters. This composited image has a roughly 185° vertical by 150° horizontal field of view.

4.3 3D Pose Estimation Pipeline

Figure 6 provides an overview of our multi-view and multi-modal 3D pose estimation pipeline. First, using the output of our compositing and unwarping pipeline (Figure 4F), we extract 2D pose estimates of the user. For this, we use a 2D pose model built on the latest v1.7.0 release of OpenPose [12] outputting the following 17 keypoints: head, 2 × shoulders, 2 × elbow, 2 × hand, torso, mid hip, 2 × pelvis, 2 × knee, 2 × ankle, and 2 × foot. First, we produce two skeletons using the left and right streams (Figure 6G, top two inset images). Then, using the resulting shoulder and hip keypoints, we create torso-normal-aligned views using a four-point perspective transform. From this viewpoint, the lower body is more proportionally correct, and we use this to produce two more 2D pose estimates (left and right streams; Figure 6G, bottom two inset images). We note that it is not uncommon for one controller to have a good view of the user, while the other view is poor or occluded. In this case, only data for one or two skeletons will be produced. In cases where no skeletons are found, the pipeline ends here and awaits better frames. For each found skeleton, we compute unit direction vectors [4] between all pairs of the 17 joints resulting in 136 direction vectors, along with joint confidence values (17 values).

In addition to the multi-view 2D pose data, we supplement our input vector with the height of the headset, and the X/Y/Z position and X/Y/Z/W rotation (quaternion space) of both controllers with respect to the head (another 15 features), as reported by the Oculus Quest 2 software (Figure 6H). Our multi-modal data is then fed into a multi-input neural network - a multilayer neural network [37] with batch normalization [27], Rectified Linear Units (RELUs) [44], dropout [62] and residual connections [21] (architecture illustrated in Figure 6I).

Our model was pre-trained on data from nine users (independent from our user study participants) performing different poses while in VR. We used a Kinect v2 and Microsoft Kinect SDK for ground truth 3D pose capture (our evaluation used a similar ground-truthing scheme; see Section 5 for more details). This training

Figure 7: The twenty lower-body poses requested in our user study. Note that while holding each pose, both arms were systematically translated within a 69x51x22 cm volume in front of the user to provide a fully-crossed arm pose x leg pose capture procedure that was experimentally feasible.

dataset contained 70,316 body instances collected over a period of 147 minutes. During training, we use a batch size of 128 and update the weights using the Adam optimizer [33] with a learning rate of 0.001 and a decay of 5×10^{-6}. For our loss term, we use mean-squared error and train the model for 500 epochs. The network is implemented in TensorFlow and trained on an NVIDIA Titan V GPU.

Our model outputs 3D joint angles for the aforementioned 17 keypoints with respect to their parent (hip is the root) with 43 degrees of freedom. We then perform forward kinematics (Figure 6J) to resolve the pose in 3D Cartesian space (Figure 6N). As a final step, we pass our pose output through an inverse kinematic solver implemented using FinalIK [59] in Unity [19] (Figure 6L). This helps to correct unnatural poses and allows us to continue animating the avatar through brief periods of tracking loss. In Unity, we also apply basic logic to ground the feet to the VR environment floor. Specifically, if a foot keypoint is detected to be near the floor, we anchor it to the floor. Otherwise, a foot is assumed to be raised. We note that our prototype implementation currently only supports one foot being raised at a time, allowing for e.g., walking, but not running. However, this is not an innate limitation, as distance from the ground plane is tracked by the VR headset, and such data could be utilized in future versions to enable e.g., jumping.

4.4 Framerate and End-to-End Latency

Our full pipeline runs at 7.2 FPS (SD=0.67). We measured the motion-to-photon latency of our full stack by having a user perform a series of rapid and distinctive poses while being recorded with a camera at 240 FPS. In subsequent analysis, we found a mean latency of ~297 ms. As noted previously, it takes ~75 ms just to receive video frames that are the input to our pipeline. Other major sources of latency come from imaging unwarping/compositing (~63 ms), body pose estimation (~128 ms), network overhead (~8 ms) and Unity (which renders graphics and runs an IK solver; ~17 ms). In a commercial implementation, some sources of latency would be eliminated (e.g., network overhead), while other processes would be deeply optimized with DSP/ASIC hardware in the headset or controllers (e.g., image unwarping). The remaining machine learning components would run with hardware acceleration.

As one point of reference, the Oculus Quest 1 performs 21-joint hand pose tracking (times two hands) – also using a view composited from four headset cameras – at 30 FPS [20] (the newer Quest 2 runs at 60 FPS [49]). This demonstrates what commercial-level optimization can achieve, and there is no fundamental reason to suggest that similar performance is not possible with our approach. The latency numbers we report above are to document our unoptimized, proof-of-concept system, and should be considered a lowerbound.

5 EVALUATION

We recruited 8 participants (mean age 22.7) for a 30 minutes study, which paid $20 in compensation. After a brief orientation, participants were fitted with an Oculus Quest 2 running our study interface and given our prototype controllers to hold for the duration of the study. An area with a plain background was used. In

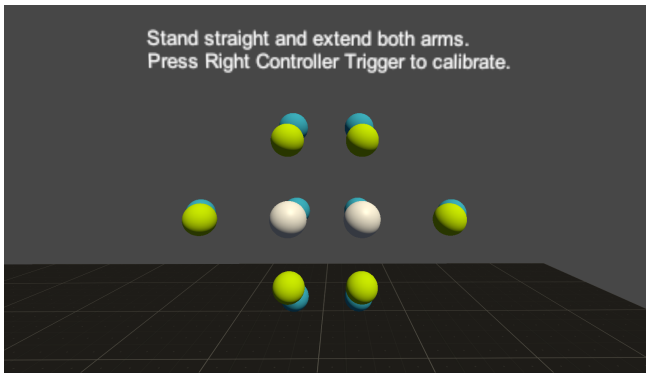

Figure 8: To help guide their arm movements through a consistent volume, participants were shown sixteen spheres arranged at two distances (arms fully extended and arms bent), arranged into triangular groupings for the left and right hands.

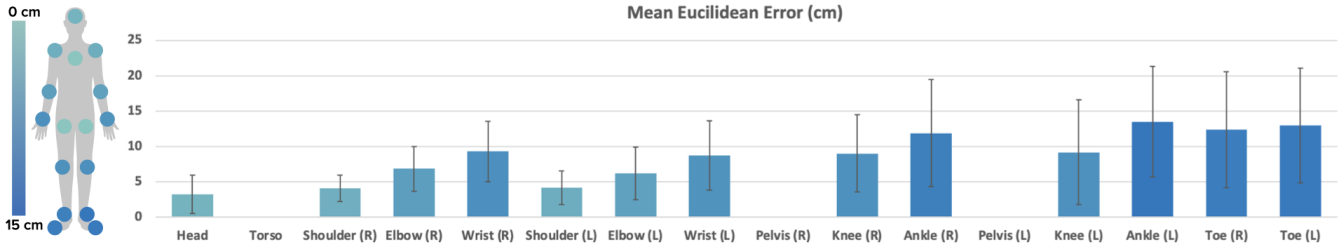

Figure 9: Mean 3D Euclidean error across the 17 body joints our system produces. Left: Heatmap of mean error across the body. Right: Bar chart of same data with error bars (standard deviation).

order to evaluate our system's pose tracking accuracy, we designed our procedure to capture a variety of body poses at a variety of controller positions, in a reasonably standardized way, that approximated the interactive volume of VR controller input informed by our earlier study.

5.1 Procedure

Upon wearing the VR headset, participants were presented with welcome text that requested they fully extend their arms in front of their bodies and pull the trigger button. This was used to calibrate arm length, after which four groups of four color-coded spheres appeared in front of the user (locked to the front of the body, occupying a fixed volume of 69 × 51 × 22 cm; Figure 8). Each group consisted of a sphere located straight ahead, at head height, at stomach height and to the side of the user at shoulder height. Each arm had groups at two distances; one for arms fully-outstretched, and another with the elbows bent, but not touching the body. Spheres turned red when intersected by the user's hand for visual confirmation. The spheres acted as a visual guide – participants were told to move one controller at a time between these four points, pausing momentarily at each sphere, completing a roughly triangular motion (i.e., straight, up, side, down, back to straight) taking ~3 seconds. We observed a mean translation speed of 1.1 m/s. Once a participant had translated one hand through a grouping (e.g., left arm at close distance), they would move to the next grouping, until all four groupings were completed. Note that during these translations, as well as when moving between sphere groups, our pipeline continually captured and processed data at 8 FPS – the task was simply designed to capture a wide variety of controller positions in front of the user.

To capture a variety of body poses, participants were asked to perform twenty different poses, seen in Figure 7. These were requested one at a time, and then inside each pose trial, the experimenter verbally instructed participants to translate the controllers through the aforementioned four sphere groups (procedure above). Eight poses had participants lift one leg from the floor. In piloting, we found that it was challenging to balance on one leg while performing the controller translations (as it shifted the user's center of mass), and so for these poses in our study, participants were permitted to drop their legs down to the floor to regain balance as needed, and then resume the controller translation when balancing again. Since our controllers continuously captured data, it meant these trials appeared more akin to a series of repeated high steps than a statically held pose. However, the data was still perfectly well suited for evaluating pose accuracy. In total, the above procedure captured 33,934 data instances (which include data from both controllers) over a cumulative period of 68 minutes.

5.2 Ground Truth Body Pose

In order to assess our system's pose tracking accuracy, it was necessary to capture a ground truth body pose. For this, we used a Kinect V2 placed 2 m in front of participants (similar to our hand location background study). The official SDK outputs 25 body joints [84] (which have a direct mapping to the 17 joints we compute) in real-world X/Y/Z coordinates with respect to the sensor. While not as accurate as a professional-grade motion capture system, we found it to be more than sufficient for the types of full-body poses we focus on (see e.g., [8] for a rigorous analysis of Kinect V2 joint tracking accuracy; at our 2 m sensing distance, mean joint error is around 4.5 cm), and obviated the need to place tracking markers on participants. We wrote a basic program to log a participant's body pose at the Kinect V2's native 30 FPS, which was timestamped for later comparison to our system's pose output.

5.3 Results and Dataset

Our study data had to be normalized to compensate for variations in user position and height, as our pipeline does not track a user's real-world position nor the real-world size of a user. Thus, to standardize our Kinect-captured ground truth data with our ControllerPose output, we use the hip center as the body origin (i.e., 0,0,0) and orient the body such that the two pelvis keypoints are facing directly forwards. We standardize both datasets (Kinect and ControllerPose) to a constant body size of 1.8 m tall. This body-size-normalization step means that smaller/larger participants have equal weighting in terms of error. As noted earlier, our machine learning model was pre-trained on data from nine users who were not recruited as participants in this study (avoiding a train/test bias).

Overall, our pipeline had a mean 3D Euclidean joint error of 6.98 cm (SD=4.2) with respect to the Kinect-captured ground truth. Accuracy broken out by joint can be seen in Figure 9. We note that hip and torso points have very low error, but this is because they directly attach to the mid-hip root node, which both our pipeline and the Kinect ground truth use as the body origin. Excluding those three points (leaving all of the limb joints), our mean 3D Euclidean joint error is 8.59 cm (SD = 5.2). Even still, this exceeded our expectations, and suggests that gross body movements can be faithfully

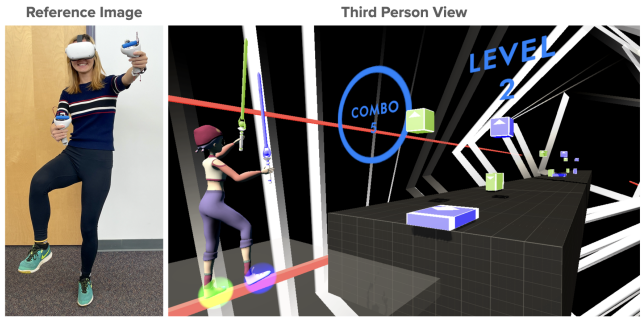

Figure 10: Feet Saber is an adaptation of *Beat Saber*, and allows the user to employ their feet in conjunction with their hands to kick and stomp incoming blocks.

captured. We also evaluated our system's spatial accuracy without using positional data reported by the VR headset and controllers (i.e., only camera-derived pose data is used as input to the model), and found error increased by 25.5% to 8.76 cm (SD=3.2).

As one point of comparison, we ran CenterHMR [65] -– a state-of-the-art 3D body pose estimation model -– and fed it recordings of our composited camera views post hoc. We found that CenterHMR had a mean joint error of 24.6 cm (SD=6.8) compared to ground truth. Note that CenterHMR has not been trained on data from our particular camera viewpoints, hence the comparisons are provided for reference only. In short, the unique viewpoints of our system, along with inherent distortions, require a custom model to perform well. However, it seems likely that a custom 3D model trained on copious data from our unique camera views would achieve even stronger performance than our present pipeline.

6 EXAMPLE APPLICATIONS

There are a myriad of applications that can make use of the full-body tracking (see Related Work) and many game titles that used Kinect's full body pose. To convey the utility of ControllerPose, we created seven demo experiences that fall into three popular categories of VR applications: games, social apps, and health. Please also see the Video Figure.

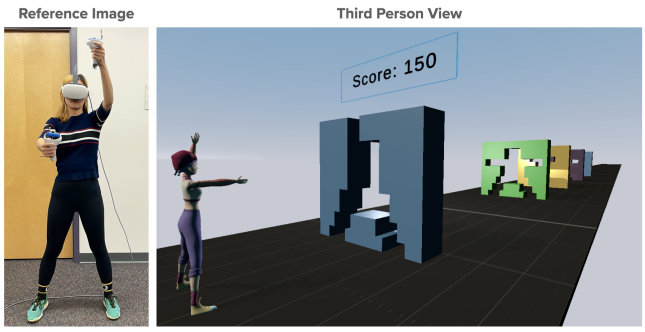

Figure 11: We created a human Tetris game, where users must assume the correct full-body pose to pass obstacles.

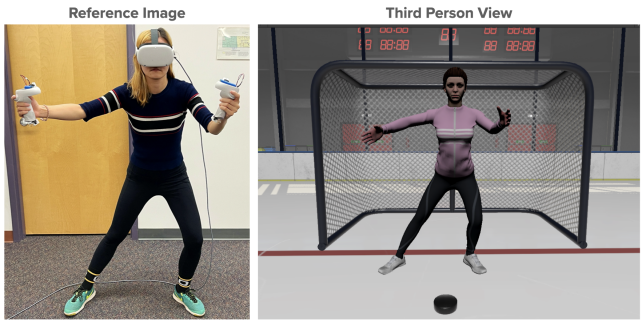

Figure 12: Our hockey goalie game requires users to use their limbs and body to block incoming shots.

6.1 Games

Games are presently the most popular category of application in virtual reality. SteamVR alone has 100+ gaming titles, along with games such as *Island 359*, *Neos VR* and *Blade and Sorcery* which have support for full body tracking and physics. To illustrate the use of our system's full-body capabilities, we've adapted one of the most popular virtual reality games to date - *Beat Saber* - and created Feet Saber (see 10). The gameplay is similar to the original game, but users can now also use their legs to stomp and kick blocks on the floor plane. For our second game example, we created a human Tetris game (Figure 11), where users must assume the correct full-body pose to pass obstacles and progress in the game. As a final example, we created a hockey goalie game (Figure 12) in which the user must utilizes every limb to deflect incoming pucks. While not a full game experience, we note that our soccer avatar (Figure 1) has the ability to kick.

6.2 Social Apps

Expressing oneself physically (e.g., hand gestures, body language, tapping one's feet impatiently) is a crucial component of social experiences. Today's consumer VR systems and the social VR apps they run generally only animate the head and arms. Not only does this limit social bandwidth, but users can experience reduced embodiment and immersion when they look down at their virtual lower body. ControllerPose can power avatars that are faithful

Figure 13: With ControllerPose, users can better express themselves in social VR experiences.

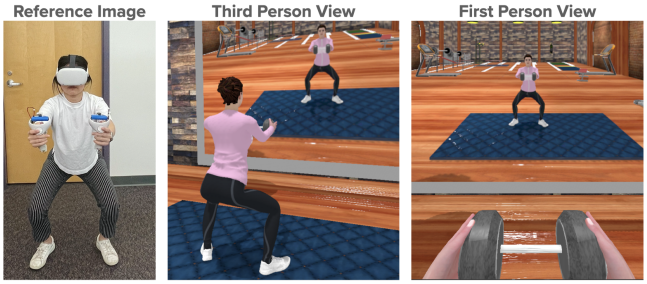

Figure 14: ControllerPose could be used to automatically detect exercises and track rep counts (e.g., squats). In this demo, users can see their reflection in a virtual gym mirror.

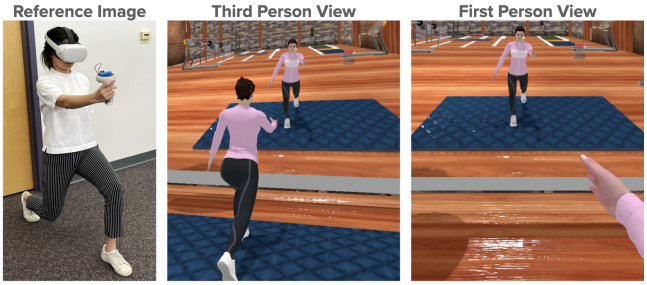

Figure 16: ControllerPose is used to count and rate lunges in this exercise demo app.

to users' real-world, full-body pose (Figure 13), as well as enable representative reflections and shadows.

6.3 Training, Exercise and Rehabilitation

ControllerPose can also enable novel applications and experiences in health, fitness and rehabilitation — apps where capturing a user's pose is immensely valuable. For example, as part of a rehabilitation regimen, an app could request the user perform balancing exercises to evaluate progress (Figure 15). ControllerPose could also be used to track physical activity and record rep counts for exercises such as squats (Figure 14) and lunges (Figure 16).

7 LIMITATIONS AND FUTURE WORK

While we believe ControllerPose offers a new and practical way to digitize the body – especially the lower body – it nonetheless has pros and cons like any technical approach. In this section we review key weaknesses and avenues of future work.

7.1 Limitations of a CV Approach

First and foremost are limitations of a computer vision approach operating from a handheld controller. Chief among these is the fact our system cannot resolve a good view of the body if the hands are too close to the user, and certainly if the arms are resting by the side of the body. In these cases, it is impossible to get a full view of the body with our current camera placement (Figure 17, left image set). However, if an additional camera was added to capture the legs (Figure 17, right image set), it seems possible to estimate body pose even from these disadvantaged views.

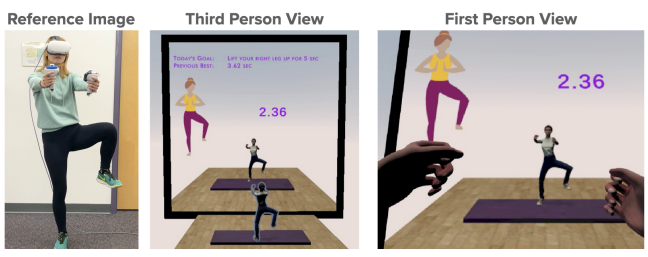

Figure 15: Balance training could be part part of a rehabilitation regimen.

It is also important to note that while computer-vision-based pose models have made tremendous strides in recent years, they are still very much an estimation. Add to this the low resolution of our cameras, and it seems unlikely such an approach could ever offer the millimeter-level tracking accuracy seen in professional optical motion capture systems (e.g., Vicon, OptiTrack). Perhaps a more achievable goal is reaching parity with battery-powered worn IMU systems, which must contend with other issues such as drift. That said, centimeter-level tracking accuracy may be more than enough for most VR experiences, especially those employing gross movements, such as walking, running, kicking, climbing, leaning, crouching, etc.

Even if higher spatial accuracies can be achieved, similar systems will have to contend with occlusion, both from the arms holding the controllers and from clothing elements such as sleeves and wrist jewelry. Also, at certain controller angles and in certain body poses, limbs can occlude one another (examples in Figure 18). We also found that some clothing can degrade the accuracy of our body pose estimation, such as baggy coats (Figure 19), but we note this is likely an issue with model training data and not an innate limitation. We also encountered cases where a user's shoes or pants were close to the color of the floor, reducing contrast to the point of intermittent tracking loss.

Finally, environmental factors can also impact system accuracy. For instance, we need good lighting as our cameras use the visible spectrum. Harsh rear lighting is particularly bad as our cameras auto-adjust their exposure levels. In the future, infrared cameras with illuminators could be considered to boost robustness. We also found that busy backgrounds can very occasionally lead to catastrophic errors in detected pose, but this was surprisingly rare – a testament to the progress made by deep learning computer vision researchers in the past few years. As described previously, our system selects a winning pose from a set of four using joint confidence scores. When busy backgrounds do cause keypoint misalignment, this generally manifests as lower confidence values, and so they are only very rarely propagated up to the live VR avatar pose.

7.2 Performance

Our current system is a proof of concept, built to assess the general feasibility of our controller-based approach. The framerate, latency and other performance characteristics are not comparable to that of a commercially engineered and optimized implementation. Put

Figure 17: Example views when the controllers/hands are resting by the side of the body. Our current cameras (green & cyan) do not capture the body, and so our pipeline fails. In the future, cameras with larger fields of view could be utilized, or one or more cameras could be added to the controller. We captured examples of what these views would look like (purple & red), and it seems likely that pose models could detect the body even from this disadvantaged location in future work.

simply, our performance stats should be considered the floor of performance, not the ceiling. Nonetheless, our proposal to add a new deep-learning-based, body-tracking process to VR headsets will inevitably add extra compute overhead, which in turn will impact battery runtime. Rather than trying to extrapolate a number, it is more reliable to find an existing commercial analog of our software pipeline. A good exemplar is the Oculus Quest 2's hand tracking feature. Like our system, this composites views from four cameras (in the headset) and tracks 21 keypoints on each hand (42 in total). This runs in realtime on the $299 hardware at 60 FPS [20, 49], concurrent with a main application consuming most of the CPU/GPU. This impressive level of performance is only possible with clever uses of hardware acceleration (e.g., the onboard Etron eSP770 camera controller performs all the panorama warping/stitching). The main Qualcomm Snapdragon XR2 chipset offers hardware accelerated AI features through its Hexagon DSP. Amazingly, this AI processing hardware is 11x faster than the predecessor Oculus Quest 1 [56] released only 16 months earlier. In general, mobile-grade hardware continues to make incredible strides in compute performance, and the ability to add new pipelines such as ours will become increasingly feasible with negligible impact.

7.3 Expanded Range of Poses

Lastly, we note that during our development process, we serendipitously noticed our system recognizing (at least in part) many unusual poses, at least in the context of a VR experience. Examples (seen in Figure 20 bottom row) include sitting with legs crossed on a couch, kneeling on the floor, standing with one foot in front of the other and sitting on the floor while tucking one's body into a loose fetal position (à la rowing). Although our pipeline was largely able to detect this body pose in the composited camera, it failed at later stages, chiefly the 3D pose estimation and the inverse kinematic posing of the avatar. However, we believe these sorts and other failure cases (Figure 20 top and middle row) of poses should be achievable in future work.

7.4 Body Pose Sensor Fusion

Our system already fuses computer-vison-derived body keypoints with IMU data from the headset and controllers. However, there are other sources of data future systems could leverage to further improve accuracy and robustness to e.g., occlusion. For instance, the cameras found in VR headsets for visual odometry (and sometimes hand tracking) are likely to be a great value, at least for the upper

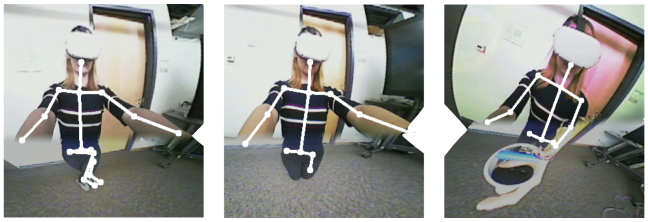

Figure 18: Pose estimation failure cases due to self-occlusion. From left to right: occlusion of lower body due to crossed legs, occlusion of feet due to kneeling and occlusion of body by the controller itself.

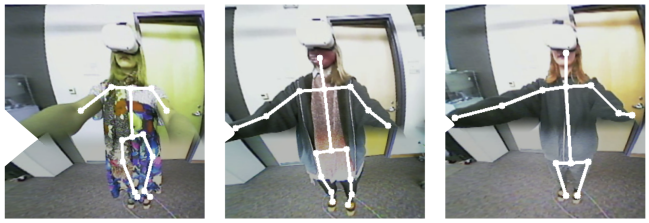

Figure 19: Pose mis-predictions due to clothing occlusion. From left to right: Incorrect foot prediction due to skirt, Incorrect hip estimation due to baggy shirt and missing ankles due to large jacket.

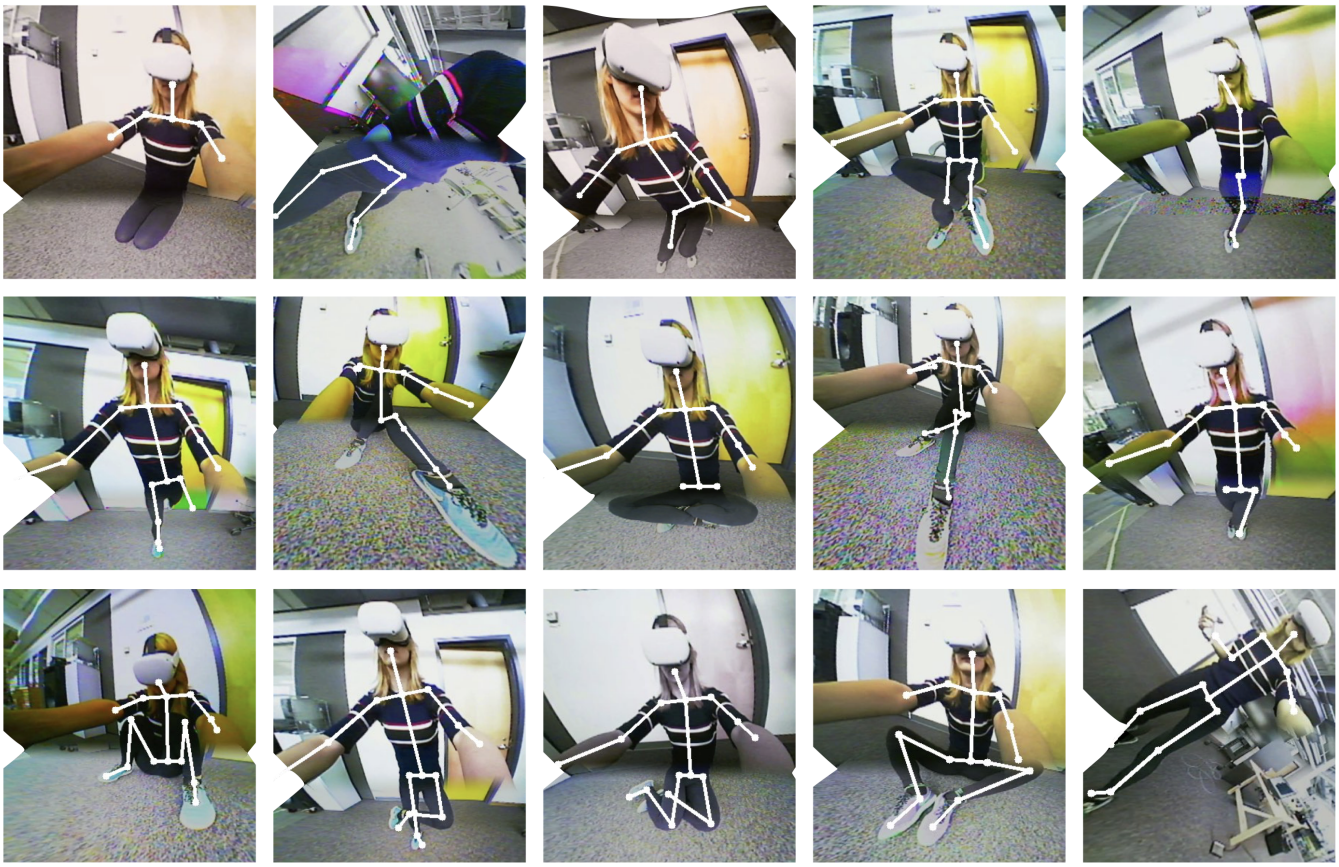

Figure 20: Apart from the range of motions showcased in our user study and demo apps, we test ControllerPose's pipeline on other unusual poses. Top row: cases where pose estimation failed. Middle row: cases where pose estimation partially worked. Bottom row: cases where pose estimation works well and finds most of the body joints correctly.

body, and also in cases where the feet project out in front of the body, such as when walking or lunging. If the user is wearing a smartwatch, IMU data could be used to better pose the wrist joint, instead of treating the hand as a linear extension of the forearm. Sound, captured by microphones in the headset, might also offer locomotion cues, such as velocity of footfalls (e.g., stomping vs. tip-toeing).

8 CONCLUSION

We have shown how integrating cameras into VR controllers provides a new perspective on body pose capture in VR. While not as spatially accurate as external tracking systems or worn IMUs, we believe the practicality of our system makes a unique contribution in the literature and could bring full-body embodiment to many more VR users. Nonetheless, our system's mean joint error of 7.34 cm should be more than sufficient for a wide range of VR experiences. To convey the performance and potential of our system, we built a series of examplary applications, most focusing on lower body interactions, as this is missing from a vast majority of VR systems today. We concluded with a discussion of limitations, both inherent and temporary, highlighting avenues for future work.

REFERENCES

[1] Karan Ahuja, Mayank Goel, and Chris Harrison. 2020. BodySLAM: Opportunistic User Digitization in Multi-User AR/VR Experiences. In *Symposium on Spatial User Interaction*. 1–8.

[2] Karan Ahuja, Chris Harrison, Mayank Goel, and Robert Xiao. 2019. MeCap: Whole-Body Digitization for Low-Cost VR/AR Headsets. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology* (New Orleans, LA, USA) *(UIST '19)*. Association for Computing Machinery, New York, NY, USA, 453–462. https://doi.org/10.1145/3332165.3347889

[3] Karan Ahuja, Rahul Islam, Varun Parashar, Kuntal Dey, Chris Harrison, and Mayank Goel. 2018. Eyespyvr: Interactive eye sensing using off-the-shelf, smartphone-based vr headsets. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 2, 2 (2018), 1–10.

[4] Karan Ahuja, Dohyun Kim, Franceska Xhakaj, Virag Varga, Anne Xie, Stanley Zhang, Jay Eric Townsend, Chris Harrison, Amy Ogan, and Yuvraj Agarwal. 2019. EduSense: Practical classroom sensing at Scale. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 3, 3 (2019), 1–26.

[5] Karan Ahuja, Andy Kong, Mayank Goel, and Chris Harrison. 2020. Direction-of-Voice (DoV) Estimation for Intuitive Speech Interaction with Smart Devices Ecosystems *(UIST '20)*. Association for Computing Machinery, New York, NY, USA, 1121–1131. https://doi.org/10.1145/3379337.3415588

[6] Karan Ahuja, Sven Mayer, Mayank Goel, and Chris Harrison. 2021. Pose-on-the-Go: Approximating User Pose with Smartphone Sensor Fusion and Inverse Kinematics. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 1–12.

[7] Karan Ahuja, Eyal Ofek, Mar Gonzalez-Franco, Christian Holz, and Andrew D Wilson. 2021. CoolMoves: User Motion Accentuation in Virtual Reality. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 5, 2 (2021), 1–23.

[8] Justin Amadeus Albert, Victor Owolabi, Arnd Gebel, Clemens Markus Brahms, Urs Granacher, and Bert Arnrich. 2020. Evaluation of the pose tracking performance of the azure kinect and kinect v2 for gait analysis in comparison with a gold standard: A pilot study. *Sensors* 20, 18 (2020), 5104.

[9] Rıza Alp Güler, Natalia Neverova, and Iasonas Kokkinos. 2018. Densepose: Dense human pose estimation in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '18)*. IEEE, 7297–7306. https://doi.org/10.1109/CVPR.2018.00762

[10] Gilles Bailly, Jörg Müller, Michael Rohs, Daniel Wigdor, and Sven Kratz. 2012. Shoesense: a new perspective on gestural interaction and wearable applications. In *Proceedings of the SIGCHI conference on human factors in computing systems*. 1239–1248.

[11] G. Bradski. 2000. The OpenCV Library. *Dr. Dobb's Journal of Software Tools* (2000).

[12] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. 2017. Realtime Multi-person 2D Pose Estimation Using Part Affinity Fields. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR '17)*. IEEE, 7291–7299. https://doi.org/10.1109/CVPR.2017.143

[13] Liwei Chan, Yi-Ling Chen, Chi-Hao Hsieh, Rong-Hao Liang, and Bing-Yu Chen. 2015. Cyclopsring: Enabling whole-hand and context-aware interactions through a fisheye ring. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*. 549–556.

[14] Ke-Yu Chen, Shwetak N Patel, and Sean Keller. 2016. Finexus: Tracking precise motions of multiple fingertips using magnetic sensing. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. 1504–1514.

[15] Beat Games. 2021. BeatSaber VR Game. https://beatsaber.com

[16] CloudHead Games. 2021. Pistol Whip. https://cloudheadgames.com/pistol-whip

[17] Oliver Glauser, Shihao Wu, Daniele Panozzo, Otmar Hilliges, and Olga Sorkine-Hornung. 2019. Interactive hand pose estimation using a stretch-sensing soft glove. *ACM Transactions on Graphics (TOG)* 38, 4 (2019), 1–15.

[18] Google. 2022. Coral. https://coral.ai

[19] John K Haas. 2014. A history of the unity game engine. (2014).

[20] Shangchen Han, Beibei Liu, Randi Cabezas, Christopher D Twigg, Peizhao Zhang, Jeff Petkau, Tsz-Ho Yu, Chun-Jung Tai, Muzaffer Akbay, Zheng Wang, et al. 2020. MEgATrack: monochrome egocentric articulated hand-tracking for virtual reality. *ACM Transactions on Graphics (TOG)* 39, 4 (2020), 87–1.

[21] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.

[22] HTC. 2020. VIVE. https://www.vive.com

[23] HTC. 2020. VIVE Accessory Trackers. https://www.vive.com/us/accessory/vive-tracker

[24] Yinghao Huang, Manuel Kaufmann, Emre Aksan, Michael J Black, Otmar Hilliges, and Gerard Pons-Moll. 2018. Deep inertial poser: Learning to reconstruct human pose from sparse inertial measurements in real time. *ACM Transactions on Graphics (TOG)* 37, 6 (2018), 1–15.

[25] Dong-Hyun Hwang, Kohei Aso, Ye Yuan, Kris Kitani, and Hideki Koike. 2020. MonoEye: Multimodal Human Motion Capture System Using A Single Ultra-Wide Fisheye Camera. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*. 98–111.

[26] Intel Corporation. 2020. RealSense. https://www.intelrealsense.com/

[27] Sergey Ioffe and Christian Szegedy. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*. PMLR, 448–456.

[28] Yasha Iravantchi, Yang Zhang, Evi Bernitsas, Mayank Goel, and Chris Harrison. 2019. Interferi: Gesture sensing using on-body acoustic interferometry. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. 1–13.

[29] Hao Jiang and Kristen Grauman. 2017. Seeing invisible poses: Estimating 3d body pose from egocentric video. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 3501–3509.

[30] Haojian Jin, Jingxian Wang, Zhijian Yang, Swarun Kumar, and Jason Hong. 2018. Rf-wear: Towards wearable everyday skeleton tracking using passive rfids. In *Proceedings of the 2018 ACM International Joint Conference and 2018 International Symposium on Pervasive and Ubiquitous Computing and Wearable Computers*. 369–372.

[31] David Kim, Otmar Hilliges, Shahram Izadi, Alex D Butler, Jiawen Chen, Iason Oikonomidis, and Patrick Olivier. 2012. Digits: freehand 3D interactions anywhere using a wrist-worn gloveless sensor. In *Proceedings of the 25th annual ACM symposium on User interface software and technology*. 167–176.

[32] Daehwa Kim, Keunwoo Park, and Geehyuk Lee. 2020. OddEyeCam: A Sensing Technique for Body-Centric Peephole Interaction Using WFoV RGB and NFoV Depth Cameras. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*. 85–97.

[33] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *ICLR (Poster)*. http://arxiv.org/abs/1412.6980

[34] Elena Kokkinara, Mel Slater, and Joan López-Moliner. 2015. The effects of visuomotor calibration to the perceived space and body, through embodiment in immersive virtual reality. *ACM Transactions on Applied Perception (TAP)* 13, 1 (2015), 3.

[35] George Alex Koulieris, Kaan Akşit, Michael Stengel, Rafał K Mantiuk, Katerina Mania, and Christian Richardt. 2019. Near-eye display and tracking technologies for virtual and augmented reality. In *Computer Graphics Forum*, Vol. 38. Wiley Online Library, 493–519.

[36] Gyeong-il Kweon and Young-ho Choi. 2010. Image-processing based panoramic camera employing single fisheye lens. *Journal of the Optical Society of Korea* 14, 3 (2010), 245–259.

[37] Julieta Martinez, Rayat Hossain, Javier Romero, and James J Little. 2017. A simple yet effective baseline for 3d human pose estimation. In *Proceedings of the IEEE International Conference on Computer Vision*. 2640–2649.

[38] Antonella Maselli and Mel Slater. 2014. Sliding perspectives: dissociating ownership from self-location during full body illusions in virtual reality. *Frontiers in human neuroscience* 8 (2014), 693.

[39] Dushyant Mehta, Srinath Sridhar, Oleksandr Sotnychenko, Helge Rhodin, Mohammad Shafiei, Hans-Peter Seidel, Weipeng Xu, Dan Casas, and Christian Theobalt. 2017. Vnect: Real-time 3d human pose estimation with a single rgb camera. *ACM Transactions on Graphics (TOG)* 36, 4 (2017), 1–14.

[40] Meta Motion. 2018. Gypsy Motion Capture System. http://metamotion.com/gypsy/gypsy-motion-capture-system.htm

[41] Meta Technologies LLC. 2020. Oculus Quest. https://www.oculus.com/quest

[42] Meta Technologies LLC. 2020. Oculus Rift. https://www.oculus.com/rift-s/

[43] Damien Michel, Ammar Qammaz, and Antonis A Argyros. 2017. Markerless 3d human pose estimation and tracking based on rgbd cameras: an experimental evaluation. In *Proceedings of the 10th International Conference on PErvasive Technologies Related to Assistive Environments*. 115–122.

[44] Vinod Nair and Geoffrey E Hinton. 2010. Rectified linear units improve restricted boltzmann machines. In *Icml*.

[45] Suranga Nanayakkara, Roy Shilkrot, Kian Peen Yeo, and Pattie Maes. 2013. EyeRing: a finger-worn input device for seamless interactions with our surroundings. In *Proceedings of the 4th Augmented Human International Conference*. 13–20.

[46] NaturalPoint Inc. 2020. OptiTrack. http://optitrack.com

[47] Evonne Ng, Donglai Xiang, Hanbyul Joo, and Kristen Grauman. 2020. You2me: Inferring body pose in egocentric video via first and second person interactions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 9890–9900.

[48] Northern Digital Inc. 2020. trakSTAR. https://www.ndigital.com/msci/products/drivebay-trakstar

[49] Oculus. 2020. Oculus Hand Tracking. https://developer.oculus.com/blog/high-frequency-hand-tracking

[50] OpenNI. 2020. OpenNI. https://structure.io/openni

[51] George Papandreou, Tyler Zhu, Liang-Chieh Chen, Spyros Gidaris, Jonathan Tompson, and Kevin Murphy. 2018. Personlab: Person pose estimation and instance segmentation with a bottom-up, part-based, geometric embedding model. In *Proceedings of the European Conference on Computer Vision (ECCV '18)*. 269–286. https://doi.org/10.1007/978-3-030-01264-9_17

[52] Mathias Parger, Joerg H. Mueller, Dieter Schmalstieg, and Markus Steinberger. 2018. Human Upper-Body Inverse Kinematics for Increased Embodiment in Consumer-Grade Virtual Reality *(VRST '18)*. Association for Computing Machinery, New York, NY, USA, Article 23, 10 pages. https://doi.org/10.1145/3281505.3281529

[53] Raspberry Pi. 2022. Compute Module 4. https://www.raspberrypi.com/products/compute-module-4/?variant=raspberry-pi-cm4001000

[54] Polhemus. 2020. Polhemus Motion Capture System. https://polhemus.com/case-study/detail/polhemus-motion-capture-system-is-used-to-measure-real-time-motion-analysis

[55] Charles Pontonnier, Georges Dumont, Asfhin Samani, Pascal Madeleine, and Marwan Badawi. 2014. Designing and evaluating a workstation in real and virtual environment: toward virtual reality based ergonomic design sessions. *Journal on Multimodal User Interfaces* 8, 2 (2014), 199–208.

[56] Qualcomm. [n.d.]. *Qualcomm Snapdragon XR2 Platform Commercially Debuts in Oculus Quest 2.* https://www.qualcomm.com/news/releases/2020/09/16/qualcomm-snapdragon-xr2-platform-commercially-debuts-oculus-quest-2

[57] Helge Rhodin, Christian Richardt, Dan Casas, Eldar Insafutdinov, Mohammad Shafiei, Hans-Peter Seidel, Bernt Schiele, and Christian Theobalt. 2016. EgoCap: Egocentric Marker-Less Motion Capture with Two Fisheye Cameras. *ACM Trans. Graph.* 35, 6, Article 162 (Nov. 2016), 11 pages. https://doi.org/10.1145/2980179.2980235

[58] G Riva and G Mantovani. 1999. The ergonomics of virtual reality: human factors in developing clinical-oriented virtual environments. In *Medicine meets virtual reality*. IOS Press, 278–284.

[59] Root Motion. 2020. FINAL IK - VRIK Solver Locomotion. http://www.root-motion.com/finalikdox/html/page16.html

[60] Takaaki Shiratori, Hyun Soo Park, Leonid Sigal, Yaser Sheikh, and Jessica K Hodgins. 2011. Motion capture from body-mounted cameras. In *ACM SIGGRAPH 2011 papers*. 1–10.

[61] SONY. 2020. PlayStationVR. https://www.playstation.com/en-us/ps-vr

[62] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research* 15, 1 (2014), 1929–1958.

[63] Thad Starner, Jake Auxier, Daniel Ashbrook, and Maribeth Gandy. 2000. The gesture pendant: A self-illuminating, wearable, infrared computer vision system for home automation control and medical monitoring. In *Digest of Papers. Fourth International Symposium on Wearable Computers*. IEEE, 87–94.

[64] StereoPi. 2022. StereoPi V2. https://www.stereopi.com/v2

[65] Yu Sun, Qian Bao, Wu Liu, Yili Fu, Michael J Black, and Tao Mei. 2021. Monocular, one-stage, regression of multiple 3d people. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 11179–11188.

[66] Superhot. 2021. Superhot VR. https://superhotgame.com

[67] Ivan E. Sutherland. 1968. A Head-Mounted Three Dimensional Display. In *Proceedings of the December 9-11, 1968, Fall Joint Computer Conference, Part I* (San Francisco, California) *(AFIPS '68 (Fall, part I))*. Association for Computing Machinery, New York, NY, USA, 757–764. https://doi.org/10.1145/1476589.1476686

[68] Jochen Tautges, Arno Zinke, Björn Krüger, Jan Baumann, Andreas Weber, Thomas Helten, Meinard Müller, Hans-Peter Seidel, and Bernd Eberhardt. 2011. Motion reconstruction using sparse accelerometer data. *ACM Transactions on Graphics (ToG)* 30, 3 (2011), 1–12.

[69] Techinsights. 2021. Bill of Materials of the Oculus Quest MH-B VR Headset. https://www.techinsights.com/products/bom-1905-810

[70] Justus Thies, Michael Zollhöfer, Marc Stamminger, Christian Theobalt, and Matthias Nießner. 2018. FaceVR: Real-time gaze-aware facial reenactment in virtual reality. *ACM Transactions on Graphics (TOG)* 37, 2 (2018), 1–15.

[71] Denis Tome, Thiemo Alldieck, Patrick Peluse, Gerard Pons-Moll, Lourdes Agapito, Hernan Badino, and Fernando De la Torre. 2020. SelfPose: 3D Egocentric Pose Estimation from a Headset Mounted Camera. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2020), 1–1. https://doi.org/10.1109/TPAMI.2020.3029700

[72] Denis Tome, Patrick Peluse, Lourdes Agapito, and Hernan Badino. 2019. xr-egopose: Egocentric 3d human pose from an hmd camera. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV '19)*. IEEE, 7728–7738. https://doi.org/10.1109/ICCV.2019.00782

[73] Vicon Motion Systems Ltd. 2020. Vicon. https://vicon.com

[74] Daniel Vlasic, Rolf Adelsberger, Giovanni Vannucci, John Barnwell, Markus Gross, Wojciech Matusik, and Jovan Popović. 2007. Practical motion capture in everyday surroundings. *ACM transactions on graphics (TOG)* 26, 3 (2007), 35–es.

[75] Johann Wentzel, Greg d'Eon, and Daniel Vogel. 2020. Improving virtual reality ergonomics through reach-bounded non-linear input amplification. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. 1–12.

[76] Wolfwhoop. 2021. Wolfwhoop FPV Cameras. https://www.amazon.com/Wolfwhoop-WT05-Transmitter-Antenna-Quadcopter/dp/B06XJMQQ6Y

[77] Erwin Wu, Ye Yuan, Hui-Shyong Yeo, Aaron Quigley, Hideki Koike, and Kris M Kitani. 2020. Back-Hand-Pose: 3D Hand Pose Estimation for a Wrist-worn Camera via Dorsum Deformation Network. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*. 1147–1160.

[78] Xsens. 2020. Xsens Motion Capture. https://www.xsens.com/motion-capture

[79] Weipeng Xu, Avishek Chatterjee, Michael Zollhoefer, Helge Rhodin, Pascal Fua, Hans-Peter Seidel, and Christian Theobalt. 2019. Mo2Cap2: Real-time Mobile 3D Motion Capture with a Cap-mounted Fisheye Camera. *IEEE Transactions on Visualization and Computer Graphics* 25, 5 (2019), 2093–2101. https://doi.org/10.1109/TVCG.2019.2898650

[80] Xing-Dong Yang, Tovi Grossman, Daniel Wigdor, and George Fitzmaurice. 2012. Magic finger: always-available input through finger instrumentation. In *Proceedings of the 25th annual ACM symposium on User interface software and technology*. 147–156.

[81] KangKang Yin and Dinesh K Pai. 2003. FootSee: an interactive animation system.. In *Symposium on Computer Animation*. Citeseer, 329–338.

[82] Yang Zhang and Chris Harrison. 2015. Tomo: Wearable, low-cost electrical impedance tomography for hand gesture recognition. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*. 167–173.

[83] Yang Zhang, Chouchang Yang, Scott E Hudson, Chris Harrison, and Alanson Sample. 2018. Wall++ room-scale interactive and context-aware sensing. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. 1–15.

[84] Zhengyou Zhang. 2012. Microsoft kinect sensor and its effect. *IEEE multimedia* 19, 2 (2012), 4–10.

[85] Mingmin Zhao, Tianhong Li, Mohammad Abu Alsheikh, Yonglong Tian, Hang Zhao, Antonio Torralba, and Dina Katabi. 2018. Through-wall human pose estimation using radio signals. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '18)*. IEEE, 7356–7365. https://doi.org/10.1109/CVPR.2018.00768

[86] Christian Zimmermann, Tim Welschehold, Christian Dornhege, Wolfram Burgard, and Thomas Brox. 2018. 3d human pose estimation in rgbd images for robotic task learning. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 1986–1992.