Reconstruction of Human Body Pose and Appearance Using Body-Worn IMUs and a Nearby Camera View for Collaborative Egocentric Telepresence

Qian Zhang* Akshay Paruchuri* YoungWoon Cha† Jiabin Huang‡ Jade Kandel*

Howard Jiang* Adrian Ilie* Andrei State* Danielle Albers Szafir* Daniel Szafir* Henry Fuchs*

*Department of Computer Science, University of North Carolina at Chapel Hill

†School of Computing, Gachon University

‡Department of Computer Science, University of Maryland

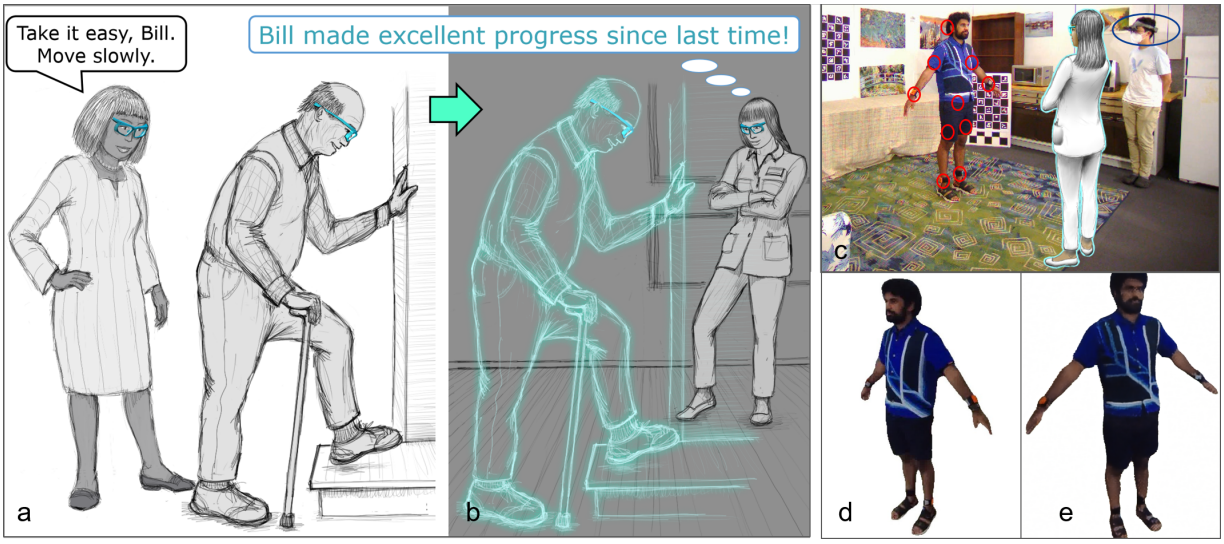

Figure 1: **Future scenario:** a) Patient outside his home, with his pose and appearance captured by his own AR glasses and IMUs, plus camera imagery from the AR glasses of his care partner. b) Patient's data being reviewed by his physical therapist. **Current scenario:** c) Overview of "home" with the "patient" on the left wearing IMUs (circled in red) , the "care partner" on the right wearing an AR device with a camera (circled in blue), and a "therapist" who can view the reconstruction remotely using VR glasses. d) Segmented input image from the camera worn by the "care partner." e) Output image for the novel viewpoint of the "therapist."

Abstract

We envision a future in which telepresence is available to users at any time and location, enabled by sensors and displays embedded in accessories worn everyday, such as wristwatches, jewelry, belt buckles, shoes, and eyeglasses. As a step toward reaching this goal, we present a collaborative approach to 3D reconstruction that combines a set of inertial measurement units (IMUs) worn by a target person with an external view from another nearby person wearing an AR headset, used for estimating the target person's body pose and reconstructing their appearance, respectively. We illustrate this approach with a prototype system in a physical therapy scenario that enables a patient to perform their exercises in the comfort of their home. Our system captures and reconstructs the patient's pose and appearance over time for interactive feedback from, and later review by, a therapist wearing a VR headset. Our results demonstrate that integrating the IMUs and an external camera yield better reconstructions than when using either of them alone. We believe our collaborative approach is orthogonal to other egocentric approaches to 3D reconstruction of human bodies in uninstrumented environments while minimizing the encumbrance imposed on the users in terms of the number and size of devices to wear.

Index Terms: Computing methodologies—Artificial intelligence—Computer vision—Reconstruction; Computing methodologies—Machine learning—Machine learning approaches—Neural networks

1 Introduction

Telemedicine methods have recently emerged as an alternative to in-person physical therapy sessions. Initially employed to reduce costs, engage with those in rural areas, and alleviate the challenges of patients with mobility issues, their use has increased significantly during the COVID-19 pandemic. To support an increasing range of remote physical therapy assessments and interventions, telepresence approaches featuring standard cameras, displays, microphones, and speakers embedded in laptops or phones have been augmented by wearable technologies such as IMUs. We anticipate a not-too-distant future in which these devices will be embedded in accessories already worn every day, such as AR eyeglasses, watches, jewelry, belt buckles, shoes, etc. These additional sensors provide opportunities for improved reconstruction of the pose and appearance of the patient, as well as vast opportunities for improved medical and

*e-mail: {qzane, akshay, kandelj, hjiang23, adyilie, andrei, dnszafir, dszafir, fuchs}@cs.unc.edu

†e-mail: youngcha@gachon.ac.kr

‡e-mail: jbhuang@umd.edu

health-related monitoring.

While telepresence enables remote social interaction, it often comes with onerous harware requirements such as instrumeting an environment with cameras, and restricts the users to the intrumented space. In contrast, collaborative egocentric telepresence has the potential to enable remote interactions anytime, anywhere. An AR headset typically has cameras, in addition to a display used to overlay computer graphics onto the real world. One of these cameras can be used to better understand the world around its user, including the pose and appearance of other human beings that are either completely or partially in the camera view. We expect AR headsets to become widely-available in the future, and that camera views from these headsets will be leveraged for the improvement of 3D reconstruction of other human beings involved in a collaborative egocentric telepresence experience.

Importantly, many aspects of patient care necessitate an involved clinician (in a clinic setting) and care partner (when at home). The availability of an additional person to work with the patient provides a unique opportunity to incorporate an additional viewpoint for enhancing the reconstruction of the patient's body pose and appearance.

As a step toward a fully-mobile telepresence system that does not rely on instrumented environments, we present a collaborative approach, which uses an external view from a camera worn by a second, nearby person. We employ today's versions of tomorrow's unencumbering sensors: a combination of body-worn IMUs and a camera from a prototype headset worn by a different person to generate novel target views. We show that using a combination of poses from IMUs and a single external camera as a reference view leads to encouraging results compared to prior methods of reconstructing body pose and appearance.

2 Related Work

2.1 IMU-based 3D Body Pose Capture

We are interested in self-contained 3D pose and appearance capture for a situation in which the user is wearing AR eyeglasses. We assume the AR glasses contain at least one IMU and one outward-looking camera to determine their pose in a given environment. The user's pose can be determined using a number of body-worn IMUs, with 10 IMUs worn on the major body bones shown to be sufficient for reliable body pose capture [40]. Other methods have employed fewer IMUs and estimations of temporal orientations and accelerations [16, 42], but we find they don't work well in our situation due to issues such as measurement noise and drift, or complex calibration procedures. Additional methods utilize both cameras and IMUs, with camera views used to constrain the IMU pose result, and IMUs used to track body poses outside of the camera views [5, 26, 37, 39, 40]. A recent method [9] uses visual-inertial sensor fusion to give accurate 3D body pose, leveraging egocentric downward-looking cameras and only 4 IMUs. We cannot employ this method because our AR headset does not feature downward-looking cameras. Since the main contribution of this paper is the improvements gained with the use of a single external camera, for now, we rely on the simple, robust 10-IMU method [40] to acquire the user's pose instead.

2.2 3D Body Models

The most widely-used body model is *SMPL* [24]. It is an unclothed body model that uses linear blend skinning to deform a predefined body mesh model. Multiple research efforts [14, 19, 21, 23, 27, 29, 32, 43] focus on estimating the body shape and pose parameters. *Octopus* [4] was proposed to estimate the shape and texture for a *SMPL* model. Most of these methods assume a weak perspective camera model, with only scale and translation parameters, while *SPEC* [20] estimates the relative body pose using a perspective camera model. These approaches only apply to fixed cameras, as they need to see the full body to estimate the body parameters. Our collaborative approach uses a head-worn camera that can get close to the target person, often generating partial views of their body. *EgoRenderer* [15] introduced a method to estimate full body pose and texture with a bulky egocentric downward-looking fish-eye camera. In our project, we strive to use a lightweight AR headset that could plausibly be worn in the future as part of a person's daily life. To that end, we opted to do the reconstruction in a collaborative manner, using a nearby person's view to reconstructing the appearance. *NeRF* [28] demonstrated a 3D reconstruction method for static scenes and objects, which has recently been extended to body-model-based deformations [11, 41]. We evaluated these methods on our datasets and found that while they work well on simple motions, they fail on complex motions (see Figure 2).

2.3 Pose-Guided Human Image Synthesis

Instead of estimating the body model in 3D, there are methods that synthesize novel views and novel human poses in 2D. Some methods are based on motion transfer [12, 22, 35], but they only work on small view changes. Other methods are based on body joints and skeletons [6, 10]. *DensePose* [13] is a representation that describes the body motion by mapping each pixel from the body image into a canonical texture space. Methods based on this representation include [3, 30]. Unfortunately, the estimation is not consistent across different frames, so when applied to video data, the generated texture map becomes very blurry. In this work, instead of relying on *DensePose* to give the correspondence between frames, we only use it as an intermediate representation for the body pose features. Our *DensePose* is generated by the *SMPL* body model, which is spatially and temporally consistent across frames.

3 Approach

Our goal is to capture the pose and appearance of a target person over time using a set of IMUs worn by the target person combined with an external view from another nearby person wearing an AR headset. Toward that end, we have devised a collaborative reconstruction approach that reaps the benefits of previous approaches, while avoiding their shortcomings. We combine the flexibility of modeling human motion using the *SMPL* body model with the transformation capabilities of mappings and the encoding power of neural networks.

Our approach takes as inputs IMU-based body poses and images from the reference view. The wearer's body pose is used to compute the parameters of a *SMPL* model that is fitted in a pre-scan step. We also fit an extra global offset T_{offset} (Equation 2) in the SMPL template using the body silhouette from reference view space. The resulting 3D mesh is first used together with images from the reference view at each time step to obtain partial reference *uv* maps via inverse texture mapping. Next, the 3D mesh is used with the partial reference *uv* maps to render partial reference images from the target viewpoint using texture mapping. Similarly, *DensePose* [13] and positional encoding (*PE*) [38] *uv* maps are rendered in the target view using texture mapping. The rendered images are fed into a conditional *GAN* (*cGAN*) [17] that outputs the final image in the target view. The *cGAN* is trained using randomly sampled frames from videos captured by the prototype headset.

We have begun to apply our approach to the scenario shown in Figure 1 (right), in which a patient performing physical therapy exercises is captured and a physical therapist can either review their performance or provide interactive feedback using a VR headset. We developed the prototype egocentric capture system[1] described in Section 3.1. Its pipeline is shown in Figure 3.

[1]Code available at: `https://github.com/qzane/CoEgoRecon`

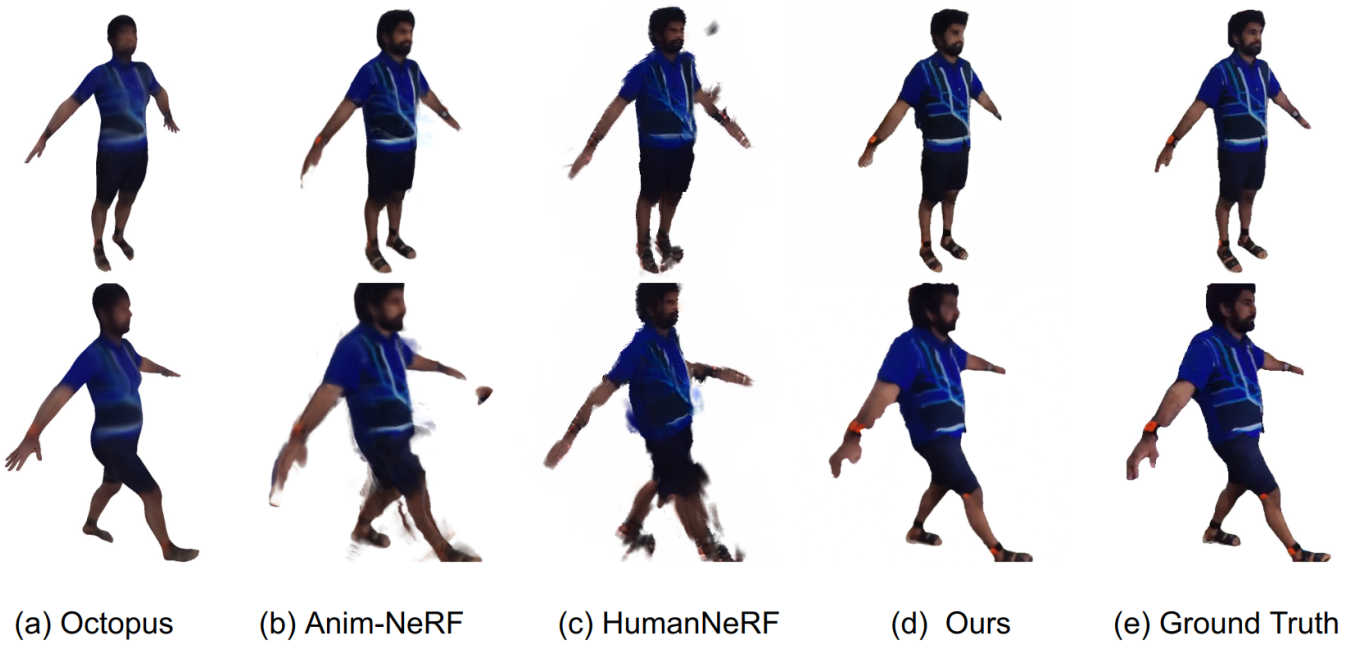

Figure 2: **Limitations of existing methods.** The first row shows the result from the simple motion dataset and the second row shows results from the complex motion dataset. The texture-and-mesh-based methods like *Octopus* (a) usually produce a blurry texture map, and cannot accurately represent body appearance. *NeRF*-based methods (b,c) don't use an explicit body model, so they can better describe body appearance. However, when the person is doing exercise with a complex body pose, the motion network fails to produce a good deformation of the body model and significant artifacts are visible in the rendered result. Our method (d) improves the rendered image quality by using a nearby person's view.

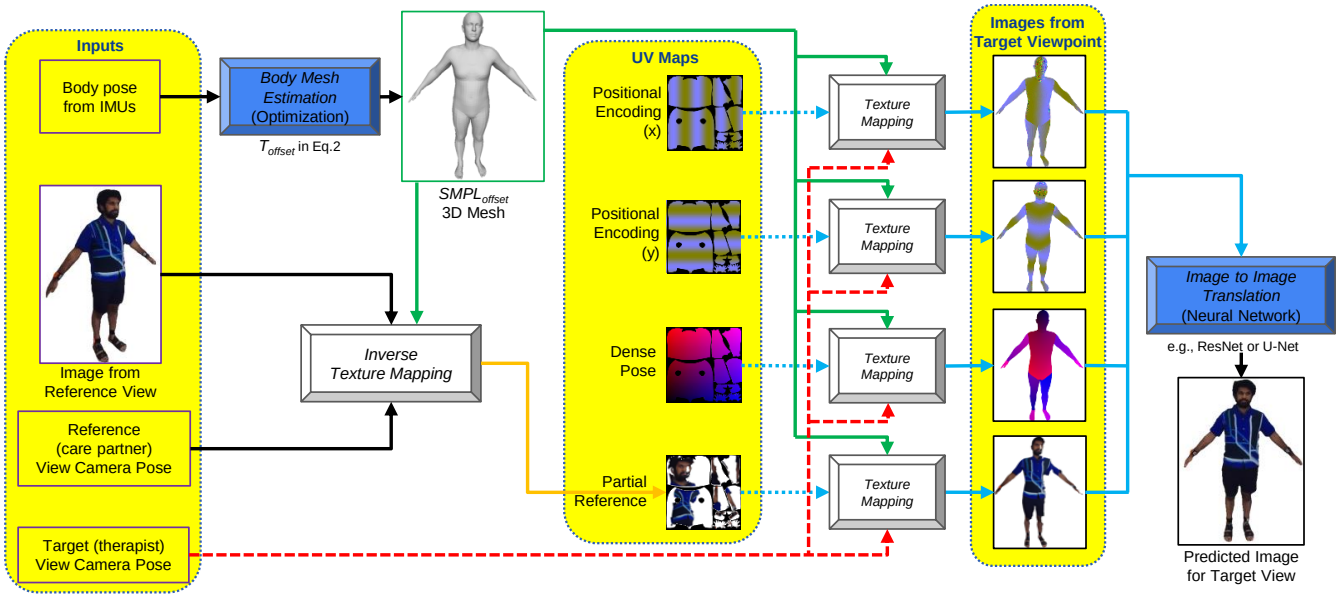

Figure 3: **Overview of our approach.** The body pose from IMUs is used to generate the $SMPL_{offset}$ 3D mesh. The image from the reference view is used together with the 3D mesh to generate a partial reference uv map through inverse texture mapping. We then use texture mapping to render the positional encoding (PE), DensePose and partial reference uv maps from the target viewpoint. Lastly, we use the rendered PE, DensePose and partial reference images as the input for an image-to-image translation network. The output is the predicted image for the target view.

3.1 Prototype Capture System

Our capture system, shown in Figure 1 (top right), consists of 10 *Xsens MTw Awinda* IMUs and an *Ximmerse Rhino X Pro* AR headset. We use the IMUs to compute the full body pose using the method from [40]. We use three headset cameras: two monochrome cameras for running SLAM to get the headset pose and one RGB camera for recording videos used for reconstruction. We plan to reduce the number of IMUs used to estimate the body pose by using a headset with downward-looking cameras and the approach in [9].

3.2 Body Mesh Estimation

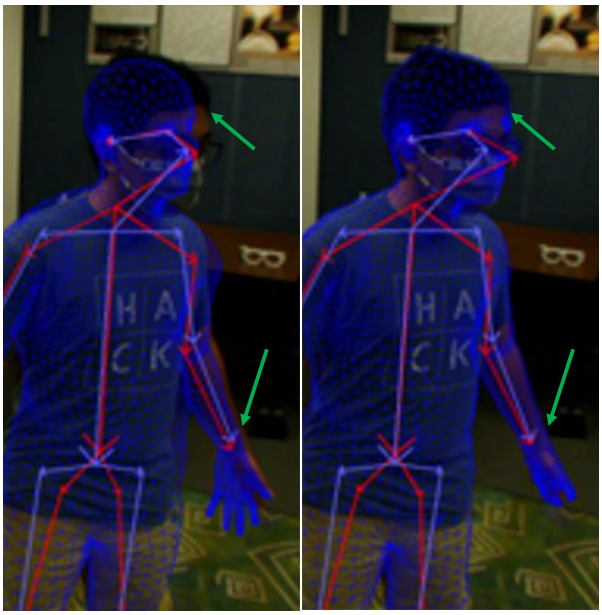

Figure 4: *SMPL* (left) vs. $SMPL_{off}$ (right). The mesh (shown in blue) is a much better fit around the hands and face (highlighted with green arrows) when adding our global offset.

We use a modified *SMPL* model to represent the estimated human body mesh, as shown in Figure 4. The original *SMPL* model [24] is a linear function that maps the shape parameters β and pose parameters θ to a set of $n = 6890$ 3D vertices points:

$$\begin{aligned} SMPL(\beta,\theta) &= W(T(\beta,\theta),J(\beta),\theta,\mathbf{W}) \\ T(\beta,\theta) &= T_{temp} + B_S(\beta) + B_P(\theta) \end{aligned} \tag{1}$$

$\beta \in R^{10}$ are the shape parameters, and $\theta \in R^{69}$ are the pose parameters that represent the rotations for 23 body joints. W is the linear blend-skinning function, T is the deformed template mesh, J represents the skeleton joints, $\mathbf{W}$ represents the blending weights, $T_{template} \in R^{6890x3}$ is the template body mesh, B_S represents the shape blend shapes, and B_P represents the pose blend shapes. The *SMPL* body model was designed for unclothed bodies, so it usually doesn't fit the body boundary well, especially around the head, hands and feet. To obtain a better fit, we introduced a global offset $T_{offset} \in R^{6890x3}$ and our modified model $SMPL_{offset}$ becomes:

$$\begin{aligned} SMPL_{offset}(\beta,\theta) &= W(T(\beta,\theta),J(\beta),\theta,\mathbf{W}) \\ T(\beta,\theta) &= T_{template} + T_{offset} + B_S(\beta) + B_P(\theta) \end{aligned} \tag{2}$$

We use the differentiable silhouette renderer of *Pytorch3D* [33] and *Adam* [18] to optimize β, θ and T_{offset}. The loss function we use for the optimization is:

$$\begin{aligned} L_{total} &= w_1 * L_{proj} + w_2 * L_{smooth} \\ L_{proj} &= L_{joints} + L_{sil} \\ L_{smooth} &= L_{edge} + L_{lap} \end{aligned} \tag{3}$$

L_{proj} is the sum of the reprojection loss for the 3D body joints projected onto the image plane L_{joints} and the loss for the silhouette L_{sil}. L_{smooth} is a smoothness loss which consists of a mesh edge length term (L1 norm) L_{proj} and a Laplacian smoothing term L_{lap}. In our experiments, we use weights $(w_1, w_2) = (30, 1)$.

We use 10 IMUs to estimate the human body pose. As shown in Figure 1 (top right), the IMUs are positioned on the back of the head, the wrists, the upper arms, the back of the pelvis, the upper legs, and the ankles. We use the gyroscope data from these IMUs to obtain the local bone rotations:

$$R^S = [s_x, s_y, s_z] \in \mathbb{R}^{3\times3} \tag{4}$$

We use ^{S} to denote the *S*keleton space. The local bone rotations allow us to directly update the pose parameters, θ, which are then used, alongside β obtained from the training phase shown in Figure 5, to render the *DensePose* [13], positional encoding (*PE*) [38] and partial reference *uv* maps in the target view.

Figure 5: **Training phase.** In this phase, the "patient" performs a few exercises, and the "care partner" records videos of their movement using an AR headset. We use this data to train our *pix2pix* [17] neural network model and also fit the shape parameters of the *SMPL* [24] model.

3.3 Camera Pose Estimation

Like most commercial IMUs, the system described in Section 3.2 produces reliable rotations combined with unreliable translations exhibiting large drifts. Consequently, an extra step is needed to align the body model and the headset video. We formulate this step as a perspective-n-point (PnP) problem [25]. Given a set of 3D body joints $P_{xyz} \in R^3$ from *SMPL* and a set of 2D body joints $P_{uv} \in R^2$ from the headset image detected by OpenPose [8], find the relative camera pose $\{R, T\}$ to align these points such that:

$$\begin{bmatrix} P_{uv} \\ 1 \end{bmatrix} = k * (P_{xyz} \times R + T) \tag{5}$$

Unlike other methods that require observing most of the body to estimate the model parameters, our method only requires detecting 4 (out of 25) joints to compute a reasonable alignment. This is advantageous in our application, because we cannot expect the care partner to look at the patient all the time. With fewer joints, the widely-used iterative PnP solver often falls into local minima when the initialization is not good enough, while the *SQPnP* [36] algorithm is more robust to tracking loss.

After the initial alignment, we use a differentiable rendering method [33] to refine the camera pose based on the segmented body silhouette. We found that the improved camera parameters from this refinement step help the optimization in Section 3.2 better cope with misalignments like the ones illustrated in Figure 4.

3.4 Rendering Images from the Target Viewpoint

Instead of simply rendering the 3D mesh from the target viewpoint, as in *Octopus* [4], we opted for a neural-network-based approach that produces the 2D image directly.

We use both positional encoding [38] and *DensePose* [13] to help encode the transform of the pixel coordinates from the canonical texture space to the target image space. The *DensePose* values are just the *uv* coordinates from -1.0 to 1.0, and the positional encoding we used is the sine functions for u and v, with i positions to encode:

$$PE(pos, 2i) = sin(pos/10000^{2i/d_{model}}) \tag{6}$$

We use $d_{model} = 3$ channels for both u and v coordinates and our experiments show that *PE* helps to make the details reconstructed better than just using *DensePose*.

We use "grid_sample" in pytorch for the texture mapping. For inverse texture mapping, there is no off-the-shelf implementation. We first apply a forward texture mapping to determine which triangles are visible in the reference view. And then, for each visible triangle, we use affine warpping to build the partial reference *uv* map.

3.5 Image-to-image translation

We use the *pix2pix* [17] framework for the final rendering step. This framework uses a conditional adversarial network (cGAN) to translate images from one domain to another. This network consists of an image generator G and a discriminator D. The generator G takes as input the label image x and a random vector noise y and outputs the rendered image. The discriminator D will try to distinguish whether a given image is a "fake" image from G or a real image from the ground truth, producing the image loss for training.

In our case, our input domain has 12 channels, 6 for the *PE* of the *uv* map, 3 for the *DensePose* and 3 for the partial image warped from the reference view to the target view. The output is a 3-channel image from the target view.

We tried both *UNet* and *ResNet* architectures for G, and find the *ResNet* architecture performs slightly better in our experiments.

In the training phase, we follow the standard approach from pix2pix [17]. We use the video from the headset as the ground truth images for the output domain. To obtain the reference view image during training, we randomly sample an image in the most recent 60 frames from our dataset. We train the cGAN for 300 epochs with a batch size of 8.

3.6 Integration with the environment reconstruction

To convey the patient's surroundings to the physical therapist, we record a video of the environment and use structure from motion (SfM) [2] to get the 3D reconstruction for the environment. We also placed a checkerboard in the room to help align the environment coordinate system and the simultaneous localization and mapping (SLAM) system running in the AR headset [7].

4 Experiments

4.1 Experimental setup

We implemented our model with *Pytorch* [31], the most popular machine learning library today. It features a linear algebra library that runs very fast on GPUs. One of its key features is the built-in automatic differentiation engine, which enables the implementation of the back-propagation (BP) [34] algorithm for solving optimization problems.

We used *Pytorch3D* [33] for differentiable rendering optimization. It is a machine learning library that focuses on 3D data. It offers efficient operations on 3D points and triangle meshes, including the loss functions we used in Equation 3. It also has a differentiable mesh renderer that supports basic shaders and camera models. The differentiable rendering feature enabled us to optimize our $SMPL_{offset}$ model (Equation 2) and the camera pose (Equation 5) with the BP algorithm.

For both training and optimization, we used *Adam* [18] with a learning rate of 0.001. *Adam* is an algorithm that adaptively updates the learning rate during the optimization process and produces results in a way that is less sensitive to the selection of initial learning rates.

We trained our model on a NVIDIA 3090 GPU, and the training typically takes around six hours.

4.2 Datasets

We collected two motion datasets with 120Hz IMU data, 30Hz RGB video data and 30Hz SLAM localization data for the headset. One dataset contains only simple motions like walking and turning around, and the other contains more complex exercises like stretching. There are 3828 frame images in the simple motion dataset and 2662 frame images in the complex motion dataset. In both datasets, we use 50% for training, 25% for validation and 25% for testing.

4.3 Evaluation metrics

We use peak signal-to-noise ratio (PSNR), structural similarity index measure (SSIM) and learned perceptual image patch similarity (LPIPS) for evaluation. PSNR is the the most popular metric for estimating image similarity, and it reflects the pixel-level differences between two images. SSIM is a newer method that is designed to take into account structural similarity between two images. LPIPS is a deep learning-based metric that measures the dissimilarity between two images by comparing the deep feature representations of image patches using a pretrained multi-layer neural network.

A higher PSNR, a higher SSIM or a lower LPIPS usually means the output image is more similar to the ground truth image.

4.4 Comparison with State-of-the-Art Methods

We compared our method with a SMPL model-based method: Octopus [4], and two model-free methods: Anim-NeRF [11] and HumanNeRF [41].

4.4.1 Image quality

The quantitative evaluation results are shown in Table 1 and the visual comparison is shown in Figure 6 for the simple motion dataset, and in Figure 7 for the complex motion dataset. We note that the model-based method generates a blurry texture map, while the model-free methods generate noticeable "ghosting" artifacts with complex motion. Our method has been successful in generating images with clear textural, while effectively minimizing the presence of artifacts.

4.4.2 Rendering Speed

Octopus is a texture mesh-based method, and it uses linear blending skinning to deform the model, so the rendering speed is very fast. *Anim-NeRF* and *HumanNeRF* are both ray sampling-based methods and query many rays to generate a single output image, so they run relatively slowly.

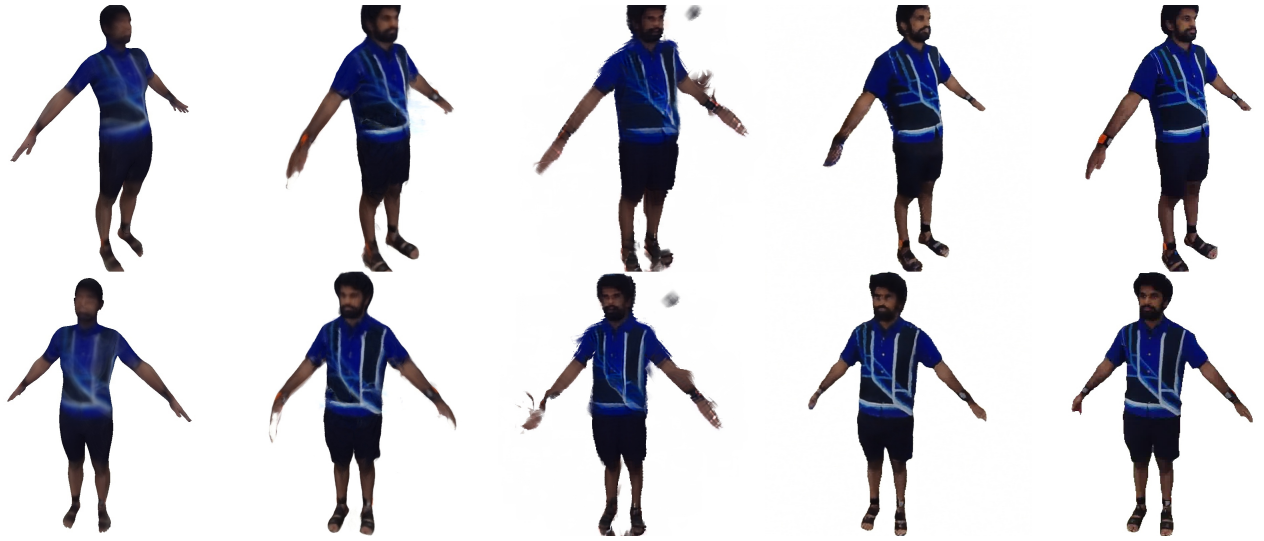

Figure 6: Performance on the simple dataset. From left to right: Octopus, Anim-NeRF, HumanNeRF, Ours, Ground Truth

Figure 7: Performance on the complex dataset. From left to right: Octopus, Anim-NeRF, HumanNeRF, Ours, Ground Truth

Figure 8: Ablation studies, from left to right: Ours w/o reference image, ours w/o PE, ours w/ UNet as a generator, Ours w/ ResNet, Ground Truth.

Simple Motion Dataset	PSNR↑	SSIM↑	LPIPS↓
Ours	**22.536**	**0.940**	**0.046**
Octopus [4]	15.463	0.887	0.136
Anim-NeRF [11]	22.510	0.931	0.082
HumanNeRF* [41]	17.272	0.865	0.123
Complex Motion Dataset	PSNR↑	SSIM↑	LPIPS↓
Ours	**23.748**	**0.948**	**0.057**
Octopus [4]	14.902	0.877	0.162
Anim-NeRF [11]	18.744	0.897	0.153
HumanNeRF* [41]	14.481	0.809	0.221

*We use *HumanNeRF* single GPU version for the experiment

Table 1: Quantitative evaluation results (complex moption dataset)

Octopus	Anim-NeRF	HumanNeRF	Ours
0.02s	30s	5s	0.3s

Table 2: Rendering speed (seconds per frame)

4.5 Ablation studies

We performed an ablation study, the results of which are shown in Fig. 8. We concluded that:

1. The reference view image helps reduce the noise in the output image.
2. The positional encoding (*PE*) helps retain more details in the image.
3. *ResNet* and *UNet* have virtually identical performance in generating the image.
4. *ResNet* is slightly better at recording details like the IMU on the right wrist.

Complex Motion Dataset	PSNR↑	SSIM↑	LPIPS↓
Ours w/o Refer	19.643	0.906	0.081
Ours w/o PE	19.541	0.906	0.101
Our w/ UNet as Generator	23.569	0.941	**0.052**
Ours (w/ ResNet)	**23.748**	**0.948**	0.057

Table 3: Ablation study

4.6 Limitations and Failure Cases

We found that reconstructing hands can be difficult (see Fig. 9), because we don't have any pose estimates for the wrist and fingers, and the reprojection and the alignment of the *SMPL* model and image are not good enough. We tried improving the result with a hand-enabled body model like *SMPL-X* [32], but we found the hand shape and thumb pose very difficult to optimize. Also, because our final rendering output is from a conditional GAN, the spatial consistency of the body appearance cannot be guaranteed.

Figure 9: Example failure case

5 Applications

This research is motivated by an ongoing collaboration with physical therapists to improve therapy outcomes and access. Traditional physical therapy sessions are in-person, which presents challenges for patients with mobility issues or in rural environments. The COVID-19 pandemic demonstrated the potential for telemedicine to enhance access to clinical treatment but highlighted therapists' lack of access to meaningful movement data for informing treatment. Wearable technologies have been employed to help track movement, performance, falls, gait, and other relevant data. However, they provide a limited view of a patient's movement. Tracking systems provide more robust data, but require greater space and consist of delicate, expensive devices that can be difficult to deploy reliably in a patient's home.

Our proposed reconstruction method using body-worn IMUs and a nearby egocentric view can help overcome limitations in current data collection methods to support telemedicine in physical therapy. We implemented our method in a prototype system for remote physical therapy that enables clinicians to collect rich data about a patient's movement during therapy exercises while only requiring the patient to wear a set of small IMUs mounted on easily-attached Velcro straps. A care partner or family member provides an external view by wearing a headset with a few miniature cameras. Prior to the therapy session, the care partner looks around the environment and moves around the patient, enabling our system to reconstruct the environment and an avatar for the patient. During the session, a remote physical therapist wearing a VR headset can see the patient from any viewpoint and provide therapy instructions and performance feedback in real-time. Data from the IMUs and the care partner's headset cameras enable pose and appearance avatar updates in near real-time, as the patient moves through their exercises.

Our prototype system allows the patient and therapist to concentrate on the therapy itself while providing robust data collection for later review. The therapist has full access to the patient's movements from any angle, and the patient does not have to remain within the field of view of a fixed camera or the active volume of a tracker, or worry about costly or complex instrumentation in their home.

Clinicians can also use our method to estimate a patient's body pose and reconstruct their appearance in subsequent playbacks of a physical therapy session. During playback, clinicians can see the patient's avatar movements in their reconstructed environment to gain contextualized insight into their therapy activities. An immersive visualization summarizes relevant motion data to allow clinicians to rapidly identify time segments of interest and recurrent patterns in the patient's movements.

Extracting joint locations from the IMU sensors, a visualization dashboard displays data critical to our clinical collaborators, including trunk angle and the relative position of the trunk angle to the feet in the coronal and sagittal planes. The visualization consists of a skeletal model with vectors describing relevant relative joint positions with length and color encoding magnitude. Clinicians can use these vectors to visualize how far the feet are from under the pelvis, which leg the patient is bearing their weight on, and how far forward or backward the patient is leaning at a given time point. A temporal heatmap summarizes this data across all collected time points, allowing clinicians to rapidly identify patterns across a set of sessions. For example, in the dashboard in Figure 10, the clinician can infer when the patient is sitting down by looking for dark red areas (when the feet are far from the pelvis) with bordering dark green marks (increased trunk angle from bending forward). The clinician can also infer that the patient tends to put more weight on one leg because the top blue row is darker than the second row, indicating that the hip is often closer to one foot than the other. Clinicians can use these patterns to quickly find movement sequences to investigate in greater detail through the reconstruction.

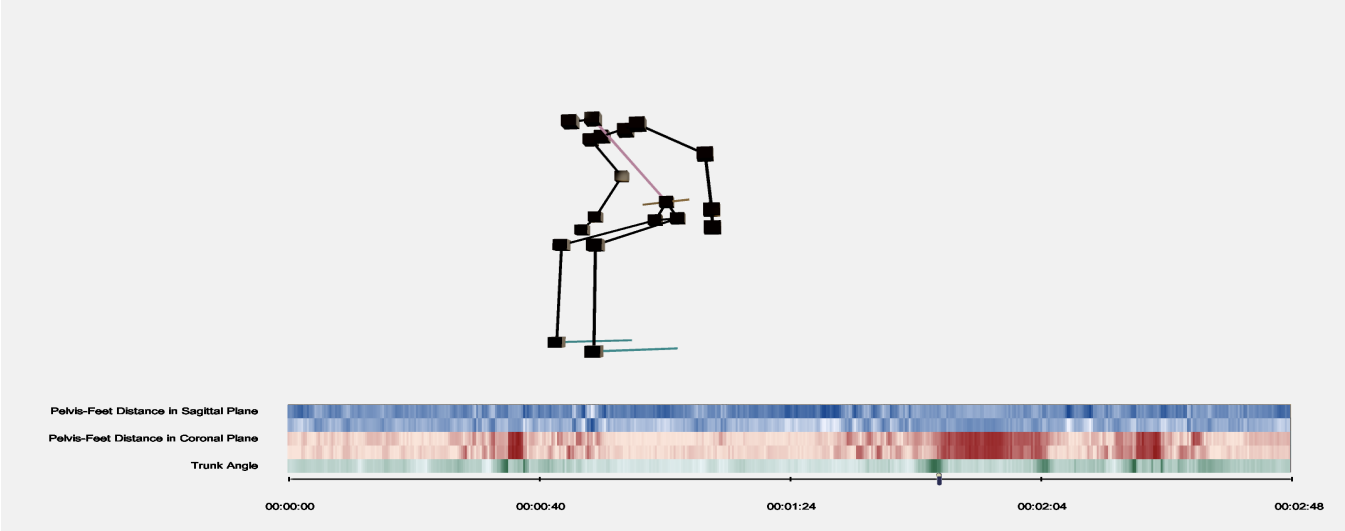

Figure 10: A prototype immersive visualization dashboard that summarizes relevant motion data for clinicians.

6 Conclusion and Future Work

We presented a novel collaborative egocentric telepresence approach which makes telepresence available anywhere, anytime, using devices that can one day be embedded in accessories worn everyday. We believe that advances in the technologies and algorithms used will enable collaborative egocentric approaches to match and eventually surpass the quality of the results currently available in fully instrumented environments.

Our method, using a combination of the pose from IMUs and a single external camera stream, has shown encouraging results, with significant improvements over prior methods of body pose and appearance capture.

In the near future, we plan to refine our technique to use fewer IMUs, preferably embedded in items likely to be worn daily, such as shoes, watches, and exercise bands. We hope to incorporate even fewer IMUs by having the patient also wear an AR headset, with both conventional forward- and downward-looking cameras [9]. We believe AR glasses with internal displays, cameras, and IMUs will be miniaturized to the form factor of today's prescription eyeglasses and worn all day, enabling scenarios such as the one shown in Fig. 1 (a,b). With this miniaturization and increasingly widespread adoption, we are optimistic that techniques such as those introduced in this paper will become widely useful for medical and healthcare applications, and also many other applications in which multiple participants with AR glasses are co-located in the same environment.

Acknowledgments

The authors would like to thank Jim Mahaney for help with setting up a capture space with various pieces of furniture and lighting fixtures to simulate a home environment. Jim Mahaney also helped extensively with experimental captures. We would also like to thank our collaborators from Ximmerse [1] for a prototype headset that enabled experimental captures. This work was partially supported by the National Science Foundation Award 1840131 and the Eunice Kennedy Shriver National Institute of Child Health & Human Development via National Institute of Health Award 1R01HD111074-01.

References

[1] Ximmerse. https://www.ximmerse.com/en/.

[2] Agisoft LLC. Metashape software, 2006.

[3] B. Albahar, J. Lu, J. Yang, Z. Shu, E. Shechtman, and J.-B. Huang. Pose with style: Detail-preserving pose-guided image synthesis with conditional stylegan. *ACM Transactions on Graphics (TOG)*, 40(6):1–11, 2021.

[4] T. Alldieck, M. Magnor, B. L. Bhatnagar, C. Theobalt, and G. Pons-Moll. Learning to reconstruct people in clothing from a single RGB camera. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, jun 2019.

[5] S. Andrews, I. H. Casado, T. Komura, L. Sigal, and K. Mitchell. Real-time physics-based motion capture with sparse sensors. In *CVMP 2016*, 2016.

[6] G. Balakrishnan, A. Zhao, A. V. Dalca, F. Durand, and J. Guttag. Synthesizing images of humans in unseen poses. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 8340–8348, 2018.

[7] C. Campos, R. Elvira, J. J. Gomez, J. M. M. Montiel, and J. D. Tardós. ORB-SLAM3: An accurate open-source library for visual, visual-inertial and multi-map SLAM. *IEEE Transactions on Robotics*, 37(6):1874–1890, 2021.

[8] Z. Cao, G. Hidalgo Martinez, T. Simon, S. Wei, and Y. A. Sheikh. Openpose: Realtime multi-person 2d pose estimation using part affinity fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

[9] Y.-W. Cha, H. Shaik, Q. Zhang, F. Feng, A. State, A. Ilie, and H. Fuchs. Mobile. egocentric human body motion reconstruction using only eyeglasses-mounted cameras and a few body-worn inertial sensors. In *2021 IEEE Virtual Reality and 3D User Interfaces (VR)*, pp. 616–625. IEEE, 2021.

[10] C. Chan, S. Ginosar, T. Zhou, and A. A. Efros. Everybody dance now. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 5933–5942, 2019.

[11] J. Chen, Y. Zhang, D. Kang, X. Zhe, L. Bao, X. Jia, and H. Lu. Animatable neural radiance fields from monocular rgb videos, 2021.

[12] C. Gao, A. Saraf, J. Kopf, and J.-B. Huang. Dynamic view synthesis from dynamic monocular video. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 5712–5721, 2021.

[13] R. A. Güler, N. Neverova, and I. Kokkinos. Densepose: Dense human pose estimation in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7297–7306, 2018.

[14] M. Habermann, W. Xu, M. Zollhofer, G. Pons-Moll, and C. Theobalt. Deepcap: Monocular human performance capture using weak supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5052–5063, 2020.

[15] T. Hu, K. Sarkar, L. Liu, M. Zwicker, and C. Theobalt. Egorenderer: Rendering human avatars from egocentric camera images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*,

pp. 14528–14538, 2021.

[16] Y. Huang, M. Kaufmann, E. Aksan, M. J. Black, O. Hilliges, and G. Pons-Moll. Deep inertial poser: Learning to reconstruct human pose from sparse inertial measurements in real time. *ACM Transactions on Graphics, (Proc. SIGGRAPH Asia)*, 37:185:1–185:15, Nov. 2018. First two authors contributed equally.

[17] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, 2017.

[18] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[19] M. Kocabas, N. Athanasiou, and M. J. Black. Vibe: Video inference for human body pose and shape estimation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[20] 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 *Proc. International Conference on Computer Vision (ICCV)*, pp. 11035–11045, Oct. 2021.

[21] N. Kolotouros, G. Pavlakos, M. J. Black, and K. Daniilidis. Learning to reconstruct 3d human pose and shape via model-fitting in the loop. In *ICCV*, 2019.

[22] Z. Li, S. Niklaus, N. Snavely, and O. Wang. Neural scene flow fields for space-time view synthesis of dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6498–6508, 2021.

[23] K. Lin, L. Wang, and Z. Liu. End-to-end human pose and mesh reconstruction with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1954–1963, 2021.

[24] M. Loper, N. Mahmood, J. Romero, G. Pons-Moll, and M. J. Black. Smpl: A skinned multi-person linear model. *ACM transactions on graphics (TOG)*, 34(6):1–16, 2015.

[25] X. X. Lu. A review of solutions for perspective-n-point problem in camera pose estimation. *Journal of Physics: Conference Series*, 1087:052009, 2018. doi: 10.1088/1742-6596/1087/5/052009

[26] C. Malleson, A. Gilbert, M. Trumble, J. Collomosse, A. Hilton, and M. Volino. Real-time full-body motion capture from video and imus. In *2017 International Conference on 3D Vision (3DV)*, pp. 449–457. IEEE, 2017.

[27] D. Mehta, O. Sotnychenko, F. Mueller, W. Xu, M. Elgharib, P. Fua, H.-P. Seidel, H. Rhodin, G. Pons-Moll, and C. Theobalt. Xnect: Real-time multi-person 3d motion capture with a single rgb camera. *Acm Transactions On Graphics (TOG)*, 39(4):82–1, 2020.

[28] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021.

[29] G. Moon and K. M. Lee. I2l-meshnet: Image-to-lixel prediction network for accurate 3d human pose and mesh estimation from a single rgb image. In *European Conference on Computer Vision*, pp. 752–768. Springer, 2020.

[30] N. Neverova, R. A. Guler, and I. Kokkinos. Dense pose transfer. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 123–138, 2018.

[31] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché Buc, E. Fox, and R. Garnett, eds., *Advances in Neural Information Processing Systems 32*, pp. 8024–8035. Curran Associates, Inc., 2019.

[32] G. Pavlakos, V. Choutas, N. Ghorbani, T. Bolkart, A. A. A. Osman, D. Tzionas, and M. J. Black. Expressive body capture: 3d hands, face, and body from a single image. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[33] N. Ravi, J. Reizenstein, D. Novotny, T. Gordon, W.-Y. Lo, J. Johnson, and G. Gkioxari. Accelerating 3d deep learning with pytorch3d. *arXiv:2007.08501*, 2020.

[34] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning internal representations by error propagation. Technical report, California Univ San Diego La Jolla Inst for Cognitive Science, 1985.

[35] S.-Y. Su, F. Yu, M. Zollhoefer, and H. Rhodin. A-nerf: Surface-free human 3d pose refinement via neural rendering. *arXiv preprint arXiv:2102.06199*, 2021.

[36] G. Terzakis and M. Lourakis. A consistently fast and globally optimal solution to the perspective-n-point problem. In *European Conference on Computer Vision*, pp. 478–494. Springer, 2020.

[37] M. Trumble, A. Gilbert, C. Malleson, A. Hilton, and J. P. Collomosse. Total capture: 3d human pose estimation fusing video and inertial sensors. In *Proceedings of the British Machine Vision Conference (BMVC)*, pp. 14.1–14.13, September 2017. doi: 10.5244/C.31.14

[38] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[39] T. von Marcard, R. Henschel, M. J. Black, B. Rosenhahn, and G. Pons-Moll. Recovering accurate 3d human pose in the wild using imus and a moving camera. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 601–617, 2018.

[40] T. Von Marcard, G. Pons-Moll, and B. Rosenhahn. Human pose estimation from video and imus. *IEEE transactions on pattern analysis and machine intelligence (TPAMI)*, 38(8):1533–1547, 2016.

[41] C.-Y. Weng, B. Curless, P. P. Srinivasan, J. T. Barron, and I. Kemelmacher-Shlizerman. HumanNeRF: Free-viewpoint rendering of moving people from monocular video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16210–16220, June 2022.

[42] X. Yi, Y. Zhou, and F. Xu. Transpose: Real-time 3d human translation and pose estimation with six inertial sensors. *ACM Transactions on Graphics*, 40(4), 08 2021.

[43] A. Zanfir, E. G. Bazavan, H. Xu, W. T. Freeman, R. Sukthankar, and C. Sminchisescu. Weakly supervised 3d human pose and shape reconstruction with normalizing flows. In *European Conference on Computer Vision*, pp. 465–481. Springer, 2020.