Pipeline for 3D reconstruction of the human body from AR/VR headset mounted egocentric cameras

Shivam Grover $^{1,+}$, Kshitij Sidana $^{1,+}$ and Vanita Jain $^{1,*}$

$^1$Bharati Vidyapeeth’s College of Engineering, New Delhi, India

$^+$Contributed equally to this work as first authors.

$^*$Corresponding author: Vanita Jain vanita.jain@bharatividyapeeth.edu

Abstract—In this paper we propose a novel pipeline for the 3D reconstruction of the full body from egocentric viewpoints. 3-D reconstruction of the human body from egocentric viewpoints is a challenging task as the view is skewed and the body parts farther from the cameras are occluded. One such example is the view from cameras installed below VR headsets. To achieve this task, we first make use of conditional GANs to translate the egocentric views to full body third person views. This increases the comprehensibility of the image and caters to occlusions. The generated third person view is further sent through the 3D reconstruction module that generates a 3D mesh of the body. We also train a network that can take the third person full body view of the subject and generate the texture maps for applying on the mesh. The generated mesh has fairly realistic body proportions and is fully rigged allowing for further applications such as real time animation and pose transfer in games. This approach can be key to new domain of mobile human telepresence.

Index Terms—Telepresence, 3D Reconstruction, Conditional GANs, Image-to-Image Translation, Virtual Reality, Generative networks

I. INTRODUCTION

The process of 3D reconstruction refers to the construction of the mesh and the corresponding texture of an object from a 2D image of it. 3D reconstruction of the human body using wearable cameras (such as virtual reality headsets) has many exciting applications such as walk-in movies, interactive TV shows, virtual meetings, remote training, and the most anticipated of all, personalized gaming experiences. While recent works which do facial reconstructions [1], pose estimation [2] [3], and environment reconstruction [4] from VR (virtual reality) headsets cameras have shown some spectacular results, no success has been seen in reconstructing the 3D mesh of the full body using only images from head-worn cameras. In our work, we not only achieve the 3D reconstruction of the full body, but we also infer the full body textures of the body all from a single pair of front-back egocentric images only. We also show the reconstructed body in novel poses and viewpoint.

In general, the goal of image-based 3D reconstruction is to infer the 3D geometry and structure of objects and scenes from one or multiple 2D images. The field of 3D reconstruction from images has been widely explored and has produced remarkable methods to do it. These methods include stereo-based techniques, shape from silhouette, or shape by space carving methods, using multiple images of the same object captured by well-calibrated cameras. This is generally achieved by placing one or more fixed cameras and sensors around the subject. The cost of setup is high and it’s feasibility is low as it requires dedicated sensors and a large space to be set up. Since it cannot be moved so easily, this lowers its portability and the mobility of the user which is a huge disadvantage for virtual reality based applications. In developing countries like India where more than 75% of the population lives in a house with less than 2 rooms, being able to afford and accommodate such a setup is like a dream.

We envision a future where to be able to achieve a sense of virtual physical presence of your peers, all you need is a head-wear gear. Virtual reality based applications have seen a rise in popularity in the last decade. With the work-from-home culture being at its peak, there is also a great demand and popularity for telepresence, virtual meetings and conferences.

While 3D reconstruction from static cameras and cameras capturing the subject from a distance is a widely explored area, 3D reconstruction from egocentric cameras is relatively new. There are certain aspects of egocentric images that pose a problem in working with them. The view of the body is not optimal for full body reconstruction since the view is massively distorted due to perspective, most of the body is occluded and while bending backwards, a major portion of the body goes out of the field of view of the camera (Fig. 1). While on the other hand the technology is cost effective and portable as it requires no additional setup or installation.

Keeping all this in mind, we present a novel pipeline for 3D reconstruction of the human body from cameras installed on VR headsets. Our pipeline includes two modules, one for view translation and one for the 3D reconstruction. In the first module, the egocentric (first person) images from the VR cameras are translated into third person full body views of the subject. This module is trained in an adversarial manner and uses an architecture similar to that proposed by [3]. The output from the first module is then sent through our second module which gives us the reconstructed mesh. We use the SMPL [6] body model which makes the output mesh very easy to import into 3D modelling software and game engines, and can further be animated as well. Our method does not require any pre-scans of the users body and can adapt to new users with varying body shapes without any tweaking of the model. We also talk about a synthetic dataset that we made for our research.
et al also seen remarkable success in reconstructing the human body from multi-angle images of the person taken in a static pose. This makes their model non-portable. For constructing the mesh and the texture, they require more RGB images has been achieved in many ways. Haberann et al. [13] [14] [15] have long been in use. The paper by Silva et al. [16] uses deep learning to classify human motion from motion capture files. They use a long short-term memory network (LSTM) [17] trained to recognize action on a simplified ontology of basic actions like walking, running or jumping. Motion capture based tracking usually require tracking markers, depth based sensors and cameras, pre-scanned body models and dedicated environments and lighting. The equipment used are usually very high in cost and low in portability. This prevents them to be used in virtual reality based applications. 3D reconstruction of humans from one or more RGB images has been achieved in many ways. Haberann et al. [18] propose a monocular real-time human performance capture. Their method, although promising, requires the camera to be at a distance from the subject which reduces the area of use. For constructing the mesh and the texture, they require a template model that they create using multi-angle images of the actor taken in a static pose. This makes their model non-adaptive and person specific. Parametric body models have also seen remarkable success in reconstructing the human body from RGB images. The work by Anguelov et al. [19] represent body shape and pose-dependent shape in terms of triangle deformations and by training on body scans containing unique structure and poses, they learn a statistical model for the shape variation. This was enhanced further by Chen et al. [20] when they also took into account the deformation of the clothing into their parameters. The SMPL model proposed by Loper et al. [21] uses a similar set of parameters for body shape and pose and gives a much more physically accurate model of the body. Using SMPL and gender specific models, the work by Pavlakos et al. [22] focuses on hands and expression alongside the pose of the body.

The work by Bogo et al. [23] uses an optimization based method to recover the parameters for SMPL from the output of CNN based keypoint detector. Kanazawa et al. [24] use adversarial training on unpaired dataset of static 3D human models and poses. Their model works in a frame-by-frame approach. Extended to videos, the work by Kocabas et al. [25] uses a recurrent architecture with an adversarial objective for inferring the shape and pose parameters. The approach for creating neural avatars by Shysheya et al. [26] splits the body into multiple parts and matches each pixel to these parts. Using the matched pixels the texture is generated. Their model relies on pose estimation and suffers significantly when there is error in the pose estimation.

**B. Egocentric 3D reconstruction of the body**

Next we look at methods for 3D reconstruction from Egocentric or wearable cameras. Reconstructing faces from wearable cameras is a unique and challenging problem because the object we want to track is distorted and largely occluded by the headset the camera is mounted on. Hence these techniques attempt to capture the full face through multiple sensors installed inside and around the headset.

To handle occlusions while having the minimum numbers of cameras, Wei et al. [27] used a multiview translation method in which they train a network that lets them augment additional views. Inspired from this we train a network which augments the third person view from the egocentric images.

Rhodin et al. [28] use two fisheye cameras attached to a helmet to predict the pose of the user in real time. Extending this idea further, Xu et al. [29] use only a single camera attached to a baseball cap and they are able to predict the 3D pose of the user in real time.

The work done by Cha et al. [30] is able to capture the motion of the user from head-worn cameras. They use a pre-scanned model of the user and transfer the motion of the user onto it. They are also able to reconstruct the environment and localize the user’s position within it. Using both audio and video for reconstructing and re-tagging the faces their work shows promising results. Their work however is very user specific. For a new user, a new pre-scan using multi-angled images of the person is needed and the pose estimation model has to be trained again for the new user. This limits the feasibility of their work since capturing the pre-scans and dataset for pose estimation require a separate dedicated setup.

**III. System Overview**

Our proposed system consists of a wearable VR headset with two cameras installed on it, one of the front and one on the back. To make it easier to visualize the setup, we have shown the camera placement and the simulated views in Fig. 2.

The camera on the front of the VR headset captures the view of the body from the front and is more prone to occlusions.

**Fig. 1.** Sample images from a camera placed under a VR headset capturing the front body. In (a), it can be seen that even in an image where the person is standing straight, there is severe distortion due to perspective. In (b) and (c) some portion of the body is occluded by the hands. In (d), the person is bending slightly backwards which causes the legs to be occluded.
Fig. 2. (a) The placement of the cameras on the VR headset. There is a camera in the front pointing downwards capturing the front and another camera on the back capturing the back. (b) The views from the front and back VR cameras. You can see that they are severely distorted due to perspective. (c) The third person view of the back and front body.

(such as the hands blocking the legs) than the camera on the back. Previous works such as that by Rhodin et al. [3] use two cameras for perceiving the depth of the image using stereoscopic vision. Our method on the other hand does not require the depth information. We use two cameras to be able to understand the body posture in scenarios where a single camera on the front won’t be enough. For example if the person is looking slightly upwards or if the person is bending backwards, the camera on the front barely captures the body but is pointed more towards the environment, but in the same situation the camera on the back of the VR headset gets a complete view of the body and is used by our model to infer the body structure and appearance. We assume that the user is in a solid colored room and the body is easily differentiable from the background.

Since there doesn’t publicly exist a dataset specifically for VR based cameras that we could use for our work, we created a simulated dataset. We will elaborate more on that in a separate section.

An overview of our reconstruction pipeline can be understood from Fig. 3. The whole process is divided into two steps: view translation and 3D reconstruction. In the view translation step, the input to the network is a vertically stacked version of the images from the front and back VR cameras (first person view) and the output is an image containing full body front and back view (third person view). The view in the input images are largely affected by perspective and occlusions and this step allows us to translate them into a more comprehensible form for the 3D reconstruction. The output from the view translation step goes as input into the second module for the 3D reconstruction and the 3D model of the person is obtained as the final output. A subset of the 3D reconstruction module is a texture generation model which generates easy-to-apply texture maps for the generated mesh.

The view translation step allows us to take advantage of the existing datasets for 3D human scans and their corresponding renders such as [26] [27] [28] and the existing state of the art methods for image to 3D reconstruction of the human body such as [23] [6] [29].

Reconstruction of the face and the facial expressions accurately is out of the scope of our system. It has already been achieved by [25] and can be used with our system as an extension. However we included the facial structure of the simulated characters in our dataset as it might prove to be useful for further research. Since the face is barely in the input images, the model will try to reconstruct the face using the existing data in its latent space.

IV. DATASET

At the time of performing this, there didn’t exist a dataset that we could’ve directly make use of for the work we are performing. The datasets released by [3] and [2] consisted of a large corpus of egocentric fish-eye images along with the detailed pose annotations for each image, but it gave no information about the occluded body parts and the body parts that were not in the field of view of the camera. Hence for our problem, we curated our own dataset.

The guidelines for making the dataset were simple. For each pair of front-back egocentric images, there should be a corresponding pair of front-back full body third person views of the person. This would allow the deep learning model to learn to cater to occlusions and situations where a body part is out of the field of the view of the camera. Fig. 4 shows how such a setup would look like. The biggest obstacle of creating such a dataset was to get the front-back third person views for each frame. We required a dataset of our subjects performing various activities and every-time the subject would rotate and move around, we would have to move the camera around him so that their relative motion is null. Only this would allow us to get the accurate front-back third person views as the subject performs various activities. To create such a system was a mammoth task and impractical.

So we decided to solve this problem by creating a large synthetic dataset that is tailored to suit our system. We used the SMPL model [6] and attached 4 cameras to each model according to the setup in Fig. 4. One camera is attached in front of the head at a distance approximately where the bottom of a VR headset would be and another camera is attached behind the head approximately where the strap of the VR headset would be. Two more cameras for the third person views are placed in front and back of the character. These two
Fig. 3. Pipeline for reconstructing the 3D model of the human body from egocentric images. The inputs from the egocentric camera are first sent through an image translation network which translates the egocentric views into third person full body views. Then the translated view is sent through the 3D reconstruction module which outputs the shape and pose parameters for the SMPL model and the texture maps are obtained using the texture generation model. The reconstructed 3D mesh can be animated and viewed from novel viewpoints.

Fig. 4. Camera setup for dataset collection. Two cameras are placed on the VR headset and two cameras are placed at a distance from the user to capture the front and back third person views.

Cameras are attached to the hip joint of the rigged skeleton of the model. This allows the cameras to move relative to the subject as it performs various activities. The subject could be back-flipping or samba dancing and the cameras will always capture the accurate front-back third person views.

For texturing the mesh, we used the texture maps released by [27] and applied them to the model. The multiple shape parameters of the SMPL model allowed us to create several variations in the body shape of the subject. We used [28] to rig and animate the models. We made the models perform over 50 distinct activities including boxing, jumping, dancing, walking (activities that might be performed in VR games or in virtual meetings) and we rendered a total of 50,000 frames and for each frame there is a pair of front-back egocentric views and a pair of front-back third person views.

The background for each frame is a solid color which allows for manually augmenting the backgrounds as per the need of the research.

V. IMAGE TO IMAGE TRANSLATION

The first step to reconstruct the user’s body is to obtain third person full body views from the egocentric views. This step is crucial as it allows the system to be able to understand the distortion of the image due to perspective and to infer the occluded and non-visible regions of the body before it is reconstructed in 3D.

The main aim of this step is to generate an output of type $y$ given an input of type $x$. Generative Adversarial Networks (GANs) [30] have performed remarkably well in the deep learning based generative area of study. Their architecture consists of two models, a generator $G$ and a discriminator $D$. The job of the generator is to generate realistic examples relative to the training dataset and the job of the discriminator is to classify an image as realistic or fake. $G$ and $D$ are both trained together in a two-player min-max situation. But GANs are only effective in generative image synthesis applications if we need to generate new examples of images. We have no control on the data being generated. To be able to control the outputs and to make use of additional information, such as class labels, or in our case an input image of type $x$ that we want to be translated into an image of type $y$, we use an extension of GANs called Conditional GANs [31].

In conditional GANs, the generator $G$ learns to generate fake samples with a condition instead of unknown noise distribution. The final objective of a conditional GAN looks like

$$L_{cGAN}(G, D) = \mathbb{E}_{x,y}[\log D(x, y)] + \mathbb{E}_{x,z}[\log(1 - D(x, G(x, z)))]$$

(1)

Our problem is of the type image to image translation and there has been some remarkable progress in this field when combined with conditional GANs. Conditional GANs have been used to achieve tasks like colorization of black and white images by Isola et al. [32], future frame prediction [33], image prediction from normal maps [34] etc. The work that we decided to use for our research by Isola et al. [5] consists of a very general image to image translation architecture which has been used to several applications by researchers later such as pose transfer [35], edges to realistic images, simulation to reality etc. They also incorporate a convolutional PatchGAN classifier for the discriminator which allows the structure to penalize at the scale of image patches. So instead of trying to check whether the image as whole is real or not, the PatchGAN checks whether each N x N patch in the image fed to the discriminator is real or not. Then the predictions by the discriminator for all patches are averaged and given out as the final output.
Fig. 5. The training of the view translation module. The generator takes as input the egocentric image and tries to generate the third person views for it. The generated image is then fed to the discriminator which classifies it as real or fake. They are both trained simultaneously until the generator starts outputting realistic third person images that correspond well with the egocentric images.

Along with the cGAN loss in (1), they also use a traditional L1 loss. This forces the generator to generate images near the ground truth output in an L1 sense while also trying to fool the discriminator into believing the generated images are real.

\[
L_{L1}(G) = \mathbb{E}_{x,y,z} \left[ \| y - G(x,z) \|_1 \right]
\]  

This results in their final objective function as,

\[
G^* = \arg \min_G \max_D L_{cGAN}(G,D) + \lambda L_{L1}(G)
\]  

Apart from the PatchGAN, their generator network uses a U-Net style architecture which allows them to establish a better relation between the input and output images that have the same low level structure such as in image colorization and simulation to reality. In our case though this feature is not as useful since our input images and output images are considerably different. Though this doesn’t prove to be a disadvantage either as the network without the U-Net architecture gave similar results to the original Pix2Pix network.

The training of the network is straightforward; the model is fed with the vertically stacked front-back pair of images from the VR camera as input and the combined third person views for ground truth. The ground truth could be fed in three different ways as seen in the Fig. 6. Fig. 6(b) might seem like a better option at first since we can make a direct correlation between the input and output visually since the egocentric front image is aligned with the third person from image and so on for the back images, but this gave low quality results as a huge portion of the image was blank and wasted. In Fig. 6(d) the third person views are scale up but there is no direct correspondence between the first person and third person views. In Fig. 6(c), the third person views are not only scaled up and also aligned horizontally with their corresponding first person views. The output images are rotated clockwise, which isn’t really an issue as long as the model is able to learn the correspondence well. The experiment results for each orientation can be seen in the Results section.

VI. 3D RECONSTRUCTION

Once we have the output from the view translation step, we are ready to reconstruct the 3D body from it. For reconstruction of the body we chose the generative human body model SMPL which is a realistic human body model and its body shape and the pose can be controlled by tweaking shape and pose parameters. Since the rig of the SMPL model has 23 joints, any pose can be defined with \( |\theta| = 3 \times 23 + 3 = 72 \) parameters; i.e. 3 for each part plus 3 for the root orientation. Similarly the body shape has 10 parameters.

Using the SMPL model will also allow us to animate the reconstructed model. The basic idea here is to fit the SMPL
model to our subject’s body images by accurately estimating the parameters required for its body shape. This has been achieved earlier by Bogo et al. [21] which fits the SMPL model to the output of a CNN keypoint detector. Other methods [37] [38] [39] include training neural networks which use the pixels to directly regress the parameters. More recent approaches such as [22] [23] use adversarial objectives to infer the parameter for SMPL. We follow the adversarial approach for our work as well. The HMR [22] model works best for still images while VIBE [23] goes a step further and employs a recurrent architecture with the adversarial objective which allows it to work remarkably well with videos.

For the scope of this research, implementing and applying HMR was more feasible and efficient since we only needed to know the body shape parameters from a single image. Along with the pose $\theta$ and the body shape $\beta$, they also use a weak-perspective camera model and solve for global rotation $R \in \mathbb{R}^{3 \times 3}$ in axis angle representation, scale $s \in \mathbb{R}$ and the translation $t \in \mathbb{R}^2$. This finally leaves them with a set of parameters to represent the reconstruction of any 3D human body which can be expressed as $\Theta = \{\theta, \beta, R, t, s\}$. $\Theta$ is an 85 dimensional vector.

The objective function that they use to train their final model is given by,

$$L = \lambda (L_{\text{reproj}} + \ell L_{\text{3D}}) + L_{\text{adv}}$$ (4)

where $L_{\text{reproj}}$ is the joint reprojection error, $L_{\text{3D}}$ is the 3D Error, $L_{\text{adv}}$ is the adversarial loss, $\lambda$ controls the relative importance of each objective and $\ell$ is a function that is 1 only if the 3D ground truth is available for the image, otherwise it is 0.

The input to this module is the third person view of the front body obtained in view translation and the output of the module is the set of parameters required to reconstruct the 3D body. By applying the obtained body shape parameters to the SMPL model we successfully reconstruct the 3D rigged mesh of the body and by applying the obtained pose parameters we can transform the model according to the pose of the person in the input image.

For the texture we trained another image-to-image translation model which takes as input the third person views generated in step 1 and outputs the corresponding texture maps. These texture maps can be used as is with the generated model and require no tweaking of the UV coordinates.

### VII. Results

In this section we will show and evaluate the results of our view translation and 3D reconstruction pipeline.

#### A. View Translation

For view translation, the first experiment that we conducted was to establish the best method to arrange the third person views before feeding it into the model as ground truth. We checked three different methods of arranging them as seen in Fig. 6. We trained a model three times on the same dataset and each time arranging the ground truth images differently. To quantitatively evaluate them, we checked the mean squared error values and the SSIM between the output images and their corresponding ground truth images on a test dataset containing 200 images. We show the average values in table I where Method A corresponds to Fig. 6(b), Method B corresponds to Fig. 6(c), and Method C corresponds to 6(d). It should be noted however that for evaluating the ground truth for Method A, we cropped and scaled up the region including the subject to match the size of the other two types of ground truth

| Metric | Method A | Method B | Method C |
|--------|----------|----------|----------|
| RMSE   | 89       | 53.2     | 40.1     |
| SSIM   | 0.67     | 0.72     | 0.89     |

### Fig. 7. Example results for the view translation step. The model is not only able to infer the body shape and the color of the person’s clothes, but also minor details such as stripes and patterns.
orientations. This is done since having a high quality larger image in the output was a driving factor for this experiment and also the average error between the images of two very small artifacts would be smaller generally.

Next we show the results of the final model on unseen input images in Fig. 7 and Fig. 11. On comparing the generated results with the ground truth we get the average SSIM value as 0.89 and RMSE value as 40.1. We further use a state of the art pose detector [40] on the generated and ground truth images to quantitatively evaluate the accuracy of the generated pose and we get an average RMSE of 10.21 between the generated joint values and the ground truth joint values.

B. 3D reconstruction

To evaluate the 3D reconstruction model individually, we first input non-generated ground truth third person views to the model and compare the SMPL model generated from it to the actual model for those images. (Fig. 8). In the example shown, there is a high level of occlusion since the right hand is not visible at all and it can be seen that the model is fairly accurate even for such a complex pose.

Furthermore, we compare the performance of the SMPL model that we are using with another state of the art method for reconstruction of the human body from a single RGB image by Saito et al. [29]. Their work PiFu is able to generate the
Fig. 10. The mesh generated from the SMPL model is compared with the mesh generated using PiFu. PiFu generates a distorted mesh of the body due to occlusions by the hand whereas the mesh recovery for the SMPL model gives fairly accurate results.

### TABLE II

**AVERAGE INFERENCE TIME FOR EACH MODULE ON A TESLA K80 GPU**

| Model                        | Inference Time |
|------------------------------|----------------|
| View Translation             | 0.6 sec        |
| Parameter estimation for SMPL| 0.12 sec       |
| Texture Map Generation       | 0.56 sec       |

Fig. 10: The mesh generated from the SMPL model is compared with the mesh generated using PiFu. PiFu generates a distorted mesh of the body due to occlusions by the hand whereas the mesh recovery for the SMPL model gives fairly accurate results.

Furthermore, their generated mesh is not rigged. Since the SMPL approach uses a rigged 3D model, it can be animated later if we import it in a virtual reality game or if we are able to extract the pose from the VR camera, we can simply transform the 3D model using the new pose. We show this in Fig. 12.

**C. Texture map generation**

To evaluate the texture map generation model, we use a real third person view rendered from an actual 3D model as input and compare the generated textures to the ground truth textures (Fig. 9). It can be seen that the model is able to infer the colors of the clothes easily and it also tries to adapt to other details such as varying types of sleeves and patterns on the t-shirt.

**D. Inference Time**

The inference times for all the steps performed on a Tesla K80 GPU are shown in Table II. One thing to note about our work is that since the 3D model is rigged, we have to infer the 3D model only once and for every consequent frame it can simply be animated by transferring the pose of the subject.

**VIII. CONCLUSION**

In this paper, we presented a novel pipeline for reconstruction of the human body from egocentric cameras. Instead of directly reconstructing the 3D model from egocentric views, we first train a model that can translate the egocentric views into third-person full body views. We use one camera on the front and one on the back of the VR headset, which allows us to get a better sense of the overall body structure and cater to occlusions by inferring a third person view of the body. From the third person views, we estimate the shape and pose parameters for the SMPL model and the corresponding texture map which can be applied to the mesh as is. We further show the reconstructed 3D model from novel view points and in novel poses and compare it to the actual 3D model. The reconstructed model can easily be imported into any 3D modeling software and game engines and can be further animated. Our work can be incorporated with egocentric pose estimation and be animated in real time as well. The mesh will only be generated once in the starting and for every consequent frame only the pose will be estimated and applied to the mesh. This will be useful for several applications such as virtual meetings and conferences, interactive VR games, walk-in movies, remote training and interactive TV shows.
Fig. 11. Example results for the view translation module.
Fig. 12. Example results for animation of the mesh. We compare the generated and the ground truth texture in novel poses and view points.
REFERENCES

[1] X. Chen, Y. Guo, B. Zhou, and Q. Zhao, “Deformable model for estimating clothed and naked human shapes from a single image,” in *The Visual Computer*, vol. 29, pp. 1187–1196, 2013.

[2] W. Xu, A. Chatterjee, M. Zollhofer, H. Rhodin, P. Fua, H.-P. Seidel, and C. Theobalt, “Mo$^2$Cap$^2$: Real-time mobile 3d motion capture with a cap-mounted fish eye camera,” *IEEE Transactions on Visualization and Computer Graphics*, pp. 1–1, 2019.

[3] H. Rhodin, C. Richardt, D. Casas, E. Insafutdinov, M. Shafiei, H.-P. Seidel, B. Schiele, and C. Theobalt, “Egocap: egocentric marker-less motion capture with two fish eye cameras,” *ArXiv*, abs/1609.07306, 2016.

[4] Y.-W. Cha, T. Price, Z. Wei, X. Lu, N. Rewkowski, R. Chabra, Z. Qin, H. Kim, Z. Su, Y. Liu, A. Ilie, A. State, Z. Xu, J.-M. Frahm, and H. Fua, “Towards fully mobile 3d face, body, and environment capture using only head-worn cameras,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 24, pp. 2993–3004, 2018.

[5] P. Isola, J.-Y. Zhu, T. Zhou, and A. Efros, “Image-to-image translation with conditional adversarial networks,” *CVPR*, 2017.

[6] M. Loper, N. Mahmood, J. Romero, G. Pons-Moll, and M. J. Black, “SMPL: A skinned multi-person linear model,” *ACM Trans. Graphics (Proc. SIGGRAPH Asia)*, vol. 34, no. 6, pp. 248:1–248:16, Oct. 2015.

[7] J. Engel, T. Schöps, and D. Cremers, “Lsd-slam: Large-scale direct monocular slam,” in *Computer Vision – ECCV 2014*, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds. Cham: Springer International Publishing, 2014, pp. 834–849.

[8] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, “Osb-slam: A versatile and accurate monocular slam system,” *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1171–1183, 2015.

[9] S. Izadi, D. Kim, O. Hilliges, D. Molyneaux, R. A. Newcombe, P. Kohli, J. Shotton, S. Hodges, D. Freeman, A. J. Davison, and A. W. Fitzgibbon, “Kinectfusion: real-time 3d reconstruction and interaction using a moving depth camera,” in *UIST* ’11, 2011.

[10] Z. Jochinke, Z. Jochinke, F. Kisby, F. Kisby, J. Altman, J. Altman, J. Rogers, J. Rogers, K. Andreason, K. Andreason, and et al., “3d camera, capture & virtual tour platform,” Jun 2020. [Online]. Available: https://masterportal.com/

[11] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohli, J. Shotton, S. Hodges, A. W. Fitzgibbon, “Kinectfusion: Real-time dense surface mapping and tracking;” *2011 10th IEEE International Symposium on Mixed and Augmented Reality*, pp. 127–136, 2011.

[12] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Computation*, vol. 9, pp. 7-–42, 2001.

[13] E. de Aguiar, C. Theobalt, C. Stoll, and H.-P. Seidel, “Marker-less deformable mesh tracking for human shape and motion capture,” *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8, 2007.

[14] J. Starck and A. Hilton, “Surface capture for performance-based animation,” *IEEE Computer Graphics and Applications*, vol. 27, 2007.

[15] D. Vlasic, I. Baran, W. Matusik, and J. Popovic, “Articulated mesh animation from multi-view silhouettes,” *ACM Trans. Graph.*, vol. 27, p. 97, 2008.

[16] R. E. da Silva, J. Ondrej, and A. Smolic, “Using lstm for automatic facial animation via multiview image translation,” *ACM Transactions on Graphics (TOG)*, vol. 38, pp. 1–16, 2019.

[17] S. Ionescu, D. Papava, V. Olaru, and C. Sminchisescu, “Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 7, pp. 1325–1339, Jul 2014.

[18] G. Varol, J. Romero, X. Martin, N. Mahmood, M. J. Black, I. Laptev, and C. Schmid, “Learning from synthetic humans,” in *CVPR*, 2017.

[19] “Mixamo.” [Online]. Available: http://www.mixamo.com/

[20] S. Saito, Z. Huang, R. Natsune, S. Morishima, A. Kanazawa, and H. Li, “Pifu: Pixel-aligned implicit function for high-resolution clothed human digitization,” *ArXiv preprint arXiv:1905.05172*, 2019.

[21] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets,” in *Advances in Neural Information Processing Systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2014, pp. 2672–2680. [Online]. Available: http://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf

[22] M. Mirza and S. Osindero, “Conditional generative adversarial nets,” *ArXiv*, abs/1411.1784, 2014.

[23] R. Zhang, P. Isola, and A. A. Efros, “Colorful image colorization,” in *ECCV*, 2016.

[24] M. Mathieu, C. Couprie, and Y. LeCun, “Deep multi-scale video prediction beyond mean square error,” 2015.

[25] X. Wang and A. Gupta, “Generative image modeling using style and structure adversarial networks,” vol. 9908, 2016, pp. 318–335.

[26] C. Chan, S. Ginosar, T. Zhou, and A. A. Efros, “Everybody dance now,” 2019 *IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 5932–5941, 2019.

[27] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” *ArXiv*, abs/1505.04597, 2015.

[28] R. A. Güler and I. Kokkinos, “Holopose: Holistic 3d human reconstruction in-the-wild,” in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019*, Long Beach, CA, USA, June 16-20, 2019. Computer Vision Foundation / IEEE, 2019, pp. 2382–2392. [Online]. Available: http://openaccess.thecvf.com/content_CVPR_2019/html/Guler_Holopose_Holistic_3D_Human_Reconstruction_In-The-Wild_CVPR_2019.html

[29] M. Omran, C. Lassner, G. Pons-Moll, P. V. Gehler, and B. Schiele, “Neural body fitting: Unifying deep learning and model-based human pose and shape estimation,” *Verona*, Italy, 2018.

[30] G. Pavlakos, L. Zhu, X. Zhou, and K. Daniilidis, “Learning to estimate 3d human pose and shape from a single color image,” *CVPR*, 2018.

[31] 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.