Compensating Motion-Induced Errors in Smartphone-Based VR Avatar Reconstruction

Friedemann Runte*
friedemann.runte@tu-dortmund.de
TU Dortmund University
Dortmund, Germany

Timo Menzel*†
timo.menzel@tu-dortmund.de
TU Dortmund University
Dortmund, Germany

Ulrich Schwanecke
ulrich.schwanecke@hs-rm.de
RheinMain University of Applied Sciences
Wiesbaden, Germany

Mario Botsch‡
mario.botsch@tu-dortmund.de
TU Dortmund University
Dortmund, Germany

Abstract

Recent developments in smartphone-based avatar reconstruction have made the creation of personalized and realistic avatars significantly more accessible. However, relying on one smartphone camera leads to capturing images sequentially, which introduces new challenges; particularly longer capture times increase the susceptibility to subject motion, which results in degraded reconstructions.

We present a novel approach for smartphone-based avatar reconstruction that combines photogrammetry, silhouette constraints, and inverse rendering to produce high-fidelity, realistic avatars free of motion-induced artifacts. By using short, motion-resilient image sequences, referred to as *sub-scans*, we considerably reduce motion-induced artifacts. Our pipeline achieves high visual quality while offering improved robustness and outperforms current state-of-the-art methods in terms of computation time and accuracy.

CCS Concepts

• **Computing methodologies** → **Mesh geometry models**.

Keywords

Virtual Humans, 3D Reconstruction, Avatars, Template Fitting

ACM Reference Format:
Friedemann Runte, Timo Menzel, Ulrich Schwanecke, and Mario Botsch. 2025. Compensating Motion-Induced Errors in Smartphone-Based VR Avatar Reconstruction. In *31st ACM Symposium on Virtual Reality Software and Technology (VRST '25), November 12–14, 2025, Montreal, QC, Canada.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3756884.3765995

1 Introduction

The growing availability of consumer-grade Virtual Reality (VR) devices has led to an increased number of use cases for VR – ranging from gaming and social applications to therapy scenarios [50]. In social VR and therapy applications in particular, the faithful and realistic visual representation of the user is of crucial importance, as it directly influences the feeling of embodiment and thus the immersion of the entire experience [16, 25, 47, 51].

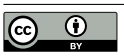
As a result, research into avatar reconstruction has risen in recent years. These reconstruction methods often use large multi-camera rigs [1, 12, 34], which produce photorealistic avatars, but require specialized personnel to operate, are expensive due to the amount of high-end cameras, and are location-bound. To overcome these issues, other approaches use monocular camera/smartphone input instead. Although this greatly reduces costs and makes it less location-bound, the usage of monocular cameras leads to longer scanning procedures, which introduce new problems, in particular unintentional motion during the scanning process. Some approaches use photogrammetry for high geometric detail but ignore the potentially occurring motion of the subject, tolerating possible artifacts [35, 56]. Others utilize silhouette constraints to improve the robustness of the fitting, but geometric detail and quality of the texture are not satisfactory [3–5]. Recent avatar reconstruction methods [17, 18, 22, 23, 41, 43, 53, 58] based on Neural Radiance Fields [36] or 3D Gaussian Splatting [24] better incorporate the occurring motion into their fitting/training process. They often use inverse rendering frameworks, but the outcomes are (often) unsuitable for VR applications, due to slow inference time, slow rendering, and visual artifacts.

We present a novel avatar reconstruction method that leverages the advantages of photogrammetry, silhouette constraints, and inverse rendering to create high-fidelity avatars from smartphone images. We split the input image sequence into distinct subsequences to generate several point clouds that do not contain motion-induced artifacts. After that, we fit a high-resolution mesh-based template model to the different resulting point clouds and additionally use silhouette constraints to compensate for remaining motions within each subsequence. This results in realistic, personalized, and VR-ready full-body avatar.

2 Related Work

This section provides an overview of the current state of research on avatar reconstruction. We first group approaches by the type of representations used for the resulting avatar models; second, we describe methods categorized by the scanning device.

*Both authors contributed equally to this research.
†Also with Lamarr Institute for Machine Learning and Artificial Intelligence.
‡Also with Lamarr Institute for Machine Learning and Artificial Intelligence.



VRST '25, Montreal, QC, Canada

ACM ISBN 979-8-4007-2118-2/2025/11
https://doi.org/10.1145/3756884.3765995

2.1 Representation of Avatars

Recently, numerous approaches have been published that utilize representations based on 3D Gaussian Splatting (3DGS) [12, 17, 19, 29, 31, 37, 38, 41, 43, 48, 55, 58]. Starting from a mesh or a point cloud, these approaches use a mixture of 3D Gaussians to represent the avatar, enabling the reconstruction of fine details (e.g. wrinkles, hair). These Gaussians are view-dependent, which further enhances realism by accounting for occlusion and perspective effects that vary with the camera angle. Early 3DGS-based approaches required significant time to compute Gaussians [19, 38, 55]; however, new methods reduce this to minutes [12, 30, 41, 48]. Furthermore, recent approaches reach 150 to 300 fps in rendering performance, which allows the use of these models for real-time applications [20, 30, 48]. However, as current VR devices typically render at 70–120 fps, the usage for VR applications remains a challenging task, as the rendered scenes usually consist of more than just one avatar (e.g., multiple avatars and the surrounding environment). Iandola et al. [21] showed that it is necessary to reduce the number of Gaussians to make 3DGS-avatars usable on VR devices. However reducing the number of Gaussians also reduces the rendering quality. The view-dependence of Gaussians, typically a key advantage of this representation, also poses challenges for VR. If the training data lacks sufficient coverage of possible views and poses, novel views create artifacts that undermine the realism of the avatar, as we show in Section 4.1.

In contrast, many approaches employ mesh-based representations for the resulting avatar models, as mesh-based models are highly efficient in terms of rendering speed, can be easily animated into novel poses, and are view-consistent [1, 4, 5, 14, 35, 52, 56]. They are widely used to represent 3D models and characters, making them compatible with current game engines and 3D programs.

Our goal is to generate realistic VR-ready full-body avatars with broad usability and applicability. For this, the disadvantages of 3DGS-based methods outweigh their benefits. Ultimately, *users* will control their avatar, leading to unpredictable new poses and views. Due to the guaranteed spatial consistency and the broad compatibility of mesh-based 3D models with current game engines (e.g., Unity, Unreal Engine), we utilize a mesh-based representation for our avatars.

2.2 Input Devices

Many approaches to avatar reconstruction use expensive multi-camera rigs consisting of up to 100 high-end cameras to capture images of a person simultaneously from all perspectives [1, 15, 28, 34, 39, 40, 47, 49]. Although these approaches yield visually appealing results, the capture method is only available in well-equipped laboratories due to the high cost of the scanner setup.

To reduce costs and make avatar reconstruction more accessible, monocular cameras or smartphones could be used instead of multi-camera rigs [3–5, 35, 52, 56, 57]. Early approaches by Alldieck et al. [3, 4, 5] capture monocular videos of a subject and use silhouette constraints to fit a mesh-based avatar. Fitting to the silhouette of each frame makes the pipeline more robust, but reduces the geometric quality as concave details do not influence the silhouette. Wenninger et al. [56] use smartphone videos and a photogrammetry-based template-fitting approach; however, their results do not match the quality of expensive camera rigs and require skilled personnel to be used correctly.

Figure 1: Point cloud from all images (left) and sub-scan capturing the left part of the subject. The full point cloud exhibits noticeable motion artifacts in both arms.

Figure 2: Comparison with Avatars for the Masses (A4M) [35]: Input image (left), result from A4M (center), our result (right). The A4M result displays a noticeably bigger arm than the input image.

More research has been conducted on avatars from monocular capture in recent years. New approaches improve visual quality [52, 57] but often require long training times (e.g. 2 days [52]).

Recently, Menzel et al. [35] proposed a method that, similar to ours, aims for broad availability, accessibility, and usability. They use a smartphone-based capture process and a mesh-based representation for their avatars. With a server-based design, their pipeline is the first one enabling non-expert users to generate VR-ready avatars using just a smartphone. Their results outperform those of Gaussian-based methods in terms of reconstruction time, rendering performance, and quality of novel pose animation. They also surpass other mesh-based approaches regarding geometric detail. Menzel et al. use a photogrammetry-based approach to fit their avatars. This allows for fast and high-detailed reconstruction, but point clouds generated using photogrammetry algorithms are prone to artifacts when input images of a non-static object, e.g., a human, are captured sequentially.

Because motions of the scanned object violate the photogrammetry assumption (the object is static and does not deform), these motions are visible in the resulting point cloud as noise. When capturing a human standing in A-pose using a smartphone, this is mainly observable in the arms, which potentially move throughout the scan (see Figure 1, left). Artifacts arise especially around the lower arms. Menzel et al. ignore this problem, incorporating these artifacts into their final avatars (see Figure 2, center).

Since we target similar requirements to Menzel et al., we employ a mesh-based representation and an approach that uses photogrammetry in combination with silhouette constraints to robustly reconstruct high-detail avatars, even with motion during scanning. In addition, we use inverse rendering to compute realistic textures.

3 Method

We combine the key ideas from different avatar reconstruction approaches into a single pipeline. We use photogrammetry to achieve high geometric detail, silhouette constraints to improve the robustness, and inverse rendering to generate a high-quality texture. To ensure reliable point clouds, despite the subject moving during the scan procedure, we introduce a new concept called *sub-scans*. We divide the captured image sequence into smaller consecutive subsequences. This reduces the subject's motion in each subsequence, resulting in point clouds with significantly fewer motion artifacts.

In the following, we first introduce our concept of sub-scans (Section 3.1), define the loss functions used in our mesh optimization (Section 3.2), provide an overview of our mesh fitting pipeline (Section 3.3, see Figure 3), and finally present our inverse rendering based texture generation (Section 3.4).

3.1 Sub-Scans

We use the iOS application provided by Menzel et al. [35] to capture two separate scans: one of the *body* and one of the *head*. The app captures 105 RGBD images in HEIC format (45 body images, 60 head images). All images are taken sequentially over a period of 2 minutes. For the body scan, the subject stands in an A-pose, which makes it susceptible to arm movement. This is not the case for the head scan, as the head is more rigid and easier to keep in the same pose. Therefore, motion artifacts most likely arise in the body scan. As this motion occurs over the full duration of the scanning procedure, it has reduced impact over shorter time intervals. To minimize the influence of motion, from the set of all body images $\mathcal{I}_b = \{\mathbf{I}_1, \ldots, \mathbf{I}_{45}\}$ we form m subsets $\mathcal{I}_b^1, \ldots, \mathcal{I}_b^m$ consisting of n consecutive images such that

$$\bigcup_{i=1}^{m} \mathcal{I}_b^i = \mathcal{I}_b. \tag{1}$$

We refer to these subsets as *sub-scans* (see Figure 3, green and yellow boxes). They form the basic concept for the following loss functions. Note that we use an overlap of one image between two consecutive sub-scans. For instance, in the case of $m = 3$ sub-scans the 45 body images are split into sub-scans $\mathcal{I}_b^1 = \{\mathbf{I}_1, \ldots, \mathbf{I}_{15}\}$, $\mathcal{I}_b^2 = \{\mathbf{I}_{15}, \ldots, \mathbf{I}_{30}\}$, and $\mathcal{I}_b^3 = \{\mathbf{I}_{30}, \ldots, \mathbf{I}_{45}\}$.

In a first step, we generate point clouds for the body and head scans using Agisoft Metashape [2]. We align both point clouds using landmarks detected with Mediapipe [33] in the input images, similar to [35]. This ensures that the head scan matches the position, orientation, and scaling of the head in the body point cloud. Furthermore, for each sub-scan, we generate corresponding point clouds $\mathcal{P}_b^1, \ldots, \mathcal{P}_b^m$. As shown in Figure 1, the sub-scan point cloud (right) covers a smaller area of the scanned person, but contains less artifacts than the full body point cloud (left), particularly in the lower arm regions where movement occurred.

In order to precisely fit our template to the sub-scans, we first need to align the sub-scan point clouds. We use the camera calibration of the full-body point cloud as initialization for the sub-scan photogrammetry. Despite the motion artifacts in the point cloud, the information from the camera calibration remains valid because large parts of the subject (e.g., legs, torso, and head) are static during the capture. This results in the sub-scan point clouds being aligned by construction.

3.2 Loss Functions

We use a template-fitting approach to reconstruct avatars. Our template mesh, created by an artist, is defined by its $N \approx 24$k vertex positions $\mathbf{V} = \{\mathbf{v}_1, \ldots, \mathbf{v}_N\}$, joints $\mathcal{Q} = \{q_1, \ldots, q_{59}\}$, and 51 blendshapes modeled after the ARKit blendshapes [6]. It can be skinned using the standard linear blend skinning function:

$$\mathbf{skin}(\mathbf{v}_k, \boldsymbol{\theta}) = \left(\sum_{q \in \mathcal{Q}} w_{k,q} \mathbf{T}_q(\boldsymbol{\theta}) \right) \mathbf{v}_k, \tag{2}$$

with $\mathbf{v}_k$ being a vertex position in homogeneous coordinates, $\boldsymbol{\theta} \in \mathbb{R}^{177}$ a vector of $59 \cdot 3 = 177$ joint angles, $\mathbf{T}_q$ the affine transformation matrix of joint q, $w_{k,q}$ the skinning weight for the vertex $\mathbf{v}_k$ and joint q. Furthermore, we can adjust the template's shape with a PCA model built from the CAESAR database [46] which can be controlled via parameters $\boldsymbol{\beta} \in \mathbb{R}^{15}$:

$$\mathbf{V} = \boldsymbol{\mu} + \mathbf{A} \cdot \boldsymbol{\beta}, \tag{3}$$

where $\boldsymbol{\mu}$ is the mean mesh computed from the CAESAR database and $\mathbf{A} \in \mathbb{R}^{3N \times 15}$ is the PCA matrix.

We use different loss functions to match pose and shape of our template to the sub-scans. In the following, we define the loss functions used in our optimization and give a short description of the influence of each loss.

3.2.1 Closest-Point-Correspondences. We build sets of closest-point correspondences C_i computed for each sub-scan $\mathcal{I}_b^i$ from point cloud $\mathcal{P}_b^i$ to our template mesh $\mathbf{V}$ with joint angles $\boldsymbol{\theta}_i$. The set of vectors of joint angles for all sub-scans is denoted as $\Theta = \{\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_m\}$. Each correspondence $c = (\mathbf{x}_c, \mathbf{v}_c) \in C_i$ is defined by a point in the point cloud $\mathbf{x}_c \in \mathcal{P}_b^i$ and a point $\mathbf{v}_c$ on a template triangle defined by barycentric coordinates. We define the closest-point-correspondence loss function as:

$$L_{\text{cpc}}(\mathbf{V}, \boldsymbol{\theta}_i, C_i) = \frac{1}{|C_i|} \sum_{c \in C_i} \| \mathbf{x}_c - \mathbf{skin}(\mathbf{v}_c, \boldsymbol{\theta}_i) \|_2^2 \tag{4}$$

This loss measures the squared distance between our skinned template and the point clouds. It serves two purposes: optimizing joint angles $\boldsymbol{\theta}_i$ to match the pose of our template with the pose of the subject in sub-scan i and non-rigidly deforming the vertex positions $\mathbf{V}$ of our A-pose template model to match the shape in the point cloud.

3.2.2 Silhouette Correspondences. We introduce a silhouette constraint to improve the robustness of the optimization, as proposed in earlier approaches [3–5]. Similarly to closest-point correspondences, we define silhouette correspondences. The silhouette is defined by the outline of the subject in the input image. We find

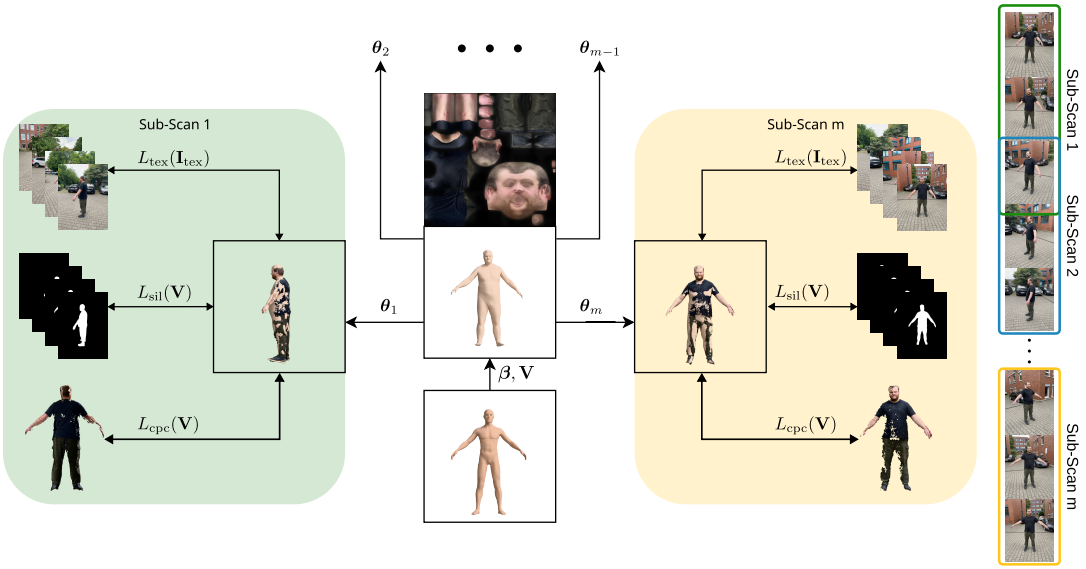


Figure 3: Overview of our sub-scan fitting with m sub-scans and sub-scan construction. We deform our template mesh using PCA parameters β and vertex positions V to get a coarsely fitted model. This is posed with joint parameters θ_j to bring the model into the correct pose for each sub-scan. We can then minimize the losses L_{cpc} and L_{sil} for each sub-scan simultaneously to compute the final geometry. After that, we minimize L_{tex} simultaneously for all sub-scans to generate the avatar's final texture.

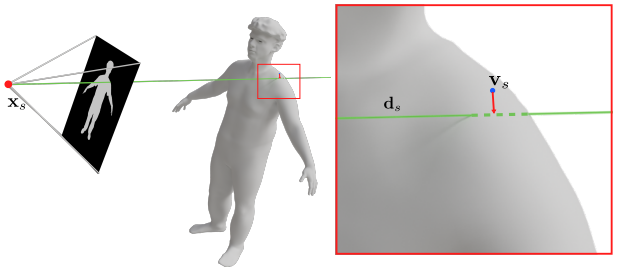


Figure 4: Visualization of one silhouette correspondence. The red dot is the camera center, the green line is the direction vector d and the blue dot is the template vertex that will be projected onto the ray. The shoulder of the template mesh is outside of the mask.

the outline of the subject by first segmenting the input images to separate the background from the subject and then computing the contour of the segmentation mask using OpenCV [8]. To compute the segmentation masks, we compared DeeplabV3 [13] and Segment Anything Model (SAM) [27]. We decided for SAM, as this model produced more accurate results.

We find correspondences by projecting the vertices of our posed model into each image, collecting all vertices outside of the segmentation mask, and finding their closest point on the outline (see Figure 4). The set of silhouette correspondences for sub-scan i is denoted as $\mathcal{S}_i$. $\mathcal{S}_i^j$ is the set of correspondences for image j of sub-scan i. As silhouette correspondences are calculated in image space, a single correspondence $s = (\mathbf{x}_s, \mathbf{d}_s, \mathbf{v}_s) \in \mathcal{S}_i^j$ is defined by a camera center $\mathbf{x}_s$, a direction vector $\mathbf{d}_s$ from the camera center through the image plane at the pixel coordinate of the contour point and the vertex position $\mathbf{v}_s$ of the posed model. We define the silhouette loss as:

$$L_{\text{sil}}(\mathbf{V}, \boldsymbol{\theta}_i, \mathcal{S}_i) = \frac{1}{n} \sum_{j=1}^{n} \sum_{s \in \mathcal{S}_i^j} \frac{1}{\left|\mathcal{S}_i^j\right|} \left\| \left(\mathbf{I} - \mathbf{d}_s \mathbf{d}_s^{\mathsf{T}}\right) \left(\mathbf{x}_s - \mathbf{skin}(\mathbf{v}_s, \boldsymbol{\theta}_i)\right) \right\|_2^2, \quad (5)$$

where $\mathbf{I}$ is the 3×3 identity matrix. The loss measures the squared distance between a skinned vertex position on the template and the line defined by the correspondence. This is used to restrict the template model to remain inside the visual hull defined by the masks of the input images.

3.2.3 Regularization. Similar to other template fitting approaches, we use a Laplacian-based regularization to penalize deviation from the template's curvature (i.e. bending):

$$L_{\Delta}(\mathbf{V}) = \frac{1}{\sum_{\mathbf{v} \in \mathbf{V}} A_{\mathbf{v}}} \sum_{\mathbf{v} \in \mathbf{V}} A_{\mathbf{v}} \left\| \Delta \mathbf{v} - \mathbf{R}_{\mathbf{v}} \Delta \overline{\mathbf{v}} \right\|_2^2, \quad (6)$$

where $\Delta \mathbf{v}$ is the cotangent Laplacian of the deformed vertex $\mathbf{v}$, $\mathbf{R}\Delta\overline{\mathbf{v}}$ is the rotated cotangent Laplacian of the undeformed vertex $\overline{\mathbf{v}}$, and $A_{\mathbf{v}}$ is the Voronoi area associated with vertex $\mathbf{v}$ [7].

3.3 Coarse-to-Fine Fitting

After defining the sub-scan-based loss functions, we now give a detailed description of our fitting process. The sub-scan point clouds cover less area than the full-body point cloud or miss parts of the scanned subject (e.g. arms). Therefore, we use the full-body point cloud, generated from all body images $\mathcal{I}_b$, to initialize the template

model. After initialization, we can use the sub-scans to finely fit the model's geometry.

3.3.1 Template Initialization. After aligning the head, body, and sub-scan point clouds, we align our template to the point clouds by computing a conformal transformation to match landmarks detected with OpenPose [10] and positions of the template's joints Q. Depending on the number of sub-scans, the point clouds only contain areas seen from few directions and thus lack volume information. This makes approximating proportions of body parts much more difficult. To compensate for this, we use the full-body point cloud for initialization, as large parts are static and can be used to coarsely match the shape of the body.

We minimize the distance between joint positions and the detected landmarks using inverse kinematics [9] to get a first approximation of the subject's pose. The pose and shape are then further refined by alternately optimizing L_{cpc} with respect to joint angles $\boldsymbol{\theta}_i$ as well as vertex positions $\mathbf{V}$ through the parameters $\boldsymbol{\beta}$ of our PCA model (see Figure 3, bottom to center).

3.3.2 Coarse Body Fitting. Accurate pose estimation, which is crucial for fine-scale sub-scan fitting, requires that the template and subject share similar body proportions. Our PCA shape model, trained on minimally clothed CAESAR scans, cannot explain the extra volume introduced by loose garments. After the initialization, we perform a coarse non-rigid fit of the model to the full-body point cloud. This lets the surface expand to match the clothing, providing a better initialization for subsequent pose refinement. This is done by minimizing L_{cpc} with respect to vertex positions using correspondences C. We additionally fine-fit to the head point cloud, as it does not contain motion-induced artifacts.

We use L_Δ to penalize deviation from the template model to ensure a smooth surface. Thus, the fitting loss is defined as

$$L_{\text{fit}}(\mathbf{V}, \boldsymbol{\theta}) = L_{\text{cpc}}(\mathbf{V}, \boldsymbol{\theta}, C) + \lambda_\Delta L_\Delta(\mathbf{V}). \tag{7}$$

This optimization is performed iteratively, with the regularization weight λ_Δ being gradually reduced. As the full-body point cloud may contain motion-induced artifacts, which are incorporated into the fitted mesh when the regularization weight becomes too low, we fit with λ_Δ in the range from 1 to 10^{-4}. In contrast, we fit the head point cloud until λ_Δ reaches 10^{-9}, as it does not contain motion-induced artifacts. We use iterative block coordinate descent for optimization.

3.3.3 Fine-Scale Sub-Scan Fitting. Although the point clouds of the sub-scans are aligned in the same coordinate system, the poses of the sub-scans do not match. Similar to the full-body point cloud, we need to adjust the pose per sub-scan. We therefore optimize L_{cpc} with respect to $\boldsymbol{\Theta}$ (see Figure 5).

After adjusting the model's pose $\boldsymbol{\theta}_i$ to each sub-scan, we fit the model using non-rigid registration by minimizing L_{cpc} and L_{sil} with respect to $\mathbf{V}$ over all sub-scans (see Figure 3, left and right). Our overall fitting loss is then defined as:

$$L_{\text{fit}}(\mathbf{V}) = \frac{1}{m}\sum_{i=1}^{m}\left[L_{\text{cpc}}(\mathbf{V}, \boldsymbol{\theta}_i, C_i) + L_{\text{sil}}(\mathbf{V}, \boldsymbol{\theta}_i, \mathcal{S}_i)\right] + \lambda_\Delta L_\Delta(\mathbf{V}). \tag{8}$$

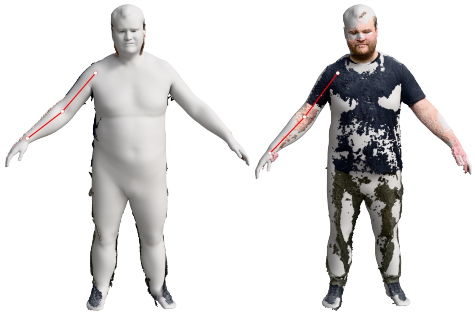


Figure 5: The template's pose is fitted to each point cloud individually while not differing too much from the first initalization. Left: Sub-scan 3, Right: sub-scan 5. Slight movements of the subject's right arm are noticeable. The moving right arm is marked with red.

Figure 6: Original masked image (left), rendered image (middle), weighting by view direction (right).

Similarly to the coarse body fitting, this loss is optimized using iterative block coordinate descent and again, we gradually reduce λ_Δ from 1 to 10^{-9}. This ensures the same level of detail as for the head scan. Additionally, we transfer the template's blendshapes to the fitted model via deformation transfer [35]. This completes the geometry fit of our pipeline.

3.4 Texture Generation

After fine-scale fitting the avatar's geometry, we compute a high-resolution texture using an inverse rendering framework. We define a loss function between our rendered mesh and the input image as:

$$\begin{aligned} &L_{\text{tex}}(\mathbf{I}_{\text{tex}}, \mathbf{I}_{\text{in}}, \mathbf{I}_{\text{ren}}) = \\ &\quad L_{\text{Ch}}(\mathbf{I}_{\text{in}}, \mathbf{I}_{\text{ren}}) + L_{\text{SSIM}}(\mathbf{I}_{\text{in}}, \mathbf{I}_{\text{ren}}) + L_{\text{TV}}(\mathbf{I}_{\text{tex}}), \end{aligned} \tag{9}$$

where $\mathbf{I}_{\text{ren}}$ is a rendered image of the fitted mesh (see Figure 6, center), $\mathbf{I}_{\text{tex}}$ is the texture of the mesh, $\mathbf{I}_{\text{in}}$ is the segmented input image with white background (see Figure 6, left), L_{Ch} is the differentiable Charbonnier L_1 loss [11], L_{SSIM} is the structural similarity index (SSIM) [54], and L_{TV} is the total variation loss (TV). We additionally weight L_{SSIM} and L_{Ch} per image by view-direction. The weighting is used to downweight parts of the image that do not face the camera and are thus less likely to be accurate. The loss function for

texture optimization over all sub-scans is then defined as

$$L_{\text{tex}}(\mathbf{V}, \mathbf{I}_{\text{tex}}) = \sum_{i=1}^{m} \sum_{j=1}^{n} L_{\text{tex}}\Big(\mathbf{I}_{\text{tex}}, \mathbf{I}_j^i, \mathbf{\Psi}(\mathbf{V}, \mathbf{I}_{\text{tex}}, \boldsymbol{\theta}_i)\Big), \tag{10}$$

where $\mathbf{I}_j^i$ is the j-th image of the i-th sub-scan, $\mathbf{\Psi}(\mathbf{V}, \mathbf{I}_{\text{tex}}, \boldsymbol{\theta}_i)$ is a function to render the fitted mesh $\mathbf{V}$ with texture $\mathbf{I}_{\text{tex}}$, posed with $\boldsymbol{\theta}_i$, on a white background.

We begin the texture generation by using Metashape to generate a texture based on our fitted geometry and the complete set of body images $\mathcal{I}_b$. However, due to subject movement, this initial texture often contains artifacts, such as parts of the background. This texture is then refined using the inverse rendering framework provided by PyTorch3D [44] – a library for deep learning with 3D data. We optimize our loss L_{tex} by minimizing the difference between our rendered avatar and the input images through adjusting the texture $\mathbf{I}_{\text{tex}}$. Because scans are performed in environments with uncontrolled lighting conditions, lighting and material parameters are unknown. Therefore, the avatar is rendered without lighting, which leads to baked-in lighting in the texture. This optimization takes advantage of the SSIM, TV, and differentiable Charbonnier L_1 loss functions from Kornia [45]. For optimization, we use the Adam optimizer [26]. Afterwards a head texture is generated via Metashape from the set of head images and merged with the body texture, similar to [35].

4 Evaluation

We compare our approach against three other recent smartphone-based avatar reconstruction methods. *RMAvatar* [43] and *iHuman* [41] are 3DGS-based approaches, *Avatars for the Masses* (A4M) [35] is a mesh-based reconstruction. Our avatars consist of a set of rigged triangle meshes with 51 blendshapes and are about 25MB of size in GLB format including texture. In this section, we show that while 3DGS-based avatar reconstruction methods are excellent in reconstructing views and poses similar to training data, they struggle with creating *novel* poses and views. This, however, is crucial for VR applications as users control their avatar, creating unpredictable poses and views.

All methods use monocular cameras/smartphones to capture the necessary data for their algorithms. A4M and our approach use photos that are taken by circumambulating a person standing in A-pose. To capture the images, we used the iOS application from A4M. RMAvatar and iHuman use short videos as input, where the subject is moving in front of the camera. We followed the input protocol suggested by the authors and recorded a short 1440×1440 video of a person rotating in A-pose in front of a static smartphone. iHuman's and RMAvatar's provided reconstruction pipeline expect scans in PeopleSnapshot format with refined SMPL [32] poses through Anim-NeRF. We therefore convert the videos by using *VideoAvatars* [4] and *Anim-NeRF* [42] and subsequently process the resulting data by the provided scripts of RMAvatar and iHuman, as proposed by the authors. We recorded the videos for iHuman and RMAvatar with the same iPhone that we used to capture the input images for A4M and our approach. We also tried to train RMAvatar and iHuman with our input images, using the same preprocessing pipeline as for the videos. However, this resulted in reconstructions of lower quality. Therefore, for a fair comparison, we used the input data that resulted in the best reconstructions for each method.

4.1 Qualitative Evaluation

Figures 7 and 8 show the results of iHuman [41], RMAvatar [43], Avatars for the Masses (A4M) [35], and our method. Figure 7 shows the reconstructions in a pose of the training sequence. For the female subject[1] all methods created plausible shapes/geometry. Regarding the color/texture the results are different. iHuman has produced visible turquoise artifacts at the arms and the result is blurry, e.g., at the legs. The other methods computed better colors/textures, with RMAvatar producing the most photorealistic results. For the male subject, the result is mostly similar. iHuman produced blurrier results than RMAvatar and our method (e.g, in the face). However, A4M failed to correctly reconstruct the person's geometry, with severe artifacts on both arms. The model's right arm contains a partial image of a car tire/headlight from the background. These artifacts result from motion during scanning. Note that the male subject was not instructed to keep his arms still, highlighting the effect of motion on reconstruction. A4M assumes that during the photo capture no movement occurs, leading to severe deformations if the assumption is violated.

Figure 8 shows two subjects in a novel pose that was not part of the training sequence. A4M's and our results produce plausible geometries and sharp textures, although the texture of our result shows fewer artifacts, e.g., the lower arm of the bottom subject. In contrast, the results of iHuman and RMAvatar are blurrier than in the training pose and show strong deformations in the arms, legs, and face, resulting in the loss of identifiable features. These differences between the training pose and the new pose result from the 3DGS reconstruction method. In contrast to the purely mesh-based approaches, the 3DGS-based approaches incorporate view-dependent material appearance through spherical harmonics. This is usually a key advantage for modeling fine details (e.g. hair); in this case, however, it becomes a disadvantage due to the lack of training data for the novel pose and novel view. Our experiments show that a short smartphone video is not sufficient to capture enough views to create novel poses faithfully. However, this is a key necessity for VR usage, as users control their avatars, leading to a variety of novel views and poses. Therefore, current smartphone-3DGS-based methods cannot be used to generate VR-ready avatars. More results and edge cases are shown in the supplementary material and video.

4.2 Quantitative Evaluation

We quantitatively evaluate our avatars by reprojecting them onto the masked input images. The masked background is replaced with white and metrics are evaluated by comparing the rendered and the input image. We report SSIM, peak-signal-to-noise ratio (PSNR), intersection over union (IoU), and computation time results in Table 1.

PSNR and SSIM measure the reprojection accuracy of the avatar on RGB images, thereby evaluating geometry and texture reconstruction quality. IoU in contrast is evaluated on binary images therefore measuring the difference between the rendered avatar's silhouette and the silhouette of the scanned person. To compute

[1]The subject asked to remain anonymous

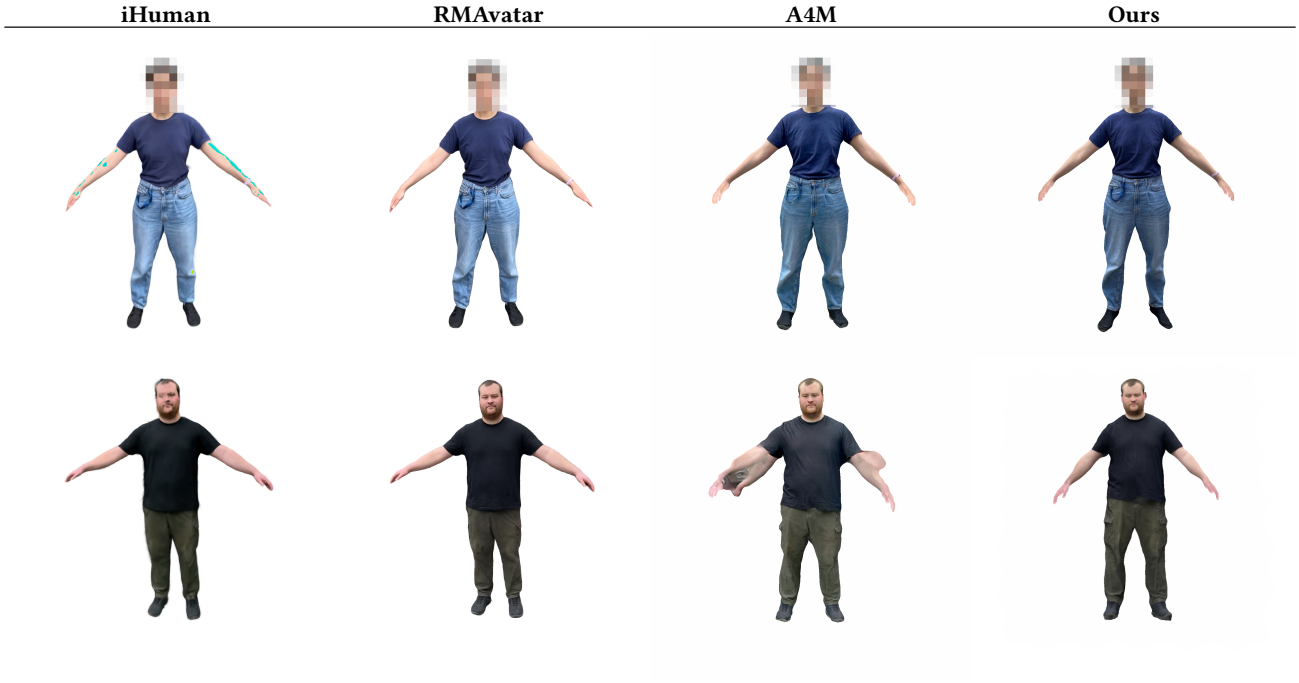


Figure 7: Comparison of the different avatar reconstructions in A-pose from the training sequence. The results from iHuman are blurry and can contain differently colored artifacts, RMAvatar produces the most photorealistic results, A4M contains visible motion artifacts in the arms. Our result is geometrically accurate and has a sharp texture. It does not contain clearly visible motion artifacts, although it is also a purely mesh-based approach.

Metric	iHuman	RMAvatar	A4M	Ours
SSIM ↑	0.957	**0.983**	0.937	0.948
PSNR ↑	25.677	**31.780**	23.747	24.395
IoU ↑	98.90 %	99.21 %	99.04 %	**99.24 %**
Time ↓	25 h	26.5 h	21 min	**19 min**
Frame Rate↑	161 fps	115 fps	**4615 fps**	**4615 fps**

Table 1: Comparison of SSIM, PSNR, IoU errors, computation time, and rendering performance (fps) for three state-of-the-art avatar reconstruction methods and our method averaged over 11 subjects (see supplementary material, subjects 1–11). Best values are in bold. In SSIM and PSNR the 3DGS-based iHuman and RMAvatar are more accurate. Our method is the most accurate with respect to IoU. Ours is the fastest, with 19 min of processing time. iHuman and RMAvatar need more than 25 h and A4M takes slightly longer with 21 min. We improve upon A4M in every metric. iHuman and RMAvatar have considerably lower rendering performance than A4M and ours.

the metrics, we used OpenCV's [8] *quality* module. Compared to 3DGS-based approaches, our avatars achieve similar image metrics: Our SSIM and PSNR are slightly below iHuman's and considerably lower than RMAvatar's but we outperform A4M in both metrics. Our method achieves the highest IoU, with RMAvatar close behind.

Our pipeline was run on a workstation equipped with an AMD Ryzen 7950X CPU and a Nvidia RTX 4090 GPU. The preprocessing and training of iHuman and RMAvatar were performed on a compute server with an AMD Ryzen Threadripper PRO 5975WX CPU and three Nvidia RTX6000 GPUs. Avatars from A4M were computed by the provided Mac Studio server (M1 Ultra 20 core CPU and 64 core GPU). We report computation times including all preprocessing steps and the actual training/fitting of the different pipelines to compare the full duration from scanning to using the avatars. Rendering speed was evaluated on a Nvidia RTX 4090 GPU powered machine in 2160×2160 resolution. Over 11 subjects, RMAvatar took the longest, with average computation times of 26.1 hours. This includes 25 hours of preprocessing with Anim-NeRF and VideoAvatars, as well as around 1.1 hours of additional training time. As iHuman uses the same preprocessing, it shares the 25 hours of preprocessing time but only adds around two minutes of extra training time. Results from A4M were received after around 21 minutes, while ours took around 19 minutes when using 3 subscans. Ours therefore outperforms every compared approach in processing time. Rendering is 28–40× faster than the 3DGS-based methods and matches our previous approach A4M, since both use the same template model. The iHuman and RMAvatar avatars consist of approximately 200k and 120k Gaussians, respectively, while A4M and our mesh-based avatars have around 24k vertices and 45k triangles.

Overall, our results are slightly worse in SSIM and PSNR compared to RMAvatar and iHuman but improve on IoU. However,

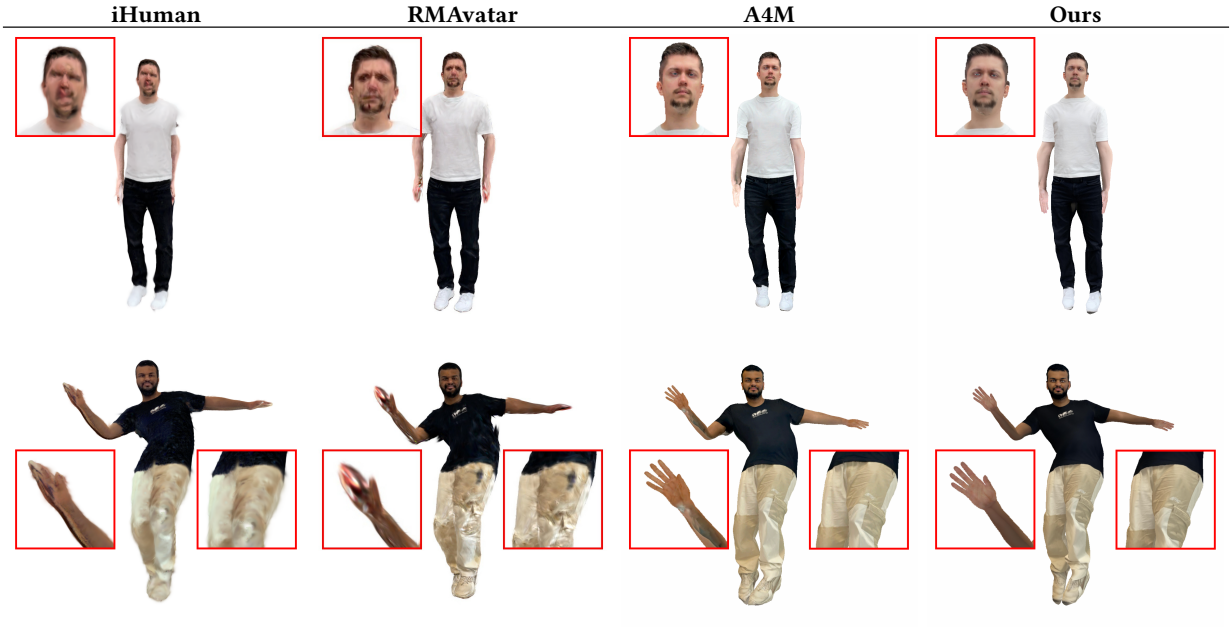


Figure 8: Comparison of the different avatar reconstructions in a novel pose. All resulting avatars can be animated using a precomputed animation but the results of iHuman and RMAvatar have visible artifacts in parts that were not seen (from this angle) in the training sequence (e.g. face, arm, or leg). In contrast, the purely mesh-based approaches do not produce these noticeable artifacts in geometry.

as shown in Figure 8, ours do not exhibit novel pose artifacts in contrast to the 3DGS-based approaches. Our pipeline runs slightly faster than A4M, while iHuman and RMAvatar take more than a day to finish the reconstruction. Compared to A4M, our results exhibit no motion artifacts and improve on reconstruction accuracy in every metric measured.

4.3 Ablation Studies

In this section, we show the influence of the number of sub-scans and the usage of silhouette constraints on the fitting quality.

4.3.1 Sub-Scans. First, we show the impact of using sub-scans by varying the number of sub-scans from two to five with one image overlap between two consecutive sub-scans. As we have 45 images for the body, higher amounts of sub-scans would result in using fewer than seven images per sub-scan, which are too few perspectives to cover enough area of the scanned person in each sub-scan. Figure 9 shows how the usage of sub-scans influences the pose alignment of the avatar to the input images and the geometric accuracy. The subject's right arm is misaligned and malformed when using no sub-scans (left). Using two sub-scans greatly improved the alignment (center), and with five sub-scans, the pose and geometry of the scanned person are matched almost perfectly (right). The geometric detail of the model is preserved in most regions. Cloth detail and skin structure are local features that are reconstructed even when using partial point clouds from five sub-scans. This is due to the fact that increasing the number of sub-scans increases the likelihood of getting motion-free sequences. Our per sub-scan aligned template fitting is then able to reconstruct pose

Figure 9: Comparison of avatar renderings in scan pose over input images (transparent) using 0 (left), 2 (middle) and 5 (right) sub-scans. An increasing number of sub-scans improves pose alignment and geometric accuracy (right arm).

and geometry faithfully. This is also visible in the texture. Without sub-scans, the texture of the subject's right arm contains areas of the background of the input images. With sub-scans, this issue is significantly reduced.

Although the error-reduction effect is generally observed, the data indicate that using too many sub-scans can increase uncertainty. When increasing the number of sub-scans from two to four, the average PSNR increases from 25.35 to 25.58. However, at five sub-scans PSNR falls to 25.49, because the effective overlap per scan becomes too small. Similar effects can be observed with SSIM and IoU. Data from our eleven scanned subjects suggest that three to four sub-scans produce the best results. A bigger amount of sub-scans can help in cases of exaggerated motion (e.g. Figure 9), however with proper instructions three sub-scans are sufficient to reduce the impact of motions in general. As computation time

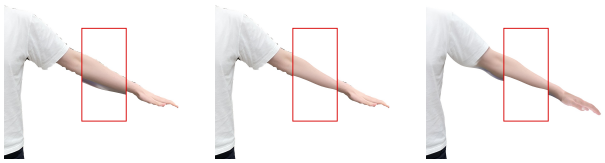


Figure 10: Without silhouette constraints (red box) on input image (left), with silhouette constraints on input image (middle). The right image shows a comparison of both avatar reconstructions: lower arm with silhouette constraints (red box) on the result without silhouette constraints. The silhouette constraint restricts the geometry to the visual hull defined by the masks, leading to a more realistic reconstruction.

Figure 11: Texture-map comparison: Metashape (left) and graph-cut (center) textures from all images versus our inverse-rendering texture generated from sub-scans (right). Only the sub-scan version removes background bleed-through and stitching artifacts, yielding a smooth, photorealistic texture.

increases with the number of sub-scans, we recommend the usage of three sub-scans for a one minute body scan.

4.3.2 Silhouettes. We use silhouette constraints to robustify the fitting process in our pipeline and reduce the influence of motions during the scan procedure even further. Figure 10 shows the influence of silhouette constraints on the reconstruction of the lower arm region. We can see that the proportions of the arms are better matched when using silhouettes. While the use of sub-scans already eliminates most artifacts, silhouette constraints lead to a better alignment with the original shape and position of the body. The choice of silhouette weight λ_{sil} is crucial. When working with exaggerated motions the overlap of silhouettes becomes extremely thin. This leads to arms becoming thinner the higher λ_{sil} is chosen. However our experiments show, that 10^{-4} is an appropriate weight for all our scans. Silhouette constraints have a positive effect on every image metric. This effect is more pronounced when dealing with a lot of motion. On average PSNR improves from 25.2 to 25.7. SSIM also improves from 0.959 to 0.960.

4.3.3 Texture Generation. We compare texture generation from all images with our inverse rendering texture generation from the sub-scans (see Figure 11). From all images (body and head scan), we compute textures using Metashape (left), a graphcut algorithm [56] (center), and our inverse rendering (right). The Metashape and graphcut textures contain visible dark artifacts and/or parts of the background. In contrast, our texture does not contain these kinds of artifacts. Due to the pose alignment through the sub-scans, the influence of motion is reduced, as the fitted avatar's model is aligning properly with the input images. This results in a smooth, realistic and coherent texture.

4.4 Limitations

Our approach reduces motion-induced errors in photogrammetry-based, template-fitting avatar reconstruction. Some limitations remain:

Imprecise Masks Silhouette constraints robustify the correspondence-driven template fitting algorithm. A necessity for the successful usage are accurate segmentation masks of the scanned subject. If the masks are too imprecise (e.g. missing limbs), these constraints may create additional artifacts.

Motion Our approach reduces the problem of motion-induced errors, but still assumes the person to be standing in A-pose, without intentional motion. While our presented method reduces the influence of slow motion, fast movements during a sub-scan can still lead to artifacts.

Clothes and Hair Clothes, skin, and hair are all represented by the same mesh. This simplification can lead can lead to artifacts for loose clothing or long hair, in particular during animations (see supplementary material/video).

5 Conclusion

We presented a novel smarpthone-based avatar reconstruction method that advances the state of the art in producing faithful and VR-ready full-body avatars. By utilizing silhouette constraints and a new sub-scan strategy, paired with inverse rendering texture generation, we outperform recent mesh- and 3DGS-based avatar reconstruction methods, in terms of computation time, accuracy, and animation quality. Our avatars are view-consistent and compatible with common game engines and VR applications without further postprocessing to faithfully represent the users.

Hair and clothing could be improved by using body-part segmentation, which would allow separate fitting to the skin mesh. This approach could also reduce motion restrictions by enabling to fit rigid parts individually. An in-headset user study could be conducted to validate the improvements for VR applications. Unlike existing 3DGS-based methods, we would like to combine textured mesh- and 3DGS-based rendering to improve the rendering quality of fine details of mesh-based avatars.

Acknowledgments

The authors are very grateful to all scanned subjects. This research has been funded by the Ministry of Culture and Science of the State of North Rhine-Westphalia through the project inVirtuo 4.0 (PB22-063-B) and by the Federal Ministry of Education and Research of Germany and the state of North Rhine-Westphalia as part of the Lamarr Institute for Machine Learning and Artificial Intelligence.

References

[1] Jascha Achenbach, Thomas Waltemate, Marc Erich Latoschik, and Mario Botsch. 2017. Fast Generation of Realistic Virtual Humans. In *Proceedings of the ACM Symposium on Virtual Reality Software and Technology*. Article 12, 10 pages. doi:10.1145/3139131.3139154

[2] Agisoft. 2025. Agisoft Metashape. https://www.agisoft.com [visited on 2025-06-05].

[3] Thiemo Alldieck, Marcus Magnor, Bharat Lal Bhatnagar, Christian Theobalt, and Gerard Pons-Moll. 2019. Learning to Reconstruct People in Clothing From a Single RGB Camera. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

[4] Thiemo Alldieck, Marcus Magnor, Weipeng Xu, Christian Theobalt, and Gerard Pons-Moll. 2018. Detailed Human Avatars from Monocular Video. In *International Conference on 3D Vision (3DV)*. 98–109. doi:10.1109/3DV.2018.00022

[5] Thiemo Alldieck, Marcus Magnor, Weipeng Xu, Christian Theobalt, and Gerard Pons-Moll. 2018. Video Based Reconstruction of 3D People Models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[6] Apple Inc. 2025. ARKit Framework - Blendshapes. https://developer.apple.com/documentation/arkit/arfaceanchor/blendshapes [visited on 2025-06-23].

[7] Mario Botsch, Leif Kobbelt, Mark Pauly, Pierre Alliez, and Bruno Lévy. 2010. *Polygon Mesh Processing*. CRC Press.

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

[9] Samuel Buss. 2004. Introduction to inverse kinematics with Jacobian transpose, pseudoinverse and damped least squares methods. *IEEE Transactions in Robotics and Automation* 17 (2004).

[10] Z. Cao, G. Hidalgo Martinez, T. Simon, S. Wei, and Y. A. Sheikh. 2019. OpenPose: Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2019).

[11] P. Charbonnier, L. Blanc-Feraud, G. Aubert, and M. Barlaud. 1994. Two deterministic half-quadratic regularization algorithms for computed imaging. In *Proceedings of 1st International Conference on Image Processing*, Vol. 2. 168–172 vol.2. doi:10.1109/ICIP.1994.413553

[12] Jianchuan Chen, Jingchuan Hu, Gaige Wang, Zhonghua Jiang, Tiansong Zhou, Zhiwen Chen, and Chengfei Lv. 2025. TaoAvatar: Real-Time Lifelike Full-Body Talking Avatars for Augmented Reality via 3D Gaussian Splatting. In *Proceedings of the Computer Vision and Pattern Recognition Conference (CVPR)*. 10723–10734.

[13] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. 2017. Rethinking Atrous Convolution for Semantic Image Segmentation. *CoRR* abs/1706.05587 (2017). arXiv:1706.05587

[14] Xiang Deng, Zerong Zheng, Yuxiang Zhang, Jingxiang Sun, Chao Xu, Xiaodong Yang, Lizhen Wang, and Yebin Liu. 2024. RAM-Avatar: Real-time Photo-Realistic Avatar from Monocular Videos with Full-body Control. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 1996–2007.

[15] Andrew Feng, Evan Suma, and Ari Shapiro. 2017. Just-in-Time, Viable, 3D Avatars from Scans. In *ACM SIGGRAPH 2017 Talks* (Los Angeles, California). Article 19, 2 pages. doi:10.1145/3084363.3085045

[16] Marie Luisa Fiedler, Erik Wolf, Nina Döllinger, David Mal, Mario Botsch, Marc Erich Latoschik, and Carolin Wienrich. 2024. From Avatars to Agents: Self-Related Cues through Embodiment and Personalization Affect Body Perception in Virtual Reality. *IEEE Transactions on Visualization and Computer Graphics* (2024). doi:10.1109/TVCG.2024.3456211

[17] Chen Guo, Junxuan Li, Yash Kant, Yaser Sheikh, Shunsuke Saito, and Chen Cao. 2025. Vid2Avatar-Pro: Authentic Avatar from Videos in the Wild via Universal Prior. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

[18] Marc Habermann, Lingjie Liu, Weipeng Xu, Gerard Pons-Moll, Michael Zollhoefer, and Christian Theobalt. 2023. HDHumans: A Hybrid Approach for High-fidelity Digital Humans. *Proc. ACM Comput. Graph. Interact. Tech.* 6, 3, Article 36 (2023), 23 pages. doi:10.1145/3606927

[19] Liangxiao Hu, Hongwen Zhang, Yuxiang Zhang, Boyao Zhou, Boning Liu, Shengping Zhang, and Liqiang Nie. 2024. GaussianAvatar: Towards Realistic Human Avatar Modeling from a Single Video via Animatable 3D Gaussians. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 634–644.

[20] Shoukang Hu, Tao Hu, and Ziwei Liu. 2024. GauHuman: Articulated Gaussian Splatting from Monocular Human Videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 20418–20431.

[21] Forrest Iandola, Stanislav Pidhorskyi, Igor Santesteban, Divam Gupta, Anuj Pahuja, Nemanja Bartolovic, Frank Yu, Emanuel Garbin, Tomas Simon, and Shunsuke Saito. 2025. SqueezeMe: Mobile-Ready Distillation of Gaussian Full-Body Avatars *(SIGGRAPH Conference Papers '25)*. Article 91, 11 pages. doi:10.1145/3721238.3730599

[22] Boyi Jiang, Yang Hong, Hujun Bao, and Juyong Zhang. 2022. SelfRecon: Self Reconstruction Your Digital Avatar from Monocular Video. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

[23] Tianjian Jiang, Xu Chen, Jie Song, and Otmar Hilliges. 2023. InstantAvatar: Learning Avatars From Monocular Video in 60 Seconds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 16922–16932.

[24] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 2023. 3D Gaussian Splatting for Real-Time Radiance Field Rendering. *ACM Transactions on Graphics* 42, 4 (2023).

[25] Do Yuon Kim, Ha Kyung Lee, and Kyunghwa Chung. 2023. Avatar-mediated experience in the metaverse: The impact of avatar realism on user-avatar relationship. *Journal of Retailing and Consumer Services* 73 (2023). doi:10.1016/j.jretconser.2023.103382

[26] Diederik P. Kingma and Jimmy Ba. 2014. Adam: A Method for Stochastic Optimization. arXiv:1412.6980 [cs.LG]

[27] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollar, and Ross Girshick. 2023. Segment Anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 4015–4026.

[28] Youngjoong Kwon, Lingjie Liu, Henry Fuchs, Marc Habermann, and Christian Theobalt. 2023. DELIFFAS: Deformable Light Fields for Fast Avatar Synthesis. In *Advances in Neural Information Processing Systems*, Vol. 36. 40944–40962.

[29] Inhee Lee, Byungjun Kim, and Hanbyul Joo. 2024. Guess The Unseen: Dynamic 3D Scene Reconstruction from Partial 2D Glimpses. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 1062–1071.

[30] Jiahui Lei, Yufu Wang, Georgios Pavlakos, Lingjie Liu, and Kostas Daniilidis. 2024. GART: Gaussian Articulated Template Models. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 19876–19887. doi:10.1109/CVPR52733.2024.01879

[31] Zhe Li, Zerong Zheng, Lizhen Wang, and Yebin Liu. 2024. Animatable Gaussians: Learning Pose-dependent Gaussian Maps for High-fidelity Human Avatar Modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 19711–19722.

[32] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. 2015. SMPL: a skinned multi-person linear model. *ACM Trans. Graph.* 34, 6, Article 248 (2015), 16 pages. doi:10.1145/2816795.2818013

[33] Camillo Lugaresi, Jiuqiang Tang, Hadon Nash, Chris McClanahan, Esha Uboweja, Michael Hays, Fan Zhang, Chuo-Ling Chang, Ming Yong, Juhyun Lee, Wan-Teh Chang, Wei Hua, Manfred Georg, and Matthias Grundmann. 2019. MediaPipe: A Framework for Perceiving and Processing Reality. In *Third Workshop on Computer Vision for AR/VR at IEEE Computer Vision and Pattern Recognition (CVPR) 2019*.

[34] S. Ma, T. Simon, J. Saragih, D. Wang, Y. Li, F. La Torre, and Y. Sheikh. 2021. Pixel Codec Avatars. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 64–73. doi:10.1109/CVPR46437.2021.00013

[35] Timo Menzel, Erik Wolf, Stephan Wenninger, Niklas Spinczyk, Lena Holderrieth, Carolin Wienrich, Ulrich Schwanecke, Marc Erich Latoschick, and Mario Botsch. 2025. Avatars for the Masses: Smartphone-Based Reconstruction of Humans for Virtual Reality. *Frontiers in Virtual Reality* 6 (2025), 1583474. doi:10.3389/frvir.2025.1583474

[36] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. 2020. NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. In *ECCV*.

[37] Gyeongsik Moon, Takaaki Shiratori, and Shunsuke Saito. 2024. Expressive Whole-Body 3D Gaussian Avatar. In *ECCV*.

[38] Arthur Moreau, Jifei Song, Helisa Dhamo, Richard Shaw, Yiren Zhou, and Eduardo Pérez-Pellitero. 2024. Human Gaussian Splatting: Real-time Rendering of Animatable Avatars. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 788–798.

[39] Wieland Morgenstern, Milena T. Bagdasarian, Anna Hilsmann, and Peter Eisert. 2024. Animatable Virtual Humans: Learning Pose-Dependent Human Representations in UV Space for Interactive Performance Synthesis. *IEEE Transactions on Visualization and Computer Graphics* 30, 5 (2024), 2644–2650. doi:10.1109/TVCG.2024.3372117

[40] Haokai Pang, Heming Zhu, Adam Kortylewski, Christian Theobalt, and Marc Habermann. 2024. ASH: Animatable Gaussian Splats for Efficient and Photoreal Human Rendering. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 1165–1175. doi:10.1109/CVPR52733.2024.00117

[41] Pramish Paudel, Anubhav Khanal, Danda Pani Paudel, Jyoti Tandukar, and Ajad Chhatkuli. 2025. iHuman: Instant Animatable Digital Humans From Monocular Videos. In *Computer Vision – ECCV 2024*. 304–323. doi:10.1007/978-3-031-73226-3_18

[42] Sida Peng, Junting Dong, Qianqian Wang, Shangzhan Zhang, Qing Shuai, Xiaowei Zhou, and Hujun Bao. 2021. Animatable Neural Radiance Fields for Modeling Dynamic Human Bodies. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*. 14294–14303. doi:10.1109/ICCV48922.2021.01405

[43] Sen Peng, Weixing Xie, Zilong Wang, Xiaohu Guo, Zhonggui Chen, Baorong Yang, and Xiao Dong. 2025. RMAvatar: Photorealistic human avatar reconstruction from monocular video based on rectified mesh-embedded Gaussians. *Graphical Models* 139 (2025), 101266. doi:10.1016/j.gmod.2025.101266

[44] Nikhila Ravi, Jeremy Reizenstein, David Novotny, Taylor Gordon, Wan-Yen Lo, Justin Johnson, and Georgia Gkioxari. 2020. Accelerating 3D Deep Learning with PyTorch3D. *arXiv:2007.08501* (2020).

[45] E. Riba, D. Mishkin, J. Shi, D. Ponsa, F. Moreno-Noguer, and G. Bradski. 2020. A survey on Kornia: an Open Source Differentiable Computer Vision Library for PyTorch.

[46] K.M. Robinette, H. Daanen, and E. Paquet. 1999. The CAESAR project: a 3-D surface anthropometry survey. In *Second International Conference on 3-D Digital Imaging and Modeling (Cat. No.PR00062)*. 380–386. doi:10.1109/IM.1999.805368

[47] Anca Salagean, Eleanor Crellin, Martin Parsons, Darren Cosker, and Danaë Stanton Fraser. 2023. Meeting Your Virtual Twin: Effects of Photorealism and Personalization on Embodiment, Self-Identification and Perception of Self-Avatars in Virtual Reality. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*. Article 499, 16 pages. doi:10.1145/3544548.3581182

[48] Zhijing Shao, Zhaolong Wang, Zhuang Li, Duotun Wang, Xiangru Lin, Yu Zhang, Mingming Fan, and Zeyu Wang. 2024. SplattingAvatar: Realistic Real-Time Human Avatars with Mesh-Embedded Gaussian Splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 1606–1616.

[49] Ashwath Shetty, Marc Habermann, Guoxing Sun, Diogo Luvizon, Vladislav Golyanik, and Christian Theobalt. 2024. Holoported Characters: Real-time Free-viewpoint Rendering of Humans from Sparse RGB Cameras. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 1206–1215.

[50] Lucia R. Valmaggia, Leila Latif, Matthew J. Kempton, and Maria Rus-Calafell. 2016. Virtual reality in the psychological treatment for mental health problems: An systematic review of recent evidence. *Psychiatry Research* 236 (2016), 189–195. doi:10.1016/j.psychres.2016.01.015

[51] Thomas Waltemate, Dominik Gall, Daniel Roth, Mario Botsch, and Marc Erich Latoschik. 2018. The Impact of Avatar Personalization and Immersion on Virtual Body Ownership, Presence, and Emotional Response. *IEEE Transactions on Visualization and Computer Graphics* 24, 4 (2018), 1643–1652. doi:10.1109/TVCG.2018.2794629

[52] Rong Wang, Fabian Prada, Ziyan Wang, Zhongshi Jiang, Chengxiang Yin, Junxuan Li, Shunsuke Saito, Igor Santesteban, Javier Romero, Rohan Joshi, Hongdong Li, Jason Saragih, and Yaser Sheikh. 2025. FRESA: Feedforward Reconstruction of Personalized Skinned Avatars from Few Images. In *Proceedings of the Computer Vision and Pattern Recognition Conference (CVPR)*. 281–291.

[53] Shaofei Wang, Bozidar Antic, Andreas Geiger, and Siyu Tang. 2024. IntrinsicAvatar: Physically Based Inverse Rendering of Dynamic Humans from Monocular Videos via Explicit Ray Tracing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 1877–1888.

[54] Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. 2004. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing* 13, 4 (2004), 600–612. doi:10.1109/TIP.2003.819861

[55] Jing Wen, Xiaoming Zhao, Zhongzheng Ren, Alexander G. Schwing, and Shenlong Wang. 2024. GoMAvatar: Efficient Animatable Human Modeling from Monocular Video Using Gaussians-on-Mesh. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2059–2069.

[56] Stephan Wenninger, Jascha Achenbach, Andrea Bartl, Marc Erich Latoschik, and Mario Botsch. 2020. Realistic Virtual Humans from Smartphone Videos. In *Proceedings of the ACM Symposium on Virtual Reality Software and Technology*. Article 29, 11 pages. doi:10.1145/3385956.3418940

[57] Xihe Yang, Xingyu Chen, Daiheng Gao, Shaohui Wang, Xiaoguang Han, and Baoyuan Wang. 2024. HAVE-FUN: Human Avatar Reconstruction from Few-Shot Unconstrained Images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 742–752.

[58] Zhichao Zhai, Guikun Chen, Wenguan Wang, Dong Zheng, and Jun Xiao. 2025. TAGA: Self-supervised Learning for Template-free Animatable Gaussian Articulated Model. In *Proceedings of the Computer Vision and Pattern Recognition Conference (CVPR)*. 21159–21169.