MoRF: Mobile Realistic Fullbody Avatars from a Monocular Video

Renat Bashirov1 Alexey Larionov1 Evgeniya Ustinova1 Mikhail Sidorenko1
David Svitov1 Ilya Zakharkin2* Victor Lempitsky3*
1Samsung Research 2ZERO10 3Cinemersive Labs

Abstract

We present a system to create Mobile Realistic Fullbody (MoRF) avatars. MoRF avatars are rendered in real-time on mobile devices, learned from monocular videos, and have high realism. We use SMPL-X as a proxy geometry and render it with DNR (neural texture and image-2-image network). We improve on prior work, by overfitting per-frame warping fields in the neural texture space, allowing to better align the training signal between different frames. We also refine SMPL-X mesh fitting procedure to improve the overall avatar quality. In the comparisons to other monocular video-based avatar systems, MoRF avatars achieve higher image sharpness and temporal consistency. Participants of our user study also preferred avatars generated by MoRF.

1. Introduction

This work focuses on fullbody human avatars with fast rendering, realism and the ease of acquisition. For our system we: 1) use short (i.e. minute-long) monocular videos as input, 2) train for a few hours on a single GPU, 3) provide sharp texture of the resulting avatar, and 4) can render in real-time on mobile devices. As far as we know, despite impressive progress, existing systems fall short of these criteria. Specifically, some rely on multi-view videos [39, 52], or a depth sensor [34, 62], some leverage implicit representations whose rendering process is far from real-time especially on mobile devices [5, 21, 36, 42, 48, 49], whilst others take days to converge on multiple GPUs [64, 68].

We therefore present a system for Mobile Realistic Fullbody (MoBF) avatars. Our avatars are realistic and exhibit temporal consistency of generated images, and they can be rendered in realtime at 30 FPS on mobile devices (Fig. 1) with Qualcomm Snapdragon 888 SoC (2020’s flagship SoC for mobile phones). Furthermore, they can be acquired from monocular videos in a few hours on a single NVIDIA RTX 3090 GPU. We rely on the deferred neural rendering (DNR) idea [60], like in numerous prior avatar systems [53, 67–69], to achieve rendering speed and realism. DNR represents 3D objects using coarse 3D geometry and neural textures, which are supplemented by a rendering image-2-image translation network to produce realistic images. For geometric modeling, we use the well-established SMPL-X body model [47].

We discovered that naively implementing the DNR approach results in underfitting blurriness or overfitting flickering. To address them, we introduce a new component into the DNR learning process that we call the neural texture warping. For each training image/frame, it fits a warping field for the neural texture, thus compensating between-frame mesh fit misalignment and facilitating the task for image-2-image UNet renderer.

The warping fields are only used for training to compensate for mesh fitting imperfections, this produces sharp and static texture for the resulting avatar. We don’t solve the task...
of clothing modelling/movement and the warping fields are discarded after the training process has converged.

We evaluate our approach on 3 datasets: self-captured, ZJU-MoCap [13, 49] and People Snapshot [3]. Self-captured and ZJU-MoCap provide ground truth for novel body and camera poses respectively. We also conduct user studies for avatar rendering quality and for mesh fitting refinements ablation. We compare against InstantAvatar [26], HF-Avatar [71], Anim-NeRF [48], HumanNeRF [66], StylePeople [17], ANR [53] and NeuMan [27].

2. Related Work

Modeling geometry of dynamic non-rigid scenes is considered in a number of recent approaches either for capturing human actors [6, 18, 20, 23, 33, 35, 36, 40, 61, 66] or general scenes [38, 46, 51]. Towards this end, 3D scans are required for supervision in [20, 23, 35, 40, 61] to learn rigid human geometry. Likewise, [6, 18, 33, 36, 38, 50, 54] utilize multi-view data to capture appearance and produce photorealistic avatars under arbitrary viewpoints and in arbitrary poses. Several methods [46, 51, 66] use monocular videos but allow free-viewpoint rendering only.

Some of the monocular fullbody avatar methods model the human geometry implicitly [36, 48, 66], others output the classical mesh+texture format [1, 2, 4, 17, 53, 59, 71]. Similarly to StylePeople [17] and ANR [53], our method is based on DNR [60] allowing for efficient convolutional rendering on a mobile device. However, StylePeople leaves modeling of geometry imperfections and frame-to-frame misalignments to the neural renderer, while ANR handles these issues by limiting the number of frames used for the neural texture optimization. HF-Avatar [71] uses the synthetic images as supervision for the neural texture, because of their consistency and good alignment with the geometry. [2] proposes a texture merging optimization procedure to prevent texture averaging among different views. Our neural texture warping allows for consistent learning of the neural texture without training on synthetic data [71], excluding most frames from texture learning [53], or overfitting the neural renderer to unmodeled variations [17].

Many of the mentioned works introduce learned sub-modules [18, 33, 35, 36, 38, 61, 66, 71] or pipeline stages [2, 4] to estimate additional geometry warping in 3D canonical (unposed or view-agnostic) space. Likewise, [33, 35, 36, 61, 66] parametrize the 3D warping fields or displacements [40, 71] as neural networks conditioned on human pose to facilitate learning of complex pose-dependent geometry. Methods for general scenes leverage other variants of conditioning. For example [46, 51], that rely on monocular videos, condition 3D warping fields on the point in time and learnable latent of a training frame respectively, while [38] introduce view-conditional warpings as they use multi-view capture. The way we introduce warping in this work is different to all above-referenced approaches, as our learnable warping operates in the canonical 2D texture space. Operating in the texture space naturally fits the DNR approach that we are based upon.

3. Method

Our approach aims to learn a rigged full-body human avatar using a relatively short (e.g. a minute-long) monocular video. We begin by briefly describing deferred neural rendering framework, as it is at the core of our system. We then introduce the neural texture warping that facilitates the training process. We also mention important architectural decisions that contribute to the avatars quality.
blue colors correspond to the two poses from misaligned. The texture space is clearly
3.2. Neural Texture Warping
Figure 3. (a) Mesh surfaces (represented by red lines) fitted to two different frames are located inconsistently. Because of the rendering network’s overfitting tendency, such misalignment can cause artifacts of the learned avatar. (b) A partial plot of $\Pi_T(I_2) - \Pi_T(I_1)$, where $\Pi_T$ is the texture space projection. The red and blue colors correspond to the two poses from (a), whilst the white color indicates that the colors match. The texture space is clearly misaligned.

3.1. Recap: Deferred Neural Rendering
Deferred neural rendering (DNR) [60] models scenes by combining a geometric proxy defined by a triangular mesh, a neural texture $T$ (with $C$ channels), and an image-to-image rendering network (renderer) $\mathcal{R}$ with a convolutional architecture. Novel views for certain camera parameters are synthesized in two steps. The first is to project the mesh onto the camera view, while superimposing the texture on the mesh using texture mapping. The resulting $C$-channeled image can then be “translated” into an RGB image (plus optionally a mask) by the rendering network. During training/fitting to a given dataset of images with known corresponding camera parameters, the latent texture $T$ and the parameters of the rendering network $\mathcal{R}$ are jointly optimized to minimize losses between training images and images predicted by $\mathcal{R}$.

3.2. Neural Texture Warping
The notable finding of [60] is that the quality of deferred neural rendering degrades very gracefully in the presence of coarseness or systematic errors in the geometric proxy. This has allowed several recent works [53, 67–69] to apply DNR to fullbody modeling using SMPL-X as an underlying geometry, despite the big gap between this geometric proxy and the actual avatar content that might include sliding or loose clothing. DNR based on SMPL-X geometry essentially “off-loads” clothing and hair modeling to the rendering network that operates in the 2D domain. While appealing for its simplicity, we discovered that such an approach frequently fails to fit the training data, because frames with similar camera parameters and body pose can differ in hair geometry or clothing draping (see Figure 3), which the DNR approach cannot model well. We propose modeling them in the texture space with 2D warping maps (see Figure 2). Our cornerstone is that the texture space by construction is a canonical space for the human body – it is invariant to body poses and camera views. At the same time it covers the whole 3D body surface. As a result, our warping approach is less ambiguous than commonly applied warping in the 3D spatial domain, and arguably is easier to train from monocular videos.

For each training frame, we learn a frame-specific warping field for the neural texture. The frame specific information is represented by a combination of the frame’s SMPL-X pose vector $p$ and a latent $z$. Those are passed through a multi-layered perceptron (MLP) $\mathcal{W}$ to obtain a style vector $w$. We predict the warping fields with an encoder-decoder convolutional architecture $E \circ F$, the input to which is a constant tensor of positional encodings of the texture coordinates. The style vector $w$ conditions the encoder part via the Adaptive Instance Normalization blocks [22]. A predicted warping field is a 2-channel tensor, that specifies per-pixel offsets of texture coordinates. See supp. mat. for the exact details of the architectures $E$, $F$, $\mathcal{W}$. Figure 4 shows the effect of the learned warping fields.

During model fitting, we use gradient descent to fit per-frame latent vectors $z$, and parameters of the warping networks $E$, $F$, $\mathcal{W}$, the neural texture $T$ and the rendering network $\mathcal{R}$, which are shared for all training frames. The latent vectors $z$ encode the variations in avatar geometry that were
not captured by the underlying SMPL-X model, and the frame-specific imperfections of SMPL-X fitting (Fig. 4).

For rendering in novel poses, we compute the warping field corresponding to the neutral A-pose \( p \) and the average latent vector \( z \). Such warping field is applied to the neural texture once. First, such an approach is suitable in the case of limited train set. Second, this helps eliminating the need for warping computations during inference, thus preserving high rendering speed.

**Texture initialization** We found that, as compared to random initialization, better results can be achieved with spectral initialization of the neural texture as in StylePeople [17]: each texel value is initialized with spectral coordinates of the corresponding point on the SMPL-X mesh graph (see examples in supp. mat.). We learn neural textures with small learning rates to avoid vanishing of the spectral information. We found that the use of spectral initialization was more pronounced in our system compared to [17]. In particular, this initialization has minor effect on image fidelity, but it significantly slows down overfitting of the rendering network, giving the warping network time to converge and facilitating temporal consistency. It also contributes in plausible inpainting of body parts not covered in the input video (see Figure 5).

### 4. Implementation details

#### 4.1. Mesh fitting

For both our system and the standard DNR model to perform well, precise and time-consistent SMPL-X mesh fitting to the training video frames (and at test time) is important. As for RGB photos, the convolutional regression-based techniques [10, 14, 28, 30, 31, 57, 58] produce somewhat accurate but inconsistent results that frequently mismatch the silhouettes. On the contrary, results of iterative non-linear optimization [7] better correspond 2D RGB images but are less plausible or unstable in 3D, since the fitting process relies on 2D image keypoints, which are poorly constrained along the depth dimension. Regression-based and optimization-based methods are unable to handle complex poses. Below we describe a few enhancements to SMPL-X optimization-based fitting that increase the stability of DNR-based avatar systems such as ours. We later validate the individual impact of these adjustments through ablation study.

**Shared shape** A person’s identity does not change between frames of a training video, thus the shape-specific parameters of SMPL-X model can be shared.

**Silhouette loss** We perform differentiable rendering of SMPL-X meshes via nvdifffrast [32]. We then apply Dice [43] loss to pull rendered SMPL-X silhouettes towards estimations of a semantic segmentation network. In the presence of loose clothing, the inclusion of this term tends to “fatten” the human shape.

**Temporal loss** The non-linear optimization-based fitting method tends to flicker on side views when the person’s farther side is not visible. To compensate for this effect, we minimize the L2 distance between adjacent body pose vectors (in axis-angle format). This term is efficient for our scenario of training videos with slow body movements. Applying it to “in-the-wild” videos with fast motions may be ineffective or even harmful.

**Automatic frame filtering** The mesh fitting procedure is vulnerable to quick movements and depth ambiguity even with temporal guidance. Some training frames may be estimated with significantly incorrect poses. This causes a conflict between the 3D mesh surface and the neural texture, resulting in the appearance of “ghost” limbs and flickering. We use a simple heuristic that we describe in supp. mat. to automatically filter out bad frames (typically, less than 5% of frames are discarded).

#### 4.2. Architectural details

**Training** As a rendering network, we use U-Net [55] (a total of \( \approx14\, 4M \) parameters). Its decoder is based on bi-linear upsampling blocks. We provide the definitions of neural layers in supp. mat. We use ADAM [29] to optimize the architecture from scratch for a target person. On a single RTX 3090 GPU, convergence to plateau takes 8 hours. Our preliminary results suggest that faster acquisition is possible, if the architecture is pretrained on a dataset of people, and then fine-tuned to a target person. Below, however, we report on experiments that fit the model to a target video from scratch.

**Ad-hoc inpainting of unseen body parts** Our minute-long training videos rarely include all body parts of the target person. Often top of the head, soles, and armpits have
We use OpenPose [8] to detect 2D keypoints, which is required for mesh fitting. We retrain Graphonomy [15] segmentation network on CCIHP dataset [37], which includes accessories like watches, belts, and glasses, etc. Meanwhile the publicly available Graphonomy trained on CHIPS dataset predicts those as background.

4.3. Loss functions

For MoRF training, we use popular loss functions from previous works [17, 60, 63] with modifications. We further elaborate on our learning objective:

$$\mathcal{L}_{\text{total}} = \sum_{i \in \mathcal{L}} \lambda_i \mathcal{L}_i \quad (1)$$

\[\mathcal{L} = \{L_1, \text{Percept}, \text{GAN}, \text{Dice, TV}_{\text{RGB}}, \text{TV}_\Delta, \Delta \rightarrow 0\}\]

**Image losses** These include L1 per-pixel loss $L_1$ and the VGG [56] based perceptual loss [12] $L_{\text{Percept}}$. To enforce the realism of the rendered images, we utilized the non-saturating adversarial loss [16] $L_{\text{GAN}}$ on predictions of multi-scale patch-based discriminators (PatchGAN from Isola et al. [25]).

**Segmentation loss** In our case, the estimated body mesh (SMPL-X) is coarse and frequently misaligned with the underlying human content. To assist the renderer in generating content outside of the rasterized mesh, the renderer predicts the human mask in addition to the RGB image. We supervise the masks on the pseudo ground truth masks (see paragraph 4.2), using Dice [43] $L_{\text{Dice}}$ segmentation loss.

**Regularization** For the warping fields, we penalize L2 norm ($L_{\Delta,0}$) to prevent large deviations from identity warping, and also apply total-variation (TV) loss [9] $L_{\text{TV}_\Delta}$ for local smoothness. We use a low weighted TV loss on generated images $L_{\text{TV}_\text{gen}}$ to slightly improve smoothness.

4.4. Avatar rendering on a mobile device

We developed a telepresence mobile app for Qualcomm Snapdragon SoC using Android, ARCore [45], and OpenGL. Our method can run natively at resolution of 640 x 640 pixels and above 30 frames per second (measured on Qualcomm Snapdragon 888, see Fig. 1). We use OpenGL shaders to generate input neural rasterizations and to infer posed SMPL-X meshes from body joint rotations. We stumbled upon OpenGL limit on the depth of the neural texture channels. We were able to pack at most 16 channels of neural rasterizations into OpenGL framebuffers, using quantization [44] into eight-bit integers. Similarly, we use post-training quantization on the weights of the rendering network. This enables the network to run at high speeds on the Qualcomm Digital Signal Processor. Quality degradation due to quantization is relatively low in our system. Because single-person avatars have low variance in appearance, the numerical range of neural network activations is small, which is advantageous for quantization.

5. Experiments

We compare our method to similar recent approaches and provide an ablation study for our mesh fitting adjustments. We conduct quantitative comparisons as well as user study comparisons. We provide sample side-by-side comparisons of the methods in supp. mat.

5.1. Self-captured dataset

We have gathered a dataset of stationary monocular videos of ten people with different demographics, body shapes, and clothes topology. We note that despite improvements in mesh fitting, it is still imperfect, therefore we captured the training videos with a specific scenario that avoids complex poses. In one-minute training videos people move slowly, turn 360°, and show their hands in front of their torso. The one-minute hold-out videos include mostly novel poses not present in training videos, namely various arm and hip movements. See samples of training and hold-out data in supp. mat.

Table 1. Quantitative metrics and the results of the user preference study, in which each method is compared against MoRF. See the text for the discussion of the results.

<table>
<thead>
<tr>
<th></th>
<th>SSIM</th>
<th>LPIPS</th>
<th>tLPIPS</th>
<th>% of users preferred MoRF</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Scenario A:</strong> 512 x 512 px, training only on A-poses</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>InstantAvatar</td>
<td>92.22</td>
<td>7.91</td>
<td>7.52</td>
<td>96.5%↑</td>
</tr>
<tr>
<td>HF-Avatar</td>
<td>93.0</td>
<td>7.25</td>
<td>4.00</td>
<td>90.0%↑</td>
</tr>
<tr>
<td>AnimNeRF</td>
<td>92.7</td>
<td>6.74</td>
<td>2.96</td>
<td>98.0%↑</td>
</tr>
<tr>
<td>MoRF</td>
<td>94.5</td>
<td>3.95</td>
<td>2.56</td>
<td>N/A</td>
</tr>
<tr>
<td><strong>Scenario B:</strong> 640 x 640 px, full training sequences</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HumanNeRF</td>
<td>87.4</td>
<td>13.00</td>
<td>19.3</td>
<td>95.0%↑</td>
</tr>
<tr>
<td>StylePeople</td>
<td>90.0</td>
<td>6.13</td>
<td>4.77</td>
<td>60.5%↑</td>
</tr>
<tr>
<td>ANR</td>
<td>90.0</td>
<td>6.14</td>
<td>4.76</td>
<td>69.2%↑</td>
</tr>
<tr>
<td>MoRF</td>
<td>89.9</td>
<td>6.23</td>
<td>4.66</td>
<td>N/A</td>
</tr>
</tbody>
</table>

We use publicly available implementations of InstantAvatar, HF-Avatar, Anim-NeRF, NeuMan and HumanNeRF. To compare with InstantAvatar, HF-Avatar, Anim-NeRF methods, we exclude the sections of training...
Figure 6. Avatars created with different monocular video-based approaches. MoRF achieves the highest visual quality in this comparison. See text for more details.

Figure 7. To illustrate the difference in temporal consistency, we show the space-time y-t slices (left). In the case of StylePeople (SP), the T-shirt pattern (top plots) abruptly tears vertically (accentuated with red arrows), resulting in (a) flickering, (b) “ghost” patterns, (c) poor clarity, and loss of pattern rendering for rotated human body (see the section labeled “side”). These artifacts are absent in our approach’s results, which also achieves good temporal stability on the hemline (bottom plots). (d) Shows an average image over a 20-frame interval (colored in light-blue).

videos where people show their hands in front of their torso (leaving about half of the frames), this forms Scenario A comparison. Anim-NeRF limits the resolution for Scenario A to 512 × 512 px. HF-Avatar’s mesh fitting adequately performed for training sequences, but struggled for hold-out sequences, therefore for them we used our
modified mesh fitting. HumanNeRF implementation does not include its own mesh fitting method, thus we use ours instead. See Table 2 for an overview of mesh fitting methods applied in our experiments.

We compare against HumanNeRF, ANR and StylePeople trained on full uncut videos at $640 \times 640$ px resolution – the maximum resolution at which we can run 30 FPS on a mobile device. This forms Scenario B comparison. For StylePeople, we use all the proposed components as per [17, 60], but take our segmentation, mesh fitting, losses, spectral texture initialization and architectures of the rendering and discriminator networks. In the comparison table, StylePeople baseline differs from MoRF only by the absence of the neural texture warping, in order to compare only novelty part of our work. To compare with ANR, we re-implement their split optimization scheme and add it to the MoRF result (in randomized order). We took novel body rendering and discriminator networks. In the comparison losses, spectral texture initialization and architectures of the other methods.

To quantify visual quality, we use three metrics, computed between real and generated images. We employ two image-based metrics: SSIM [65] and LPIPS [70]. To measure temporal stability, we use tLPIPS [11].

We conduct a user study (the right-most column of Table 1) in which we show pairs of sequences, one of which is the MoRF result (in randomized order). We took novel body motion sequences from AMASS [41] dataset. We asked people to indicate their preference, and thus obtain the percentage of responses when MoRF was preferred over each of the other methods.

The results are shown in Table 1. In our comparison, AnimNeRF and InstantAvatar produce NeRF blob artifacts. HumanNeRF similarly exhibits NeRF artifacts in motion, and its training diverged for some training videos, with the reasons unrelated to MoRF’s mesh fitting quality (see supp. mat. for evaluation scores and images). Due to such divergence, HumanNeRF has very high values for LPIPS and tLPIPS metrics. Non-NeRF HF-Avatar method suffers from significant blurring. In comparison to StylePeople and ANR, our approach outperforms both of them in temporal stability while slightly underperforming in SSIM and LPIPS metrics. We explain it by noticing, that when turning off warping fields on inference, the rendered avatar’s texture appears in a neutral position (Fig. 4), which is almost certainly misaligned with the real dynamic garments from the hold-out ground truth. Thus, the metrics capture mostly pixel-wise discrepancy between garments, which might not match the human perception of pleasant avatars.

Qualitatively, we observe at least as good image quality for MoRF as that of StylePeople and ANR. The advantage of our method is validated by the user study on the videos, where MoRF is preferred over other methods. We also provide a qualitative comparison of MoRF with StylePeople in our mobile AR app in supp. mat.

In Figure 6 we show the qualitative comparisons between the methods. In Figure 7, we further compare the temporal stability of MoRF avatars with StylePeople avatars using time-space slices. Here we created avatar images with a novel animation sequence and centered them on a single SMPL-X mesh vertex: on a T-shirt pattern, or the hemline. We demonstrate extreme failure cases for StylePeople that do not occur with MoRF. As seen, our method provides good vertical stability of rendered parts, confirming that the tasks of mesh misalignment absorption and translation for latent neural renderings into color images are disentangled.

Mesh fitting ablation study In Table 3, we perform an ablation study on mesh fitting modifications described in Section 4.1. We start with the SMPLify-X baseline [7] and add the improvements one-by-one. We retrain the MoRF avatars for the new fits, and perform a user study on pairs of videos: one is the baseline, the other is a modification. Every pair is shown to 25 people, one person judges at most 6 pairs every 6 hours. Although on average any modification is favored by the user study over SMPLify-X, on particular avatar subjects the score can significantly drop, e.g. as low as 12% for the temporal loss. However, when all modifications are used, the lowest preference score among all subjects was 52%, indicating the consistently positive impact of the proposed fitting improvements on the avatar quality.

<table>
<thead>
<tr>
<th>SMPLify-X</th>
<th>+Shared shape +Frames filter</th>
<th>+Temporal loss +Silhouette loss</th>
<th>+All</th>
</tr>
</thead>
<tbody>
<tr>
<td>N/A</td>
<td>57.6%</td>
<td>58.0%</td>
<td>64.4%</td>
</tr>
</tbody>
</table>

5.2. ZJU-MoCap

To evaluate on ZJU-MoCap dataset, we follow the protocol/split from HumanNeRF and compare to HumanNeRF and StylePeople with two types of SMPL mesh fits: type A) the original multi-view fits from ZJU-MoCap, and type B) our monocular fitting results. For both types of fits, HumanNeRF exhibits ghosting and NeRF blob artifacts. Despite MoCap origin of this dataset, it’s original SMPL multi-
<table>
<thead>
<tr>
<th>Subject 377</th>
<th>Subject 394</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSNR↑</td>
<td>SSIM↑</td>
</tr>
<tr>
<td>HumanNeRF</td>
<td>28.54</td>
</tr>
<tr>
<td>StylePeople</td>
<td>29.30</td>
</tr>
<tr>
<td>MoRF</td>
<td>29.03</td>
</tr>
</tbody>
</table>

Table 4. Evaluation on ZJU-MoCap dataset. (∗) LPIPS × 10³.

HumanNeRF StylePeople MoRF

Figure 8. Results on ZJU-MoCap dataset (renderings of novel views far from the train view). Avatars of subjects 377 (left), 394 (right) are shown, trained on multi-view (top) or on our monocular (bottom) mesh fits. Zoom-in is highly recommended.

view fits (type A) are less aligned with ground truth images than our monocular fits (type B). As a result, given type A fits, we find out that MoRF performs worse that StylePeople baseline (Tab. 4, Fig. 8 – zoom-in highly recommended) because neutral texture warping could not converge. On the contrary, given well aligned type B mesh fits, MoRF produces sharper output renderings and less flicker. This sets the application scope of our rendering method: the expected effect is achieved with our mesh fitting and capture scenario.

5.3. People Snapshot

We have also found that MoRF works well on People Snapshot data. See the comparison on two identities from People Snapshot with InstantAvatar and StylePeople in Fig. 9.

Figure 9. Results on PeopleSnapshot produced by InstantAvatar (left), StylePeople (middle) and MoRF (right). The least texture averaging is demonstrated by MoRF.

ticeable NeRF blob artifacts, and HumanNeRF has the least of them.

Compared to NeRF-based methods, mesh-based rendering methods (StylePeople, ANR and MoRF) are more stable and do not produce blob artifacts. Moreover the inference speed for the fastest NeRF (InstantAvatar) is 15 FPS for 512 × 512 px on desktop GPU, compared to 30 FPS for 640 × 640 px on mobile GPU for MoRF.

Compared to StylePeople and ANR, MoRF can produce sharper and more temporally stable results given our well aligned mesh fits. This effect is noticeable on texture rich regions like on a T-shift.

For the results of HF-Avatar see Figure 6. The NeuMan method struggles for static monocular videos like in our self-captured dataset, see supp. mat. for the exact details.

6. Discussion

We have presented a fullbody avatar system that has high realism, can be rendered in real-time on a mobile devices, and is learned from monocular videos. We evaluate our method on a self-captured dataset on novel poses, prove MoRF is faster and produces better rendering results compared to NeRFs. On ZJU-MoCap dataset, we find out that MoRF is sensitive to mesh fits quality. MoRF produces sharper and more temporally stable results (compared to StylePeople and ANR) given our well aligned mesh fits, the effect is noticeable on texture rich regions like on a T-shift.

The limitations of our method are: 1) MoRF does not model clothing movement or loose clothing, 2) there is no prior to render train-blind areas, 3) texture seams are not handled and warping artifacts could appear there, using spherical texture without seams is a future work.
References


