PointAvatar: Deformable Point-based Head Avatars from Videos

Yufeng Zheng[1,2] Wang Yifan[3] Gordon Wetzstein[3] Michael J. Black[2] Otmar Hilliges[1]
[1]ETH Zurich, [2]Max Planck Institute for Intelligent Systems, [3]Stanford University

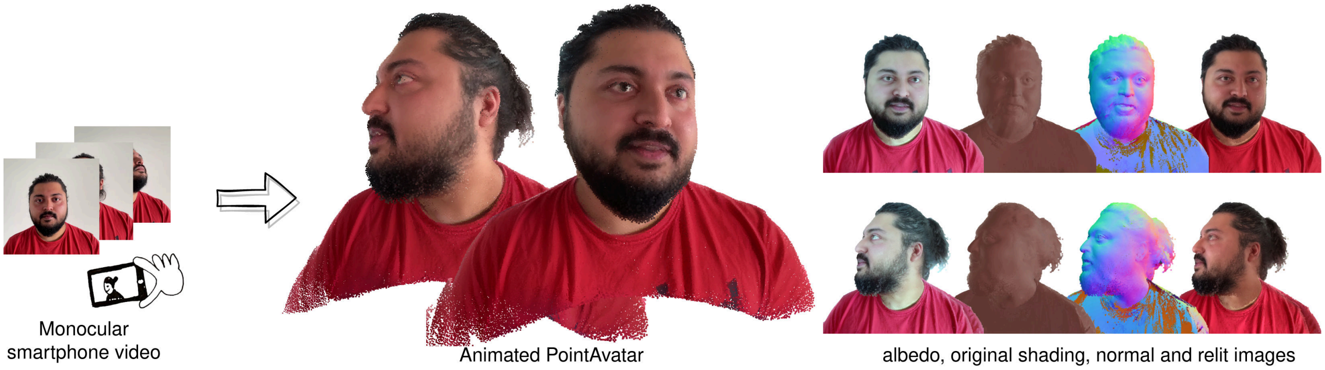


Figure 1. **PointAvatar** learns lighting-disentangled point-based head avatars from a monocular RGB video captured by a smartphone.

Abstract

The ability to create realistic animatable and relightable head avatars from casual video sequences would open up wide ranging applications in communication and entertainment. Current methods either build on explicit 3D morphable meshes (3DMM) or exploit neural implicit representations. The former are limited by fixed topology, while the latter are non-trivial to deform and inefficient to render. Furthermore, existing approaches entangle lighting and albedo, limiting the ability to re-render the avatar in new environments. In contrast, we propose PointAvatar, a deformable point-based representation that disentangles the source color into intrinsic albedo and normal-dependent shading. We demonstrate that PointAvatar bridges the gap between existing mesh- and implicit representations, combining high-quality geometry and appearance with topological flexibility, ease of deformation and rendering efficiency. We show that our method is able to generate animatable 3D avatars using monocular videos from multiple sources including hand-held smartphones, laptop webcams and internet videos, achieving state-of-the-art quality in challenging cases where previous methods fail, e.g.*, thin hair strands, while being significantly more efficient in training than competing methods.*

1. Introduction

Personalized 3D avatars will enable new forms of communication and entertainment. Successful tools for creating avatars should enable easy data capture, efficient computation, and create a photo-realistic, animatable, and relightable 3D representation of the user. Unfortunately, existing approaches fall short of meeting these requirements.

Recent methods that create 3D avatars from videos either build on 3D morphable models (3DMMs) [26, 36] or leverage neural implicit representations [32, 33, 35]. The former methods [8, 13, 22, 23] allow efficient rasterization and inherently generalize to unseen deformations, but they cannot easily model individuals with eyeglasses or complex hairstyles, as 3D meshes are limited by a-priori fixed topologies and surface-like geometries. Recently, neural implicit representations have also been used to model 3D heads [5, 11, 16, 54]. While they outperform 3DMM-based methods in capturing hair strands and eyeglasses, they are significantly less efficient to train and render, since rendering a single pixel requires querying many points along the camera ray. Moreover, deforming implicit representations in a generalizable manner is non-trivial and existing approaches have to revert to an inefficient root-finding loop, which impacts training and testing time negatively [10, 18, 25, 45, 54].

To address these issues, we propose PointAvatar, a novel avatar representation that uses point clouds to represent the canonical geometry and learns a continuous deformation

contact: yufeng.zheng@inf.ethz.ch
project page: https://zhengyuf.github.io/PointAvatar/

	Efficient Rendering	Easy Animation	Flexible Topology	Thin Strands	Surface Geometry
Meshes	✓	✓	✗	✗	✓
Implicit Surfaces	✗	✗	✓	✗	✓
Volumetric NeRF	✗	✗	✓	✓	✗
Points (ours)	✓	✓	✓	✓	✓

Table 1. **PointAvatar** is efficient to render and deform which enables straightforward rendering of full images during training. It can also handles flexible topologies and thin structures and can reconstruct good surface normals in surface-like regions, *e.g.*, skin.

field for animation. Specifically, we optimize an oriented point cloud to represent the geometry of a subject in a canonical space. For animation, the learned deformation field maps the canonical points to the deformed space with learned blendshapes and skinning weights, given expression and pose parameters of a pretrained 3DMM. Compared to implicit representations, our point-based representation can be rendered efficiently with a standard differentiable rasterizer. Moreover, they can be deformed effectively using established techniques, *e.g.*, skinning. Compared to meshes, points are considerably more flexible and versatile. Besides the ability to conform the topology to model accessories such as eyeglasses, they can also represent complex volume-like structures such as fluffy hair. We summarize the advantanges of our point-based representation in Tab. 1.

One strength of our method is the disentanglement of lighting effects. Given a monocular video captured in unconstrained lighting, we disentangle the apparent color into the intrinsic albedo and the normal-dependent shading; see Fig. 1. However, due to the discrete nature of points, accurately computing normals from point clouds is a challenging and costly task [6, 17, 29, 37], where the quality can deteriorate rapidly with noise, and insufficient or irregular sampling. Hence we propose two techniques to (a) robustly and accurately obtain normals from learned canonical points, and (b) consistently transform the canonical point normals with the non-rigid deformation. For the former, we exploit the low-frequency bias of MLPs [38] and estimate the normals by fitting a smooth signed distance function (SDF) to the points; for the latter, we leverage the continuity of the deformation mapping and transform the normals analytically using the deformation's Jacobian. The two techniques lead to high-quality normal estimation, which in turn propagates the rich geometric cues contained in shading to further improve the point geometry. With disentangled albedo and detailed normal directions, PointAvatar can be relit and rendered under novel scene lighting.

As demonstrated using various videos captured with DSLR, smartphone, laptop cameras, or obtained from the internet, the proposed representation combines the advantages of popular mesh and implicit representations, and surpasses both in many challenging scenarios. In summary, our contributions include:

1. We propose a novel representation for 3D animatable avatars based on an explicit canonical point cloud and continuous deformation, which shows state-of-the-art photo-realism while being considerably more efficient than existing implicit 3D avatar methods;
2. We disentangle the RGB color into a pose-agnostic albedo and a pose-dependent shading component;
3. We demonstrate the advantage of our methods on a variety of subjects captured through various commodity cameras, showing superior results in challenging cases, *e.g.*, for voluminous curly hair and novel poses with large deformation.

2. Related Work

Point clouds for neural rendering. Point-based neural rendering is gaining attention thanks to the flexibility and scalability of point representations [43]. Aliev *et al.* [1] introduce one of the first approaches to point-based neural rendering, in which each point is augmented with a neural feature. These features are projected to a set of multi-scale 2D feature maps using a standard point rasterizer, then converted to an RGB image. Recently, Point-NeRF [48] combines points with volumetric rendering, achieving perceptually high-quality results while being efficient to train. Both of the above methods obtain point geometry through off-the-shelf multi-view stereo methods and keep the points fixed during optimization. Other approaches [40, 52] propose to jointly optimize the point geometry using differentiable point rasterization [24, 47, 51]. Our method not only jointly optimizes both the point geometry and colors, but also the deformation of the point cloud through a forward deformation field proposed in [10, 54].

Head avatars from 2D. Learning photo-realistic head avatars from 2D observations is an emerging research topic in the computer vision community. Based on 3DMMs [26, 36], recent works [13, 23] leverage differentiable neural rendering to learn the detailed facial appearance and complete head geometry for a single subject. The idea is extended in [8, 22] to enable generative or one-shot head avatars. Another line of work leverages neural implicit representations. NerFace [11] and RigNeRF [2] extend photo-realistic and flexible neural radiance fields (NeRF) [33] to model the dynamic head geometry and view-dependent appearance. IMavatar [54] employs neural implicit surfaces [21, 32, 35, 50] and learns an implicit deformation field for generalizable animation. NeRF-based head avatars have been extended to a multi-subject scenario [5, 16] and combined with InstantNGP [34] for fast appearance acquisition [12, 49, 55]. To the best of our knowledge, our work is the first to learn deformable point-based head avatars.

Point-based body and clothing avatars. PoP [31] learns

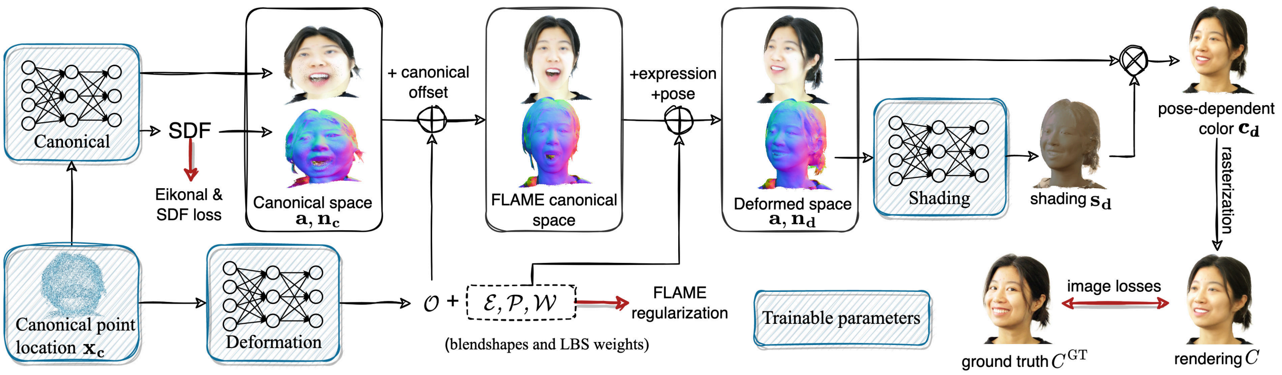

Canonical
SDF
Eikonal & SDF loss
Canonical space $\mathbf{a}, \mathbf{n_c}$
+ canonical offset
FLAME canonical space
+expression +pose
Deformed space $\mathbf{a}, \mathbf{n_d}$
Shading
shading $\mathbf{s_d}$
pose-dependent color $\mathbf{c_d}$
rasterization
Canonical point location $\mathbf{x_c}$
Deformation
$\mathcal{O}$ + $\mathcal{E}, \mathcal{P}, \mathcal{W}$
(blendshapes and LBS weights)
FLAME regularization
Trainable parameters
image losses
ground truth C^{GT}
rendering C

Figure 2. **Method pipeline.** We model the human head as a learned deformable point cloud consisting of the point locations $\mathbf{x_c}$, normals $\mathbf{n_c}$ and albedo $\mathbf{a}$, which describe the subject's geometry and intrinsic appearance in the canonical space. To deform points given a target expression and pose in a controllable and interpretable way, we warp $\mathbf{x_c}$ into the FLAME canonical space via a learned offset $\mathcal{O}$, and then deform further to the target deformed space by applying blendshapes and skinning using learned personalized expression basis $\mathcal{E}$, pose correctives $\mathcal{P}$ and skinning weights $\mathcal{W}$. After obtaining the deformed geometry, a small shading MLP is used to obtain shading $\mathbf{s_d}$ from the deformed normals $\mathbf{n_d}$. These are multiplied with the albedo colors $\mathbf{a}$ to produce the shaded colors $\mathbf{c_d}$, which are rendered to images via differentiable rasterization. PointAvatar leverages a combination of per-pixel and image-based losses to obtain photo-realism, while the FLAME regularization term encourages controllable and generalizable animations. GT stands for ground truth.

to model pose-dependent clothing geometry by mapping points from the minimally-clothed SMPL [28] surface to the clothing surface and demonstrates impressive geometric quality on various clothing types. Point-based clothing representation can be extended to mitigate point sparsity issues for long skirts [27, 30]. Our work is principally different in two ways: 1) We learn the deformation, geometry, and appearance jointly from scratch, without explicitly relying on a 3DMM template. 2) We learn from monocular videos, whereas previous work requires 3D scans [27, 30, 31].

3. Method

Given a monocular RGB video of a subject performing various expressions and poses, our model jointly learns (1) a point cloud representing the pose-agnostic geometry and appearance of the subject in a canonical space; (2) a deformation network that transforms the point cloud into new poses using FLAME [26] expression and pose parameters extracted from the RGB frames; (3) a shading network that outputs a per-point shading vector based on the point normals in the deformed space. The three components can be jointly optimized by comparing the rendering of the shaded points in the deformed space with the input frames. Once trained, we can synthesize new sequences of the same subject under novel poses, expressions and lighting conditions. Figure 2 presents an overview of our method.

3.1. Point-based Canonical Representation

Our canonical point representation consists of a set of learnable points $\mathcal{P}_c = \{\mathbf{x_c}^i\}$ with $i = \{1, 2, \dots, N\}$, where $\mathbf{x_c} \in \mathbb{R}^3$ denotes the optimizable point locations. We empirically choose to learn points in an unconstrained canonical space without enforcing them to correspond to a predefined pose, and found that this leads to better geometry quality (implementation explained in Sec. 3.2).

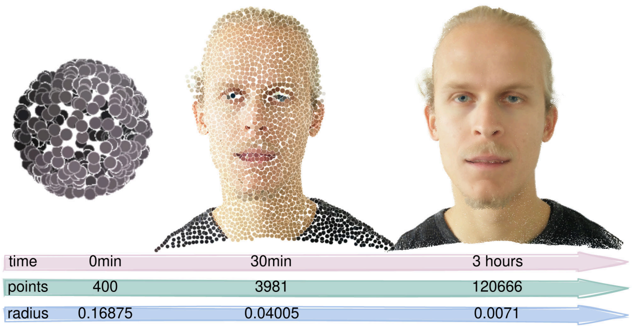


Figure 3. The **Coarse-to-fine** optimization strategy upsamples points and reduces point radii periodically during training, enabling fast convergence and detailed final reconstruction.

For optimization, we initialize with a sparse point cloud randomly sampled on a sphere and periodically upsample the point cloud while reducing the rendering radii. This coarse-to-fine optimization scheme enables fast convergence of training because the initial sparse points are efficient to deform and render, and can quickly approximate the coarse shape, whereas denser points in the late training stage lead to good reproduction of details. A schematic of this procedure is provided in Fig. 3. Furthermore, at the end of every epoch, we prune away invisible points that have not been projected onto any pixel with visibility above a predefined threshold; this further accelerates training.

Our method disentangles the point colors into a pose-agnostic albedo component and a pose-dependent shading component (explained later in Sec. 3.3), where the shading component is inferred from the point normals in the deformed space. We first discuss how to compute the normals and the albedo in canonical space, followed by the learned transformation of point locations and normals into the deformed space (see Sec. 3.2).

Canonical normals. Point normals are local geometric descriptors that can be estimated from neighboring point locations. However, estimating normals in this way yields poor results, especially when the points are sparse, noisy and irregularly sampled, as is the case during early training. To alleviate this issue, we estimate the point normals from an SDF defined by the canonical points. Normals are then defined as the spatial gradient of the SDF:

$$\mathbf{n_c} = \nabla_{\mathbf{x_c}} \mathrm{SDF}\left(\mathbf{x_c}\right). \tag{1}$$

Specifically, we represent the SDF with an MLP and fit the zero-level set to the canonical point locations using a data term and an Eikonal regularizer, defined as:

$$\mathcal{L}_{\text{sdf}} = \left\|\mathrm{SDF}\left(\mathbf{x_c}\right)\right\|^2 \text{ and } \mathcal{L}_{\text{eik}} = \left(\left\|\nabla_{\mathbf{x_e}} \mathrm{SDF}\left(\mathbf{x_e}\right)\right\| - 1\right)^2, \tag{2}$$

where the Eikonal samples $\mathbf{x_e}$ consist of the original and perturbed point locations [14]. We update the SDF using current point locations at each training step.

Canonical albedo. We use an MLP to map the point locations $\mathbf{x_c}$ to the albedo colors $\mathbf{a} \in \mathbb{R}^3$, similar to [9]. Compared with directly modeling the albedo of each point as an individual per-point feature, the inductive bias of MLPs automatically enforces a local smoothness prior on albedo colors [3, 4, 15, 42]. For efficiency, we use a shared MLP to compute the canonical normals and albedo in practice, *i.e.*,

$$\left[\mathrm{SDF}\left(\mathbf{x_c}\right); \mathbf{a}\right] = \mathrm{MLP_c}\left(\mathbf{x_c}\right). \tag{3}$$

3.2. Point Deformation

In order to achieve controllable animations, PointAvatar is deformed using the same expression and pose parameters as FLAME [26], a parametric head model learned from 4D scans. Our deformation takes a two-step approach shown in Fig. 2. In the first stage, we warp the canonical points $\mathbf{x_c}$ to their location in an intermediate pose $\mathbf{x_o}$, which corresponds to a predefined mouth-opened canonical pose of the FLAME template. In the second stage, we use the target FLAME expression and pose parameters to transform $\mathbf{x_o}$ to the deformed space based on learned blendshapes and skinning weights. Our experiments (see Sec. 4.4) empirically show that the proposed two-staged deformation helps to avoid bad local minima during optimization and yields more accurate geometries.

To improve expression fidelity and to account for accessories such as eyeglasses, we learn personalized deformation blendshapes and skinning weights, similar to IMavatar [54]. Specifically, a coordinate-based MLP is used to map each canonical point $\mathbf{x_c}$ to (1) an offset $\mathcal{O} \in \mathbb{R}^3$, which translates $\mathbf{x_c}$ to its corresponding location in the FLAME canonical space $\mathbf{x_o}$; (2) n_e expression blendshapes and n_p pose blendshapes, denoted as $\mathcal{E} \in \mathbb{R}^{n_e \times 3}$ and $\mathcal{P} \in \mathbb{R}^{n_p \times 9 \times 3}$; (3) LBS weights $\mathcal{W} \in \mathbb{R}^{n_j}$ associated with the n_j bones. The point location in deformed space $\mathbf{x_d}$ is then computed as:

$$\mathbf{x_o} = \mathbf{x_c} + \mathcal{O} \tag{4}$$

$$\mathbf{x_d} = \mathrm{LBS}\left(\mathbf{x_o} + \mathrm{B}_P\left(\theta; \mathcal{P}\right) + \mathrm{B}_E\left(\psi; \mathcal{E}\right), \mathrm{J}\left(\psi\right), \theta, \mathcal{W}\right), \tag{5}$$

where LBS and J define the standard skinning function and the joint regressor defined in FLAME, and B_P and B_E denote the linear combination of blendshapes, which outputs the additive pose and expression offsets from the animation coefficients θ and ψ and the blendshape bases $\mathcal{P}$ and $\mathcal{E}$. Despite the similarity of this formulation to IMavatar [54], our forward deformation mapping only needs to be applied once in order to map canonical point locations $\mathbf{x_c}$ to the deformed space, and therefore does not need the computationally costly correspondence search of IMavatar [54].

Normal deformation. The deformation mapping defined in Eq. (5) is differentiable w.r.t. the input point locations. This allows us to transform canonical normals analytically with the inverse of the deformation Jacobian

$$\mathbf{n_d} = l\mathbf{n_c}\left(\frac{\mathrm{d}\mathbf{x_d}}{\mathrm{d}\mathbf{x_c}}\right)^{-1}, \tag{6}$$

where l is a normalizing scalar to ensure the normals are of unit length. The formulation can be obtained via Taylor's theorem. Please see Supp. Mat. for the detailed proof.

3.3. Point Color

The color of each point in the deformed space $\mathbf{c_d}$ is obtained by multiplying the intrinsic albedo $\mathbf{a} \in \mathbb{R}^3$ with the pose-dependent lighting effects, which we refer to as shading $\mathbf{s_d} \in \mathbb{R}^3$:

$$\mathbf{c_d} = \mathbf{a} \circ \mathbf{s_d}, \tag{7}$$

where $\circ$ denotes the Hadamard product. As we assume our input video is captured under a fixed lighting condition and camera position, we model the shading as a function of the deformed point normals $\mathbf{n_d}$. In practice, we approximate their relation using a shallow MLP:

$$\mathbf{s_d} = \mathrm{MLP}_s\left(\mathbf{n_d}\right). \tag{8}$$

By conditioning the albedo only on the canonical locations (see Sec. 3.1) and the shading only on the normal directions, our formulation achieves *unsupervised* albedo disentanglement. Even though our lighting model is simple and minimally constrained, our method can enable rudimentary relighting by changing the shading component (see Sec. 4.2).

3.4. Differentiable Point Rendering

One advantage of our point-based representation is that it can be rendered efficiently using rasterization. We adopt PyTorch3D's [39] differentiable point renderer. It splats each point as a 2D circle with a uniform radius, arranges them in z-buffers, and finally composites the splats using alpha compositing. The alpha values are calculated as $\alpha = 1 - d^2/r^2$, where d is the distance from the point center to the pixel center and r represents the splatting radius. Then, the transmittance values are calculated as $T_i = \prod_{k=1}^{i-1}(1-\alpha_k)$, where i denotes the index of the sorted points in the z-buffer. The point color values are integrated to produces the final pixel color:

$$c_{\text{pix}} = \sum_i \alpha_i T_i c_{\text{d},i}. \tag{9}$$

3.5. Training Objectives

The efficiency of point-based rendering allows us to render the whole image at each training step. This makes it possible to apply perceptual losses on the whole image, whereas implicit-based avatar methods are often limited to per-pixel objectives. Specifically, we adopt a VGG feature loss [19], defined as

$$\mathcal{L}_{\text{vgg}}\left(\mathbf{C}\right) = \left\|\mathrm{F}_{\text{vgg}}(\mathbf{C}) - \mathrm{F}_{\text{vgg}}(\mathbf{C}^{\text{GT}})\right\|, \tag{10}$$

where $\mathbf{C}$ and $\mathbf{C}^{\text{GT}}$ denote the predicted and ground truth image, and $\mathrm{F}_{\text{vgg}}(\cdot)$ calculates the features from the first four layers of a pre-trained VGG [41] network.

In addition to $\mathcal{L}_{\text{vgg}}$, we follow prior work and adopt the losses of IMavatar [54]:

$$\mathcal{L}_{\text{RGB}} = \left\|\mathbf{C} - \mathbf{C}^{\text{GT}}\right\|,$$
$$\mathcal{L}_{\text{flame}} = \lambda_e\|\mathcal{E} - \widehat{\mathcal{E}}\|_2 + \lambda_p\|\mathcal{P} - \widehat{\mathcal{P}}\|_2 + \lambda_w\|\mathcal{W} - \widehat{\mathcal{W}}\|_2,$$
$$\mathcal{L}_{\text{mask}} = \left\|\mathbf{M} - \mathbf{M}^{\text{GT}}\right\|.$$

Here, $\mathbf{M}$ and $\mathbf{M}^{\text{GT}}$ denote the predicted and ground truth head mask, $\widehat{\mathcal{E}}, \widehat{\mathcal{P}}$ and $\widehat{\mathcal{W}}$ are pseudo ground truth defined by the nearest FLAME vertex. The mask prediction at each pixel is obtained by $m_{\text{pix}} = \sum_i \alpha_i T_i$, and the pseudo ground truth mask is obtained using an off-the-shelf foreground estimator [20].

Our total loss is

$$\mathcal{L} = \lambda_{\text{rgb}}\mathcal{L}_{\text{RGB}} + \lambda_{\text{mask}}\mathcal{L}_{\text{mask}} + \lambda_{\text{flame}}\mathcal{L}_{\text{flame}} + \lambda_{\text{vgg}}\mathcal{L}_{\text{vgg}}. \tag{11}$$

	L1 ↓	LPIPS ↓	SSIM ↑	PSNR ↑
IMavatar [54]	0.033; 0.050	0.178; 0.261	0.874; 0.770	22.4; 18.7
NerFace [11]	0.030; 0.045	0.126; 0.187	0.877; 0.782	22.7; 19.6
Ours	**0.021**; **0.036**	**0.094**; **0.145**	**0.899**; **0.802**	**26.6**; **22.3**
NHA [13]	0.022;0.029	0.086; 0.123	0.890; 0.837	25.7; 21.6
Ours (no cloth)	**0.017**; **0.021**	**0.077**; **0.100**	**0.912**; **0.863**	**28.6**; **25.8**

Table 2. **Quantitative comparison.** The first and second number in each cell represent scores for lab-capture sequences (IMavatar and NerFace datasets) and casual videos (smartphone, webcam and internet videos), respectively. PointAvatar outperforms implicit-based IMavatar and NerFace by a large margin in perceptual quality reflected by LPIPS [53]. Compared to 3DMM-based NHA, our method not only generates complete avatars with shoulders and clothing, but also performs better in the head region.

The loss weights can be found in Supp. Mat. together with other implementation details.

4. Experiments

Datasets. We compare our approach with state-of-the-art (SOTA) methods on 2 subjects from IMavatar [54] and 2 subjects from NerFace [11]. Additionally, we evaluate different methods on 1 subject collected from the internet, 4 subjects captured with hand-held smartphones, and 1 subject from a laptop webcam. These settings pose new challenges to avatar methods due to limited head pose variation, automatic exposure adjustment or low image resolution. In Supp. Mat., we evaluate reconstructed point geometry on a synthetic dataset rendered from the MakeHuman project [7]. For all subjects, we use the same face-tracking results for all methods.

Baselines. We compare our method with three SOTA baselines, including (1) NerFace [11], which leverages dynamic neural radiance fields [33], (2) neural head avatar (NHA) [13] based on 3D morphable mesh models [26] (3) and IMavatar [54], which builds on neural implicit surfaces [50] and learnable blendshapes. Together with our method, which represents the head geometry with deformable point clouds, our experiments reveal the strength and weaknesses of each geometric representation in the scenario of head avatar reconstruction. Through our experiments, we demonstrate the efficiency, flexibility and photorealism achieved by point-based avatars.

4.1. Comparison with SOTA Methods

In Tab. 2, we quantitatively compare our point-based avatar method with SOTA baselines using conventional metrics including L1, LPIPS [53], SSIM [46] and PSNR. Since NHA [13] only models the head region, we compare with NHA without considering the clothing region. PointAvatar achieves the best accuracy among all methods on both lab-captured DSLR sequences and videos from more casual capture settings, *e.g.*, with smartphones.

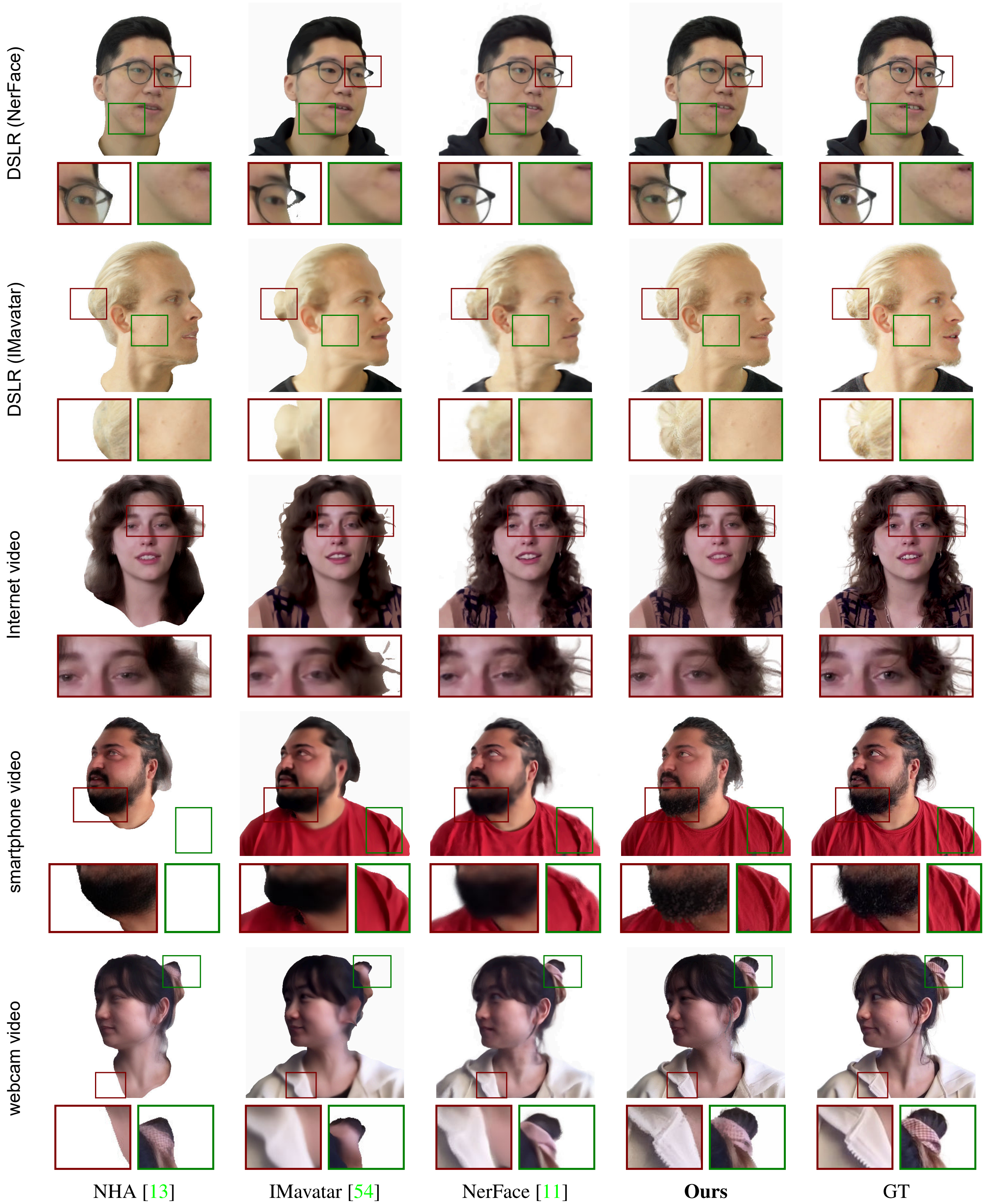


Figure 4. **Qualitative comparison.** PointAvatar produces photo-realistic and detailed appearance compared to SOTA methods, especially apparent in skin details and hair textures. Our point-based method is also flexible enough to capture challenging geometries such as eyeglasses and thin hair strands, which cannot be handled by mesh-based methods.

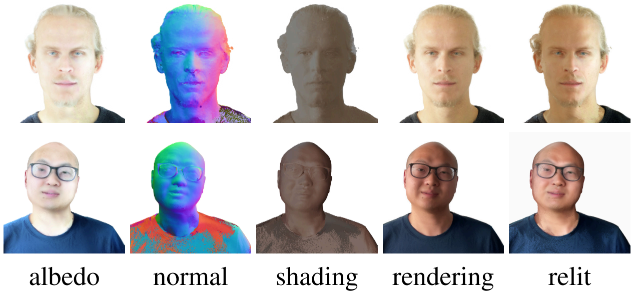


Figure 5. **Self-supervised lighting disentanglement.** PointAvatar disentangles albedo and normal-depedent shading from a single video captured with a fixed lighting condition. After training, PointAvatar can be relit.

Eyeglasses and detailed structures. In Fig. 4, we show qualitatively that our method can successfully handle the non-face topology introduced by eyeglasses, which pose severe challenges for 3DMM-based methods. NHA cannot model the empty space between the frame of the eyeglasses because of its fixed topology and, instead, learns a convex hull painted with dynamic textures. In the second example of a man with a hair bun, even though the bun is topologically consistent with the template mesh, NHA still produces a worse hair silhouette than other methods. IMavatar, based on implicit surfaces, is theoretically capable of modeling the geometry of eyeglasses, but it also fails to learn part of the thin frame of glasses. Furthermore, for IMavatar to learn such thin structures, accurate foreground segmentation masks are required, which are hard to obtain in-the-wild. In Supp. Mat., we show that our method can be trained without mask supervision or known backgrounds.

Surface-like skin and volumetric hair. Both NHA and IMavatar enforce a surface constraint by design. While such a constraint helps with surface geometry reconstruction and multi-view consistency, it also causes problems when modeling volumetric structures such as the curly hair in the third example (Fig. 4). Our point-based method is free to model volumetric structures where needed, but also encourages surface-like point geometry in the skin and clothing regions via point pruning, which removes unseen inside points. We show that our method renders sharp and realistic images even for extreme head poses. In contrast, NerFace is a flexible volumetric NeRF-based avatar method rendered in a volumetric manner. While it is capable of modeling thin structures of any topology, its under-constrained nature leads to poor geometry and compromises rendering quality in uncommon poses. Even for near frontal poses, shown in the first and third examples, our method still produces sharper skin details than NerFace.

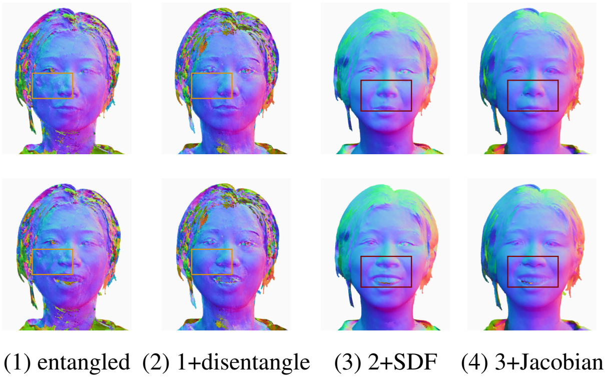


Figure 6. **Geometry ablation.** Disentangling shading and albedo improves facial normals compared to an entangled rendering model. Moreover, obtaining canonical point normals from the SDF further improved smoothness. Transforming normals with the spatial Jacobian of the deformation field enables the capture of blendshape-related normal changes, *e.g*., around the nasal line.

4.2. Lighting Disentanglement

Our method disentangles rendered colors into intrinsic albedo colors and normal-dependent shading values. By changing the shading component, we can alter the lighting directions as shown in Fig. 5 (see Supp. Mat. for details of relighting). In the following, we demonstrate that lighting disentanglement, along with the proposed normal estimation techniques, improve the geometric details of the reconstruction. In column 2 of Fig. 6, we show that, disentangling shading and albedo itself improves facial geometry (see the cheek area). We further ablate two of our design choices for obtaining better point normals. First, we compare our SDF-based normal estimation with direct normal estimation, which calculates normals by approximating the local plane of neighboring points. Normals obtained through the SDF are less noisy, especially in highly detailed regions such as the hair, eyes and mouth, where direct normal estimation often fails. Second, we compare our proposed normal transformation using the deformation Jacobian vs. transforming the normals simply with the rotation matrix of the point deformation. The latter ignores the spatial changes of blendshapes and LBS weights. As column 4 shows, our normal transformation method takes into account the effects of additive blendshapes, and is able to produce correct normals around the nasal line when smiling.

4.3. Training Efficiency

In Tab. 3 and Fig. 7, we show that PointAvatar is considerably faster to train and render than implicit head avatar methods. Thanks to the coarse-to-fine learning strategy and point pruning, our method only needs to render a small number of points with large radii in early training stages, which significantly speeds up training. With the ability to

Method	Training time (hour)	Rendering time per image (train)
IMavatar	48h	100s
NerFace	54h	4s
Ours	6h	0.1s - 1.5s (varies with point numbers)

Table 3. **Training and rendering time.** Compared to implicit-based methods, our method is significantly faster in training and is able to render full-images much more efficiently.

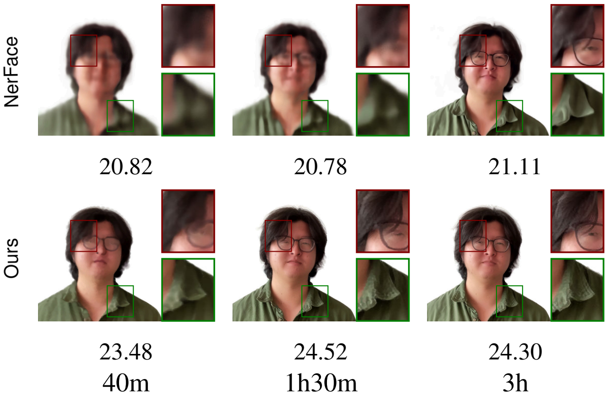
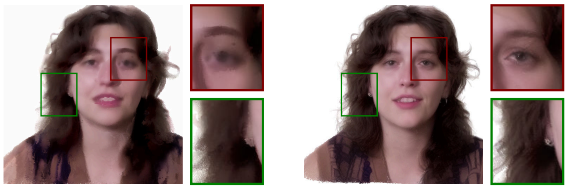


Figure 7. **Training efficiency.** PointAvatar converges much faster than implicit-based methods. Here we show NerFace [11] as an example, and indicate the average PSNR on the test set.



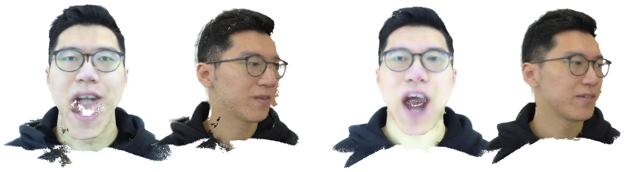
wo/ VGG with VGG

Figure 8. **Ablation: The VGG loss** improves photo-realism compared to only using a per-pixel L1 loss. PointAvatar is able to render full images efficiently during training, enabling the usage of various image- and patch-based losses.

render full images efficiently during training, PointAvatar can be trivially combined with various image- and patch-based training objectives, which target certain properties, *e.g.*, photo-realism, much more effectively than pixel-based losses. In Fig. 8, we ablate our VGG feature loss, revealing its significant effect in boosting photo-realism.

4.4. Ablation: Free Model Canonical Space

As depicted in Fig. 9, having an additional freely-learned canonical space, compared to forcing the model to directly learn in a predefined FLAME space, improves the canonical point geometry and generalization to novel poses significantly. A possible reason is that the deformations, modeled by an MLP, are easier to optimize compared to point locations. Without the canonical offset $\mathcal{O}$, the model overfits and learns wrong canonical geometries.



without canonical offset with canonical offset

Figure 9. **Ablation: The canonical offset** improves the canonical 3D geometry and therefore boosts generalization to novel head poses. For both cases, we show the canonical point representation (transformed to the FLAME canonical space if using canonical offset) and a deformed representation in a novel pose.

5. Discussion

We propose PointAvatar, a deformable point-based avatar representation that features high flexibility, efficient rendering and straightforward deformation. We show that PointAvatar is able to handle various challenging cases in modeling head avatars, including eyeglasses, voluminous hair, skin details and extreme head poses. Despite training only on a monocular video with fixed lighting conditions, PointAvatar achieves detailed facial and hair geometry and disentangles lighting effects from the intrinsic albedo.

There are several exciting future directions. (1) Our shading MLP maps normal directions to pose-dependent lighting effects, without further disentangling them into environment maps and surface reflectance properties, limiting relighting capability. Future work could leverage a more constrained, physically-based, rendering model. (2) We render points with uniform radius but some regions require denser and finer points to be modeled accurately, *e.g.*, eyes and hair strands. Rendering with different point sizes could potentially achieve detailed reconstructions with fewer points, which speeds up training and rendering further. (3) Future work could combine PointAvatar with neural rendering, potentially boosting photo-realism. (4) Our method cannot faithfully model the reflection of eyeglass lenses, which can be improved by modeling transparency and reflection [44]. (5) Point-based representations can suffer from sparsity issues under large deformations. We discuss several post-processing methods in Supp. Mat..

Acknowledgments an Conflicts: Yufeng Zheng is supported by the Max Planck ETH Center for Learning Systems. Wang Yifan is partially funded by the SNF postdoc mobility fellowship. MJB has received research gift funds from Adobe, Intel, Nvidia, Meta/Facebook, and Amazon. MJB has financial interests in Amazon, Datagen Technologies, and Meshcapade GmbH. While MJB is a part-time employee of Meshcapade, his research was performed solely at, and funded solely by, the Max Planck Society.

References

[1] Kara-Ali Aliev, Artem Sevastopolsky, Maria Kolos, Dmitry Ulyanov, and Victor Lempitsky. Neural point-based graphics. In *European Conference on Computer Vision (ECCV)*, pages 696–712. Springer, 2020. 2

[2] ShahRukh Athar, Zexiang Xu, Kalyan Sunkavalli, Eli Shechtman, and Zhixin Shu. Rignerf: Fully controllable neural 3d portraits. In *Computer Vision and Pattern Recognition (CVPR)*, 2022. 2

[3] Jonathan T Barron and Jitendra Malik. Shape, albedo, and illumination from a single image of an unknown object. In *Computer Vision and Pattern Recognition (CVPR)*, pages 334–341. IEEE, 2012. 4

[4] Jonathan T Barron and Jitendra Malik. High-frequency shape and albedo from shading using natural image statistics. In *Computer Vision and Pattern Recognition (CVPR)*, 2022. 4

[5] Alexander W. Bergman, Petr Kellnhofer, Wang Yifan, Eric R. Chan, David B. Lindell, and Gordon Wetzstein. Generative neural articulated radiance fields. In *NeurIPS*, 2022. 1, 2

[6] Alexandre Boulch and Renaud Marlet. Fast and robust normal estimation for point clouds with sharp features. In *Computer graphics forum*. Wiley Online Library, 2012. 2

[7] Leyde Briceno and Gunther Paul. Makehuman: a review of the modelling framework. In *Proceedings of the 20th Congress of the International Ergonomics Association (IEA 2018) Volume V: Human Simulation and Virtual Environments, Work With Computing Systems (WWCS), Process Control 20*, pages 224–232. Springer, 2019. 5

[8] Marcel C. Buehler, Abhimitra Meka, Gengyan Li, Thabo Beeler, and Otmar Hilliges. Varitex: Variational neural face textures. In *International Conference on Computer Vision (ICCV)*, 2021. 1, 2

[9] Pol Caselles, Eduard Ramon, Jaime Garcia, Xavier Giro-i Nieto, Francesc Moreno-Noguer, and Gil Triginer. Sira: Relightable avatars from a single image. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 775–784, 2023. 4

[10] Xu Chen, Yufeng Zheng, Michael J Black, Otmar Hilliges, and Andreas Geiger. Snarf: Differentiable forward skinning for animating non-rigid neural implicit shapes. In *International Conference on Computer Vision (ICCV)*, 2021. 1, 2

[11] Guy Gafni, Justus Thies, Michael Zollhöfer, and Matthias Nießner. Dynamic neural radiance fields for monocular 4d facial avatar reconstruction. In *Computer Vision and Pattern Recognition (CVPR)*, pages 8649–8658, June 2021. 1, 2, 5, 6, 8

[12] Xuan Gao, Chenglai Zhong, Jun Xiang, Yang Hong, Yudong Guo, and Juyong Zhang. Reconstructing personalized semantic facial nerf models from monocular video. *ACM Transactions on Graphics, (Proc. SIGGRAPH Asia)*, 41(6), 2022. 2

[13] Philip-William Grassal, Malte Prinzler, Titus Leistner, Carsten Rother, Matthias Nießner, and Justus Thies. Neural head avatars from monocular rgb videos. In *Computer Vision and Pattern Recognition (CVPR)*, 2022. 1, 2, 5, 6

[14] Amos Gropp, Lior Yariv, Niv Haim, Matan Atzmon, and Yaron Lipman. Implicit geometric regularization for learning shapes. In *Computer Vision and Pattern Recognition (CVPR)*, 2020. 4

[15] Roger Grosse, Micah K Johnson, Edward H Adelson, and William T Freeman. Ground truth dataset and baseline evaluations for intrinsic image algorithms. In *International Conference on Computer Vision (ICCV)*, pages 2335–2342. IEEE, 2009. 4

[16] Yang Hong, Bo Peng, Haiyao Xiao, Ligang Liu, and Juyong Zhang. Headnerf: A real-time nerf-based parametric head model. In *Computer Vision and Pattern Recognition (CVPR)*, 2022. 1, 2

[17] Hui Huang, Dan Li, Hao Zhang, Uri Ascher, and Daniel Cohen-Or. Consolidation of unorganized point clouds for surface reconstruction. *ACM Transactions on Graphics, (Proc. SIGGRAPH Asia)*, 28:176:1–176:7, 2009. 2

[18] Boyi Jiang, Yang Hong, Hujun Bao, and Juyong Zhang. Selfrecon: Self reconstruction your digital avatar from monocular video. In *Computer Vision and Pattern Recognition (CVPR)*, 2022. 1

[19] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European Conference on Computer Vision (ECCV)*, 2016. 5

[20] Zhanghan Ke, Jiayu Sun, Kaican Li, Qiong Yan, and Rynson W.H. Lau. Modnet: Real-time trimap-free portrait matting via objective decomposition. In *AAAI*, 2022. 5

[21] Petr Kellnhofer, Lars Jebe, Andrew Jones, Ryan Spicer, Kari Pulli, and Gordon Wetzstein. Neural lumigraph rendering. In *Computer Vision and Pattern Recognition (CVPR)*, 2021. 2

[22] Taras Khakhulin, Vanessa Sklyarova, Victor Lempitsky, and Egor Zakharov. Realistic one-shot mesh-based head avatars. In *European Conference of Computer Vision (ECCV)*, 2022. 1, 2

[23] Hyeongwoo Kim, Pablo Garrido, Ayush Tewari, Weipeng Xu, Justus Thies, Matthias Nießner, Patrick Pérez, Christian Richardt, Michael Zollöfer, and Christian Theobalt. Deep video portraits. *ACM Transactions on Graphics (TOG)*, 37(4):163, 2018. 1, 2

[24] Christoph Lassner and Michael Zollhofer. Pulsar: Efficient sphere-based neural rendering. In *Computer Vision and Pattern Recognition (CVPR)*, pages 1440–1449, 2021. 2

[25] Ruilong Li, Julian Tanke, Minh Vo, Michael Zollhofer, Jurgen Gall, Angjoo Kanazawa, and Christoph Lassner. Tava: Template-free animatable volumetric actors. In *European Conference on Computer Vision (ECCV)*, 2022. 1

[26] Tianye Li, Timo Bolkart, Michael. J. Black, Hao Li, and Javier Romero. Learning a model of facial shape and expression from 4D scans. *ACM Transactions on Graphics, (Proc. SIGGRAPH Asia)*, 36(6):194:1–194:17, 2017. 1, 2, 3, 4, 5

[27] Siyou Lin, Hongwen Zhang, Zerong Zheng, Ruizhi Shao, and Yebin Liu. Learning implicit templates for point-based clothed human modeling. In *European Conference of Computer vision (ECCV)*, 2022. 3

[28] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. SMPL: A skinned multi-person linear model. *ACM Transactions on Graphics, (Proc. SIGGRAPH Asia)*, 34(6):248:1–248:16, Oct. 2015. 3

[29] Dening Lu, Xuequan Lu, Yangxing Sun, and Jun Wang. Deep feature-preserving normal estimation for point cloud filtering. *Computer-Aided Design*, 125:102860, 2020. 2

[30] Qianli Ma, Jinlong Yang, Michael J. Black, and Siyu Tang. Neural point-based shape modeling of humans in challenging clothing. In *International Conference on 3D Vision (3DV)*, 2022. 3

[31] Qianli Ma, Jinlong Yang, Siyu Tang, and Michael J. Black. The power of points for modeling humans in clothing. In *International Conference on Computer Vision (ICCV)*, Oct. 2021. 2, 3

[32] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *Computer Vision and Pattern Recognition (CVPR)*, pages 4460–4470, 2019. 1, 2

[33] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *European Conference of Computer vision (ECCV)*, 2020. 1, 2, 5

[34] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Trans. Graph.*, 41(4):102:1–102:15, July 2022. 2

[35] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *Computer Vision and Pattern Recognition (CVPR)*, pages 165–174, 2019. 1, 2

[36] Pascal Paysan, Reinhard Knothe, Brian Amberg, Sami Romdhani, and Thomas Vetter. A 3d face model for pose and illumination invariant face recognition. In *2009 sixth IEEE international conference on advanced video and signal based surveillance*, pages 296–301. IEEE, 2009. 1, 2

[37] Xiaojuan Qi, Renjie Liao, Zhengzhe Liu, Raquel Urtasun, and Jiaya Jia. Geonet: Geometric neural network for joint depth and surface normal estimation. In *Computer Vision and Pattern Recognition (CVPR)*, pages 283–291, 2018. 2

[38] Nasim Rahaman, Aristide Baratin, Devansh Arpit, Felix Draxler, Min Lin, Fred Hamprecht, Yoshua Bengio, and Aaron Courville. On the spectral bias of neural networks. In *International Conference on Machine Learning*, pages 5301–5310. PMLR, 2019. 2

[39] Nikhila Ravi, Jeremy Reizenstein, David Novotny, Taylor Gordon, Wan-Yen Lo, Justin Johnson, and Georgia Gkioxari. Accelerating 3d deep learning with pytorch3d. *arXiv:2007.08501*, 2020. 5

[40] Darius Rückert, Linus Franke, and Marc Stamminger. Adop: Approximate differentiable one-pixel point rendering. *ACM Transactions on Graphics (TOG)*, 41(4):1–14, 2022. 2

[41] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 5

[42] Marshall Tappen, William Freeman, and Edward Adelson. Recovering intrinsic images from a single image. *Advances in Neural Information Processing Systems*, 15, 2002. 4

[43] A. Tewari, J. Thies, B. Mildenhall, P. Srinivasan, E. Tretschk, W. Yifan, C. Lassner, V. Sitzmann, R. Martin-Brualla, S. Lombardi, T. Simon, C. Theobalt, M. Nießner, J. T. Barron, G. Wetzstein, M. Zollhöfer, and V. Golyanik. Advances in neural rendering. *Computer Graphics Forum*, 41(2):703–735, 2022. 2

[44] Dor Verbin, Peter Hedman, Ben Mildenhall, Todd Zickler, Jonathan T. Barron, and Pratul P. Srinivasan. Ref-NeRF: Structured view-dependent appearance for neural radiance fields. *Computer Vision and Pattern Recognition (CVPR)*, 2022. 8

[45] Shaofei Wang, Katja Schwarz, Andreas Geiger, and Siyu Tang. Arah: Animatable volume rendering of articulated human sdfs. In *European Conference on Computer Vision (ECCV)*, 2022. 1

[46] Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004. 5

[47] Olivia Wiles, Georgia Gkioxari, Richard Szeliski, and Justin Johnson. Synsin: End-to-end view synthesis from a single image. In *Computer Vision and Pattern Recognition (CVPR)*, pages 7467–7477, 2020. 2

[48] Qiangeng Xu, Zexiang Xu, Julien Philip, Sai Bi, Zhixin Shu, Kalyan Sunkavalli, and Ulrich Neumann. Point-nerf: Point-based neural radiance fields. In *Computer Vision and Pattern Recognition (CVPR)*, 2022. 2

[49] Yuelang Xu, Lizhen Wang, Xiaochen Zhao, Hongwen Zhang, and Yebin Liu. Manvatar : Fast 3d head avatar reconstruction using motion-aware neural voxels, 2022. 2

[50] Lior Yariv, Yoni Kasten, Dror Moran, Meirav Galun, Matan Atzmon, Basri Ronen, and Yaron Lipman. Multiview neural surface reconstruction by disentangling geometry and appearance. *Advances in Neural Information Processing Systems*, 33, 2020. 2, 5

[51] Wang Yifan, Felice Serena, Shihao Wu, Cengiz Öztireli, and Olga Sorkine-Hornung. Differentiable surface splatting for point-based geometry processing. *ACM Transactions on Graphics (TOG)*, 38(6):1–14, 2019. 2

[52] Qiang Zhang, Seung-Hwan Baek, Szymon Rusinkiewicz, and Felix Heide. Differentiable point-based radiance fields for efficient view synthesis. In *ACM Transactions on Graphics, (Proc. SIGGRAPH Asia)*, SA '22, New York, NY, USA, 2022. Association for Computing Machinery. 2

[53] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Computer Vision and Pattern Recognition (CVPR)*, pages 586–595, 2018. 5

[54] Yufeng Zheng, Victoria Fernández Abrevaya, Marcel C. Bühler, Xu Chen, Michael J. Black, and Otmar Hilliges. I M Avatar: Implicit morphable head avatars from videos. In *Computer Vision and Pattern Recognition (CVPR)*, 2022. 1, 2, 4, 5, 6

[55] Wojciech Zielonka, Timo Bolkart, and Justus Thies. Instant volumetric head avatars. In *Computer Vision and Pattern Recognition (CVPR)*, June 2023. 2