CAT-NeRF: Constancy-Aware Tx^{2}Former for Dynamic Body Modeling

Haidong Zhu Zhaoheng Zheng Wanrong Zheng Ram Nevatia
University of Southern California
{haidongz|zhaoheng.zheng|wanrongz|nevatia@usc.edu}

Abstract

This paper addresses the problem of human rendering in the video with temporal appearance constancy. Reconstructing dynamic body shapes with volumetric neural rendering methods, such as NeRF, requires finding the correspondence of the points in the canonical and observation space, which demands understanding human body shape and motion. Some methods use rigid transformation, such as SE(3), which cannot precisely model each frame's unique motion and muscle movements. Others generate the transformation for each frame with a trainable network, such as neural blend weight field or translation vector field, which does not consider the appearance constancy of general body shape. In this paper, we propose CAT-NeRF for self-awareness of appearance constancy with Tx^{2}Former, a novel way to combine two Transformer layers, to separate appearance constancy and uniqueness. Appearance constancy models the general shape across the video, and uniqueness models the unique patterns for each frame. We further introduce a novel Covariance Loss to limit the correlation between each pair of appearance uniquenesses to ensure the frame-unique pattern is maximally captured in appearance uniqueness. We assess our method on H36M and ZJU-MoCap and show state-of-the-art performance.

1. Introduction

Rendering an animated person in a video from a novel viewpoint is helpful for several applications, such as game design and simulation, and involves implicit inference of the 3-D human shape and pose. High-quality human reconstructions require modeling the detailed appearance of each frame, which are expensive in computation and ignore the constant appearance of the same person in the sequence. In addition, when encountering occlusions, framewise reconstruction cannot fill in these occluded patterns with knowledge from current viewpoints, which we can find in other frames as the constant appearance of the same person.

In this work, we focus on mining the appearance constancy among the frames for rendering dynamic body shapes based on the neural radiance field (NeRF) [24]. NeRF implicitly records the density and color of the object, making viewpoint-free rendering possible without requiring explicit geometric modeling. NeRF constructs two spaces for the animation of an object: an observation space for each frame reflecting the observed object shape, and a canonical space for a specific pose shared by all frames. The alignment between these spaces requires understanding both shapes and dynamic motions.

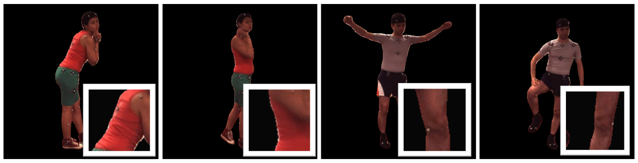

(a) Clothes Change (b) Muscle Movements

Figure 1. For the dynamic body in the video sequence, we have some unique framewise patterns from (a) change of the patterns on the clothes and (b) change of the muscles in addition to the constant appearances and shapes that are shareable across the frames.

To align the points between these two spaces, researchers use SE(3) [29] or a translation vector field [32,35] for building correspondences between the canonical and observation spaces. The translation vector field predicts the correspondence with a trainable neural network, while SE(3) uses rigid transformation for rotation matrix computation between body parts. Considering the non-rigid transformation of the human body shapes, finding such correspondence between these two spaces requires both understanding of rigid transformation for global shapes and non-rigid local movements and motions. We show two examples in Figure 1. Based on constant shapes and appearance, dynamic body shapes introduce unique framewise appearance from changing patterns on (a) the clothes and (b) muscle that cannot be captured with rigid transformation.

To solve this problem, we separate appearance constancy and uniqueness between the frames based on the neural blend weight fields [32] with **C**onstancy **A**wareness **T**x^{2}Former, abbreviated as CAT-NeRF. We apply a temporal-constant feature to model the constant appearance shared across all frames and a set of framewise features for each frame to capture the uniqueness. Appearance constancy can help find the missing pattern with the knowledge from other frames when encountering the unseen parts, while appearance uniqueness is to include more frame-specific motions and patterns.

Since the temporal-constant and framewise features capture patterns of different levels, the model needs to distinguish the appearance constancy and uniqueness in the feature sequence. We introduce a novel Covariance Loss to minimize the correlation score in the framewise feature. By limiting the similar information shared across the framewise features, we make these features focus on the frame it is representing and maximally extract the unique patterns in each frame. This can also simultaneously maximize the appearance constancy captured by the temporal-constant feature since the model needs to store these patterns for modeling the dynamic body shapes during optimization. We include more discussion in Sec. 3.3.

In addition, considering some appearances are only shared by a small set of frames, directly mining the appearance constancy or uniqueness with Covariance Loss fails to capture these patterns in either level of features. We introduce Transformer-on-Transformer (Tx^{2}Former), a new way of combining the Transformer layers to fuse the framewise features and focus on useful information based on the current frame. The first Transformer [46] layer equally combines all the framewise features, and the second Transformer layer takes the feature of the current frame along with the average-pooled output of the first Transformer to select the helpful information for the specific frame. Unlike the typical Transformer that only takes the output of previous layers as input, introducing the feature of the current frame helps the network focus on what needs to be selected from sequential features via self-attention. We assess our method on two public datasets, ZJU-MoCap [33] and H36M [13], and show state-of-the-art performance.

In summary, our contributions are as follows: 1) we introduce separating constant and unique appearance for dynamic human body rendering, 2) we introduce CAT-NeRF with a Tx^{2}Former for mining the appearance constancy across the frames and fusing different levels of features across the video, and 3) we introduce a Covariance Loss to mine and preserve unique patterns for each frame.

2. Related Work

Neural Radiance Field. NeRF [24] introduces 2-D images from different viewpoints to reconstruct a cubic neural radiance field for storing the RGB color value and density for each point in the cube. Recently some papers [20, 27, 29, 32, 33, 35] introduce decomposing the neural radiance from the observation space to canonical space for modeling the movement of an object. By predicting the correspondences in the canonical space [29,32,35], deformable NeRF finds the connection between every point in the observation space and the canonical space and uses the moving object in the video for the construction of a unique object. Nerfies [29] and Neural 3-D video synthesis [19] construct a framewise deformation field for aligning the points in different scenes. HyperNeRF [30] builds the hyper-space for recording the topological changes. Although these methods perform well on scene reconstruction, it is difficult to apply directly for the animatable body shapes since they heavily rely on the memory of the projection.

Body Shape Reconstruction and Animation. Constructing the human body shape requires complicated hardware by most methods [5,6,9,10,43]. Recently, researchers have mainly developed two different methods: statistic-based methods [3, 7, 11, 14, 16, 28, 37, 55, 56] and data-based methods [25,26,38,39,41,52,54]. Statistic methods use a predefined body shape with a default linear skinned model for human shape reconstruction. Recent work has also introduced some non-statistics models based on the body shape reconstruction for 3-D human body shape reconstruction. [38, 39] introduce using implicit functions for 3-D estimation. [51] introduced using normals to correct the reconstruction model generated with SMPL shape. With the development of NeRF [24], researchers also introduced using the implicit function [20, 27] and other 3-D representations [1, 40, 44, 50, 53] for modeling the static body shapes.

To animate the body shapes and render them in the scene, animatable NeRF, different from the deformable methods, projects the body shape into a canonical space [18, 32, 33, 49] or a common shape shapes [34, 36] for projecting the human body shape from different frames into a shared shape or space. Recently, researchers have used statistical methods such as SMPL [22] and SMPL-X [31] representations for body templates. However, these statistical skinned models cannot precisely model the body shapes in the scene, considering different body shapes and clothes. Researchers have proposed different methods to bridge the gap between these two models. Methods such as NARF [27] and A-NeRF [42] do not make explicit canonical space modeling. SNARF [4] and Animatable NeRF [32] utilize the neural blend weight field with frame-level correction for building such correspondence with statistical methods, while TAVA [18] uses the linear blend skinning (LBS) and apply a constant change for modeling muscles and clothing dynamics. No existing methods take both unique framewise dynamics and temporal constancy into consideration.

3. Method

For each person in a video, we have n camera viewpoints ($n \geq 1$) recording the same sequence from n synchronized and geometry calibrated cameras and generate a frame sequence $\{v_i\}_{i=1,2,...,N}$. i is the current frame number, and N

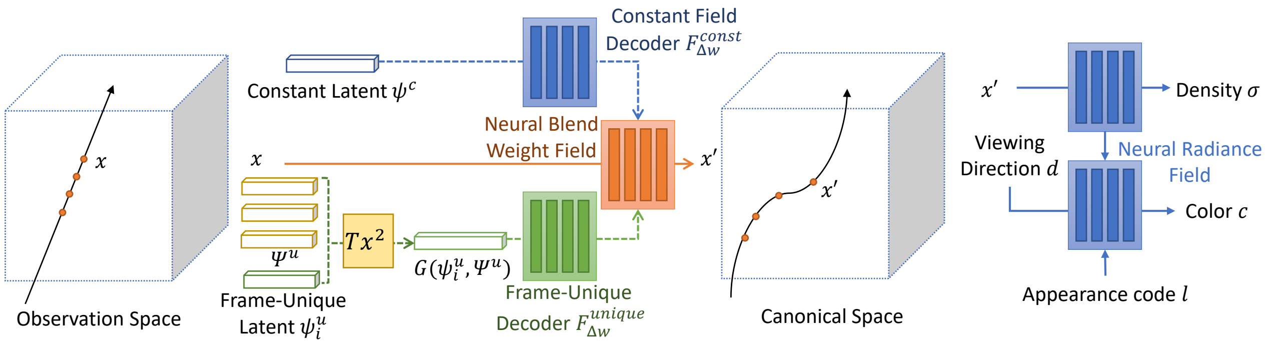


Figure 2. Architecture for the proposed CAT-NeRF for frame uniqueness and temporal constancy decomposition. Rectangles in the boxes are trainable MLP layers. We apply the constant and frame-unique field decoders for temporal constant and unique features, respectively.

is the length of the sequence. We show the architecture of CAT-NeRF in Figure 2. During training, we have both camera positions and corresponding groundtruth images, while during inference, with novel camera viewpoints, we render the image with the corresponding feature vectors by predicting the value of each pixel.

We first briefly review Animatable NeRF [32] in 3.1. We then introduce CAT-NeRF in Sec. 3.2. For framewise uniqueness, we introduce the covariance loss in Sec. 3.3, followed by the overall objective for training in Sec. 3.4.

3.1. Animatable NeRF for Human Modeling

To represent a dynamic scene or object in the video, Animatable NeRF [32] constructs two spaces: one observation space representing the shape we observe for each individual frame and one canonical space shared by all the frames describing the same object with a default pose. For a point x in the observation space, it constructs the scene or object of a frame within a video with two fields representing the density value $\sigma(x)$ and RGB value $c(x)$ of each point following

$$\begin{aligned} \sigma(x), z_i(x) &= F_\sigma(\gamma_x(T_{oc}(x'))); \\ c(x) &= F_c(z_i(x),\ \gamma_d(d),\ l)) \end{aligned} \tag{1}$$

where x' is the point in the canonical space corresponding to x. d is the observation direction and l is the specific feature representative for each frame. z is a learnable representation. γ_d and γ_x are the two position encoding functions following [27]. T_{oc} is the transformation function to find the corresponding point x in the observation space from the point x' in the canonical space, which is formulated as the neural blend weight field [2, 12]. By separating the human body shape with K parts based on the linear skinned model, Animatable NeRF builds the function T_{oc} as

$$T_{oc}(x') = (\sum_{k=1}^{K} w(x')_k G_k) x' \tag{2}$$

where G_k is the $SE(3)$ transformation matrix for the corresponding body part and $w(v)$ is the weight for each point x' in the canonical space. In addition, Animatable NeRF utilizes a frame-wise latent code ψ_i for bridging the differences between the statistic model $w_s(x', S_i)$ and $w_i(x')$ for frame i. The final blend weight is calculated as

$$w_i(x') = norm(F_{\Delta w}(x', \psi_i) + w_s(x', S_i)) \tag{3}$$

where $w_s(\cdot)$ is from the statistical shape S_i.

3.2. CAT-NeRF: Constant and Unique Appearance

Although $F_{\Delta w}(x', \psi_i)$ helps bridge the differences between the statistic model and the rendered shape, framewise corrections only use information from current frames and do not utilize information from the whole sequence. Modeling the appearance of dynamic shapes needs to consider both constant appearances that are shared by all the frames and the framewise unique appearances for the unique dynamics that only appear in one or a few frames.

Based on neural blend weight fields, we introduce CAT-NeRF to deal with the constant and unique appearance patterns via mining the temporal constancy in the sequence. Specifically, we use ψ_i^u to store framewise uniqueness for frame i, and ψ^c for appearance constancy shared among all the frames. In this way, we decompose the dynamic body rendering of the neural blend weight field following

$$F_{\Delta w}(x') = F_{\Delta w}^{const}(x_i, \psi^c) + F_{\Delta w}^{unique}(x_i, G(\psi_i^u, \Psi^u)) \tag{4}$$

where $F_{\Delta w}^{const}$ and $F_{\Delta w}^{unique}$ represent two networks to decode the corresponding temporal constant and framewise adjustment of the appearance for the dynamic person, respectively. G is the Tx^{2}Former for combining framewise features for frame i as Figure 3.

Since some appearances of the body are shared among a small set of frames that are neither constant nor framewise unique, we introduce Transformer-on-Transformer (Tx^{2}Former) as a novel way of combining two Transformer [46] layers. Tx^{2}Former combines the framewise-unique features to model the appearances only shared by a set of

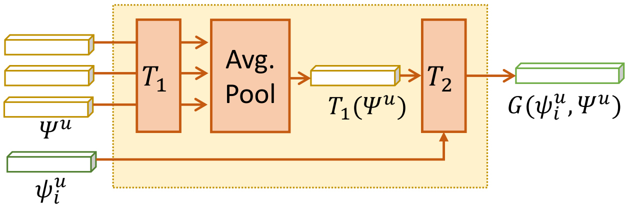


Figure 3. With the collection of frame-unique latent Ψ^u and current frame ψ_i^u, we use T_1 for fusing the features in the neighbor frames and T_2 for combining the features with the current frame.

frames. We have two stacks of Transformer layers [46] to fuse the framewise feature for frame i with other frames in the video. The final output of $G(\psi_i^u, \Psi^u)$ follows

$$G(\psi_i^u, \Psi^u) = T_2(\psi_i^u, T_1(\Psi^u)) \tag{5}$$

where T_1 and T_2 are two stacks of Transformer encoder layers. The first Transformer encoder fuses all the unique features to generate a global understanding of what is included for all the frames in this sequence besides appearance constancy ψ^c. We average pool the output T_1 and concatenate it with the corresponding framewise-unique shape ψ_i^u of the current frame before sending it to T_2.

Discussion: Unlike other vision transformers [8, 21, 45] that require the output of each patch used in the input, the reconstruction of the current frame relies more on ψ_i^u and compared with the output of the other frames. In addition, using the original Transformer generates $N-1$ outputs that are not used for rendering since we only focus on the output features for frame i. Moreover, traditional Transformers compute the self-attention across each pair of frames, which still focuses on the frame-level understanding without sequential knowledge. In Tx^{2}Former, concatenating the pooled features and ψ_i^u shortens the sequence length for the input of T_2, making the model focus on ψ_i^u and select the sequence-level information from T_1's output based on ψ_i^u.

3.3. Covariance Loss for Unique Patterns

To split the latent representation for framewise uniqueness ψ_i^u and the constancy ψ^c shared by all frames, we introduce a covariance-related loss to separate the information stored in these two representations in individual frames. For a video $i \in \{1, 2, ..., N\}$, where N is the overall number of frames, we have the collection of frame unique representations $\Psi^u = (\psi_1^u, \psi_2^u, ..., \psi_n^u)$ for framewise uniqueness. The covariance loss is as follows

$$L_{cov} = \frac{\sum ||cov(\Psi^u, \Psi^u)|| - \sum Diag(||cov(\Psi^u, \Psi^u)||)}{(N-1)^2} \tag{6}$$

$cov(\cdot)$ is the covariance matrix for Ψ^u, and $\sum$ represents the sum of every element in its bracket. $||\cdot||$ indicates using the absolute value for every element.

For a covariance matrix, each element $cov_{i,j}$ represents the covariance value between the two features ψ_i and ψ_j. Since the diagonal value for the covariance matrix is the covariance between a variable to itself, it is the variance of this variable. Since 0 variance indicates all the elements in the vector are 0, we remove this item in our loss function.

Discussion: When the covariance value between two variables is not zero, these two variables tend to vary in the same or opposite direction. Since the range for the covariance value is in $(-\infty, +\infty)$, we use the absolute value for each element to change its range to $[0, +\infty)$. A smaller absolute value for covariance indicates that these two variables are less likely to vary together. In this case, we set 0 as the minimum of the loss function and our target for optimization. By reducing the correlation between each pair of the framewise-unique features, ψ_i^l for frame i is less likely to vary along with other frames and increasingly represent the uniqueness of each frame. This also allows the constant feature ψ^c to capture the maximum amount of the temporal constancy information and reduce the reuse of the same feature representations in ψ_i^u.

To show that L_{cov} can minimize temporal constancy information in ψ^u across frames, for each feature ψ_i, we can decompose it into two vectors: a, which is related to other feature ψ_j at timestamp j, $i \neq j$, and b, which is unique and not shared with other features. In CAT-NeRF, we use the learnable feature vector ψ^u to represent the frame's unique feature and ψ^c to represent the constancy. Thus we have

$$\psi_i = \psi^u + \psi^c = a + b$$

If two features are related to each other, the decoded results are similar after decoding with the same network and have strong connections. On the contrary, if two features represent two distinct shapes sharing no similarity, the correlation between them should be minimized. Thus by applying the correlation loss on ψ^u, we allow the sharable feature a to be minimally captured by ψ^u and make ψ^c learn the constancy. In the meantime, since the constant feature is captured by ψ^c, ψ^u has more capability of capturing the framewise-unique feature b and includes more fine-grained details for each frame in the sequence.

3.4. Objective

To train the model, we follow [32] to build the objective function for training ψ^c, ψ_i^u, $F_{\Delta w}^{const}$, $F_{\Delta w}^{unique}$, $G(\cdot)$, F_σ and F_c jointly. The final objective for training is

$$\begin{aligned} L &= L_{cov} + L_{rgb} + L_{nsf} \\ L_{rgb} &= \sum_{r \in R} ||\tilde{C}_i(r) - C_i(r)||_2 \\ L_{nsf} &= \sum_{x \in \mathcal{X}} ||w_i(x) - w^{can}(T_{oc}(x))||_1 \end{aligned} \tag{7}$$

where R is the collection for all the rays that go through the pixel and $\mathcal{X}$ represents all the points sampled in the volumetric field. $||\cdot||_k$ is the k-norm value. L_{rgb} assess the differences between the final rendered color $\tilde{C}_i(r)$ with the groundtruth value $C_i(r)$ for each pixel. Since the blend weight fields between the points in observation space x and canonical space $T_{oc}(x)$ should be the same for Eq. 3, we follow [32] for establishing L_{nsf} to minimize the difference.

4. Experiments

In this section, we present our settings and results. We first show the dataset description in Sec. 4.1, followed by the implementation details in Sec. 4.2. With these experimental details, we show our results in Sec. 4.3 and 4.4.

4.1. Datasets

In our experiments, we compare our methods with baseline methods on two different datasets: H36M [13] and ZJU-MoCap [33]. These two datasets capture the moving pattern of different poses of the same person from different camera viewpoints whose viewpoints are available. We follow [32] to select the frames and generate the splits for training and inference in our experiment.

H36M [13] includes videos of different poses for the same and unique person from 4 different camera viewpoints. In our experiment, we follow [32] to select the videos from subjects S1, S5, S6, S7, S8, S9 and S11. We use the first three viewpoints (0, 1 and 2) for training and the remaining for inference for the four camera viewpoints. For novel view synthesis, the number of frames for training and testing for these subjects varies between 30 and 60 for each camera viewpoint. For novel pose synthesis, we use 49 to 200 frames for each subject for evaluation. The size of the image is set to 1002×1000.

ZJU-MoCap [33] includes videos captured from 21 different cameras to collect different human poses. We follow [32] to use the videos in four categories, “Twirl”, “Taichi”, “Warmup”, and “Punch1” in our experiment. We select four viewpoints from positions 0, 6, 12 and 18 from the dataset for training and use the remaining 17 viewpoints for inference. For novel view synthesis, the number of frames for training and testing for these subjects varies between 60 and 400 for each camera viewpoint. For novel pose synthesis, we use 346 to 1,000 frames for each subject for evaluation. The size of the image for each frame is 1024×1024.

4.2. Implementation Details

Training and Inference. To extract the SMPL shapes for RGB frames, we follow [15] to generate the SMPL reconstruction. We follow [2, 12] to generate the neural blend weight field to find the three nearest points on the skinned model for building the field. For each batch, we sample 4,096 rays and for each ray, we sample 64 points.

To train the network for novel camera viewpoints, we follow [32] to implement our network. For ψ^c and ψ^u, we set the number for both dimensionalities as 128. To train the model for novel poses, we first get a model for a novel view with constant and unique features, respectively. After that, we copy the frame-unique features reconstructed for the current poses to the novel poses and use the smooth-L1 loss for the novel constant feature for body shape and the original feature generated from the novel view. For both training steps, we use the Adam optimizer [17] and set the initial learning rate as $5e-4$ and decay it to $\frac{1}{10}$ after 1,000 epochs with an exponential training scheduler following [32]. The number of epochs is set to 400. During inference, we use the pretrained constant and unique features to decode the density and RGB value for each point in the field. Our network takes 6-8 hours to train on a novel viewpoint setting and 20-30 hours to train on the novel pose setting for each subject on an Nvidia A40 or A100 GPU.

Metrics. For our experiment, we have two different metrics between the projection of new images with our generated results for comparison, PSNR and SSIM [48]. PSNR and SSIM describe the quality of the reconstructed image to the original image. Higher values represent better performance for both metrics.

Baseline Methods. In our experiment, we compare our method with Neural Texture [44], NHR [50] and Animated NeRF [32]. We also report SMPLpix [34] on H36M, along with NeuralBody [33] and HumanNeRF [49] on ZJU-MoCap, since the authors did not provide numbers on the other datasets. We follow [32] to use the body shape reconstruction from SMPL [22] as the input for Neural Texture [44] as the coarse mesh to render the image and use the points sampled from the SMPL body shape reconstruction as the input for NHR. In addition, we also compare a randomly generated image with untrained Animatable NeRF for PSNR and SSIM since these two scores for randomly generated images are not the lowest theoretical values (0) for both of these two metrics. For all these methods, we follow the setting of Animatable NeRF[1].

4.3. Numerical Results

We show our numerical results on H36M [13] and ZJU-MoCap [33], along with the ablation study for L_{cov} and Tx^{2}Former architecture, in this subsection. We present other ablation studies, such as hyperparameter selections, in the **supplementary materials**.

Results on H36M. We show the numerical results for H36M dataset rendered for novel view and novel pose in Table 1 and Table 2 respectively. For the results in the tables, NT is the result for Neural Texture [44] and AN is the result from Animatable NeRF [32].

[1] https://github.com/zju3dv/animatable_nerf

Splits	PSNR (↑)						SSIM (↑)					
	Rand	NT	NHR	SMPLpix	AN	Ours	Rand	NT	NHR	SMPLpix	AN	Ours
S1	17.79	20.98	21.08	22.01	22.05	**24.52**	0.784	0.860	0.872	0.882	0.888	**0.905**
S5	18.19	19.87	20.64	23.35	23.27	**24.23**	0.781	0.855	0.872	0.879	0.892	**0.901**
S6	18.08	20.18	20.40	21.09	21.13	**24.24**	0.769	0.816	0.830	0.860	0.854	**0.875**
S7	16.51	20.47	20.29	22.03	22.50	**24.11**	0.753	0.856	0.868	0.888	0.890	**0.899**
S8	16.94	16.77	19.13	22.22	22.75	**23.66**	0.762	0.837	0.871	0.895	0.898	**0.904**
S9	18.26	22.96	23.04	23.99	24.72	**25.95**	0.770	0.873	0.879	0.902	0.908	**0.909**
S11	18.98	21.71	21.91	22.05	24.55	**25.26**	0.756	0.859	0.871	0.889	0.902	**0.905**
Average	17.82	20.42	20.93	22.39	23.00	**24.57**	0.768	0.851	0.866	0.885	0.890	**0.900**

Table 1. Results of novel view synthesis on H36M dataset. NT and AN represents Neural Textures and Animatable NeRF respectively. (↑) indicates higher results are better.

Splits	PSNR (↑)						SSIM (↑)					
	Rand	NT	NHR	SMPLpix	AN	Ours	Rand	NT	NHR	SMPLpix	AN	Ours
S1	16.51	20.09	20.48	21.90	21.37	**23.34**	0.741	0.837	0.853	0.875	0.868	**0.888**
S5	17.80	20.03	20.72	23.01	22.29	**23.32**	0.749	0.843	0.860	0.878	0.875	**0.888**
S6	17.94	20.42	20.47	21.89	22.69	**24.55**	0.805	0.844	0.856	0.865	0.884	**0.891**
S7	15.92	20.03	19.66	22.12	22.22	**22.72**	0.723	0.838	0.852	0.873	**0.878**	**0.878**
S8	16.36	16.69	18.83	22.01	21.78	**22.90**	0.750	0.824	0.855	0.889	0.882	**0.895**
S9	17.53	22.20	22.18	23.91	23.72	**24.74**	0.738	0.851	0.860	0.890	0.886	**0.892**
S11	19.64	21.72	22.12	22.45	23.91	**24.24**	0.747	0.854	0.867	0.875	0.889	**0.891**
Average	17.39	20.17	20.64	22.47	22.55	**23.68**	0.750	0.841	0.858	0.878	0.880	**0.889**

Table 2. Results of novel pose synthesis on H36M dataset. NT and AN represents Neural Textures and Animatable NeRF respectively.

For the results on the novel view setting shown in Table 1, we outperform the state-of-the-art baseline method, Animatable NeRF [32], on all splits in the dataset for both metrics we assessed. Since the PSNR and SSIM for a randomly generated image are not 0, we achieve a 30.3% and 8.2% relative improvement compared with the differences between our baseline method, Animatable NeRF, and the randomly generated images. Constant features find the appearance constancy across different frames in the video and make full use of all the features, as well as reduce randomness with the assistance of constant appearances in the other frames, while the framewise feature can refine the details based on the constant appearances.

In addition to the results for novel viewpoints, we show the results on novel poses in Table 2 comparing with the baseline methods on both metrics. Our method consistently improves on most of the splits for both metrics. Specifically, compared with Animatable NeRF, the best baseline method in our experiment on H36M, we have 21.9% and 6.9% relative improvements on PSNR and SSIM, respectively.

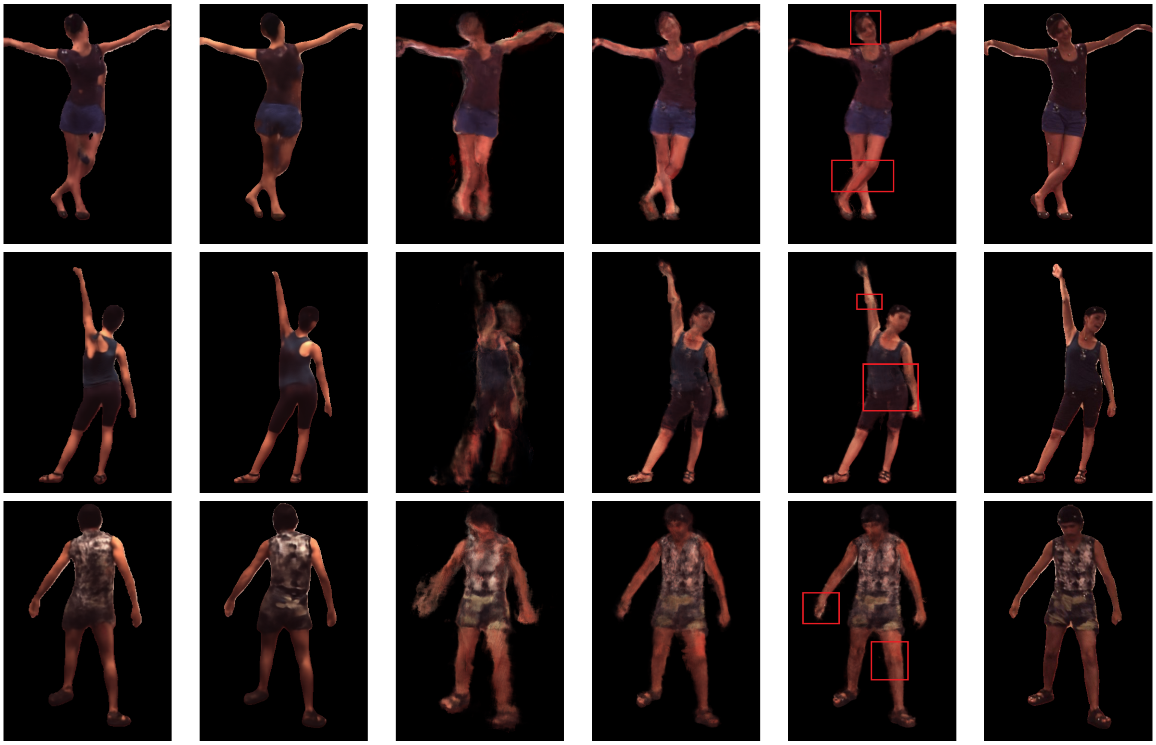
Results on ZJU-MoCap. In addition to the results on the H36M dataset, we also show our average results for PSNR and SSIM for the ZJU-MoCap dataset in Table 3. For the numerical results, we have similar PSNR and SSIM compared with NeuralBody and HumanNeRF on the novel view setting and best performance on novel pose synthesis. Compared with Animatable NeRF, we achieve a 19.1% relative improvement compared to PSNR on novel views. Compared with the H36M dataset, ZJU-MoCap is larger, but its action variations are comparatively fewer, resulting in the improvements being smaller but still consistent.

Ablation for Covariance Loss. To assess the covariance loss L_{cov} in Eq. 6, we replace it with three variations: 1) No extra loss for splitting, 2) Correlation loss L_{corr}, and 3) KL Divergence L_{KLD} with the $N(0,1)$ distribution. L_{corr} is the product of the correlation scores between every two dimensionalities in Ψ following

$$L_{corr} = \frac{\sum(x_1-\bar{x_1})(x_2-\bar{x_2})...(x_n-\bar{x_n})}{\sqrt{\sum(x_1-\bar{x_1})^2\sum(x_2-\bar{x_2})^2...\sum(x_n-\bar{x_n})^2}} \tag{8}$$

where x_i is the i^{th} dimension of training features.

We show the results in Table 4 on the s1p subject of the H36M dataset. We note that, without any extra loss, our network already outperforms the original Animatable NeRF, indicating the constant features help find the appearance constancy in the sequence for rendering. In addition, all three methods have better results than the one with no additional loss applied, while Covariance loss in Eq. 6 has the best performance. Using product instead of sum and the lack of examples compared with the feature dimensionality make L_{corr} and L_{KLD} less likely to be optimized. In con-



(a) Neural Texture (b) NHR (c) D-NeRF (d) AN (e) ours (f) Groundtruth

Figure 4. Visualization of novel view reconstruction for the three test examples in H36M dataset. AN stands for Animatable NeRF. We highlight the significant improvements within boxes in our reconstructions.

trast, L_{cov} is capable of finding a stable solution to push ψ^u to find the uniqueness for each frame.

Ablation for $G(\cdot)$***.*** Instead of using Tx^{2}Former for fusing frame-unique feature, we show the ablations of different variations in Table 5. We compare Tx^{2}Former (denotes as Tx^2) with 1) unprocessed frame-level latent ψ_i^u, 2) average pooling of all Ψ^u, 3) the original combination of two transformer layers Tx and 4) replacing the first layer with average pooling. We also compare these methods on s1p object of the H36m dataset and show that Tx^{2}Former has the best performance, indicating Tx^{2}Former being able to mine the neighbor features and find out the missing features in ψ_i^u.

4.4. Visualization Results

Comparison with other methods. We show some rendered images in Figure 4 on the H36M dataset with novel view reconstruction results. In addition to the Neural Texture [44], NHR [50] and Animatable NeRF [32], we also compared with D-NeRF [35] for reconstruction. D-NeRF utilizes the SE (3) transformations to find the corresponding points in the canonical space for the points observed.

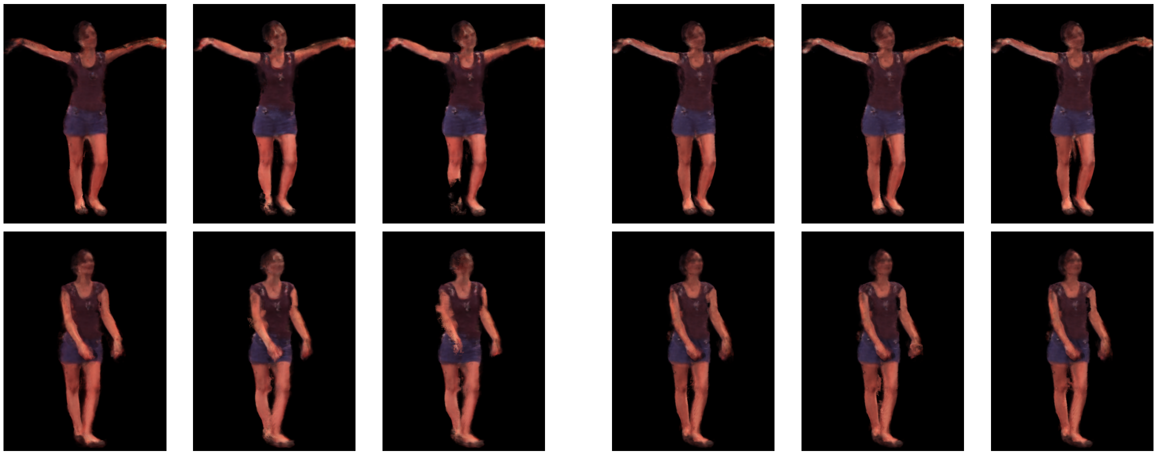
With the introduction of decoupling of the constant and frame-unique appearance with the blend weight between the statistic shape and the final reconstructed shape, our model can capture the body shape more accurately and with clearer boundaries. For example, in the first and third rows, our method is capable of making the correct prediction for the position of the two legs and having a clearer facial appearance. We have clearer rendered clothing results for the sec-

Metric	Setting	Rand	NT	NHR	NB	HN	AN	Ours
PSNR	View	17.83	22.61	23.25	28.90	**29.01**	27.10	28.87
	Pose	18.19	21.55	21.88	23.06	23.20	23.16	**23.62**
SSIM	View	0.801	0.899	0.905	0.967	**0.966**	0.949	0.955
	Pose	0.790	0.860	0.863	0.879	0.885	0.893	**0.899**

Table 3. Results of PSNR and SSIM for novel view and pose synthesis on ZJU-MoCap dataset. NT, HN and AN represents Neural Textures, HumanNeRF and Animatable NeRF. 'Pose' and 'View' represent the novel pose and novel view settings respectively.

Loss Type	AN	No Loss	L_{corr}	L_{KLD}	L_{cov}
PSNR	22.05	22.86	23.52	23.67	**24.52**

Table 4. Ablation for using different loss functions to replace L_{cov} in Eq. 7, along with original Animatable NeRF (denotes as AN) results. All methods except AN are results for our model. 'No Loss' is to remove L_{cov} in L of Eq. 7.



(a) $\Delta t = 0$ (b) $\Delta t = 4$ (c) $\Delta t = 8$

Animatable NeRF

(a) $\Delta t = 0$ (b) $\Delta t = 4$ (c) $\Delta t = 8$

Ours

Figure 5. Visualization of novel view reconstruction for the test examples in H36M dataset with neural blend weight field from other frames. Δt indicates the differences of the frame id in the temporal sequence.

Loss Type	ψ_i^l	Avg	Tx	Avg+T_2	Tx^2
PSNR	22.05	23.09	23.19	23.82	**24.52**

Table 5. Ablation for different frame-unique features selection as $G(\cdot)$. Avg is average pooling and Tx is one-layer encoder.

ond example than Animated NeRF. Using the constant features to capture such appearance constancy allows our network to find and preserve the temporal similarities instead of being dominated by unique patterns.

Evaluation for Appearance Constancy. Since we separate the temporal constant and framewise-unique features for dynamic human body shape rendering, in this experiment, we assess how much information has been captured by the constant vector as the appearance constancy. We exchange the frame-unique representations ψ^u between different frames to see how big a difference the rendered image has over the original rendered results. We select two frames each time, one from t and another from $t + \Delta t$, and apply the neural blend weight field from frame $t + \Delta t$ to frame t. The images of the person are captured with the same camera viewpoint but from different timestamps. If the constant feature captures enough appearance constancy, changes in the frame-unique feature vector will not have big differences in the final rendered image since the temporal constant feature is shared between two frames.

We show the results in Figure 5. With the framewise features from a different timestamp, Animatable NeRF reconstructs a blurred image with general appearances unchanged, indicating most patterns and appearances are similar between the frames. As a comparison, CAT-NeRF stores such appearance constancies in the sequence. For example, in the second row, when we use the framewise features from other frames, the facial appearance of Animatable NeRF is not stable, while ours show a more accurate rendering. When such appearances, such as facial patterns, are constant and shared across all the frames, our model can construct a more confident result from the sequential inputs.

5. Conclusion

We introduce CAT-NeRF for mining and separating the appearance constancy and uniqueness for dynamic body shape rendering. The constant feature predicts the constant appearances that are shareable for all frames for the same person in the sequence, while the unique framewise feature simulates the unique dynamics in the sequential rendering. In addition, we introduce Tx^{2}Former for combining different levels of features and a specific Covariance Loss to ensure the unique appearance for each frame is correctly captured. Our method outperforms state-of-the-art methods on the public datasets, H36M [13] and ZJU-MoCap [33].

Acknowledgement. This research is based upon work supported in part by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via [2022-21102100007]. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of ODNI, IARPA, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation therein.

References

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

[2] Bharat Lal Bhatnagar, Cristian Sminchisescu, Christian Theobalt, and Gerard Pons-Moll. Loopreg: Self-supervised learning of implicit surface correspondences, pose and shape for 3d human mesh registration. *NeurIPS*, 33:12909–12922, 2020. 3, 5

[3] Federica Bogo, Angjoo Kanazawa, Christoph Lassner, Peter Gehler, Javier Romero, and Michael J Black. Keep it smpl: Automatic estimation of 3d human pose and shape from a single image. In *ECCV*, pages 561–578, 2016. 2

[4] Xu Chen, Yufeng Zheng, Michael J Black, Otmar Hilliges, and Andreas Geiger. Snarf: Differentiable forward skinning for animating non-rigid neural implicit shapes. In *ICCV*, pages 11594–11604, 2021. 2

[5] Alvaro Collet, Ming Chuang, Pat Sweeney, Don Gillett, Dennis Evseev, David Calabrese, Hugues Hoppe, Adam Kirk, and Steve Sullivan. High-quality streamable free-viewpoint video. *TOG*, 34(4):1–13, 2015. 2

[6] Paul Debevec, Tim Hawkins, Chris Tchou, Haarm-Pieter Duiker, Westley Sarokin, and Mark Sagar. Acquiring the reflectance field of a human face. In *SIGGRAPH*, pages 145–156, 2000. 2

[7] Junting Dong, Qing Shuai, Yuanqing Zhang, Xian Liu, Xiaowei Zhou, and Hujun Bao. Motion capture from internet videos. In *ECCV*, pages 210–227. Springer, 2020. 2

[8] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 4

[9] Mingsong Dou, Sameh Khamis, Yury Degtyarev, Philip Davidson, Sean Ryan Fanello, Adarsh Kowdle, Sergio Orts Escolano, Christoph Rhemann, David Kim, Jonathan Taylor, et al. Fusion4d: Real-time performance capture of challenging scenes. *ACM ToG*, 35(4):1–13, 2016. 2

[10] Kaiwen Guo, Peter Lincoln, Philip Davidson, Jay Busch, Xueming Yu, Matt Whalen, Geoff Harvey, Sergio Orts-Escolano, Rohit Pandey, Jason Dourgarian, et al. The relightables: Volumetric performance capture of humans with realistic relighting. *ACM TOG*, 38:1–19, 2019. 2

[11] Peter Hedman, Pratul P Srinivasan, Ben Mildenhall, Jonathan T Barron, and Paul Debevec. Baking neural radiance fields for real-time view synthesis. In *ICCV*, pages 5875–5884, 2021. 2

[12] Zeng Huang, Yuanlu Xu, Christoph Lassner, Hao Li, and Tony Tung. Arch: Animatable reconstruction of clothed humans. In *CVPR*, pages 3093–3102, 2020. 3, 5

[13] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. Human3. 6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *TPAMI*, pages 1325–1339, 2013. 2, 5, 8

[14] Wen Jiang, Nikos Kolotouros, Georgios Pavlakos, Xiaowei Zhou, and Kostas Daniilidis. Coherent reconstruction of multiple humans from a single image. In *CVPR*, pages 5579–5588, 2020. 2

[15] Hanbyul Joo, Tomas Simon, and Yaser Sheikh. Total capture: A 3d deformation model for tracking faces, hands, and bodies. In *CVPR*, pages 8320–8329, 2018. 5

[16] Angjoo Kanazawa, Michael J Black, David W Jacobs, and Jitendra Malik. End-to-end recovery of human shape and pose. In *CVPR*, pages 7122–7131, 2018. 2

[17] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5

[18] Ruilong Li, Julian Tanke, Minh Vo, Michael Zollhofer, Jurgen Gall, Angjoo Kanazawa, and Christoph Lassner. Tava: Template-free animatable volumetric actors. *arXiv preprint arXiv:2206.08929*, 2022. 2, 13

[19] Zhengqi Li, Simon Niklaus, Noah Snavely, and Oliver Wang. Neural scene flow fields for space-time view synthesis of dynamic scenes. In *CVPR*, pages 6498–6508, 2021. 2

[20] Lingjie Liu, Marc Habermann, Viktor Rudnev, Kripasindhu Sarkar, Jiatao Gu, and Christian Theobalt. Neural actor: Neural free-view synthesis of human actors with pose control. *ACM SIGGRAPH Asia*, 2021. 2

[21] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, pages 10012–10022, 2021. 4

[22] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J Black. Smpl: A skinned multi-person linear model. *TOG*, 34(6):1–16, 2015. 2, 5

[23] Ricardo Martin-Brualla, Noha Radwan, Mehdi SM Sajjadi, Jonathan T Barron, Alexey Dosovitskiy, and Daniel Duckworth. Nerf in the wild: Neural radiance fields for unconstrained photo collections. In *CVPR*, pages 7210–7219, 2021. 11

[24] 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 *ECCV*, pages 405–421. Springer, 2020. 1, 2, 11

[25] Ryota Natsume, Shunsuke Saito, Zeng Huang, Weikai Chen, Chongyang Ma, Hao Li, and Shigeo Morishima. Siclope: Silhouette-based clothed people. In *CVPR*, pages 4480–4490, 2019. 2

[26] Michael Niemeyer, Lars Mescheder, Michael Oechsle, and Andreas Geiger. Differentiable volumetric rendering: Learning implicit 3d representations without 3d supervision. In *CVPR*, pages 3504–3515, 2020. 2

[27] Atsuhiro Noguchi, Xiao Sun, Stephen Lin, and Tatsuya Harada. Neural articulated radiance field. In *ICCV*, pages 5762–5772, 2021. 2, 3

[28] Ahmed AA Osman, Timo Bolkart, and Michael J Black. Star: Sparse trained articulated human body regressor. In *ECCV*, pages 598–613, 2020. 2

[29] Keunhong Park, Utkarsh Sinha, Jonathan T Barron, Sofien Bouaziz, Dan B Goldman, Steven M Seitz, and Ricardo Martin-Brualla. Nerfies: Deformable neural radiance fields. In *ICCV*, pages 5865–5874, 2021. 1, 2

[30] Keunhong Park, Utkarsh Sinha, Peter Hedman, Jonathan T Barron, Sofien Bouaziz, Dan B Goldman, Ricardo Martin-Brualla, and Steven M Seitz. Hypernerf: A higher-dimensional representation for topologically varying neural radiance fields. *arXiv preprint arXiv:2106.13228*, 2021. 2

[31] Georgios Pavlakos, Vasileios Choutas, Nima Ghorbani, Timo Bolkart, Ahmed AA Osman, Dimitrios Tzionas, and Michael J Black. Expressive body capture: 3d hands, face, and body from a single image. In *CVPR*, pages 10975–10985, 2019. 2

[32] Sida Peng, Junting Dong, Qianqian Wang, Shangzhan Zhang, Qing Shuai, Xiaowei Zhou, and Hujun Bao. Animatable neural radiance fields for modeling dynamic human bodies. In *ICCV*, pages 14314–14323, 2021. 1, 2, 3, 4, 5, 6, 7, 11, 12

[33] Sida Peng, Yuanqing Zhang, Yinghao Xu, Qianqian Wang, Qing Shuai, Hujun Bao, and Xiaowei Zhou. Neural body: Implicit neural representations with structured latent codes for novel view synthesis of dynamic humans. In *CVPR*, pages 9054–9063, 2021. 2, 5, 8

[34] Sergey Prokudin, Michael J Black, and Javier Romero. Smplpix: Neural avatars from 3d human models. In *WACV*, pages 1810–1819, 2021. 2, 5

[35] Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-nerf: Neural radiance fields for dynamic scenes. In *CVPR*, pages 10318–10327, 2021. 1, 2, 7

[36] Amit Raj, Julian Tanke, James Hays, Minh Vo, Carsten Stoll, and Christoph Lassner. Anr: Articulated neural rendering for virtual avatars. In *CVPR*, pages 3722–3731, 2021. 2

[37] Javier Romero, Dimitrios Tzionas, and Michael J Black. Embodied hands: Modeling and capturing hands and bodies together. *TOG*, 36(6):1–17, 2017. 2

[38] Shunsuke Saito, Zeng Huang, Ryota Natsume, Shigeo Morishima, Angjoo Kanazawa, and Hao Li. Pifu: Pixel-aligned implicit function for high-resolution clothed human digitization. In *ICCV*, pages 2304–2314, 2019. 2

[39] Shunsuke Saito, Tomas Simon, Jason Saragih, and Hanbyul Joo. Pifuhd: Multi-level pixel-aligned implicit function for high-resolution 3d human digitization. In *CVPR*, pages 84–93, 2020. 2

[40] Aliaksandra Shysheya, Egor Zakharov, Kara-Ali Aliev, Renat Bashirov, Egor Burkov, Karim Iskakov, Aleksei Ivakhnenko, Yury Malkov, Igor Pasechnik, Dmitry Ulyanov, et al. Textured neural avatars. In *CVPR*, pages 2387–2397, 2019. 2

[41] Vincent Sitzmann, Michael Zollhöfer, and Gordon Wetzstein. Scene representation networks: Continuous 3d-structure-aware neural scene representations. *NeurIPS*, 32, 2019. 2

[42] Shih-Yang Su, Frank Yu, Michael Zollhöfer, and Helge Rhodin. A-nerf: Articulated neural radiance fields for learning human shape, appearance, and pose. *NeurIPS*, 34:12278–12291, 2021. 2

[43] Zhuo Su, Lan Xu, Zerong Zheng, Tao Yu, Yebin Liu, and Lu Fang. Robustfusion: Human volumetric capture with data-driven visual cues using a rgbd camera. In *ECCV*, pages 246–264, 2020. 2

[44] Justus Thies, Michael Zollhöfer, and Matthias Nießner. Deferred neural rendering: Image synthesis using neural textures. *TOG*, 38(4):1–12, 2019. 2, 5, 7

[45] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *ICML*, pages 10347–10357, 2021. 4

[46] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *NeurIPS*, 30, 2017. 2, 3, 4, 11

[47] Shaofei Wang, Katja Schwarz, Andreas Geiger, and Siyu Tang. Arah: Animatable volume rendering of articulated human sdfs. In *ECCV*, 2022. 13

[48] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *TIP*, 13(4):600–612, 2004. 5

[49] Chung-Yi Weng, Brian Curless, Pratul P Srinivasan, Jonathan T Barron, and Ira Kemelmacher-Shlizerman. Humannerf: Free-viewpoint rendering of moving people from monocular video. In *CVPR*, pages 16210–16220, 2022. 2, 5, 13

[50] Minye Wu, Yuehao Wang, Qiang Hu, and Jingyi Yu. Multi-view neural human rendering. In *CVPR*, pages 1682–1691, 2020. 2, 5, 7

[51] Yuliang Xiu, Jinlong Yang, Dimitrios Tzionas, and Michael J Black. Icon: Implicit clothed humans obtained from normals. *arXiv preprint arXiv:2112.09127*, 2021. 2

[52] Lior Yariv, Yoni Kasten, Dror Moran, Meirav Galun, Matan Atzmon, Basri Ronen, and Yaron Lipman. Multiview neural surface reconstruction by disentangling geometry and appearance. *NeurIPS*, 33:2492–2502, 2020. 2

[53] Jae Shin Yoon, Lingjie Liu, Vladislav Golyanik, Kripasindhu Sarkar, Hyun Soo Park, and Christian Theobalt. Pose-guided human animation from a single image in the wild. In *CVPR*, pages 15039–15048, 2021. 2

[54] Zerong Zheng, Tao Yu, Yixuan Wei, Qionghai Dai, and Yebin Liu. Deephuman: 3d human reconstruction from a single image. In *ICCV*, pages 7739–7749, 2019. 2

[55] Haidong Zhu, Ye Yuan, Yiheng Zhu, Xiao Yang, and Ram Nevatia. Open: Order-preserving pointcloud encoder decoder network for body shape refinement. In *ICPR*, pages 521–527, 2022. 2

[56] Haidong Zhu, Zhaoheng Zheng, and Ram Nevatia. Gait recognition using 3-d human body shape inference. In *WACV*, pages 909–918, 2023. 2