Drivable Avatar Clothing: Faithful Full-Body Telepresence with Dynamic Clothing Driven by Sparse RGB-D Input (Supplementary Document)

Donglai Xiang
donglaix@cs.cmu.edu
Carnegie Mellon University
USA

Fabian Prada
fabianprada@meta.com
Meta Reality Labs Research
USA

Zhe Cao
zhecao@berkeley.edu
Meta Reality Labs Research
USA

Kaiwen Guo
guokaiwen_neu@126.com
Meta Reality Labs Research
USA

Chenglei Wu
chengleiwu@gmail.com
Meta Reality Labs Research
USA

Jessica Hodgins
jkh@cmu.edu
Carnegie Mellon University
USA

Timur Bagautdinov
timurb@meta.com
Meta Reality Labs Research
USA

1 RELATED WORK (CONTINUED)

Template-Based Performance Capture. Our work is also related to a group of methods that track human surface by deforming a person-specific or category-specific template or avatar, using either classical optimization [Habermann et al. 2019; Robertini et al. 2016; Xiang et al. 2020; Xu et al. 2018] or network prediction [Habermann et al. 2021, 2020; Jiang et al. 2023; Li et al. 2022]. They achieve better temporal coherency than template-free methods that regress human shape for each frame [Li et al. 2020; Saito et al. 2019, 2020; Xiu et al. 2023, 2022], but focus on reconstructing human geometry and rather than modeling dynamic appearance.

2 ABLATION STUDIES ON N-ICP

We conduct ablation studies on our design of the N-ICP algorithm. The results are shown in Tab. 1. The most naive baseline is to simply use the point cloud as the input feature, shown on the first row of the table. On the second row, we add the closest point residual to the input feature, which provides useful information for surface alignment and enables an iterative update of the deformation parameters. The following rows suggest that the energy gradient derived from the residuals can provide more effective guidance, similar to its critical role in traditional nonlinear optimization. The last two rows verify the benefit of iterative parameter update compared with a one-shot prediction by the network.

Table 1: Ablation studies on different types of input for N-ICP. The evaluation metric is the Mean Squared Error (MSE) in mm^2 between surfaces. $\mathbf{P}$, $\mathbf{r}$ and $\mathbf{g}$ refer to the point cloud, residual and gradient defined in Sec. 4.2. N denotes the number of update iterations. When $N = 1$, the network makes a one-shot prediction. Our full method is shown in the last row.

Input Type	MSE (mm^2)
$\mathbf{P}$ ($N = 1$)	101.19
$\mathbf{P}, \mathbf{r}$ ($N = 3$)	82.72
$\mathbf{P}, \mathbf{g}$ ($N = 3$)	49.60
$\mathbf{P}, \mathbf{r}, \mathbf{g}$ ($N = 1$)	72.30
$\mathbf{P}, \mathbf{r}, \mathbf{g}$ ($N = 3$, full)	**48.47**

3 DETAIL OF COMPARISON WITH SENSING-BASED BASELINES (SEC. 6.3)

Here, we provide the implementation detail for the sensing-based baselines for the experiment in Sec. 6.3 in the main paper. We first fuse the sparse input depth maps into a single Truncated Signed Distance Field (TSDF) volume [Curless and Levoy 1996; Dong et al. 2022], and then extract from it an explicit mesh representation. Using the fused geometry, we can then warp the input RGB images from the source views to any target view. However, the warped image is usually imperfect because the fused geometry is often incomplete and noisy. Therefore, we follow the idea of LookinGood [Martin-Brualla et al. 2018] and train a U-Net to complete the warped image. This baseline essentially learns to inpaint complete human appearance from partial input only in the screen space, and struggles to achieve 3D-aware temporal consistency in the output. As explained in the main paper, this experiment is not intended to be a full-scale comparison against state-of-the-art sensing-based approaches, but to better understand our method in comparison to a modest baseline along this line of work given similar input.

4 COMPARISON WITH CLOTHING CODEC AVATARS AND DRESSING AVATARS

We highlight the difference in formluation between our method, Clothing Codec Avatars (CCA) [Xiang et al. 2021] and Dressing Avatars (DA) [Xiang et al. 2022] in Tab. 2. In terms of driving signal, CCA and DA take body and face motion as input, while our method additionally uses sparse RGB-D views. DA and our

Work done when DX was a visiting researcher at Meta. ZC and CW are now at Google.

Table 2: Comparison between Clothing Codec Avatars (CCA) [Xiang et al. 2021], Dressing Avatars (DA) [Xiang et al. 2022] and our method.

	CCA	DA	Ours
RGB-D driving input			✓
Loose clothing dynamics		✓	✓
Physical simulation		✓	
Ground truth registration	✓	✓	
Faithful output			✓

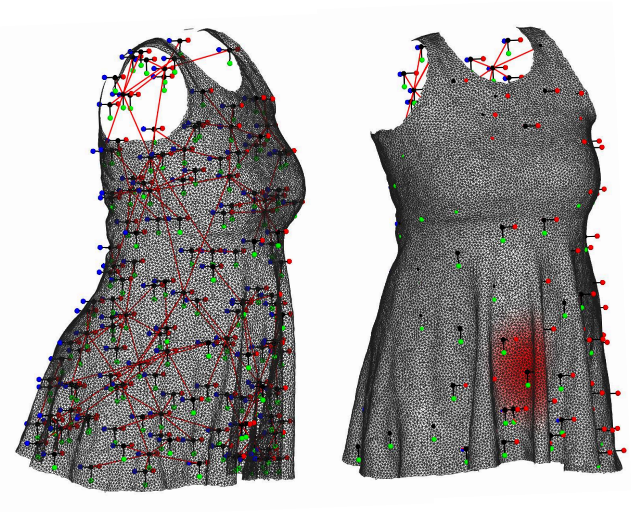


Figure 1: A visualization of the deformation graph $\mathcal{E}$ used in the dress example. On the left side, we show the coordinate frame at each graph node and their connectivity by the red lines. On the right side, the region of influence by a node located in the center is shown in red.

method can generate richer and more realistic dynamics for loose clothing than CCA, but DA requires a proprietary implementation of real-time cloth simulation. CCA and DA utilize ground truth clothing registration to train their models, while our method does not require such pre-processing. Finally, thanks to the additional RGB-D input and our model design, our output is more faithful to the actual clothing motion than those two previous methods.

5 IMPLEMENTATION DETAIL

5.1 Clothing Deformation Graph

In Fig. 1, we provide a visual illustration of the deformation graph $\mathcal{E}$ in the inner layer of the clothing deformation model $\mathcal{D}$ (Sec. 4 of the main paper) for the dress example. The parameters for the deformation graph include the rotation and translation for each node:

$$\boldsymbol{\theta} = \{\mathbf{r}_k, \mathbf{t}_k\}_{k=1}^{K}, \ \mathbf{r}_k, \mathbf{t}_k \in \mathbb{R}^3, \tag{1}$$

where $\mathbf{r}_k$ is the axis-angle representation of a 3D rotation. We use a total of $K = 125$ nodes for each example.

5.2 Training Setup

5.2.1 *N-ICP.*

When training the N-ICP module, we adopt a regularization term for deformation graph that compares the difference in transformation between adjacent nodes:

$$L_{\text{DG-Reg}} = \frac{1}{K(K-1)} \sum_{1 \le j \ne k \le K} \|T_j \mathbf{m}_{jk} - T_k \mathbf{m}_{jk}\|^2, \tag{2}$$

where T_j and T_k denote the SE(3) transformation for the $j-$th and $k-$th nodes respectively, and $\mathbf{m}_{ij}$ denotes the middle point between the rest positions of the $j-$th and $k-$th nodes. Then the total loss function for training N-ICP is written as

$$L_{\text{N-ICP}} = \frac{1}{N} \sum_{i=1}^{N} L_{\text{ICP}}(\boldsymbol{\theta}^{(i)}, \bar{\mathbf{P}}) + \lambda_{\text{DG-Reg}} L_{\text{DG-Reg}}, \tag{3}$$

where the balancing weight is set to $\lambda_{\text{DG-Reg}} = 1 \times 10^{-3}$. The trainable parameters in N-ICP are those in PointNet $\mathcal{M}$. The input and output of the PointNet $\mathcal{M}$ are converted to the root body coordinate of the subject given the tracked body pose $\boldsymbol{\rho}$ to be invariant to the global orientation and translation. We use the AdamW optimizer with an initial learning rate of 1×10^{-5}.

Initialization. We find it crucial to initialize the parameters in the last layer of the PointNet with values close to zero, so that $\boldsymbol{\theta}^{(i)} \approx \mathbf{0}$ for $i = 1, \ldots, N$ at the first training iteration, with $\boldsymbol{\theta}^{(0)}$ set to $\mathbf{0}$. In this way, thanks to the two-layer clothing deformation model (Sec. 4 in the main paper), $\mathcal{D}(\boldsymbol{\theta}^{(i)})$ is close enough to the ICP target to generate meaningful gradient at the beginning of the training process, and gradually converges to the desired minimum. In practice, we initialize the last layer of the network by random sampling from a uniform distribution $U[-\varepsilon, \varepsilon]$ where $\varepsilon = 1 \times 10^{-6}$.

Discussion on supervision. N-ICP is trained in a self-supervised manner, because the loss function $L_{\text{N-ICP}}$ does not involve the "ground truth" deformation parameters. The reasons are two-fold. First, it takes extra processing time efforts obtain the ground truth. Second, the problem of estimating reliable "ground truth" deformation parameters is challenging by itself. Unless the garment under capture has been specially designed to encode correspondences in a printed pattern [Halimi et al. 2022], otherwise, the principal approach is to run offline ICP between the deformation model and MVS geometry. In this way, the "ground truth" essentially offers no more information than directly supervising N-ICP by MVS. The self-supervised formulation, instead, allows solving a global optimization by sharing the information across all frames.

5.2.2 *Texel-Conditioned Clothed Avatars.*

We use the following loss functions to train the texel-conditioned clothed avatars (Sec. 5 of the main paper)

$$L_{\text{avatars}} = \sum_i \lambda_{\text{i}} L_{\text{i}}, \ i \in \{\text{RGB, mask, reg, part, ID-MRF}\}. \tag{4}$$

L_{RGB} and L_{mask} are the standard L_1 losses for RGB and mask respectively; L_{reg} is the Laplacian regularization terms for body and clothing meshes. L_{part} is similar to L_{mask} but identify background, body and clothing in three different categories. Following [Feng et al. 2022], we use the ID-MRF loss [Wang et al. 2018], a stronger form of perceptual loss to encourage sharpness for high-frequency texture in the clothing region. We use $\lambda_{\text{RGB}} = 0.2$, $\lambda_{\text{mask}} = \lambda_{\text{part}} = 500.0$, $\lambda_{\text{reg}} = 100.0$, $\lambda_{\text{ID-MRF}} = 1.0$. The gradient of loss functions defined in the image space (RGB, mask, part and ID-MRF) with respect to the network parameters are back-propagated through a differentiable rasterizer. We use the AdamW optimizer with a learning rate of 1×10^{-3}.

Color Augmentation. In order to deal with the domain gap in illumination and color when the directly applying the avatars to the novel capture environment (Sec. 6.4 in the main paper), we apply a random color augmentation to texel-aligned RGB features $\mathbf{F}_I$ using the 'ColorJitter' function in TorchVision[1] at training time. Notice that we leave the ground truth images used for supervision in L_{avatars} unchanged, so that the network always preserves the original appearance in the *output*, despite a different color mode in the *input* feature $\mathbf{F}_I$ when we direct apply the model to the novel environment. The output appearance only changes after fine-tuning with ground truth images in the novel environment.

5.3 Preprocessing and Postprocessing

5.3.1 Input Preprocessing. Our method takes RGB and depth images as input. When training and testing using data from the dense capture studio, we run image-based part segmentation and transfer the result to the MVS mesh by projection and visibility check. This operation allows us to extract the clothing region. The MVS mesh may include floating noise, which we remove by checking the mesh connectivity and setting a threshold on the minimal number of vertices in a connected component. Then, we rasterize the segmented mesh to RGB views to "simulate" a depth image.

When training and testing in the novel environment, we use the RGB-D images from calibrated Kinect sensors as input. We also run image-based part segmentation to extract the clothing regions. Then we use TSDF fusion [Curless and Levoy 1996] and Marching Cubes [Lorensen and Cline 1987] to form a mesh from the extracted depth images, which allows us to perform similar connectivity check as above to remove noise from the depth sensors.

5.3.2 Temporal Smoothing. Due to the unstructured point cloud input, the output of N-ICP may have undesirable jittering. We apply temporal smoothing to the output of N-ICP by taking the average on the vertex positions in a small temporal window, which is feasiable because the N-ICP output shares a consistent registered topology across all the frames. The filtered meshes are then used to unwrap texel-aligned features and as input to the texel-conditioned avatars as shown in Fig. 2 of the main paper. We find no need to apply additional smoothing on the final output of texel-conditioned avatars if the provided initial tracking is temporally stable and the floating depth noise has been removed in the preprocessing step.

5.3.3 Collision. To resolve the collision between the body and clothing layers, which is usually slight in the results, we follow Clothing Codec Avatars [Xiang et al. 2021] (Sec. 6) to project the clothing vertices in collision beyond the nearest body points by a slight margin. More sophisticated ways to handle collision based on geometry or learning [Tan et al. 2022] may be incorporated, which we leave for future work.

5.4 Network Architecture

5.4.1 N-ICP. N-ICP takes an unstructured point cloud as input, so we adopt the PointNet++ [Qi et al. 2017] architecture. To specify the architecture, we reuse the notation of Set Abstraction function from [Qi et al. 2017]:

$$\mathrm{SA}(K, r, [l_1, \ldots, l_d]),$$

where K denotes the number of grouping centers, r denotes the radius of the grouping regions, and l_i denotes the output size of a fully connected layer in the Multi-Layer Perceptron (MLP). We also denote a standalone MLP as $\mathrm{FC}([l_1, \ldots, l_d])$. Then the architecture of the network $\mathcal{M}$ can be described as

$$[\mathbf{p}, \mathbf{r}] \rightarrow \mathrm{SA}(32, 0.1, [16, 16, 32]) \rightarrow \mathrm{SA}(32, 0.2, [64, 64, 128]) \rightarrow$$
$$\mathrm{SA}(32, 0.4, [256, 256, 256]) \rightarrow \mathrm{SA}(32, 0.8, [256, 256, 512]) \rightarrow$$
$$\mathrm{MaxPool} \rightarrow \bigoplus \mathbf{g} \rightarrow \mathrm{FC}([512, 512, 512, 750]) \rightarrow \Delta\boldsymbol{\theta},$$

where $\mathbf{p}$ and $\mathbf{r}$ denote the point coordinate and residual as defined in Sec. 4 of the main paper, and $\bigoplus \mathbf{g}$ denotes the operation to concatenate the result from the previous step with gradient input $\mathbf{g}$.

5.4.2 Texel-Conditioned Clothed Avatars. The overall architecture of the texel-conditioned avatar models is shown in Fig. 2. Given the texel-aligned features $\mathbf{F}_I, \mathbf{F}_D$ unwrapped from the initial tracking results $\mathbf{D}$ as input, the encoder produces a feature map that is spatially aligned with the input. The encoded feature maps are then decoder into a vertex offset map $\delta\mathbf{G}$, from which the offsets are extracted and then applied on top of the initial tracking to obtain the output geometry $\mathbf{G}$. The geometry $\mathbf{G}$ and the viewpoint $\mathbf{v}$ are used together to compute the view-conditioning, including the viewing vector expressed in the local Tangent-Bitangent-Normal (TBN) frame [Xiang et al. 2022] as well as its reflected direction. The view-dependent U-Net takes in the view conditioning and the view-independent texture to produce an additive view-dependent offset. With the final geometry $\mathbf{G}$, we also compute the ambient occlusion, which is fed into the shadow U-Net to produce a multiplicative shadow map. The view-dependent texture is then upsampled to 2k resolution by a upsampling network.

To specify the architecture of the individual networks above, we define the blocks shown in Fig. 3.

(1) Convolutional encoder consists of the network blocks in the following table. Following DVA [Remelli et al. 2022], we find that using a U-Net at 64×64 resolution instead of a bottleneck structure helps to preserve the UV-space detail in the output.

Block	Output Size ($C \times H \times W$)
ConvBlock(6, 16, 1)	$16 \times 512 \times 512$
ConvDownBlock(16, 32, 1)	$32 \times 256 \times 256$
ConvDownBlock(32, 64, 1)	$64 \times 128 \times 128$
ConvDownBlock(64, 64, 1)	$64 \times 64 \times 64$
U-Net(64, 64, 32)	$32 \times 64 \times 64$

(2) View-independent decoder consists of the network blocks in the following table. Here, the "RepeatChannels" operation repeats the channels of the input feature for the geometry and texture

[1] https://github.com/pytorch/vision. We use the following parameters: brightness=0.5, contrast=0.5, saturation=0.5, hue=0.2.

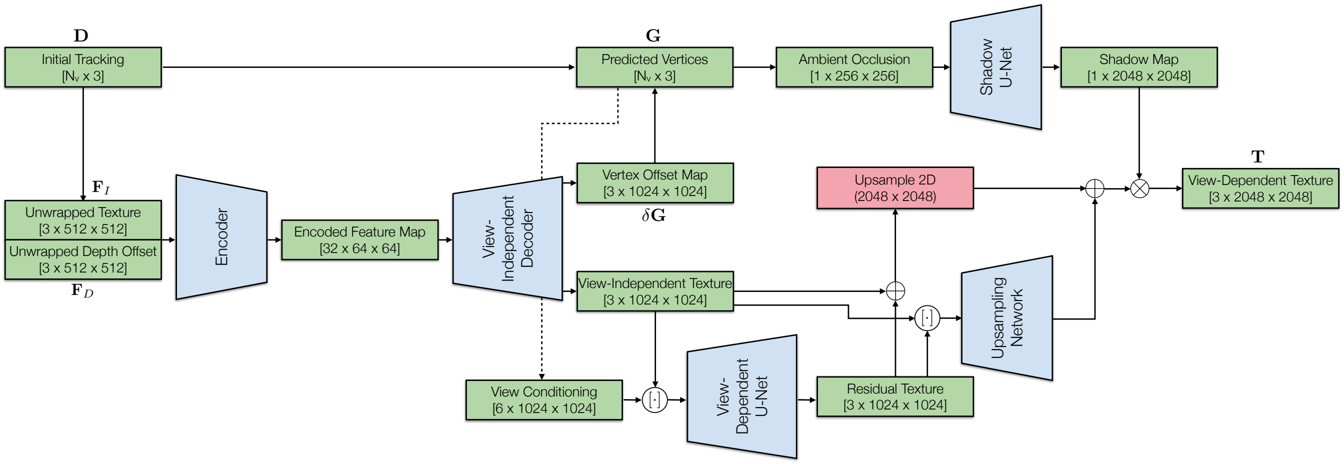


Figure 2: The network architecture for the texel-conditioned clothed avatars. It consists of the following five components: (1) a convolutional encoder that encodes the texel-aligned input, (2) a view-independent decoder that outputs vertex and texture maps, (3) a view-dependent U-Net that regresses view-dependent variation in the texture, (4) a shadow network that takes in ambient occlusion and computes a multiplicative shadow map, and (5) an upsampling network that predicts the residual after increasing the spatial resolution from 1024 to 2048.

branches. The following "ConvUpBlocks" processing them separately in different groups. The output is then evenly split into a vertex offset map and a texture map.

Block	Output Size ($C \times H \times W$)
ConvBlock(32, 32, 1)	$32 \times 64 \times 64$
RepeatChannels	$64 \times 64 \times 64$
ConvUpBlock(64, 32, 2)	$32 \times 128 \times 128$
ConvUpBlock(32, 16, 2)	$16 \times 256 \times 256$
ConvUpBlock(16, 8, 2)	$8 \times 512 \times 512$
ConvUpBlock(8, 8, 2)	$8 \times 1024 \times 1024$
Conv2D(8, 6, 2, k=1)	$6 \times 1024 \times 1024$
SplitChannels	$(2 \times)\ 3 \times 1024 \times 1024$

(3) View-dependent U-Net is a single block "U-Net(9, 4, 3)" defined in Fig. 3.

(4) Shadow U-Net is an upsampling operation on the input ambient occlusion map from 256 resolution to 2048, followed by a block "U-Net(1, 2, 1)".

(5) Upsampling network is defined in the following table. Here the "PixelShuffle(r)" is an operation that rearranges a tensor from shape $(C \times r^2) \times H \times W$ to $C \times (H \times r) \times (W \times r)$.

Block	Output Size ($C \times H \times W$)
Conv2D(6, 2, 1)	$2 \times 1024 \times 1024$
LReLU(0.2)	$2 \times 1024 \times 1024$
Conv2D(2, 12, 1)	$12 \times 1024 \times 1024$
PixelShuffle(2)	$3 \times 2048 \times 2048$

5.5 Training Data Preparation

In this section, we describe how we prepare the assets required to train the avatars. Given the multi-view images captured by more than one hundred synchronized cameras, we run 2D keypoint detection, part segmentation, and Multi-View Stereo (MVS). The 2D body keypoints are triangulated to estimate 3D keypoints. For each vertex in the MVS output, we aggregate its category label from each camera view by checking its projection in the image segmentation, and then perform a majority voting, followed by a Markov Random Field (MRF) to ensure spatial smoothness. We also use the method in [Zhang et al. 2017] to estimate an underlying body template and the body pose for each frame given the 3D keypoints and segmented MVS mesh. The whole process is similar to [Xiang et al. 2021], except that we do not perform clothing registration offline in the style of ClothCap [Pons-Moll et al. 2017]. Instead, we define a deformation model $\mathcal{D}(\boldsymbol{\theta})$, and train the N-ICP network to track the clothing in a self-supervised manner. The clothing template is created from the segmented clothing region in the MVS mesh in a rest-pose frame with some manual cleanup and remeshing.

REFERENCES

Brian Curless and Marc Levoy. 1996. A volumetric method for building complex models from range images. In *SIGGRAPH 1996 Conference Papers.*

Wei Dong, Yixing Lao, Michael Kaess, and Vladlen Koltun. 2022. ASH: A modern framework for parallel spatial hashing in 3D perception. *IEEE transactions on pattern analysis and machine intelligence* (2022).

Yao Feng, Jinlong Yang, Marc Pollefeys, Michael J Black, and Timo Bolkart. 2022. Capturing and Animation of Body and Clothing from Monocular Video. In *SIGGRAPH Asia 2022 Conference Papers.*

Marc Habermann, Weipeng Xu, Michael Zollhoefer, Gerard Pons-Moll, and Christian Theobalt. 2019. Livecap: Real-time human performance capture from monocular video. *ACM Transactions On Graphics (TOG)* 38, 2 (2019).

Marc Habermann, Weipeng Xu, Michael Zollhoefer, Gerard Pons-Moll, and Christian Theobalt. 2021. A Deeper Look into DeepCap. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2021).

Marc Habermann, Weipeng Xu, Michael Zollhofer, Gerard Pons-Moll, and Christian Theobalt. 2020. Deepcap: Monocular human performance capture using weak supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Oshri Halimi, Tuur Stuyck, Donglai Xiang, Timur Bagautdinov, He Wen, Ron Kimmel, Takaaki Shiratori, Chenglei Wu, Yaser Sheikh, and Fabian Prada. 2022. Pattern-Based Cloth Registration and Sparse-View Animation. *ACM Transactions on Graphics (TOG)* 41, 6 (2022).

Yue Jiang, Marc Habermann, Vladislav Golyanik, and Christian Theobalt. 2023. HiFECap: Monocular High-Fidelity and Expressive Capture of Human Performances. In *Proceedings of British Machine Vision Conference*.

Ruilong Li, Yuliang Xiu, Shunsuke Saito, Zeng Huang, Kyle Olszewski, and Hao Li. 2020. Monocular real-time volumetric performance capture. In *European Conference on Computer Vision*.

Zhe Li, Zerong Zheng, Hongwen Zhang, Chaonan Ji, and Yebin Liu. 2022. Avatarcap: Animatable avatar conditioned monocular human volumetric capture. In *European Conference on Computer Vision*.

William E Lorensen and Harvey E Cline. 1987. Marching cubes: A high resolution 3D surface construction algorithm. In *SIGGRAPH 1987 Conference Papers*.

Ricardo Martin-Brualla, Rohit Pandey, Shuoran Yang, Pavel Pidlypenskyi, Jonathan Taylor, Julien Valentin, Sameh Khamis, Philip Davidson, Anastasia Tkach, Peter Lincoln, et al. 2018. LookinGood: enhancing performance capture with real-time neural re-rendering. *ACM Transactions on Graphics (TOG)* 37, 6 (2018).

Gerard Pons-Moll, Sergi Pujades, Sonny Hu, and Michael J Black. 2017. ClothCap: Seamless 4D clothing capture and retargeting. *ACM Transactions on Graphics (TOG)* 36, 4 (2017).

Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. 2017. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in Neural Information Processing Systems* 30 (2017).

Edoardo Remelli, Timur Bagautdinov, Shunsuke Saito, Chenglei Wu, Tomas Simon, Shih-En Wei, Kaiwen Guo, Zhe Cao, Fabian Prada, Jason Saragih, et al. 2022. Drivable Volumetric Avatars using Texel-Aligned Features. In *SIGGRAPH 2022 Conference Papers*.

Nadia Robertini, Dan Casas, Helge Rhodin, Hans-Peter Seidel, and Christian Theobalt. 2016. Model-based outdoor performance capture. In *2016 International Conference on 3D Vision (3DV)*.

Shunsuke Saito, Zeng Huang, Ryota Natsume, Shigeo Morishima, Angjoo Kanazawa, and Hao Li. 2019. Pifu: Pixel-aligned implicit function for high-resolution clothed human digitization. In *Proceedings of the IEEE International Conference on Computer Vision*.

Shunsuke Saito, Tomas Simon, Jason Saragih, and Hanbyul Joo. 2020. PIFuHD: Multi-Level Pixel-Aligned Implicit Function for High-Resolution 3D Human Digitization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Qingyang Tan, Yi Zhou, Tuanfeng Wang, Duygu Ceylan, Xin Sun, and Dinesh Manocha. 2022. A Repulsive Force Unit for Garment Collision Handling in Neural Networks. In *European Conference on Computer Vision*.

Yi Wang, Xin Tao, Xiaojuan Qi, Xiaoyong Shen, and Jiaya Jia. 2018. Image inpainting via generative multi-column convolutional neural networks. *Advances in neural information processing systems* 31 (2018).

Donglai Xiang, Timur Bagautdinov, Tuur Stuyck, Fabian Prada, Javier Romero, Weipeng Xu, Shunsuke Saito, Jingfan Guo, Breannan Smith, Takaaki Shiratori, et al. 2022. Dressing Avatars: Deep Photorealistic Appearance for Physically Simulated Clothing. *ACM Transactions on Graphics (TOG)* 41, 6 (2022).

Donglai Xiang, Fabian Prada, Timur Bagautdinov, Weipeng Xu, Yuan Dong, He Wen, Jessica Hodgins, and Chenglei Wu. 2021. Modeling clothing as a separate layer for an animatable human avatar. *ACM Transactions on Graphics (TOG)* 40, 6 (2021).

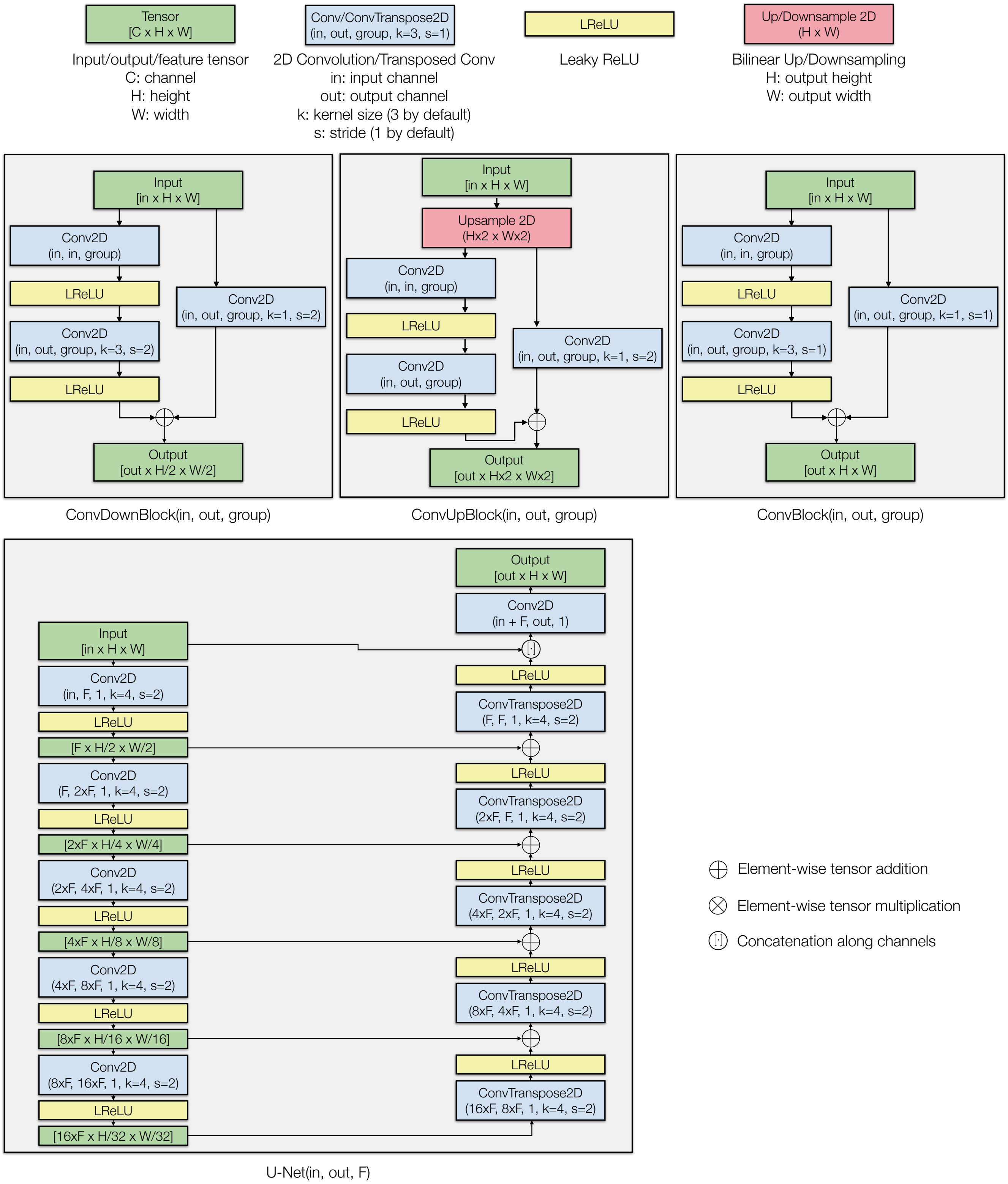
Donglai Xiang, Fabian Prada, Chenglei Wu, and Jessica Hodgins. 2020. MonoClothCap: Towards temporally coherent clothing capture from monocular RGB video. In *2020 International Conference on 3D Vision (3DV)*.

Yuliang Xiu, Jinlong Yang, Xu Cao, Dimitrios Tzionas, and Michael J Black. 2023. ECON: Explicit Clothed humans Optimized via Normal integration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Yuliang Xiu, Jinlong Yang, Dimitrios Tzionas, and Michael J Black. 2022. Icon: Implicit clothed humans obtained from normals. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Weipeng Xu, Avishek Chatterjee, Michael Zollhöfer, Helge Rhodin, Dushyant Mehta, Hans-Peter Seidel, and Christian Theobalt. 2018. Monoperfcap: Human performance capture from monocular video. *ACM Transactions on Graphics (TOG)* 37, 2 (2018).

Chao Zhang, Sergi Pujades, Michael J. Black, and Gerard Pons-Moll. 2017. Detailed, Accurate, Human Shape Estimation From Clothed 3D Scan Sequences. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.



ConvDownBlock(in, out, group)

ConvUpBlock(in, out, group)

ConvBlock(in, out, group)

U-Net(in, out, F)

Figure 3: Network blocks used in the architecture of texel-aligned avatars.