

DressRecon: Freeform 4D Human Reconstruction from Monocular Video

Supplementary Material

6. Video Results

Please see the [attached webpage](#) for video results.

7. Implementation Details

7.1. Consistent 4D Neural Fields

Signed distance fields. We initialize canonical signed distance fields as a sphere with radius 0.1m. Following standard practice, we apply positional encodings to all 3D points ($L_{xyz} = 10$) and timestamps ($L_t = 6$) before passing into MLPs. The appearance code ω_t has 32 channels.

After MLP_{SDF} computes the signed distance d at a 3D point, we convert the signed distance to a volumetric density $\sigma \in [0, 1]$ for volume rendering. Similar to VolSDF [71], this is done using the cumulative Laplace distribution $\sigma = \Gamma_\beta(d)$, where β is a global learnable scalar parameter that controls the solidness of the object, approaching zero for solid objects. This representation allows us to extract a mesh as the zero level-set of the SDF.

Cycle consistency regularization. Given a forward warping field $\mathcal{W}^+(t) : \mathbf{X} \rightarrow \mathbf{X}_t$ and a backward warping field $\mathcal{W}^-(t) : \mathbf{X}_t \rightarrow \mathbf{X}$, we introduce a cycle consistency term, similar to NSFF [35]. A sampled 3D point in camera coordinates should return to its original location after passing through a backward and forward warping:

$$\mathcal{L}_{\text{cyc}} = \sum_{\mathbf{X}_t} \|\mathcal{W}^+(\mathcal{W}^-(\mathbf{X}_t, t), t) - \mathbf{X}_t\|_2^2 \quad (16)$$

7.2. Hierarchical Gaussian Motion Fields

Bag-of-bones skinning deformation. Our motion model uses the motion of B bones (defined as 3D Gaussians, typically $B = 25$) to drive the motion of canonical geometry. Given 3D Gaussians, we compute dense 3D motion fields by blending the $\text{SE}(3)$ transformations of canonical Gaussians with skinning weights \mathbf{W} :

$$\mathbf{X}_t = \mathcal{W}^+(\mathbf{X}, t) = \left(\sum_{b=1}^B \mathbf{W}^{+,b} \mathbf{G}_t^b (\mathbf{G}^b)^{-1} \right) \mathbf{X} \quad (17)$$

$$\mathbf{X} = \mathcal{W}^-(\mathbf{X}_t, t) = \left(\sum_{b=1}^B \mathbf{W}_t^{-,b} \mathbf{G}^b (\mathbf{G}_t^b)^{-1} \right) \mathbf{X}_t \quad (18)$$

where \mathbf{G}_t^+ are forward warps from canonical to time t Gaussians, \mathbf{G}_t^- are backward warps from time t to canonical Gaussians, and \mathbf{W}^+ are forward skinning weights.

Similar to SCANimate [49] and LASR [66], we define a forward skinning weight function $\mathcal{S}^+ : \mathbf{X} \rightarrow \mathbb{R}^B$ which

computes the normalized influence of each Gaussian bone on a canonical 3D point. At a coarse level, skinning weights are defined as the Mahalanobis distance from \mathbf{X} to the canonical Gaussians:

$$\mathbf{W}_\sigma^+ = (\mathbf{X} - \boldsymbol{\mu})^\top \mathbf{Q} (\mathbf{X} - \boldsymbol{\mu}), \quad (19)$$

where $\boldsymbol{\mu} \in \mathbb{R}^{B \times 3}$ are canonical bone centers, $\mathbf{Q} = \mathbf{V}^\top \boldsymbol{\Lambda} \mathbf{V}$ are canonical bone precision matrices, $\mathbf{V} \in \mathbb{R}^{B \times \text{SO}(3)}$ are canonical bone orientations, and $\boldsymbol{\Lambda}^{B \times 3 \times 3}$ are time-invariant axis-aligned diagonal scale matrices.

In addition to a coarse component, we find it helpful to use delta skinning weights to model fine geometry. Delta skinning weights are computed by a coordinate MLP:

$$\mathbf{W}_\Delta^+ = \text{MLP}_{\Delta,+}(\mathbf{X}, t) \in \mathbb{R}^B \quad (20)$$

The final skinning function is a normalized sum of coarse and fine components:

$$\mathbf{W}^+ = \mathcal{S}^+(\mathbf{X}, t) = \text{softmax}(-\mathbf{W}_\sigma^+ - \mathbf{W}_\Delta^+), \quad (21)$$

where the negative sign ensures that faraway Gaussian bones (which have a larger Mahalanobis distance) are assigned a lower skinning weight after softmax.

Backward skinning weights are computed analogously with the time t Gaussians, which have center $\boldsymbol{\mu}_t$, orientation \mathbf{V}_t , and time-invariant scale $\boldsymbol{\Lambda}$. We also need the transformation \mathbf{G}_t^- from each time t Gaussian to the canonical Gaussian, as well as the backward skinning MLP_{Δ}^- .

7.3. Optimization

Sampling. Due to the expensive per-ray computation in volume rendering, optimization with batch gradient descent is challenging. As a result, previous methods randomly sample entire images [69] to compute the reconstruction terms, leading to small batch sizes (typically 16 images per batch) and noisy gradients. We implement an efficient data-loading pipeline with memory-mapping that allows per-pixel measurements (e.g., RGB, flow, features) to load directly from disk without accessing the full image. This allows loading pixels from significantly more images in a single batch (e.g. 256 images on a GPU).

Hyperparameters. We use the AdamW optimizer with learning rate 0.0005. We use 48k iterations of optimization for all experiments. On a single RTX 4090 GPU, it takes about 8 hours to optimize the neural implicit body model and 15 seconds to render each frame. 3D Gaussian refinement is performed for another 48k iterations of optimization, taking about 8 hours to optimize and 0.1 seconds to

Table 7. **Summary of losses and loss weights.** Our final loss is a weighted sum of reconstruction terms (color, optical flow, normal, feature, and segmentation) and regularization terms (eikonal, cycle-consistency, gaussian consistency, camera prior, and joint prior).

Loss	Weight	Description
\mathcal{L}_c	$\lambda_c = 0.1$	L2 loss, rendered RGB vs. the input image
\mathcal{L}_f	$\lambda_f = 0.5$	L2 loss, rendered 2D flow vs. computed flow from VCNPlus [65]
\mathcal{L}_n	$\lambda_n = 0.03$	L2 loss, rendered normals vs. computed normals from Sapiens [31]
\mathcal{L}_ϕ	$\lambda_\phi = 0.01$	L2 loss, rendered features vs. computed features from DINOv2 [45]
\mathcal{L}_s	$\lambda_s = 0.3$	L2 loss, rendered masks vs. computed masks from SAM [32]
\mathcal{L}_{eik}	$\lambda_{\text{eik}} = 0.001$	Encourage numerical gradients of canonical SDF to have unit norm
\mathcal{L}_{cyc}	$\lambda_{\text{cyc}} = 0.05$	Encourage backward and forward warping fields to be inverses
$\mathcal{L}_{\text{gauss}}$	$\lambda_{\text{gauss}} = 0.2$	Sinkhorn divergence between canonical 3D Gaussians and SDF
$\mathcal{L}_{\text{cloth}}$	$\lambda_{\text{cloth}} = 0.1$	Minimize the magnitude of clothing deformation

render each frame. We attribute the long training times to the use of numerical normal loss, which requires computing finite differences, as well as the use of large MLPs and ray-marching for geometry modeling and neural skinning. We leave speeding up as future work.

Our loss weights are described in Tab. 7. Loss weights λ are searched once and kept across all experiments. At each iteration, we sample 96 images and take 16 pixel samples per image. For training efficiency, input images are cropped to a tight bounding box around the object and resized to 256x256. To prevent floater artifacts from appearing outside the tight crop, 90% of pixel samples are taken from the tight bounding box and 10% of pixel samples are taken from the full un-cropped image.