HVH: Learning a Hybrid Neural Volumetric Representation for Dynamic Hair Performance Capture - Supplementary Material

1. Appendix

1.1. Dataset and Capture system

We use a multi-camera system with around 100 synchronized color cameras that produces 2048×1334 resolution images at 30 Hz. The cameras are focused at the center of the capture system and distributed spherically at a distance of one meter to provide as many viewpoints as possible. Camera intrinsics and extrinsics are calibrated in an offline process. We captured three sequences of different hair styles and hair motions. In the first sequence, we have one actor with a short high pony tail performing nodding and rotating. In the second sequence, we have one actor with a curly long releasing style hair and leaning her head towards four directions(left, right, up and down) and rotating. In the third sequence, we have one actor with a long high pony tail performing nodding and rotating.

1.2. Diversity in hair styles

Given that the main focus of this work is dynamic hair capture and tracking, we selected several hairstyles with a certain level of diversity, like long curly open hair, mid-length fluffy straight pony tail, and long curly pony tail, that exhibit complex dynamic behavior where hair does not move rigidly with the head—hence are particularly well-suited for analyzing the performance of the proposed approach. Regarding generalization, the 3D scene flow formulation and the hair decoder are agnostic to specific hair structure and color; the hair tracking algorithm depends on artist prepared guide strands and, together with the optical flow, requires sufficient contrast for hair strands and background. Given its strand-based nature, our method might not be suitable for specific hairstyles like buzz cut or afro-textured hair, where it is challenging to create the initialization of the strands. However, we want to point out that our 3D scene flow formulation, which is agnostic to hair style, alone already improves MVP (as shown in the experiments).

1.3. Baselines

We compare against several volume-based or implicit function based baseline methods [3,5,7] for spatio-temporal modeling.

MVP [5] presents an efficient 4D representation for dynamic scenes with humans which is capable of doing animation and novel view synthesis. It combines explicitly tracked head mesh with volumetric primitives to model the human appearance and geometry with better completeness. The volumetric primitives can be aligned onto an unwrapped 2D UV-map from a tracked head mesh and can be regressed from a 2D convolutional neural network that leverages shared spatially computation. Similar to Neural Volumes [4], a differentiable volumetric ray marching algorithm is designed to render 2D rgb images on MVP in real time. We use $N_p = 4096$ volumetric primitives with a voxel resolution $8 \times 8 \times 8$ on each sequence with a ray marching step size around $dt = 1mm$. We use a global latent size of 256.

Non-rigid NeRF [7] presents an implicit function based representation for dynamic scene reconstruction and novel view synthesis based on NeRF [6]. It utilizes a hierarchical model by disentangling a dynamic scene into a canonical frame NeRF and its corresponding deformation field which is parameterized by another MLP. In our experiments, we use 128 sampling points for both coarse and fine level sampling. We use the original implementation from the authors here. We train different models for each sequences and each model is trained for at least 300k iterations until convergence.

NSFF [3] is another implicit function based representation for dynamic scenes that is also based on NeRF [6]. It learns a per-frame NeRF that is additionally conditioned on the time index. It brings optical flow as additional supervision and learns a 3D scene flow in parallel with the per-frame NeRF for enforcing temporal consistency. NSFF is able to perform both spatial and temporal interpolation on a given video sequence. We use a setting of 256 sampling points in our experiments, using [2] as a substitute for generating optical flow. We use the original implementation from the authors here. We train different models for each sequences and each model is trained for at least 300k iterations until convergence.

1.4. Training Details

For both tracking optimization and HVH training, We deploy Adam [1] for optimization. For hair tracking, we use a learning rate of 1. We set the weighting coefficients of each losses as $\omega_{hdir} = 3$, $\omega_{hpos} = 1$, $\omega_{len} = 3$, $\omega_{tang} = 3$ and $\omega_{cur} = 1e4$. For each time step, 100 iterations are taken for optimization to solve the possible hair strands at next frame out. For HVH, we set weighting parameters for each objective as $\lambda_{flow} = 1$. $\lambda_{geo} = 0.1$, $\lambda_{vol} = 0.01$, $\lambda_{cub} = 0.01$ and $\lambda_{KL} = 0.001$. All models are trained with approximately 100-150k iterations. We use a latent code size of 256 and per-strand hair code size of 256, raymarching step size around $dt = 1mm$ and around $N_p = 5500$ volumetric primitives with a voxel resolution $8 \times 8 \times 8$ for each sequence depending on the number of guide hairs. For each sequence, we have roughly 30 strands for guide hair and we sample 50 points on each strands.

1.5. Novel View Synthesis

We show a larger version of comparison figure between different methods in Figure 1. For completeness, we also include visualizations from a perframe NeRF model which takes a perframe temporal code as input liker non-rigid NeRF [7].

1.6. Ablation Studies

Temporal Consistency. We show a bigger version of rendering results on unseen sequence in Figure 2.

Hair Decoder structure. As part of the hair decoder ablation, we compare our method with a naive decoder that uses the same volume decoder as MVP [5] for hair volumes. There are two major differences: 1) the naive decoder does not take the per-strand hair feature as input; 2) The design of the naive decoder does not take into account the hair specific structure where it regresses the same slab as for head tracked mesh and we take the first N_{hair} volumes as the output. In this way, the naive decoder discards all intrinsic geometric structural information while doing convolutions in each layers. We show the hair volumes layout in Figure 3. In the naive design, the hair strands are randomly squeezed into a square UV-map which could break the inner connections of each hair. In our design, we groom the hair strands into the their directions which could preserve the hair specific geometric structure. We compare different designs of decoder on Seq01. As in Table 1, our hair structure aware decoder produces a smaller image reconstruction error and better SSIM, a result of inductive bias of the designed hair decoder.

We additionally compare two different designs of the hair decoder where we do late and early fusion of the per-strand hair feature and the global latent feature. We show two different designs in Figure 4. Table 1 shows that the late fusion model performs better than early fusion model. This could be because the late fusion model transfers the 1d global latent code into a spatially varying feature tensor which is a more expressive form of feature representation.

decoder	MSE	SSIM	PSNR
naive	45.68/75.15	0.9549/0.9220	31.83/29.54
early fus.	43.75/71.08	0.9533/0.9259	31.97/29.82
late fus.	41.89/65.96	0.9543/0.9280	32.17/30.09

Table 1. **Decoder structure.** We compare different designs of the hair decoder. We report all metrics on both training and testing and we use a to separate them where on the left are the results of novel synthesis on training sequence.

1.7. Visualization of Flow

Please see Figure 5 for a visualization of the rendered flow from our representation. Compared to the optical flow from [2], our rendered 2D flow has less noise on the background. This is because that we only define our 3D scene flow on the volumetric primitives instead of the whole space. With the help of the coarse level geometry like the hair strands and head tracked mesh, the scene flow of most part of the empty space will naturally be zero. This could help us eliminate the noise from the background optical flow to certain degree.

Run Time Analysis. We report the rendering time of one iamge at resolution 1024 × 667 for each methods here. MVP [5] takes 0.223s. Ours takes 0.254s. NSFF takes 28.68s. NRNeRF [7] takes 41.29s. All tests are conducted under a single Nvidia Tesla V100 GPU.

1.8. Hair Tracking Analysis

In Figure 6, we plot different hair properties over time. We report four different metrics describing how well the tracked hairs fit the per-frame reconstruction and how well it preserves its length and curvature. In the first two rows, we report the MSE between the tracked hair and the tracked hair at first frame in terms of curvature and length. In the last two rows, we report the cosine distance between the direction of each nodes on the tracked guide hair and the direction of its neighbor from the reconstruction and the Chamfer distance between the tracked guide hair nodes and the reconstruction. As we can see the length and curvature are relatively preserved across frames and the affinity between the per-frame reconstruction and the tracked guide hair is relatively high.

References

[1] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 2

[2] Till Kroeger, Radu Timofte, Dengxin Dai, and Luc Van Gool. Fast optical flow using dense inverse search. In *European*

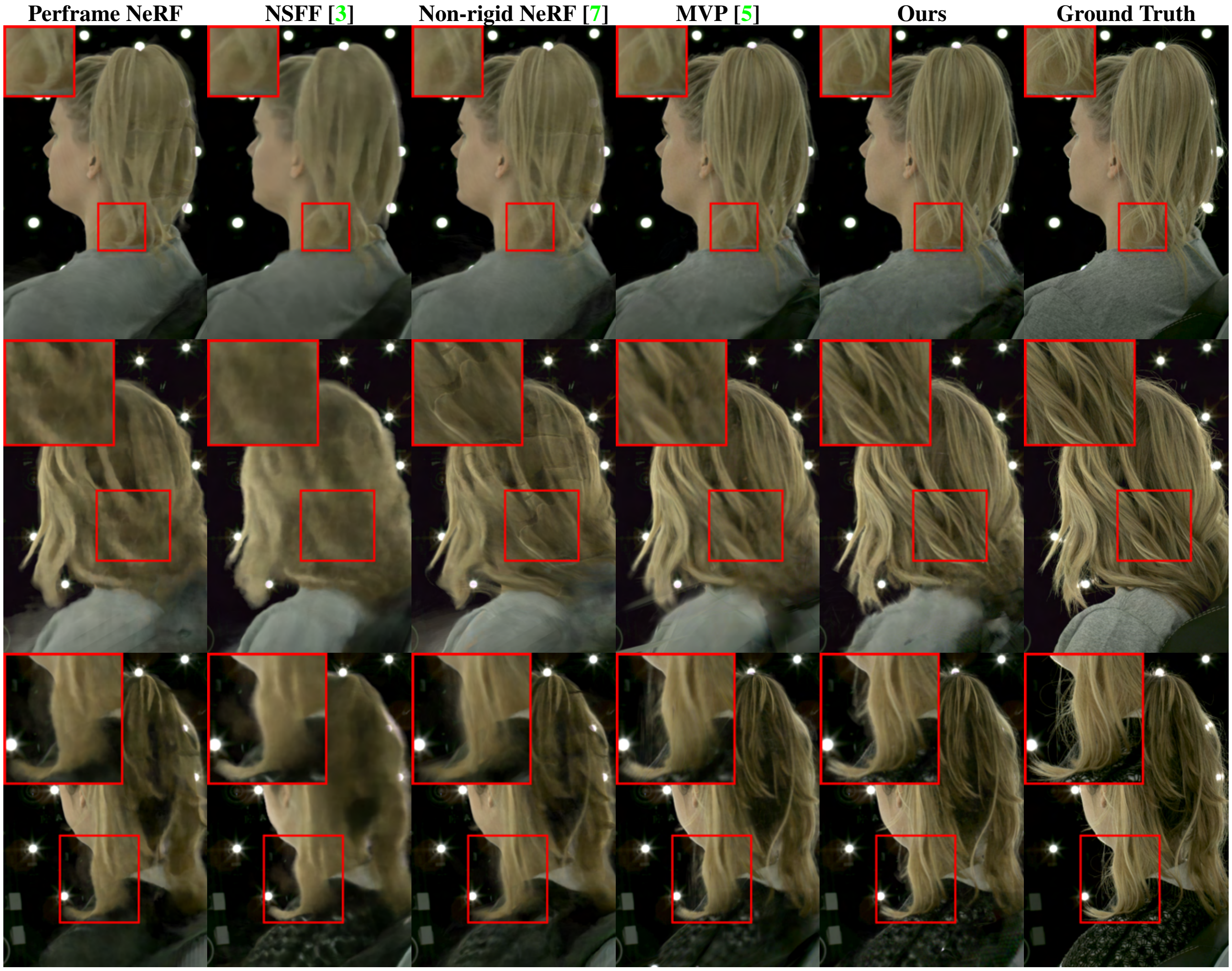

Figure 1. **Comparison on novel view synthesis between different methods.**

Conference on Computer Vision, pages 471–488. Springer, 2016. 1, 2, 5

[3] Zhengqi Li, Simon Niklaus, Noah Snavely, and Oliver Wang. Neural scene flow fields for space-time view synthesis of dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6498–6508, 2021. 1, 3

[4] Stephen Lombardi, Tomas Simon, Jason Saragih, Gabriel Schwartz, Andreas Lehrmann, and Yaser Sheikh. Neural volumes: Learning dynamic renderable volumes from images. *ACM Transactions on Graphics (TOG)*, 38(4), July 2019. 1

[5] Stephen Lombardi, Tomas Simon, Gabriel Schwartz, Michael Zollhoefer, Yaser Sheikh, and Jason Saragih. Mixture of volumetric primitives for efficient neural rendering. *ACM Transactions on Graphics (TOG)*, 40(4), July 2021. 1, 2, 3, 4

[6] 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 on computer vision*, pages 405–421. Springer, 2020. 1

[7] Edgar Tretschk, Ayush Tewari, Vladislav Golyanik, Michael Zollhöfer, Christoph Lassner, and Christian Theobalt. Non-rigid neural radiance fields: Reconstruction and novel view synthesis of a dynamic scene from monocular video. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12959–12970, 2021. 1, 2, 3

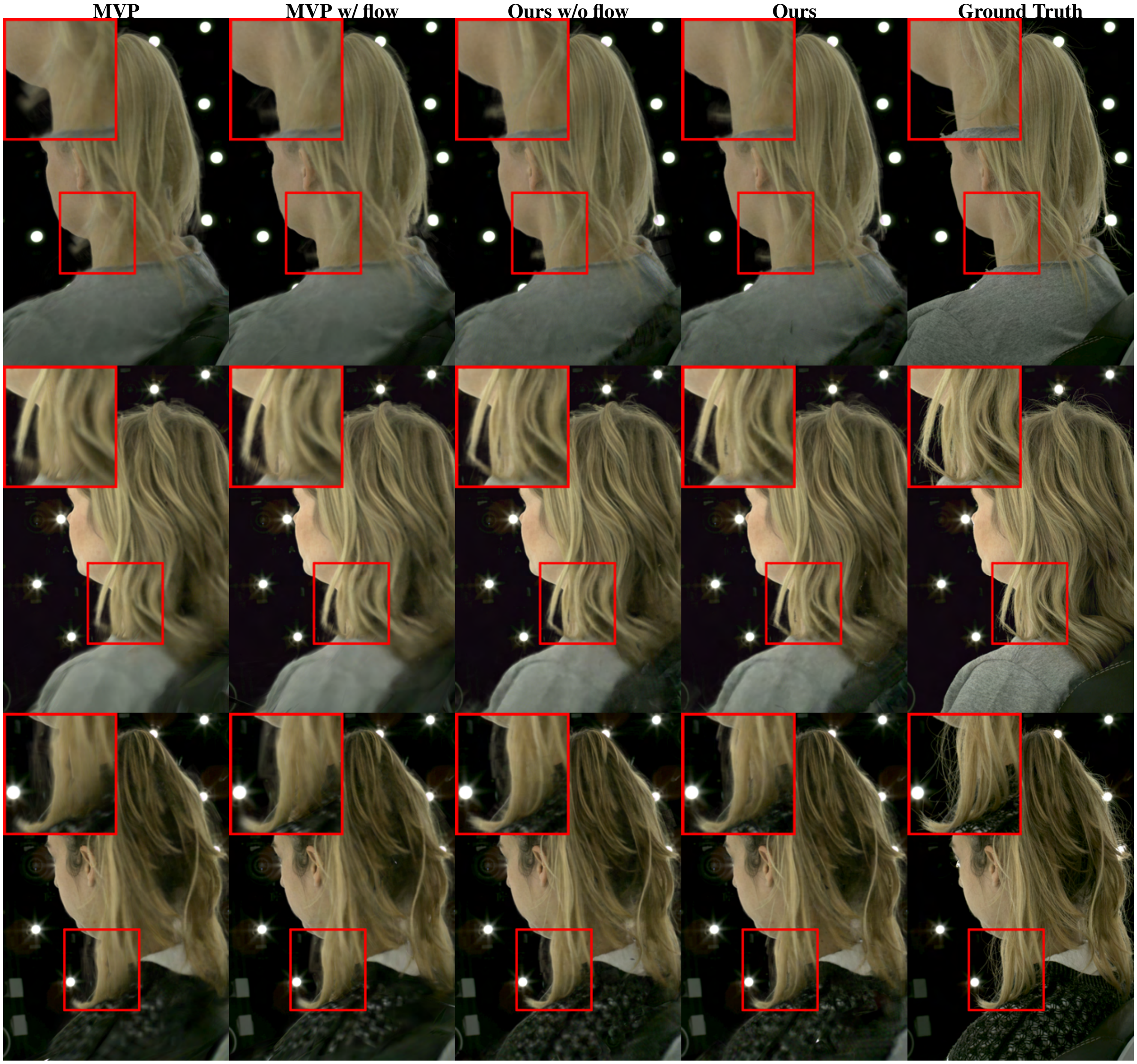

Figure 2. **Ablation of temporal consistency.** We compare MVP [5] and ours with different variations.

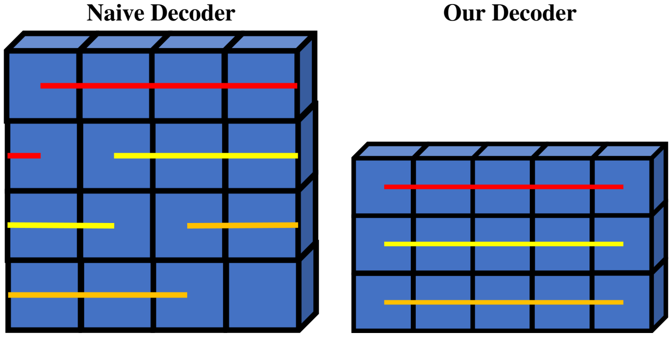

Figure 3. **Hair volumes layout.** We show the hair volume layout of both naive decoder and ours.

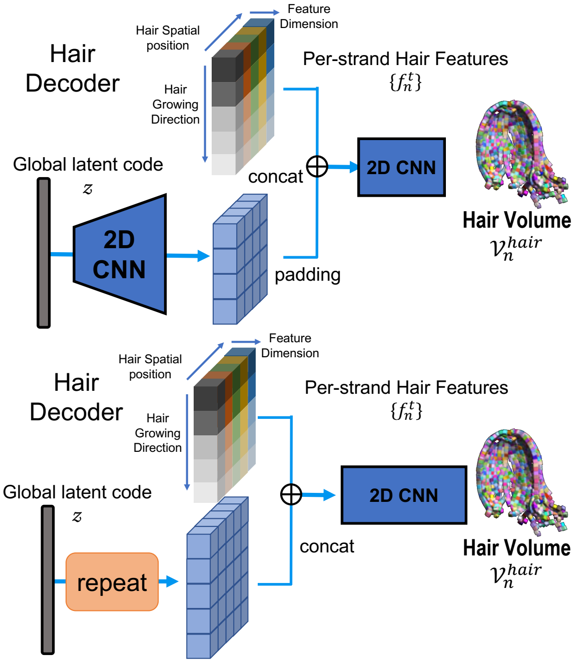

Figure 4. **Architecture of the hair decoder.** We show late fusion on the top and early fusion on the bottom. The late fusion model first deconvolves the 1D global latent code into a 2D feature map and then concatenate it with the per-strand hair features. A 2D CNN is used afterwards to generate the hair volumes. The early fusion model first repeat the 1D global latent vector spatially and then concatenate the repeated feature map with per-strand hair features. The concatenated features are than fed into a deeper 2D CNN to generate the hair volumes.

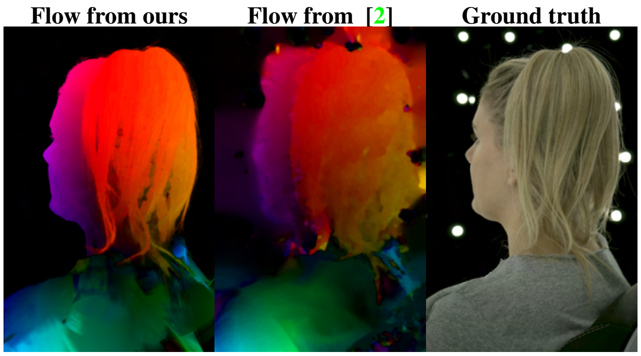

Figure 5. **Visualization of flow.** We show the rendered 3D scene flow into 2D flow in the first column and the openCV optical flow [2] in the second column. The last column shows the ground truth image as reference.

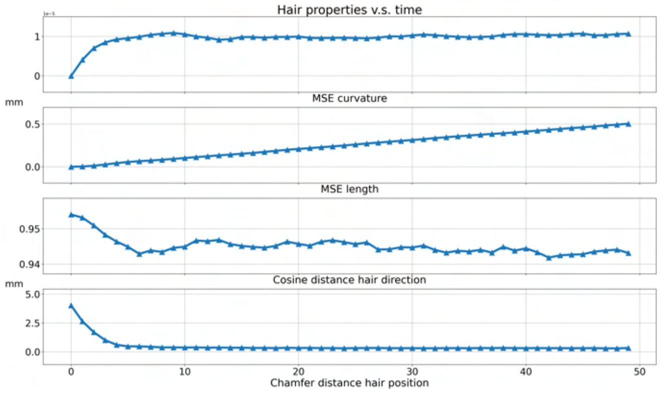

Figure 6. **Plot of tracked hair properties v.s. time.** As we can see, the hair properties like length and curvature are not changing too much across time and hair Chamfer distance are relatively small.