Supplementary Material for
OmnimatteRF: Robust Omnimatte with 3D Background Modeling

A. Additional Qualitative Results

Video files of results presented in the main paper (all videos from the `Movies`, `Kubrics`, `Wild`, and `DAVIS` datasets) are available on our project website as part of the supplementary material. We highly recommend watching them on: `https://omnimatte-rf.github.io`

1. For our method (OmnimatteRF), we include results (inputs with masks, foreground layers, background layer, background depth map) for every video.
2. For D^{2}NeRF [5], we use the best result among all configurations provided by the authors for every video. If none of the configurations successfully reconstruct non-empty static and dynamic layers, we drop the video files and only show a frame in Fig. A2.
3. For Omnimatte [3] and Layered Neural Atlas (LNA) [2], we include videos from `Wild`, `Movies`, and `Kubrics`. Results of `DAVIS` can be found in prior works.

B. Random Initialization

As is also discussed in Omnimatte [3], different random initializations can lead to varying results of the foreground layers. We show two examples in Fig. A1.

In all our experiments, the random seed is set to 3.

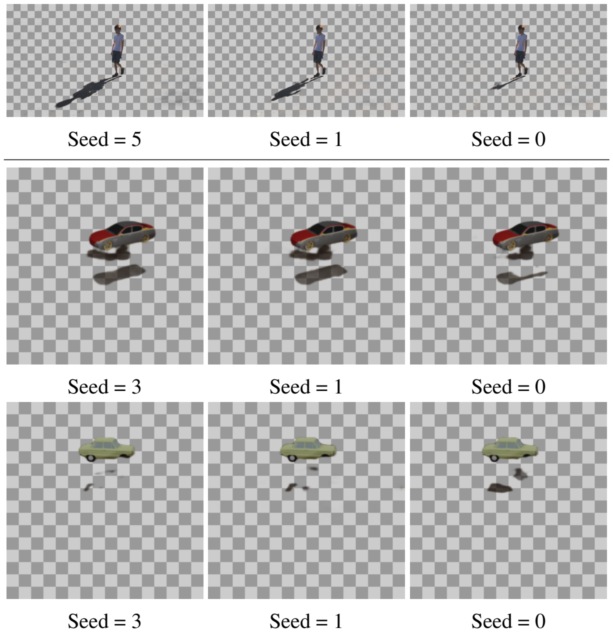

Figure A1. **Effect of random initialization.** Top: for the `Wild/dogwalk` video, different seeds lead to different amount of hallucinated shadow of the person. Bottom: for the `Kubrics/cars` video, seeds influence how shadows are associated to the objects.

C. Additional Implementation Details

C.1. Mask Generation

Our method and Omnimatte relies on coarse mask videos that outlines every object of interest. The synthetic `Kubrics` and `Movies` videos have ground truth object masks and we use them directly. To obtan mask for an in-the-wild video, we use one of the two workflows:

1. We first process the video a the pretrained Mask R-CNN model (`X101-FPN`) from Detectron 2 [6]. Then, we manually select a mask in every frame that best capture the object.
2. We use the Roto Brush tool in Adobe After Effects to track the object. This method is useful when Mask R-CNN fails produce good masks for a video. In particular, we processed `Wild/dance` and `Wild/solo` manually.

It takes about 10 minutes of manual work to generate a mask sequence for a 200-frame video.

C.2. Network Architecture

Our foreground network is based on the U-Net architecture of Omnimatte, which is detailed in their supplementary [3]. To adopt their network to OmnimatteRF, we replace the background noise input by the 2D feature map E_t. Each pixel in E_t is the positional encoding of the 3D vector (x, y, t) where (x, y) is the pixel location and t is the frame number. The positional encoding scheme is the same as proposed in NeRF [4], with $L = 10$ frequencies.

Method	Steps	Training (hours)	Rendering (s/image)
Omnimatte	12,000	2.7	2.5
D^{2}NeRF	100,000	4.5	4.8
LNA	400,000	8.5	0.40
Ours	15,000	3.8	3.5
Omnimatte	12,000	1.2	0.95
D^{2}NeRF	100,000	3.2	2.2
LNA	400,000	8.5	0.21
Ours	15,000	2.5	1.2

Table A1. **Running Time Measurement.** We measure and compare the time it takes to train F2B3 and baseline methods. **Top**: `Movies`, `Wild` (480 × 270, `DAVIS` has a similar resolution of 428 × 240). **Bottom**: `Kubrics` (256 × 256).

For background, we use the Vector-Matrix decomposition model in TensoRF [1] with the MLP feature decoder. Our initial grid has the same resolution $N_0 = 128$, and the final grid is limited to $N = 640$. The vectors and matrices are upsampled at steps 2000, 3000, 4000, 5500.

C.3. Hyper-parameters

For all videos, we use a learning rate of 0.001 for the foreground network, which is exponentially decayed from the 10,000 step at a rate of $0.1\times$ per 10,000 steps. We find the decay crucial in preventing the foreground training from diverging. The mask bootstrapping loss $\mathcal{L}_{\text{mask}}$ has an initial weight of 50, which is first reduced to 5 when the loss value (before weighting) drops to below 0.02, and then turned off after the same number of steps. We document weights of other loss terms in Table A2.

Background network learning rate scheduling and $\mathcal{L}_{\text{bg-reg}}$ weight are identical as the original TensoRF [1].

In general, we use the same set of hyper-parameters for most videos, and only add additional terms when artifacts are observed.

C.4. Running Time Measurement

We measure and report the time it takes to train OmnimatteRF and baseline methods in Table A1. All measurements are conducted on a workstation with an eight-core AMD R7-2700X CPU and a single NVIDIA RTX3090 GPU.

Our method takes longer to train than Omnimatte due to the addition of the 3D background radiance field.

Optimizing the background model only, as in the clean background retraining process, takes about 30 minutes per video.

References

[1] Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. Tensorf: Tensorial radiance fields. In *European Conference on Computer Vision (ECCV)*, 2022. 2

[2] Yoni Kasten, Dolev Ofri, Oliver Wang, and Tali Dekel. Layered neural atlases for consistent video editing. *ACM Transactions on Graphics (TOG)*, 40(6):1–12, 2021. 1

[3] Erika Lu, Forrester Cole, Tali Dekel, Andrew Zisserman, William T Freeman, and Michael Rubinstein. Omnimatte: Associating objects and their effects in video. In *CVPR*, 2021. 1

[4] 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*, 2020. 1

[5] Tianhao Wu, Fangcheng Zhong, Andrea Tagliasacchi, Forrester Cole, and Cengiz Oztireli. D^{2}nerf: Self-supervised decoupling of dynamic and static objects from a monocular video. In *Advances in Neural Information Processing Systems*, 2022. 1

[6] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. `https://github.com/facebookresearch/detectron2`, 2019. 1

Video	Steps	$\mathcal{L}_{\text{recons}}$	$\mathcal{L}_{\alpha\text{-reg}}$	$\mathcal{L}_{\alpha\text{-warp}}$	$\mathcal{L}_{\text{flow}}$	$\mathcal{L}_{\text{depth}}$	$\mathcal{L}_{\text{distort}}$
All	15,000	1	0.01 (L1) / 0.005 (L0)	0.01	1	0	0
Wild/bouldering	-	-	-	-	-	0.1	0.01
`DAVIS`	10,000	-	-	-	-	1	0

Table A2. **Hyper-parameters.** We document the hyper-parameters (number of steps and weights of loss terms) in our experiments. The first row is the configuration shared by most videos. Remaining rows are videos with different configurations, and - means unchanged from the shared number.

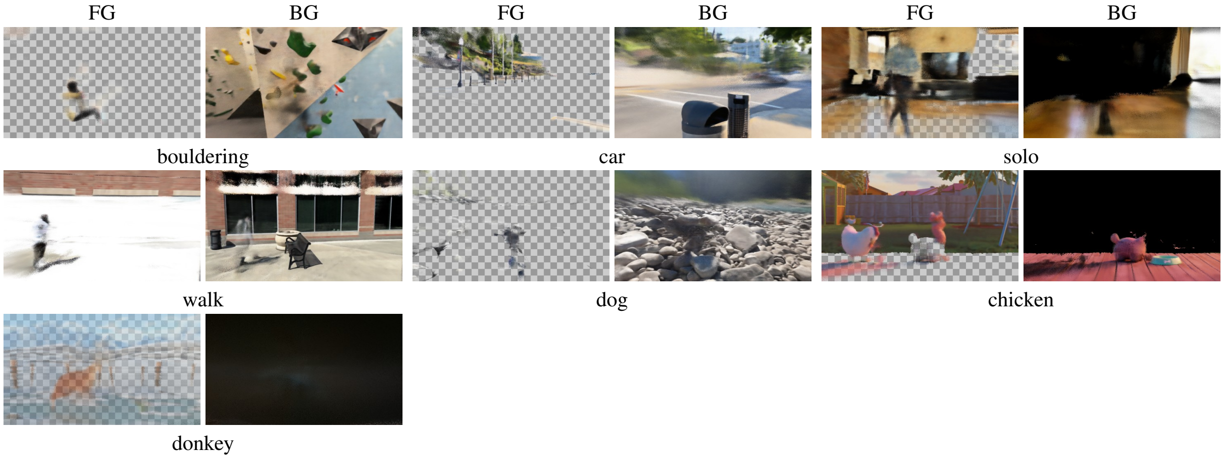

Figure A2. **D^{2}NeRF results for failed scenes.**