Supplementary Material for: Omnimatte360: Associating Objects and their Effects in Unconstrained Monocular Video

A. Network Architectures (Sec. 3.1)

Table A.1 the network architecture for the UNets use in our model. Each Resblock consist of two 3×3 convolution layer with swish activations and a skip connection between the input and output. A Downsample / Upsample Resblock has an additional downsampling / upsampling layer before the initial convolution.

Network Blocks	Out Channels
3x3 Conv	4
Resblock x 3	4
Downsample Resblock	4
Resblock x 3	8
Downsample Resblock	8
Resblock x 3	16
Downsample Resblock	16
Resblock x 3	32
Downsample Resblock	32
Resblock x 3	32
Resblock	32
Resblock	32
Skip-Resblock x 3	32
Upsample Resblock	32
Skip-Resblock x 3	32
Upsample Resblock	32
Skip-Resblock x 3	16
Upsample Resblock	16
Skip-Resblock x 3	8
Upsample Resblock	8
Skip-Resblock x 3	4

Table A.1. Architecture for the UNets used in our model.

B. Loss hyperparameters (Sec. 3.3)

We use the following weights for the loss term:

- $\lambda_1 = 1.0$ for the reconstruction loss $\mathcal{L}_{recon}$
- $\lambda_2 = 1.0$ for the projection consistency loss $\mathcal{L}_{proj}$
- $\lambda_3 = 0.01$ for the mask loss $\mathcal{L}_{mask}$
- $\lambda_4 = 1.0$ for the disparity loss.
- $\lambda_5 = 5 \cdot 10^{-4}$ and $\gamma = 2$ for the sparsity loss $\mathcal{L}_{sp}$

The schedule for these weights are described in Section 4.1.

C. Additional Training Details

For the videos from DAVIS [3], we resize the frames to a resolution of 160×320. Our models are trained for $30k$ iterations per-scene with a batch size of 4 using 4 V100s. Training our model takes approximately an hour per-scene. We implement our code base in JAX/Flax.

D. Editing Effects Implementation Details (Sec. 4.4)

Depth-based editing effects such as synthetic defocus or rerendering along a smoothed camera path require depth in for the foreground layers in addition to the background layer. Our optimization does not directly produce foreground depth, but we can extract an approximate foreground depth D_a^i from the output alpha matte $\mathcal{A}_a^i$ and the input single-layer depth map D_a. We use image erosion on the alpha matte to construct a high-confidence binary mask that covers the internal areas of the foreground without boundary pixels. We then apply this mask to the depth map D_a to produce a partial foreground depth map with valid pixels only inside the high-confidence mask. We then outpaint this partial foreground depth map using standard tools (Inpaint node in NUKE [1]) to fill depth values everywhere in the image. The result is a depth map D_a^i that "bleeds" the foreground object's depth outside the foreground object mask.

To construct the depth-based effects, we apply the effect separately to each layer using their corresponding depth maps: D_a^i for the foreground objects, $\mathcal{D}_a^{bg}$ for the background, then composite using the alpha mattes $\mathcal{A}_a^i$. For synthetic defocus, we apply the same depth-based defocus

effect (ZDefocus in NUKE) to each layer, then composite. For 3D rerendering (camera stabilization), we separately reproject each layer, then composite.

Treating the layers separately in this manner better captures fine details that may be missed by the single-layer depth map D_a. See supplementary video for comparison with single-layer depth-based warping or depth-based defocus.

E. Additional Results

We provide additional layer decomposition results on 40 videos from the DAVIS [3] dataset. These results can be viewed by opening `main.html` in the supplementary zip file in a chrome browser. For each video, we show the background and object layers predicted by our method as well as Omnimatte [2]. The webpage additionally provides a slider interface to compare these output with the input video as well as with each other.

References

[1] The Foundry Visionmongers Ltd. Nuke, 2018. 1

[2] Erika Lu, Forrester Cole, Tali Dekel, Andrew Zisserman, William T. Freeman, and Michael Rubinstein. Omnimatte: Associating objects and their effects in video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4507–4515, June 2021. 2

[3] Jordi Pont-Tuset, Federico Perazzi, Sergi Caelles, Pablo Arbeláez, Alexander Sorkine-Hornung, and Luc Van Gool. The 2017 DAVIS challenge on video object segmentation. *arXiv:1704.00675*, 2017. 1, 2