Colorizing Monochromatic Radiance Fields: Supplementary Material

Anonymous submission

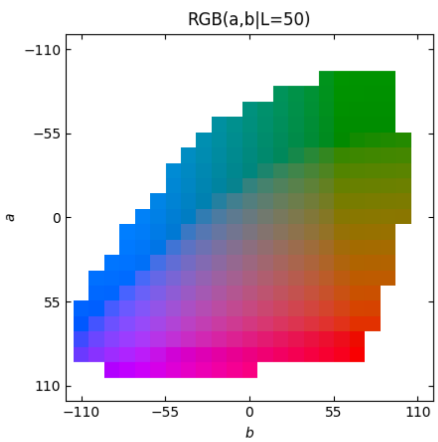

Figure S1: The quantized ab space.

Quantization process

We propose a classification-based color injection module to inject color information into the color representation and preserve the vividness of the results. Inspired by previous methods (Zhang, Isola, and Efros 2016), we use a classification objective during training. A fundamental process for such an operation is to quantize the color space, so we first quantize the possible ab space into $Q = 313$ discrete colors with gird size 10, as visualized in Fig. S1. Note that in all possible $22 \times 22 = 484$ colors, there are only 313 valid colors. As described in the main paper, we use a soft-label operation $\mathcal{S}(\cdot)$ to convert $\mathbf{Y}_{\mathcal{P}}$ to soft label $\mathbf{Z}_{\mathcal{P}}$. This visualization also demonstrates that similar colors are adjacent in this quantized color space, hence we could use the nearest neighbor algorithm to find colors closest to $\mathbf{Y}_{\mathcal{P}}(\mathbf{p})$.

Complete results

We display the complete qualitative experimental results (corresponding to Fig. 6 of the main paper) on our synthetic monochromatic data (all ten datasets shown in Fig. 5 of the main paper) in our supplementary video. In this video, we show a monochromatic input reference, our results, and the baselines described in the main paper: 1) Vid (Lei and Chen 2019)+NeRF; 2) ARF (Zhang et al. 2022); 3) CLIP-NeRF (Wang et al. 2022); 4) CT^2 (Weng et al. 2022)+NeRF.

The video also contains our results on real monochromatic data (corresponding to Fig. 9 of the main paper), demonstrating our possible application on creating colors for neuromorphic sensors such as the spike camera (Huang et al. 2022) or rejuvenating old digital archives in the form of radiance fields.

In the main paper, we present animated results to better demonstrate the multi-view results of our model and the compared baselines. We recommend that readers to further check our supplementary video for animated results with higher resolution and frame rate.

Implementation details

We build our framework based on an unofficial NeRF implementation[1]. Following the settings in (Mildenhall et al. 2020), we use 63-dimension positional encoding for points position and 27-dimension positional encoding for view direction. As described in the main paper, an eight-layer MLP with 256 channels is used for points encoding, and an additional linear layer is used to produce density σ. The luminance representation and color representation have separate directional encoding MLPs, which take the concatenation of points embedding and encoded view direction as input and produce a 128-dimensional view embedding. Lastly, an output layer is employed to produce the predicted color with a specific dimension (1 for luminance representation and 313 for color representation). All MLP layers use ReLU (Agarap 2018) activation function except the output layers, which use Sigmoid (Han and Moraga 1995) activation function. We use the author-provided code and weights in the external colorization module CT^2 (Weng et al. 2022). The parameters of the colorization module are fixed throughout training.

Regarding the implementation of the compared baselines, for ARF (Zhang et al. 2022), we follow their original paradigm to first construct a monochromatic NeRF in Plenoctrees (Yu et al. 2021), then we use a ground truth color image as style image to colorize the scene. In CLIP-NeRF (Wang et al. 2022), we use the author-provided code for color editing, we use prompts that directly describe the scene colors (e.g., in FLOWER scene, the prompt is "red

[1]https://github.com/kwea123/nerf_pl

flower in green leaves".). In Vid (Lei and Chen 2019)+NeRF and CT^2 (Weng et al. 2022)+NeRF, we use the "colorize-then-fuse" paradigm, i.e, we first colorize all the input images and use these colorized images for vanilla NeRF construction.

Training details

In the training process, we first construct the luminance and density representation without patch sampling for 30 epochs, we then fix the parameters in points encoding and MLPs used to predict σ and l. In the color representation prediction process, as described in the "Proposed Method" section of the main paper, we employ the patch sampling scheme similar to GRAF (Schwarz et al. 2020). In every epoch, we first query the luminance and density representation for the monochromatic image patch. We then use our classification-based color injection module to supervise our color representation. We also apply the re-balance loss weights to balance the loss based on color rarity, following prior arts (Weng et al. 2022; Zhang, Isola, and Efros 2016).

About error metrics

We choose error metrics commonly used in image colorization and NeRFs to measure the performance of our model. Fréchet Inception Score (FID) (Heusel et al. 2017) is also used in colorization, measuring the similarity between two Gaussians (the ground truth and the predicted images). However, the test set of NeRFs only contains several images, which makes FID biased and unsuitable for measuring performance.

References

Agarap, A. F. 2018. Deep learning using rectified linear units (relu). *arXiv preprint arXiv:1803.08375*.

Han, J.; and Moraga, C. 1995. The influence of the sigmoid function parameters on the speed of backpropagation learning. In *International Workshop on Artificial Neural Networks*.

Heusel, M.; Ramsauer, H.; Unterthiner, T.; Nessler, B.; and Hochreiter, S. 2017. GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium. In *Proc. of Neural Information Processing Systems*.

Huang, T.; Zheng, Y.; Yu, Z.; Chen, R.; Li, Y.; Xiong, R.; Ma, L.; Zhao, J.; Dong, S.; Zhu, L.; et al. 2022. 1000× faster camera and machine vision with ordinary devices. *Engineering*.

Lei, C.; and Chen, Q. 2019. Fully Automatic Video Colorization With Self-Regularization and Diversity. In *Proc. of Computer Vision and Pattern Recognition*.

Mildenhall, B.; Srinivasan, P. P.; Tancik, M.; Barron, J. T.; Ramamoorthi, R.; and Ng, R. 2020. NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. In *Proc. of European Conference on Computer Vision*.

Schwarz, K.; Liao, Y.; Niemeyer, M.; and Geiger, A. 2020. GRAF: Generative Radiance Fields for 3D-Aware Image Synthesis.

Wang, C.; Chai, M.; He, M.; Chen, D.; and Liao, J. 2022. CLIP-NeRF: Text-and-image driven manipulation of neural radiance fields. In *Proc. of Computer Vision and Pattern Recognition*.

Weng, S.; Sun, J.; Li, Y.; Li, S.; and Shi, B. 2022. CT^2: Colorization Transformer via Color Tokens. In *Proc. of European Conference on Computer Vision*.

Yu, A.; Li, R.; Tancik, M.; Li, H.; Ng, R.; and Kanazawa, A. 2021. Plenoctrees for real-time rendering of neural radiance fields. In *Proc. of Computer Vision and Pattern Recognition*.

Zhang, K.; Kolkin, N.; Bi, S.; Luan, F.; Xu, Z.; Shechtman, E.; and Snavely, N. 2022. ARF: Artistic radiance fields. In *Proc. of European Conference on Computer Vision*.

Zhang, R.; Isola, P.; and Efros, A. A. 2016. Colorful Image Colorization. In *Proc. of European Conference on Computer Vision*.