

NeuroPump: Simultaneous Geometric and Color Rectification for Underwater Images

- Supplementary Material -

A Overview

The supplementary material first provides in-depth derivations for the main paper's §3.1 and §4.1, and complete experiment results on our benchmark dataset. In addition, we include synthesized **videos** of novel views and new optical parameters on our project page <https://ygsu.github.io/NeuroPump.github.io/>.

B Refraction

Derivation of main paper's Equation 15, *i.e.*

$$\mathbf{d}_w(\mathbf{x}) = \text{Refract}(\mathbf{d}_a(\mathbf{x}), \phi_a(\mathbf{x}), \phi_w(\mathbf{x})). \quad (1)$$

It can be further elucidated using Rodrigues' rotation formula:

$$\begin{aligned} \mathbf{d}_w(\mathbf{x}) &= \mathbf{d}_a(\mathbf{x}) \cos(\phi_a(\mathbf{x}) - \phi_w(\mathbf{x})) \\ &\quad + (\mathbf{u}(\mathbf{x}) \times \mathbf{d}_a(\mathbf{x})) \sin(\phi_a(\mathbf{x}) - \phi_w(\mathbf{x})) \\ &\quad + (\mathbf{u}(\mathbf{x}) \cdot \mathbf{d}_a(\mathbf{x})) \mathbf{u}(\mathbf{x}) (1 - \cos(\phi_a(\mathbf{x}) - \phi_w(\mathbf{x}))). \end{aligned} \quad (2)$$

The rotation axis $\mathbf{u}(\mathbf{x})$ is given by:

$$\mathbf{u}(\mathbf{x}) = \mathbf{d}_a(\mathbf{x}) \times \mathbf{g}. \quad (3)$$

where \mathbf{g} is the camera lens case's normal vector.

C Single image geometry rectification

Derivation of the geometric rectification factor h (*i.e.* main paper's Equation 21). As shown in Fig. 1, we assume the world origin is at the camera optical center $O = (0, 0, 0)$, and the camera sensor's principal point is $m = (0, 0)$. For an object point $\mathbf{x} = (x, y, z)$ in the world space, its corresponding pixel coordinate is $\mathbf{x} = (u, v)$ when imaged underwater due to refraction. However, its actual pixel coordinate should be $\mathbf{x}' = (u', v')$, and the relationship between \mathbf{x}' and \mathbf{x} is given by:

$$\begin{aligned} (u', v') &= (hu, hv), \\ h &= \frac{\|\mathbf{x}'m\|_2}{\|\mathbf{x}m\|_2} = \frac{\|\mathbf{x}M\|_2}{\|\mathbf{x}'M\|_2} \\ &= \frac{s \tan(\phi_a(\mathbf{x})) + (z - s) \tan(\phi_w(\mathbf{x}))}{z \tan(\phi_a(\mathbf{x}))}. \end{aligned} \quad (4)$$

In practice, z and s are unknown, and we can solve h in three different cases:

(1) When s is small, *e.g.* $s \approx 0$, h can be simplified as follows:

$$h = \frac{\tan(\phi_w(\mathbf{x}))}{\tan(\phi_a(\mathbf{x}))}. \quad (5)$$

(2) When z is uniform across object points, *e.g.* all object points lie on the same plane that is parallel to the camera lens sensor and lens case. Once s/z is obtained, h becomes:

$$h = \frac{\tan \phi_w + (s/z)(\tan(\phi_a(\mathbf{x})) - \tan(\phi_w(\mathbf{x})))}{\tan(\phi_a(\mathbf{x}))}. \quad (6)$$

(3) When z varies across object points, each object point's depth is needed to solve for accurate h .

D Experiments

We provide the complete comparison between our NeuroPump and other baselines on our benchmark dataset in Fig. 2 (excluding Totoro, as it is already included in the main paper's Figure 4) and Tab. 2.

We also present a complexity comparison of our method and all baselines in Tab. 1 for reference only, as runtime is not our focus.

As shown in the **videos** of novel view synthesis, our NeuroPump's final rectification results clearly outperform other baselines both quantitatively and qualitatively. Moreover, our NeuroPump can perform new optical parameter synthesis because it explicitly decouples and learn these optical parameters.

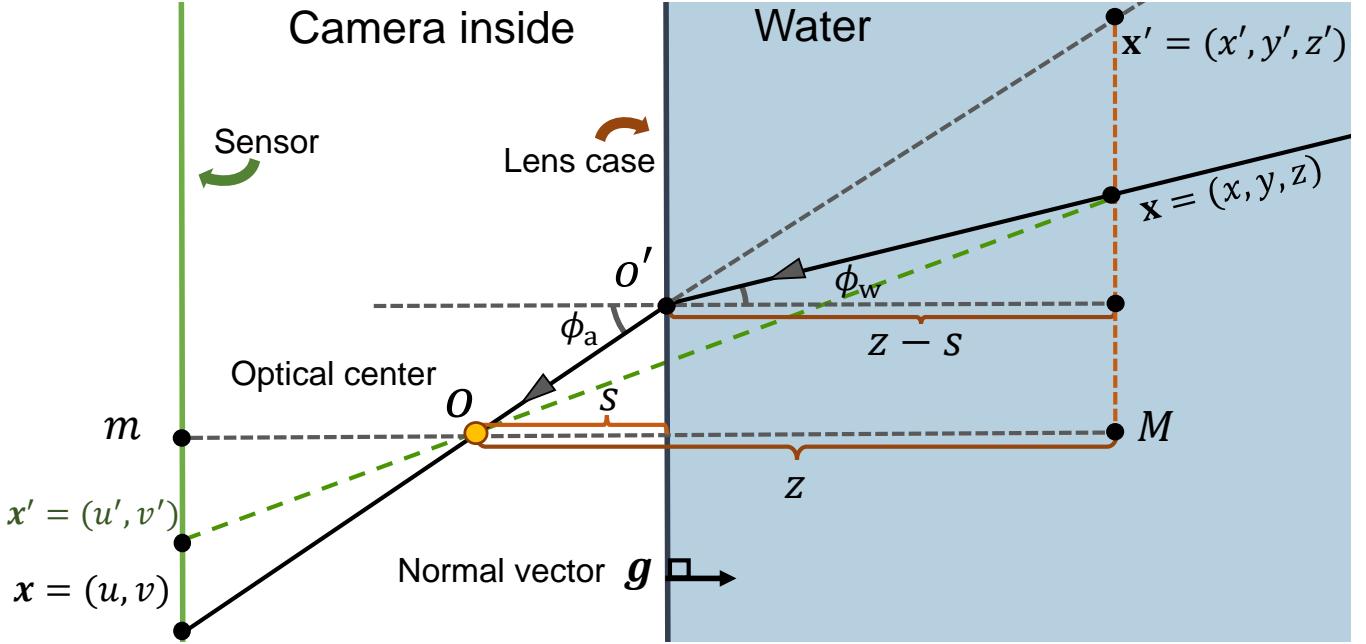


Figure 1: Underwater imaging. We assume the world origin is at the camera optical center $O = (0, 0, 0)$, and the camera sensor's principal point is $m = (0, 0)$. Given an object point $x = (x, y, z)$ in the world space, its corresponding pixel coordinate is $x = (u, v)$ when imaged underwater due to refraction. While x 's actual pixel coordinate should be $x' = (u', v')$. Moreover, g is the normal vector of lens case interface.

Table 1: Runtime and model size. Metrics are calculated for image resolution 910×540. Note that Mip360 (Lav)+color-rectification only applies color rectifications during test and color rectification models are not re-trained, thus they share the same training time.

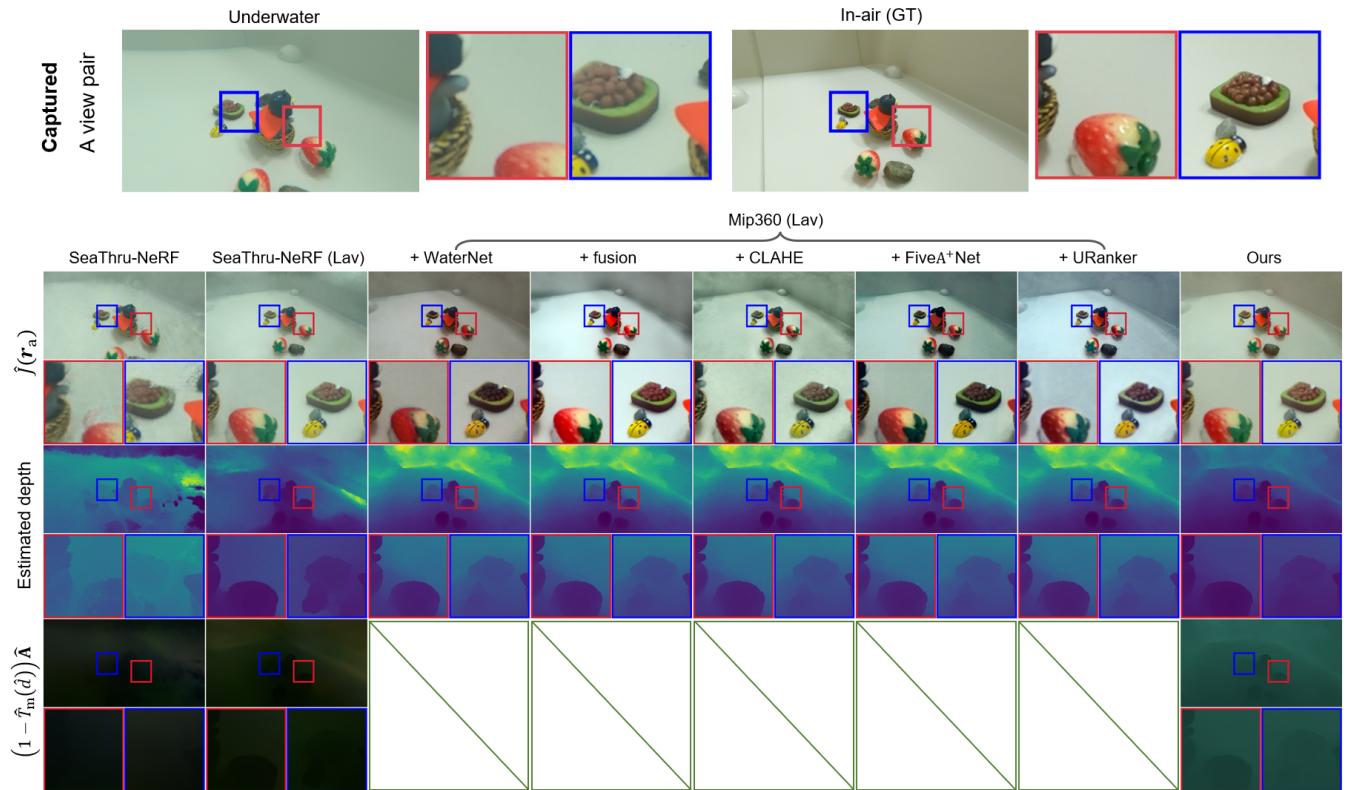
Method	Train (sec/iter.)	Test (sec/img.)	#Params.
Seathru-NeRF [5]	~0.135	~14	8,197,262
Seathru-NeRF (Lav [4])	~0.135	~14	8,197,262
Mip360 (Lav) [2] + WaterNet [6]	~0.144	~16.3	10,098,161
Mip360 (Lav) + fusion [1]	~0.144	~16.4	9,007,493
Mip360 (Lav) + CLAHE [8]	~0.144	~15.8	9,007,493
Mip360 (Lav) + FiveA ⁺ Net [3]	~0.144	~16	9,016,483
Mip360 (Lav) + URanker [7]	~0.144	~16.9	18,445,262
Ours	~0.144	~16.5	9,007,499

E More dataset acquisition details

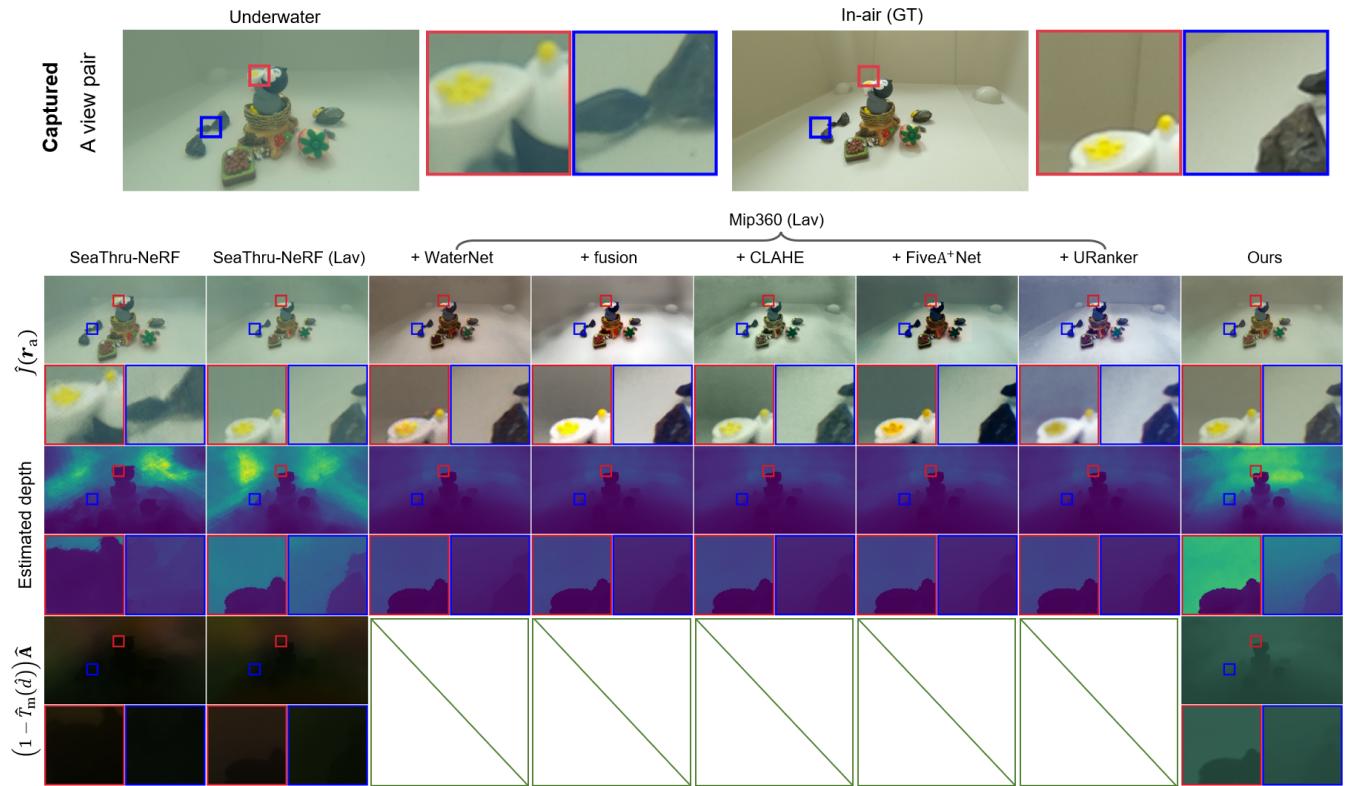
In addition to data acquisition details in main paper §5.1, camera pose consistency during in-air and underwater capturing is also worth noting. Therefore, instead of moving the camera directly, we mount the camera onto a tripod with an overhead boom, and only rotate a turntable beneath the water tank, and control the camera remotely over Wi-Fi.

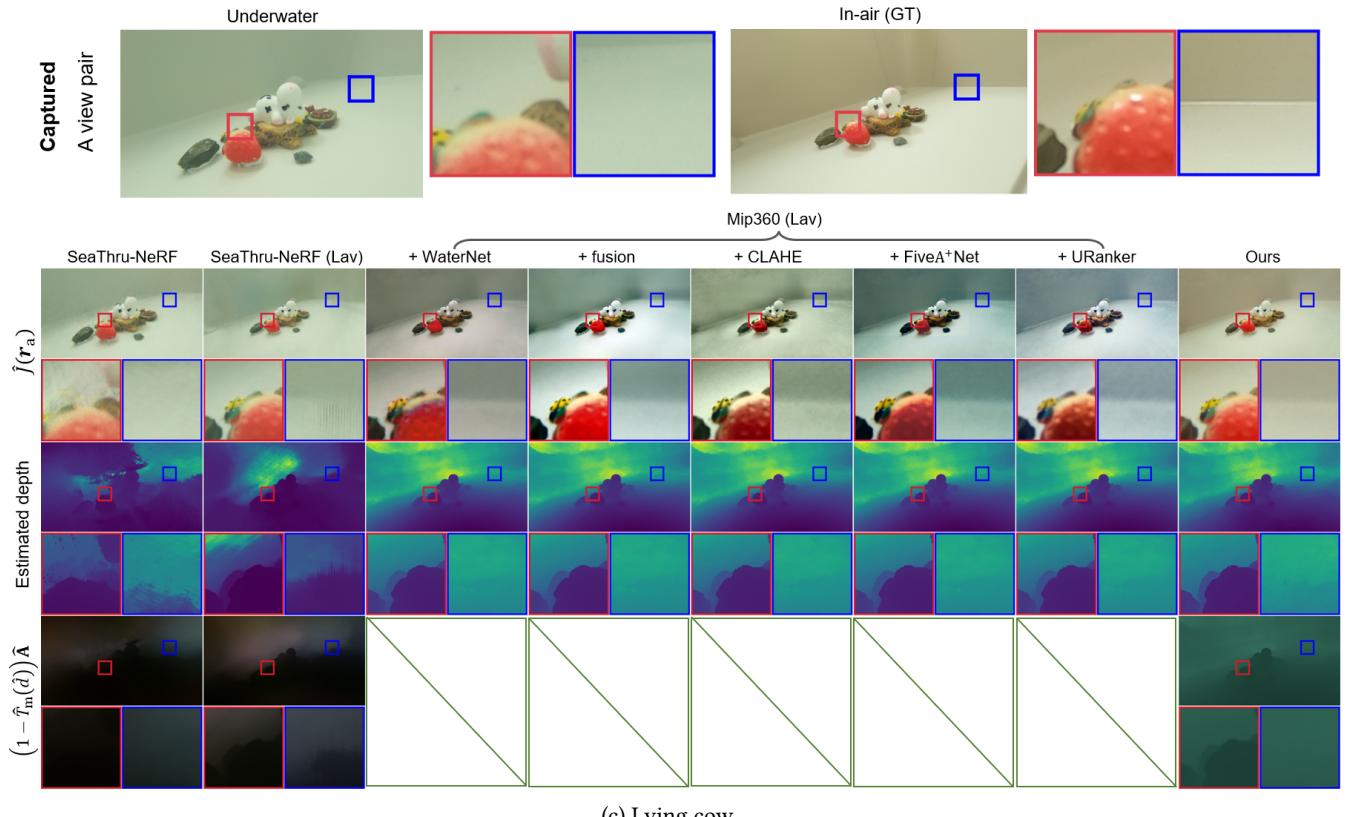
Table 2: Complete quantitative comparison on joint geometric and color rectification (i.e. obtaining the in-air image $\hat{j}(\mathbf{r}_a)$, the final target) on our benchmark dataset. Note that the metrics of the Mip360 (Lav) + Others is worse than ours, because the color rectification methods focus on improving color contrast, saturation and brightness that may be different from the real in-air GT images.

Data	Model	$\hat{j}(\mathbf{r}_a)$ (geometry & color)		
		PSNR↑	SSIM↑	RMSE↓
Penguin hero	Seathru-NeRF [5]	16.8309	0.7847	0.1446
	Seathru-NeRF (Lav [4])	21.2786	0.8850	0.0865
	Mip360 [2] (Lav) + WaterNet [6]	15.0396	0.8417	0.1806
	Mip360 (Lav) + fusion [1]	19.9828	0.8655	0.1003
	Mip360 (Lav) + CLAHE [8]	20.3073	0.8132	0.0965
	Mip360 (Lav) + FiveA ⁺ Net [3]	16.7797	0.8380	0.1449
	Mip360 (Lav) + URanker [7]	18.9338	0.8449	0.1132
	Ours	23.0724	0.8965	0.0702
Penguin flower	Seathru-NeRF [5]	16.6865	0.7820	0.1467
	Seathru-NeRF (Lav [4])	20.6086	0.8867	0.0934
	Mip360 [2] (Lav) + WaterNet [6]	17.7489	0.8557	0.1311
	Mip360 (Lav) + fusion [1]	19.5272	0.8556	0.1057
	Mip360 (Lav) + CLAHE [8]	20.3904	0.8246	0.0957
	Mip360 (Lav) + FiveA ⁺ Net [3]	16.7996	0.8348	0.1446
	Mip360 (Lav) + URanker [7]	18.9476	0.8372	0.1129
	Ours	22.7369	0.8934	0.0732
Lying cow	Seathru-NeRF [5]	17.6290	0.8115	0.1320
	Seathru-NeRF (Lav [4])	23.4194	0.9201	0.0675
	Mip360 [2] (Lav) + WaterNet [6]	15.4484	0.8546	0.1768
	Mip360 (Lav) + fusion [1]	19.0020	0.8770	0.1123
	Mip360 (Lav) + CLAHE [8]	17.0293	0.7615	0.1410
	Mip360 (Lav) + FiveA ⁺ Net [3]	15.6349	0.8511	0.1655
	Mip360 (Lav) + URanker [7]	18.9539	0.8663	0.1128
	Ours	25.7708	0.9214	0.0516
Totoro	Seathru-NeRF [5]	17.3704	0.8236	0.1355
	Seathru-NeRF (Lav [4])	20.4499	0.9101	0.0951
	Mip360 [2] (Lav) + WaterNet [6]	12.2523	0.8381	0.2479
	Mip360 (Lav) + fusion [1]	18.5969	0.8730	0.1187
	Mip360 (Lav) + CLAHE [8]	19.0212	0.8415	0.1119
	Mip360 (Lav) + FiveA ⁺ Net [3]	14.2387	0.8536	0.1941
	Mip360 (Lav) + URanker [7]	15.7144	0.8356	0.1639
	Ours	22.4898	0.9113	0.0751
Hamster	Seathru-NeRF [5]	17.0561	0.8135	0.1403
	Seathru-NeRF (Lav [4])	18.6212	0.8814	0.1172
	Mip360 [2] (Lav) + WaterNet [6]	15.2562	0.8317	0.1757
	Mip360 (Lav) + fusion [1]	16.8127	0.8219	0.1450
	Mip360 (Lav) + CLAHE [8]	17.2735	0.8291	0.1380
	Mip360 (Lav) + FiveA ⁺ Net [3]	12.9441	0.7956	0.2254
	Mip360 (Lav) + URanker [7]	14.9388	0.7435	0.1793
	Ours	19.7682	0.8816	0.1028



(a) Penguin hero





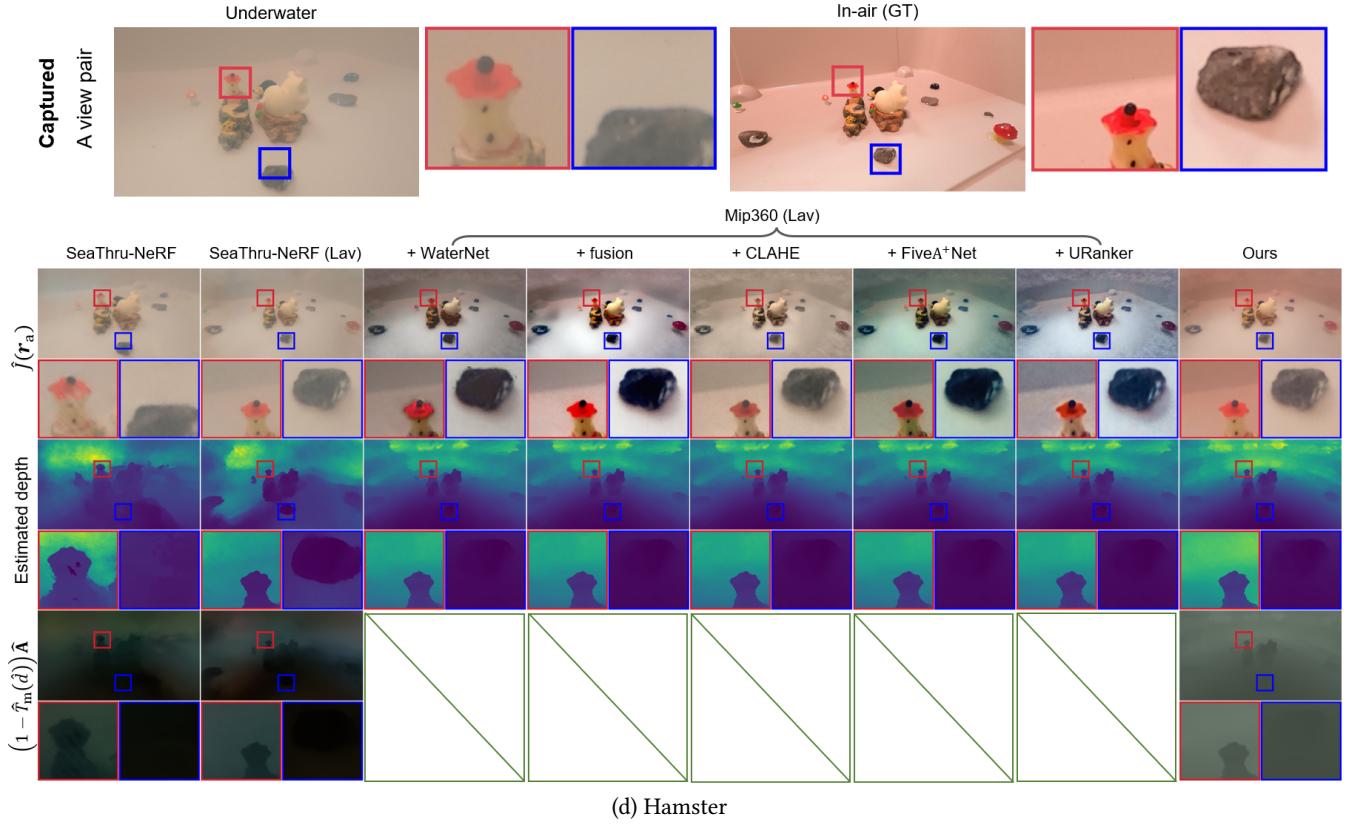


Figure 2: Complete qualitative comparison between NeuroPump and other baselines on our benchmark dataset (excluding Totoro, which is already included in the main paper’s Figure 4). For geometry, clearly, the original SeaThru-NeRF’s result looks like a zoomed-in version with corner stretching of the in-air ground truth, because it cannot rectify geometry distortion caused by refraction. For colors, note that SeaThru-NeRF and SeaThru-NeRF (Lav) cannot accurately rectify underwater color, and the estimated depth (the 2nd row) and back-scatter (the 3rd row) are also inferior. Mip360 (Lav)-based methods show differences in brightness, saturation and color compared to the in-air ground truth. In comparison, our NeuroPump shows clear advantages in all results.

References

- [1] Codruta O. Ancuti, Cosmin Ancuti, Christophe De Vleeschouwer, and Philippe Bekaert. 2018. Color Balance and Fusion for Underwater Image Enhancement. *IEEE Transactions on Image Processing (TIP)* 27, 1 (2018), 379–393.
- [2] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. 2022. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 5470–5479.
- [3] JingXia Jiang, Tian Ye, Sixiang Chen, Erkang Chen, Yun Liu, Shi Jun, Jinbin Bai, and Wenhao Chai. 2023. Five A+ Network: You Only Need 9K Parameters for Underwater Image Enhancement. In *34th British Machine Vision Conference 2023, BMVC 2023, Aberdeen, UK, November 20-24, 2023*. BMVA.
- [4] Jean-Marc Lavest, Gérard Rives, and Jean-Thierry Lapresté. 2000. Underwater camera calibration. In *Proc. Eur. Conf. Comput. Vis. (ECCV)*. Springer, 654–668.
- [5] Deborah Levy, Amit Peleg, Naama Pearl, Dan Rosenbaum, Derya Akkaynak, Simon Korman, and Tali Treibitz. 2023. SeaThru-NeRF: Neural Radiance Fields in Scattering Media. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 56–65.
- [6] Chongyi Li, Chunle Guo, Wenqi Ren, Runmin Cong, Junhui Hou, Sam Kwong, and Dacheng Tao. 2020. An Underwater Image Enhancement Benchmark Dataset and Beyond. *IEEE Trans. Image Processing (TIP)* 29 (2020), 4376–4389.
- [7] C. Guo *et al.* 2023. Underwater Ranker: Learn Which Is Better and How to Be Better. In *AAAI 2023*.
- [8] Karel Zuiderveld. 1994. Contrast limited adaptive histogram equalization. *Graphics gems* (1994), 474–485.