

# 1 NeuroPump: Simultaneous Geometric and Color Rectification for 2 Underwater Images 3 - Supplementary Material-

4  
 5  
 6  
 7  
 8  
 9  
 10 Anonymous Author(s)  
 11  
 12  
 13  
 14  
 15  
 16  
 17  
 18  
 19  
 20  
 21  
 22  
 23  
 24  
 25  
 26  
 27  
 28  
 29  
 30  
 31  
 32  
 33  
 34  
 35  
 36  
 37  
 38  
 39  
 40  
 41  
 42  
 43  
 44  
 45  
 46  
 47  
 48  
 49  
 50  
 51  
 52  
 53  
 54  
 55  
 56  
 57  
 58  
 59  
 60  
 61  
 62  
 63  
 64  
 65  
 66  
 67  
 68  
 69  
 70  
 71  
 72  
 73  
 74  
 75  
 76  
 77  
 78  
 79  
 80  
 81  
 82  
 83  
 84  
 85  
 86  
 87  
 88  
 89  
 90  
 91  
 92  
 93  
 94  
 95  
 96  
 97  
 98  
 99  
 100  
 101  
 102  
 103  
 104  
 105  
 106  
 107  
 108  
 109  
 110  
 111  
 112  
 113  
 114  
 115  
 116

## ACM Reference Format:

Anonymous Author(s). 2025. NeuroPump: Simultaneous Geometric and Color Rectification for Underwater Images - Supplementary Material-. In *Proceedings of xxxxxxxxx (ACM MM'25)*. ACM, New York, NY, USA, 8 pages.  
<https://doi.org/XXXXXXX.XXXXXXX>

## 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 in the zip file.

## B Refraction

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

$$d_w(x) = \text{Refract}(d_a(x), \phi_a(x), \phi_w(x)). \quad (1)$$

It can be further elucidated using Rodrigues' rotation formula:

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

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

$$\mathbf{u}(x) = d_a(x) \times g, \quad (3)$$

where  $g$  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 principle point is  $m = (0, 0)$ . For 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. However, its actual pixel coordinate should be  $x' = (u', v')$ , and the relationship between  $x'$

---

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

ACM MM'25, Dublin, Ireland

© 2025 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM ISBN 978-x-xxxx-xxxx-x/YYY/MM

<https://doi.org/XXXXXXX.XXXXXXX>

Anonymous Author(s)  
 Submission Id: 3091

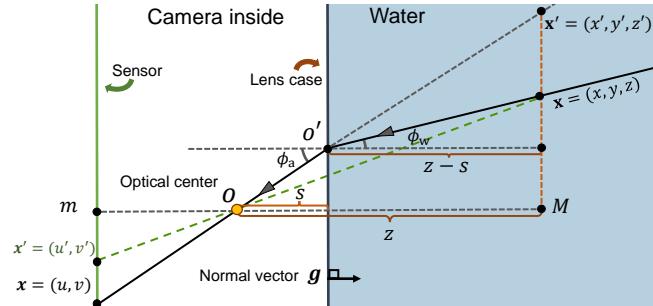


Figure 1: Underwater imaging. We assume the world origin is at the camera optical center  $O = (0, 0, 0)$ , and the camera sensor's principle point is  $m = (0, 0)$ . Giving an object point  $x = (x, y, z)$  in the world space, its corresponding pixel coordinate is  $x = (u, v)$  when imaged underwater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

and  $x$  is given by:

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

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

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

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

(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:

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

(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.

**Table 1: Runtime and model size.** Metrics are calculated for image resolution  $910 \times 540$ . Note that Mip360 (Lav)+color-rectification only apply 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 <sup>+</sup> 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

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

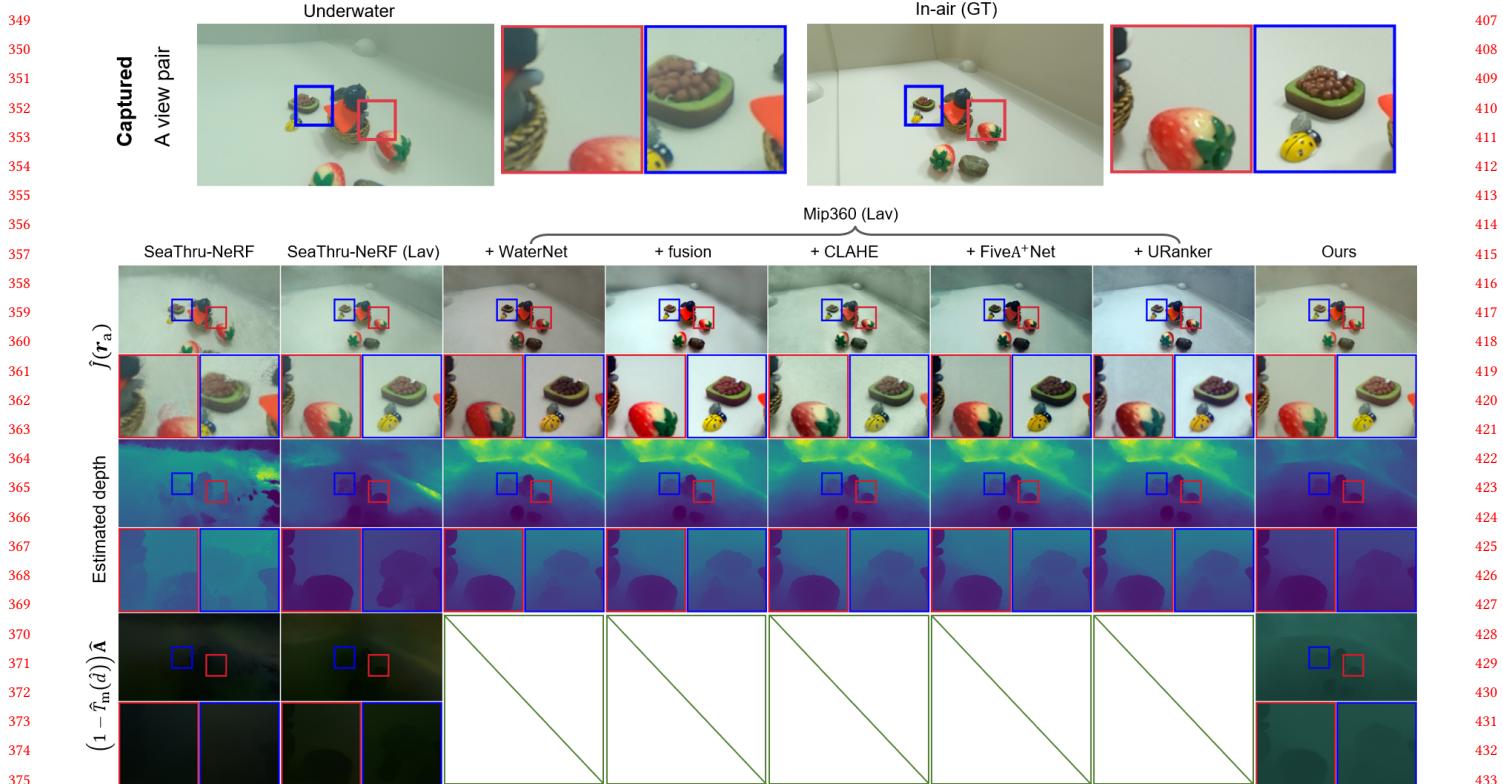
As shown in the **videos** of novel view synthesis, our NeuroPump's final rectification results clearly outperform other baselines both quantitatively and qualitatively. Moreover, our NeuroPump can perform new optical parameter synthesis because it explicitly decouple and learn these optical parameter.

## E More dataset acquisition details

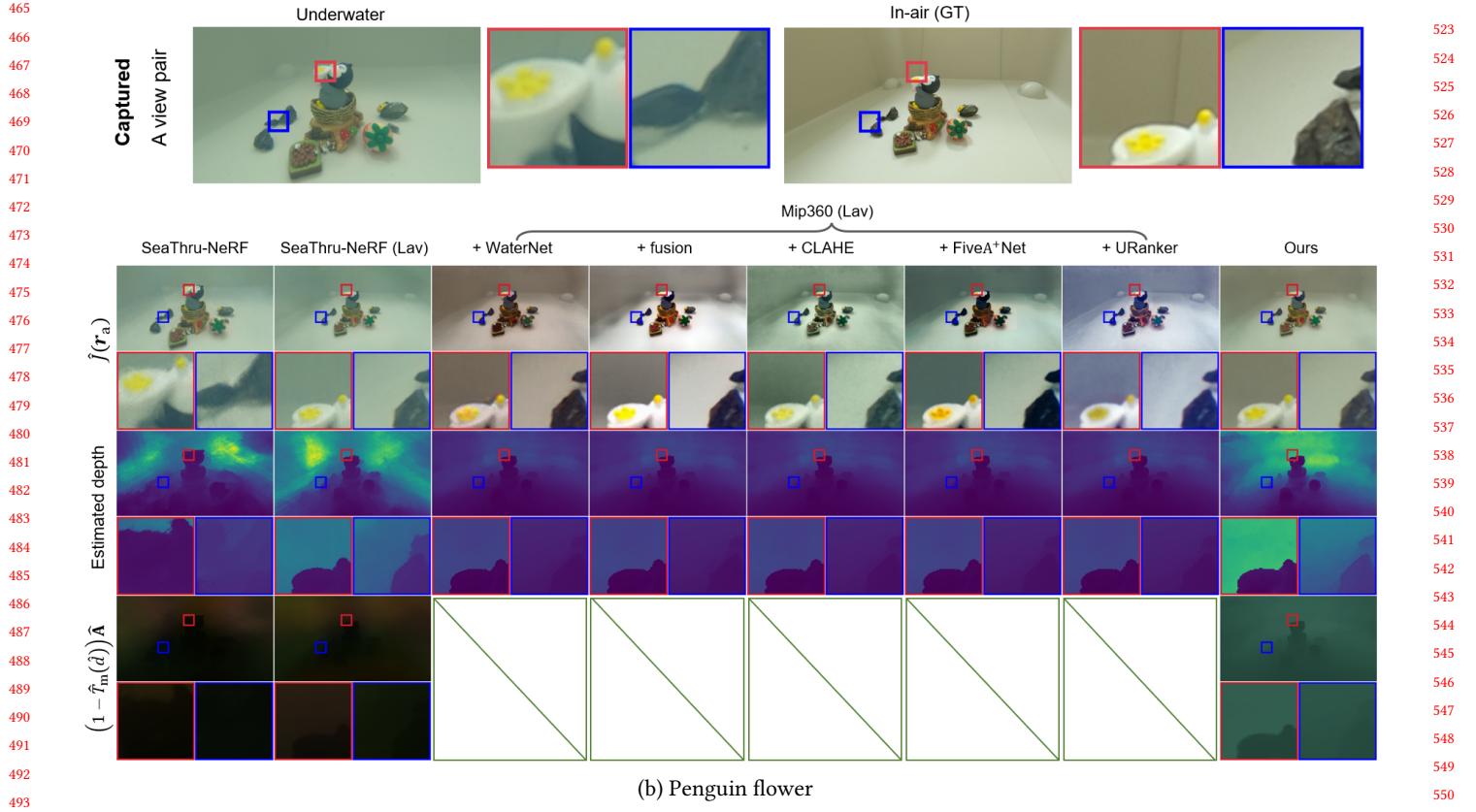
In addition to data acquisition details in main paper §5.1, camera pose consistency during in-air and underwater capturing is also worth noting. Therefore, instead of moving the camera directly, we mount the camera onto a tripod with an overhead boom, and only rotate a turntable beneath the water tank,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

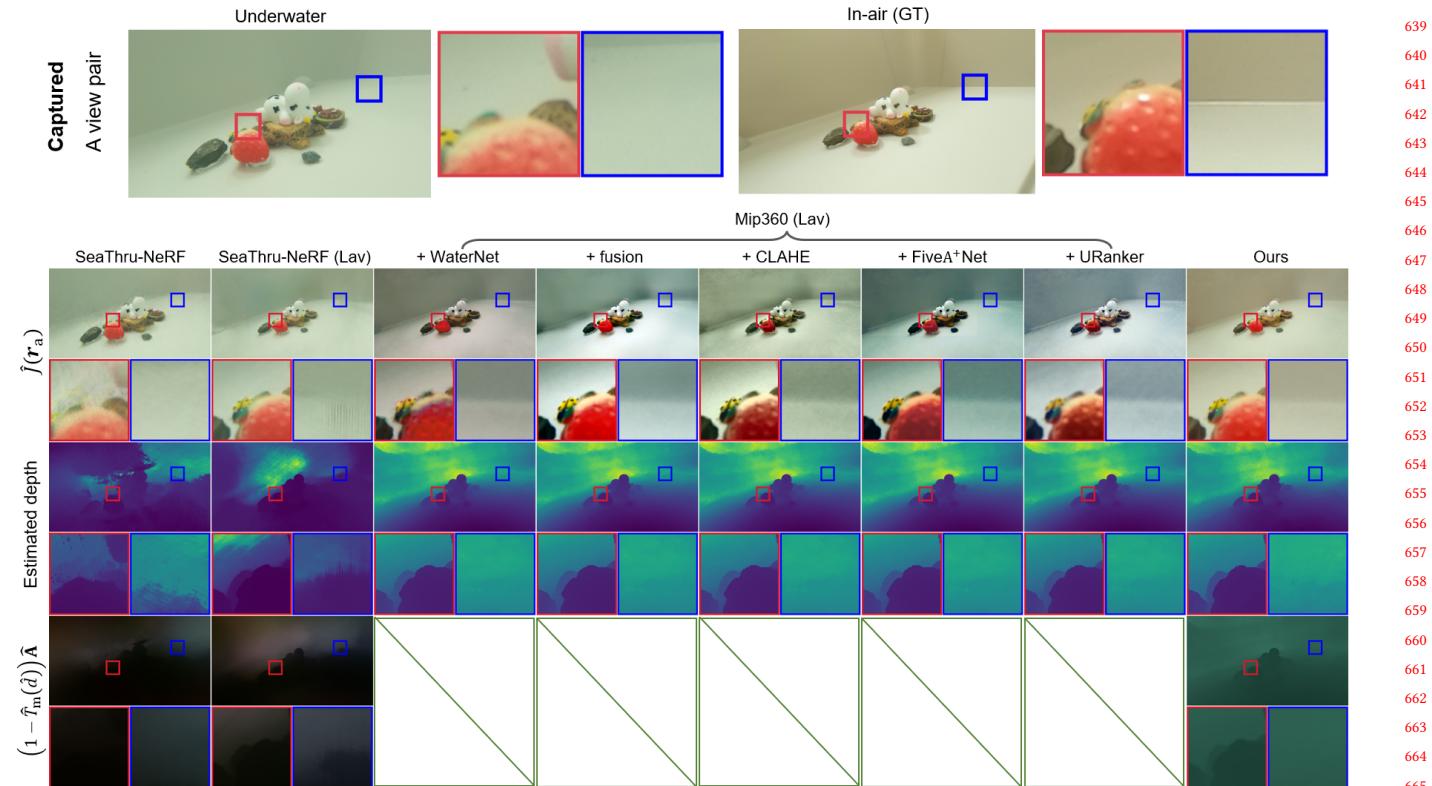
**Table 2: Complete quantitative comparison on joint geometric and color rectification (i.e. obtaining the in-air image  $\hat{J}(\mathbf{r}_a)$ , the final target) on our benchmark dataset. Note that the metrics of the Mip360 (Lav) + Others is worse than ours, because the color rectification methods 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 <sup>+</sup> Net [3]	16.7797	0.8380	0.1449
	Mip360 (Lav) + URanker [7]	18.9338	0.8449	0.1132
	Ours	<b>23.0724</b>	<b>0.8965</b>	<b>0.0702</b>
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 <sup>+</sup> Net [3]	16.7996	0.8348	0.1446
	Mip360 (Lav) + URanker [7]	18.9476	0.8372	0.1129
	Ours	<b>22.7369</b>	<b>0.8934</b>	<b>0.0732</b>
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 <sup>+</sup> Net [3]	15.6349	0.8511	0.1655
	Mip360 (Lav) + URanker [7]	18.9539	0.8663	0.1128
	Ours	<b>25.7708</b>	<b>0.9214</b>	<b>0.0516</b>
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 <sup>+</sup> Net [3]	14.2387	0.8536	0.1941
	Mip360 (Lav) + URanker [7]	15.7144	0.8356	0.1639
	Ours	<b>22.4898</b>	<b>0.9113</b>	<b>0.0751</b>
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 <sup>+</sup> Net [3]	12.9441	0.7956	0.2254
	Mip360 (Lav) + URanker [7]	14.9388	0.7435	0.1793
	Ours	<b>19.7682</b>	<b>0.8816</b>	<b>0.1028</b>

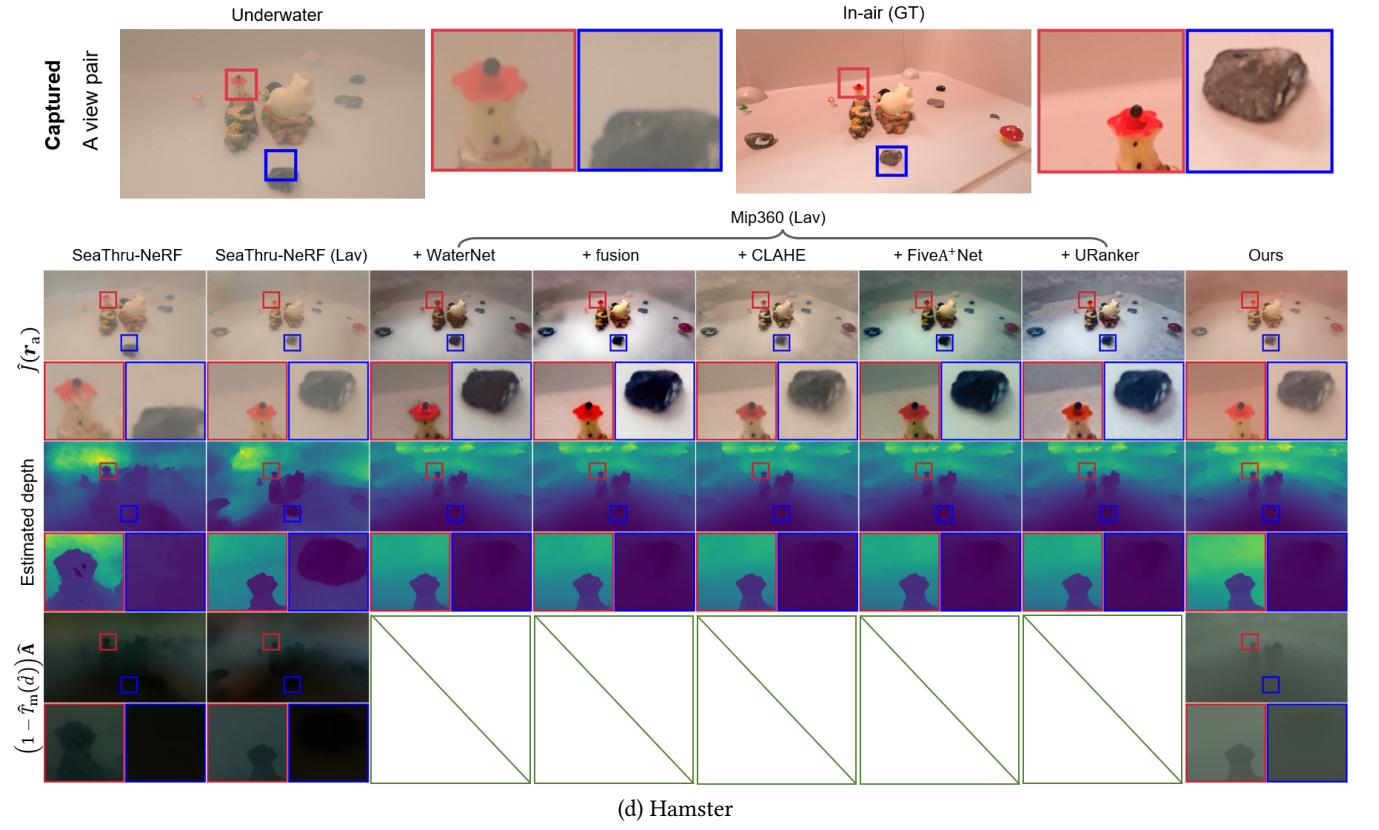


(a) Penguin hero





(c) Lying cow



**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.

697  
698  
699  
700  
701  
702  
703  
704  
705  
706  
707  
708  
709  
710  
711  
712  
713  
714  
715  
716  
717  
718  
719  
720  
721  
722  
723  
724  
725  
726  
727  
728  
729  
730  
731  
732  
733  
734  
735  
736  
737  
738  
739  
740  
741  
742  
743  
744  
745  
746  
747  
748  
749  
750  
751  
752  
753  
754  
755  
756  
757  
758  
759  
760  
761  
762  
763  
764  
765  
766  
767  
768  
769  
770  
771  
772  
773  
774  
775  
776  
777  
778  
779  
780  
781  
782  
783  
784  
785  
786  
787  
788  
789  
790  
791  
792  
793  
794  
795  
796  
797  
798  
799  
800  
801  
802  
803  
804  
805  
806  
807  
808  
809  
810  
811  
812

## 813 References

- 814 [1] Codruța O. Ančuti, Cosmin Ančuti, Christophe De Vleeschouwer, and Philippe  
815 Bekaert. 2018. Color Balance and Fusion for Underwater Image Enhancement.  
816 *IEEE Transactions on Image Processing (TIP)* 27, 1 (2018), 379–393.
- 817 [2] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter  
818 Hedman. 2022. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In  
819 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition  
(CVPR)*. 5470–5479.
- 820 [3] Jingxia Jiang, Tian Ye, Jinbin Bai, Sixiang Chen, Wenhao Chai, Shi Jun, Yun Liu,  
821 and Erkang Chen. 2023. Five A<sup>+</sup> Network: You Only Need 9K Parameters for  
822 Underwater Image Enhancement. *arXiv preprint arXiv:2305.08824* (2023).
- 823 [4] Jean-Marc Lavest, Gérard Rives, and Jean-Thierry Lapresté. 2000. Underwater  
824 camera calibration. In *Proc. Eur. Conf. Comput. Vis. (ECCV)*. Springer, 654–668.
- 825 [5] Deborah Levy, Amit Peleg, Naama Pearl, Dan Rosenbaum, Derya Akkaynak,  
826 Simon Korman, and Tali Treibitz. 2023. SeaThru-NerF: Neural Radiance Fields in  
827 Scattering Media. In *Proceedings of the IEEE/CVF Conference on Computer Vision  
and Pattern Recognition (CVPR)*. 56–65.
- 828 [6] Chongyi Li, Chunli Guo, Wenqi Ren, Runmin Cong, Junhui Hou, Sam Kwong,  
829 and Dacheng Tao. 2020. An Underwater Image Enhancement Benchmark Dataset  
830 and Beyond. *IEEE Trans. Image Processing (TIP)* 29 (2020), 4376–4389.
- 831 [7] C. Guo *et al.* 2023. Underwater Ranker: Learn Which Is Better and How to Be  
832 Better. In *AAAI 2023*.
- 833 [8] Karel Zuiderveld. 1994. Contrast limited adaptive histogram equalization. *Graphics  
834 gems* (1994), 474–485.
- 835 [881]
- 836 [882]
- 837 [883]
- 838 [884]
- 839 [885]
- 840 [886]
- 841 [887]
- 842 [888]
- 843 [889]
- 844 [890]
- 845 [891]
- 846 [892]
- 847 [893]
- 848 [894]
- 849 [895]
- 850 [896]
- 851 [897]
- 852 [898]
- 853 [899]
- 854 [900]
- 855 [901]
- 856 [902]
- 857 [903]
- 858 [904]
- 859 [905]
- 860 [906]
- 861 [907]
- 862 [908]
- 863 [909]
- 864 [910]
- 865 [911]
- 866 [912]
- 867 [913]
- 868 [914]
- 869 [915]
- 870 [916]
- 871 [917]
- 872 [918]
- 873 [919]
- 874 [920]
- 875 [921]
- 876 [922]
- 877 [923]
- 878 [924]
- 879 [925]
- 880 [926]
- 881 [927]

871 [928]

872 [929]

873 [930]

874 [931]

875 [932]

876 [933]

877 [934]

878 [935]

879 [936]

880 [937]

881 [938]

882 [939]

883 [940]

884 [941]

885 [942]

886 [943]

887 [944]

888 [945]

889 [946]

890 [947]

891 [948]

892 [949]

893 [950]

894 [951]

895 [952]

896 [953]

897 [954]

898 [955]

899 [956]

900 [957]

901 [958]

902 [959]

903 [960]

904 [961]

905 [962]

906 [963]

907 [964]

908 [965]

909 [966]

910 [967]

911 [968]

912 [969]

913 [970]

914 [971]

915 [972]

916 [973]

917 [974]

918 [975]

919 [976]

920 [977]

921 [978]

922 [979]

923 [980]

924 [981]

925 [982]

926 [983]

927 [984]