
Looking Into the Water by Unsupervised Learning of the Surface Gradient - Supplementary Material

Anonymous Author(s)

Affiliation

Address

email

1 Implementation Details

The architecture of our model consists of two parts. Both parts are implemented using SIREN models [3]. The image model I_ϕ is a 3-layer SIREN with a hidden layer size of 256. The surface height model H_θ is a 2-layer SIREN with hidden layer sizes of 128. We also utilized positional encoding [4] and similar to a previous work [5] we also found that the optimal bandwidth scale factor κ was 8.

Hyperparameter tuning is conducted on the first batch of 10 consecutive frames per sequence. Main results (Table 1, main paper) are computed over all non-overlapping frame batches. We perform grid search using Eq. 1, evaluating reconstruction quality via PSNR, SSIM, and LPIPS. We tune the hyper-parameters controlling ω_0 of both SIREN models separately for the synthetic and real-world data. This is due to the difference in image sizes and the difference between simulated and real waves. We search over the range of 5 to 75, and also tune the learning rate. The two sets of hyperparameters are as follows:

Real1 [2]: For I_ϕ : $\omega_0 = 45$ with learning rate = $1e-4$. For H_θ : $\omega_0 = 15$ and learning rate = $1e-3$, the first training stage ran for 700 iterations and the second training stage ran for 1000 iterations.

Synthetic: For I_ϕ : $\omega_0 = 10$ with learning rate = $1e-3$. For H_θ : $\omega_0 = 15$ and learning rate = $1e-3$, the first training stage ran for 700 iterations and for the second training stage we found that 500 iterations suffice.

All experiments were conducted on a GeForce 4090 RTX GPU, memory footprint on GPU is roughly 15 GB depending on the length of the input sequence and the size of the images, the networks we use are relatively shallow. Run time is around 5 minutes for a batch of 10 images on *Real1*, on-par with other methods. In terms of total compute, to reproduce all the main results of our paper, it would take roughly 2-hours on a GeForce 4090 RTX machine. Code and data are provided in the supplementary material with scripts to run all experiments in this paper.

2 Training SIREN with the Signal Gradient

We describe in more detail the training signal of our model, highlighting an aspect that was proposed in the original SIREN work [3], namely, the possibility of training the model by matching the spatial gradient of the output to the spatial gradient of the data.

Starting from our loss function,

$$\mathcal{L}(\theta, \phi) = \sum_t |I_{\theta, \phi}^t - I_t| \quad (1)$$

we expand the terms by plugging in explicitly the computation of the predicted image $I_{\theta,\phi}^t$ using our model, and the image formation of the observed image I_t given the image formation model,

$$\begin{aligned} |I_{\theta,\phi}^t - I_t| = \\ \left| I_{\phi} \left(x_{\text{reg}} + \left(1 - \frac{1}{n} \right) h_0 \nabla H_{\theta}(x, t) \right) - J \left(x_{\text{reg}} + \left(1 - \frac{1}{n} \right) h_0 \nabla H(x, t) \right) \right| \end{aligned} \quad (2)$$

where we use Snell’s law as described in the main paper, and $J(\cdot)$ denotes the ground truth undistorted image evaluated at the given pixel positions.

We note that the observed image is obtained through the physical image formation process which involves different water surface heights and Snell’s law, and that the predicted image is computed using the same computational model. The only differences between the two terms is that for the predicted image, both the water surface H_{ϕ} and the underlying clean image I_{θ} are estimated by our model.

Eq. 2 highlights the fact that the height network H_{θ} is trained by matching its gradient to the gradient of the ground truth height H . This provides a physics driven justification for training the signal using gradient supervision, which was originally proposed as one of the advantages of SIREN.

3 Quantitative Results & Videos on Surface Height Reconstruction

We examine our distortion and surface height estimation results from the synthetic set described in the main paper. The quantitative results evaluate the distortion estimation using the measures RMSE \downarrow , EPE \downarrow and AAE $^{\circ}$ \downarrow . The average endpoint error (EPE) [6] is the Euclidean norm of the difference between the predicted and the true distortion vectors, averaged over all image pixels. Average angular error (AAE) measures the error in degrees between the distortion vectors, averaged over all pixels. Surface height estimation is measured with the scale-invariant measure SILog \uparrow [1]. This measure is not compared with [8] which does not estimate surface height.

Tables 1, 2, 3 present the results on the 10 frames of each wave. Wave1 is a Gaussian wave, and Waves 2&3 are ripple waves. In all of them our results outperform NDIR [8], where the largest difference is in Wave3.

We also provide videos demonstrating the height estimation of our model for the aforementioned 3 sequences. Along with the estimated height and the simulated ground truth height, in each video we show the derived pixel offset distortion of our method, the offsets predicted directly by NDIR and the ground truth offsets resulting from the simulation. Also presented in the videos are the ground truth "Measurement" images (produced by ray tracing in the simulator) alongside a warping of our predicted clean image by using our predicted offsets. If either of our predicted clean image or our predicted offsets are faulty then the predicted "Backwarp" image should not resemble the ground truth measurement image. Figures 2, 3, 4 show single frames from these 3 videos.

61

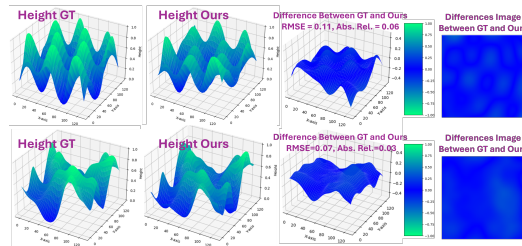


Figure 1: Example of differences between the ground-truth water surface and our prediction. Left to right: we show ground-truth and our prediction of the water surface, a 3D surface plot of the differences, and a top-down view for two types of waves (top: ripple, bottom: Gaussian).

Table 1: Quantitative comparison between ours and [8] on **wave1 (Gaussian)**.

Frame	Method	RMSE ↓	EPE ↓	AAE ^o ↓	SILog ↑
0	Ours	0.64	0.79	82.41	0.95
	[8]	0.64	0.77	91.24	
1	Ours	0.63	0.74	18.38	0.48
	[8]	0.67	0.79	21.92	
2	Ours	0.63	0.75	15.93	0.50
	[8]	0.64	0.77	14.77	
3	Ours	0.62	0.75	22.19	0.47
	[8]	0.64	0.76	19.65	
4	Ours	0.61	0.75	21.85	0.41
	[8]	0.65	0.77	21.14	
5	Ours	0.65	0.80	20.83	0.44
	[8]	0.74	0.85	22.60	
6	Ours	0.65	0.81	21.10	0.55
	[8]	0.71	0.88	23.57	
7	Ours	0.63	0.79	22.95	0.58
	[8]	0.67	0.79	23.14	
8	Ours	0.67	0.82	23.47	0.58
	[8]	0.65	0.79	23.58	
9	Ours	0.66	0.81	19.07	0.41
	[8]	0.64	0.78	18.20	
Average	Ours	0.64	0.78	26.82	0.54
	[8]	0.66	0.80	27.98	

Table 2: Quantitative comparison between ours and [8] on **wave2 (ripple)**.

Frame	Method	RMSE ↓	EPE ↓	AAE ^o ↓	SILog ↑
0	Ours	0.54	0.65	22.49	0.56
	[8]	0.54	0.63	23.59	
1	Ours	0.53	0.63	18.43	0.66
	[8]	2.24	2.81	113.00	
2	Ours	0.53	0.64	18.71	0.83
	[8]	0.92	1.11	34.87	
3	Ours	0.53	0.64	20.02	0.69
	[8]	0.82	1.01	31.11	
4	Ours	0.54	0.65	34.34	0.65
	[8]	0.81	1.00	36.31	
5	Ours	0.53	0.63	29.78	0.62
	[8]	0.85	1.05	59.72	
6	Ours	0.53	0.63	25.08	0.60
	[8]	0.83	1.04	45.68	
7	Ours	0.54	0.63	23.01	0.65
	[8]	0.73	0.91	34.09	
8	Ours	0.53	0.62	22.00	0.73
	[8]	0.71	0.85	31.49	
9	Ours	0.53	0.64	25.57	0.77
	[8]	0.74	0.91	32.30	
Average	Ours	0.53	0.64	23.94	0.68
	[8]	0.92	1.13	44.22	

Table 3: Quantitative comparison between ours and [8] on **wave3 (ripple)**.

Frame	Method	RMSE ↓	EPE ↓	AAE ^o ↓	SILog ↑
0	Ours	0.32	0.39	9.96	0.59
	[8]	0.54	0.65	17.44	
1	Ours	0.33	0.40	10.45	0.60
	[8]	1.63	2.11	85.65	
2	Ours	0.34	0.41	10.92	0.60
	[8]	1.18	1.51	52.86	
3	Ours	0.34	0.41	10.69	0.61
	[8]	1.03	1.31	45.19	
4	Ours	0.33	0.41	10.64	0.58
	[8]	1.08	1.38	45.47	
5	Ours	0.33	0.40	10.03	0.52
	[8]	1.21	1.60	51.80	
6	Ours	0.35	0.42	11.63	0.52
	[8]	1.34	1.75	58.51	
7	Ours	0.33	0.40	11.81	0.58
	[8]	1.32	1.72	63.68	
8	Ours	0.34	0.41	12.33	0.62
	[8]	1.22	1.58	65.13	
9	Ours	0.31	0.38	12.04	0.62
	[8]	1.03	1.33	58.30	
Average	Ours	0.33	0.40	11.05	0.58
	[8]	1.16	1.49	54.40	

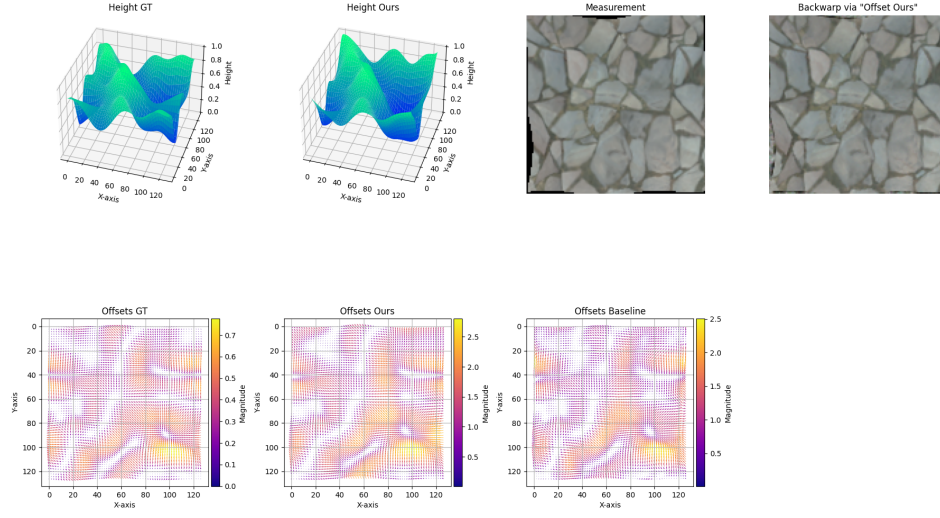


Figure 2: Example of distortion and surface height estimation on wave 1 (Gaussian). Our method outperforms [8] (see Table 1). The full video is provided with the supplementary materials.

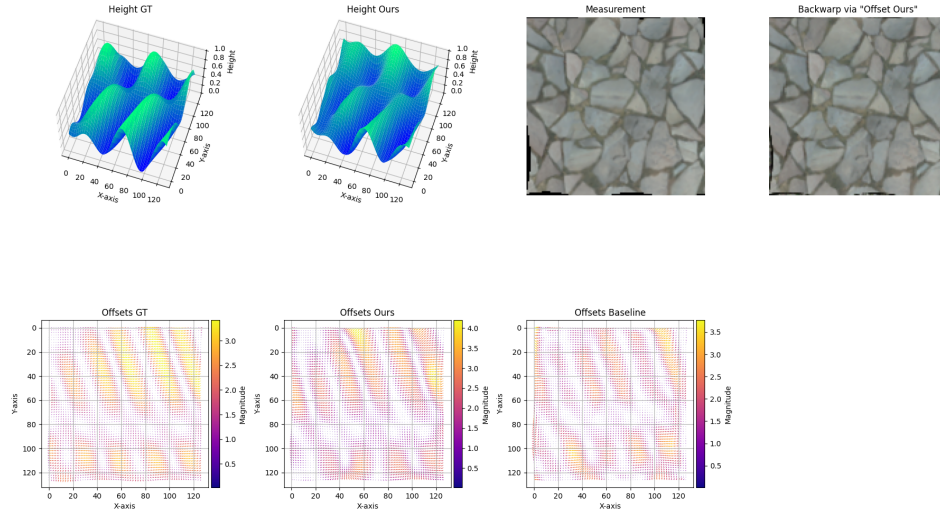


Figure 3: Example of distortion and surface height estimation on wave 2 (ripple). Our method outperforms [8] (see Table 2). The full video is provided with the supplementary materials.

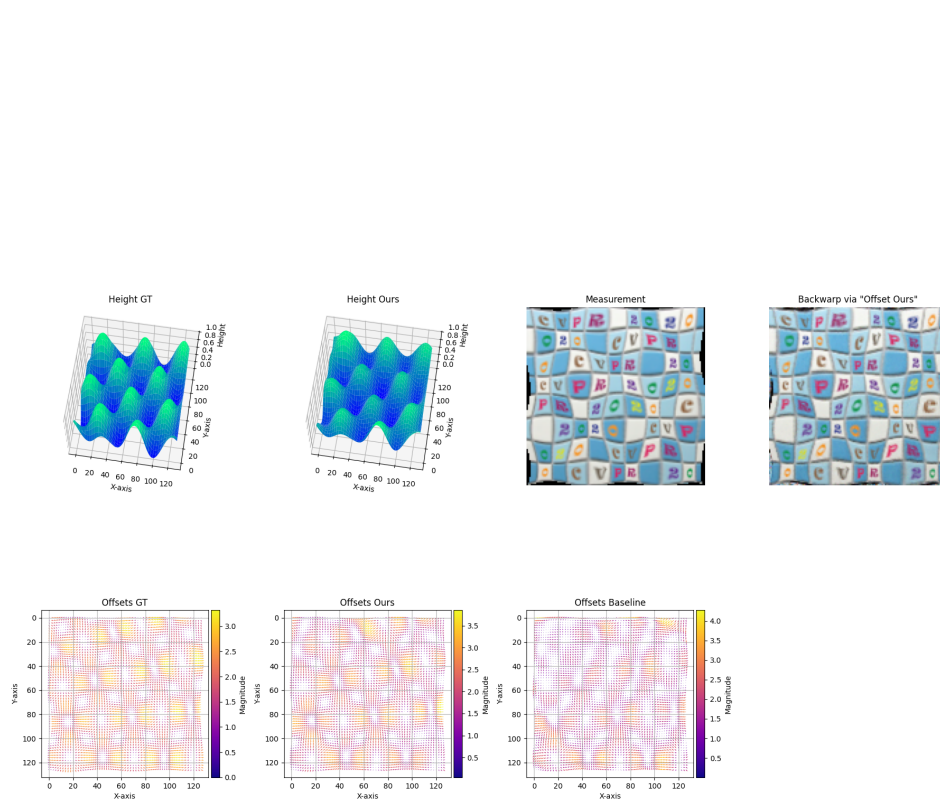


Figure 4: Example of distortion and surface height estimation on wave 3 (ripple). Our method outperforms [8] (see Table 3). The full video is provided with the supplementary materials.

62 References

- 63 [1] D. Eigen, C. Puhrsch, and R. Fergus. Depth map prediction from a single image using a
64 multi-scale deep network. *NeurIPS*, 27, 2014.
- 65 [2] J. G. James, P. Agrawal, and A. Rajwade. Restoration of non-rigidly distorted underwater images
66 using a combination of compressive sensing and local polynomial image representations. In
67 *ICCV*, pages 7839–7848, 2019.
- 68 [3] V. Sitzmann, J. Martel, A. Bergman, D. Lindell, and G. Wetzstein. Implicit neural representations
69 with periodic activation functions. *NeurIPS*, 33:7462–7473, 2020.
- 70 [4] M. Tancik, P. Srinivasan, B. Mildenhall, S. Fridovich-Keil, N. Raghavan, U. Singhal, R. Ra-
71 mamanoorthi, J. Barron, and R. Ng. Fourier features let networks learn high frequency functions in
72 low dimensional domains. *Advances in neural information processing systems*, 33:7537–7547,
73 2020.
- 74 [5] S. Thapa, N. Li, and J. Ye. Dynamic fluid surface reconstruction using deep neural network. In
75 *CVPR*, pages 21–30, 2020.
- 76 [6] B. Ummenhofer, H. Zhou, J. Uhrig, N. Mayer, E. Ilg, A. Dosovitskiy, and T. Brox. Demon:
77 Depth and motion network for learning monocular stereo. In *CVPR*, pages 5038–5047, 2017.