# Leveraging Near-Field Lighting for Monocular Depth Estimation from Endoscopy Videos

Akshay Paruchuri[†], Samuel Ehrenstein, Shuxian Wang, Inbar Fried, Stephen M. Pizer, Marc Niethammer, and Roni Sengupta[†]

Department of Computer Science
University of North Carolina at Chapel Hill

## A Overview of Appendices

Our appendices contain the following additional details and results:

- In Sec. B, we present point cloud results from our approach. We also provide point cloud results from LightDepth [7] for comparison.
- In Sec. C, we provide additional information regarding our approach. Sec. C.1 includes additional details about the proposed per-pixel lighting (PPL) representation and its usage in our proposed approach. In Sec. C.2, we provide additional details regarding our approach to teacher-student learning, including a full algorithm table. Sec. C.3 contains additional details regarding our implementation, including our chosen loss weights and backbone model size.
- Sec. D includes additional information regarding our experiments, chiefly the training splits we utilized that match [7], as well as additional qualitative results.
- Sec. E describes our limited code release included as a part of these supplementary materials. The code includes our proposed loss functions and model files for reference, as well as pre-trained models. We will release our full code, including the training code and various baselines, in the near future.
- Sec. F includes additional information regarding our implementation of LightDepth [7].
- We provide additional, qualitative results on bronchoscopy data in Sec. G.
- Sec. H includes additional details regarding the clinical dataset utilized in our work and to be released in the near future.

## B Point Clouds

We present point cloud results from our monocular depth estimation approach in order to show that our produced depths can eventually be used for the task of 3D mesh reconstruction and subsequent 3D analysis. In Fig. 1 it can be seen that our produced point clouds are superior to the point clouds produced by our re-implementation of LightDepth [7]. We also provide the raw *.ply* files for each point cloud in Fig. 1 as a part of our supplementary materials.

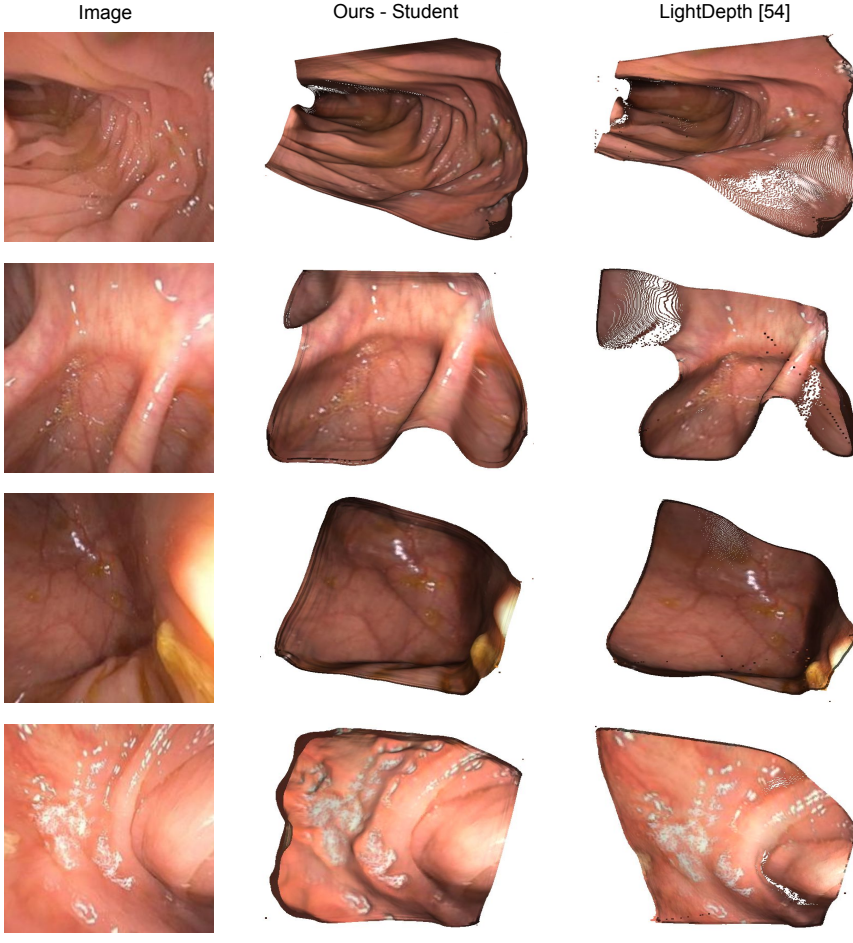


**Fig. 1:** A qualitative view of 3D point clouds generated by our approach and in contrast to LightDepth [7]. We also provide the raw *.ply* files for each point cloud as a part of our supplementary materials.

# C Additional Info on Methods

## C.1 Per-Pixel Lighting Representation and Losses

In our main paper, we noted that we utilize the conventional pinhole camera framework, positioned at the origin in the global coordinate system and looking down the z-axis. This camera is defined by the intrinsic matrix $K$. A point in the world space, denoted as $X = (x, y, z)$, is mapped to a pixel coordinate $(u, v)$ using:

$$(u, v, 1)^T \sim K(x, y, z)^T . \tag{1}$$

Subsequently, we noted that we only take into account visible surfaces of the colon, and therefore assume that the depth map and corresponding normal map is a complete description of in-view surfaces. $X(u, v) \in \mathbb{R}^3$ describes the 3D location of a pixel $(u, v)$ on the surface of the organ. The depth map can then be defined by $D(u, v) = X(u, v)_3$, corresponding to the z-component of $X(u, v)$. $X(u, v)$ itself can be recovered from the following:

$$X(u, v) = D(u, v)K^{-1}(u, v, 1)^T . \tag{2}$$

For completeness, we now also note that given $n(X)$ is the normal at the point $X$, then the normal map can be defined by $N(u, v) = n(X(u, v))$. $N$ is subsequently computed based on the following:

$$N = \frac{\frac{\partial X}{\partial u} \times \frac{\partial X}{\partial v}}{\left\| \frac{\partial X}{\partial u} \times \frac{\partial X}{\partial v} \right\|} . \tag{3}$$

Additionally, we note that our learning objective $\mathcal{L}_{PPS-corr}$ can be formulated as follows:

$$\mathcal{L}_{PPS-corr} = 1 - \frac{\sum_{h=1}^{H} \sum_{w=1}^{W} (I_{gray_{hw}} - \bar{I})(PPL_{F_{hw}} - \overline{PPL}_F)}{\sqrt{\sum_{h=1}^{H} \sum_{w=1}^{W} (I_{gray_{hw}} - \bar{I})^2 \sum_{hw} (PPL_{F_{hw}} - \overline{PPL}_F)^2}} \tag{4}$$

The simplified version of this formulation is presented in our main paper as follows:

$$\mathcal{L}_{PPS-corr} = 1 - \text{corr}(I_{gray}, PPL_F) \tag{5}$$

As noted in the main paper, the self-supervised loss function variant will enable us to train on real clinical data where ground-truth depth information is unavailable. Code implementations of both our supervised and self-supervised loss function variants are included as a part of these supplementary materials.

## C.2 Teacher-Student Transfer Learning for Sim2Real

Our approach with teacher-student training on labeled data $\mathcal{D}_L$ and unlabeled data $\mathcal{D}_U$ can be summarized in algo. 1 as follows:

Where the losses used alongside our proposed losses $\mathcal{L}_{PPS-sup}$ and $\mathcal{L}_{PPS-corr}$ correspond to the scale-shift invariant ($\mathcal{L}_{SSI}$), regularization ($\mathcal{L}_{REG}$), and virtual-normal ($\mathcal{L}_{VNL}$) losses described in prior works [3, 6].

**Algorithm 1** Teacher-Student Learning for Endoscopic Data

**Require:** labeled synthetic dataset: $\mathcal{D}_L$, unlabeled real dataset: $\mathcal{D}_U$
Initialize teacher and student models with identical architectures:
    $M_T$ (Teacher), $M_S$ (Student) $\leftarrow InitializeModels()$
Train $M_T$ on $\mathcal{D}_L$: $M_T \leftarrow Train(M_T, \mathcal{D}_L)$
    $\mathcal{L} = \alpha_{SSI}\mathcal{L}_{SSI} + \alpha_{REG}\mathcal{L}_{REG} + \alpha_{VNL}\mathcal{L}_{VNL} + \alpha_{PPS-sup}\mathcal{L}_{PPS-sup}$
Freeze $M_T$ to prevent further updates
Prepare mixed dataset $\mathcal{D}_M$ combining $\mathcal{D}_L$ and $\mathcal{D}_U$.
**for** each batch $b$ in $\mathcal{D}_M$ **do**
    **if** $b$ is from $\mathcal{D}_L$ **then**
        Train $M_S$ on $\mathcal{D}_L$: $M_S \leftarrow Train(M_S, \mathcal{D}_L)$
            $\mathcal{L} = \alpha_{SSI}\mathcal{L}_{SSI} + \alpha_{REG}\mathcal{L}_{REG} + \alpha_{VNL}\mathcal{L}_{VNL} + \alpha_{PPS-sup}\mathcal{L}_{PPS-sup}$
    **else if** $b$ is from $\mathcal{D}_U$ **then**
        Train $M_S$ on $\mathcal{D}_U$: $M_S \leftarrow Train(M_S, \mathcal{D}_U)$
            $\mathcal{L} = \alpha_{SSI}\mathcal{L}_{SSI*} + \alpha_{REG}\mathcal{L}_{REG*} + \alpha_{VNL}\mathcal{L}_{VNL*} + \alpha_{PPS-corr}\mathcal{L}_{PPS-corr}$
            Compute $\mathcal{L}_{SSI*}$, $\mathcal{L}_{REG*}$, and $\mathcal{L}_{VNL*}$ using pseudo-supervision from $M_T$.
    **end if**
**end for**

### C.3 Implementation

As a part of training our approach, we use the following $\alpha$ values that describe loss weights as shown in algo. 1: $\alpha_{SSI} = 1.0$, $\alpha_{REG} = 0.1$, $\alpha_{VNL} = 10.0$, $\alpha_{PPS-sup} = 0.1$, and $\alpha_{PPS-corr} = 1.0$. We utilize the ViT-Small version of the backbone architecture that's also utilized in Depth Anything [8].

## D Experiments

### D.1 Dataset Splits

We utilized the same dataset splits as LightDepth [7] and include the exact sequences used for the C3VD [2] dataset in Tab. 1. Our clinical dataset includes 80 sequences with oblique views (7,293 frames), 14 sequences with en-face views (832 frames), and 20 sequences with down-the-barrel (axial) views (10,216 frames). The oblique views and en-face views are from a to-be-released dataset described in Sec. H. The down-the-barrel (axial) views are from the Colon10K [5] dataset. The exact clinical sequences used for training and testing are included as *train.txt* and *val.txt* files in our supplementary materials folder.

**Table 1:** Dataset Splits for C3VD [2]

| Sequence | Texture | Video | Frames | Set |
|---|---|---|---|---|
| Cecum | 1 | b | 765 | Train |
| Cecum | 2 | b | 1120 | Train |
| Cecum | 2 | c | 595 | Train |
| Cecum | 4 | a | 465 | Train |
| Cecum | 4 | b | 425 | Train |
| Sigmoid Colon | 1 | a | 800 | Train |
| Sigmoid Colon | 2 | a | 513 | Train |
| Sigmoid Colon | 3 | b | 536 | Train |
| Transcending | 1 | a | 61 | Train |
| Transcending | 1 | b | 700 | Train |
| Transcending | 2 | b | 102 | Train |
| Transcending | 2 | c | 235 | Train |
| Transcending | 3 | b | 214 | Train |
| Transcending | 4 | b | 595 | Train |
| Descending Down | 4 | a | 74 | Train |
| Cecum | 1 | a | 276 | Test |
| Cecum | 2 | a | 370 | Test |
| Cecum | 3 | a | 730 | Test |
| Sigmoid | 3 | a | 610 | Test |
| Transcending | 2 | a | 194 | Test |
| Transcending | 3 | a | 250 | Test |
| Transcending | 4 | a | 384 | Test |
| Descending Up | 4 | a | 74 | Test |

### D.2 Qualitative Results

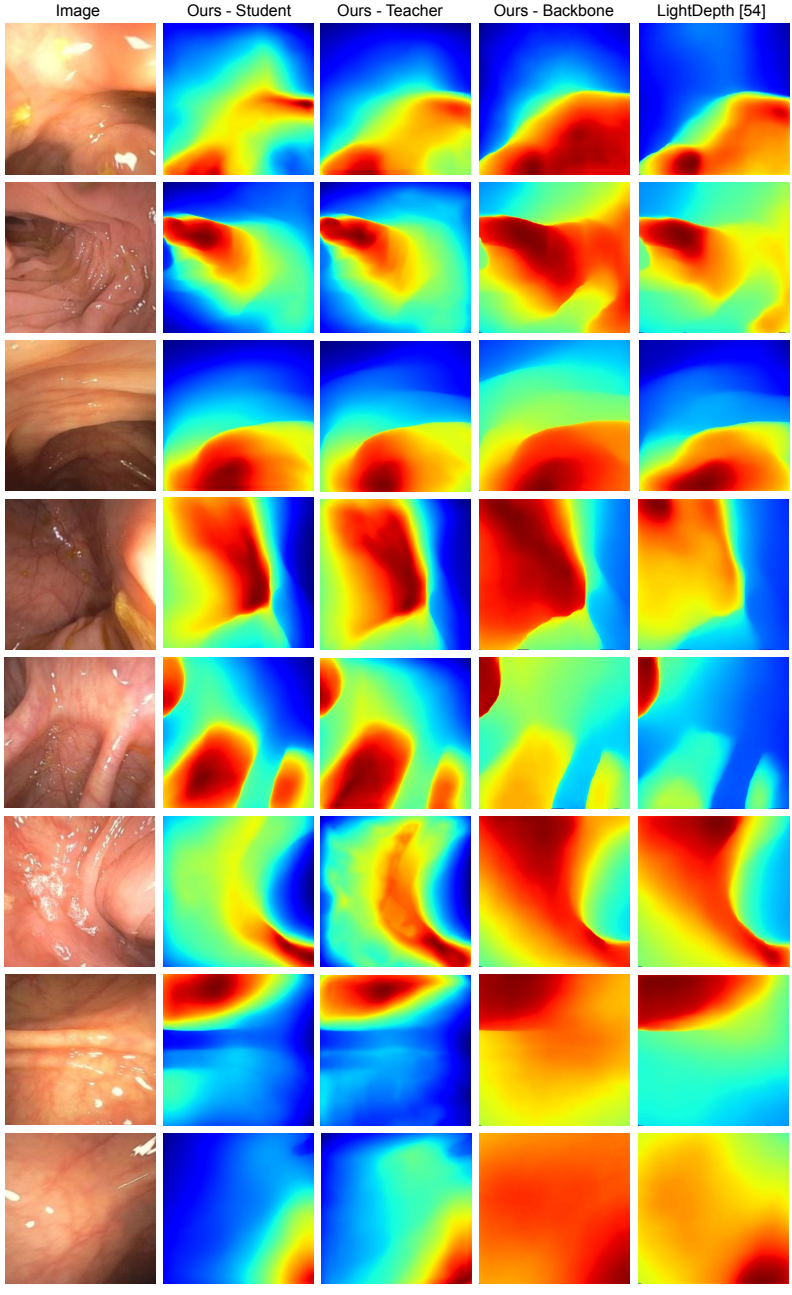


**Fig. 2:** Qualitative evaluation on clinical data. Red = further distance from the camera and blue is closer.

## E Limited Code Release

Our limited code release can be accessed via a link to a GitHub repo on our project website: `https://ppsnet.github.io/`. The GitHub repo contains a *PPSNet.py* file that serves as the full model file for our proposed approach. *calculate_PPL.py* from [4] is also included for reference alongside our proposed loss functions in *PPS_losses.py*. We also provide limited evaluation code for the C3VD [2] dataset and pre-trained models. We will release our full codes, including the training code and various baselines, in the near future.

## F LightDepth Implementation

For the purpose of comparison to state-of-the-art monocular depth estimation specific to endoscopy, we attempted to implement LightDepth [7] on our own, as the authors did not release their code. The authors describe an architecture with two ViT branches: one initialized with the weights from DPT-Hybrid, used for depth prediction; and one trained from scratch, used for prediction of albedo. Specifically, this albedo predictor computes hue and saturation values for each pixel, and then converts to RGB color space from HSV assuming V=100%. We were unable to successfully implement this two-headed depth and albedo prediction. Instead, we compute the albedo $\rho$ by analytically solving the per-pixel rendering equation used (eq. 3 in [7]):

$$\mathcal{I}(d_i, \rho_i, g) = \left( \frac{\sigma_0}{||d_i \mathbf{r}_i - \mathbf{x}_l||^2} R(\psi_i) \cos\theta_i \rho_i g \right)^{1/\gamma}$$

with $\psi_i$, $R(\psi_i)$, and $\theta_i$ depending only on constants and the predicted depth map (whose value at pixel $i$ is $d_i$).

Assuming a colocated light and camera and setting $\sigma_0 = g = 1$, we can rearrange to get:

$$\rho_i = \frac{d_i^2 \mathcal{I}_i^\gamma}{R(\psi_i) \cos\theta_i}$$

where $\mathcal{I}_i$ is the R, G, or B value of the $i$th pixel, and we apply this separately for each channel. This calculated $\rho_i$ replaced the predicted albedo $\rho_i$ in all loss calculations. We additionally add a regularization loss

$$\mathcal{L}_{albvar} = \frac{1}{3}(\mathrm{Var}(\rho^R) + \mathrm{Var}(\rho^G) + \mathrm{Var}(\rho^B))$$

where $\mathrm{Var}(\rho^R)$ is the variance of the R channel calculated albedo value over the entire batch, and likewise for G and B. This provided a slight improvement in results, the idea being that in endoscopy the surface albedo should be fairly uniform.

Starting from the pretrained DPT-Hybrid weights from [3], we fine-tuned our LightDepth implementation for a single epoch on the train split of C3VD used in [7] at a learning rate of 4e-6, as any further training was found to overfit and reduce accuracy.

## G Qualitative Results on Bronchoscopy

In Fig. 3, we present qualitative depth map results on bronchoscopy data. Despite not on bronchoscopy data, we note that 'Ours - Student' and 'Ours - Backbone' both provide superior results to the baseline provided by Depth Anything [8], with 'Ours - Student' being slightly better than 'Ours - Backbone' in certain areas with relatively farther depths. We plan to explore bronchoscopy data as an additional bio-application in subsequent work. We will release a small dataset of bronchsocopy frames for qualitative evaluation upon this work's acceptance.

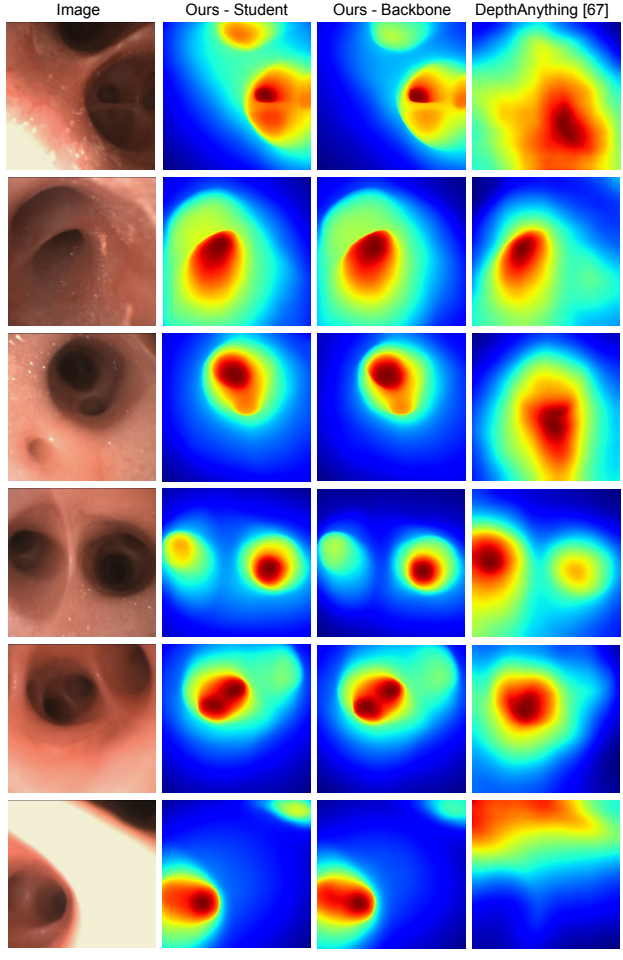


**Fig. 3:** Qualitative evaluation on bronchoscopy data. Red = further distance from the camera and blue is closer. Note that 'Ours - Student' and 'Ours - Backbone' provide significantly higher quality depth estimations than Depth Anything [8].

## H Clinical Dataset

The clinical dataset used in our work consists of 80 sequences with oblique views (7,293 frames), 14 sequences with en-face views (832 frames), and 20 sequences with down-the-barrel (axial) views (10,216 frames). The down-the-barrel (axial) views are from the Colon10K [5] dataset. The oblique views and en-face views are from a dataset which will be released in a separate submission [1] to a different conference. The anonymized submission in question is included as a part of our supplementary materials.

## References

1. Anonymous: Structure-preserving image translation for depth estimation in colonoscopy (2024), paper submitted to another conference that releases a portion of the clinical data used in this paper.
2. Bobrow, T.L., Golhar, M., Vijayan, R., Akshintala, V.S., Garcia, J.R., Durr, N.J.: Colonoscopy 3d video dataset with paired depth from 2d-3d registration. Medical Image Analysis **90**, 102956 (Dec 2023). https://doi.org/10.1016/j.media.2023.102956, http://dx.doi.org/10.1016/j.media.2023.102956
3. Eftekhar, A., Sax, A., Bachmann, R., Malik, J., Zamir, A.: Omnidata: A scalable pipeline for making multi-task mid-level vision datasets from 3d scans (2021)
4. Lichy, D., Sengupta, S., Jacobs, D.W.: Fast light-weight near-field photometric stereo (2022)
5. Ma, R., McGill, S.K., Wang, R., Rosenman, J., Frahm, J.M., Zhang, Y., Pizer, S.: Colon10k: a benchmark for place recognition in colonoscopy. In: 2021 IEEE 18th International Symposium on Biomedical Imaging (ISBI). pp. 1279–1283. IEEE (2021)
6. Ranftl, R., Lasinger, K., Hafner, D., Schindler, K., Koltun, V.: Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer (2020)
7. Rodriguez-Puigvert, J., Batlle, V.M., Montiel, J.M.M., Martinez-Cantin, R., Fua, P., Tardos, J.D., Civera, J.: Lightdepth: Single-view depth self-supervision from illumination decline (2023)
8. Yang, L., Kang, B., Huang, Z., Xu, X., Feng, J., Zhao, H.: Depth anything: Unleashing the power of large-scale unlabeled data (2024)