

# DarkGS: Learning Neural Illumination and 3D Gaussians Relighting for Robotic Exploration in the Dark

Tianyi Zhang<sup>1</sup>, Kaining Huang<sup>1</sup>, Weiming Zhi<sup>1</sup>, Matthew Johnson-Roberson<sup>1</sup>

**Abstract**—Humans have the remarkable ability to construct consistent mental models of an environment, even under limited or varying levels of illumination. We wish to endow robots with this same capability. In this paper, we tackle the challenge of constructing a photorealistic scene representation under poorly illuminated conditions and with a moving light source. We approach the task of modeling illumination as a learning problem, and utilize the developed illumination model to aid in scene reconstruction. We introduce an innovative framework that uses a data-driven approach, *Neural Light Simulators* (NeLiS), to model and calibrate the camera-light system. Furthermore, we present DarkGS, a method that applies NeLiS to create a relightable 3D Gaussian scene model capable of real-time, photorealistic rendering from novel viewpoints. We show the applicability and robustness of our proposed simulator and system in a variety of real-world environments. Code released at <https://tyz1030.github.io/proj/darkgs.html>

## I. INTRODUCTION

Robots and autonomous vehicles have been routinely deployed in poorly illuminated environments for critical missions and tasks such as exploration, inspection, transportation, search and rescue, etc. (see Fig. 1a). Imaging systems consisting of one or multiple RGB cameras and light sources are often equipped on the robot to illuminate and sense the surrounding environment. The streamed image sequence can be further used in downstream tasks, e.g. navigation, mapping, and visualization, to boost the robot autonomy and human understanding of the environment.

Scene reconstruction, or the capability to create accurate internal representations of the environment, is vital for robots operating in unknown environments. Previous vision-only approaches largely rely on identifying common feature points over a set of multiple-view images, and then minimizing a reprojection error [4]. Such procedures like Structure-from-Motion (SfM) or Simultaneous Localization and Mapping (SLAM) also estimate the camera poses of the images. Based on these camera poses, Neural Radiance Field (NeRF) [5] is capable of achieving photorealistic scene reconstruction by optimizing a photometric loss between the representation and the images. However, while achieving huge success in the graphics community, the transition of objective function from reprojection error to photometric loss has raised a new challenge to the robotics community: Can we still build consistent scene representations with a moving light source on the robot platform?

<sup>1</sup>Authors are with the Robotics Institute, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213, USA {tianyiz4, kaining2, wzhi, mkj}@andrew.cmu.edu

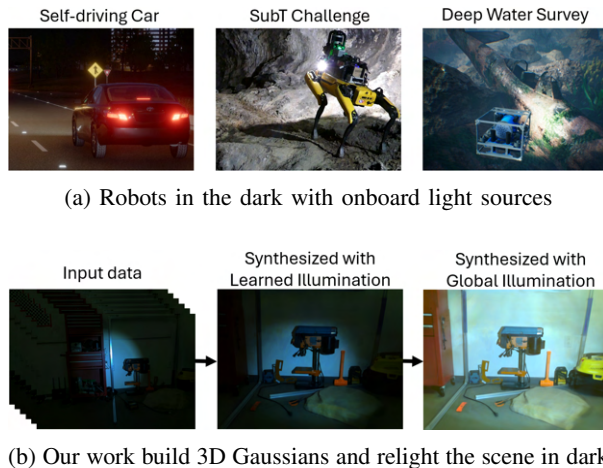


Fig. 1: Robotic imaging systems working in the dark consist of cameras and light sources. Examples as shown in (a): Carla Simulator [1], Team CoStar in SubT Challenge [2] and HoloOcean underwater robot simulator [3]. In this work, we propose a pipeline that calibrates the camera-light system which helps photorealistic scene reconstruction and relighting from images collected in the dark.

In this work, we identify *illumination-inconsistency* as the main challenge in building a photorealistic scene representation from images taken with a moving light source. That is, as the robot operates in the environment, the effect of moving light source results in images captured of the same region appearing visually different. Handcrafted feature descriptors that serve as backbone in SfM and SLAM [6]–[8] are designed to be invariant to changes in pixel intensities. Such changes also have limited effects on the optimization pipeline, which uses reprojection errors as objective functions. Hence, the classic SfM methods are generally robust against *illumination-inconsistency*. However, on the way towards photorealistic scene reconstruction and switching to optimization of the photometric loss, we find that state-of-the-art NeRF methods [5], [9] fail on the images taken in the dark with a moving light source. Although variants such as RawNeRF [10] and Relightable 3DG [11] claim to be able to handle dark images in RAW format and model the reflection properties of the scene, they still fail to handle the varying illumination.

Concretely, the problem tackled in this paper is as such: Given a sequence of images taken in poorly illuminated environments, with one major light source moving with one camera as a rigid body, reconstruct the scene by minimizing

photometric loss and achieve photorealistic novel view image synthesis (Fig. 1b). Our contributions are as follows:

- 1) A pipeline that consists of light source modeling, camera-light calibration, building 3D Gaussians and scene relighting from illumination-inconsistent images.
- 2) Neural Light Simulators (NeLiS), a data-driven and physically interpretable illumination model and software for light source modeling and calibration.
- 3) Dark Gaussian Splatting (DarkGS), a variant of the 3D Gaussian Splatting (3DGS) model that builds photorealistic scene representations under poorly illuminated conditions and relights the scene with global illumination, based on COLMAP [12] and NeLiS results.

## II. RELATED WORK

### A. Extroceptive Sensor Calibration on Robots

Autonomous robots are usually equipped with perceptual sensors such as cameras and LiDARs. Taking camera calibration as an example, a calibration target with AprilTag [13] or checkerboard pattern is often used. With observations of the target from different perspectives, the focal length, center of projection, distortion coefficient, and pose of the camera can be estimated [14]. Similar target-based approaches have been used to calibrate LiDARs [15], radars [16] and acoustic sensors [17] for downstream sensor fusion tasks [18]. Analogous to sensor calibrations mentioned above, in this paper, we propose to use a target consisting of AprilTags [13] and blank space for calibrating the light in a camera-light system, including estimating the transformation between camera's and light's coordinate system, radiant intensity distribution (RID) and light fall-off curve of the light source.

### B. Light Calibration

Existing methods use various kinds of customized calibration targets: [19][20] propose to use a target of AprilTags and pins to infer the position of point light source from the shadows of the pins. Alternatively, [21][22] propose to use a Lambertian sphere and estimate the light source parameters by learning to reconstruct the sphere. Furthermore, [23] shows that the location of the light source can also be recovered from the shadow of a sphere. With a Lambertian plane and customized markers, [24] shows that it is possible to calibrate a point light source with a single image. [25] shows the most relevance to our work, which calibrates the pose of a light source and the metric scale given the RID curve. However, the above work either assumes an oversimplified model, e.g. a point light source with inverse square light falloff, in a pure dark environment, or that the RID is given, which makes the method less applicable. In this work, we use a minimum calibration target design that consists of only AprilTags [13] and a Lambertian plane similar to [25]. With NeLiS we show that modeling and learning RID, light falloff and ambient light together with camera-light pose can improve the quality of image reconstruction and make our calibration method more feasible for real-world robot deployments.

### C. NeRF and 3D Gaussian Splatting

Based on the success of SfM [4], NeRF learns to represent scenes using a continuous function, e.g. a multilayer perceptron (MLP), and can achieve photorealistic novel-view image synthesis [26]. Popular variations of NeRF such as Mip-NeRF[27] and Instant-NGP [28] have improved NeRF by rendering accuracy and speed. Recent innovation introduced as 3DGS [9] has revolutionized this field by replacing the neural network backbone with spatial Gaussians, allowing rendering speed of 100+ fps while maintaining the accuracy and differentiability of the system. For our robotic setup, the problem of varying illumination is attempted by [29], [30] which model the physical property of the scene, e.g. roughness, albedo, and normal, instead of the radiance as a constant. However, these works oversimplify the light model as a Lambertian point light source, co-centered with the camera, making it not applicable on many real robotic platforms. Gaussianshader [31] and Relightable 3DG [11] introduce physical properties into the 3DGS framework but model illumination as a constant that does not change frame-by-frame. According to our experiments, none of the above mentioned methods handles the illumination-inconsistency issue on a real-robot imaging system. In our proposed DarkGS, by modeling the physical property of the scene and taking advantage of NeLiS, we can not only build a consistent 3DGS from poorly illuminated images, but also relight the scene with global illumination.

### D. Photometric Stereo

Photometric stereo refers to the estimation of the shape of an object with multiple images taken from a single viewpoint with different lighting orientations [32]. Early approaches confine this problem to known light source and Lambertian surfaces with uniform albedo. Further work has allowed spatially varying albedo [33] and unknown light [34]. However, our robotic setup differs from this problem by the fact that our camera and light source always move together as a rigid body. And we choose to tackle this problem within the framework of 3DGS [9] which allows us to synthesize novel views with photorealistic quality.

## III. METHODOLOGY

Our proposed workflow for building NeLiS and DarkGS is shown in Fig. 2. We build NeLiS by calibrating the camera-light system at a planar calibration target. By accounting for the calibration image sequence and hand measurement of the light pose, an in-the-loop human operator can initialize the light parameters using our intuitive GUI. Since the calibration target can be localized by AprilTags [13], a shading model can render the image with the given parameters to approximate the calibration image. Then, we run the optimization to minimize the photometric error of the calibration image sequence and rendered image. After obtaining a calibrated NeLiS, the robot can be deployed in a dark environment, using the collected images to build a DarkGS, finetune the model, and relight the scene.

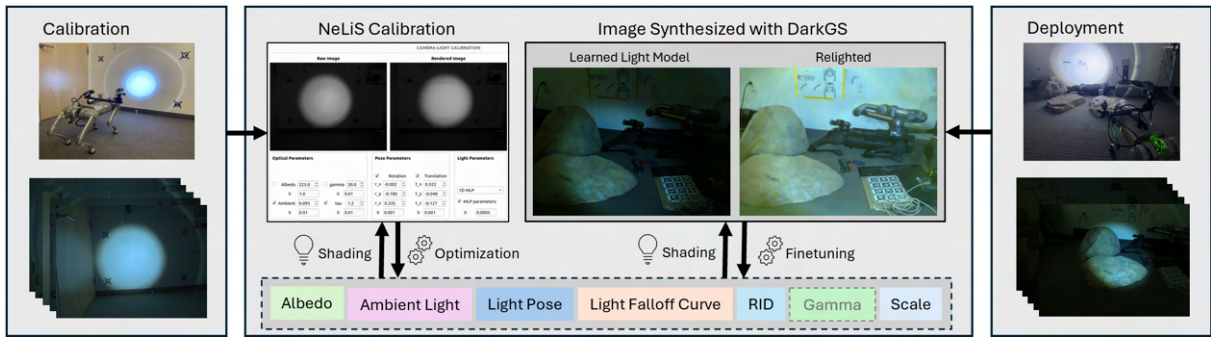


Fig. 2: Our proposed workflow: Images for camera-light calibration are first collected at a calibration target. With NeLiS, we manually initialize the parameters and then optimize the light model. The model can then be used to build DarkGS, present the scene with learned or relighted illumination. (Learning Gamma tone mapping is supported by NeLiS but not further discussed in this paper.)

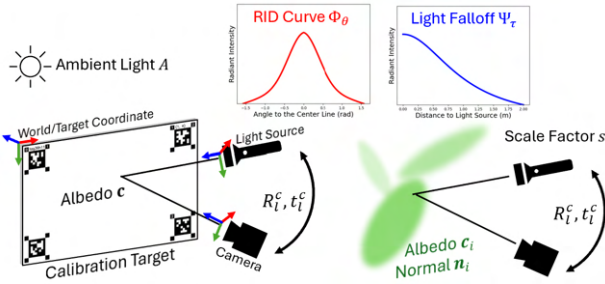


Fig. 3: Our shading model: (Left) In NeLiS, camera poses are localized by AprilTags on the calibration target. The pose of light  $R_l^c$  and  $t_l^c$ , albedo  $\mathbf{c}$  of the calibration target, ambient light  $A$ , RID  $\Phi_\theta$ , and light falloff function  $\Psi_\tau$  will be learned. (Right) In building DarkGS, each Gaussian  $g_i$  is modeled with an albedo  $\mathbf{c}_i$  and normal  $\mathbf{n}_i$ . The ambient light  $A$  and scale  $s$  will also be optimized in this process.

In this section, we will discuss the details of the NeLiS model in III-A, DarkGS in III-B and NeLiS frontend software and workflow in III-C.

### A. Shading Model for NeLiS

The calibration data is taken by capturing photos at a calibration target from different views while the light source moves with the camera as a rigid body. The calibration target is a white plane with four AprilTags positioned at four corners of a rectangle (as shown in Fig. 3). We attach the origin of the world coordinate to the top-left corner of the calibration target. We assume that the camera is precalibrated with distortions removed. One of the key problems for NeLiS to solve is estimating the relative pose of light to the camera  $R_l^c \in SO(3)$  and  $t_l^c \in \mathbb{R}^3$ .

Given the true size of the calibration target, we can apply Perspective-n-Point algorithm (PnP) [35] to extract  $R_c^w \in SO(3)$  and  $t_c^w \in \mathbb{R}^3$  which transform points from the camera coordinate to the world coordinate. The position of the light source is then given by  $\mathbf{o}_l = R_c^w t_l^c + t_c^w \in \mathbb{R}^3$ . Since we align the  $z$  axis of the light coordinate with the centerline of the light, the direction of the centerline can be denoted by

$\omega_l = R_c^w R_l^c [0, 0, 1]^T \in \mathbb{R}^3$ . Both  $\mathbf{o}_l$  and  $\omega_l$  are in the world coordinate frame.

We only use the area bounded by 4 AprilTags as the Region of Interest (ROI) to do calibrations. We assume that this area is a Lambertian plane and has the same normal  $\mathbf{n} \in \mathbb{R}^3$  and diffusive albedo  $\mathbf{c} \in \mathbb{R}^\lambda$  anywhere on the plane ( $\lambda = 1$  for grayscale images and  $\lambda = 3$  for RGB). For each pixel in the ROI, we find the intersection of the corresponding camera ray and the target plane in the world coordinate system, denoted by  $\mathbf{x} \in \mathbb{R}^3$ . To infer the incident radiance at  $\mathbf{x}$ , one needs to model the RID, light falloff function, and ambient light.

1) *RID*: RID is commonly modeled as a function of the angle between the centerline  $\omega_l$  of the light and light ray  $\omega_x = \mathbf{x} - \mathbf{o}_l$ . Previous work assume that RID is given [22], [24], [25]. However, this assumption may not hold for in-the-wild robot deployment. Instead, we remove this dependency by learning a neural RID from calibration data:

$$\Phi_\theta(\mathbf{x}) = \text{MLP}_\theta(\cos^{-1}(\frac{\omega_x}{\|\omega_x\|_2} \cdot \omega_l)) \quad (1)$$

here  $\theta$  denotes the parameters of the MLP.

2) *Light Falloff Curve*: Inverse square law is widely used to model light falloff, based on the assumption of a point light source. However, when objects are closer to the light source, the inverse square law starts to fail. We choose to model the light falloff in the form of a Lorentzian function of the distance  $\|\omega_x\|_2$ , as suggested by [36]:

$$\Psi_\tau(\mathbf{x}) = \frac{1}{\tau + \|\omega_x\|_2^2} \quad (2)$$

Instead of estimating  $\tau$  from hand measurement of the light surface [36], we designate it to be a learnable parameter.

3) *Ambient Light*: Imaging and lighting problems are often studied in a dark room to better simplify and constrain the model by removing ambient light. However, a perfectly dark space for calibration might not be accessible for real-world robot deployment. Instead, we model ambient light as a learnable parameter  $A$ . The incident radiance at the point  $\mathbf{x}$  can thereby be modeled as:

$$I_{\mathbf{x}} = \Psi_{\tau}(\mathbf{x})\Phi_{\theta}(\mathbf{x}) + A \quad (3)$$

4) *BRDF*: Although there are bidirectional reflectance distribution function (BRDF) models capable of handling complex reflection effects, they exceed the scope of this paper. We opt for the Lambertian reflection model, eliminating the need to optimize any parameters in our BRDF. We use  $f_r(\omega_{\mathbf{x}}, \mathbf{n}, \mathbf{c}) = \max(\omega_{\mathbf{x}} \cdot \mathbf{n}, 0)\mathbf{c}$  as our BRDF (Lambert cosine law included in BRDF for simplicity, following convention of NeRV [37]), giving the rendering equation:

$$\hat{L}_{\mathbf{x}} = I_{\mathbf{x}}f_r(\omega_{\mathbf{x}}, \mathbf{n}, \mathbf{c}) \quad (4)$$

With captured pixel intensity  $L_{\mathbf{x}}$ , we use L1 loss and formulate the NeLiS optimization problem as:

$$\min_{\theta, A, \tau, R_i^c, t_i^c, \mathbf{c}} \sum_{\mathbf{x} \in \text{ROI}} \|L_{\mathbf{x}} - \hat{L}_{\mathbf{x}}\|_1 \quad (5)$$

### B. Building DarkGS

Within the framework of 3DGS, we model the scene with a point cloud of Gaussians  $\mathbf{G}$  (as shown in Fig. 3 right). Each Gaussian  $g_i$  in the point cloud encompasses attributes including position  $\mathbf{p}_i$ , covariance  $\Sigma_i$ , opacity  $\alpha_i$ , albedo  $\mathbf{c}_i$  and normal  $\mathbf{n}_i$ , that  $g_i = \{\mathbf{p}_i, \Sigma_i, \alpha_i, \mathbf{c}_i, \mathbf{n}_i\} \in \mathbf{G}$ .

Given  $\mathbf{p}_i$ , the incident radiance  $I_i$  can be calculated by Eq. 3. With  $\mathcal{N}$  ordered points for pixel  $(u, v)$ , the rendering equation then becomes:

$$\hat{L}_{u,v} = \sum_{i \in \mathcal{N}} I_i f_r(\omega_i, \mathbf{n}_i, \mathbf{c}_i) \alpha_i \prod_j^{i-1} (1 - \alpha_j) \quad (6)$$

1) *Scale Recovery*: The framework of 3DGS and its variants are often based on monocular SfM solutions such as COLMAP [4]. While calibration using AprilTags recovers poses in true metric scale with NeLiS, monocular SfM only gives up-to-scale poses for building 3D Gaussians. Here, we introduce a scaling factor  $s > 0$  as a learnable parameter so that we can obtain the positions with scale  $\mathbf{p}'_i = s\mathbf{p}_i$ . With captured pixel intensity  $L_{u,v}$ , the 3D Gaussian can be built by solving the follow optimization problem:

$$\min_{A, \mathbf{G}, s} \sum \|L_{u,v} - \hat{L}_{u,v}\|_1 \quad (7)$$

2) *Training Warm-Up*: In some cases, a large discrepancy between the initial scale and the true scale often leads to divergence and local minimum at the beginning of optimization. We therefore propose a warm-up strategy to tackle this problem. The overall idea behind this warm-up strategy is that we overwrite our light pose to be co-centered and parallel with the camera. In this co-centered configuration, the baseline between camera and light will be zero, so that the scale difference between the baseline and the point cloud will have a minor effect on shading at coarse level. Then, we gradually recover the pose to the calibration result over the first  $k$  iterations. During this process, Eq. 7 is being solved and  $s$  will be optimized. We define a warm-up factor that grows with iterations, so that in the  $m^{\text{th}}$  iteration, this

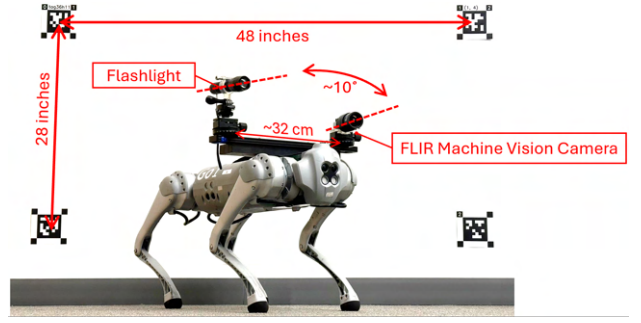


Fig. 4: Our experiment setup: The imaging system is installed on a legged robot platform (Unitree GO1). We use a FLIR machine vision camera to stream the images in RAW format. The calibration target (as shown behind the dog) is a white wall with four AprilTags positioned in a rectangle.

factor is  $\frac{m}{k}$ . Here, we denote Lie exponential map and Lie logarithm map as follows:

$$\exp(\cdot) : \mathbb{R}^3 \rightarrow SO(3), \quad \log(\cdot) : SO(3) \rightarrow \mathbb{R}^3 \quad (8)$$

so that in warm-up stage we replace  $R_i^c$  and  $t_i^c$  with:

$$\hat{R}_i^c = \exp\left(\frac{m}{k} \log(R_i^c)\right), \quad \hat{t}_i^c = \frac{m}{k} t_i^c \quad (9)$$

3) *Relighting*: Once the DarkGS model is built, we can relight the scene by replacing the components in Eq. 3, i.e.  $\Psi_{\tau}$ ,  $\Phi_{\theta}$  and  $A$ , by carefully designed values and functions. For example, replacing the MLP function in  $\Phi_{\theta}$  with a constant can create Lambertian illumination without a strong pattern; replacing the light falloff function  $\Psi_{\tau}$  with a constant can simulate the illumination from a parallel light source.

### C. NeLiS frontend: Human-in-the-loop calibration

Although front-end software is not the main contribution of this paper, it provides an interactive interface for NeLiS visualization, manual initialization of parameters, and hyperparameter tuning, as partially shown in Fig. 2 and elaborated on our [github](#). We find that the NeLiS model is prone to local minimum, especially when learning an MLP and camera-light pose from scratch at the same time. In addition, optimizing MLP and albedo together creates an under-constrained problem, often leading to numerical instability. With our GUI, we first initialize the camera-light pose with a Gaussian distribution as initial RID, and fine-tune the pose, albedo, and other parameters. Then we train an MLP-based RID with pose and albedo frozen, preventing divergence in concurrently learning multiple parameters. Once the model has converged, we unfreeze all the learnable parameters and optimize them together to reach the global minimum.

## IV. EXPERIMENTS

### A. Experiments Setup

Our imaging system consists of a FLIR machine vision camera (Model Firefly S) and a light source, together mounted on a rigid structure on top of our legged robot, as shown in Fig. 4. The baseline between the camera



Fig. 5: None of the existing methods can solve the problem: Results of Vanilla 3DGS [9], RawNeRF [10] Relightable 3DGS [11] show heavy artifacts and fail to converge. The key reason for the failures is that the existing method does not model the illumination change.

and the light source is approximately 32 cm, measured by hand. Although the ray-tracing problem can be simplified by placing the light source and camera close enough and modeling them as co-centered [29], in many applications a long baseline between them is a preferred design worth studying. For example, deep-sea robots usually have a long baseline between light and camera to avoid massive light backscattering from the water.

We experiment with three different light sources: a flashlight, a diving light, and a flood light. To build NeLiS, we take  $\sim 40$  calibration images from different ranges and perspectives for each light configuration and use 25% of the images as a testing set to evaluate the NeLiS performance. For building DarkGS, we take 50  $\sim$  150 images for each scene. All images are streamed in RAW format without any tone mapping or white balancing. Parameters optimized in both NeLiS and DarkGS use Adam optimizer [38] with LieTorch [39] integration.

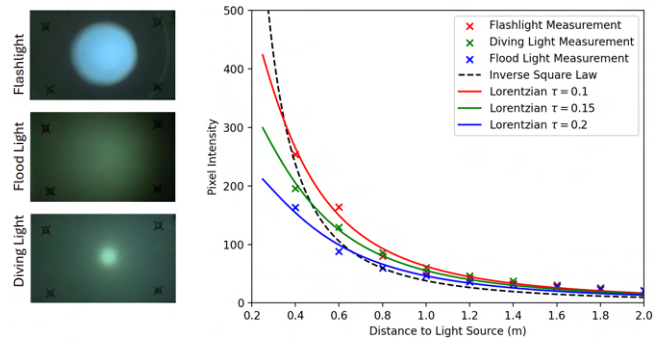
### B. Can existing 3DGS methods do the job?

We first investigate whether existing 3DGS methods can reconstruct the scene using the images we collected with our robotic setup in the dark. We experimented with: 1) vanilla 3DGS [9] which models the scene with constant radiance 2) RawNeRF [10] which is developed to reconstruct scene from RAW HDR images 3) Relightable 3DGS [11] which models the shading and environmental light map. To make a fair comparison, we apply gamma tone mapping to images for vanilla 3DGS [9] and Relightable 3DGS [11] which brightens the image and smooth the discrepancy between illuminated and under-illuminated areas. We also replace RawNeRF’s backbone MLP-based renderer [10] with Gaussian render [9].

As shown in Fig. 5, we find that none of the existing methods is able to build valid scene reconstructions. We see an excessive amount of artifacts in the center area of the synthesized image. The key reason is that the capability to model varying illumination is missing from existing methods.

### C. Why do we need to learn RID?

As shown in Fig. 6a, different light has different RID patterns. Existing methods often model RID as known [25], or general functions such as the power of cosine functions [22] or a Gaussian distribution. However, for most light sources, and all light sources we used in this research, RID is not given. Modeling them with general functions such as a



(a) RID

(b) Light Fall-off Curve

Fig. 6: (a) Light sources on real robots have various RID patterns; (b) Real world measurements of light falloff show that the inverse square law is insufficient to model any of our light sources, but Lorentzian functions [36] with learnable parameter  $\tau$  fit them well.

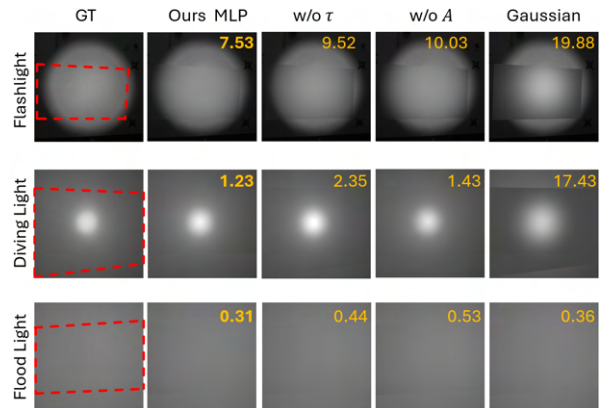


Fig. 7: Ablation study: We show that our model with an MLP-based RID, learnable light fall-off parameter  $\tau$  and ambient light  $A$  can effectively improve the rendering performance. **MSE** of the testing set are highlighted in yellow. The **red dashed box** bounds the regions to be rendered.

Gaussian distribution will not be adaptive and expressive enough to reflect the true RID of different light patterns for building a photorealistic renderer. The ablation study in Fig. 7 (columns 2 and 5) shows that while a Gaussian distribution fits the RID of flood light well, it performs much worse on the other two types of light. In comparison, our MLP-based model fits all 3 kinds of light with the best performance, showing good adaptivity to different light patterns.

### D. Why do we need to learn light falloff?

For simplicity, previous studies [22], [24], [25] use an inverse square law to model the light falloff. It is not a good fit for real-world light systems. We measure the light falloff of the center point of 3 different light sources as shown in Fig. 6b. The plot show that when the range is close, light falloff does not follow the inverse square law as the point light source assumption starts to fail. However, with a Lorentzian model suggested by [36], the light falloff can be better fitted



Fig. 8: Visualization of results from multiple scenes: We show that with DarkGS, we can reconstruct the scene with RAW images from robotic deployments in dark environments, and relight the scene to reveal more information that is corrupted in the RAW image due to uneven and partial illumination. Results as shown are all from the flashlight setup which is the most challenging according to our numerical results.

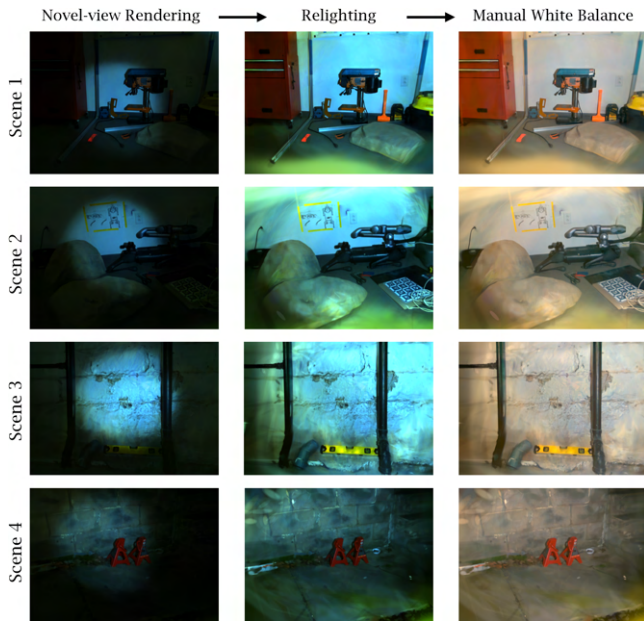


Fig. 9: Novel-view rendering, relighting with virtual Lambertian illumination, and manual white balance.

with our introduced learnable parameter  $\tau$ . We further show numerical results in Fig. 7 (columns 2 and 3) that by learning  $\tau$ , the rendering performance on testing set of all 3 kinds of lights gets improved, which implies that our light fall-off model better approximates the true light falloff.

#### E. Do we need a perfectly dark environment?

No. Ambient light is universal in all kinds of environment and a perfectly dark room for calibration is rare for robot deployments in the wild. Neither our calibration with NeLiS nor deployment experiments with DarkGS are performed in a perfectly dark environment. As shown in Fig. 7 (columns 2 and 4), we show that with our modeling of ambient light  $A$ , the performance of the testing set is improved on all three different lights.

#### F. Results Visualization

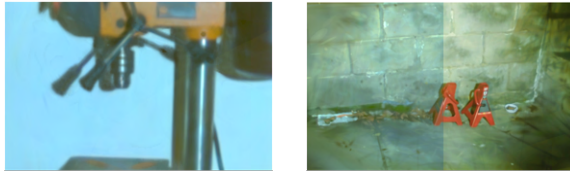
We deploy our system in various real-world environments and the results are shown in Fig. 8. Compared with ground truth, our model is able to reconstruct the image with photo-realistic quality with learned illumination. Then we replace the light source in our model with a virtual Lambertian light to create a global illumination that illuminates the entire field of view (FOV) of the camera with photorealistic rendering quality.

We further visualize the novel-view rendering results in Fig 9. In the first column in Fig 9, we show that the learned light effects generalize to novel views well, with visual quality comparable to training views in Fig. 8. We relight the scene (column 2) in a way similar to Fig. 8, and manual white balance (column 3).

### V. LIMITATIONS

A limitation of our work is that our model currently does not support modeling objects with heavy shadows or specular

reflection (Fig. 10a). However, recent concurrent works [11], [40] have shown the potential to integrate them into the 3DGS framework. It is also obvious that our relighted image is in green-blue tone (Fig. 10b). This is because 1) We are sticking with the RAW image format throughout the pipeline, so images appear green as the RAW image nature 2) Our LED light has a high color temperature (cool color) so that the relighted image appears blue.



(a) Shadow & Specularity (b) Auto White Balancing

Fig. 10: Limitations: (a) Our current method is not able to model shadows and specularity. (b) The relighted image is still in green-blue tone due to the nature of RAW images format and that automatic white balancing is missing from our pipeline.

## VI. CONCLUSIONS AND FUTURE WORK

This work aims to solve the problem of building 3D Gaussians and scene relighting from images taken by a moving camera-light system. Our proposed pipeline consists of NeLiS, a camera-light simulation and calibration model, and DarkGS which build's photolistic representation for scenes in the dark. The results show that our proposed pipeline can build relightable Gaussians from images taken by the robot platform deployed in the dark, while other current approaches are not able to do the job. Ablations show that components in our proposed model effectively improve the performance of our shading model, so we can learn RID with an MLP for arbitrary light pattern, better approximate the light-falloff curve, and allow us to calibrate and deploy the system in environments that are not perfectly dark. These efforts make our model more applicable in real-world robot deployments. Future work includes modeling shadows and non-Lambertian objects, and bringing white balance and tone mapping into the loop.

## ACKNOWLEDGEMENTS

This work is supported by NOAA NA22OAR0110624.

## REFERENCES

- [1] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An open urban driving simulator," in *Proceedings of the 1st Annual Conference on Robot Learning*, 2017, pp. 1–16.
- [2] A. Agha, K. Otsu, B. Morrell, *et al.*, "Nebula: Quest for robotic autonomy in challenging environments; team costar at the darpa subterranean challenge," *arXiv preprint arXiv:2103.11470*, 2021.
- [3] E. Potokar, S. Ashford, M. Kaess, and J. G. Mangelson, "Holocean: An underwater robotics simulator," in *2022 International Conference on Robotics and Automation (ICRA)*, 2022, pp. 3040–3046.
- [4] S. Agarwal, Y. Furukawa, N. Snavely, *et al.*, "Building rome in a day," *Communications of the ACM*, vol. 54, pp. 105–112, 2011.
- [5] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "Nerf: Representing scenes as neural radiance fields for view synthesis," in *ECCV*, 2020.
- [6] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, pp. 91–110, 2004.
- [7] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardós, "ORB-SLAM: A versatile and accurate monocular SLAM system," *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.
- [8] C. Campos, R. Elvira, J. J. Gómez, J. M. M. Montiel, and J. D. Tardós, "ORB-SLAM3: An accurate open-source library for visual, visual-inertial and multi-map SLAM," *IEEE Transactions on Robotics*, vol. 37, no. 6, pp. 1874–1890, 2021.
- [9] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis, "3d gaussian splatting for real-time radiance field rendering," *ACM Transactions on Graphics*, vol. 42, no. 4, Jul. 2023.
- [10] B. Mildenhall, P. Hedman, R. Martin-Brualla, P. P. Srinivasan, and J. T. Barron, "NeRF in the dark: High dynamic range view synthesis from noisy raw images," *CVPR*, 2022.
- [11] J. Gao, C. Gu, Y. Lin, *et al.*, "Relightable 3d gaussian: Real-time point cloud relighting with brdf decomposition and ray tracing," *arXiv:2311.16043*, 2023.
- [12] J. L. Schönberger and J.-M. Frahm, "Structure-from-motion revisited," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
- [13] E. Olson, "Apriltag: A robust and flexible visual fiducial system," in *2011 IEEE International Conference on Robotics and Automation*, 2011, pp. 3400–3407.
- [14] J. Rehder, J. Nikolic, T. Schneider, T. Hinzmann, and R. Siegwart, "Extending kalibr: Calibrating the extrinsics of multiple imus and of individual axes," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2016, pp. 4304–4311.
- [15] J. Beltrán, C. Guindel, A. de la Escalera, and F. García, "Automatic extrinsic calibration method for lidar and camera sensor setups," *IEEE Transactions on Intelligent Transportation Systems*, 2022.
- [16] J. Domhof, J. F. P. Kooij, and D. M. Gavrilu, "A joint extrinsic calibration tool for radar, camera and lidar," *IEEE Transactions on Intelligent Vehicles*, vol. 6, no. 3, pp. 571–582, 2021.
- [17] D. Yang, B. He, M. Zhu, and J. Liu, "An extrinsic calibration method with closed-form solution for underwater opti-acoustic imaging system," *IEEE Transactions on Instrumentation and Measurement*, vol. 69, no. 9, pp. 6828–6842, 2020.
- [18] J. Song, L. Zhao, and K. A. Skinner, "Lirafusion: Deep adaptive lidar-radar fusion for 3d object detection," *arXiv preprint arXiv:2402.11735*, 2024.
- [19] H. Santo, M. Waechter, M. Samejima, Y. Sugano, and Y. Matsushita, "Light structure from pin motion: Simple and accurate point light calibration for physics-based modeling," in *European Conference on Computer Vision (ECCV)*, 2018.
- [20] H. Santo, M. Waechter, W.-Y. Lin, Y. Sugano, and Y. Matsushita, "Light structure from pin motion: Geometric point light source calibration," *International Journal of Computer Vision (IJCV)*, 2020.
- [21] L. Ma, Y. Liu, J. Liu, *et al.*, "A fast led calibration method under near field lighting based on photometric stereo," *Optics and Lasers in Engineering*, vol. 147, p. 106749, 2021.
- [22] L. Ma, X. Liu, Y. Liu, X. Pei, and S. Guo, "Robust point light source calibration method for near-field photometric

- stereo using feature points selection,” *Appl. Opt.*, vol. 62, no. 36, pp. 9512–9522, Dec. 2023.
- [23] W. Jin, M. Zhu, J. Liu, B. He, and J. Yu, “Shadow-based lightsource localization with direct camera–lightsource geometry,” *IEEE Transactions on Instrumentation and Measurement*, vol. 73, pp. 1–12, 2024.
- [24] L. Ma, J. Liu, X. Pei, Y. Hu, and F. Sun, “Calibration of position and orientation for point light source synchronously with single image in photometric stereo,” *Opt. Express*, vol. 27, no. 4, pp. 4024–4033, Feb. 2019.
- [25] Y. Song, F. Elibol, M. She, D. Nakath, and K. Köser, *Light pose calibration for camera-light vision systems*, 2020.
- [26] G. Chen and W. Wang, “A survey on 3d gaussian splatting,” *arXiv preprint arXiv:2401.03890*, 2024.
- [27] J. T. Barron, B. Mildenhall, M. Tancik, P. Hedman, R. Martin-Brualla, and P. P. Srinivasan, “Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields,” *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 5835–5844, 2021.
- [28] T. Müller, A. Evans, C. Schied, and A. Keller, “Instant neural graphics primitives with a multiresolution hash encoding,” *ACM Trans. Graph.*, vol. 41, no. 4, 102:1–102:15, Jul. 2022.
- [29] S. Bi, Z. Xu, P. Srinivasan, *et al.*, “Neural reflectance fields for appearance acquisition,” *arXiv preprint arXiv:2008.03824*, 2020.
- [30] T. Zhang and M. Johnson-Roberson, “Beyond nerf underwater: Learning neural reflectance fields for true color correction of marine imagery,” *IEEE Robotics and Automation Letters*, vol. 8, no. 10, pp. 6467–6474, 2023.
- [31] Y. Jiang, J. Tu, Y. Liu, *et al.*, “Gaussianshader: 3d gaussian splatting with shading functions for reflective surfaces,” *arXiv preprint arXiv:2311.17977*, 2023.
- [32] R. Woodham, “Photometric method for determining surface orientation from multiple images,” *Optical Engineering*, vol. 19, Jan. 1992.
- [33] A. Chakrabarti and K. Sunkavalli, “Single-image rgb photometric stereo with spatially-varying albedo,” in *2016 Fourth International Conference on 3D Vision (3DV)*, IEEE, 2016, pp. 258–266.
- [34] R. Basri, D. Jacobs, and I. Kemelmacher, “Photometric stereo with general, unknown lighting,” *International Journal of Computer Vision*, vol. 72, pp. 239–257, May 2007.
- [35] M. A. Fischler and R. C. Bolles, “Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography,” *Commun. ACM*, vol. 24, no. 6, 381–395, Jun. 1981.
- [36] A. Ryer, *Light Measurement Handbook*. International Light, 1997.
- [37] P. P. Srinivasan, B. Deng, X. Zhang, M. Tancik, B. Mildenhall, and J. T. Barron, “Nerv: Neural reflectance and visibility fields for relighting and view synthesis,” in *CVPR*, 2021.
- [38] D. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *International Conference on Learning Representations*, Dec. 2014.
- [39] Z. Teed and J. Deng, “Tangent space backpropagation for 3d transformation groups,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.
- [40] L. Bolanos, S.-Y. Su, and H. Rhodin, “Gaussian shadow casting for neural characters,” 2024.