SunFaded: Illumination-Aware Gaussian Splatting for Dark Scenes with Camera-Mounted Active Lighting

Supplementary Material

7. Network Details

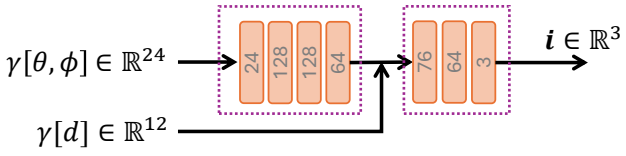

Figure 7. Network Structure

The architecture of our lighting MLP is illustrated in Fig. 7. We first encode the light direction (θ, ϕ) with a positional encoding $\gamma[\theta, \phi] \in \mathbb{R}^{24}$ and feed it into a four-layer MLP to obtain a 64-dimensional feature. This feature is then concatenated with the encoded distance $\gamma[d] \in \mathbb{R}^{12}$, forming a 76-dimensional vector that is processed by a second three-layer MLP. The final output $i \in \mathbb{R}^3$ represents the predicted RGB lighting for each tile.

8. Implementation Details

We detail the optimization scheme described in Sec. 5.2; the full procedure is summarized in Algorithm 1. We adopt a three-stage optimization schedule to jointly learn the 2D Gaussian representation $\{G_i\}$, the global ambient term E, and the lighting model M_{light}. In the first stage (iterations 1–10000), we initialize the 2D Gaussians and a global ambient illumination E, and optimize them using an unlit reconstruction loss $\mathcal{L}_{\text{unlit}}$ on the unlit renderings I_m^u, weighted by the illumination map W_m, together with normal and depth regularization terms $\mathcal{L}_{\text{normal}}$ and $\mathcal{L}_{\text{depth}}$ obtained from the depth prior network f_{prior}. This stage drives the representation toward an illumination-invariant geometry and albedo. In the second stage (iterations 10000–30000), we introduce the lighting MLP M_{light} and the light pose $\mathbf{l}$. For the first half of this stage (10000–20000 iterations), we keep the Gaussians and E fixed and update only M_{light} and $\mathbf{l}$ under the photometric loss $\mathcal{L}_{\text{r}}(I_m^r, I_m)$, so as to accurately calibrate the lighting given the current geometry. In the second half (20000–30000 iterations), we jointly optimize $\{G_i\}$, E, M_{light}, and $\mathbf{l}$ under the same rendering loss, allowing geometry, appearance, and lighting to be refined in a coherent manner. In the third stage (iterations 30000–40000), we freeze E and M_{light} and refine the geometry with an albedo-based depth prior: we render albedo maps A_m, feed them into f_{prior} to obtain D_m^{albedo} and N_m^{albedo}, and optimize a combination of depth, normal, and photometric losses using the same rendering pipeline as in Stage 2. During iterations 30000–35000, we update only the Gaussian opacities $\{\alpha_i\}$ to clean up floaters and adjust visibility, and in the last 5000 iterations we further update all Gaussian attributes while keeping the lighting branch fixed, which improves the constructed 2DGS representation $\{G_i\}$.

9. Evaluations on the Opacity Schedule

In Sec. 4.3, we introduce a strategy that predicts depth from rendered albedo images and then uses it to further train the Gaussian set G_i. During this stage, we freeze all Gaussian attributes and only optimize the opacity. In this section, we perform an ablation study on this opacity-only optimization scheme. As shown in Fig. 10 and Tab. 6, enabling opacity-only optimization leads to sharper texture details and fewer artifacts, confirming the effectiveness of the proposed strategy.

Table 6. **Ablation study on the proposed opacity-based optimization strategy.**

	PSNR	SSIM	LPIPS
w/o opacity-only opt.	39.90	0.9717	0.0313
all	**41.02**	**0.9739**	**0.0286**

10. Results after Brightness Enhancement

Since our method focuses on reconstruction in dark scenes, we apply a consistent brightness enhancement to selected images to facilitate comparison in dark regions. As shown in Fig. 8, our approach not only produces a more faithful appearance in illuminated areas, but also preserves finer details in the dark regions.

11. Additional Visual Comparisons for Appearance Modeling

Since DarkGS and WildGaussians also perform appearance modeling to remove illumination, their outputs correspond to scenes without explicit lighting effects. For a fair comparison, we manually apply the brightness adjustment to their results, as shown in Fig. 9. To provide clearer references, Fig. 9e and 9d show the original ground-truth image and its brightness-enhanced version, respectively.

Algorithm 1: Three-Stage Optimization

Input: Multi-view RGB images $\{I_m\}_{m=1}^{M}$ with camera poses; depth prior network f_{prior}
Output: Optimized 2D Gaussian set $\{G_i\}$, ambient term E, lighting MLP M_{light}, and light pose $\mathbf{l}$

Stage 1: Unlit Scene Optimization
Initialize 2D Gaussians $\{G_i\}$ and global ambient illumination E.
for $t = 1$ **to** 10000 **do**
  $\{G_i\}, E \xrightarrow{Splatting} I_m^u, D_m, N_m$.
  $I_m \xrightarrow{\mathcal{K}_\sigma} W_m$.
  $I_m \xrightarrow{f_{\text{prior}}} D_m^{\text{p}}, N_m^{\text{p}}$.
  Update E and the parameters of $\{G_i\}$ by gradient descent on

$$\mathcal{L} = \mathcal{L}_{\text{unlit}}(I_m, W_m, I_m^u) + 0.01\mathcal{L}_{\text{normal}}(N_m, N_m^{\text{p}}) + 0.001\mathcal{L}_{\text{depth}}(D_m, D_m^{\text{p}}).$$

Stage 2: Optimization for Light Modeling
Initialize lighting MLP M_{light} and light pose $\mathbf{l}$.
for $t = 10000$ **to** 30000 **do**
  $\{G_i\} \xrightarrow{Splatting} A_m, D_m, N_m$.
  $(D_m, \mathbf{l}) \xrightarrow{M_{\text{light}}} \mathbf{i}_m$.
  $(A_m, N_m, \mathbf{i}_m) \xrightarrow{\text{Shading}} I_m^{\text{diff}}$.
  $I_{\text{diff},m}, E, A_m \xrightarrow{\text{Combine}} I_m^r$.
  Compute

$$\mathcal{L} = \mathcal{L}_{\text{r}}(I_m^r, I_m).$$

  if $t < 20000$ **then**
    Update M_{light} and $\mathbf{l}$ by gradient descent on $\mathcal{L}$.
  else
    Update $\{G_i\}$, E, M_{light} and $\mathbf{l}$ by gradient descent on $\mathcal{L}$.

Stage 3: Geometry Refinement with Albedo-based Depth Prior
Freeze E and M_{light} for this stage.
for $t = 30000$ **to** 40000 **do**
  $\{G_i\} \xrightarrow{Splatting} A_m, D_m, N_m$.
  $\{G_i\}, A_m, D_m, N_m, \mathbf{l}, E \xrightarrow{M_{\text{light}}} I_m^r$ `// Same rendering pipeline as in lines 12--14`
  $A_m \xrightarrow{f_{\text{prior}}} D_m^{\text{albedo}}, N_m^{\text{albedo}}$.
  Compute

$$\mathcal{L} = 0.01\mathcal{L}_{\text{normal}}(N_m, N_m^{\text{albedo}}) + 0.001\mathcal{L}_{\text{depth}}(D_m, D_m^{\text{albedo}}) + \mathcal{L}_{\text{r}}(I_m^r, I_m).$$

  if $t \leq 35000$ **then**
    Update only the opacities $\{\sigma_i\}$ by gradient descent on $\mathcal{L}$.
  else
    Update all Gaussian attributes in $\{G_i\}$ by gradient descent on $\mathcal{L}$.
return $\{G_i\}$, E, M_{light}, $\mathbf{l}$

(a) VAST (b) DarkGS (c) GS-W (d) Ours (e) GT (brightness-adjusted)

Figure 8. **Brightness-adjusted qualitative comparisons on the DarkRobotic dataset.**

(a) DarkGS (b) WildGaussians (c) Ours (d) GT (brightness-adjusted) (e) GT

Figure 9. **Qualitative comparisons for appearance modeling on the DarkRobotic dataset.**

12. Depth Prior Evaluation

In the main paper, Tab. 1 reports the quantitative results with and without the depth prior introduced in Sec. 4.3. Here, we complement those results with qualitative visualizations. As shown in Fig. 13, even without the depth prior our method can still recover a broadly correct overall geometry, but the fine geometric details are noticeably improved when the depth prior is employed.

13. Dataset Construction

In this section, we detail the process of constructing our dataset. We capture images using an iPhone 15 with the built-in flash enabled, and use Adobe Lightroom to record RAW images. During acquisition, we fix the ISO, exposure time, and white balance. After capture, we process the RAW images using `rawpy` to convert them into PNG format, and then feed the resulting images into COLMAP for reconstruction and camera pose estimation.

14. Additional Visualization Results

In this section, we present additional qualitative results on the DarkPhone dataset. As shown in Fig. 14, we apply a simple brightness enhancement to all rendered images to facilitate visual comparison. Our method produces visually competitive results compared to the baselines.

15. View-Consistency Videos

To further demonstrate view consistency, we provide dense novel-view renderings for Ours, DarkGS, and GS-W in the supplementary PDF. Please view the animations using Adobe Acrobat.

16. Discussion of Limitations

As illustrated in Fig. 12, we present several examples to discuss the main limitations of our method. First, our approach struggles to accurately model strong specular highlights (green boxes), since we currently adopt a simple reflection model; incorporating more expressive illumination and reflection models may alleviate this issue. Second, our method cannot explicitly handle cast shadows (red boxes). We attempted to treat shadows as artifacts and suppress them using an uncertainty-based masking strategy, but this did not lead to visible improvements, likely because the optimization objective becomes overly entangled with too many competing factors. Finally, our approach is unable to separately model fixed ambient light sources in the scene (e.g., ceiling lamps and diffuse environmental lighting), which manifests as over-bright regions near the light sources, such as the top-right area in the figure.

Figure 12. **Typical failure cases of our method.**

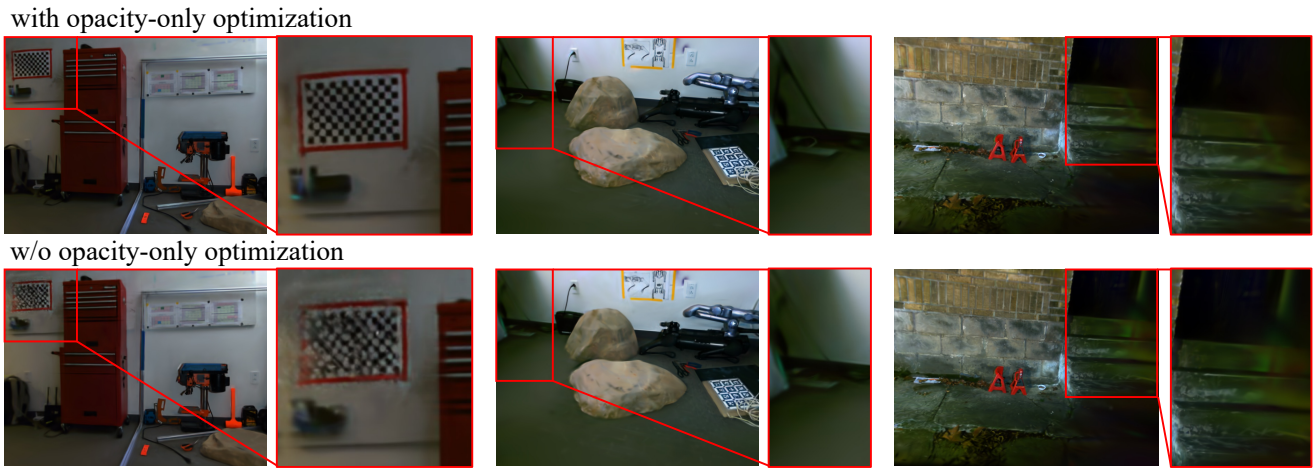

Figure 10. **Ablation study on the proposed opacity-based optimization strategy.**

Figure 11. **Dense novel-view renderings for view-consistency comparison.** From left to right: Ours, DarkGS, and GS-W.

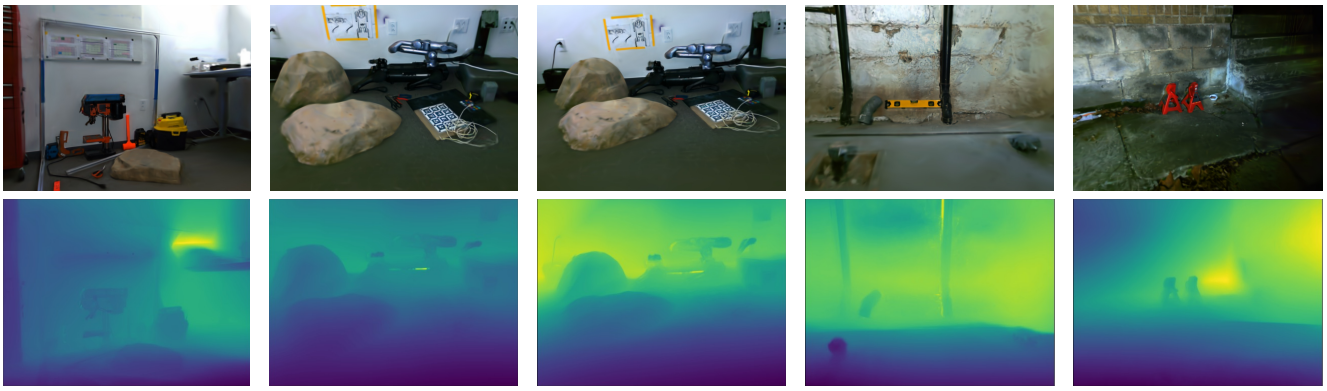

Figure 13. **Depth visualization from our method without the depth prior.**

(a) VAST (b) 2DGS (c) GS-W (d) Ours (e) GT

Figure 14. **Qualitative comparisons on the DarkPhone dataset.**