Gated Fields: Learning Scene Reconstruction from Gated Videos (Supplementary Information)

Andrea Ramazzina[1*] Stefanie Walz[1,2*] Pragyan Dahal[3] Mario Bijelic[4,5] Felix Heide[4,5]

[1]Mercedes-Benz [2]Saarland University [3]Politecnico di Milano [4]Torc Robotics [5]Princeton University

In this supplementary document, we present additional information and experiments in support of the main manuscript. In Section 1, we provide additional details on the proposal network f_p. In Section 2, we provide training details, with focus on the hyperparameters used during optimization, followed by an explanation in Section 3 of the architecture of the networks forming the neural field f_G. In Section 4, we present further qualitative results for the 3D reconstruction task. Section 5 provides details on the creation of the pointcloud which we use as ground truth. Section 6 and Section 7, respectively provide further information about the shadow estimation process as well as the modelling of the illuminator source. Additional results are presented in Sections 8, 9 and 10. Specifically, in Section 8 we show example render results obtained by varying gating proprieties. In Section 9, we show additional results of scene decomposition generated by our method. In Section 10, we compare the 2D depths synthesized by Gated Fields to other state-of-the-art depth estimation and scene reconstruction methods. Finally, in Section 11, additional details of the dataset are listed.

Contents

[*]These authors contributed equally to this work.

1. Proposal Network f_p

In this section, we describe details of the proposal network we employ for sampling in our method. We follow [1] and use a proposal network sampler, comprised of two networks, to identify the portions of the ray most likely to contribute to the final pixel color, as well as to reduce the number of samples that have to be queried by f_{Gd}.
For each camera ray $\mathbf{r}$, we sample an initial set of points using a piece-wise sampler as in [5]. These points are then fed into the first proposal network, and the weights $\hat{\mathbf{w}}_1$ are computed. Following the hierarchical sampling procedure of [2], these weights are then used to sample a new set of points which are then fed to the second proposal network, to compute $\hat{\mathbf{w}}_2$ which are used to finally sample the ray points to be evaluated by f_G.
We use two fused MLPs with hash encoding [3] to respectively produce 120 and 48 samples. We start by updating the proposal network every training step iteration, and progressively lowering the frequency in the first 5000 steps to 1 update every 5 training iterations. Both MLPs are trained to produce a weights distribution $\hat{\mathbf{w}}$ consistent with the weights $\mathbf{w}$ computed by f_{Gd}

$$\mathcal{L}_{prop}(\mathbf{t}, \mathbf{w}, \hat{\mathbf{t}}, \hat{\mathbf{w}}) = \sum_i \frac{1}{w_i} \max(0, w_i - \text{bound}(\hat{\mathbf{t}}, \hat{\mathbf{w}}, T_i))^2, \tag{1}$$

where $\mathbf{t}$ is the ordered set of samples distances $t_{i=1,..,N}$ of the samples along the ray, T_i is the interval $T_i = [t_i, t_{i+1})$, and the function bound$(\cdot)$ computes the total sum of weights $\hat{w}$ overlapping within an interval T

$$\text{bound}(\hat{\mathbf{t}}, \hat{\mathbf{w}}, T) = \sum_{j: T \cap \hat{T} \neq \emptyset} \hat{w}_j. \tag{2}$$

2. Training Details

Next, we provide additional training details. We learn the networks f_p and f_G by training for 35,000 steps while simultaneously optimizing the camera poses $\mathbf{o}_c$ and different hardware-related parameters, namely the camera-to-laser transformation matrices $\mathbf{T}$ and $\mathbf{R}$, the illuminator profile properties η, $\boldsymbol{\Xi}$, $\boldsymbol{\Omega}$, $\boldsymbol{\Theta}$ as well as the gating parameters including number of accumulated laser pulses m_k before read-out, laser pulse duration $t_{l,k}$, camera exposure $t_{g,k}$, and delay ξ_k between laser pulse emission and gated exposure for all three slices $k \in \{0, 1, 2\}$, the general distance offset d_0 for the range intensity profiles to compensate for internal signal processing delays.
The total training loss is

$$\mathcal{L} = \lambda_{1A}\mathcal{L}_{cA} + \lambda_{1P}\mathcal{L}_{cP} + \lambda_2\mathcal{L}_d + \lambda_3\mathcal{L}_s + \lambda_4\mathcal{L}_{nc} + \lambda_5\mathcal{L}_\alpha, \tag{3}$$

where $\lambda_{1A} = 10^0$, $\lambda_{1P} = 10^{-1}$, $\lambda_2 = 10^{-2}$, $\lambda_3 = 10^{-4}$, $\lambda_4 = 10^{-4}$, $\lambda_5 = 10^{-4}$. We treat the first 1000 steps as "warm-up" phase, and only optimize the density field f_{Gd} through the loss $\mathcal{L}_d$. We find that this bootstrapping schema helps correct disjoint learning of geometry and appearance, avoiding getting stuck in early local minima. For the reflectance regularization loss $\mathcal{L}_\alpha$, we sample $\epsilon_{\mathbf{x}}$ uniformly from the interval $(-10^{-3}, 10^{-3})^3$, and $\epsilon_{\mathbf{d}}$ initially from $(-10^{-1}, 10^{-1})^3$, decreasing to 0 in 20'000 iterations. For the shadow loss $\mathcal{L}_s$, we use a value of $\epsilon_i = 0.1$ to filter out all the dark pixels.

3. Network Details

Next, we provide additional details on the network architecture we base our model on. Following [3], we map a position $\mathbf{x}$ in space to its hash-encoded position which is used as input for the density field, which is a shallow fused MLP whose architecture is presented in Tab. 1. The hash encoding has 16 levels, 2 features per level, and a maximum resolution of 2048.

		Density field f_{Gd}		
Layer #	**Layer**	**Activation**	**Input Shape**	**Output Shape**
0	Linear	ReLU	$hash_{dim}$	128
1	Linear	ReLU	128	128
2	Linear	ReLU	128	128
3	Linear	ReLU	128	128
$4a$	Linear	exp	128	1
$4b$	Linear	None	128	chi_{dim}

Table 1. Architecture details of the MLP f_{Gd}.

Here, $chi_{dim} = 20$ is the size of the spatial embedding χ. The scene reflection field $f_{G\alpha}$ takes as input such embedding χ, concatenated with the camera viewing direction $\mathbf{d}$ and laser incident direction ω, projected using spherical harmonics

Scene Reflection field $f_{G\alpha}$

Layer #	Layer	Activation	Input Shape	Output Shape
0	Linear	ReLU	$chi_{dim} + 2 \cdot shd_{dim}$	128
1	Linear	ReLU	128	128
2	Linear	ReLU	128	128
3	Linear	ReLU	128	128
4	Linear	Sigmoid	128	1

Table 2. Architecture details of the MLP $f_{G\alpha}$.

Ambient Light Field f_{Gp}

Layer #	Layer	Activation	Input Shape	Output Shape
0	Linear	ReLU	$chi_{dim} + shd_{dim}$	128
1	Linear	ReLU	128	128
2	Linear	ReLU	128	128
3	Linear	ReLU	128	128
4	Linear	Sigmoid	128	1

Table 3. Architecture details of the MLP f_{Gp}.

as basis. Its network architecture is described in Tab. 2. Similarly, the scene reflection field f_{Gp} takes as input the spatial embedding χ concatenated with the encoded camera viewing direction. Its network architecture is described in Tab. 3. Finally, the normal field f_{Gn} takes as input the spatial embedding χ concatenated with the hashed $\mathbf{x}$. Its network architecture is reported in Tab. 4.

Density field f_{Gn}

Layer #	Layer	Activation	Input Shape	Output Shape
0	Linear	ReLU	$chi_{dim} + hash_{dim}$	128
1	Linear	ReLU	128	128
2	Linear	ReLU	128	128
3	Linear	ReLU	128	128
4	Linear	L2 Norm	128	3

Table 4. Architecture details of the MLP f_{Gn}.

4. Additional 3D Reconstruction Results

In this section, we provide additional qualitative results for the 3D reconstruction from our method. Six examples of the Gated Fields are shown in Fig. 1. For each scene, we present a 3D visualization with color-coded normals (bottom row), as well as the synthesized neural depth for a sample view, next to the LiDAR scan for comparison (middle row). We also include the corresponding RGB and Gated captures for reference (top row).

Gated Fields is able to reconstruct accurately the scene surfaces of objects at both close and further distances. Inspecting the trees or car regions, our method can reconstruct finer structures and surfaces details, for both day and night time scenes. Moreover, the surface and normal areas only visible from a limited range of views (such as the car roofs) are being plausibly synthesized.

5. Groundtruth Pointcloud Generation

In order to evaluate our proposed method, we construct a large-scale ground-truth point cloud by aggregating multiple LiDAR scans and removing noisy points. In particular, we use LIO-SAM [4] tailored to our sensor setup. The algorithm takes as input the measurements from the IMU and GNSS sensor in addition to the LiDAR point cloud. The mapping module consists of three components, namely LiDAR point cloud ego-motion correction, factor graph-based IMU motion prediction, and global map optimization as illustrated in Fig. 2.

Point Cloud Ego-motion Correction mitigates the distortions caused by the ego-motion in the LiDAR points. This step leverages the raw IMU data projected into the LiDAR frame to estimate the rotational transformation at the timestamp of the LiDAR scan. Furthermore, the lidar odometry information is incorporated to compute the initial and final poses during the scan. The IMU measurements and the liDAR odometry are subsequently utilized to de-skew each point in the point cloud, hence compensating for ego-motion.

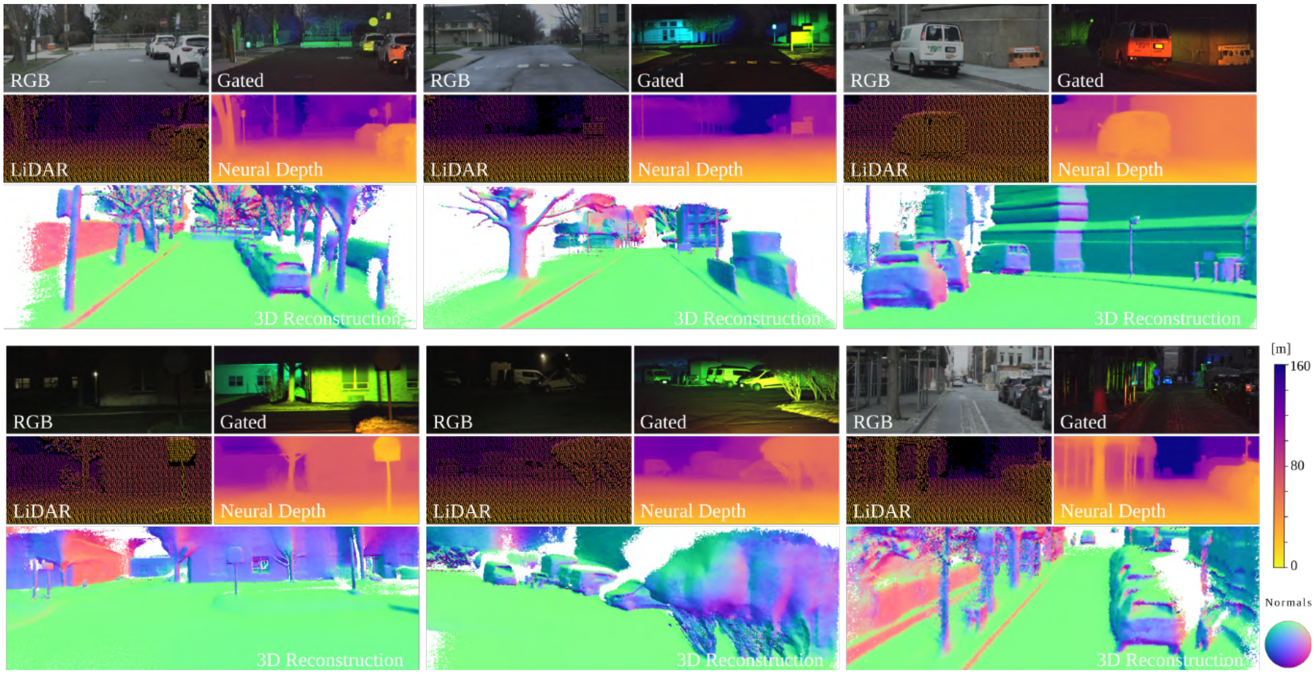


Figure 1. From a single video of gated captures, we reconstruct an accurate scene representation and render depth projections as accurate as LiDAR scans, and we recover 3D geometry and normals at day and night.

IMU Preintegration or IMU motion prediction provides LiDAR odometry at the IMU frequency. It performs a fusion of LiDAR odometry computed by the mapping module with the IMU preintegration leveraging factor graph-based optimization. It needs to be noted that the LiDAR odometry is available at $10Hz$ while the IMU raw measurements are obtained at $400Hz$. This step serves a dual purpose: first providing the mapping module with an accurate initial guess for the frame-to-local map transformation computation and second, providing the point cloud deskewing step with the required LiDAR poses.

Global Map Optimization takes as input the ego-motion corrected point cloud and the initial guess of the LiDAR pose in the map computed through the IMU motion prediction. Additionally, a Global Navigation Satellite System (GNSS) sensor can be added to reduce the possible drifts in the long-term motion. The point cloud registration is performed by aligning the deskewed point cloud with a local map using a combined approach. The algorithm employs plane fitting to refine the surface features and an iterative optimization process to compute the 6DOF transformation between the LiDAR point cloud and the local map. Each LiDAR pose is then used to construct a lidar odometry factor for a factor graph, which in turn is optimized to get poses used to obtain accumulated LiDAR point cloud. The output of the algorithm is the accumulated LiDAR point cloud and keyframe poses for each point cloud are used to generate the accumulated cloud.

The algorithm has been specifically tailored to enhance mapping accuracy. This customization involved the deliberate selection of parameters that prioritize precision over real-time capability. We obtain the accumulated LiDAR point cloud, together with the refined positions and orientations of the LiDAR, Gated, and RGB camera sensors for their corresponding measurement timestamps. Fig. 3 showcases a representation of the environment through the accumulated LiDAR point cloud. Each figure provides a dual perspective, featuring both the complete view and a zoomed-in section corresponding to the RGB and gated images for night and daytime scenes.

6. Shadow Estimation

Next, we provide additional details on the shadow indicator estimation. The shadow indicator ψ of a point $\mathbf{x}$ sampled on the camera ray is obtained by integrating the volumetric density along the illuminator ray $\mathbf{r}_{ill}$. If such computation is performed for each sample along the camera ray, the computational demand scales quadratically with the number of points sampled along the camera ray N.

Thanks to the proposal network f_p, we need to evaluate only a limited number of samples per ray, hence limiting the additional computational burden of the shadow indicator estimation. To speed this step up, we find that, in practice, we get satisfactory results by only evaluating ψ at the point $\mathbf{x}_{distance}$ computed from the expected ray termination distance. This change reduces

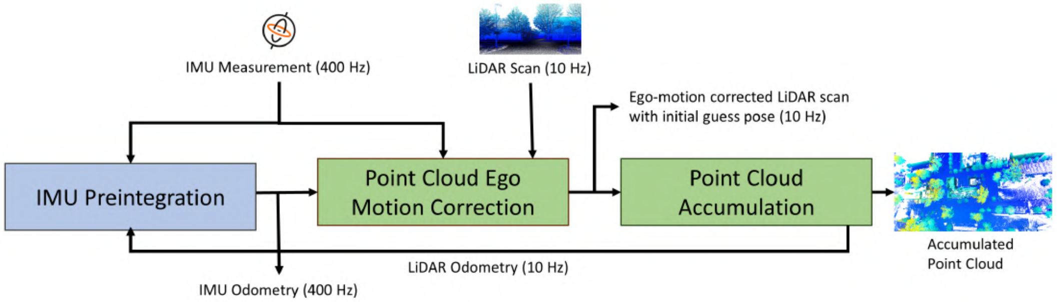


Figure 2. Illustration of the steps of the mapping algorithm used to generate the accumulated LiDAR point cloud.

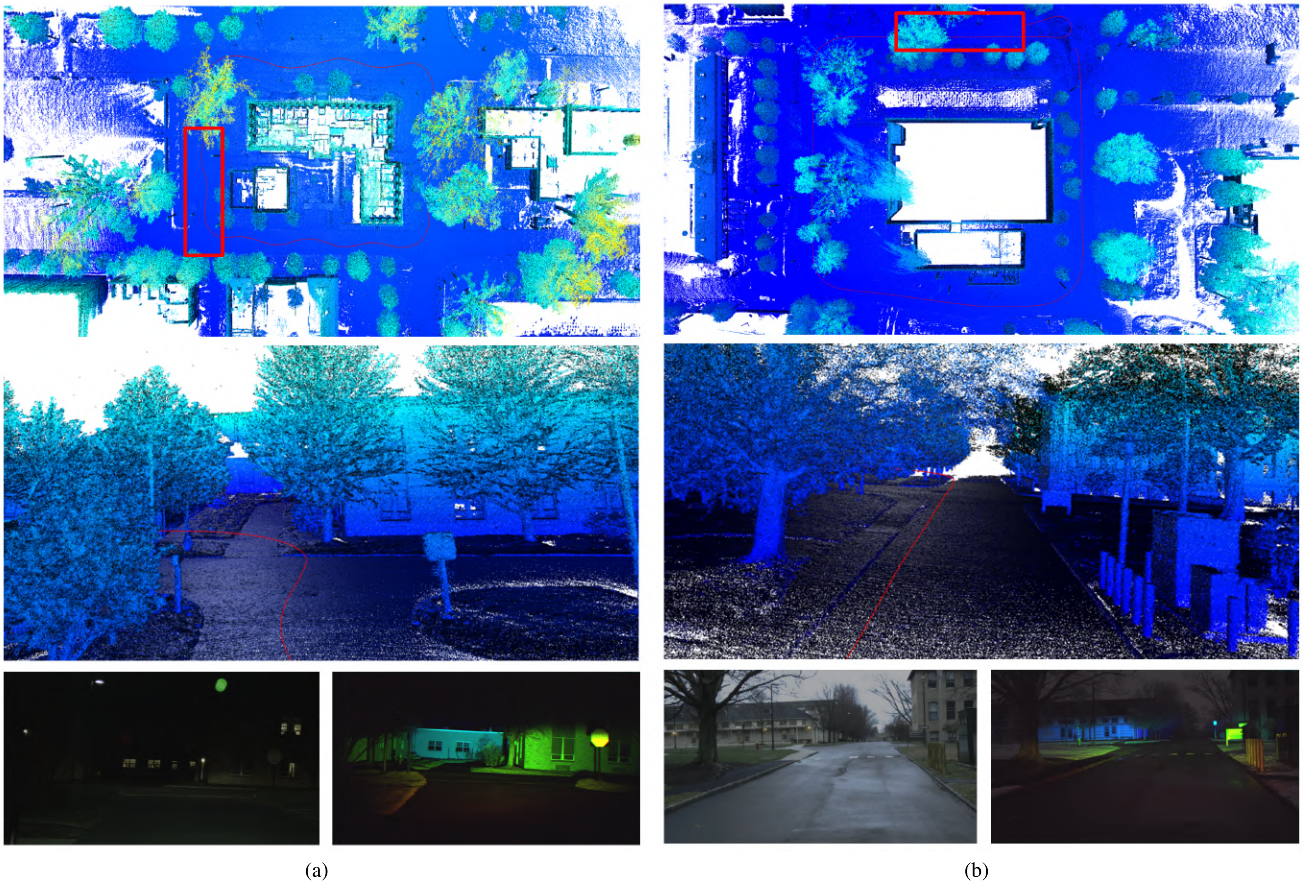


Figure 3. A comprehensive scene representation featuring the accumulated LiDAR point cloud and the vehicle poses illustrated by red trajectory (top). The midsection provides a zoomed-in view aligned with the timestamps of the RGB image in the bottom left corner and the gated image in the bottom right corner.

the computational complexity from quadratic to linear.

7. Illumination Source Modelling

The illuminator consists of two vertical-cavity surface-emitting laser (VCSEL) modules, which illuminate the scene with a laser pulse. We observe that the intensity is maximum at the cross-section center, and diminishes exponentially when diverging from it. We model this distribution as a 2D higher-order Gaussian, mapping the vertical and horizontal displacement angle $\gamma = (\gamma_v, \gamma_h)$ to an attenuation coefficient ι. This model allows for the quasi-constant illuminator intensity at the center-area of the beam, while allowing for exponential decay as we depart from it. To corroborate our model, we show in Fig. 4 a comparison between the cross-section illuminator intensity ι as a function of γ_v, γ_h and the corresponding measured values recorded in a control environment setting. It is possible to note that our model adheres well to the flat-top shape of the vertical profile, as well as the Gaussian for the horizontal section, with a negligible error in most of the domain.

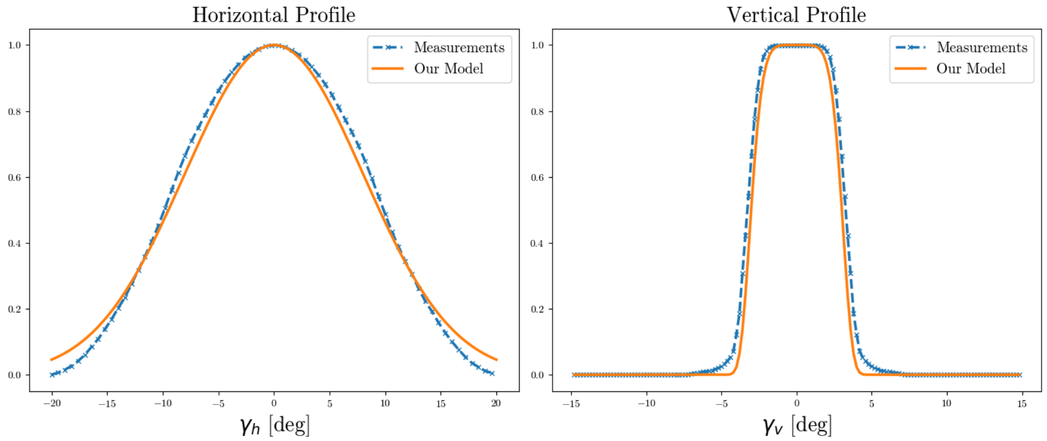


Figure 4. Cross-section comparisons of the illuminator profiles with our model and with controlled-environment measurements. Our modelling closely matches the real measured values, without the need for a complex formulation.

8. Novel Gating Generation

We can not only render a capture from a novel view, but also change any other gated camera parameter. For example, it is possible to re-render the capture varying the delay ξ between laser pulse emission and gated exposure. We present examples in Fig. 5a, where in each row we re-render the same scene with increasing delay ξ for all three slices. In the first column, the close objects are visible in the first and second slices (hence visualised in yellow color). In the second column, we can see that, increasing ξ, the same objects are now only visible in the first slice (hence visualized in red color). Further increasing the delay, such objects are not visible in any of the three slices (hence visualized in black).

Another parameter we can vary is the number of laser pulses m_k. We show in Fig. 5c re-rendering examples of two scenes where we increase m_k for all the three slices. On the left (first two columns), are shown renderings with low number of pulses. As expected, the intensity across the image is low. By increasing the number of pulses, we see the scene intensity increasing, especially for retroreflective materials like the traffic signs.

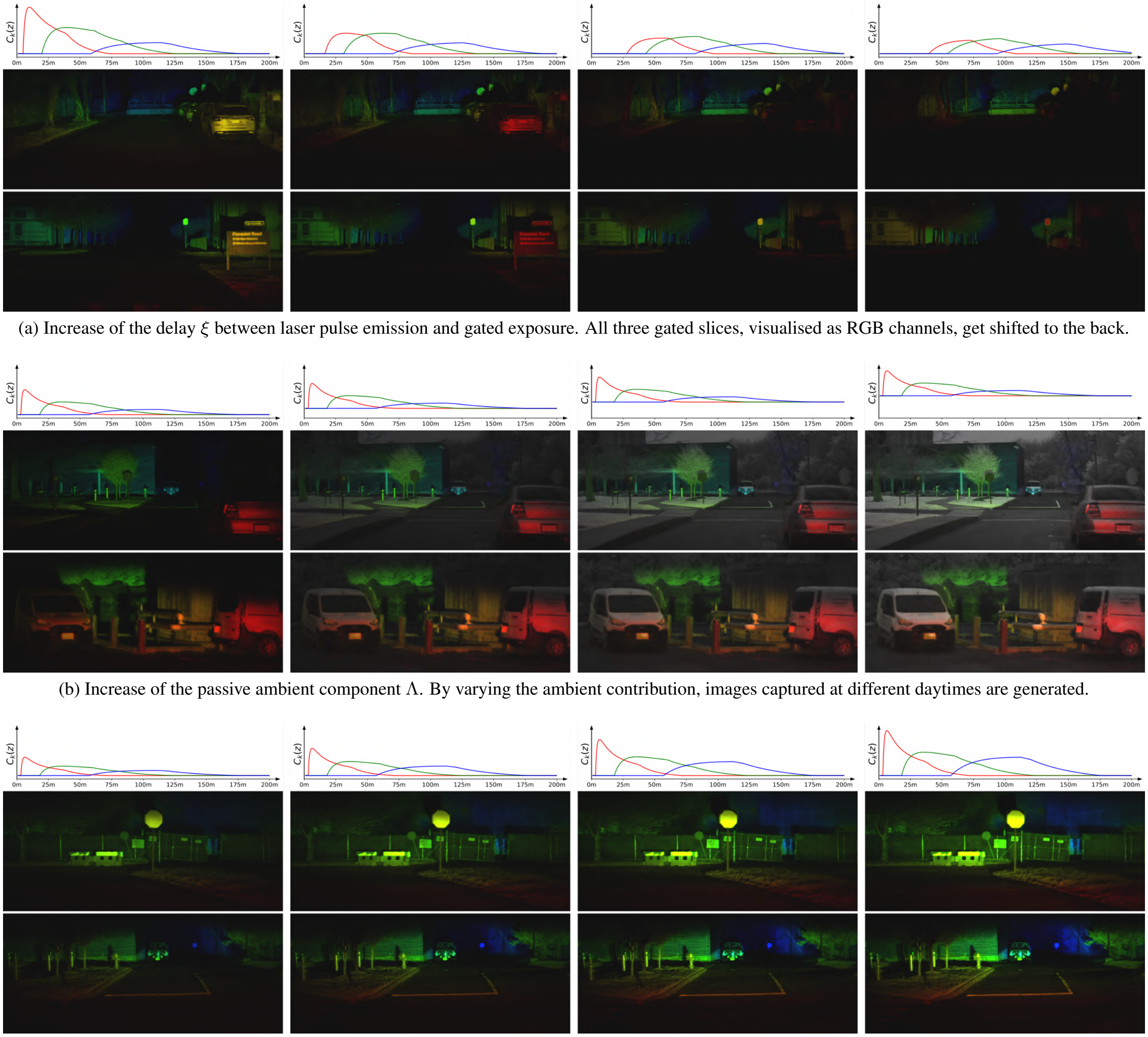
Moreover, our method also enables to vary the ambient light component in the scene. Examples of such modification are being shown in Fig. 5b. On the right, we can see how increasing the ambient light component, the gated effects are less visible. In the first column (on the left) we see the same scene without ambient component. This operation allows us to capture a scene during daytime and re-render it as if it was collected at night (assuming no other visible active lighting).

9. Scene Decomposition by Inverse Rendering

We explicitly leverage the gated imaging formation model in our method to reconstruct the input gated captures. This enables us to disjointly learn normal, ambient lighting, and reflectance of the scene. We show results in Fig. 6, where we visualize the gated capture (ground truth) and the rendered counterpart (reconstruction), followed by the ambient light, normal, laser illumination, effective reflectance (given by the multiplication of reflectance, cosine term, and illumination), shadow value and expected depth.

As visible in the third column, Gated Fields is able to accurately reconstruct the scene geometry and proprieties both when the ambient light is prevalent (i.e., in daylight conditions) as well as in its absence (i.e., at night). Furthermore, the reconstructed normals are coherent with the representation and overall smooth on flat surfaces like road or buildings. The following column represents the intensity of the illuminator light. This intensity depends on the relative 3d position of each point with respect to the illuminator source. More details on the light source model are provided in Sec. 7.

Next, we show the reflectance, multiplied with the cosine term (i.e., $|\omega \cdot \mathbf{n}|$) and the illumination, to better show the active contribution components. Here, it is visible how certain materials like traffic signs or car bodies have a high reflectance, while areas like the road and side walks have a low reflectance. We also show the shadow values, in the second-to-last column. It



(a) Increase of the delay ξ between laser pulse emission and gated exposure. All three gated slices, visualised as RGB channels, get shifted to the back.

(b) Increase of the passive ambient component Λ. By varying the ambient contribution, images captured at different daytimes are generated.

(c) Increase of the number of laser pulses m_k for all three slices. Since the laser power increases, illuminated objects become brighter.

Figure 5. Rendering with novel gating parameters. By increasing or decreasing the different gating parameters, physically correct novel rendered images are generated.

is possible to note here the shadows in black, cast by the different objects in the scene occluding the illuminator light beam. Finally, we also show the rendered depth, which we discuss in detail in Sec. 10.

10. Additional Qualitative Depth Results

In this section, we provide additional results for depth estimation. Qualitative results are shown in Figs. 7 to 12. As visible in the zoom-ins in Fig. 7, Gated Fields is able to reconstruct finer details like wires and trees branches also at far distances, while depth estimation algorithms and other neural reconstruction methods either remove such details or only reconstruct a coarser geometry of them. This improvement over baseline methods is visible in areas more exposed to the illuminator light pulse, as shown in Figs. 8 and 9. Here, poles and traffic signs are being reconstructed with more accurate edges. Here, it is also visible how LiDAR-based method struggles to reconstruct such details, especially if only seen at far distances, since only a

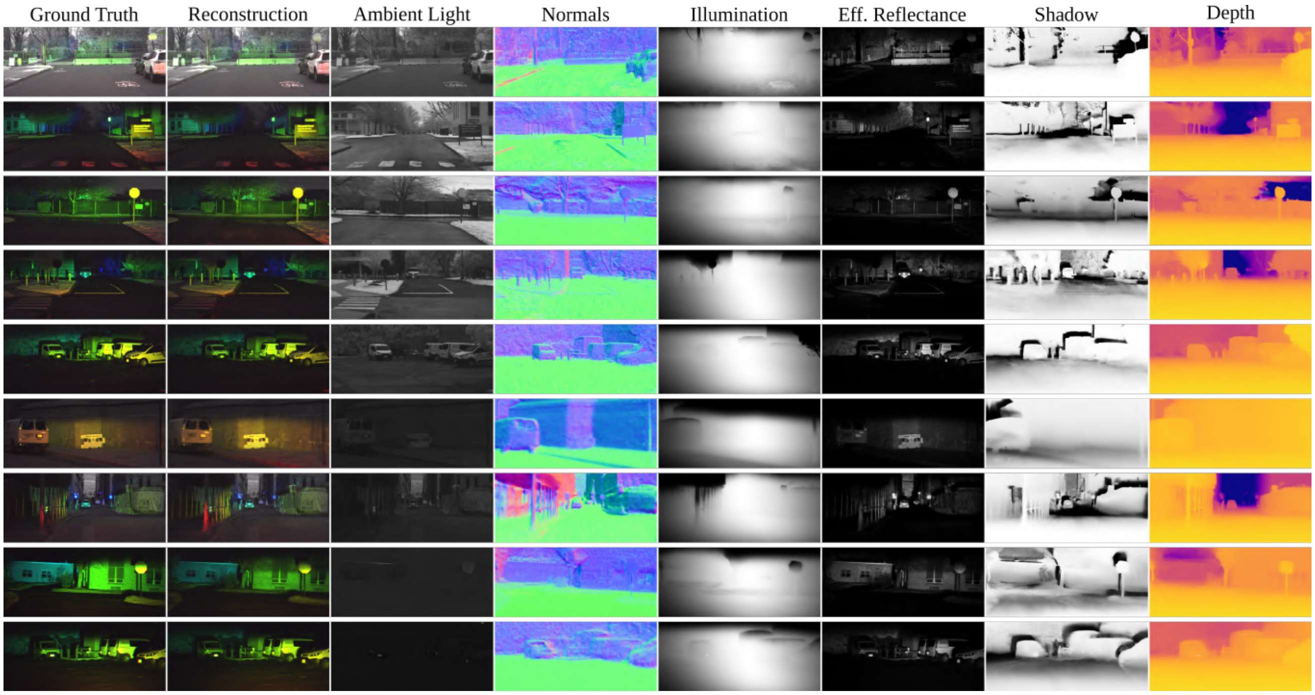


Figure 6. The proposed method allows to reconstruct the given scene and decomposes it into effective reflectance, normals, depth, shadow, laser illumination and passive contribution of the environment for each point.

few LiDAR points are being collected on those structures. The difference is accentuated in nighttime sequences, where Gated Fields is able to reconstruct the scene without the need of ambient light, thanks to the active component generated from the illuminator light pulses. On the contrary, other methods only retrieve small parts of the initial scene and tend to completely fail without explicit depth supervision, as visible in Figs. 10 and 11. By incorporating the gated image formation model, Gated Fields is able to disambiguate areas only visible from a limited range of views, see Fig. 12, where our method is the only one to reconstruct the finer structure of the bush branches, while baseline methods only provide a coarser representation of it.

11. Gated Fields Dataset

We provide here additional details on the Gated Fields dataset, which we collected and curated in order to evaluate our model. Specifically, we collected across North America a set of 10 sequences, 5 during the day and 5 during the night, using a test vehicle which we set up with a Gated stereo and RGB stereo setups, as well as a LiDAR and a GNSS. After data collection, we filter the dataset removing all the sequences containing moving actors, as we are focusing on static scenes. Subsequently, we estimate camera and LiDAR poses as well as construct a ground-truth point cloud, see again Sec. 5.
We show a subset of the scenes in Figs. 13 to 15. These scenes include several different scenarios, such as sub-urban and countryside, visible in Fig. 13 and Fig. 14. A subset of the sequences was also collected in city environments, namely New York and Philadelphia. In Figs. 14 and 15, we also present some of the sequences we recorded at night time. In particular, the scenes ”Forrest” and ”Hedge” are being collected both day and night. This enables us to study the impact of day and night-time conditions on the quality of the reconstructed scene.

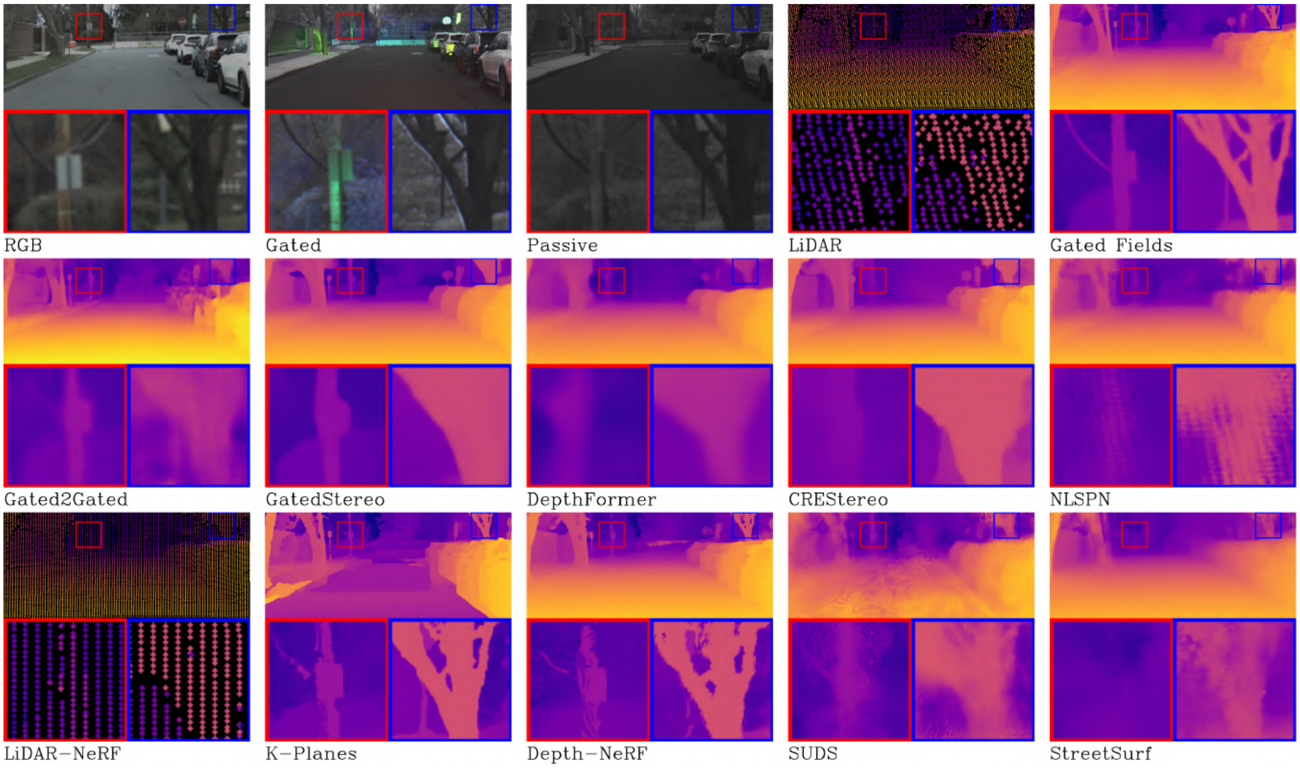


Figure 7. Qualitative comparison of Gated Fields and existing depth estimation methods. For each example, we show the corresponding RGB image, the colored gated image, and the LiDAR measurements. Our method is able to reconstruct much finer details, such as thin wires (red) and tree branches (blue), than state-of-the-art methods.

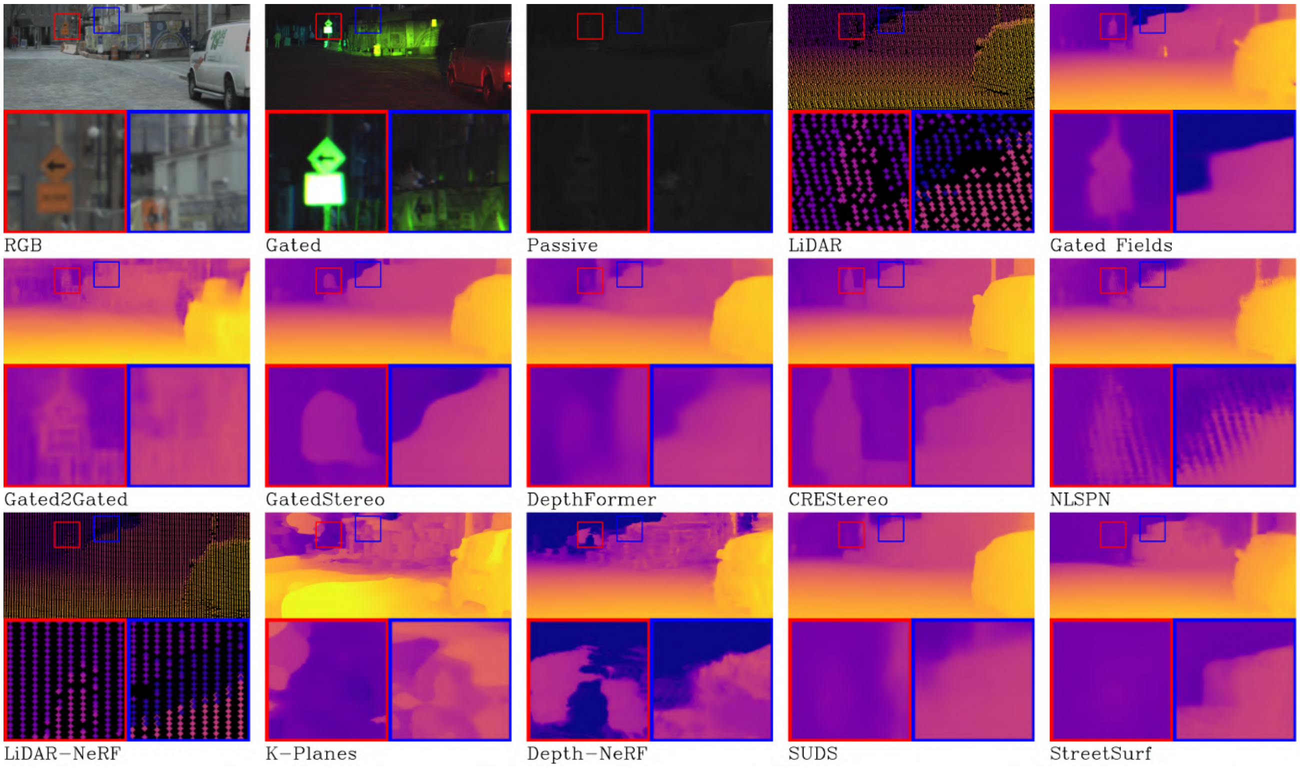


Figure 8. Additional comparison of Gated Fields and existing depth estimation methods. Gated Fields is able to provide accurate depth even for objects in low-contrast areas.

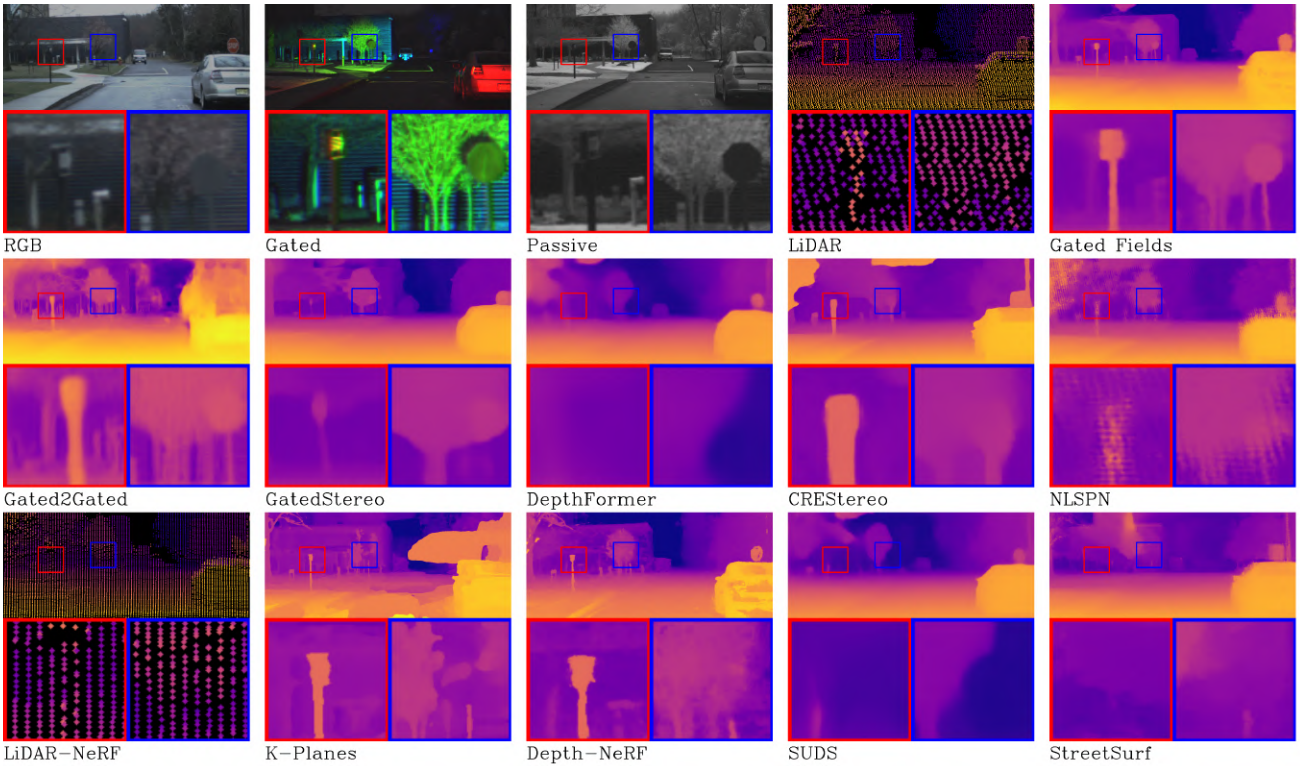


Figure 9. Additional comparison of Gated Fields and existing depth estimation methods for texture-deficient regions. RGB based methods (DepthFormer, CreStereo, K-Planes, SUDs and StreetSurf) suffer in low-contrast regions and e.g. are missing the stop sign (blue) in their depth estimates.

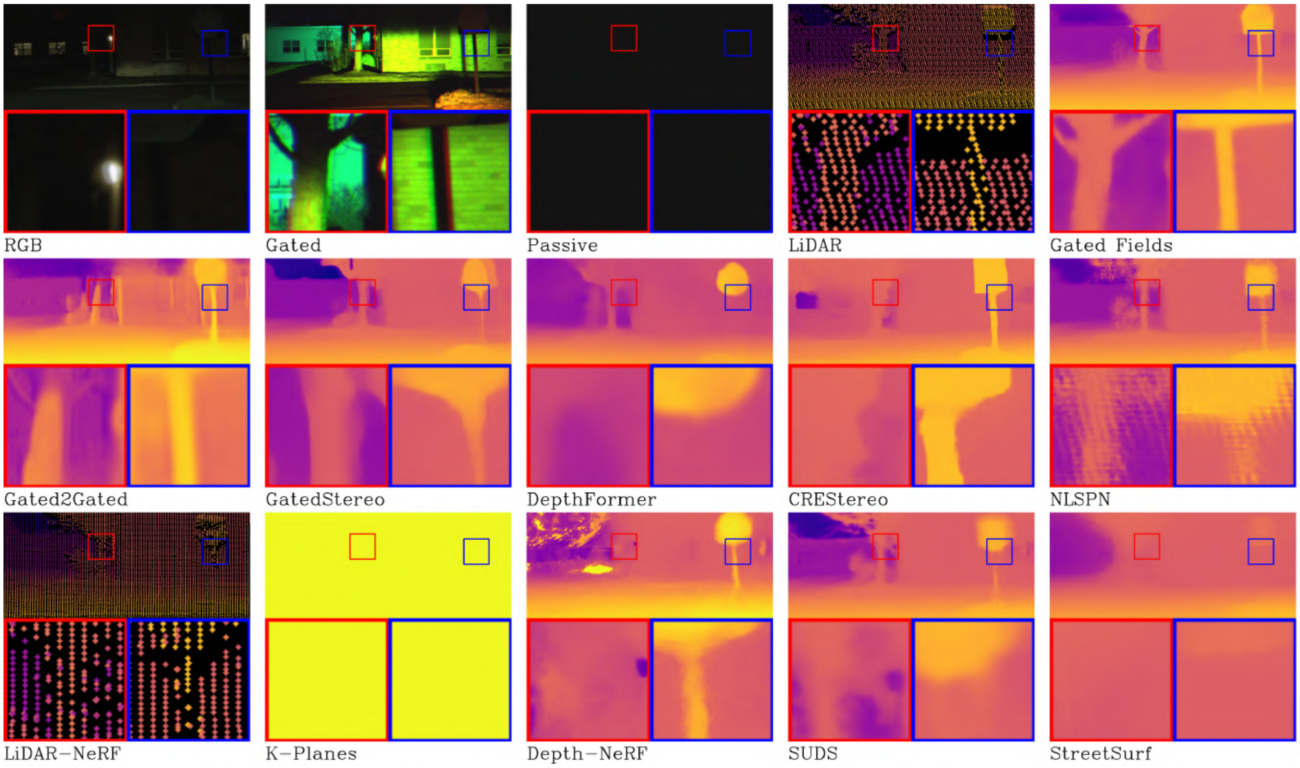


Figure 10. Additional comparison of Gated Fields and existing depth estimation methods for night time conditions. Due to the active illumination, Gated Fields is capable to reconstruct objects with accurate edges even for far distances.

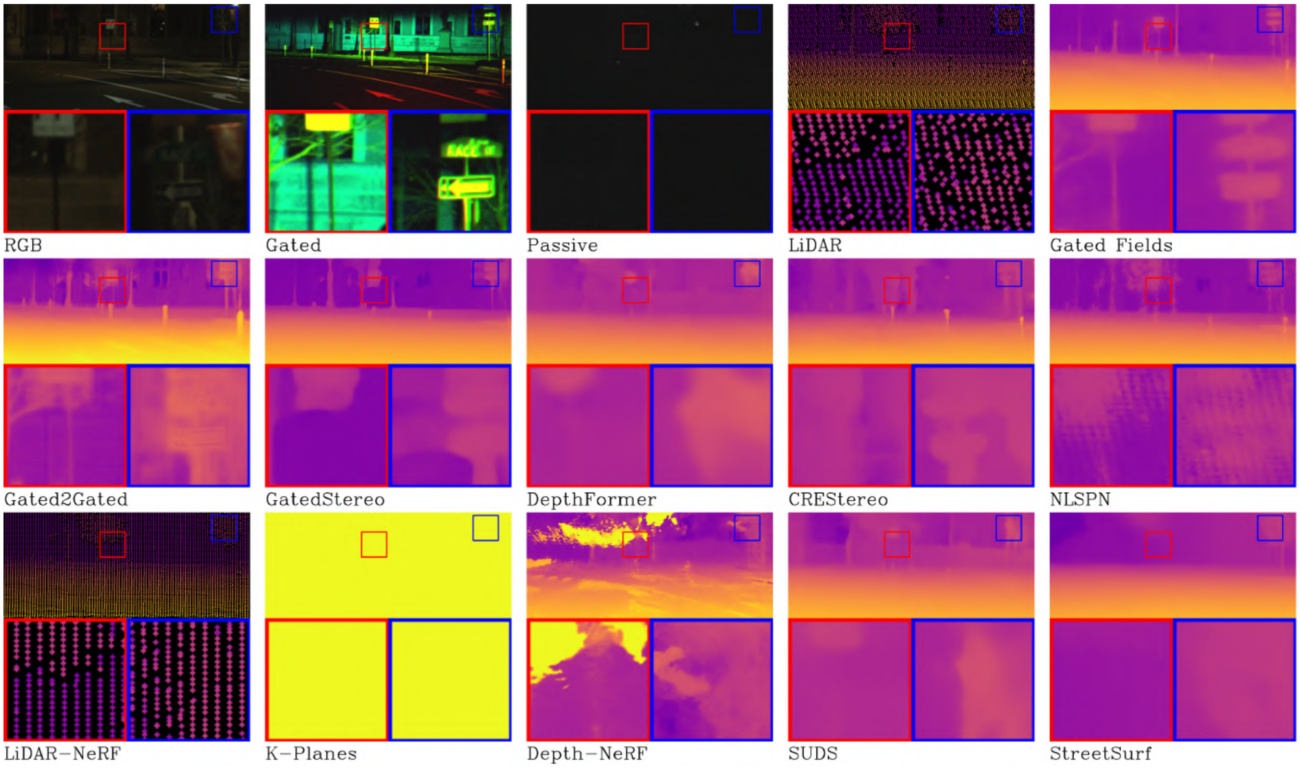


Figure 11. Additional comparison of Gated Fields and existing depth estimation methods for fine structures in night time conditions. Especially K-Planes has problems to reconstruct accurate depth due to missing contrast information in the RGB images.

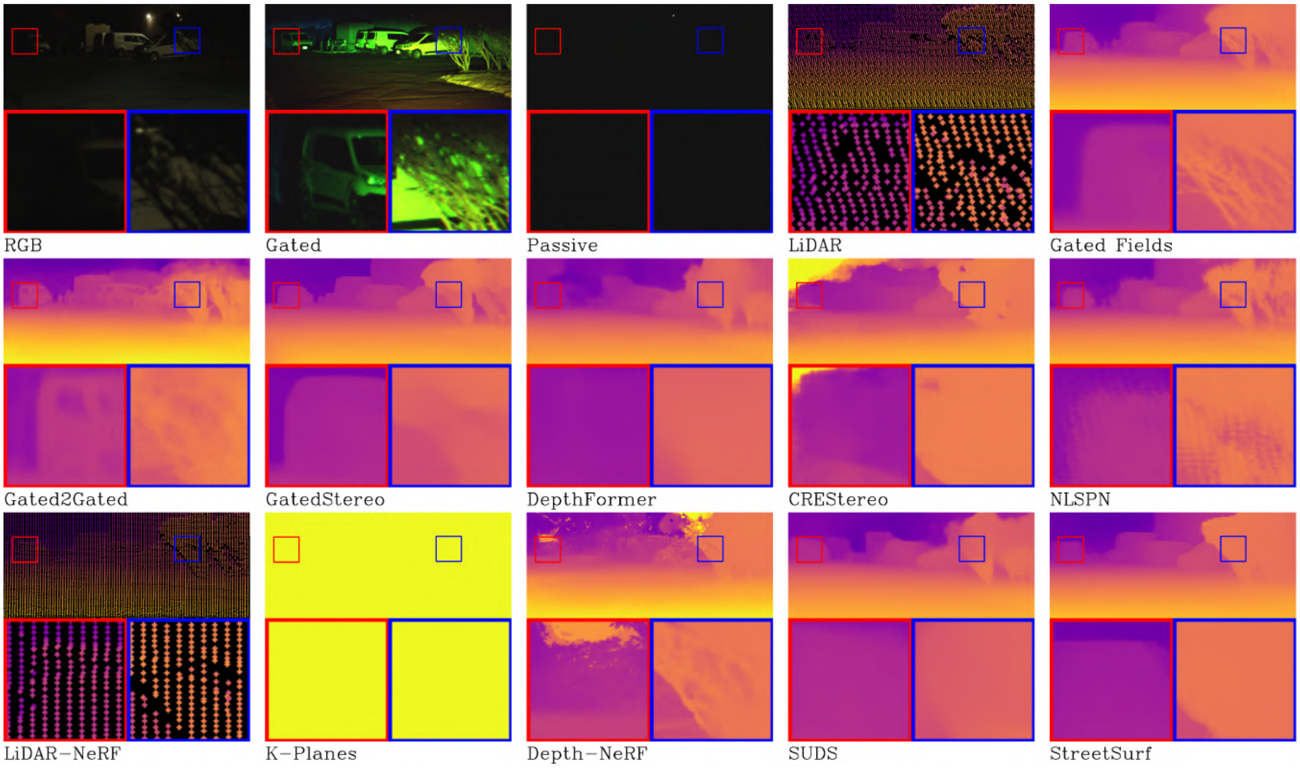


Figure 12. Additional comparison of Gated Fields and existing depth estimation methods for night time conditions. RGB based methods suffer from the lack of ambient light and are not able to provide reasonable depth estimate.

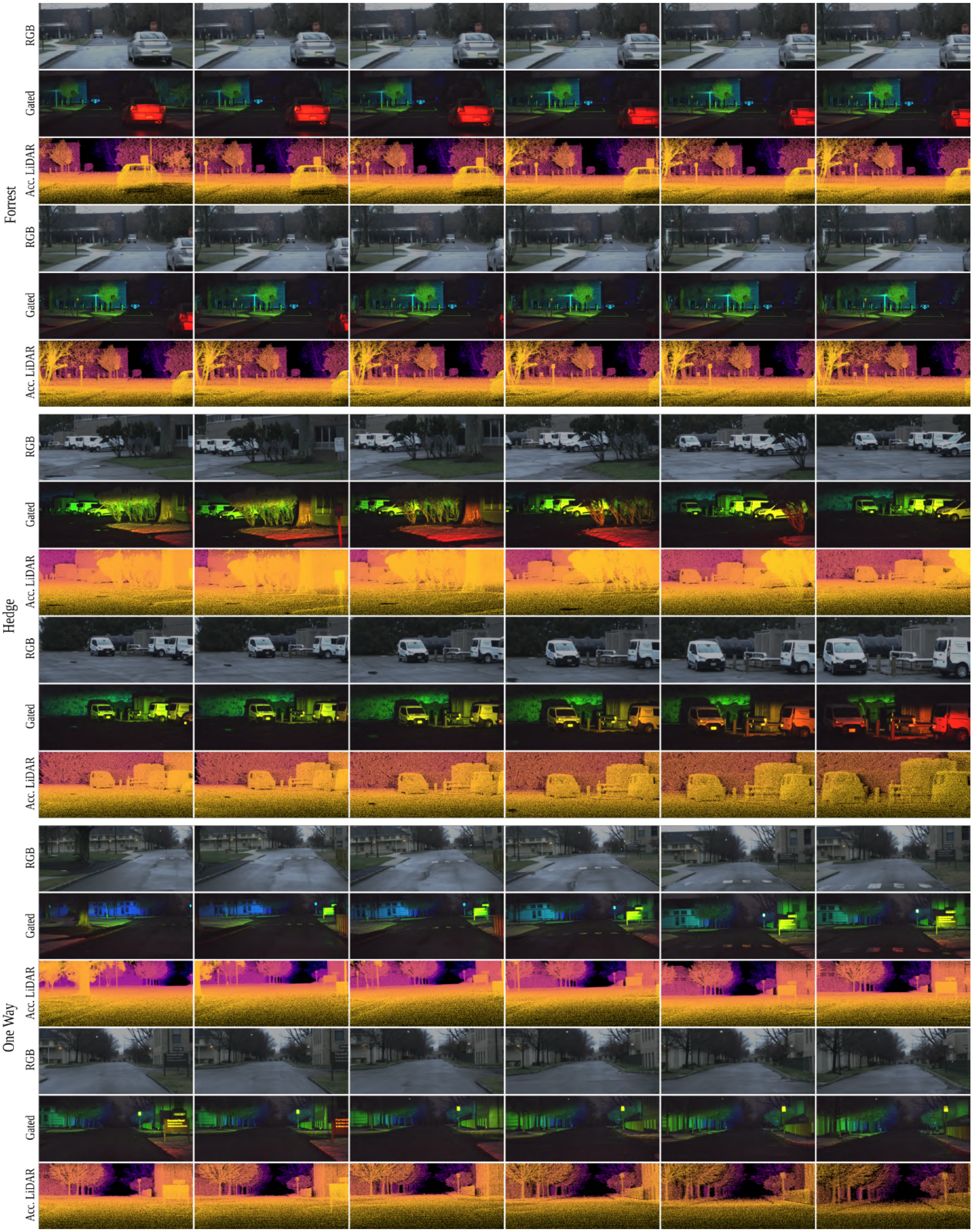


Figure 13. Examples from our Gated Fields Dataset. We show RGB, gated image and accumulated LiDAR ground-truth depth.

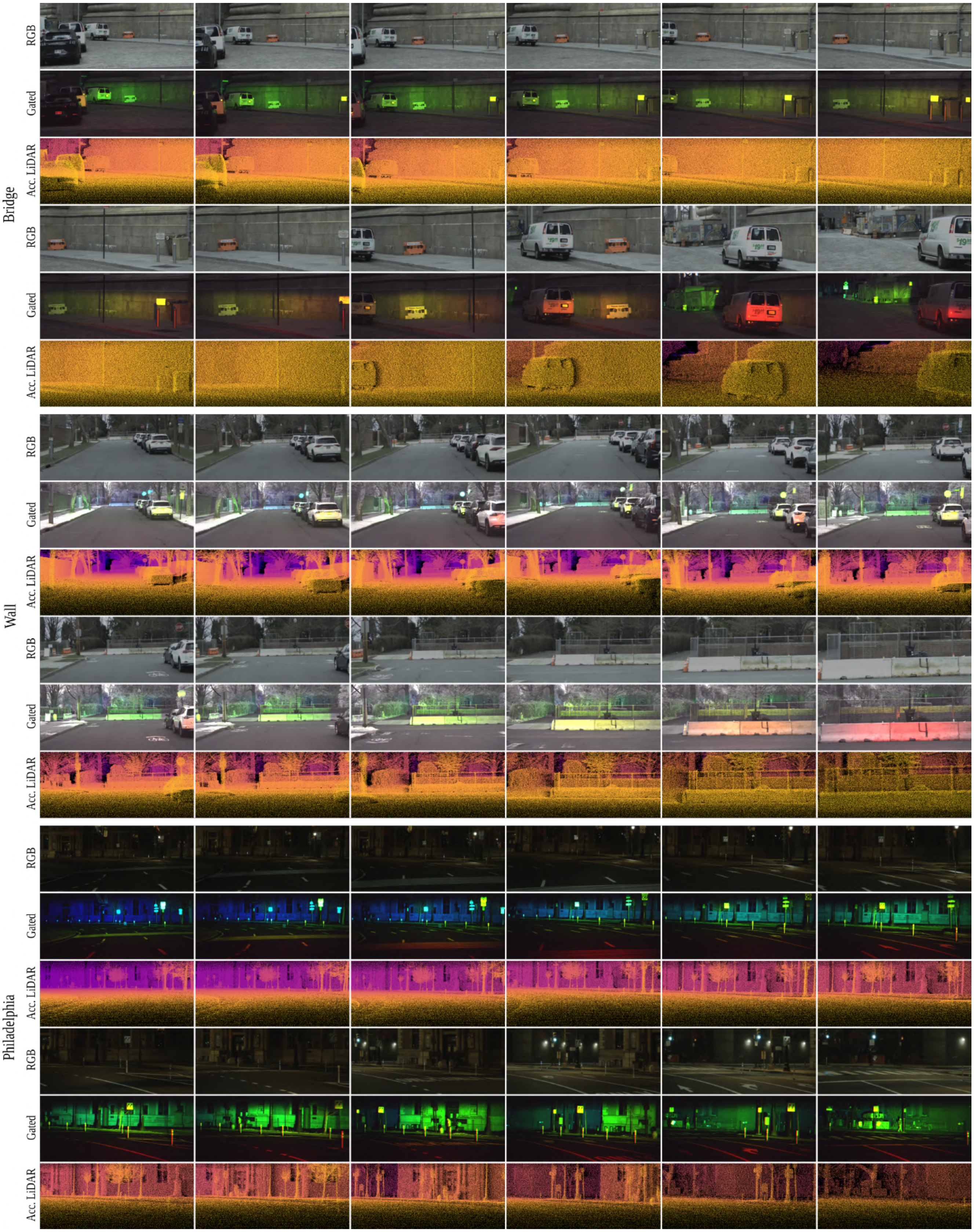


Figure 14. Examples from our Gated Fields Dataset. We show RGB, gated image and accumulated LiDAR ground-truth depth.

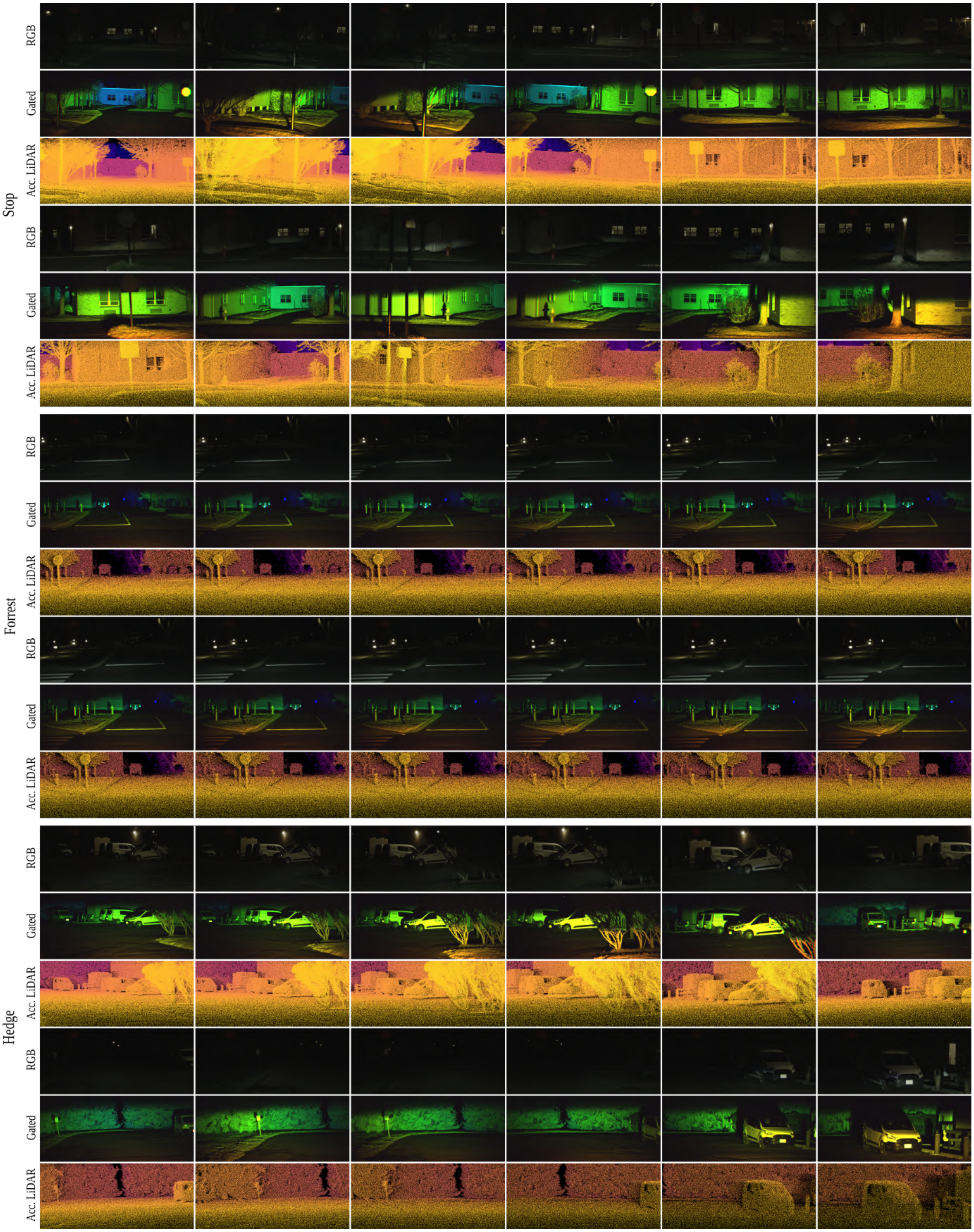


Figure 15. Examples from our Gated Fields Dataset. We show RGB, gated image and accumulated LiDAR ground-truth depth.

References

[1] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5470–5479, 2022. 2

[2] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020. 2

[3] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Transactions on Graphics (ToG)*, 41(4):1–15, 2022. 2

[4] Tixiao Shan, Brendan Englot, Drew Meyers, Wei Wang, Carlo Ratti, and Daniela Rus. Lio-sam: Tightly-coupled lidar inertial odometry via smoothing and mapping. In *2020 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 5135–5142. IEEE, 2020. 3

[5] Matthew Tancik, Ethan Weber, Evonne Ng, Ruilong Li, Brent Yi, Terrance Wang, Alexander Kristoffersen, Jake Austin, Kamyar Salahi, Abhik Ahuja, et al. Nerfstudio: A modular framework for neural radiance field development. In *ACM SIGGRAPH 2023 Conference Proceedings*, pages 1–12, 2023. 2