Gated2Depth: Real-time Dense Lidar from Gated Images (Supplement)

Tobias Gruber[1,3] Frank Julca-Aguilar[2] Mario Bijelic[1,3] Werner Ritter[1] Klaus Dietmayer[3] Felix Heide[2,4]

[1]Daimler AG [2]Algolux [3]Ulm University [4]Princeton University

1. Additional Dataset Details

In this section, we describe additional details on the synthetic and measured datasets used for the evaluations in the main draft and this supplemental document.

1.1. Synthetic Dataset

Recently, a growing set of synthetic datasets for driving tasks have been proposed, all generated by advanced real-time game engines [4, 8, 12, 13, 14]. While these existing datasets contain RGB images and depth maps, they do not contain enough information to synthesize realistic gated measurements that require NIR reflectance modeling and passive sunlight-illumination. We modify the GTA5-based simulation framework from [12] to address this issue. Using Renderdoc[1], every drawcall and buffer can be read out from the GPU while playing GTA5 [12]. Renderdoc is a free MIT licensed stand-alone graphics debugger that is able to read every drawcall and buffer from the GPU while playing GTA5 and was first used in [12] for creating automotive synthetic datasets. Adrian Courrèges[2] provides a detailed description of the rendering process of GTA5. For the simulation of gated images, the G buffer and the depth-stencil buffer are of interest to us. Figure 1 visualizes the drawcall data that can be extracted, namely diffuse reflectances, normal vectors, specular reflectance, glossiness, sunlight illumination, active lights and depth map. Moreover, the projection matrix has to be extracted in order to obtain 3D points. To generate high-resolution synthetic measurement data, GTA5 is played in 4k resolution (3840x2146) on a workstation with a Nvidia Titan Xp. For best rendering quality, we enable anti-aliasing (FXAA and MSAAx2). For each recording session, we drive randomly through Los Santos by using VAutodrive[3] and capture a frame every 10 seconds.

The gated simulation consists of five processing steps that will be explained in the following: Camera adaptation, NIR model, laser illumination, gating and saturation and blooming. Figure 2 shows intermediate example outputs of this processing pipeline.

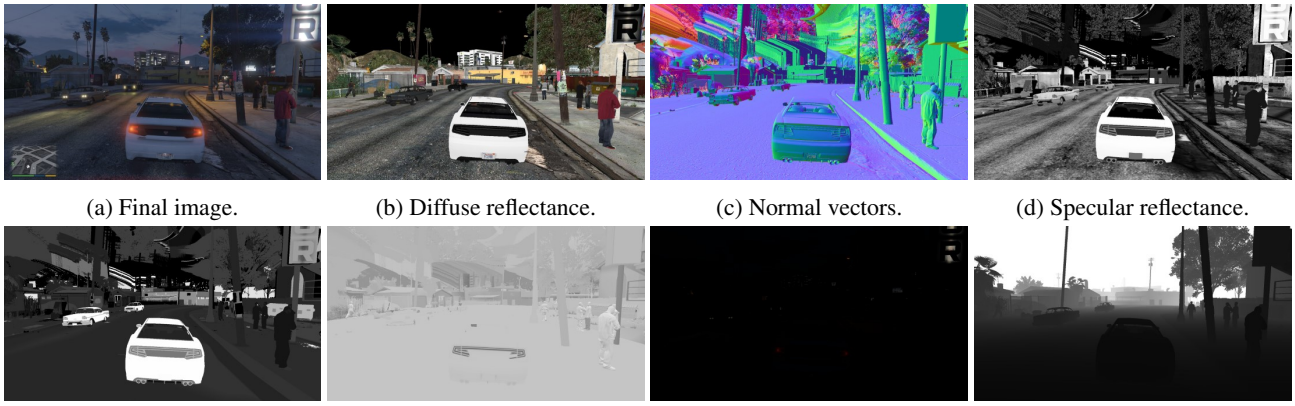
1.1.1 Camera Adaptation

With the projection matrix extracted from the GTA5 rendering process, each pixel can be projected into 3D space. These 3D points can be projected in the gated frame with the calibrated camera parameters of the prototype gated camera. If, despite 4k resolution, there are pixels in the frame where no 3D points are projected on, information is interpolated by nearest neighbor interpolation. In this way, the GTA5 frame is warped into the gated frame and both resolution and intrinsic calibration are adapted as illustrated in Figure 3.

1.1.2 NIR model

GTA5 does not provide any kind of NIR reflectance and no NIR image formation model. Therefore, the reflectance of the objects in the NIR regime is unknown. Inspired by Megan Kennedy[4], we simulate the approximate NIR reflectance based on the diffuse RGB reflectance. The following steps are performed: First, the original image is inverted (Figure 4b). Then,

[1] renderdoc.org
[2] www.adriancourreges.com/blog/2015/11/02/gta-v-graphics-study
[3] de.gta5-mods.com/scripts/vautodrive
[4] digital-photography-school.com/create-infrared-effect-photoshop



(a) Final image. (b) Diffuse reflectance. (c) Normal vectors. (d) Specular reflectance.

(e) Glossiness. (f) Sunlight illumination. (g) Active lights. (h) Depth map.

Figure 1: Raw data extracted with renderdoc from the drawcalls of GTA5 (Rockstar Games).

original and inverted image are overlaid by selecting for each pixel in each channel the maximum value of both (Figure 4c), namely

$$I_{\text{NIR}} = \max\left(I_{\text{RGB}}, 1 - I_{\text{RGB}}\right). \tag{1}$$

In order to obtain a monochrome image, after swapping the red and blue channel, the following RGB to gray conversion is applied

$$Y = 0.229R + 0.587G + 0.114B, \tag{2}$$

where Y describes the NIR luminance (Figure 4d). Since the image lightens up a lot due to the maximum overlay, gamma correction with $\gamma = 0.25$ is finally applied in order to darken the image (Figure 4e). In Figure 4, all steps from the RGB image to the NIR simulated version are shown. These NIR simulations match qualitatively with the measurements acquired from the prototype system.

1.1.3 Laser Illumination

Due to the mounting position of the laser illuminator below the bumper, shadows appear in the gated images. Matching the mounting position of the flash illumination allows us to accurately model the shadows in the experimental gated images. This effect can be taken into account by transforming the 3D points into a laser coordinate system and then determining which points are illuminated and which not.

1.1.4 Gating

As described in the main paper, gating is applied by using

$$I\left(r\right) = \alpha\, C\left(r\right) = \int_{-\infty}^{\infty} g\left(t - \xi\right) \kappa\left(t, r\right) \mathrm{d}t, \tag{3}$$

where ξ is the designated delay between start of illumination and start of exposure, $g\left(t - \xi\right)$ is the rectangular gating function with duration t_G and $\kappa\left(t, r\right)$ is the temporal scene response assumed to be

$$\kappa\left(t, r\right) = \alpha p\left(t - \frac{2r}{c}\right) \beta\left(r\right), \tag{4}$$

where $p\left(t - 2r/c\right)$ is also rectangular with duration t_L. The atmospheric effects can be modeled by

$$\beta(r) = \frac{P_{\text{laser}} \tau_{\text{optics}}}{4\pi r^2 \tan\left(\frac{\theta_H}{2}\right) \tan\left(\frac{\theta_V}{2}\right)} \frac{\rho^2}{F_{\text{num}}^2} \frac{\lambda}{hc} e^{-2\gamma r} \tag{5}$$

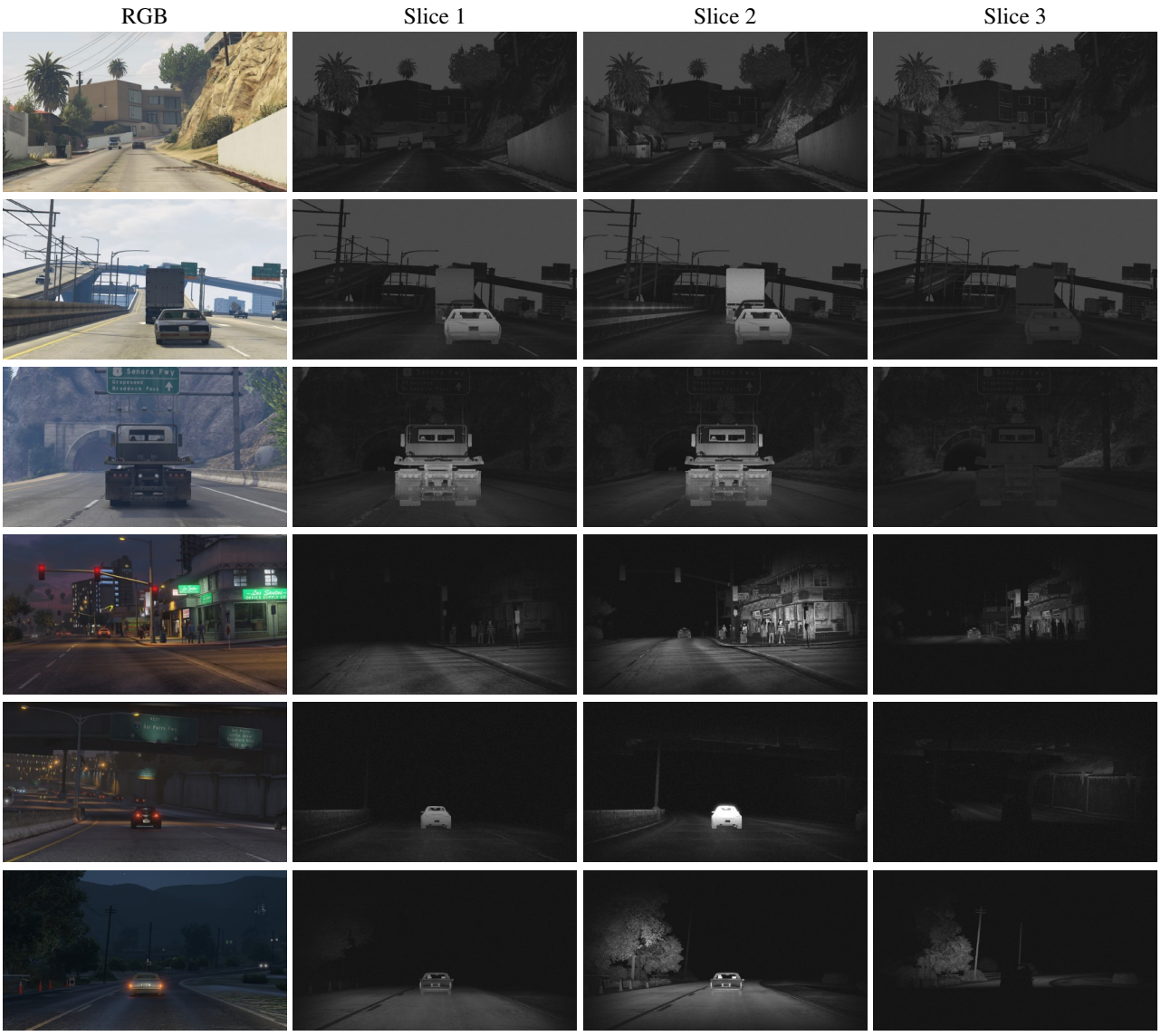
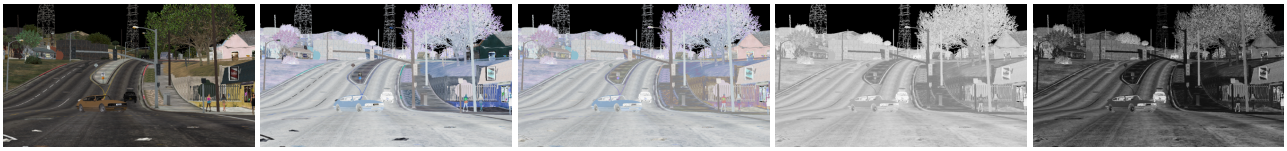


Figure 2: This figure shows simulation examples of gated slices. We show for each example the reference RGB image and all three slices.

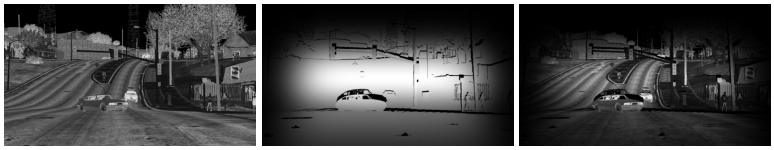
Figure 3: Steps for adapting the intrinsic calibration and resolution from the original GTA camera to the gated camera.

where P_{laser} is the laser power, θ_H, θ_V the horizontal/vertical field of illumination, ρ the pixel pitch, F_{num} the aperture, λ the wavelength, h the Planck constant and γ atmospheric attenuation coefficient. In order to improve the SNR, multiple pulses are required before read-out as described in Table 2. Details on the laser and camera specifications can be found in Table 1.



(a) RGB. (b) RGB inverted. (c) Maximum overlay. (d) Grayscale conversion. (e) NIR.

Figure 4: From left to right: original RGB image, inverted RGB image, maximum overlay of original and inverted RGB image, grayscale converted overlay image, final heuristically simulated NIR color model image after gamma correction.



(a) NIR reflectance. (b) Illumination. (c) Full gated image.

Figure 5: From left to right: heuristically simulated NIR reflectance, laser illumination, combined NIR reflectance and laser illumination.

Laser		
Laser Power	P_{laser}	500 W
Wavelength	λ	808 nm
Horizontal Field of Illumination	θ_H	24°
Vertical Field of Illumination	θ_V	8°
Camera		
Pixel pitch	ρ	10 µm
Aperture	F_{num}	1.2
Optical transmission	τ_{optics}	0.64
Focal length	f	23 mm
Horizontal Field of View	θ_H	31.1°
Vertical Field of View	θ_V	17.8°
Resolution		1280x720

Table 1: Laser and camera specifications.

The photon-to-pixel conversion gain, delays due to signal runtime, and the dark level of the camera are all calibrated with distant-dependent measurements at night on targets with defined reflectivity. Figure 6 shows the measurements of a target with 50 % reflectivity and the model from Eq. (3). For simulation, we cannot use Chebychev approximations as in the main paper for least-squares estimation because these are only valid within a certain range. Moreover, directly using the calibrated physical model allows us to quickly change the parameters of the gating system and create gated images of arbitrary range-intensity-profiles. For daylight image simulations we simply add a passive component according to the passive exposure duration and a varying ambient illumination level for each image.

For the final image, we applied the Poissonian-Gaussian noise model given by

$$z = I(r) + \eta_p\left(I(r)\right) + \eta_g, \tag{6}$$

where the noise parameters are calibrated from a BrightwayVision BrightEye camera using Foi et al. [3].

Three slices are simulated with the parameters that are set up in the real camera as shown in Table 2. These settings allow to theoretically record slices from 3-72 m, 18-123 m and 57-176 m respectively. The exact ranges have to be calibrated due to signal runtimes and rise and fall times of the laser and the gate. Figure 2 shows example simulations.

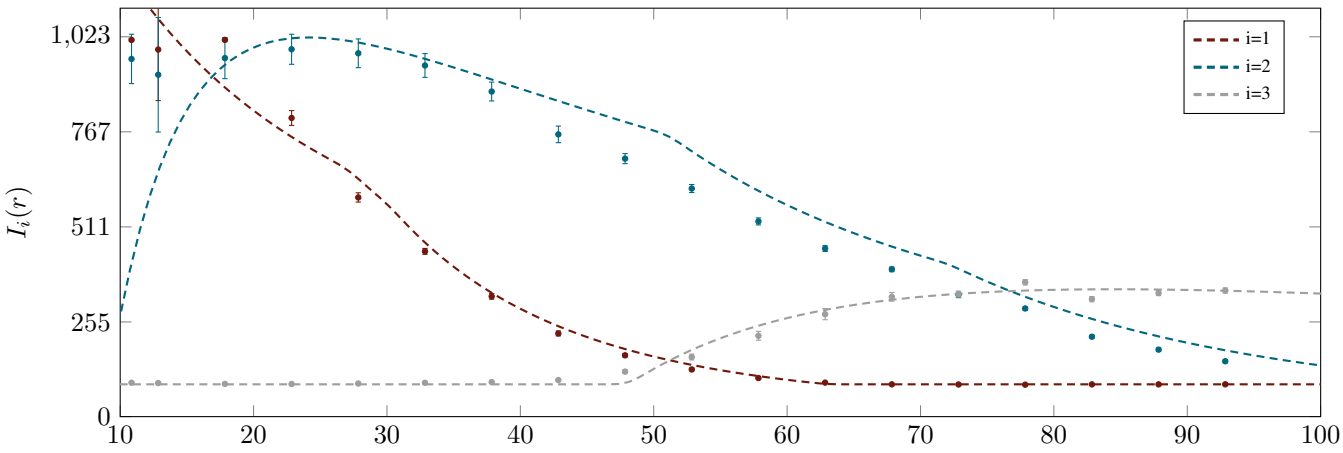


Figure 6: Measurements on a 50 % reflective target of the three range-intensity profiles $I_i(r), i \in \{1, 2, 3\}$ used in this work, and our modeled range-intensity profiles according to (3) plotted with distance r [m].

	t_L	t_G	ξ	pulses
Slice 1	240 ns	220 ns	260 ns	202
Slice 2	280 ns	420 ns	400 ns	591
Slice 3	370 ns	420 ns	750 ns	770

Table 2: Gating parameters.

(a) Full gated image. (b) Slice 1. (c) Slice 2. (d) Slice 3.

Figure 7: Gating applied according to the gated parameters in Table 2.

(a) No effects. (b) With camera noise, saturation and blooming.

Figure 8: We include three important camera effects: noise, saturation and blooming. This figure shows the influence of these components.

1.1.5 Saturation and Blooming

In addition to the noise model with calibrated parameters from the camera prototype, we include saturation and blooming effects into the simulation. We mask out the saturated areas by thresholding the image at the sensors bit depth. Then, multiple airy-disk filters with decreasing sigma are applied and added to the image. Finally, the image is clipped to the sensor bit depth.

Figure 9: The sensor setup of our test vehicle for validation is shown. The car is equipped with a gated camera, consisting of a flood-light flash source, gated imager and a scanning lidar Velodyne HDL64-S3D as reference.

1.1.6 Stereo

Obtaining a second RGB image from another perspective is super challenging because it would require another rendering pass from a slightly different perspective. However, in GTA5 there exist only a few different camera positions with large displacement.

1.2. Real Dataset

For experimental validation, we have equipped a testing vehicle with a standard RGB stereo camera (Aptina AR0230), lidar system (Velodyne HDL64-S3) and a gated camera (BrightwayVision BrightEye) with flood-light source integrated into the front bumper, shown in Figure 9. Both cameras are mounted behind the windshield, while the lidar is mounted on the roof. The stereo cam runs at 30 Hz with a resolution of 1920x1080 pixels. The gated camera provides 10bit images with a resolution of 1280x720 at a framerate of 120 Hz, which we split up in three slices plus an additional ambient capture without active illumination. The car is equipped with two vertical-cavity surface-emitting laser (VCSEL) modules, which are diffused, with a wavelength of 808 nm and a pulsed optical output peak power of 500 W each. The peak power is limited due to eye-safety regulations. Our reference lidar systems is running with 10 Hz and yields 64 lines. All sensors are calibrated and time-synchronized. During a four-week acquisition time in Germany, Denmark and Sweden, we recorded 17,686 frames in different cities (Hamburg, Kopenhagen, Gothenburg, Vårgårda, Karlstad, Örebro, Västerås, Stockholm, Uppsala, Gävle, Sundsvall, Kiel). In total, we recorded about 24 h of sensor data. We select samples every 5 s. The whole sensor setup runs with Robot Operating System (ROS)[5] that allows to record all sensors in a common framework. For time synchronization, we extended the ROS approximate time synchronizer such that the sequential gated slices are assembled before time synchronizing with the lidar system and the stereo camera, otherwise it is not guaranteed that the time synchronized sequential gated slices belong to the same recording shot.

2. Outdoor Automotive ToF vs. Indoor Consumer ToF

In contrast to consumer ToF cameras for indoor applications, automotive applications require long ranges and hence high laser power. However, due to eye-safety regulations, laser illumination power is limited and hinders a high signal especially in the NIR spectrum from larger ranges. Changing to SWIR spectrum would enable the use of much higher illumination powers but currently also at a much higher cost. Current automotive ToF applications, therefore, suffer under bright daylight conditions, where it is difficult to illuminate the scene such that the active illumination overcomes the passive component. Morever, in constrast to indoor ToF systems, shadows are inevitable because larger illuminators are required and cannot be placed around the camera.

[5] ros.org

Figure 10: Accumulating three gated slices yields a single unmodulated exposure, full gated image, with continuous illumination.

Gated and ToF sensing can both be expressed as amplitude-modulated correlation imaging [1, 6], where a sensor acquires modulated exposures, e.g. gated or sinusoidal modulation, of an amplitude-modulated flood-lit scene. As such, in future work, the proposed network and training approach might also be applied to ToF correlation cameras with the phase images as input instead of gated slices. In contrast to existing methods that use custom CCD sensors, such as [1] with concurrent gating of 1.5 ns width, or [9] as a correlation sensor, we use a much simpler and low-cost CMOS sensor with 30 ns rise time and sequential capture. The main benefit of our method is that it not only exploits the correlations between the gated slices at pixel level but incorporates non-local semantic context.

3. Design of Exposure Profiles

The design of the exposure profile, also called range-intensity profile, is important for depth estimation but not trivial. In this work we do not focus on optimal exposure profile design, but show that depth can be estimated from non-local image information and using this context information helps to achieve dense reconstructions, and tackle multi-path and shadow effects that appear in pixel-based methods. The parameters that are used in this work are shown in Table 2.

Since the slices have to be recorded sequentially, increasing the number of slices means increasing the overall acquisition time for a set of slices, if the SNR per slice is kept constant. In case of the BrightwayVision BrightEye system, capturing a single slice takes about 8.3 ms. For a system that can run at 30 Hz we selected three active illuminated slices and a passive image (defined as below).

We manually design the exposure profiles. The three slices have to be overlapping for depth estimation. Covering a long range by a small number of slices means increasing the width of the slices, thus long gate and laser duration. The higher the delay and thus the distance of the slice, the lower the number of returning photons. In order to integrate more photons on the chip, we increased both the number of pulses, laser duration and gate duration for slices further away (see Table 2). The first slice starts after the end of the laser pulse plus a guard interval of 20 ns and therefore covers the close range up to 60-70 m. To cover ranges up to 150 m, another broad slice starting at approximately 50 m is required. In order to have overlapping slices, another slice in between is selected. Due to eye safety and heat development, the total laser duration is constraint to be less than 500 µs in a slice iteration of 33 ms. This limits both range and the number of active slices.

We use the laser recovery time in between to record a passive slice without any laser illumination, *passive image*. This facilitates gated imaging during day where the passive image can be simply subtracted. The different number of pulses in the three slices results in different passive components. Therefore, additional passive exposures are added to the first two slices to ensure the same passive component in each gated image. Although the overall image lightens up, this simplifies the removal of the passive component. By integrating all three slices, a *full gated* image is obtained that provides a fully illuminated scene understanding without gating.

4. Effect of Motion

Our gated camera prototype captures frames at 120 Hz which means capturing three active and one passive type takes 33 ms. At a speed of 100 km/h the car moves less than 1 m during this acquisition time, which is sufficient for all tested driving scenarios. For adequate training and evaluation, we performed egomotion compensation of the lidar pointcloud.

5. Full Gated images

Accumulating the three gated slices provides a single NIR exposure without gating and with continuous illumination which we refer to as full gated image, see Figure 10. Figure 11 and Table 3 show that the proposed method based on three slices significantly outperforms estimation from a full gated image input. Hence, the depth cues contained in the differences between the individual gated slices are essential for the proposed approach.

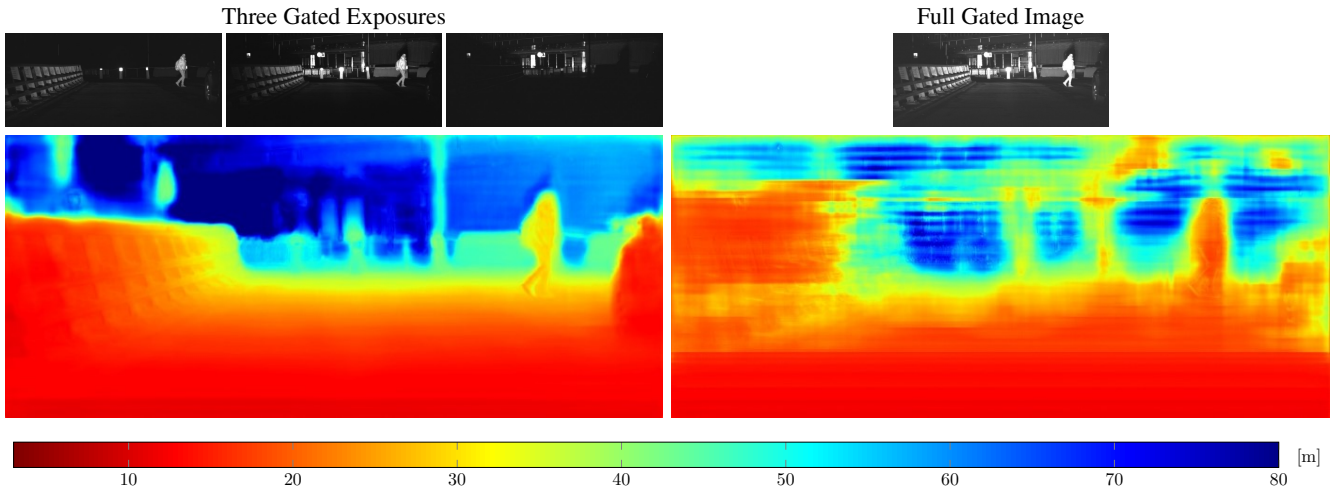


Figure 11: Depth estimation using three gated slices and a full gated image. See Figure 15 for this capture and further comparisons.

6. Least-Squares Baseline

In this section, we give more details about the pixel-based least-squares algorithm. Specifically, for a single pixel, we stack the measurements $z_{\{1,2,3\}}$ for a sequence of delays $\xi_{\{1,2,3\}}$ in a single vector $\mathbf{z} = [z_1, \ldots, z_3]$. We can estimate the depth and albedo jointly as

$$\hat{r}_{LS} = \underset{r,\alpha}{\operatorname{argmin}} \left|\left|\mathbf{z} - \alpha\tilde{\mathbf{C}}(r)\right|\right|_2^2, \tag{7}$$

where $\tilde{\mathbf{C}}(r) = [\tilde{C}_1(r), \ldots, \tilde{C}_3(r)]$ is a Chebychev intensity profile vector. Since the range-intensity profiles are non-linear, we solve this nonlinear least-squares estimation using the Levenberg-Marquardt optimization method. We rely on auto-differentiation and use the SciPy wrapper leastsq around MINPACK's lmdir and lmder algorithms [11]. As prior estimate, distance is set to 50 m and albedo to 0.5. Note that least-squares depth estimation takes multiple minutes for a full image, and that the method requires high signal-to-noise ratio of each measurement pixel.

7. Effect of Ambient Illumination

Ambient illumination limits the performance of gated imaging because it cannot be avoided to integrate an additional passive component on the sensor. The probably easiest method to remove this passive component is to record an additional passive capture with no laser illumination and subtract it from the active slices. In order to prevent the laser from overheating, the number of laser pulses in a certain time is limited and therefore a passive image can be obtained at no cost during laser recovery.

We noticed that the Gated2Depth network is able to learn to deal with the ambient illumination and then subtraction of the passive component is not required. Pixel-based methods such as regression tree [1] and least-squares require to subtract the passive component before, because these methods do not have the full image context for dealing with ambient illumination. It is possible to estimate the ambient illumination in the least-squares approach as additional open parameter but these experiments were unstable and yielded worse results than with passive image subtraction.

8. Network evaluation

In this section, we describe the systematic design evaluation we followed to find the presented *Gated2DepthNet* model and loss functions. While we performed this evaluation using the real data, with corresponding lidar data as ground-truth, we also present additional synthetic validation at the end of this section, further validating the proposed method. We designed our experiments with three goals in mind: high accuracy, high efficiency (we aim at models that offer real-time inference), and high quality dense maps (discouraging models that overfit the sparse lidar data and produce low quality dense outputs).

Model	**RMSE** $[m]$	**ARD**	**MAE** $[m]$	$\boldsymbol{\delta_1}$ [%]	$\boldsymbol{\delta_2}$ [%]	$\boldsymbol{\delta_3}$ [%]
Input data:						
Vanilla	8.87	0.17	4.31	84.31	91.77	94.78
Vanilla + passive img.	8.85	0.17	4.29	84.31	91.76	94.72
Fine-tuned Vanilla	8.70	0.16	4.08	85.94	92.48	95.20
Net. architecture:						
2 Down conv.	10.94	0.24	5.93	72.79	87.37	92.41
5 Down conv.	8.98	0.20	4.50	84.11	91.82	94.89
3D conv.	10.85	0.27	6.40	71.65	86.86	91.98
Loss function:						
Proposed (ℓ_1 loss)	8.87	0.17	4.31	84.31	91.77	94.78
Proposed (proposed loss)	8.56	0.16	3.97	86.52	92.87	95.42

Table 3: Dense depth estimation results for our proposed model and different configurations considering input data, network architecture and loss functiions.

First, we define a vanilla model as a network that only consists of the U-net variant proposed in the main document. To train this vanilla model, no multi-scale, smoothness or adversarial loss component is used. The model is trained using only real data, and a plain ℓ_1 loss. We then evaluate different configurations of the input data, network architecture, and loss functions.

As part of the input data evaluation, in addition to using only gated images, we evaluate the use of a *passive image* of the scene as further channel (*Vanilla + passive img.*). We refer to the passive image as a gray scale image additionally captured as described in Section 3. As a natural alternative approach, we also evaluate a model that is pre-trained using the synthetic dataset and then fine-tune it using the real data (*Fine-tuned vanilla*).

As alternative network architectures, we evaluate the use of 3D convolutions (*3D conv.*), and different number of down/up convolutions (2/5 *down conv.*). For the 3D conv. variants, we keep the number of layers and filters per layer in the vanilla generator, and introduce filters with depth 1 or 2 in the temporal (gated) dimension in the first layer, and 3 in the rest of the layers. Note that for the first layer we limit the depth up to 2 due to the number of gated slices (3) – a depth 3 filter would correspond to a 2D filter as used in the vanilla model. For the pooling layers, similarly to the vanilla model, we define kernels of size 2x2 in the height and width dimension; and 1 for the temporal dimension. As shown in the main manuscript, some of the pixel-wise methods are able to partially capture accurate depth details. In terms of a network architecture, this suggests that simpler generators, for example with only two down/up concolutions, might also perform well on depth estimation and at the same time offer faster inference. We then evaluate different numbers of down/up convolutions in the generator and measure the accuracy of these alternative versions.

Table 3 shows the results of the model analysis. Evaluating different input data configurations, we observe that the passive image does not significantly improve accuracy. Using synthetic data for pre-training the vanilla network shows a considerable improvement over the vanilla, giving insight into the effect of the adversarial loss component. Recall, that (as described in the main manuscript) our proposed model relies on a a discriminator pretrained on synthetic data and keeps it fixed during training on real data. We visualize the improvement for the individual loss function components further below in Figure 13.

Among the evaluated network architectures, we did not find improvements when using 3D convolutions or modifying the number of down/up convolutions. Based on these results, our proposed model considers only 3 gated images as input, 2D convolutions, and uses a generator pretrained on synthetic data.

We can see in Table 3 that our proposed model outperforms all other evaluated models. Figure 12 shows a qualitative comparison on real data, comparing the proposed model and the vanilla one. In comparison to the vanilla model, we can see that our proposed model produces dense depth estimations with fine details. We can also see that the proposed model removes artefacts around the objects and at the top borders of the maps.

We also show synthetic results, comparing against the ground-truth dense depth maps present in this case. Figure 13 shows the the outputs of our model trained with the different proposed loss functions and the vanilla model. We can see that networks trained with multi-scale loss and GAN produce dense depth estimations with more fine grained details, and better defined edges. Networks trained with the smoothness component, on the other hand, produce less noisy depth estimations.

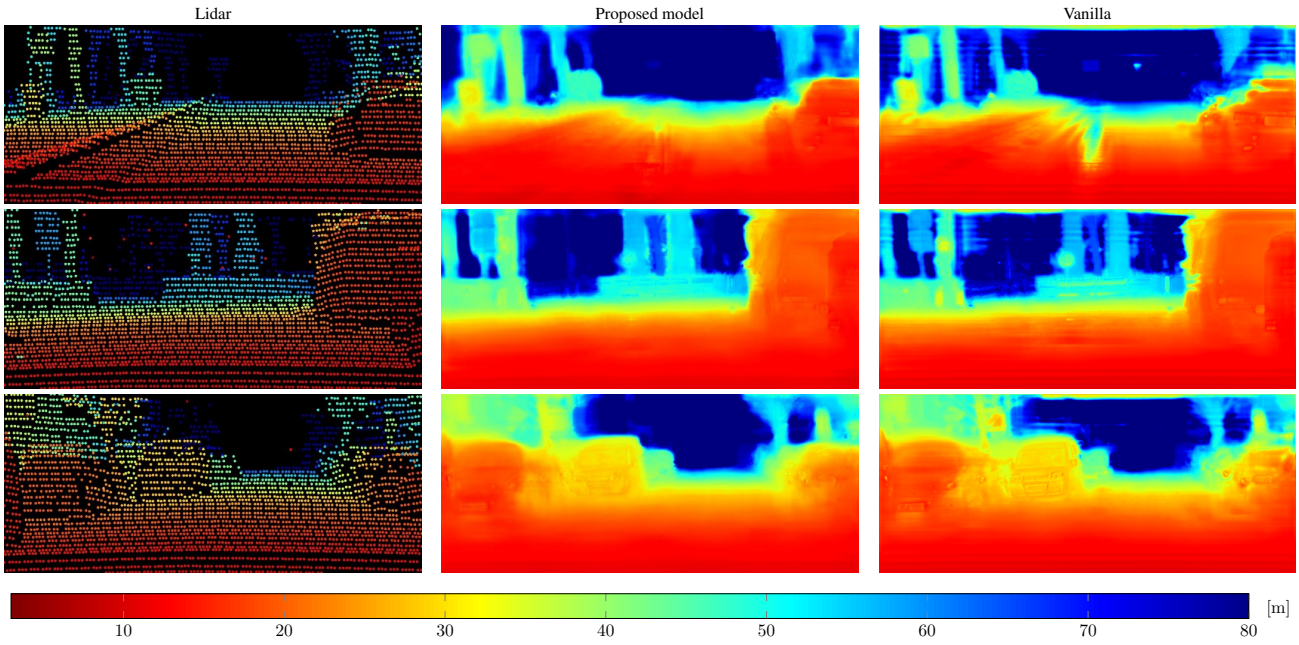


Figure 12: Experimental qualitative results of our proposed model and the vanilla model. Each row show an output for an specific example. The first column illustrates the sparse lidar data of each example

9. Additional 3D Visualizations

We show an additional 3D pointcloud visualization in Figure 14. These visualizations demonstrate the detail and high-density generated from our proposed method. Only at the edges of objects some smearing can be observed.

10. Reference Methods

In this section, we provide additional details about the reference methods. We finetuned the unsupervised monocular method [5] on the stereo images of the training split starting from the best available model. The results improve slightly compared to the model without finetuning and our method still outperforms this approach in all metrics, offering multiple meters improvement in MAE. Our dataset provides only sparse lidar points as ground truth data and there are no accumulated pointcloud scans. Since the best available stereo model [2] already generalizes well to our data, we waive finetuning. The traditional stereo algorithm [7] runs directly on a FPGA and is already adapted to our camera settings.

11. Additional Evaluations

In this section, we show additional results comparing the proposed method to existing approaches. Experimental and simulated results are shown in Figures 15, 19, 20, 21, 22, 23, 24, 25 and 26 and Figures 17, 27, 28, 29, 30, 31, 32, 33 and 34, respectively. These additional reconstructions illustrate how detailed the depth map of our approach is compared to other reference methods. Our proposed method is able to visualize all particular extremities of a pedestrian (see Figure 15). Moreover, even small objects at large distances are visible while other methods such as lidar interpolation simply wipes them out (see Figures 21, 22 and 25). The day time results in Figures 24, 25 and 26 indicates that this method also works during day time without any significant drop of performance due to ambient light present.

Figures 16 and 18 visualize the depth-dependent MAE. Specifically, this evaluation validated that how our method outperforms all monocular and stereo depth estimation methods by a large margin. Only the method that uses ground truth lidar points as input [10] is able to achieve comparable performance.

References

[1] Amit Adam, Christoph Dann, Omer Yair, Shai Mazor, and Sebastian Nowozin. Bayesian time-of-flight for realtime shape, illumination and albedo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(5):851–864, 2017. 7, 8, 13, 14, 15, 16, 17, 18,

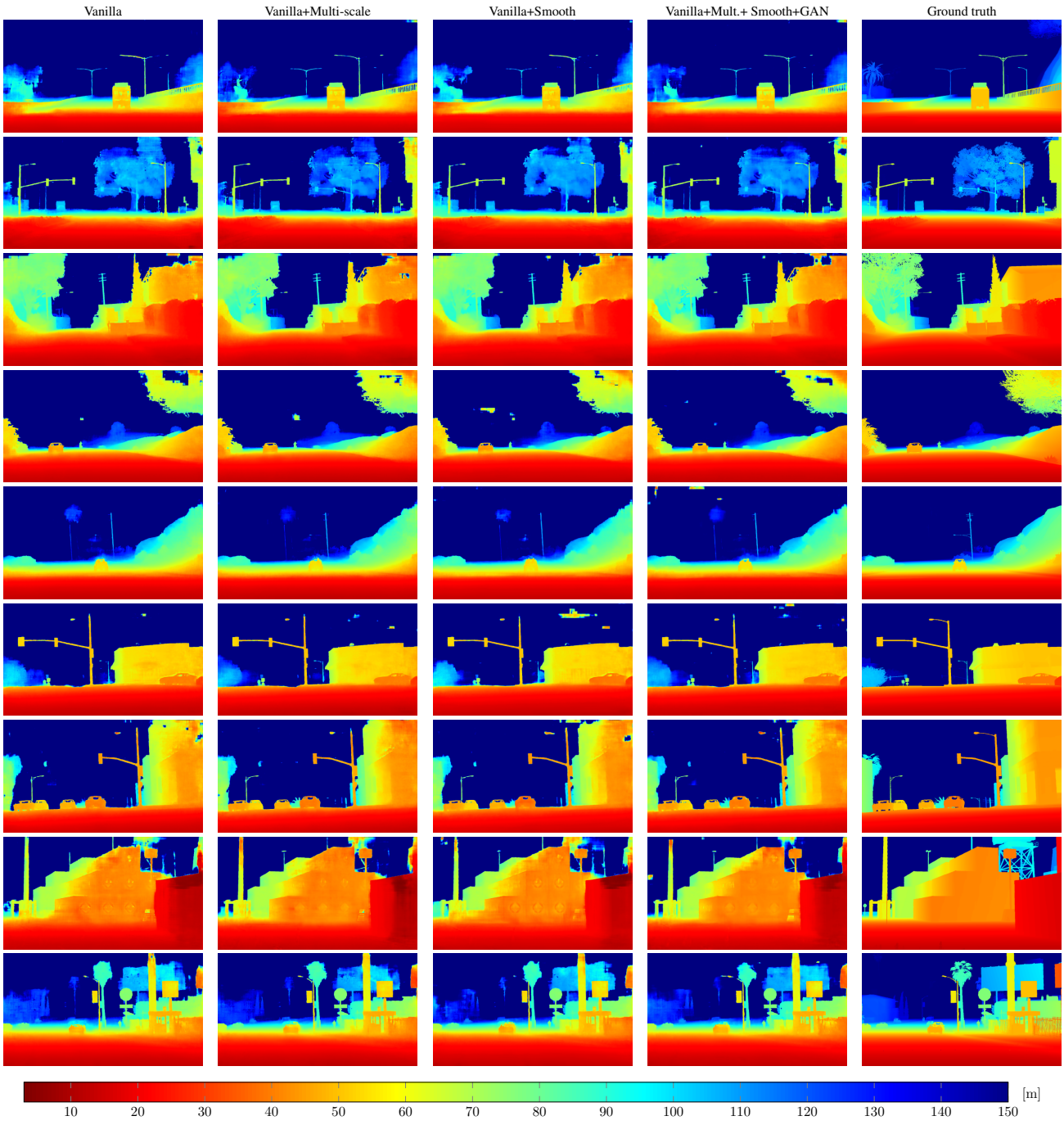


Figure 13: Synthetic qualitative results. Each row shows results for a specific example. The first four columns show the outputs for our evaluated networks, the fifth column shows the ground-truth.

19, 20, 21

[2] Jia-Ren Chang and Yong-Sheng Chen. Pyramid stereo matching network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5410–5418, 2018. 10, 13, 14, 15, 16, 17, 18, 19, 20, 21

[3] Alessandro Foi, Mejdi Trimeche, Vladimir Katkovnik, and Karen Egiazarian. Practical poissonian-gaussian noise modeling and fitting for single-image raw-data. *IEEE Transactions on Image Processing*, 17(10):1737–1754, 2008. 4

[4] Adrien Gaidon, Qiao Wang, Yohann Cabon, and Eleonora Vig. Virtual worlds as proxy for multi-object tracking analysis. In

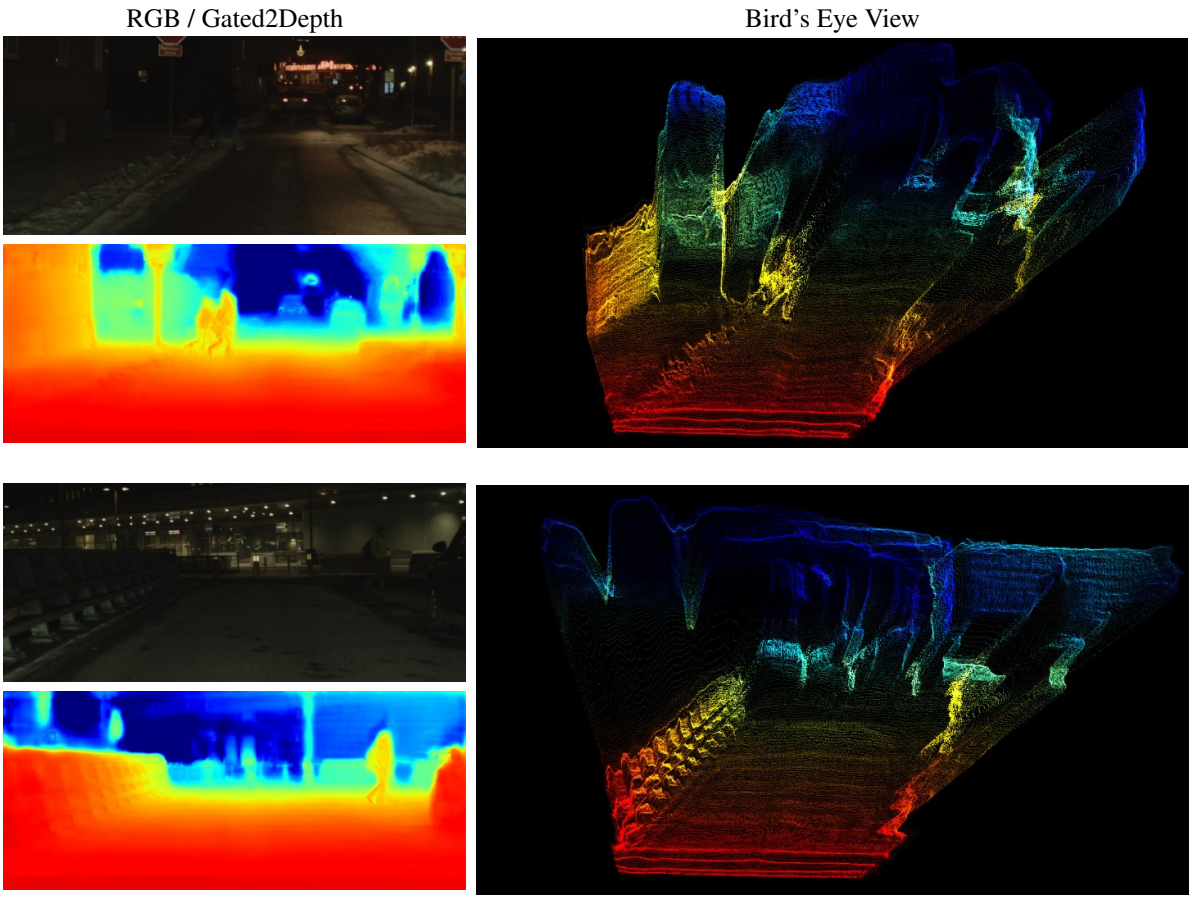


Figure 14: Additional 3D point cloud visualization. Color coding is the same as in Figure 15.

Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016. 1

[5] Clément Godard, Oisin Mac Aodha, and Gabriel J Brostow. Unsupervised monocular depth estimation with left-right consistency. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 10, 13, 14, 15, 16, 17, 18, 19, 20, 21

[6] Felix Heide, Wolfgang Heidrich, Matthias Hullin, and Gordon Wetzstein. Doppler time-of-flight imaging. *ACM Transactions on Graphics (ToG)*, 34(4):36, 2015. 7

[7] Heiko Hirschmuller. Stereo processing by semiglobal matching and mutual information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(2):328–341, Feb 2008. 10, 13, 15, 16, 17, 18

[8] Matthew Johnson-Roberson, Charles Barto, Rounak Mehta, Sharath Nittur Sridhar, Karl Rosaen, and Ram Vasudevan. Driving in the matrix: Can virtual worlds replace human-generated annotations for real world tasks? In *IEEE International Conference on Robotics and Automation*, pages 1–8, 2017. 1

[9] Robert Lange. 3D time-of-flight distance measurement with custom solid-state image sensors in CMOS/CCD-technology. 2000. 7

[10] Fangchang Ma and Sertac Karaman. Sparse-to-dense: Depth prediction from sparse depth samples and a single image. In *IEEE International Conference on Robotics and Automation*, pages 1–8, 2018. 10, 13, 14, 15, 16, 17, 18, 19, 20, 21

[11] Jorge J. Moré, Danny C. Sorensen, Kenneth E. Hillstrom, , and Burton S. Garbow. The MINPACK project. *Sources and Development of Mathematical Software*, 1984. 8

[12] Stephan R. Richter, Zeeshan Hayder, and Vladlen Koltun. Playing for benchmarks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2232–2241, 2017. 1

[13] Stephan R. Richter, Vibhav Vineet, Stefan Roth, and Vladlen Koltun. Playing for data: Ground truth from computer games. In *Proceedings of the IEEE European Conf. on Computer Vision*, volume 9906 of *LNCS*, pages 102–118. Springer International Publishing, 2016. 1

[14] German Ros, Laura Sellart, Joanna Materzynska, David Vazquez, and Antonio M. Lopez. The SYNTHIA dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Jun 2016. 1

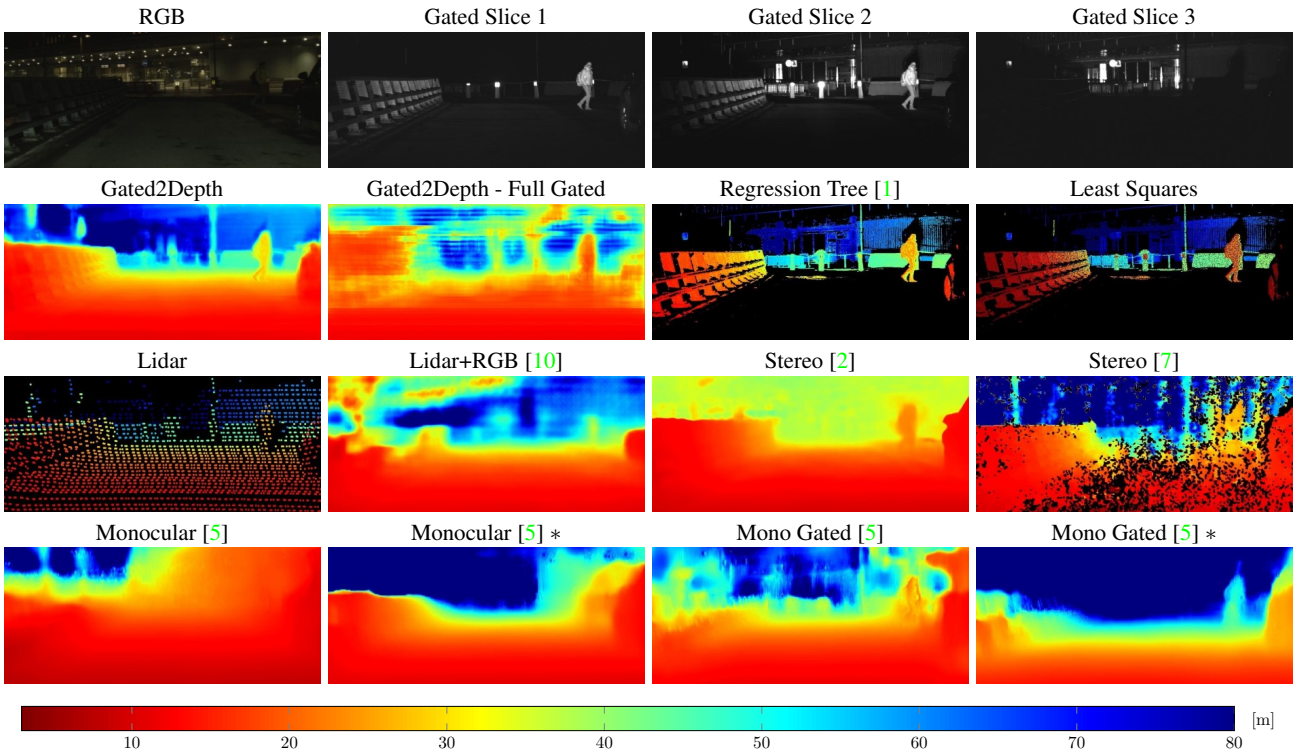


Figure 15: Qualitative results for our method and reference methods over real night time examples. For each example, we include the corresponding RGB and all three gated slices, along with the lidar measurements. Our method generates more accurate and detailed maps over the full depth range in comparison to the other methods.

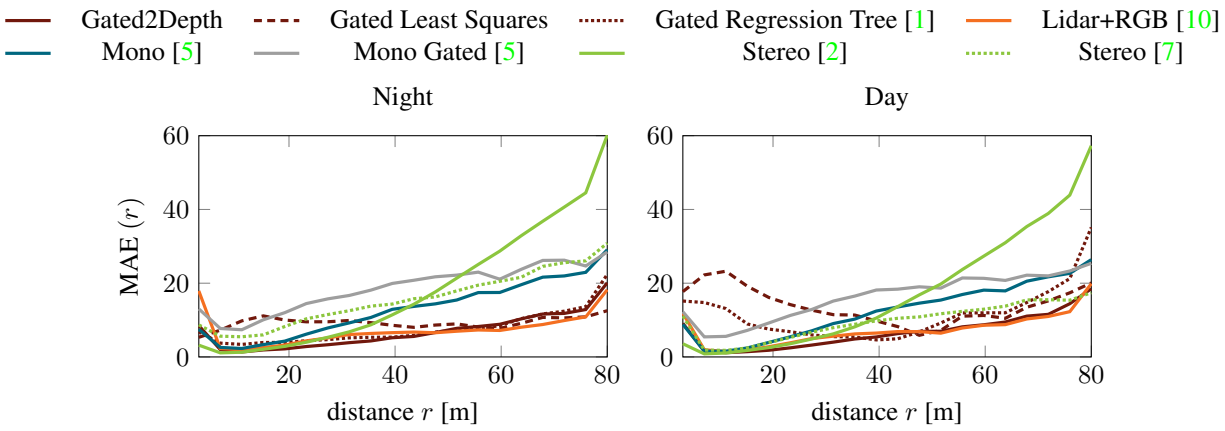


Figure 16: Depth-dependent accuracy graphs for night and day (Real).

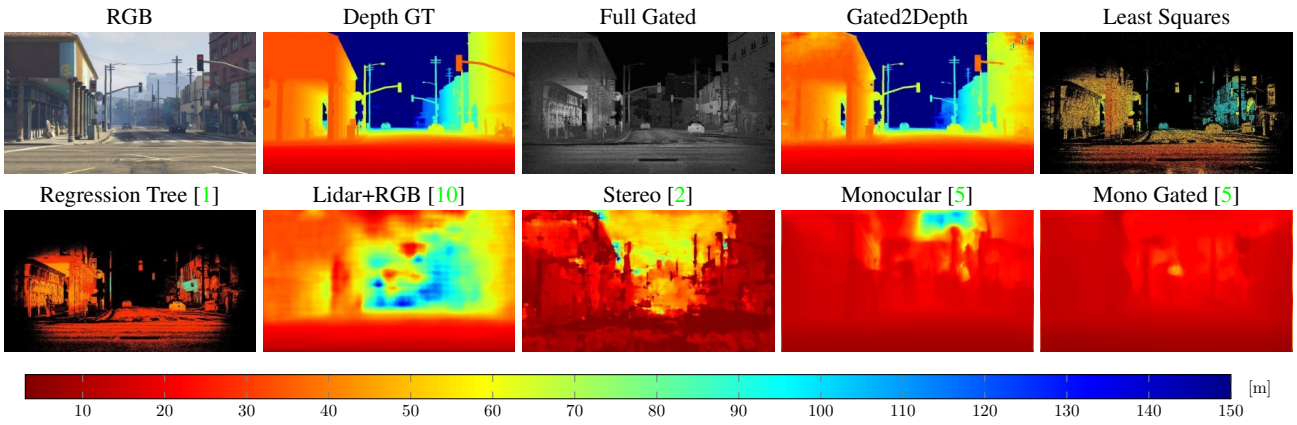


Figure 17: Qualitative results for our method and reference methods over simulated night time examples. For each example, we include the corresponding RGB and full gated image, along with the Depth GT and lidar sampling for training [10]. Our method generates more accurate and detailed maps over the full depth range in comparison to the other methods.

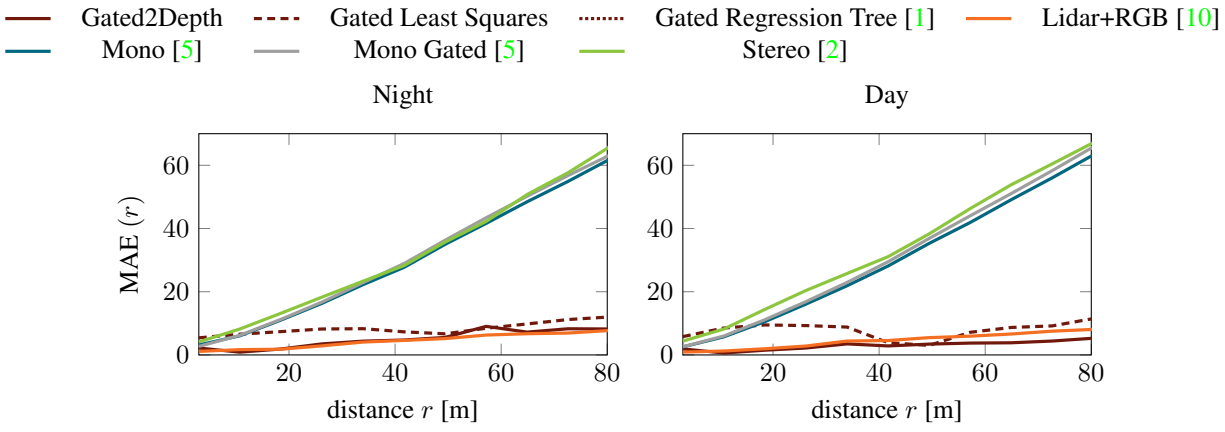


Figure 18: Depth-dependent accuracy graphs for night and day (Synthetic).

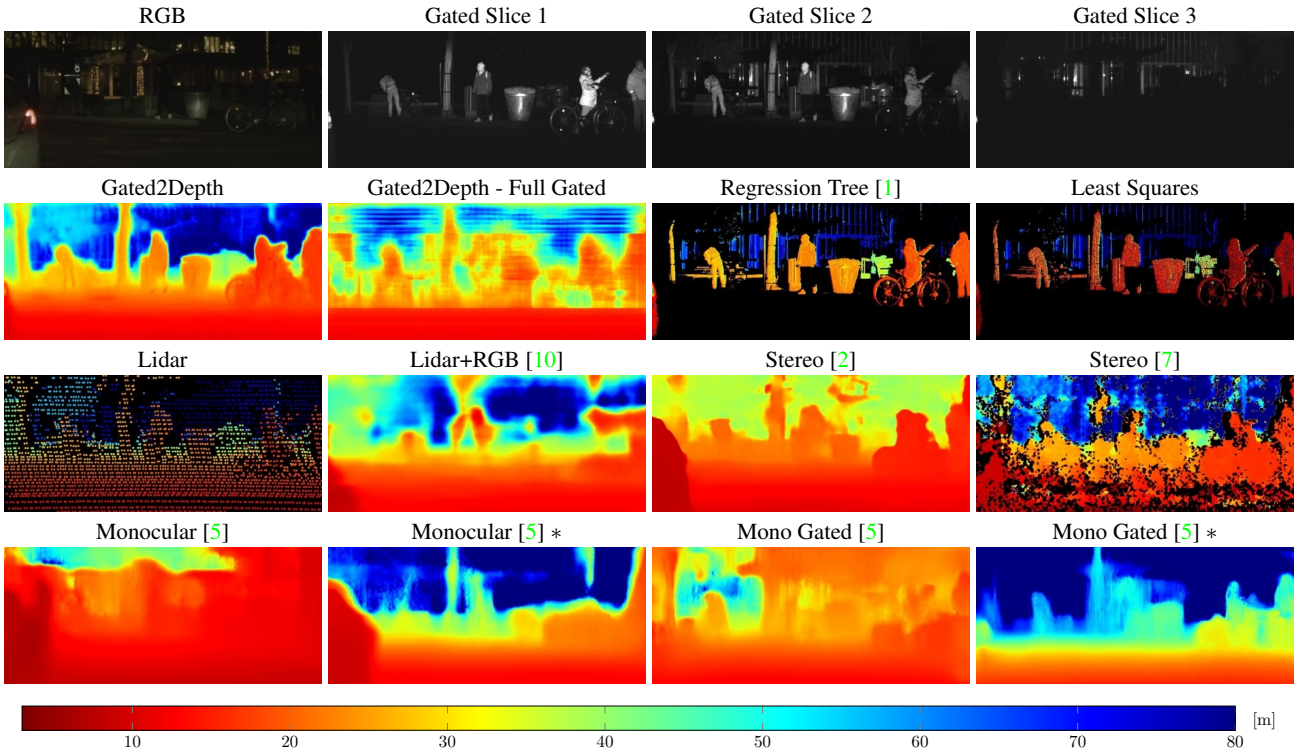


Figure 19: Additional result with same comparisons as in Figure 15.

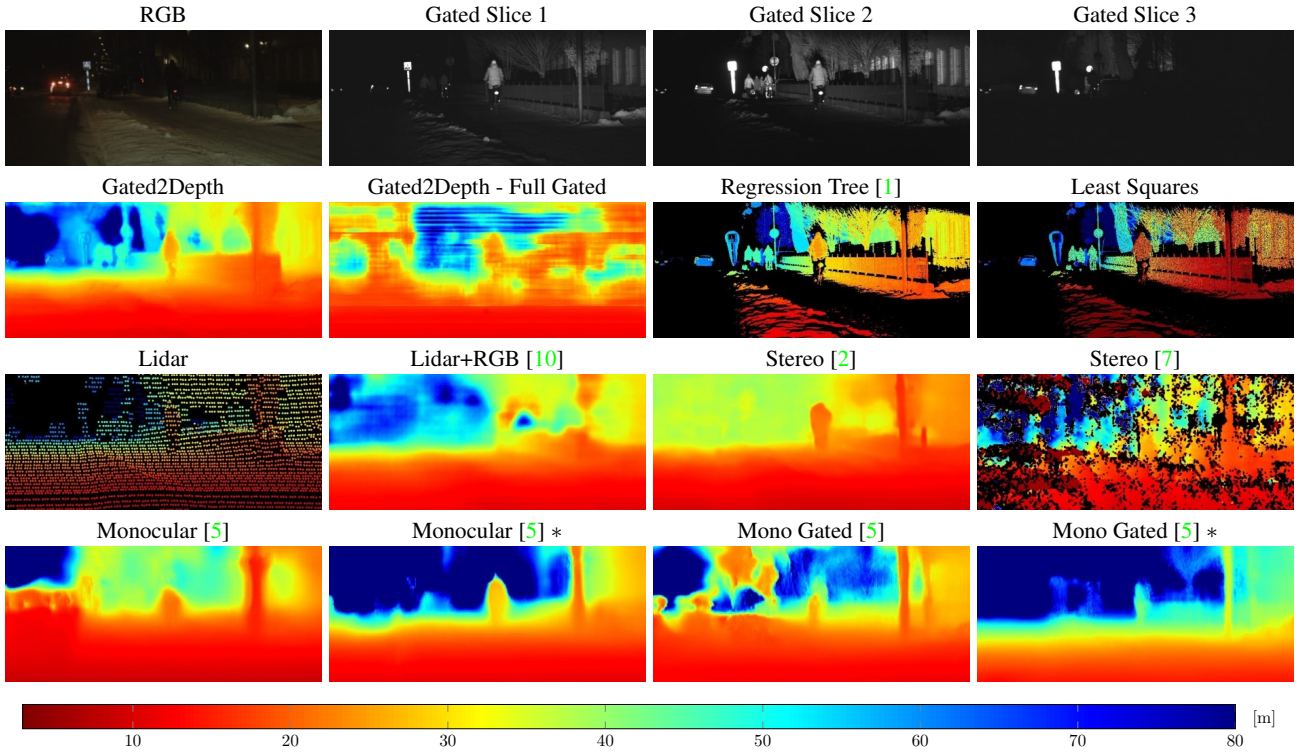


Figure 20: Additional result with same comparisons as in Figure 15.

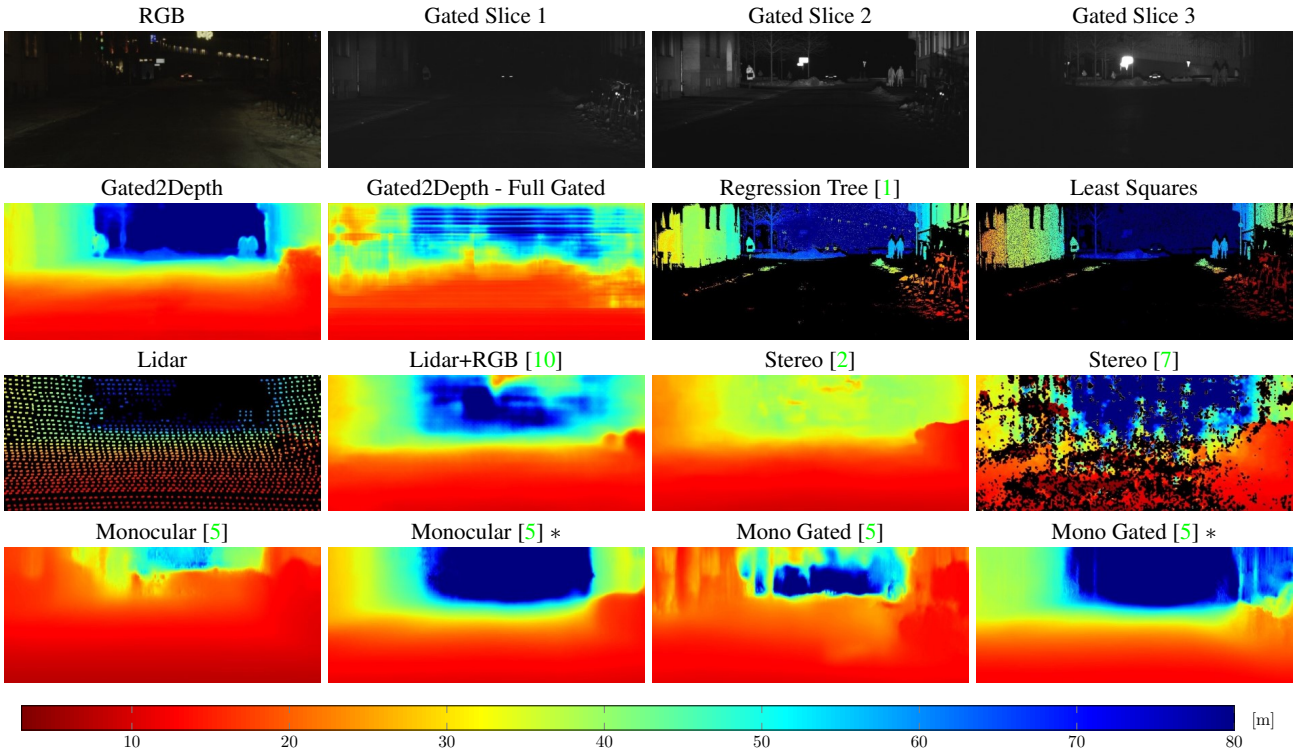


Figure 21: Additional result with same comparisons as in Figure 15.

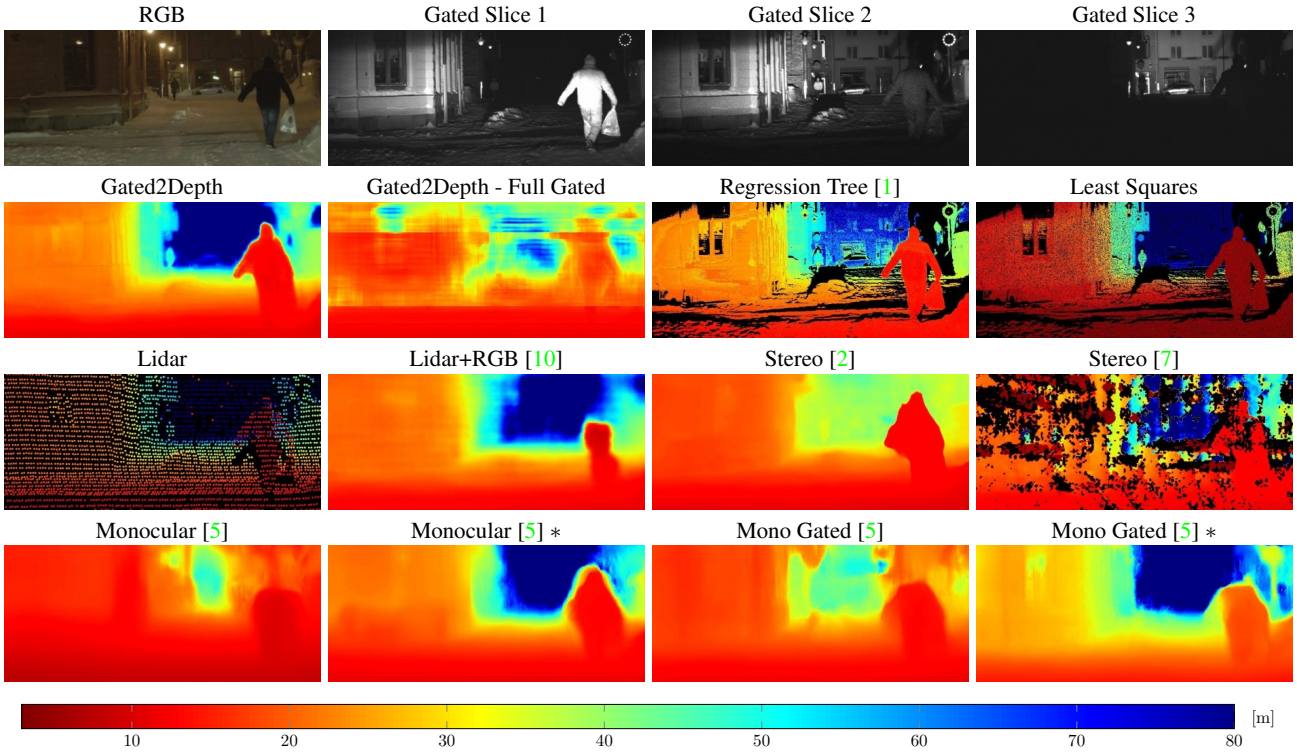


Figure 22: Additional result with same comparisons as in Figure 15.

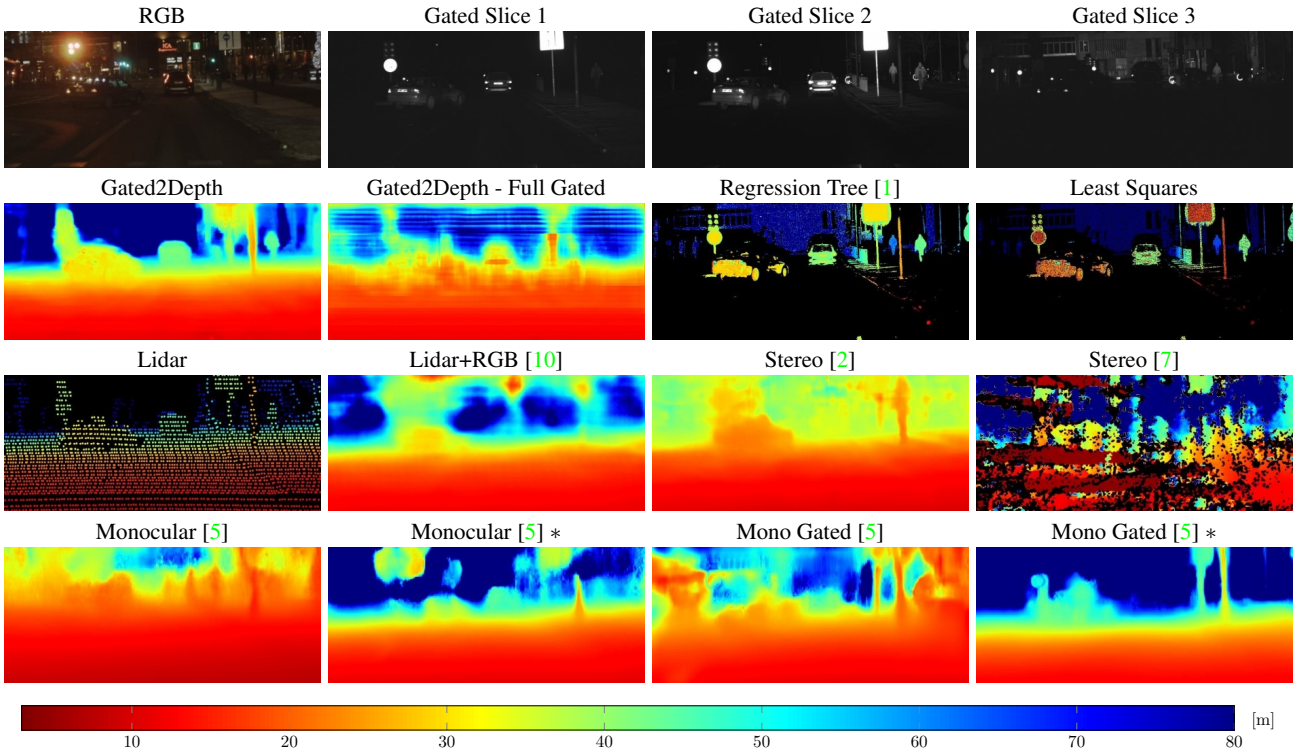


Figure 23: Additional result with same comparisons as in Figure 15.

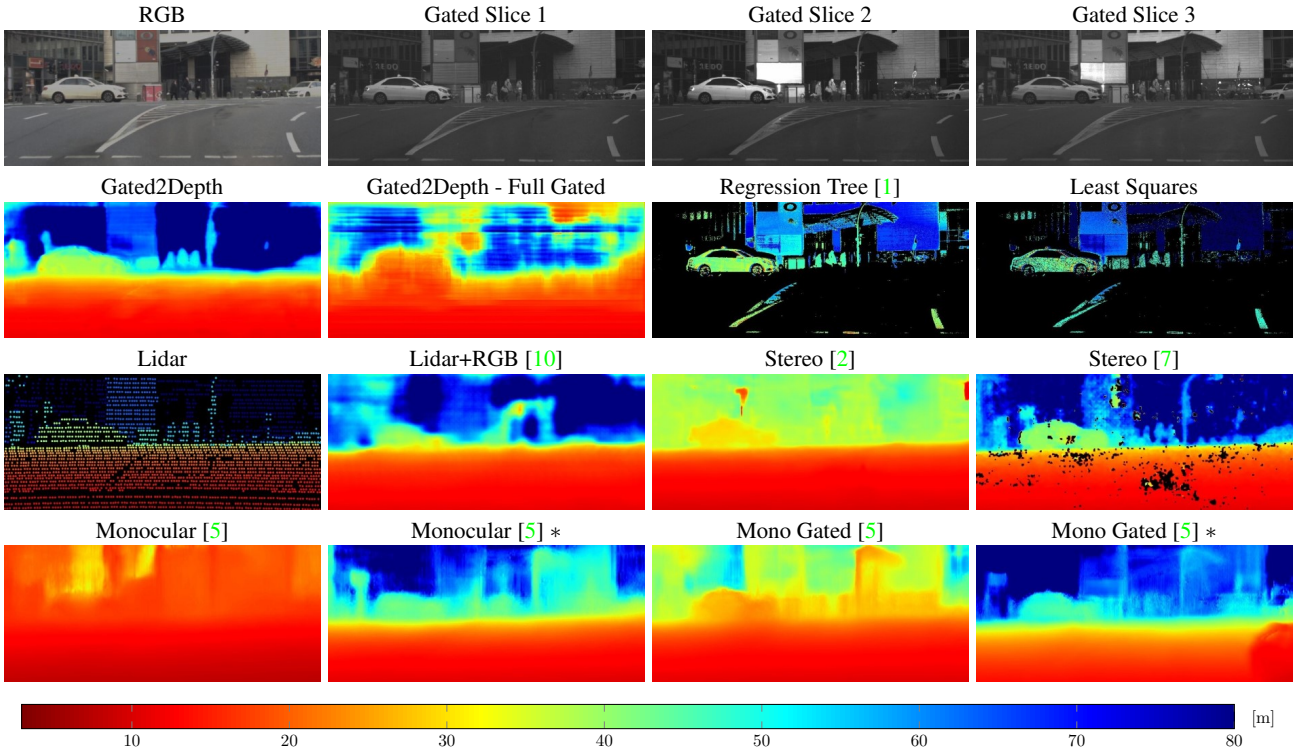


Figure 24: Additional result with same comparisons as in Figure 15.

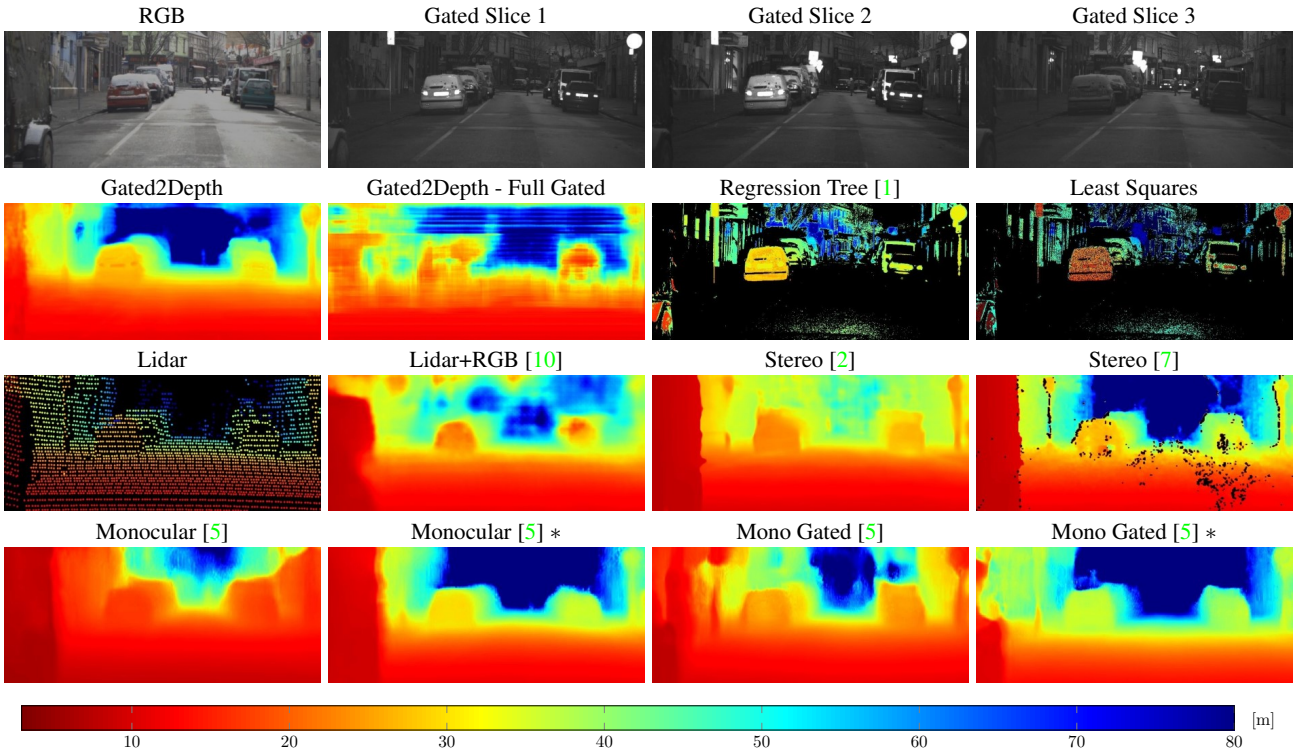


Figure 25: Additional result with same comparisons as in Figure 15.

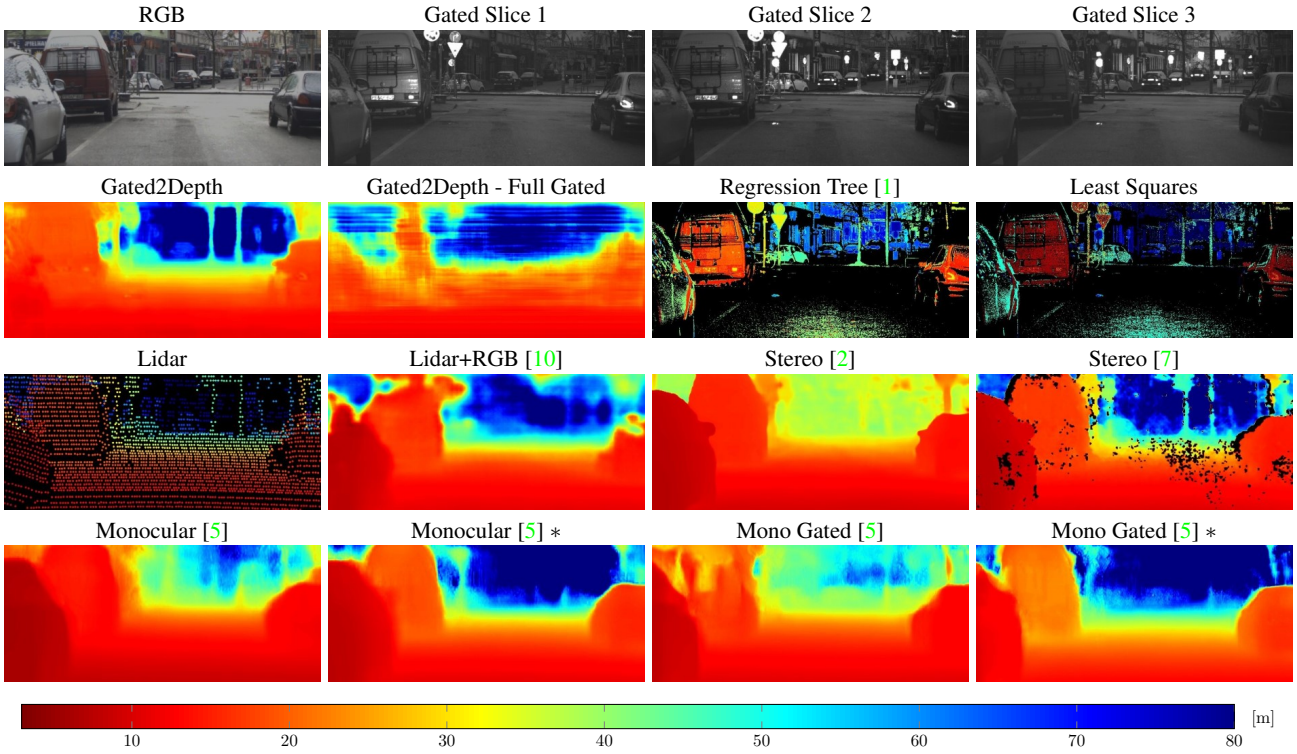


Figure 26: Additional result with same comparisons as in Figure 15.

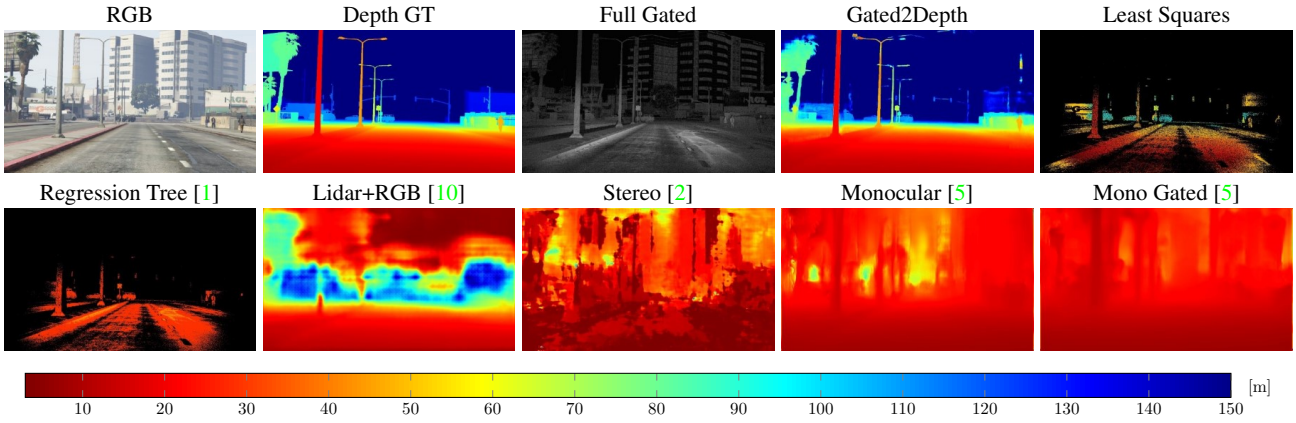


Figure 27: Additional simulated result with same comparisons as in Figure 17.

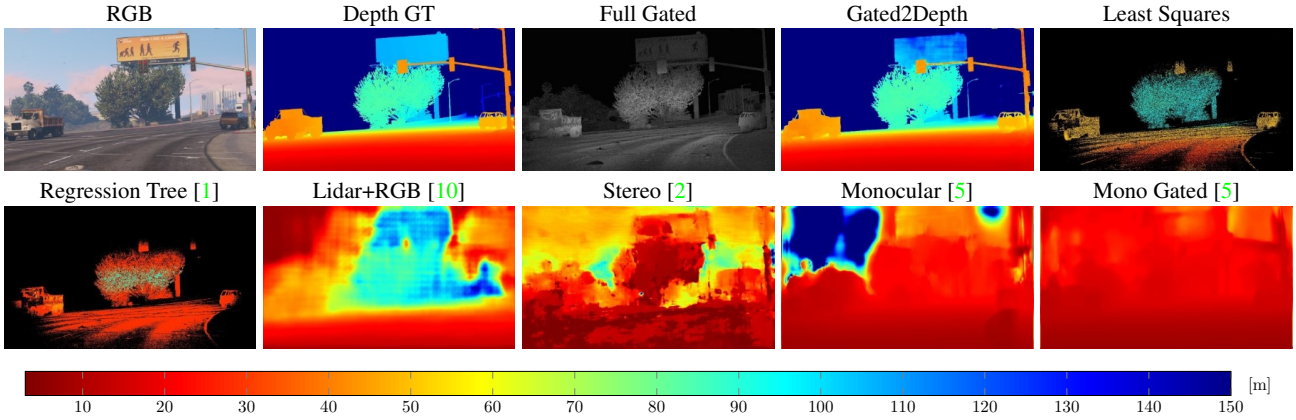


Figure 28: Additional simulated result with same comparisons as in Figure 17.

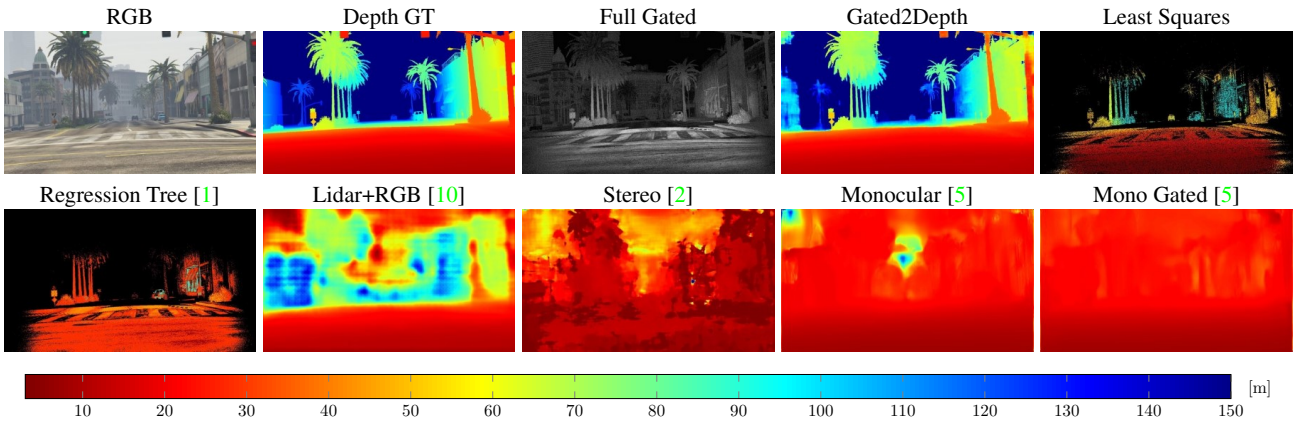


Figure 29: Additional simulated result with same comparisons as in Figure 17.

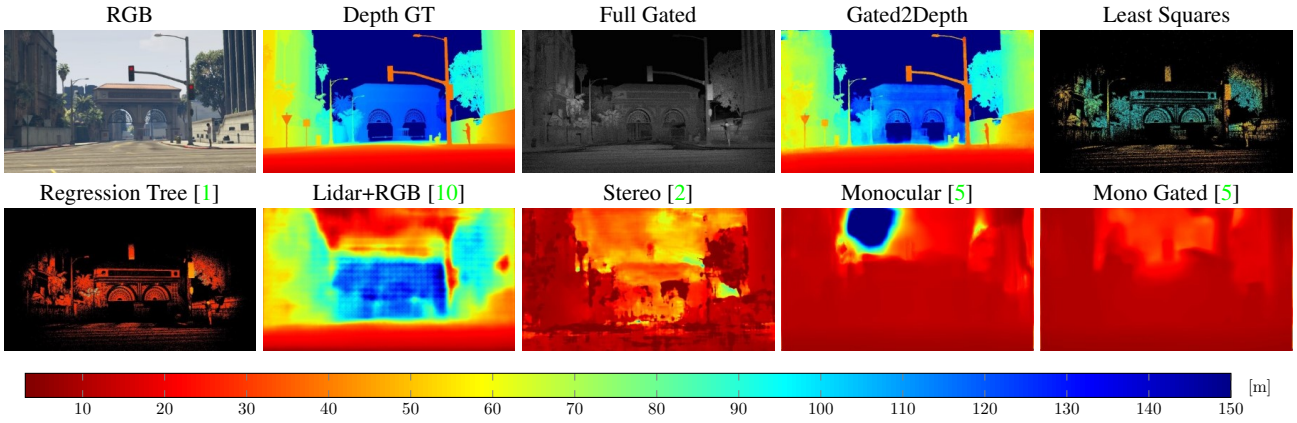


Figure 30: Additional simulated result with same comparisons as in Figure 17.

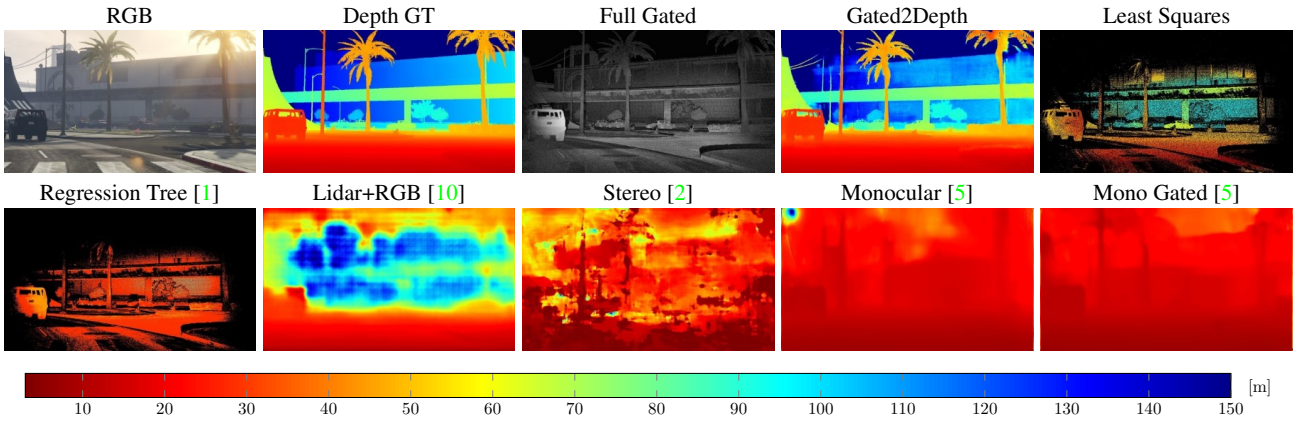


Figure 31: Additional simulated result with same comparisons as in Figure 17.

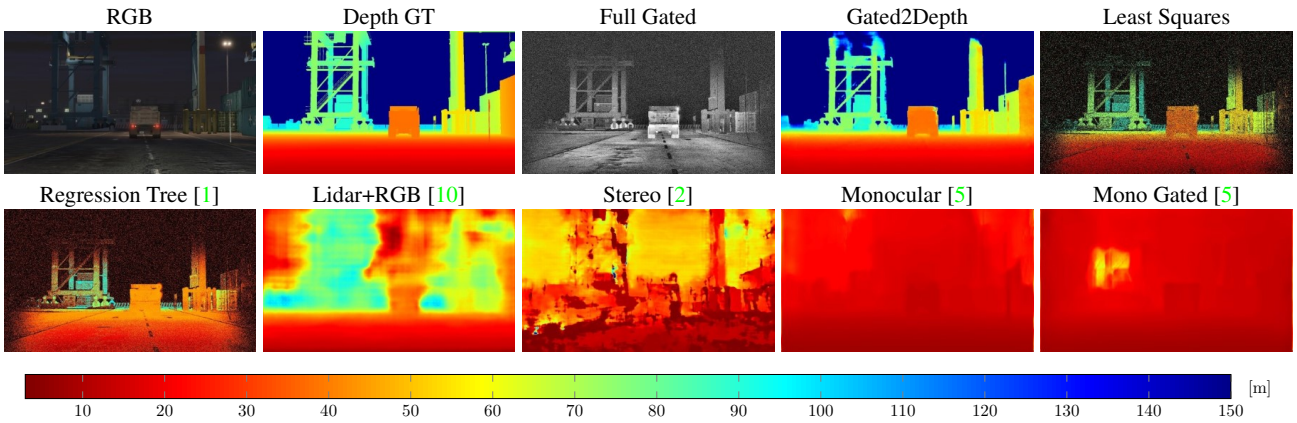


Figure 32: Additional simulated result with same comparisons as in Figure 17.

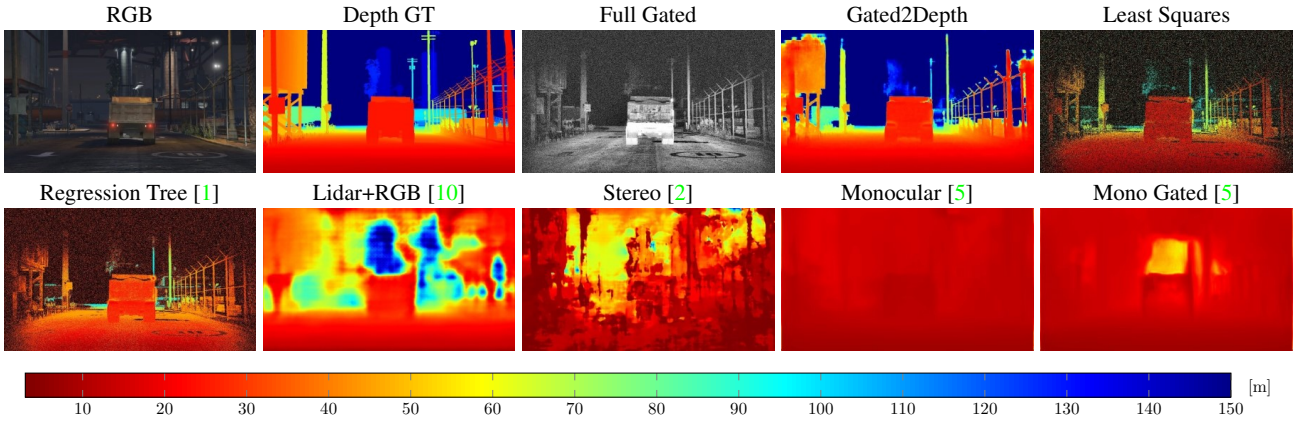


Figure 33: Additional simulated result with same comparisons as in Figure 17.

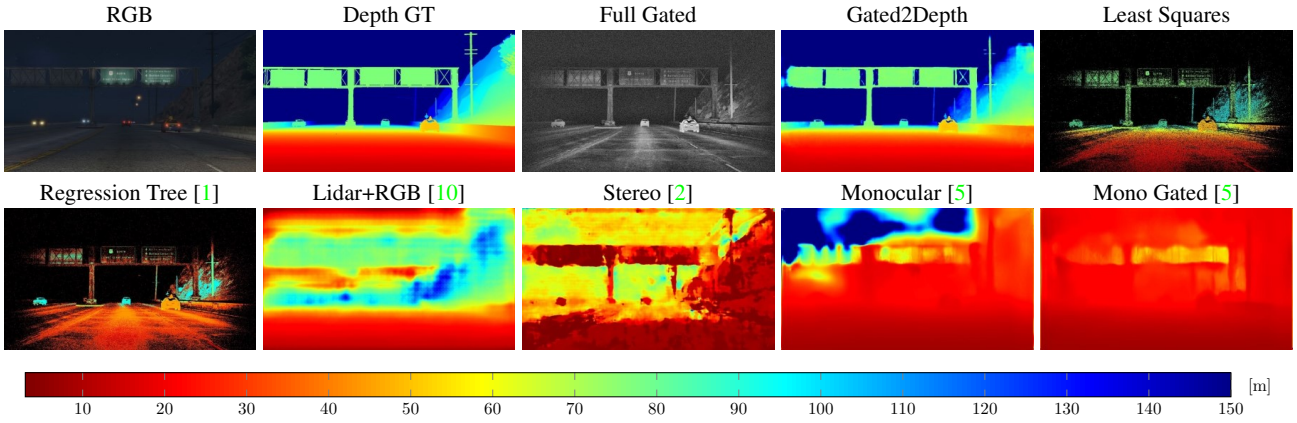


Figure 34: Additional simulated result with same comparisons as in Figure 17.