Too Tiny to See: Hazardous Obstacle Detection Dataset and Evaluation (Supplementary Information)

Topi Miekkala[1] Samuel Brucker[2] Stefanie Walz[2] Filippo Ghilotti[2]
Andrea Ramazzina[3] Dominik Scheuble[3] Pasi Pyykönen[1] Mario Bijelic[2,4] Felix Heide[2,4]
[1]VTT Technical Research Centre of Finland [2]Torc Robotics [3]Mercedes-Benz [4]Princeton University
https://light.princeton.edu/2T2S

This supplemental document provides additional details to support the findings in the main manuscript. Section 1 describes the proposed lost cargo dataset and explains the automatic object placement process. Section 2 outlines the implementation of the 2T2S metric and the reference metrics. Section 3 presents the implementation and fine-tuning of the depth estimation methods. Section 4 includes additional ablation studies for the 2T2S implementation, and Section 5 provides further quantitative and qualitative results.

Contents

1. Lost Cargo Dataset

We use a single experimental setup with two sensor configurations mounted on different vehicles. All sequences are calibrated, ensuring consistent data across setups. The first is a portable sensor rig mounted on the roof of a test vehicle, integrating four modalities: an automotive RGB stereo camera (OnSemi AR0230), a gated stereo system (BrightwayVision BrightEye), an automotive RCCB stereo camera (OnSemi AR0820AT), and a reference LiDAR sensor (Velodyne VLS128). The gated system consists of two cameras and an active NIR illumination source. One camera is housed in the main sensor box alongside the RGB stereo pair, while the second gated camera is placed in a satellite box to provide a 0.75 m baseline. The system operates at 120 Hz, capturing near-infrared images (808 nm, 10-bit, 1280×720). Each cycle generates three active slices and two passive images with low and high exposure times, resulting in an effective 24 Hz repetition rate. Active illumination is supplied by two VCSEL modules mounted at the front of the vehicle, triggered by the left gated camera. The RGB stereo pair has a 0.23 m baseline and records 1920×1024 images at 30 Hz with 12-bit quantization. The RCCB stereo system uses OnSemi AR0820AT sensors, capturing 16-bit HDR imagery at 15 Hz with a resolution of 3848×2168. The cameras are mounted with a 0.75 m baseline, distributed between the main and satellite sensor boxes. The stereo systems are synchronized and calibrated, ensuring aligned epipolar geometry for disparity estimation. Ground-truth depth is provided by a Velodyne VLS128 LiDAR, operating at 905 nm and 10 Hz, with a vertical field of view of 40° across 128 scanning lines. Its

range exceeds 200 m for reflective targets. All sensors are recorded in a unified ROS framework with extended synchronization to align gated slices, stereo frames, RCCB images, and LiDAR sweeps, ensuring consistent temporal alignment across modalities for training and evaluation.

The second sensor setup is a rig mounted on another test vehicle, integrating two LiDAR sensors (Luminar H3, Livox HAP). The Luminar H3 operates at 1550 nm and 10 Hz, with a vertical field of view of 30° across 64 scan lines and a detection range greater than 200 m for reflective targets. Notably, the Luminar H3 implements modifiable scan patterns allowing adjustments to the scan line density vertically along the image. For our data collection, we set an optimal Gaussian shape scan pattern, with the focus point μ = -2.0 and σ = 3.5. The Livox HAP operates at 905 nm and 10 Hz, with a vertical field of view of 25° and detection range of 200 m for reflective targets. In our measurements the Livox HAP operates with a non-repeating scan pattern with a higher density focus in the middle ROI.

The dataset also includes ground-truth meshes extracted from each lost-cargo item as described in the main paper. The meshes for the winter dataset objects 'Biker', 'Bumper' and 'Exhaust' were created with the Livox approach. All other objects 'Stacked pallets', 'Small tire' and 'Large tire' were scanned with the polycam app. The summer dataset objects 'Stacked pallets', 'Bumper', 'Large tire', 'Muffler' and 'Small tire' were all scanned with the polycam app. For the mesh approximation parameters shown in Eqs. 1-12 of the main paper, k was set to 20 and τ was set to 0.8. r was set to $[0.02, 0.06, 0.08, 0.10]$ for BPA. Voxel size v was set to 0.1 m when generating $\mathcal{G}(x, y, z)$. λ_c was set to 1.0, λ_e was set to 1.0, λ_l was set to 0.1, and λ_n was set to 0.01. The evaluation crop distance d_e was set to 1.0 m.

The dataset consists of two recording campaigns. The summer recordings were collected in suburban North America under clear daytime conditions, with objects placed at 15, 30, 45, 60, 75, and 90 meters from the camera. The winter recordings were collected in rural northern Europe on a snow-covered highway, under both day and night conditions, with objects placed at 25, 50, 75, and 100 meters. Both campaigns include captures preformed at day and nighttime. Each sequence contains captures at multiple distances under identical environmental and background conditions. Distances of 15 m and 100 m are excluded from evaluation because objects fall outside the effective field of view of individual sensors—either beyond the physical operating range of lidars in winter or outside the vertical field of view of long-range cameras (30° opening angle). In total, the dataset comprises 14 sequences. With lost placement variations and unique lost-cargo placements a combination of 90 test examples can be found. Figures 3 to 6 illustrate sensor outputs for the closest object placements (25 m in winter, 30 m in summer), ordered by sequence number.

1.1. Object Mesh Alignment and Calibration

To ensure consistent evaluation across distances and sensor modalities, lost-cargo objects must be placed at fixed positions within each sequence. Since the dataset was recorded in real environments, direct placement of meshes is not possible; instead, object meshes must be aligned to point clouds to achieve accurate positioning. An overview of the automated placement procedure is shown in Fig. 1.

The procedure consists of two stages. Stage 1 is executed once per sequence and determines the reference placement of objects within the scene layout. Stage 2 replicates this placement across all defined distances by positioning the meshes at the corresponding depth locations.

Each dataset sample includes a configuration file that specifies: (i) paths to the generated object meshes, (ii) object rotations, and (iii) the region-of-interest (ROI) given by the minimum and maximum depth where objects are expected. Object rotations are manually determined by aligning the meshes to the closest available point cloud, ensuring realistic orientation relative to the scene.

1.1.1 Stage 1: Object placement within the layout.

Stage 1 is executed once for each lost-cargo sequence to automatically determine the placement of objects within the respective layout. In the following we are presenting each consecutive step per paragraph.

ROI cropping and ground removal. Given a point cloud $\mathcal{P} = \{\mathbf{p}_i \in \mathbb{R}^3 \mid i = 1, \ldots, M\}$, where M is the number of points, we first restrict the set to a region-of-interest (ROI) defined by a minimum and maximum depth $(d_{\min}, d_{\max})$. The cropped point cloud is

$$\mathcal{P}_{\text{ROI}} = \{\mathbf{p}_i \in \mathcal{P} \mid d_{\min} \leq \|\mathbf{p}_i\|_2 \leq d_{\max}\}. \tag{1}$$

Next, a morphological ground filter is applied to remove points belonging to the road plane, resulting in the filtered cloud $\mathcal{P}_{\text{filt}}$.

Clustering. We apply DBSCAN clustering to $\mathcal{P}_{\text{filt}}$ to obtain a set of clusters $\mathcal{C} = \{C_1, \ldots, C_K\}$, where each $C_k \subseteq \mathcal{P}_{\text{filt}}$ represents a coherent group of points.

Ground-truth mesh sampling. For each object mesh $\mathcal{M}$, the encoding is loaded and rotated according to the configuration file. A set of N = 10000 points is sampled from the mesh surface to form a ground-truth point cloud

$$\mathcal{P}_{\mathcal{M}} = \{\mathbf{q}_j \in \mathbb{R}^3 \mid j = 1, \ldots, N\}. \tag{2}$$

Using the computation steps of 2T2S metric, both $\mathcal{P}_{\mathcal{M}}$ and the cluster point sets $\{C_k\}$ are encoded into feature vectors $f(\cdot)$, yielding $\mathbf{f}_{\mathcal{M}} = f(\mathcal{P}_{\mathcal{M}})$ and $\mathbf{f}_{C_k} = f(C_k)$.

Matching. Hungarian matching is performed between ground-truth encodings and cluster encodings by minimizing a pairwise cost matrix

$$\mathcal{L}(i, k) = \|\mathbf{f}_{\mathcal{M}_i} - \mathbf{f}_{C_k}\|_2, \tag{3}$$

ensuring each mesh sample $\mathcal{P}_{\mathcal{M}_i}$ is matched to the most similar cluster C_k.

Mesh alignment. Finally, the mesh center is aligned to the center of the matched cluster. If $\boldsymbol{\mu}(C_k)$ denotes the centroid of cluster C_k, then the translation Δ applied to the mesh is

$$\Delta = \boldsymbol{\mu}(C_k) - \boldsymbol{\mu}(\mathcal{P}_{\mathcal{M}_i}), \tag{4}$$

where $\boldsymbol{\mu}(\cdot)$ computes the mean 3D position of a point set. The mesh is then shifted by Δ to achieve proper placement.

1.1.2 Stage 2: Alignment across distances.

Stage 2 aligns the reference point cloud $\mathcal{P}_1$ from the closest distance (used in Stage 1) with the point clouds $\mathcal{P}_2$ captured at larger distances, in order to transfer object placements consistently.

Initial alignment. An initial alignment is performed using the region-of-interest (ROI) and measurement distance specified in configuration files. Let $(\mathbf{d}^{(1)}_{\min}, \mathbf{d}^{(1)}_{\max})$ and $(\mathbf{d}^{(2)}_{\min}, \mathbf{d}^{(2)}_{\max})$ denote the ROI coordinates of $\mathcal{P}_1$ and $\mathcal{P}_2$, respectively. The alignment ensures that these ranges overlap such that the expected object placement regions are consistent. $\mathcal{P}_1$ and $\mathcal{P}_2$ are cropped:

$$\tilde{\mathcal{P}}_1 = \{\mathbf{p} \in \mathcal{P}_1 \mid \mathbf{d}^{(1)}_{\min} \leq \mathbf{p} \leq \mathbf{d}^{(1)}_{\max}\}. \tag{5}$$

$$\tilde{\mathcal{P}}_2 = \{\mathbf{p} \in \mathcal{P}_2 \mid \mathbf{d}^{(2)}_{\min} \leq \mathbf{p} \leq \mathbf{d}^{(2)}_{\max}\}. \tag{6}$$

A rigid transformation (R, t) is estimated by Iterative Closest Point (ICP) to align $\tilde{\mathcal{P}}_2$ with $\tilde{\mathcal{P}}_1$:

$$(R^*, t^*) = \arg\min_{R,t} \sum_{\mathbf{p} \in \tilde{\mathcal{P}}_1} \min_{\mathbf{q} \in \tilde{\mathcal{P}}_2} \|R\mathbf{p} + t - \mathbf{q}\|_2^2, \tag{7}$$

where $R \in SO(3)$ and $t \in \mathbb{R}^3$. The optimized transformation (R^*, t^*) is applied to $\mathcal{P}_2$ which yields the aligned full point cloud $\mathcal{P}'_2 = \{R^*\mathbf{p} + t^* \mid \mathbf{p} \in \mathcal{P}_2\}$.

Mesh transfer and pose refinement. Since object meshes have already been aligned in $\mathcal{P}_1$ during Stage 1, their positions are now also aligned in $\mathcal{P}'_2$. If $\mathbf{m}$ denotes a mesh center in $\mathcal{P}_1$, its aligned position in $\mathcal{P}'_2$ is

$$\mathbf{m}_{\mathcal{P}_2} = \mathbf{m}. \tag{8}$$

Next, $\mathcal{P}'_2$ is cropped according to the ROI defined in Stage 1:

$$\tilde{\mathcal{P}}'_2 = \{\mathbf{p} \in \mathcal{P}'_2 \mid d^{(1)}_{\min} \leq \|\mathbf{p}\|_2 \leq d^{(1)}_{\max}\}. \tag{9}$$

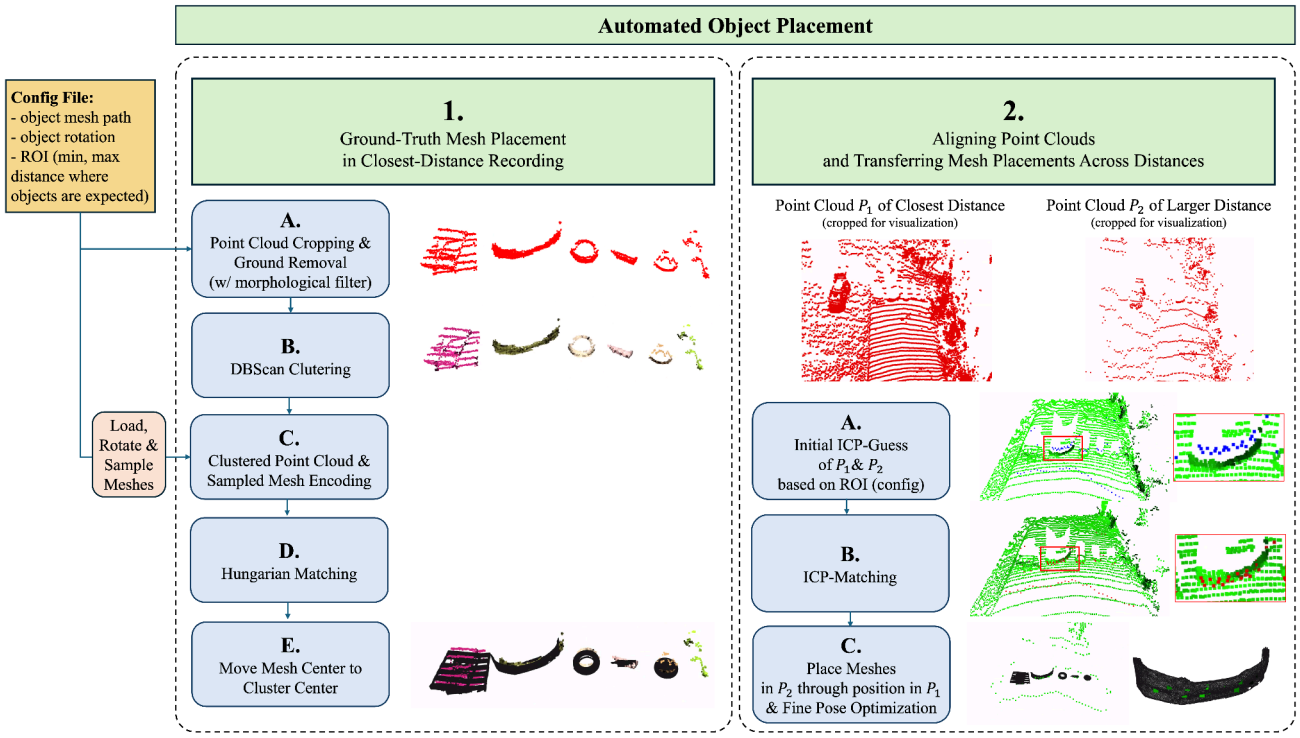

Figure 1. **Automated Object Placement.** Overview of the two-stage process: stage one places the ground-truth mesh in the recording with the closest distance, and stage two applies this placement to all other distances and sensors.

Then for each object mesh placed in $\tilde{\mathcal{P}}_2'$, we apply the fine pose optimization operations as presented in Section 4 of the main paper by Eqs. (15)–(22). The optimizer is implemented in PyTorch3D [9] and minimizes the loss function in Eq. (15) of the main paper to find and optimal transform $\mathcal{T}_\text{o}$ shown in Eq. (21). The learning rate is initially set to 0.07, and the optimization is guided by learning rate scheduling, which halves the learning rate when the loss does not decrease for five consecutive epochs. The optimization is considered complete when the scheduler has dropped the learning rate to less than 0.001 of the original rate. After applying $\mathcal{T}_\text{o}$ to obtain $\mathcal{M}_\text{e}$ from Eq. 22 of the paper, we crop $\tilde{\mathcal{P}}_2'$ to obtain the evaluation point cloud $\mathcal{P}_e$ of the paper:

$$\mathcal{P}_\text{e} \;=\; \left\{\, \mathbf{p} \in \tilde{\mathcal{P}}_2' \;\middle|\; d_k(\mathbf{p};\, \mathcal{M}_e) \le \tau \,\right\}, \tag{10}$$

$$d_k(\mathbf{p};\, \mathcal{M}_e) \;=\; \text{kNN-dist}\Big(\mathbf{p},\, \mathcal{M}_e,\, k\Big), \tag{11}$$

where k is equivalent to d_e in the paper, and is set to 1 m.

2. Metrics Implementation Details

For point cloud operations and metric computations, we mainly utilize the Open3D [13], PyTorch3D [9] and Pointcept [4] libraries. Instead of applying metrics globally to full point clouds, all metrics are applied to $\mathcal{P}_0$ consisting only of ground truth object points, and to the extracted crop $\mathcal{P}_e$ consisting only of object points detected by the depth perception method.

Depth-Based Metrics. To evaluate depth-based metrics MAE, RMSE, RMSE Log, Silog and Abs Rel, the evaluation cloud $\mathcal{P}_\text{e}$ and sampled ground truth cloud $\mathcal{P}_0$ are first normalized to coordinate range $[-1, 1]$ and rendered into depth images

$$I \in \mathbb{R}^{H \times W}$$

where H = 512 and W = 512. We render using PyTorch3D library, utilizing the alpha compositing for point-based rendering [14]. The depth metric calculations include only pixels which have non-zero values in both images. We define the set of non-zero pixels as

$$\Omega(I) = \{\, (i,j) \mid I(i,j) \neq 0,\; 1 \le i \le H,\; 1 \le j \le W \,\}.$$

Depth metrics are then computed only over

$$\Omega(I_1) \cap \Omega(I_2).$$

Spatial Metrics. Similarly to depth-based metrics, we first normalize the point clouds to range $[-1, 1]$. For Chamfer Distance (CD) [1] we employ the PyTorch3D implementation to compute the value between $\mathcal{P}_0$ and $\mathcal{P}_e$ as

$$d_{\text{CD}}(P, Q) = \frac{1}{|P|} \sum_{p \in P} \min_{q \in Q} \|p - q\|_2^2 \; + \; \frac{1}{|Q|} \sum_{q \in Q} \min_{p \in P} \|q - p\|_2^2.$$

Voxel IoU is computed as

$$\text{IoU}(V_{\text{pred}}, V_{\text{gt}}) = \frac{|\, V_{\text{pred}} \cap V_{\text{gt}} \,|}{|\, V_{\text{pred}} \cup V_{\text{gt}} \,|}$$

for a voxel grid V which is computed as

$$V(P; \mathrm{v_s}) = \{\, v(p) \;\mid\; p \in P \,\}, \quad P \subset \mathbb{R}^3$$

where $\mathrm{v_s}$ is voxel size, and

$$v(p) = \left(\left\lfloor \tfrac{x}{\mathrm{v_s}} \right\rfloor, \left\lfloor \tfrac{y}{\mathrm{v_s}} \right\rfloor, \left\lfloor \tfrac{z}{\mathrm{v_s}} \right\rfloor \right).$$

2T2S Metric. Our novel metric is based on comparing deep feature embeddings learned by a point cloud feature encoder. The choice of network layers used for metric calculation relates to their individual architectures. Our chosen point cloud encoders are typically organized into consecutive blocks, where each block encapsulates a series of transformations, and learned weights to extract complex semantic embeddings. In this work, we focus on extracting sets of initial layers of these blocks to include features from both early and later stage layers of networks. All networks were trained in the Pointcept framework [4] for a point cloud classification task. The ModelNet40 [11] dataset was chosen for its variety of object shapes, with partially similar characteristics as point cloud representations of lost cargo objects. The networks except for PTV3 were trained for 100 epochs using an Nvidia Titan RTX 24GB GPU, using the cross entropy loss, and optimizing with stochastic gradient descent. PTV3, which was trained on an Nvidia RTX 4090 16GB GPU due to the hardware supporting the FlashAttention [5] mechanism, using a combination of cross entropy loss and Lovasz Loss [2], and optimizing with AdamW [7]. Figure 2 shows the correlation of qualitative results and metric value for the objects 'Stacked pallets' and 'Bumper' from the winter dataset.

3. Depth Estimation Implementation Details

This Section, adds more details how depth from images (gated, rgb, rccb) was obtained. In total, we evaluate the following methods [3, 6, 12] with different modality combinations as input. From those methods [6] did not require fine tuning due to the foundational nature of the method, while [3, 12] were fine tuned described subsequently. For fine tuning, we collected an additional holdout dataset with 600 samples that did not overlap with the lost-cargo test set under winter and summer conditions. This data set contains all camera images with synchronized Velodyne VLS-128 point clouds. For supervision the pointclouds are projected into respective camera images and the methods are trained fully supervised applying an L1-loss:

$$\mathcal{L}_{\text{depth}} \;=\; \frac{\left\| M \odot (D_{\text{pred}} - D_{\text{lidar}}) \right\|_1}{\|M\|_1}, \tag{12}$$

where $D_{\text{pred}} \in \mathbb{R}^{B \times H \times W}$ denotes the predicted depth maps, $D_{\text{lidar}} \in \mathbb{R}^{B \times H \times W}$ the LiDAR-projected ground-truth depth images, and $M \in \{0, 1\}^{B \times H \times W}$ is a binary mask indicating the valid LiDAR pixels ($M_{b,i,j} = 1$ if $D^{(b)}_{\text{lidar}}(i, j) > 0$, and 0 otherwise). The operator $\odot$ denotes element-wise multiplication, and $\| \cdot \|_1$ the element-wise L1 norm.

All methods were fine-tuned on a single NVIDIA A10G GPU with batch-size 1.

For Gated RCCB Stereo we use [3] with weights pre-trained on the Gated Stereo dataset [3]. The model is finetuned separately for summer and winter, each for 5 epochs with a learning rate of 0.0001, while keeping the image encoders frozen due to compute constraints on the A10 GPUs. This method is selected for its focus on high-resolution inputs and outputs. Gated images are rectified and cropped to heights 104–616 and widths 128–1152 and provided at 1024×512 px and RCCB images are rectified and cropped to heights 319–1855 and widths 388–3460 and provided at 3072×1536 px, with depth maps predicted at the RCCB resolution.

For both RGB and RCCB stereo, we use IGEV-Stereo [12] for its strong zero-shot performance and detailed depth predictions. The model is fine-tuned from the KITTI checkpoint for 10 epochs with a learning rate of 0.0002. RGB images are

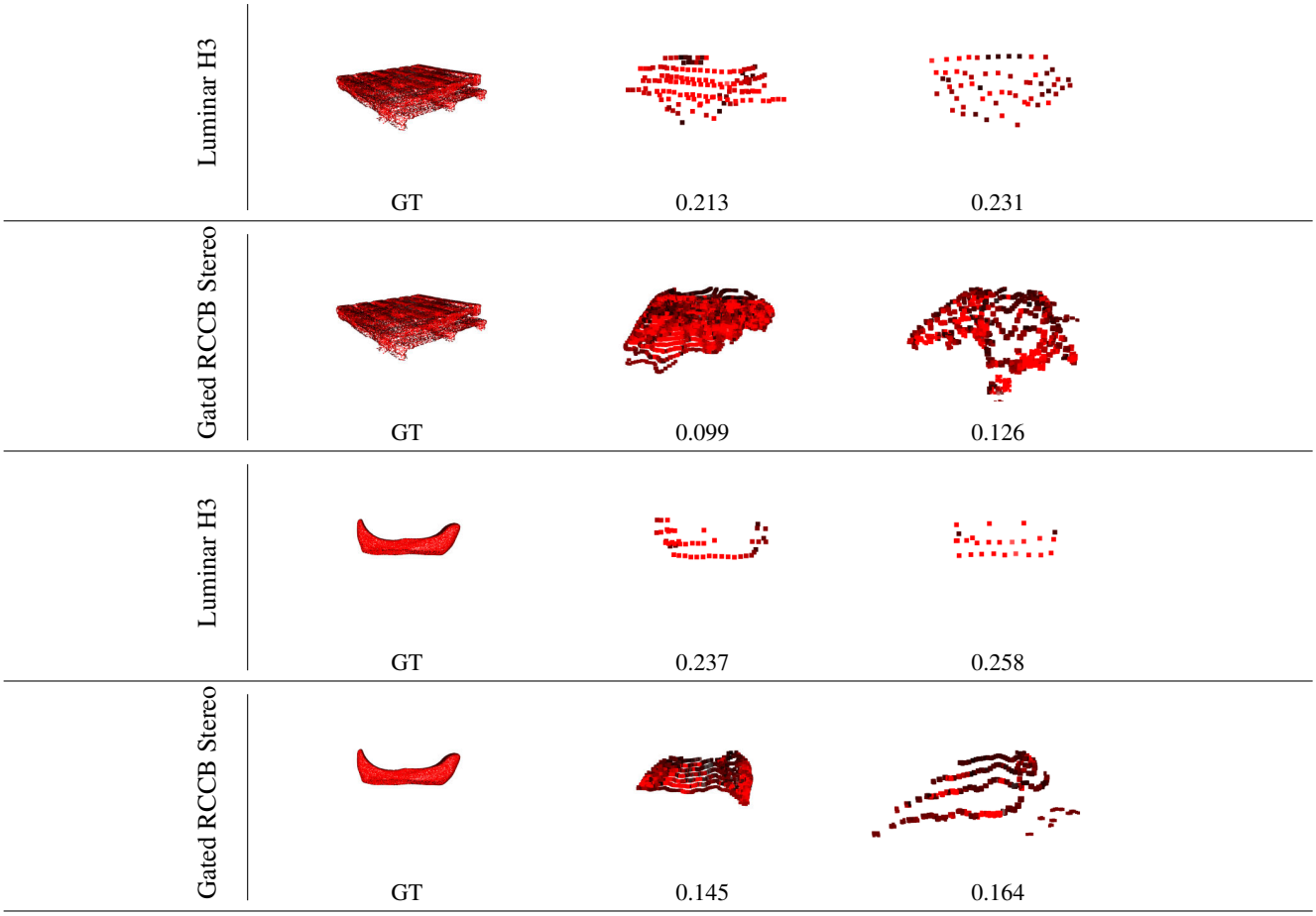

Figure 2. **Visual quality of objects and their corresponding 2T2S metric value**. From left to right the ground truth mesh, the reconstructions at 25 meters range and 50 meters range are respectively presented for the Luminar H3 LiDAR and for Gated RCCB Stereo [3]. The LiDAR yields sparse point clouds with less spatial detail, while the Gated RCCB Stereo method provides significantly more density and and better shape geometry, resulting in a better (lower) 2T2S (our) metric.

rectified and downsampled to 960×512 px, while RCCB images are rectified, cropped as in Gated RCCB Stereo, and downsampled to 1024×512 px, with depth maps matching each input resolution. Inputs are downsampled from full resolution due to compute limitations.

For the monocular RGB baseline, we use Metric3Dv2 [6], a recently proposed depth foundation model with strong zero-shot performance in metric depth estimation. We do not finetune the model, as our goal is to evaluate the capability of published monocular foundation models directly. Specifically, we adopt the metric3d_vit_small weights provided by the authors. The input is the left rectified RGB image at full resolution (1920×1024), and the output depth map matches this resolution.

3.1. Depth Map Creation

Example depth maps for all methods are shown in Fig. 7. Depth maps were created for lost-cargo distances of 30, 45, 60, 75, 90 meters for summer sequences and 25, 50, 75 and 100 meters for winter sequences. Depth maps are projected into 3D space using the intrinsic calibration of the camera. For each pixel (u, v) with corresponding depth value d, the 3D point in the camera coordinate system is obtained as

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = d \cdot K^{-1} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}, \tag{13}$$

where K denotes the intrinsic calibration matrix. The resulting 3D points are subsequently transformed into the LiDAR coordinate frame of the Velodyne VLS-128 by applying the extrinsic calibration parameters (R, t):

$$\mathbf{p}_{\text{LiDAR}} = R\,\mathbf{p}_{\text{cam}} + t, \tag{14}$$

with $R \in SO(3)$ and $t \in \mathbb{R}^3$ representing the rotation and translation, respectively. This procedure ensures that the reconstructed 3D structure from the depth maps is spatially aligned with the LiDAR reference frame and enables the placement of ground-truth lost-cargo meshes in the correct location.

4. Additional Ablations

In support of the ablation experiments in the main manuscript in Section 6 and Figure 6, we provide further information here. Corresponding quantitative numbers from the graphs are shown in Tables 1 and 2.

To add further details, for each architecture, 4 intermediate feature layers are selected for feature aggregation to assess the similarity at different down sampling stages. For Point Transformer V3 [10] we aggregate the output of the attention layers of the first 4 'enc' blocks. For SparseUNet [4] and OA-CNN [8], the outputs of the first convolution layer in the first 4 'enc' blocks are used. As in the other metric estimations, the point cloud coordinates are normalized to a range $[-1, 1]$ before passing to the network. Additionally, the point clouds are preprocessed by grid sampling at a resolution of 0.01 m.

5. Additional Results

In this Section, we present further result tables and figures supporting the findings in Section 6 of the main manuscript. Tables 3 an 4 show quantative numbers for each perception method at averaged accross all object categories at measurement distance per row. Figures 8, 9, 10, 11, 12 extend Figure 4 of the main manuscript with examples of summer captures, while Figures 13, 14, 15, 16, 17 show object examples for the winter captures.

References

[1] Harry G. Barrow, Jay M. Tenenbaum, Robert C. Bolles, and Helen C. Wolf. Parametric correspondence and chamfer matching: Two new techniques for image matching. In *Proceedings of the 5th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 659–663. Morgan Kaufmann Publishers Inc., 1977. 5

[2] Maxim Berman, Amal Rannen Triki, and Matthew B. Blaschko. The Lovász-softmax loss: A tractable surrogate for the optimization of the intersection-over-union measure in neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4413–4421, 2018. 5

[3] Samuel Brucker, Stefanie Walz, Mario Bijelic, and Felix Heide. Cross-spectral gated-rgb stereo depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21654–21665, 2024. 5, 6, 10, 11, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26

[4] Pointcept Contributors. Pointcept: A codebase for point cloud perception research. https://github.com/Pointcept/Pointcept, 2023. 4, 5, 7

[5] Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. FlashAttention: Fast and memory-efficient exact attention with IO-awareness. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. 5

[6] Mu Hu, Wei Yin, Chi Zhang, Zhipeng Cai, Xiaoxiao Long, Hao Chen, Kaixuan Wang, Gang Yu, Chunhua Shen, and Shaojie Shen. Metric3d v2: A versatile monocular geometric foundation model for zero-shot metric depth and surface normal estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024. 5, 6, 10, 11, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26

[7] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*, 2019. 5

[8] Bohao Peng, Xiaoyang Wu, Li Jiang, Yukang Chen, Hengshuang Zhao, Zhuotao Tian, and Jiaya Jia. Oa-cnns: Omni-adaptive sparse cnns for 3d semantic segmentation, 2024. 7

[9] Nikhila Ravi, Jeremy Reizenstein, David Novotny, Oliver Gordon, Wan-Yen Lo, Justin Johnson, and Andrea Vedaldi. Pytorch3d: An open-source library for 3d deep learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2020. 4

[10] Xiaoyang Wu, Li Jiang, Peng-Shuai Wang, Zhijian Liu, Xihui Liu, Yu Qiao, Wanli Ouyang, Tong He, and Hengshuang Zhao. Point transformer v3: Simpler, faster, stronger, 2024. 7

[11] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1912–1920, 2015. 5

[12] Gangwei Xu, Xianqi Wang, Xiaohuan Ding, and Xin Yang. Iterative geometry encoding volume for stereo matching. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 21919–21928, 2023. 5, 10, 11, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26

[13] Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. Open3d: A modern library for 3d data processing. *arXiv preprint arXiv:1801.09847*, 2018. 4

[14] Matthias Zwicker, Hanspeter Pfister, Jeroen Van Baar, and Markus Gross. Surface splatting. *Proceedings of the 28th annual conference on Computer graphics and interactive techniques (SIGGRAPH)*, pages 371–378, 2001. 4

Sector [m]	Method	SparseUNet	PTV3	OA-CNN	SPVCNN
30	Gated RCCB Stereo	0.148	0.103	0.137	0.152
	Luminar H3	0.379	0.204	0.201	0.256
	RCCB Stereo	0.216	0.147	0.161	0.187
	RGB Mono	0.135	0.118	0.146	0.167
	RGB Stereo	0.148	0.150	0.168	0.199
	VLS-128	0.341	0.201	0.202	0.256
45	Gated RCCB Stereo	0.137	0.099	0.130	0.142
	Luminar H3	0.380	0.210	0.221	0.285
	RCCB Stereo	0.269	0.166	0.168	0.201
	RGB Mono	0.188	0.139	0.158	0.188
	RGB Stereo	0.175	0.168	0.178	0.212
	VLS-128	0.347	0.211	0.216	0.286
60	Gated RCCB Stereo	0.117	0.096	0.146	0.157
	Luminar H3	0.342	0.199	0.230	0.313
	RCCB Stereo	0.246	0.175	0.174	0.211
	RGB Mono	0.232	0.163	0.172	0.209
	RGB Stereo	0.202	0.183	0.184	0.220
	VLS-128	0.354	0.211	0.234	0.318
75	Gated RCCB Stereo	0.174	0.124	0.151	0.174
	Luminar H3	0.338	0.197	0.207	0.272
	RCCB Stereo	0.273	0.180	0.175	0.219
	RGB Mono	0.250	0.182	0.183	0.222
	RGB Stereo	0.220	0.191	0.200	0.243
	vls-128	0.284	0.216	0.245	0.341
90	Gated RCCB Stereo	0.153	0.142	0.165	0.191
	Luminar H3	0.371	0.198	0.224	0.300
	RCCB Stereo	0.256	0.194	0.191	0.233
	RGB Mono	0.280	0.197	0.187	0.226
	RGB Stereo	0.303	0.198	0.204	0.272
	VLS-128	0.426	0.215	0.232	0.334

Table 1. **Feature Encoder Architectures Ablation Study: Summer Captures 2T2S Scores.** Complementing Figure 6 of the main manuscript, we report quantitative results of averaged metric values for different feature encoder architectures, on summer captures and for all distances.

Sector [m]	Method	SparseUNet	PTV3	SPVCNN	OA-CNN
25	Gated RCCB Stereo	0.140	0.083	0.137	0.115
	Luminar H3	0.335	0.193	0.241	0.195
	RCCB Stereo	0.149	0.096	0.158	0.133
	RGB Mono	0.160	0.144	0.187	0.160
	RGB Stereo	0.208	0.147	0.205	0.170
	VLS-128	0.337	0.196	0.244	0.194
50	Gated RCCB Stereo	0.148	0.098	0.164	0.132
	Luminar H3	0.361	0.195	0.258	0.204
	RCCB Stereo	0.241	0.163	0.212	0.169
	RGB Mono	0.227	0.180	0.223	0.181
	RGB Stereo	0.291	0.202	0.251	0.199
	VLS-128	0.365	0.202	0.281	0.215
75	Gated RCCB Stereo	0.183	0.131	0.193	0.156
	Luminar H3	0.347	0.196	0.276	0.212
	RCCB Stereo	0.278	0.191	0.257	0.209
	RGB Mono	0.248	0.191	0.226	0.187
	RGB Stereo	0.296	0.210	0.272	0.216
	VLS-128	0.342	0.202	0.314	0.229

Table 2. **Feature Encoder Architectures Ablation Study: Winter Captures 2T2S Scores.** Complementing Figure 6 of the main manuscript, we report quantitative results of averaged metric values for different feature encoder architectures, on winter captures and for all distances.

Sector [m]	Method	Silog	RMSE Log	MAE	Abs Rel	RMSE	CD	Voxel IoU	2T2S
30	Gated RCCB Stereo	7.007	0.074	0.286	0.059	0.364	0.070	**0.377**	**0.152**
	Luminar H3	3.060	0.039	0.150	0.032	0.190	0.106	0.231	**0.256**
	RCCB Stereo	7.592	0.082	0.327	0.068	0.406	0.071	0.389	**0.187**
	RGB Mono	6.208	0.073	0.302	0.063	0.366	**0.053**	0.367	**0.167**
	RGB Stereo	10.459	0.106	0.428	0.089	0.518	0.103	0.306	**0.199**
	VLS-128	**2.982**	**0.035**	**0.126**	**0.026**	**0.172**	0.149	0.210	**0.256**
45	Gated RCCB Stereo	8.188	0.085	0.329	0.069	0.418	**0.072**	**0.382**	**0.142**
	Luminar H3	**2.153**	**0.030**	**0.122**	**0.026**	**0.145**	0.141	0.175	**0.285**
	RCCB Stereo	8.607	0.090	0.357	0.074	0.448	0.084	0.326	**0.201**
	RGB Mono	9.263	0.101	0.418	0.088	0.494	0.098	0.321	**0.188**
	RGB Stereo	10.453	0.106	0.430	0.089	0.525	0.106	0.269	**0.212**
	VLS-128	4.879	0.053	0.206	0.042	0.266	0.164	0.101	**0.286**
60	Gated RCCB Stereo	7.862	0.082	0.321	0.068	0.404	**0.069**	**0.380**	**0.157**
	Luminar H3	4.132	0.057	0.224	0.048	0.280	0.150	0.105	**0.313**
	RCCB Stereo	9.023	0.095	0.379	0.079	0.473	0.086	0.320	**0.211**
	RGB Mono	7.276	0.080	0.331	0.070	0.404	0.104	0.251	**0.209**
	RGB Stereo	10.116	0.104	0.428	0.089	0.517	0.094	0.281	**0.220**
	VLS-128	**2.989**	**0.039**	**0.141**	**0.028**	**0.190**	0.241	0.082	**0.318**
75	Gated RCCB Stereo	9.039	0.095	0.397	0.083	0.472	0.098	**0.357**	**0.174**
	Luminar H3	**1.252**	**0.020**	**0.083**	**0.017**	**0.099**	**0.091**	0.155	**0.272**
	RCCB Stereo	9.293	0.100	0.405	0.085	0.502	0.096	0.275	**0.219**
	RGB Mono	11.329	0.121	0.521	0.109	0.598	0.147	0.248	**0.222**
	RGB Stereo	9.998	0.102	0.418	0.087	0.505	0.105	0.219	**0.243**
	VLS-128	5.738	0.068	0.309	0.066	0.328	0.126	0.089	**0.341**
90	Gated RCCB Stereo	9.480	0.105	0.442	0.093	0.521	0.100	**0.317**	**0.191**
	Luminar H3	2.578	0.030	0.125	0.027	0.146	0.189	0.084	**0.300**
	RCCB Stereo	10.410	0.111	0.451	0.095	0.546	0.106	0.205	**0.233**
	RGB Mono	11.354	0.116	0.477	0.099	0.577	0.168	0.209	**0.226**
	RGB Stereo	10.579	0.113	0.476	0.098	0.567	0.188	0.122	**0.272**
	VLS-128	**1.554**	**0.026**	**0.107**	**0.022**	**0.129**	**0.045**	0.134	**0.334**

Table 3. **Metric scores for all distance sectors for summer dataset.** The values of 2T2S metric are seen generally increasing along with measurement distance: **Gated RCCB Stereo** [3], **Luminar H3**, **RCCB Stereo** [12], **RGB Mono** [6], **RGB Stereo** [12], **VLS-128**. The best scoring depth perception method for each reference metric at each distance is highlighted.

Sector [m]	Method	Silog	RMSE Log	RMSE	MAE	Abs Rel	CD	Voxel IoU	2T2S
25	Gated RCCB Stereo	4.973	0.062	0.298	0.255	0.054	0.056	0.346	**0.137**
	Luminar H3	3.794	0.048	0.235	0.186	0.039	0.041	**0.431**	**0.241**
	RCCB Stereo	3.847	0.050	0.246	0.207	0.044	**0.024**	0.416	**0.158**
	RGB Mono	7.864	0.087	0.426	0.354	0.074	0.079	0.309	**0.187**
	RGB Stereo	5.938	0.072	0.357	0.296	0.062	0.051	0.317	**0.205**
	VLS-128	**3.088**	**0.040**	**0.195**	**0.158**	**0.033**	0.055	0.352	**0.244**
50	Gated RCCB Stereo	8.908	0.097	0.468	0.389	0.081	0.090	0.310	**0.164**
	Luminar H3	4.907	0.060	0.295	0.242	0.050	0.085	0.271	**0.258**
	RCCB Stereo	5.835	0.066	0.325	0.264	0.055	**0.040**	**0.415**	**0.212**
	RGB Mono	9.563	0.102	0.504	0.418	0.088	0.099	0.270	**0.223**
	RGB Stereo	10.455	0.110	0.548	0.458	0.095	0.114	0.174	**0.251**
	VLS-128	**3.846**	**0.047**	**0.230**	**0.181**	**0.038**	0.092	0.187	**0.281**
75	Gated RCCB Stereo	8.424	0.093	0.450	0.379	0.079	0.120	0.253	**0.193**
	Luminar H3	3.231	0.046	0.226	0.203	0.043	0.095	0.121	**0.276**
	RCCB Stereo	10.623	0.109	0.543	0.455	0.093	0.155	0.168	**0.257**
	RGB Mono	10.646	0.116	0.575	0.481	0.102	0.099	**0.256**	**0.226**
	RGB Stereo	9.752	0.101	0.502	0.424	0.088	0.116	0.143	**0.272**
	VLS-128	**2.820**	**0.046**	**0.224**	**0.197**	**0.042**	**0.085**	0.112	**0.314**

Table 4. **Metric scores for all distance sectors for winter dataset: Gated RCCB Stereo** [3], **Luminar H3**, **RCCB Stereo** [12], **RGB Mono** [6], **RGB Stereo** [12], **VLS-128**. The best scoring depth perception method for each reference metric at each distance is highlighted.

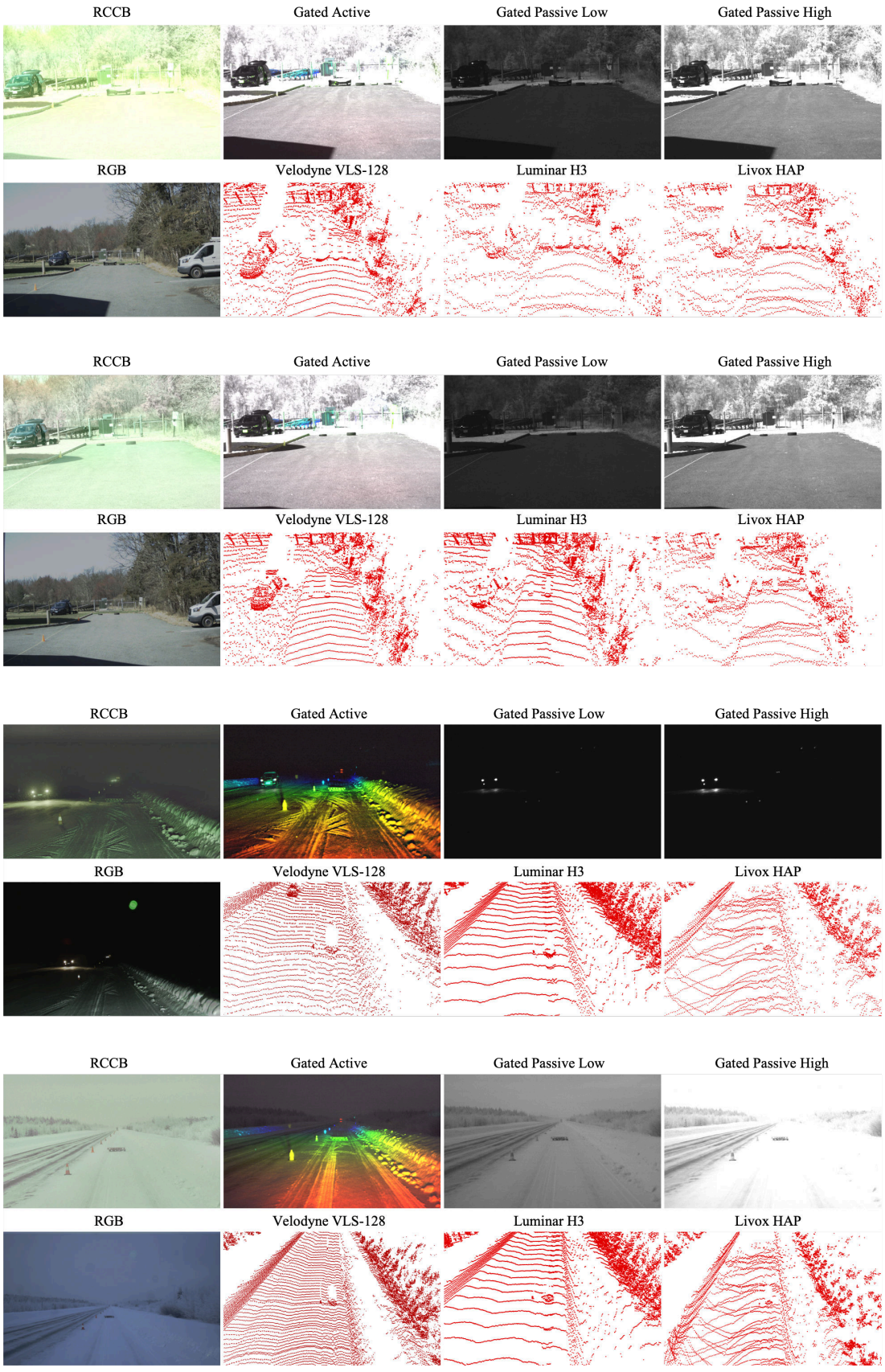

Figure 3. **Dataset Visualization 1** Sequence Nr. 1 - 4.

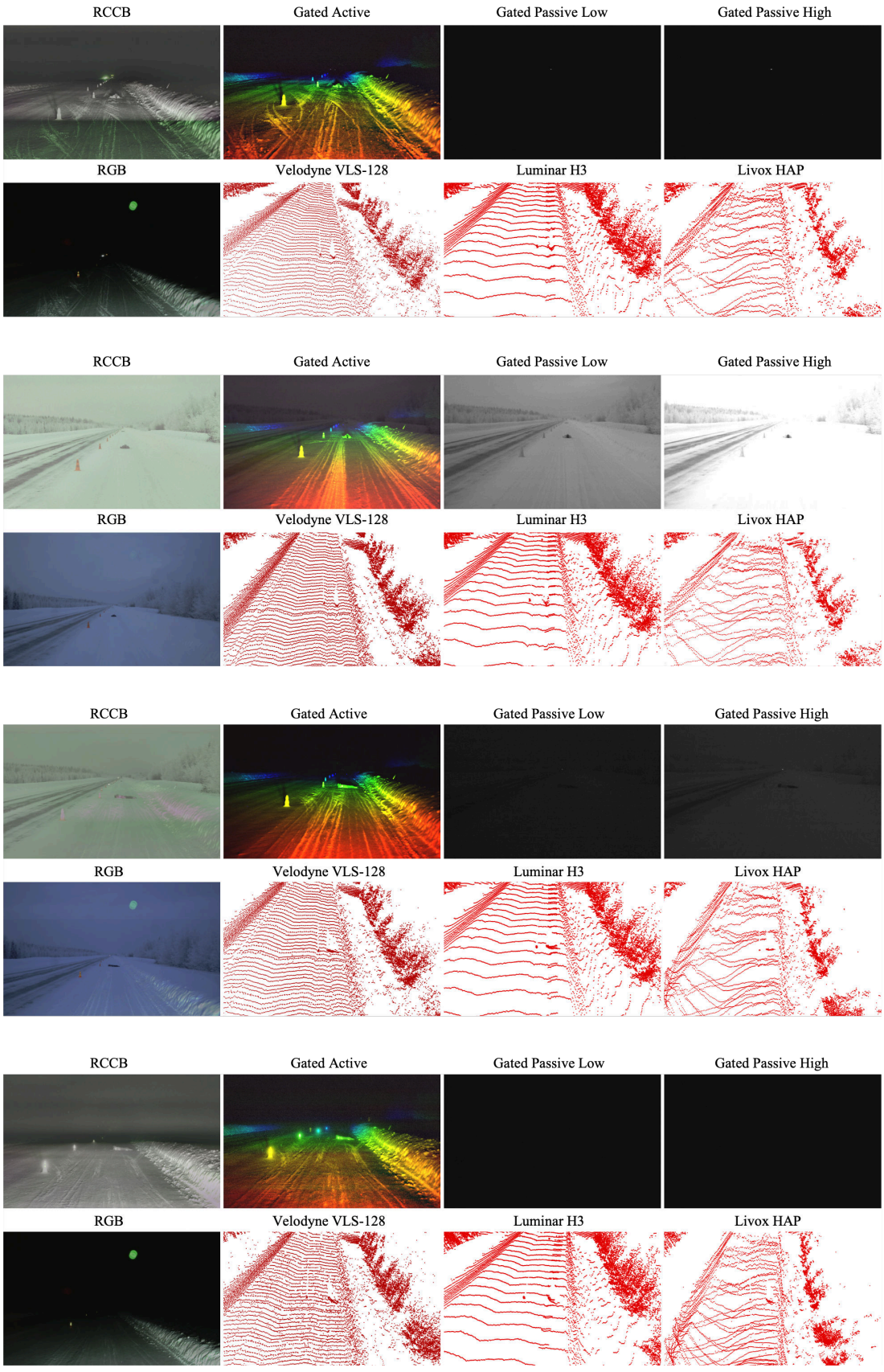

Figure 4. **Dataset Visualization 2** Sequence Nr. 5 - 8

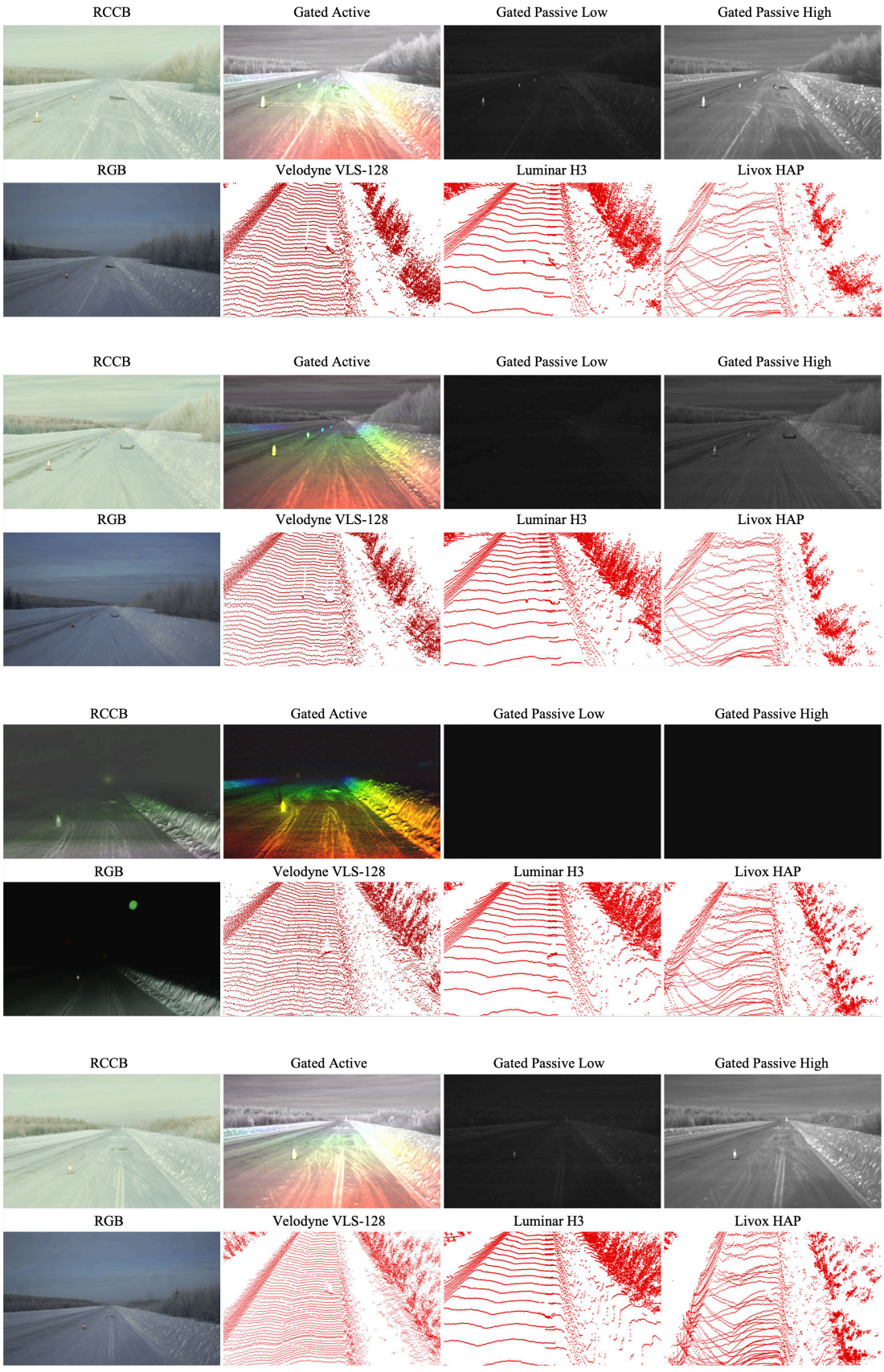

Figure 5. **Dataset Visualization 3** Sequence Nr. 9 - 12

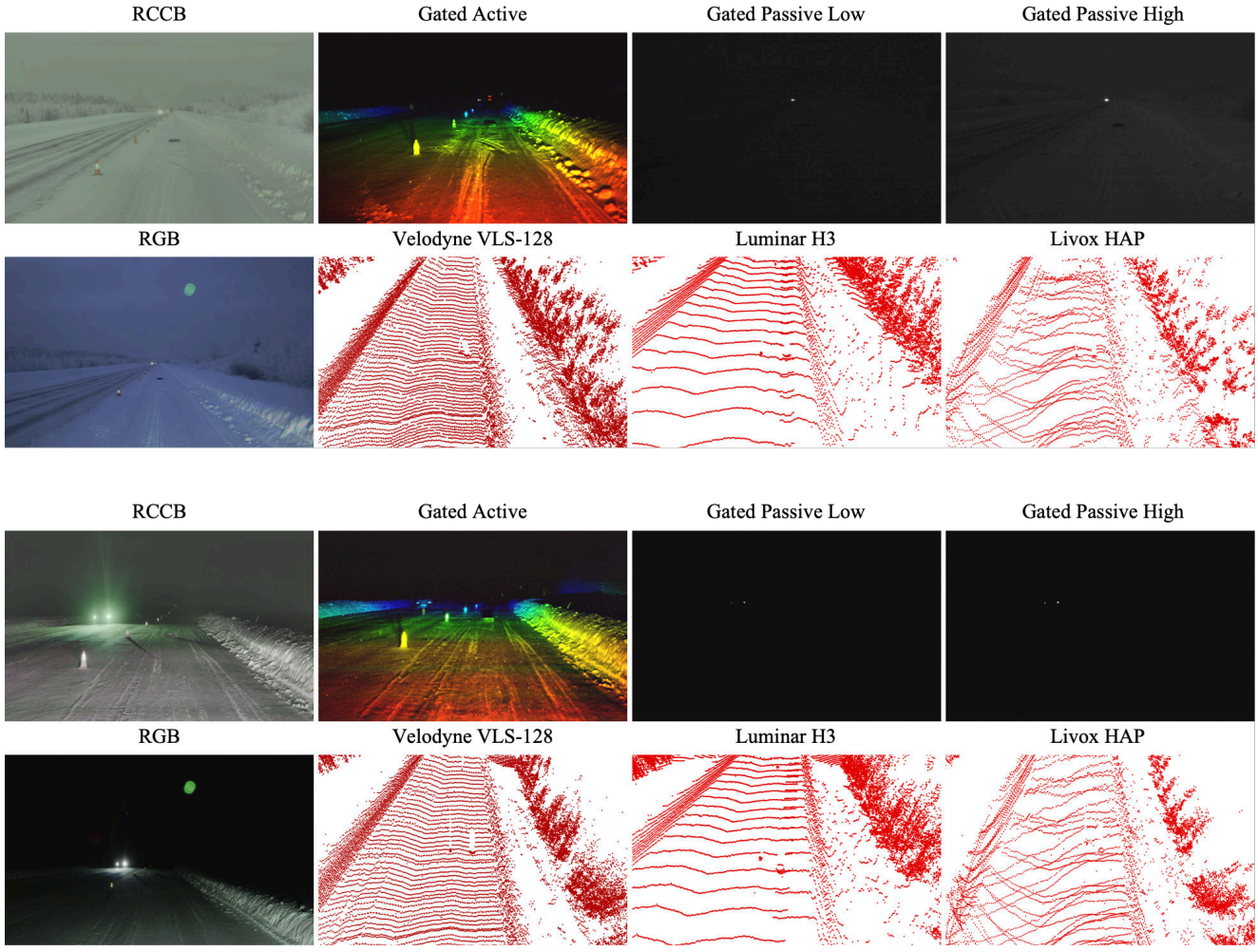

Figure 6. **Dataset Visualization 4** Sequence Nr. 13 - 14

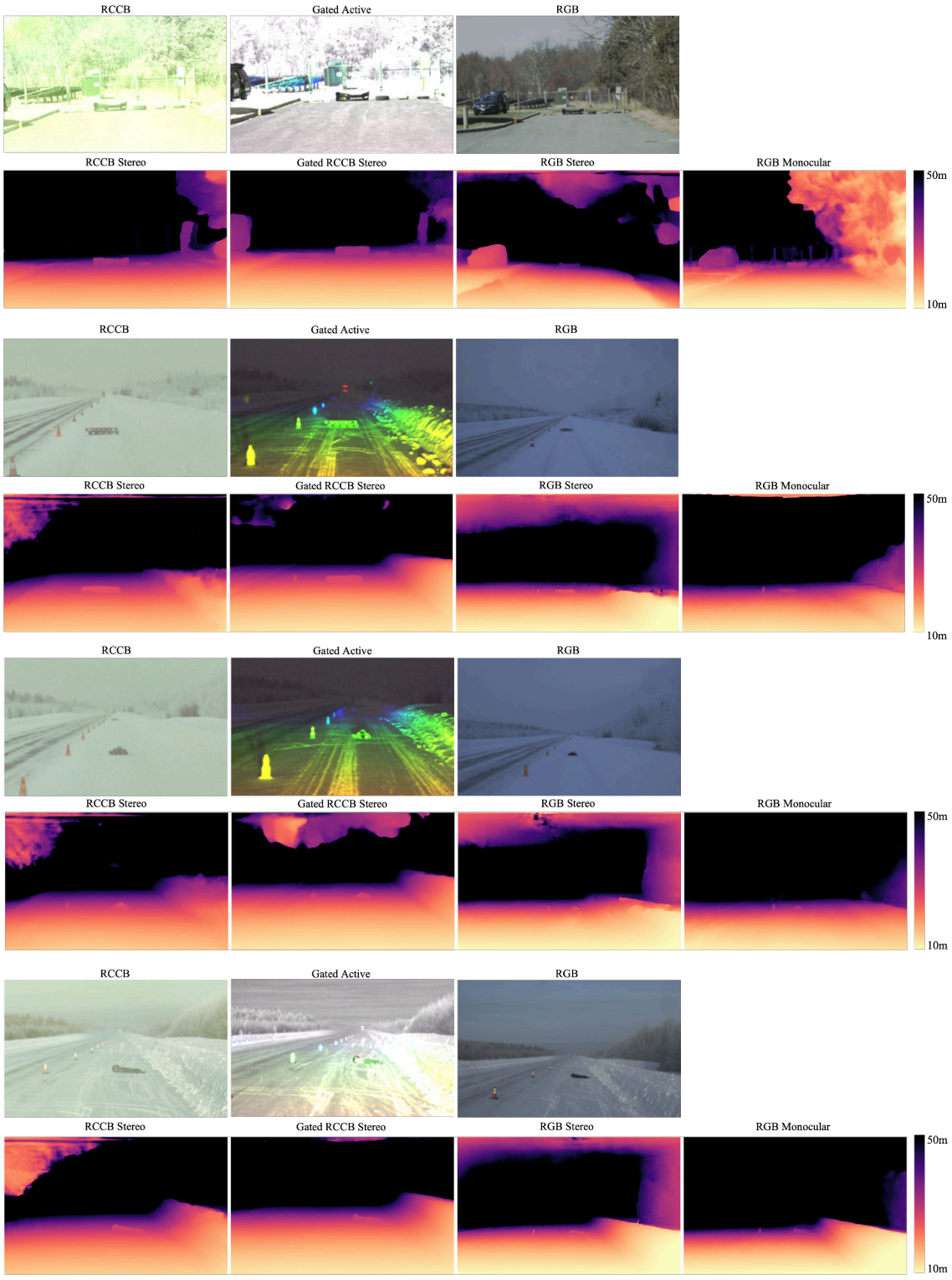

Figure 7. **Depth Map Visualizations** Lost cargo objects are placed at 25 meters in these examples, with reference images taken directly from the raw dataset and preprocessed for depth generation as detailed in Section 3.

Figure 8. **Summer Recording: Small Tire.** Visual results of lost cargos captured at different distances, with different depth estimation methods, comprising cross spectral Gated RCCB fusion [3], RGB Stereo [12], RCCB Stereo [12] and RGB Monocular [6], and LiDAR sensors from Luminar and Velodyne.

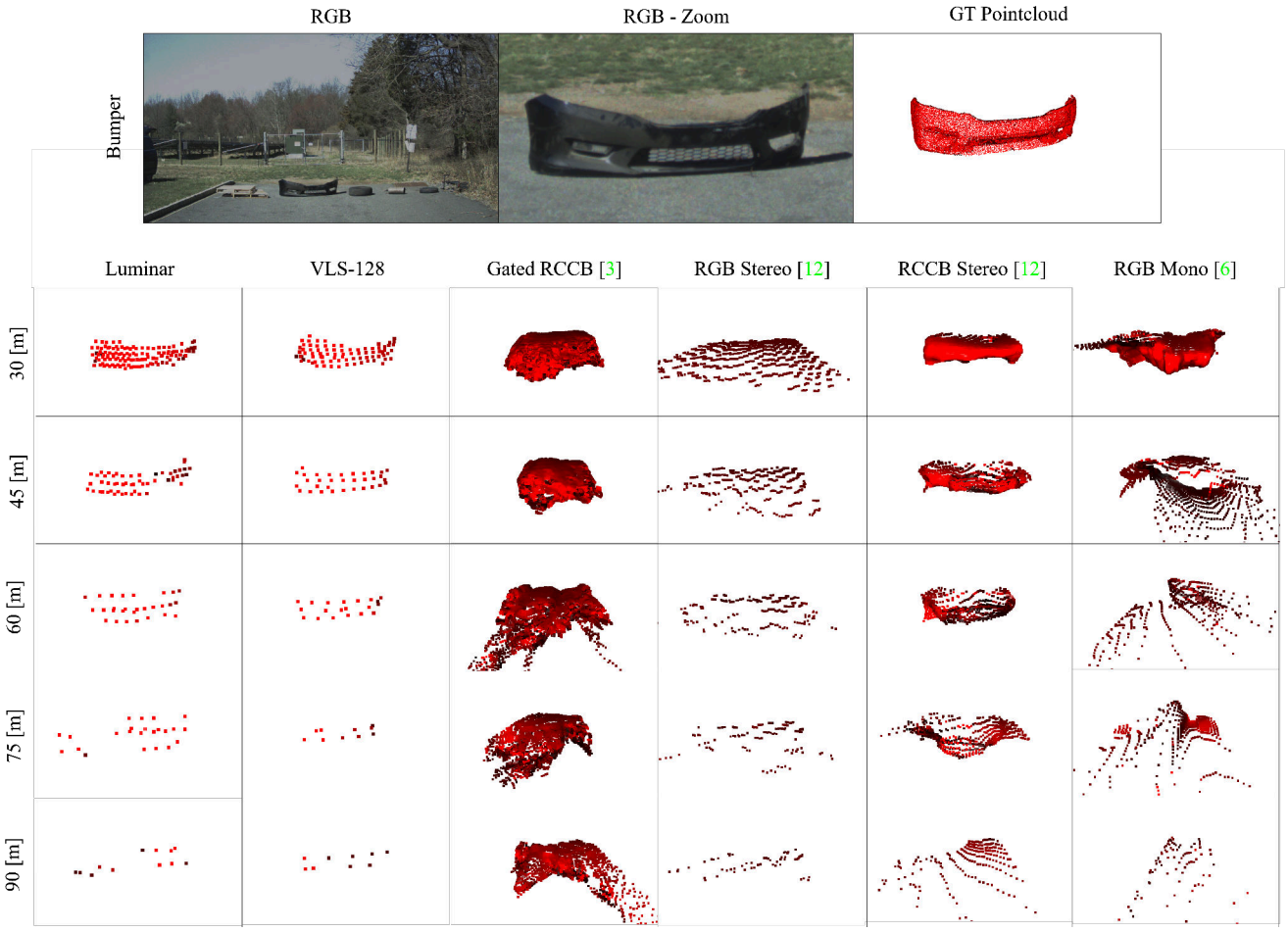

Figure 9. **Summer Recording: Bumper.** Visual results of lost cargos captured at different distances, with different depth estimation methods, comprising cross spectral Gated RCCB fusion [3], RGB Stereo [12], RCCB Stereo [12] and RGB Monocular [6], and LiDAR sensors from Luminar and Velodyne.

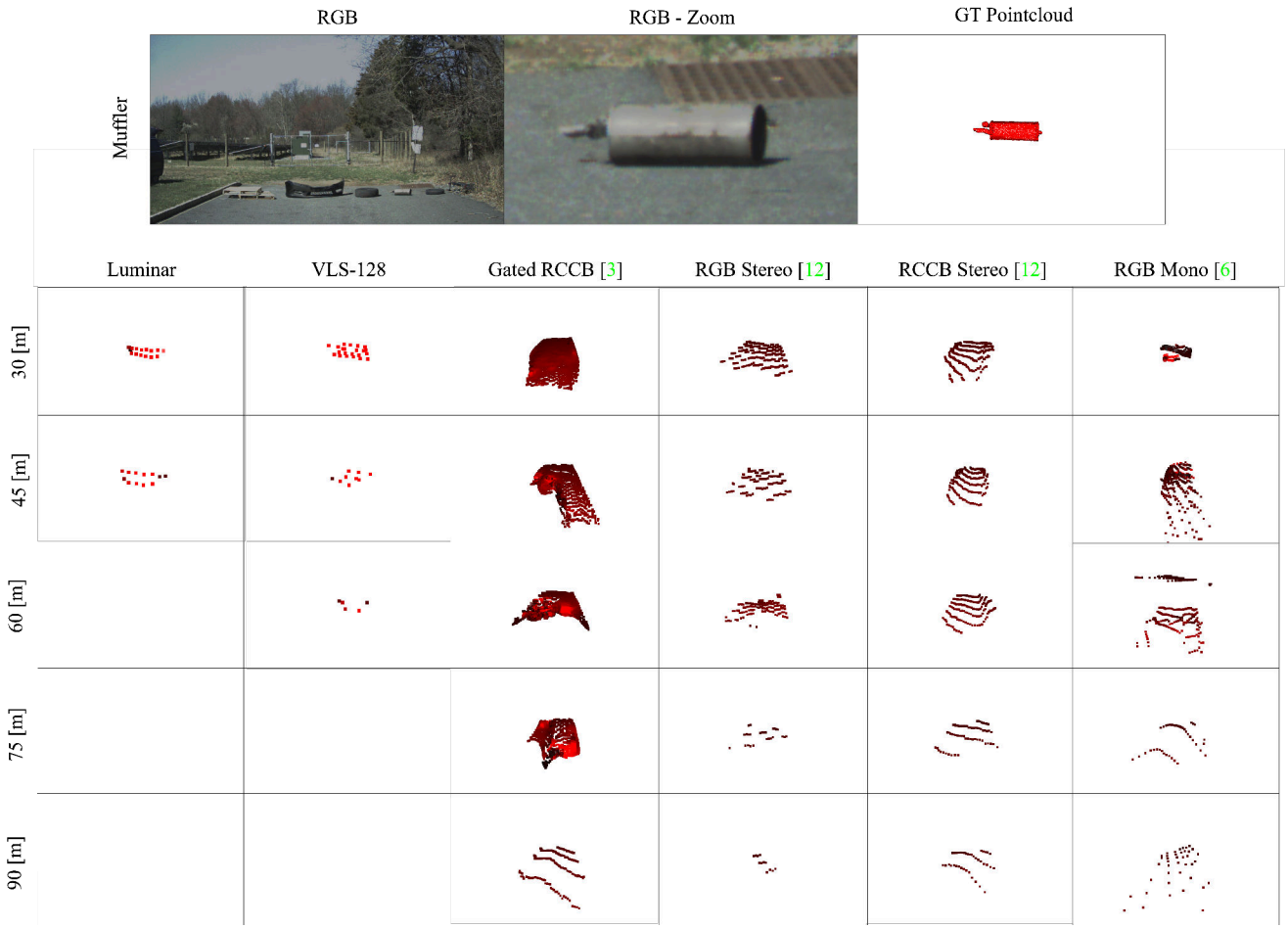

Figure 10. **Summer Recording: Exhaust.** Visual results of lost cargos captured at different distances, with different depth estimation methods, comprising cross spectral Gated RCCB fusion [3], RGB Stereo [12], RCCB Stereo [12] and RGB Monocular [6], and LiDAR sensors from Luminar and Velodyne.

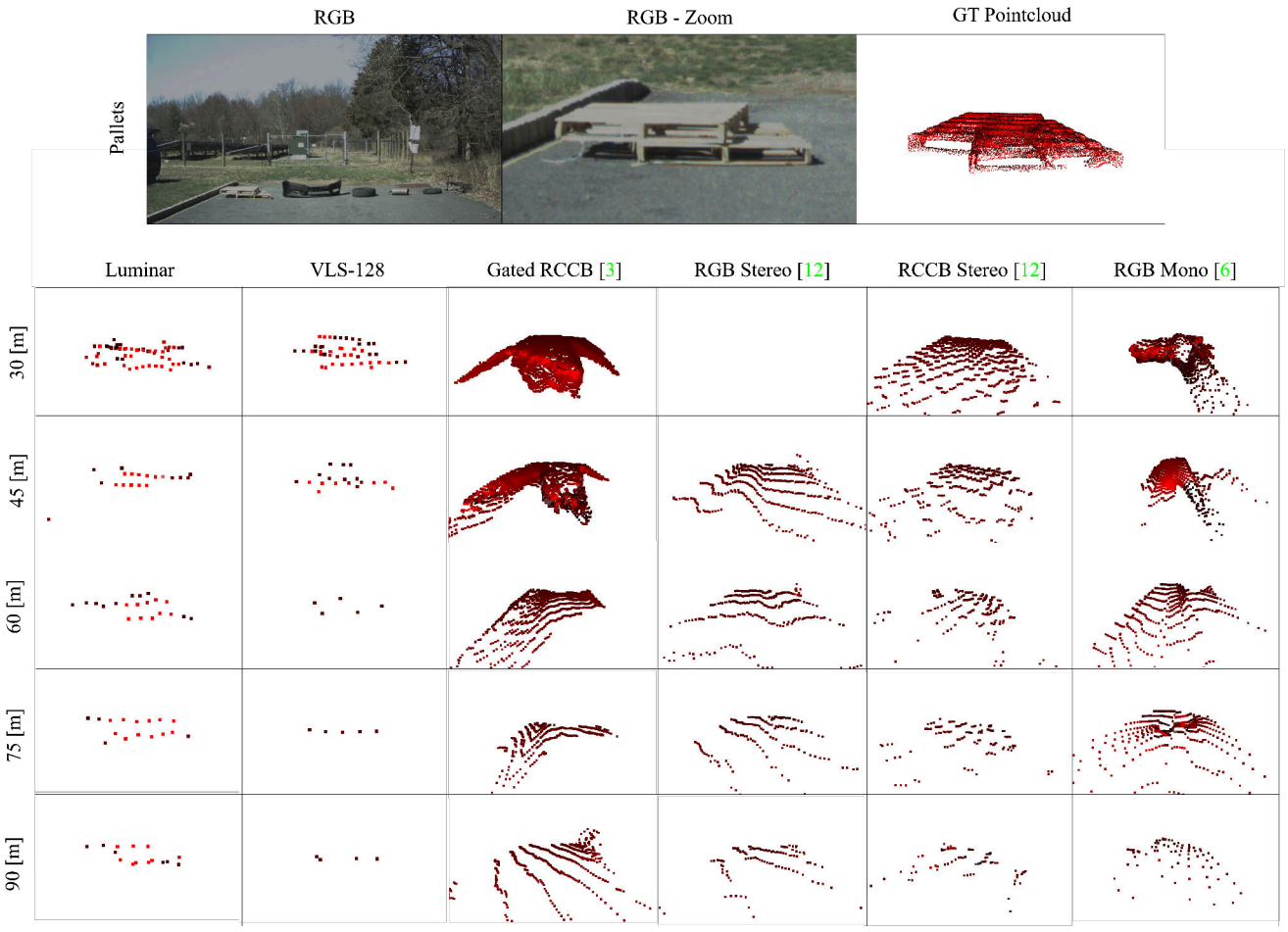

Figure 11. **Summer Recording: Pallets.** Visual results of lost cargos captured at different distances, with different depth estimation methods, comprising cross spectral Gated RCCB fusion [3], RGB Stereo [12], RCCB Stereo [12] and RGB Monocular [6], and LiDAR sensors from Luminar and Velodyne.

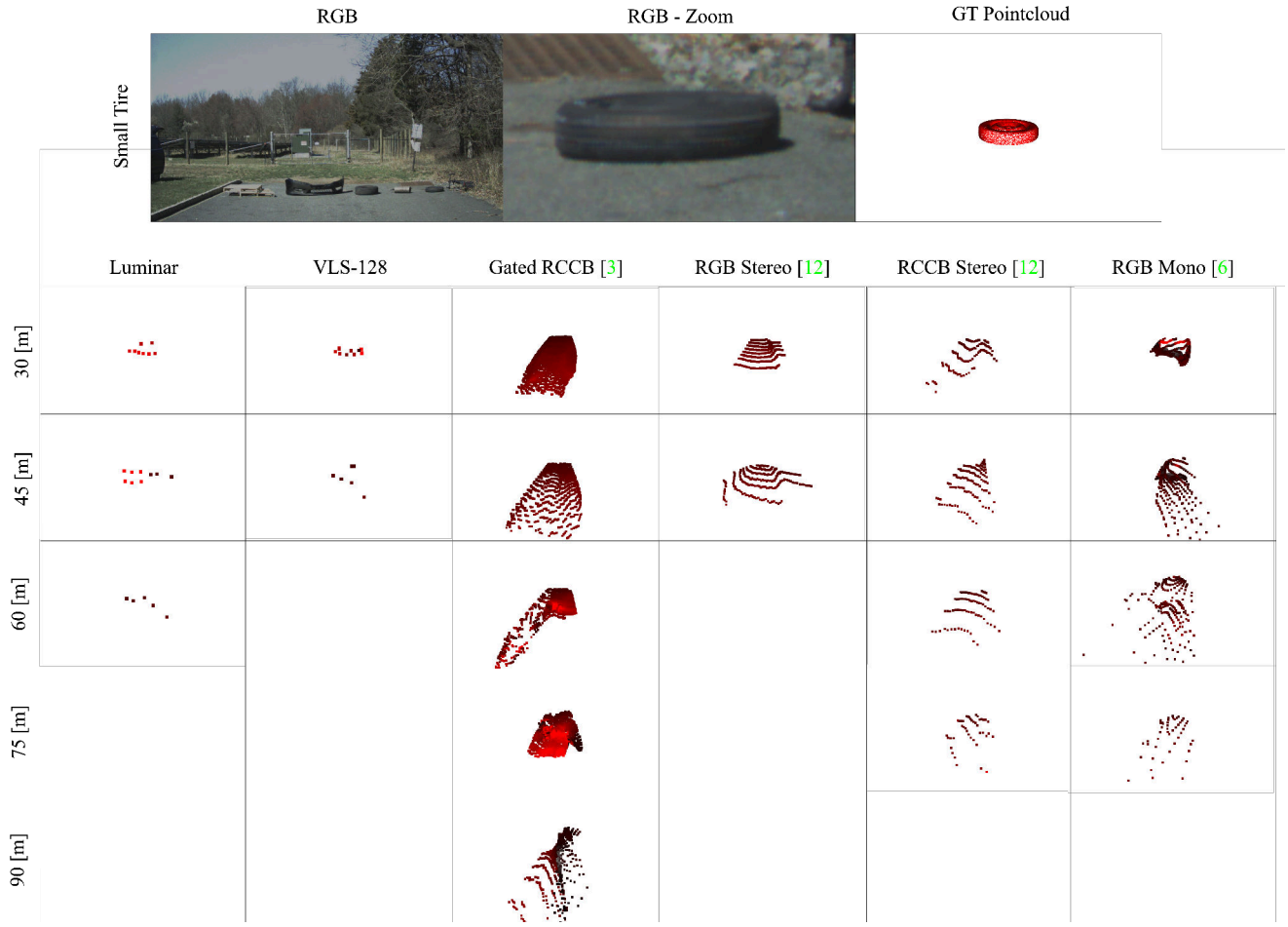

Figure 12. **Summer Recording: Small Tire.** Visual results of lost cargos captured at different distances, with different depth estimation methods, comprising cross spectral Gated RCCB fusion [3], RGB Stereo [12], RCCB Stereo [12] and RGB Monocular [6], and LiDAR sensors from Luminar and Velodyne.

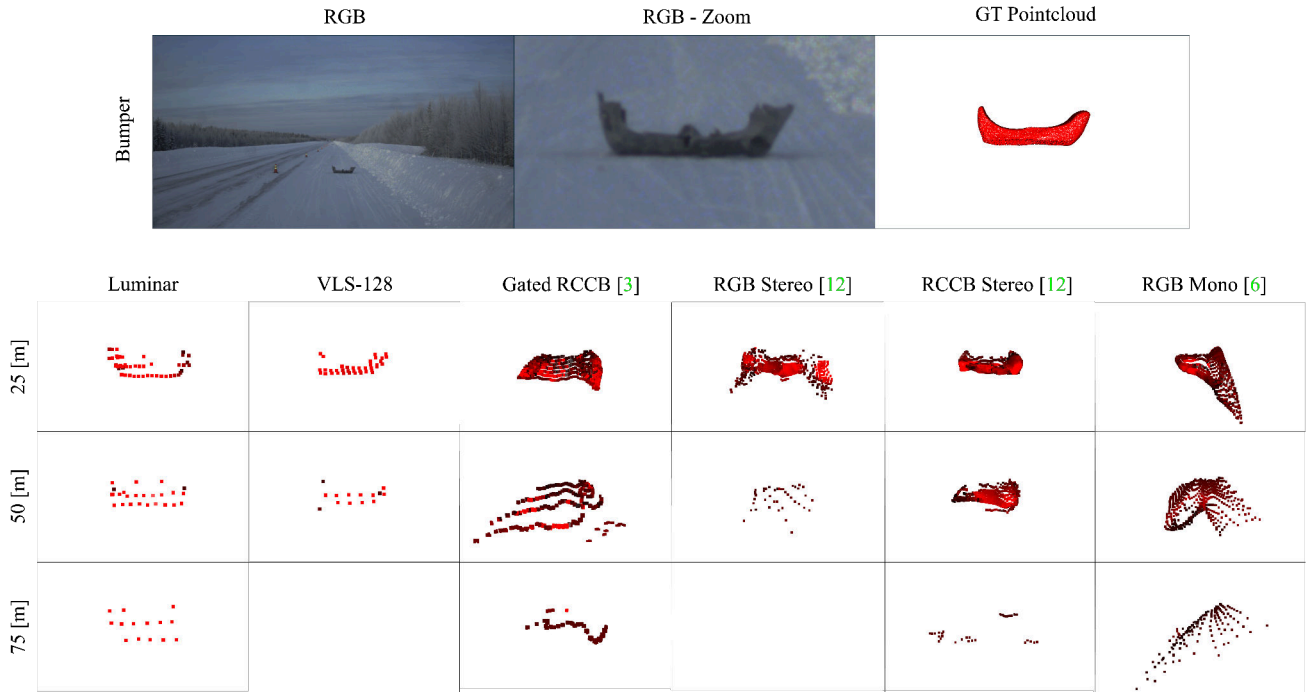

Figure 13. **Winter Recording: Bumper.** Visual results of lost cargos captured at different distances, with different depth estimation methods, comprising cross spectral Gated RCCB fusion [3], RGB Stereo [12], RCCB Stereo [12] and RGB Monocular [6], and LiDAR sensors from Luminar and Velodyne.

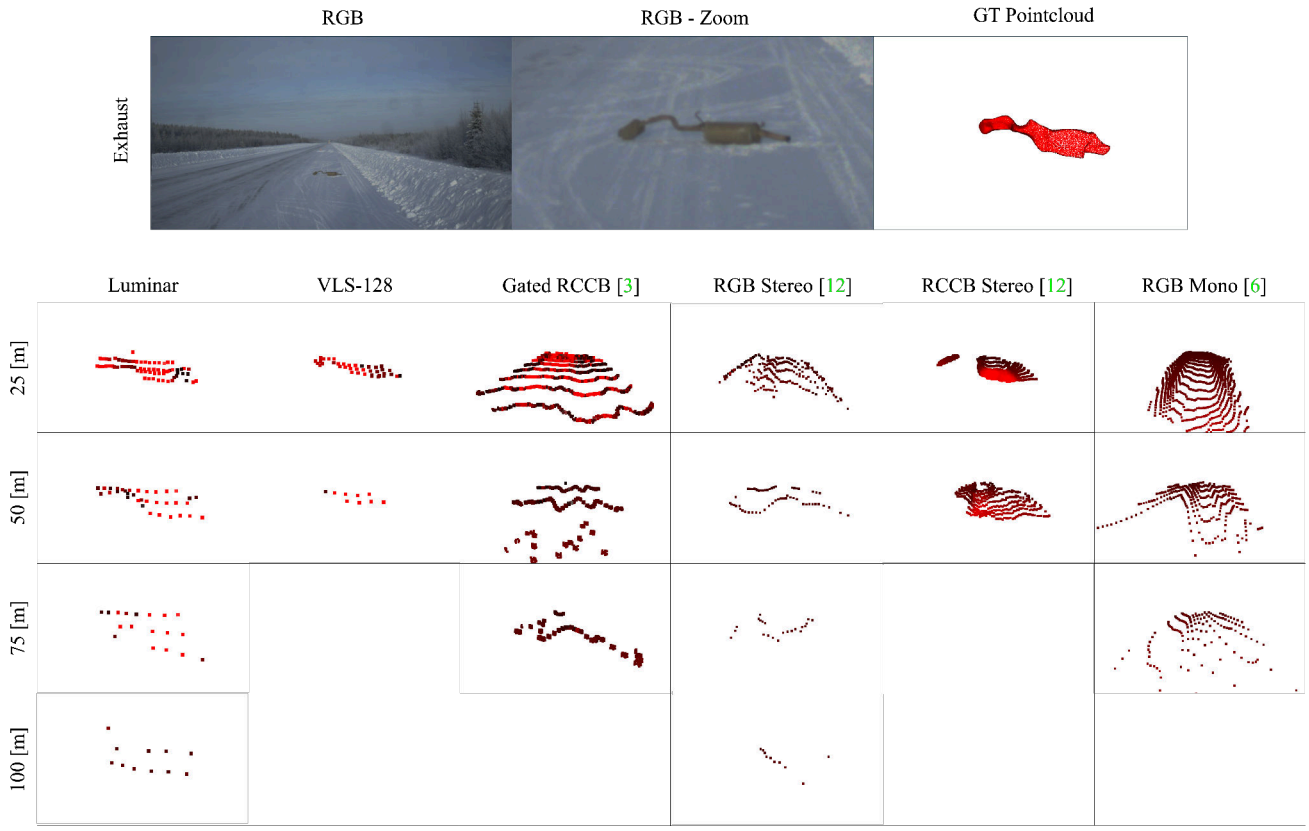

Figure 14. **Winter Recording: Exhaust.** Visual results of lost cargos captured at different distances, with different depth estimation methods, comprising cross spectral Gated RCCB fusion [3], RGB Stereo [12], RCCB Stereo [12] and RGB Monocular [6], and LiDAR sensors from Luminar and Velodyne.

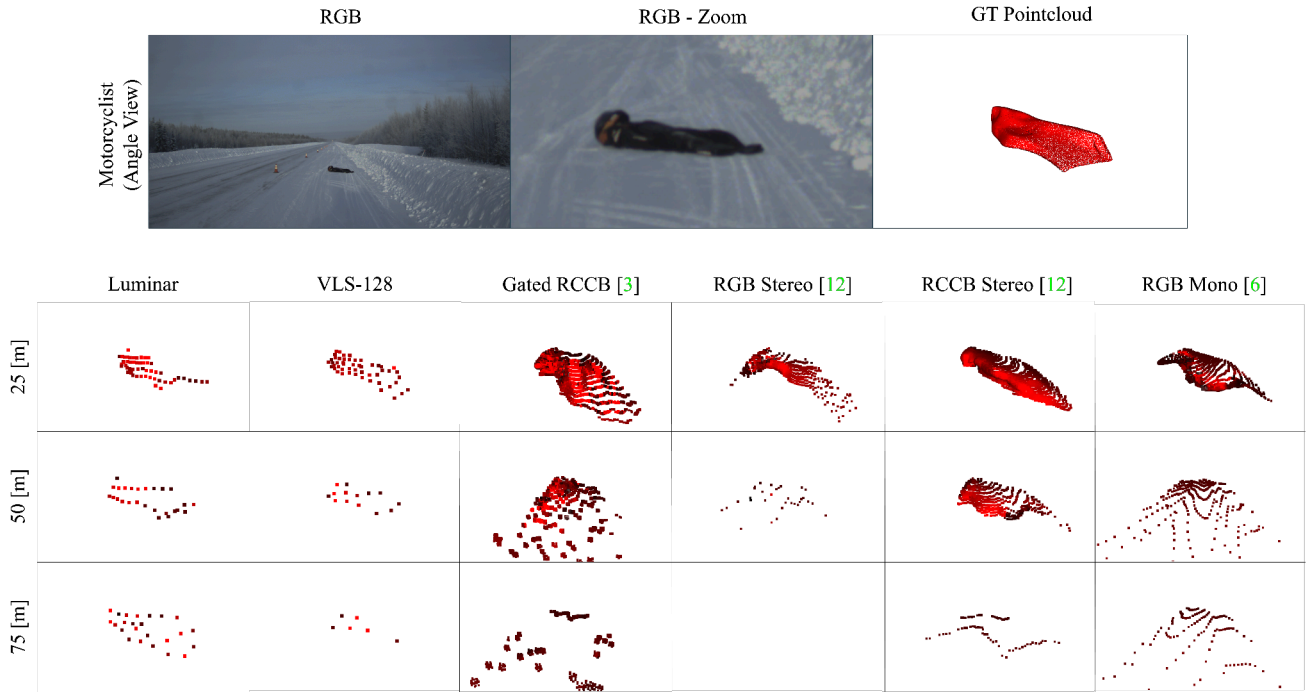

Figure 15. **Winter Recording: Angle View Motorcyclist.** Visual results of lost cargos captured at different distances, with different depth estimation methods, comprising cross spectral Gated RCCB fusion [3], RGB Stereo [12], RCCB Stereo [12] and RGB Monocular [6], and LiDAR sensors from Luminar and Velodyne.

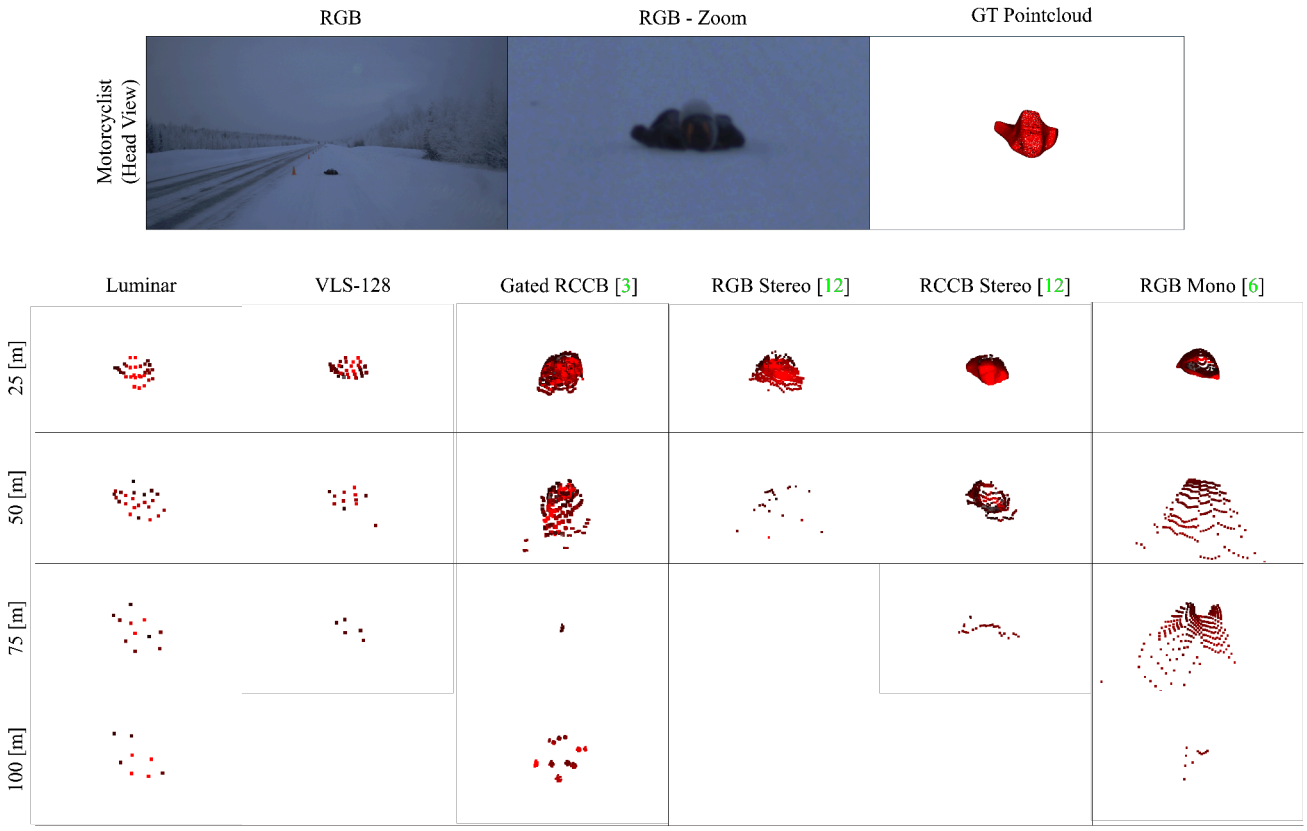

Figure 16. **Winter Recording: Head View Motorcyclist.** Visual results of lost cargos captured at different distances, with different depth estimation methods, comprising cross spectral Gated RCCB fusion [3], RGB Stereo [12], RCCB Stereo [12] and RGB Monocular [6], and LiDAR sensors from Luminar and Velodyne.

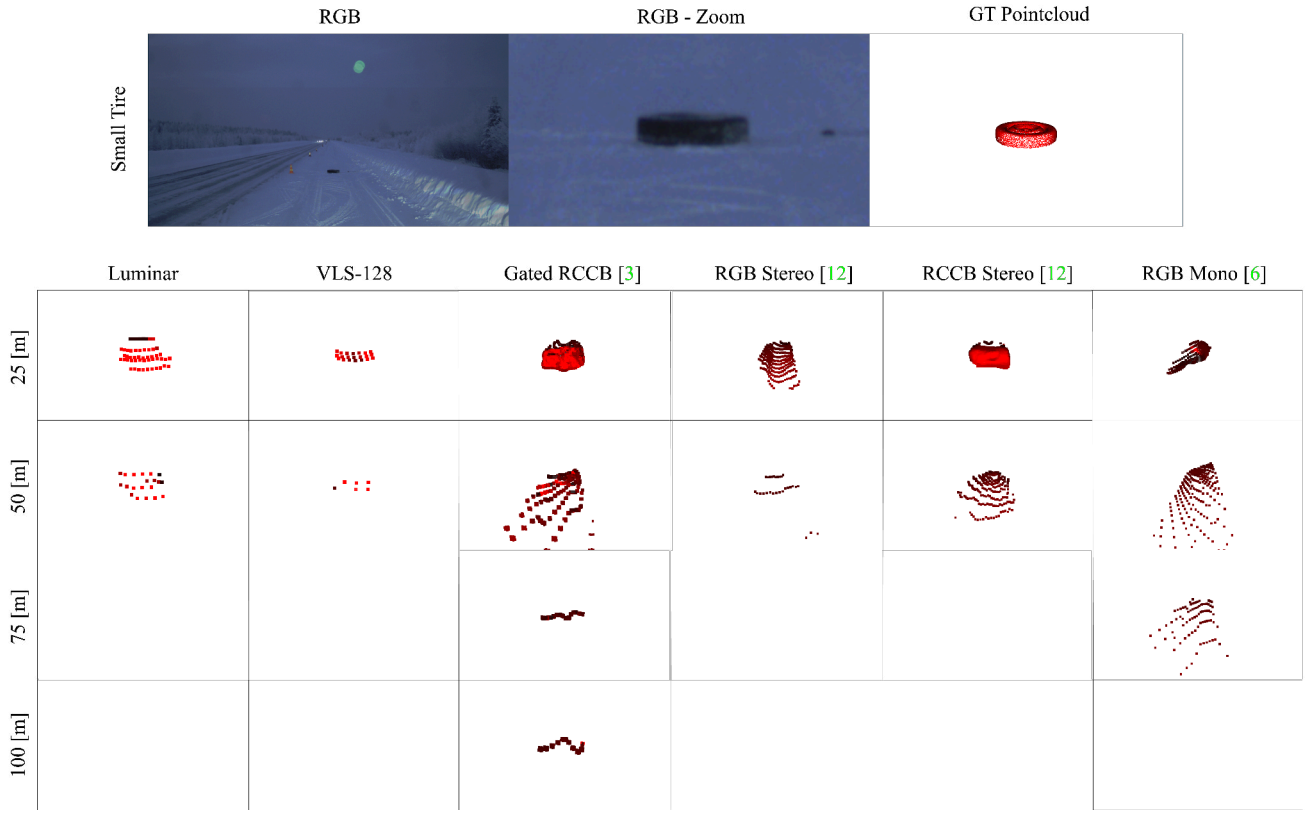

Figure 17. **Winter Recording: Small Tire.** Visual results of lost cargos captured at different distances, with different depth estimation methods, comprising cross spectral Gated RCCB fusion [3], RGB Stereo [12], RCCB Stereo [12] and RGB Monocular [6], and LiDAR sensors from Luminar and Velodyne.