

CARD: A Multi-Modal Automotive Dataset for Dense 3D Reconstruction in Challenging Road Topography

Supplementary Material

1. Sensors Calibration

Accurate road geometry requires consistent calibration of cameras, LiDARs, the IMU, and the wheel contact points. We therefore use a two stage procedure: Camera-LiDAR calibration, followed by calibrating the rig, which is referenced at the IMU, to each wheel.

Camera-LiDAR calibration. For each recording day, we perform three calibration sessions. In each session, we record ~ 500 synchronized LiDAR and stereo image samples while observing a calibration board at different distances, orientations, and positions. Camera intrinsics and extrinsics are estimated from board detections. The front LiDAR is registered to the cameras by fitting a common plane between LiDAR returns and the calibration board in the images. The rear LiDAR and the IMU are then calibrated with respect to the front LiDAR using the batch optimization stage of our two stage IMU-LiDAR fusion pipeline based on MC2SLAM [6]. This step yields individual calibration parameters for both Hesai LiDARs and the pose of the IMU in the front LiDAR frame.

Rig-Wheel calibration with Leica. After calibrating cameras, LiDARs, and IMU, we perform an additional rig and wheel calibration once per recording day with a Leica 3D Disto device. The calibration board is mounted on flat ground in three distinct poses. For each pose, the Leica device measures the three dimensional coordinates of eight marked points on the board, as shown in Fig. 1a. With the vehicle and the board kept fixed, we also measure for each wheel the wheel center and the corresponding ground contact point, as shown in Fig. 1b. In the same session, the stereo cameras observe the board, which allows us to register Leica measurements with the board points reconstructed in the camera frame. The combined set of board and wheel measurements defines a full calibration between the wheels, the vehicle base frame, and the sensor rig. We repeat the Leica session three times to ensure consistency and jointly refine all parameters to minimize reprojection error.

2. Wheel Excitation

Wheel excitation is the trajectory of each wheel’s contact point along the ground. The transformations between the IMU frame and wheel-ground contact points are calibrated for the vehicle in standstill, as described in Sec. 1. Transforming the optimized SLAM trajectory to the wheel contact points yields a rigid-body baseline trajectory. To recover the true wheel excitation, which is induced from the

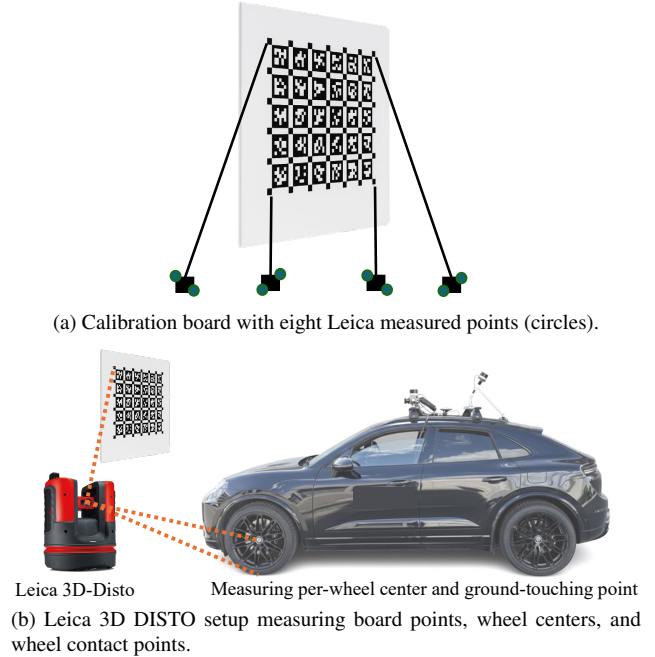


Figure 1. Overview of the rig and wheel calibration procedure. (a) Leica measurements of calibration board points. (b) Combined measurement of board points and wheel geometry, used to link the wheels, vehicle base frame, and sensor rig with sub centimeter accuracy.

terrain profile, we must account for the suspension travel relative to this rigid baseline.

Limited by the tire footprint and suspension travel, a subset of the global point cloud P_w is extracted along the approximate trajectory for wheel w . Projecting these ground points onto the 1D manifold defined by the wheel center trajectory and the up vector, under a parallel-suspension assumption, produces a 1D representation of the ground height relative to the statically calibrated wheel contact point.

For every point $p \in P_w$, we find the two closest adjacent trajectory points at timestamps $i - 1$ and i with accompanying rotation matrices and translation vectors $(R_{i-1}, t_{i-1}), (R_i, t_i) \in (\mathbb{R}^{3 \times 3}, \mathbb{R}^3)$. Let d_i denote the cumulative arclength of trajectory pose i from the start of the wheel trajectory. To interpolate the correct position on the manifold, the relative offset to the first trajectory pose is

computed as

$$\alpha = \left\langle \frac{t_i - t_{i-1}}{\|t_i - t_{i-1}\|}, p - t_{i-1} \right\rangle / \|t_i - t_{i-1}\|. \quad (1)$$

The projected distance of p along the trajectory is then

$$d_p = (1 - \alpha) d_{i-1} + \alpha d_i \quad (2)$$

where d_{i-1} and d_i are the arc lengths from the start of the trajectory for poses $i - 1$ and i , respectively. Finally, the relative height of the point w.r.t. the trajectory is

$$h_p = (1 - \alpha) (R_{i-1}^\top (p - t_{i-1}))_z + \alpha (R_i^\top (p - t_i))_z. \quad (3)$$

To obtain a smooth 1-D excitation profile, we use Ceres to fit a cubic spline in arclength–height space over the set $\{(d_p, h_p) \mid p \in P_w\}$. Applying a robust Tukey loss ensures good outlier suppression, such that spurious measurements from dynamic objects or raised dust are effectively ignored. The final wheel-excitation signal is obtained by evaluating this spline at the trajectory arclengths $\{d_i\}$ and translating the nominal wheel trajectory by the corresponding relative heights.

3. Ground-Truth Aggregation Details

This section gives the full specification of the LiDAR aggregation pipeline used to construct the quasi-dense ground truth. All steps are implemented in our released script and are applied per driving sequence. Unless otherwise stated, all coordinates are in metres.

We denote coordinate frames by $a, b \in \{w, r, c, \ell_f, \ell_r\}$, corresponding to world (w), vehicle/rig (r), camera (c), front LiDAR (ℓ_f), and rear LiDAR (ℓ_r). A 3D point expressed in frame a is written as ${}^a\mathbf{p} \in \mathbb{R}^3$, and its homogeneous counterpart as ${}^a\tilde{\mathbf{p}} = [({}^a\mathbf{p})^\top, 1]^\top$.

A rigid transform ${}^aT_b \in \text{SE}(3)$ maps coordinates from frame b to frame a :

$${}^a\tilde{\mathbf{p}} = {}^aT_b {}^b\tilde{\mathbf{p}}, \quad {}^aT_c = {}^aT_b {}^bT_c. \quad (4)$$

Thus, ${}^wT_r(t)$ is the rig pose in the world at time t , which is the base pose always referenced in the IMU center below the front LiDAR, and rT_s is the time-invariant extrinsic calibration from sensor s to the rig.

Each LiDAR burst b from sensor $s \in \{\ell_f, \ell_r\}$ provides a finite set of points

$$\mathcal{P}_b = \{{}^s\mathbf{p}^{(k)} \in \mathbb{R}^3\}. \quad (5)$$

A. Motion compensation and voxel grid

For burst b from sensor s with end timestamp t_b^{end} , we first form the sensor pose in the world:

$${}^wT_s(b) = {}^wT_r(t_b^{\text{end}}) {}^rT_s. \quad (6)$$

Each LiDAR point ${}^s\mathbf{p}^{(k)}$ is then transformed to the world frame:

$${}^w\mathbf{p}^{(k)} = \Pi({}^wT_s(b) {}^s\tilde{\mathbf{p}}^{(k)}), \quad (7)$$

where Π drops the homogeneous coordinate. For the front LiDAR ($s = \ell_f$), we additionally crop to the forward half-space in the sensor frame to avoid overlap with the rear unit and reduce redundant support.

Voxelization. We discretize the world into a regular grid of cubic voxels of edge length

$$s_{\text{vox}} = 0.10 \text{ m}.$$

The voxel index of a world-frame point ${}^w\mathbf{p}$ is

$$\mathbf{i}({}^w\mathbf{p}) = \lfloor {}^w\mathbf{p} / s_{\text{vox}} \rfloor \in \mathbb{Z}^3. \quad (8)$$

In implementation, \mathbf{i} is stored via a hash for efficiency, but conceptually we operate on the integer index \mathbf{i} .

B. Baseline-adaptive multi-LiDAR voting

Viewpoints per voxel. For each voxel v we keep:

- a list of distinct front-LiDAR sensor origins in world coordinates, $\mathcal{O}_v^F = \{{}^w\mathbf{o}_j\}$,
- a rear-LiDAR hit count n_v^R ,
- an axis-aligned bounding box accumulated over all points assigned to v .

For a front burst with sensor pose ${}^wT_{\ell_f}(b)$, the sensor origin in the world is the translation part ${}^w\mathbf{o}$. We append ${}^w\mathbf{o}$ to \mathcal{O}_v^F only if it is at least

$$d_{\min} = 0.03 \text{ m}$$

away from all existing origins in \mathcal{O}_v^F , enforcing a minimal motion baseline between “distinct” views.

Per-voxel motion baseline. The effective motion baseline for voxel v is the maximum pairwise distance between its front viewpoints:

$$b_v = \begin{cases} \max_{j,\ell} \|{}^w\mathbf{o}_j - {}^w\mathbf{o}_\ell\|_2, & |\mathcal{O}_v^F| \geq 2, \\ 0, & \text{otherwise.} \end{cases} \quad (9)$$

Baseline-dependent vote quotas. Given a vote range $[v_{\min}, v_{\max}]$ and baseline anchors

$$b_n = 0.20 \text{ m}, \quad b_f = 0.90 \text{ m},$$

we define

$$\text{clip}(b) = \min\{\max(b, b_n), b_f\}, \quad (10)$$

$$\hat{v}(b) = v_{\max} - \frac{\text{clip}(b) - b_n}{b_f - b_n} (v_{\max} - v_{\min}), \quad (11)$$

$$\text{req}(b; v_{\min}, v_{\max}) = \text{round}(\hat{v}(b)). \quad (12)$$

Small baselines ($b \approx b_n$) require many confirmations v_{\max} , whereas large baselines ($b \approx b_f$) allow fewer confirmations v_{\min} .

For the front and rear LiDAR we use

$$(v_{\min}^F, v_{\max}^F) = (1, 4), \quad (v_{\min}^R, v_{\max}^R) = (1, 2),$$

and obtain per-voxel quotas

$$n_v^F \geq \text{req}(b_v; v_{\min}^F, v_{\max}^F), \quad (13)$$

$$n_v^R \geq \text{req}(b_v; v_{\min}^R, v_{\max}^R), \quad (14)$$

where $n_v^F = |\mathcal{O}_v^F|$ is the number of distinct front viewpoints that hit v .

Rigid voxel constraint. We also track the axis-aligned bounding box of all world-frame points assigned to voxel v and require that its spatial extent remains small:

$$\|{}^w\mathbf{p}_{\max,v} - {}^w\mathbf{p}_{\min,v}\|_2 \leq \varepsilon_{\text{rigid}}, \quad \varepsilon_{\text{rigid}} = 0.20 \text{ m}. \quad (15)$$

This constraint acts as a temporal consistency filter, rejecting voxels whose content moves or deforms over time.

Global static voxel set. A voxel v belongs to the global static set $\mathcal{V}_{\text{static}}$ if and only if it satisfies all three conditions:

$$v \in \mathcal{V}_{\text{static}} \iff \begin{cases} n_v^F \geq \text{req}(b_v; v_{\min}^F, v_{\max}^F), \\ n_v^R \geq \text{req}(b_v; v_{\min}^R, v_{\max}^R), \\ \|{}^w\mathbf{p}_{\max,v} - {}^w\mathbf{p}_{\min,v}\|_2 \leq \varepsilon_{\text{rigid}}. \end{cases} \quad (16)$$

This static grid is computed once per sequence and cached for all frames.

C. Dynamic pruning with ICP flow

Even with baseline voting and rigidity checks, sometimes moving objects can accumulate into apparently static voxels, such as pedestrians. We therefore add an ICP-based dynamic pruning stage.

LiDAR-scan pairing. Let b_i denote a LiDAR scan with sensor pose ${}^wT_s(b_i)$ and origin ${}^w\mathbf{c}_i$. We form scan pairs (b_i, b_j) such that

$$\|{}^w\mathbf{c}_j - {}^w\mathbf{c}_i\|_2 \geq b_{\text{flow}}, \quad b_{\text{flow}} = 0.50 \text{ m}. \quad (17)$$

Residual-based motion votes. For each pair (b_i, b_j) we align their world-frame point clouds using voxelized generalized ICP, estimating a rigid transform ${}^wT_{w'}^{\text{ICP}}$ that maps the points from burst b_j to the reference burst b_i (both are expressed in w). For every point ${}^w\mathbf{p} \in \mathcal{P}_{b_i}$ we compute the post-alignment residual

$$\delta({}^w\mathbf{p}) = \min_{\mathbf{q} \in \mathcal{P}_{b_j}} \|{}^w\mathbf{p} - {}^wT_{w'}^{\text{ICP}} {}^w\mathbf{q}\|_2. \quad (18)$$

If $\delta({}^w\mathbf{p})$ exceeds

$$\tau_{\text{flow}} = 0.10 \text{ m},$$

we cast a “moved” vote for the voxel v with index $\mathbf{i}({}^w\mathbf{p})$. For voxel v we track m_v (number of moved votes) and c_v (total comparisons). A voxel is flagged as dynamic if

$$m_v \geq V_{\text{dyn}} \quad \text{and} \quad \frac{m_v}{c_v} \geq \eta_{\text{dyn}}, \quad (19)$$

with $V_{\text{dyn}} = 2$ and $\eta_{\text{dyn}} = 0.7$. All such voxels are removed from $\mathcal{V}_{\text{static}}$.

D. Per-frame static point selection

For each camera timestamp t_{cam} , we select LiDAR bursts whose time windows overlap a sensor-specific interval around t_{cam} . We utilize the specific intervals of $[-3, 13]$ s for the front LiDAR and $[0, 30]$ s for the rear LiDAR. Let $\mathcal{B}(t_{\text{cam}})$ denote the selected bursts.

Multi-LiDAR redundancy per frame. Within $\mathcal{B}(t_{\text{cam}})$, a voxel must be confirmed by the two distinct LiDAR units. For voxel v we track the set of sensors

$$\mathcal{S}_v = \{s : \exists b \in \mathcal{B}(t_{\text{cam}}), {}^w\mathbf{p} \in \mathcal{P}_b \text{ with } \mathbf{i}({}^w\mathbf{p}) = v\}, \quad (20)$$

and define the per-frame redundant set

$$\mathcal{V}_{\text{multi}}^{(t)} = \{v : |\mathcal{S}_v| \geq 2\}. \quad (21)$$

Static world cloud for the frame. Stacking all points from selected bursts,

$$\mathcal{P}_{\text{all}}^{(t)} = \bigcup_{b \in \mathcal{B}(t_{\text{cam}})} \mathcal{P}_b, \quad (22)$$

we keep only those whose voxel belongs to both the global static set and the per-frame multi-LiDAR set:

$$\mathcal{P}_{\text{static}}^{(t)} = \{{}^w\mathbf{p} \in \mathcal{P}_{\text{all}}^{(t)} : \mathbf{i}({}^w\mathbf{p}) \in \mathcal{V}_{\text{static}} \cap \mathcal{V}_{\text{multi}}^{(t)}\}. \quad (23)$$

Per-voxel MAD outlier cleaning. Residual dynamic points can still fall into otherwise static voxels. For each voxel v with point set $P_v \subset \mathcal{P}_{\text{static}}^{(t)}$ we compute the centroid

$${}^w\boldsymbol{\mu}_v = \frac{1}{|P_v|} \sum_{{}^w\mathbf{p} \in P_v} {}^w\mathbf{p}, \quad (24)$$

and radii $r_i = \|{}^w\mathbf{p}_i - {}^w\boldsymbol{\mu}_v\|_2$. We form a robust scale estimate (median absolute deviation) and keep only points that satisfy

$$r_i \leq \sigma_{\text{vox}} \text{MAD}_v, \quad \sigma_{\text{vox}} = 1.5. \quad (25)$$

Voxels with fewer than $N_{\text{vox}}^{\min} = 2$ surviving inliers are dropped. This yields the cleaned static world cloud $\tilde{\mathcal{P}}_{\text{static}}^{(t)}$ used for projection.

For each timestamp, we additionally append a “keyframe” front-LiDAR burst, which is the first burst ending after t_{cam} , to guarantee a dense near-range sampling around the evaluation timestamp.

E. Projection and occlusion

Projecting on the left camera. For camera c with intrinsics K_c and extrinsics rT_c we form the world-to-camera transform

$${}^cT_w(t_{\text{cam}}) = ({}^wT_r(t_{\text{cam}}) {}^rT_c)^{-1}. \quad (26)$$

Every static world point is transformed to the camera frame,

$${}^c\mathbf{p} = \Pi({}^cT_w(t_{\text{cam}}) {}^w\tilde{\mathbf{p}}), \quad (27)$$

Ego-vehicle mask. We precompute a 2D binary ego mask $M_{\text{ego}}(u, v)$ in the undistorted camera image space, marking pixels occupied by the hood. Any LiDAR point whose projection falls inside M_{ego} is discarded before monocular consensus and visibility reasoning.

Hidden-point removal. To obtain only camera-visible points, we apply the hidden point removal (HPR) algorithm [5] implemented in Open3D [13]. Denoting by \mathcal{P}_c the current camera-frame cloud, we run two passes of HPR with radii

$$R_1 = \alpha_1 D, \quad R_2 = \alpha_2 D, \quad (28)$$

where D is the diagonal length of the cloud bounding box and $(\alpha_1, \alpha_2) = (100, 210)$. We apply a two-pass HPR strategy. The first pass uses a strict radius (R_1) to aggressively remove static background points that are occluded by dynamic objects, which may appear as single scan lines. In the second pass, we re-introduce the keyframe burst and apply a relaxed radius (R_2). This restores valid sparse points at long range that were over-aggressively culled by the dense near-field points in the first pass, ensuring distant road geometry is preserved.

The resulting camera-frame point cloud for timestamp t_{cam} is saved.

For a small subset of sequences with particularly challenging dynamics, we optionally run a monocular depth consensus step using a pre-trained DepthAnything model [11]. This step is never used to densify or inpaint depth, it acts purely as a conservative rejector on top of the LiDAR-only pipeline above.

Monocular depth estimation (MDE) consensus. Given the undistorted RGB image, DepthAnything predicts a per-pixel depth map $\hat{z}_{\text{DA}}(u, v)$ up to an unknown scale. We align this scale to LiDAR by minimizing a robust one-dimensional least-squares problem over overlapping pixels:

$$\begin{aligned} z_{\text{LiDAR}}(u, v) : \text{LiDAR rasterization}, \quad (29) \\ s^* = \underset{s > 0}{\operatorname{argmin}} \sum_{(u, v) \in \Omega} (z_{\text{LiDAR}}(u, v) - s \hat{z}_{\text{DA}}(u, v))^2, \quad (30) \end{aligned}$$

where Ω contains pixels where both signals are valid and we drop extreme percentiles for robustness. We then form a metrically aligned prediction

$$z_{\text{DA}}(u, v) = s^* \hat{z}_{\text{DA}}(u, v). \quad (31)$$

Per-point consistency test. For each remaining LiDAR point ${}^c\mathbf{p} = (x_c, y_c, z_c)$ we project to pixel coordinates (u, v) and read the aligned DepthAnything depth $z_{\text{DA}}(u, v)$. If no prediction is available at that pixel or if $z_c > r_{\text{max}}$ with

$$r_{\text{max}} = 40 \text{ m},$$

we keep the point unmodified. Otherwise, we compute absolute and relative errors

$$e_{\text{abs}} = |z_{\text{DA}}(u, v) - z_c|, \quad e_{\text{rel}} = \frac{e_{\text{abs}}}{z_c}. \quad (32)$$

A LiDAR point is kept if it satisfies the joint tolerance

$$e_{\text{rel}} \leq \tau_{\text{rel}} \quad \text{and} \quad e_{\text{abs}} \leq \max(\tau_{\text{abs}}, \tau_{\text{rel}} z_c), \quad (33)$$

with

$$\tau_{\text{rel}} = 0.14, \quad \tau_{\text{abs}} = 0.10 \text{ m}.$$

Points that violate this condition are removed and treated as residual movers. Importantly, we never add monocular points to the ground truth, MDE is used exclusively as a sanity check on the LiDAR geometry and only on demand. Finally, we present the qualitative output of our ground-truth generation pipeline in Fig. 2. The figure visualizes the sensor setup, the intermediate point cloud aggregation, and the final dense geometry projected into the image frame for annotation and evaluation.

F. Numerical hyper-parameters

Table 1 summarizes the numerical settings used across all sequences for the pipeline described above.

4. Annotation Pipeline

Labeling road-surface irregularities from images is inherently ambiguous. In practice, we observe a variety of structures: speed bumps and humps with varying profiles and

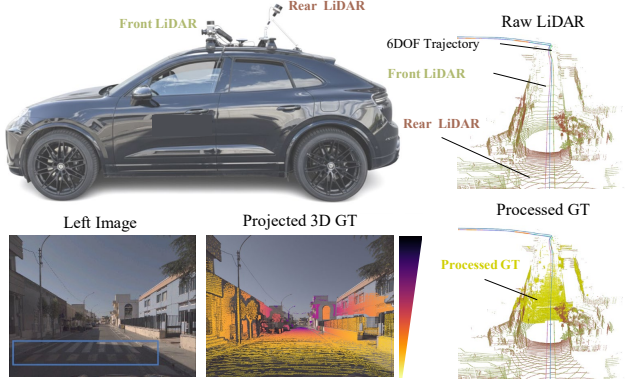


Figure 2. Qualitative Overview of the Ground-Truth Generation Pipeline. The ego-vehicle (top left) captures data using front and rear LiDARs. The right column compares the Raw LiDAR along the 6-DoF trajectory (top) against the final Processed GT (bottom), showing the densification effect. The bottom row illustrates the downstream utility: a Left Image with a speed bump annotation and its corresponding Projected 3D GT depth map.

Stage	Parameter	Value
Voxel grid	Voxel size s_{vox}	0.10 m
	Min. front view spacing d_{min}	0.03 m
	Rigidity radius $\varepsilon_{\text{rigid}}$	0.20 m
Baseline voting	Near / far baseline b_n, b_f	0.20/0.90 m
	Front votes $v_{\text{min}}^F, v_{\text{max}}^F$	1/4
	Rear votes $v_{\text{min}}^R, v_{\text{max}}^R$	1/2
Per-frame selection	Min. LiDARs per voxel $ \mathcal{S}_v $	≥ 2
ICP pruning	Flow baseline b_{flow}	0.50 m
	Residual threshold τ_{flow}	0.10 m
	Moved votes V_{dyn}	2
	Moved fraction η_{dyn}	0.7
Voxel MAD	MAD multiplier σ_{vox}	1.5
	Min. inliers $N_{\text{vox}}^{\text{min}}$	2
Visibility	HPR multiplier (1st / 2nd pass)	100/210
DepthAnything consensus	Rel. error τ_{rel}	0.15
	Abs. floor τ_{abs}	0.10 m
	Max. range r_{max}	40 m

Table 1. Numerical settings for the LiDAR aggregation pipeline. All values are fixed across sequences. There are a few examples which required manual filtering by controlling the τ_{rel} of the MDE

markings, elevated circular asphalt patches, and, on the other hand, potholes, cracks, subsidence, and other local depressions. Our main goal in CARD is not fine-grained categorization, but providing bounding boxes that enable height-based evaluation on regions where the road topography deviates from a locally planar surface.

Moreover, the perception of what constitutes a bump versus a mild elevation, or a pothole versus a shallow depression, is subjective and depends on context and surrounding texture. To reduce label noise and keep the annotation

protocol tractable, we adopt a coarse but robust strategy with two image-based classes. The positive class represents speed bumps and other elevated structures from the road plane, excluding sidewalks or large structural irregularities. On the other hand, the negative class represents potholes and local depressions. These are annotated as bounding boxes at the image level to support per-object height evaluation.

In addition, CARD contains off-road segments, such as gravel, grass, or unpaved construction areas, where the entire drivable surface is intrinsically irregular and does not admit a stable reference plane. For such scenes, we provide a per-sequence off-road flag rather than per-image bounding boxes, since there is no single, well-defined defect region.

4.1. Labeling Strategy

To ensure high-quality ground truth and facilitate a targeted analysis of these road features, we employ a semi-automated pipeline that utilizes a dedicated object detector to verify and refine the irregularities across the dataset.

4.2. Semi-Automated Dataset Annotation

We employ a strategy based on YOLOv8 [4], combining a manually annotated seed set with large-scale model-assisted labeling.

- 1. Initial frame selection.** We first perform a comprehensive manual review of all frames to identify images containing at least one visible road irregularity, positive or negative.
- 2. Manual annotation subset.** From this subset, we randomly sample 40% for manual annotation. Using [3], annotators draw bounding boxes for all clearly visible potholes and speed bumps in this subset.
- 3. Internal consistency split.** The manually annotated subset is randomly partitioned into training (70%), validation (20%), and test (10%) splits. We use this split to train the verification model and evaluate the consistency of our labeling criteria.
- 4. Model-assisted labeling.** We train a YOLOv8 detector on the annotated training set and run it on the remaining 60% of the master list to obtain preliminary bounding boxes. All predictions are subsequently reviewed, and a custom script is used to efficiently correct errors. This semi-automated loop substantially accelerates the annotation process while ensuring the final labels adhere to the definitions established in the seed set.

4.3. Training Protocol

We fine-tune YOLOv8 [4] on our two-class road-irregularity labels. The model is trained for 150 epochs with an input resolution of 1024×1024 pixels on an NVIDIA RTX 6000 Ada Generation GPU. Training completes in approximately 8.1 hours.

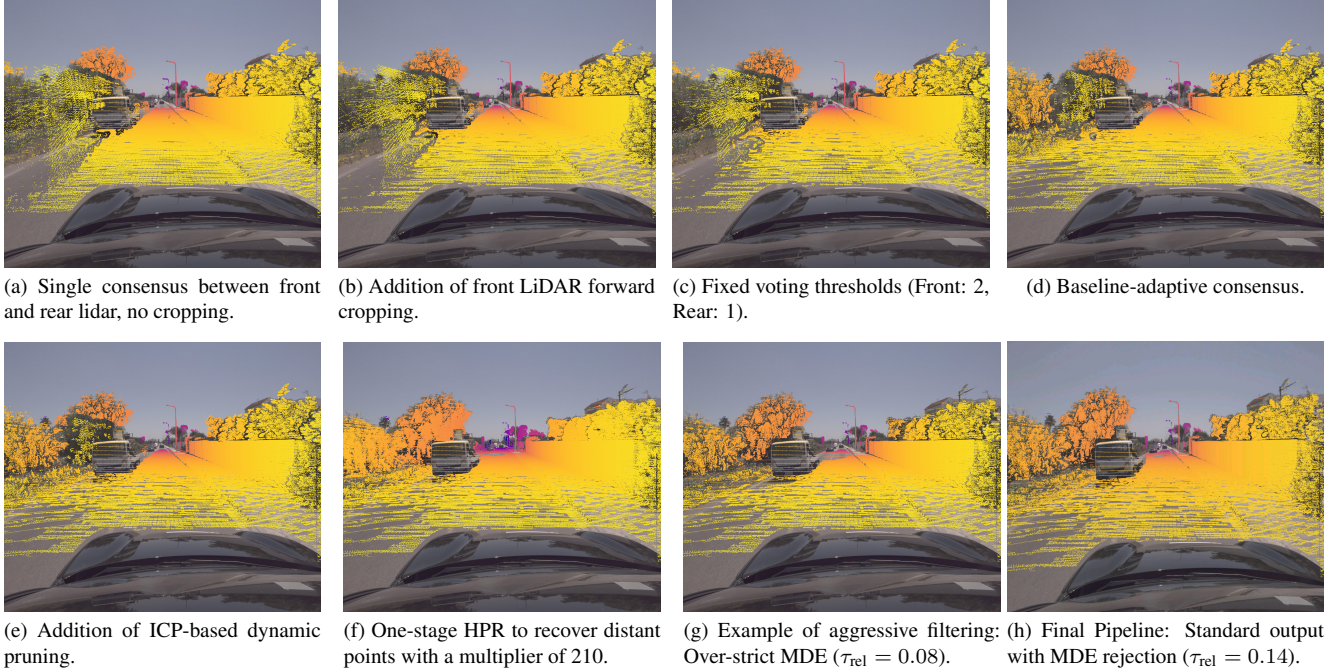


Figure 3. Qualitative ablation of the ground-truth aggregation pipeline. (a)–(c) show the progressive cleanup using cropping and adaptive-voting thresholds. (d) introduces our baseline-adaptive consensus to handle variable point density. (e) adds ICP-based dynamic pruning to remove ghosting from moving objects. (f) demonstrates the two-stage Hidden Point Removal (HPR), which successfully recovers sparse valid points at long range. (g) illustrates a failure case where overly strict monocular consistency checks erode valid road geometry in the distance. (h) shows the final configuration used in our main experiments.

4.4. Annotation Consistency Analysis

To validate the quality and learnability of our ground truth, we evaluate the detector on the validation and test splits using standard metrics. High detection scores in this context indicate that the visual definitions of “positive” and “negative” irregularities are distinct and consistently applied across the dataset.

The results on the held-out test set (Table 2b) demonstrate a high degree of label consistency. The model achieves an mAP@.50 of 0.994 for the positive class, confirming that elevated structures (speed bumps) are annotated with high precision and low ambiguity.

For the negative class (potholes), the model achieves an mAP@.50 of 0.876. This performance confirms that the distinct visual features of potholes are learnable, the slight gap compared to the positive class reflects the inherent subjectivity in defining the boundary between a pothole and a minor depression, as noted in our annotation protocol.

5. Quantitative Results

All quantitative evaluations presented in this supplementary material are computed using the largest version of each baseline model. We report metrics for both depth and height estimation across all scenes, as well as specific per-

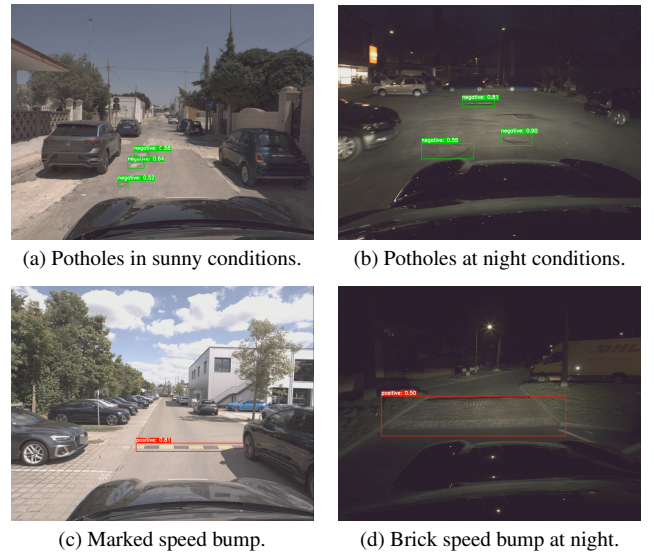


Figure 4. Qualitative YOLOv8 detections for potholes (negative class) and speed bumps (positive class) under diverse illumination conditions.

bounding-box evaluations.

All metrics are computed over a valid pixel set \mathcal{V} . Let

Class	Instances	Precision	Recall	mAP@.50	mAP@.50-.95
All classes	1,725	0.932	0.903	0.952	0.804
Negative	970	0.897	0.839	0.919	0.793
Positive	755	0.968	0.967	0.986	0.814

(a) Validation set.

Class	Instances	Precision	Recall	mAP@.50	mAP@.50-.95
All classes	870	0.878	0.912	0.935	0.793
Negative	521	0.793	0.833	0.876	0.755
Positive	349	0.964	0.991	0.994	0.831

(b) Test set.

Table 2. Verification of label consistency: YOLOv8 detection metrics on (a) validation and (b) test splits.

d_i and \hat{d}_i denote the ground truth and predicted depth for pixel i , respectively. A pixel i is considered valid ($i \in \mathcal{V}$) if and only if both d_i and \hat{d}_i fall within the sensor range $[d_{\min}, d_{\max}]$, where $d_{\min} = 0.1$ m and $d_{\max} = 80.0$ m.

5.1. Depth Evaluation

We employ standard depth estimation metrics:

- **Absolute Relative Error (AbsRel):**

$$\text{AbsRel} = \frac{1}{|\mathcal{V}|} \sum_{i \in \mathcal{V}} \frac{|d_i - \hat{d}_i|}{d_i} \quad (34)$$

- **Squared Relative Error (SqRel):**

$$\text{SqRel} = \frac{1}{|\mathcal{V}|} \sum_{i \in \mathcal{V}} \frac{\|d_i - \hat{d}_i\|^2}{d_i} \quad (35)$$

- **Root Mean Squared Error (RMSE):**

$$\text{RMSE} = \sqrt{\frac{1}{|\mathcal{V}|} \sum_{i \in \mathcal{V}} \|d_i - \hat{d}_i\|^2} \quad (36)$$

- **RMSE (log):**

$$\text{RMSE}_{\log} = \sqrt{\frac{1}{|\mathcal{V}|} \sum_{i \in \mathcal{V}} \|\log d_i - \log \hat{d}_i\|^2} \quad (37)$$

- **Threshold Accuracy (δ_n):** The percentage of pixels where the ratio between prediction and ground truth is below a threshold:

$$\delta_n = \frac{1}{|\mathcal{V}|} \sum_{i \in \mathcal{V}} \mathbb{I} \left(\max \left(\frac{d_i}{\hat{d}_i}, \frac{\hat{d}_i}{d_i} \right) < 1.25^n \right) \quad (38)$$

We report results for $n \in \{1, 2, 3\}$.

5.2. Height Evaluation Metrics

To evaluate road irregularities, we convert absolute depth maps into height maps relative to the road surface for the current vehicle position. For each timestamp, we recover the 3D positions of the four wheels' ground contact points from the calibrated sensor rig and wheel-excitation signals (see Secs. 1 and 2). In the vehicle frame, we construct the local road plane as the unique plane spanned by these four contact points, which serves as a physically meaningful reference surface.

Both ground-truth and predicted depth maps are back-projected into 3D and transformed into this common vehicle frame. For each 3D point, we compute its signed distance h to the corresponding per-frame road plane, positive values indicate points above the road surface. This yields dense height maps that directly encode deviations of the road geometry with respect to the nominal road surface. As evaluation metrics, we report:

- **Height Absolute Difference (AbsDiff):** The mean absolute difference between the predicted and ground truth height:

$$\text{AbsDiff} = \frac{1}{|\mathcal{V}|} \sum_{i \in \mathcal{V}} |h_i - \hat{h}_i| \quad (39)$$

- **Height RMSE:** The root mean squared error of the height deviation:

$$\text{RMSE}_h = \sqrt{\frac{1}{|\mathcal{V}|} \sum_{i \in \mathcal{V}} (h_i - \hat{h}_i)^2} \quad (40)$$

- **Height Accuracy ($\delta@ \tau$):** The fraction of pixels where the absolute height error is within a specific threshold τ :

$$\delta@ \tau = \frac{1}{|\mathcal{V}|} \sum_{i \in \mathcal{V}} \mathbb{I} (|h_i - \hat{h}_i| < \tau) \quad (41)$$

We evaluate at thresholds $\tau = 10$ cm and $\tau = 25$ cm.

5.3. Fine-tuning Analysis on CARD

To probe whether the limitations of monocular methods are primarily due to domain shift, we fine-tune UniDepth-V2-L [7] on the CARD training split. We use our dense depth maps as metric-depth supervision and mask valid points within $[0.05, 80]$ m. Starting from the checkpoint, we freeze the encoder backbone and optimize only the decoder parameters. The model is trained for 10 epochs with an L1 depth loss and AdamW optimizer (learning rate 1×10^{-5} , weight decay 10^{-2}) with a cosine annealing schedule, using mixed-precision training with batch size 4. We select the checkpoint with the lowest validation loss on held-out CARD sequences.

Table 3 summarizes the effect of this CARD-specific fine-tuning. We observe consistent improvements in global scene metrics, as an example full-scene RMSE from 1.825 to 1.749, AbsRel from 0.046 to 0.042, indicating that our supervision benefits overall depth quality. However, on local bounding boxes around road irregularities, the gains are marginal. These structures induce only centimetre-scale depth changes, so even perceptually relevant differences correspond to small errors in depth space and remain largely negligible to global metrics. This suggests that resolving fine-grained road topography requires architectures and training strategies explicitly tailored to road geometry, beyond shallow decoder-only fine-tuning. However, as demonstrated in the main paper, incorporating the affine-invariant losses from MoGe [8, 9] yields substantial improvements in the topography-specific metrics. Thus, we leave the development of such specialized architectures for road geometry as future work.

6. Qualitative Results

Figure 6 illustrates the visual diversity of the CARD dataset through random image crops, categorizing scenes into positive irregularities, negative irregularities, off-road terrain, and general road contexts.

To highlight the quality of our data, Fig. 5 presents examples of our densified ground truth. As shown, the generated height maps are dense and clearly resolve fine-grained road features, such as speed bumps. Crucially, the aggregation pipeline effectively mitigates artifacts from dynamic objects, resulting in clean representations of the static road surface.

Additionally, Figs. 7 to 10 present specific qualitative comparisons of surface reconstruction. In these examples, we visualize the input context, ground truth height, and predictions from monocular baselines alongside the stereo reference. The error maps indicate that monocular estimators often oversmooth high-frequency geometry, leading to height estimation errors on potholes and bumps, whereas the stereo baseline better preserves these structural details.

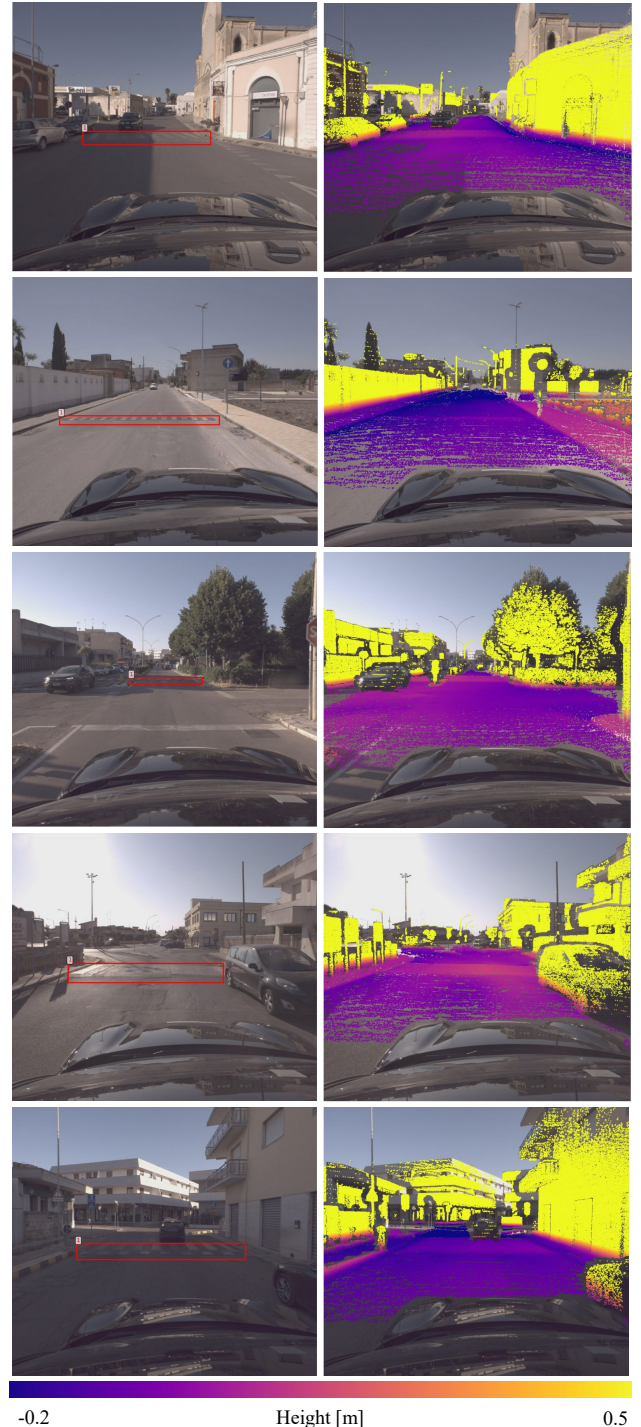


Figure 5. Projected Densified Ground Truth. We show input RGB images (left) alongside their corresponding projected dense height maps (right).

Method	AbsRel ↓		SqRel ↓		RMSE ↓		RMSElog ↓		δ_1 ↑		δ_2 ↑		δ_3 ↑	
	F	B	F	B	F	B	F	B	F	B	F	B	F	B
UniDepthV2-L [†]	0.046	0.029	0.204	0.022	1.825	0.382	0.074	0.032	0.981	0.992	0.995	0.993	0.999	0.993
UniDepthV2-L(Fine-tuned)	0.042	0.027	0.183	0.020	1.749	0.355	0.069	0.030	0.989	0.991	0.997	0.993	0.998	0.993

Table 3. Impact of Fine-tuning on UniDepthV2 by L1 depth loss. Comparisons of metrics before and after fine-tuning on the CARD training set. We report results for both the Full and per-box metrics. While fine-tuning improves global scene metrics (Full), the per-box metrics targeting road irregularities show diminishing returns, indicating that standard fine-tuning is insufficient to fully resolve local topography. Both models utilized median-scaling in testing. [†] denotes median-scaled metrics.

Method	AbsRel ↓		SqRel ↓		RMSE ↓		RMSElog ↓		δ_1 ↑		δ_2 ↑		δ_3 ↑	
	F	B	F	B	F	B	F	B	F	B	F	B	F	B
DAV2 [12]	1.537	1.456	29.122	20.811	20.264	14.493	0.926	0.895	0.008	0.001	0.025	0.003	0.085	0.017
DepthPro [1]	0.317	0.288	2.779	1.058	6.086	3.080	0.390	0.355	0.280	0.223	0.722	0.744	0.907	0.944
MoGeV2 [9]	0.108	0.092	0.349	0.143	2.440	1.013	0.123	0.089	0.919	0.963	0.993	0.993	0.998	0.993
Metric3DV2 [2]	0.665	0.664	5.812	4.383	9.493	6.589	0.511	0.506	0.017	0.001	0.283	0.236	0.935	0.972
UniDepthV2 [7]	0.078	0.062	0.314	0.074	2.242	0.764	0.099	0.062	0.975	0.992	0.995	0.993	0.998	0.993
FS (stereo) [10]	0.040	0.014	0.411	0.006	2.162	0.185	0.088	0.016	0.976	0.994	0.989	0.994	0.994	0.994

(a) Depth (without median-scaling). Raw predictions. FS [10] is a stereo model.

Method	AbsRel ↓		SqRel ↓		RMSE ↓		RMSElog ↓		δ_1 ↑		δ_2 ↑		δ_3 ↑	
	F	B	F	B	F	B	F	B	F	B	F	B	F	B
DAV2 [†] [12]	0.096	0.055	0.548	0.084	4.107	0.683	0.154	0.060	0.895	0.976	0.969	<u>0.989</u>	<u>0.990</u>	<u>0.992</u>
DepthPro [†] [1]	0.100	0.045	0.830	0.075	3.270	0.602	0.142	0.050	0.908	0.974	0.971	<u>0.989</u>	0.986	0.993
MoGeV2 [†] [9]	0.045	0.027	<u>0.177</u>	0.022	1.807	0.337	0.074	0.029	<u>0.980</u>	0.989	<u>0.996</u>	0.993	0.999	0.993
Metric3DV2 [†] [2]	0.060	0.039	0.165	<u>0.034</u>	1.836	0.474	0.085	0.042	0.976	<u>0.990</u>	0.997	0.993	0.999	0.993
UniDepthV2 [†] [7]	<u>0.046</u>	<u>0.029</u>	0.204	0.022	<u>1.825</u>	<u>0.382</u>	0.074	<u>0.032</u>	0.981	0.992	0.995	0.993	0.999	0.993
FS (stereo) [10]	0.036	0.009	0.401	0.003	2.126	0.137	0.086	0.011	0.977	0.994	0.989	0.994	0.994	0.994

(b) Depth (with per-image scaling). [†]Monocular models median-scaled.

Method	AbsDiff ↓		RMSE ↓		$\delta@10$ cm ↑		$\delta@25$ cm ↑	
	F	B	F	B	F	B	F	B
DAV2 [†] [12]	0.181	0.106	<u>0.371</u>	0.113	0.578	0.620	0.836	0.929
DepthPro [†] [1]	0.249	0.086	0.817	0.092	0.630	0.726	0.839	0.944
MoGeV2 [†] [9]	0.119	0.051	0.633	0.056	<u>0.801</u>	0.893	0.936	<u>0.987</u>
Metric3DV2 [†] [2]	0.117	0.074	0.356	0.081	0.675	0.754	0.911	0.969
UniDepthV2 [7]	0.117	<u>0.055</u>	0.456	<u>0.061</u>	0.807	<u>0.860</u>	0.948	0.989
FS (stereo) [10]	0.169	0.017	1.508	0.021	0.907	0.992	0.958	0.999

(c) Height (with scale). We report only scaled height metrics (Full vs Boxes). [†]Monocular values use the same per-image scaling as in depth. FS is stereo.

Table 4. Comprehensive CARD benchmark. Each subtable reports *Full* (F) and *Boxes* (B) side by side.



(a) Positive Irregularities: Speed bumps, raised manholes, and plateau bumps.



(b) Negative Irregularities: Potholes, subsidence, and road cracks.



(c) Off-Road Terrain: Unpaved surfaces, gravel, and unstructured geometry.



(d) General Road Scenes: diverse pavement types and nominal driving conditions.

Figure 6. Diversity of the CARD Dataset. We display random image crops sampled across the dataset to illustrate variability in geometry and environment. The samples cover (a) positive vertical obstacles, (b) negative surface defects, (c) unstructured off-road terrain, and (d) general road contexts.

Positive Example

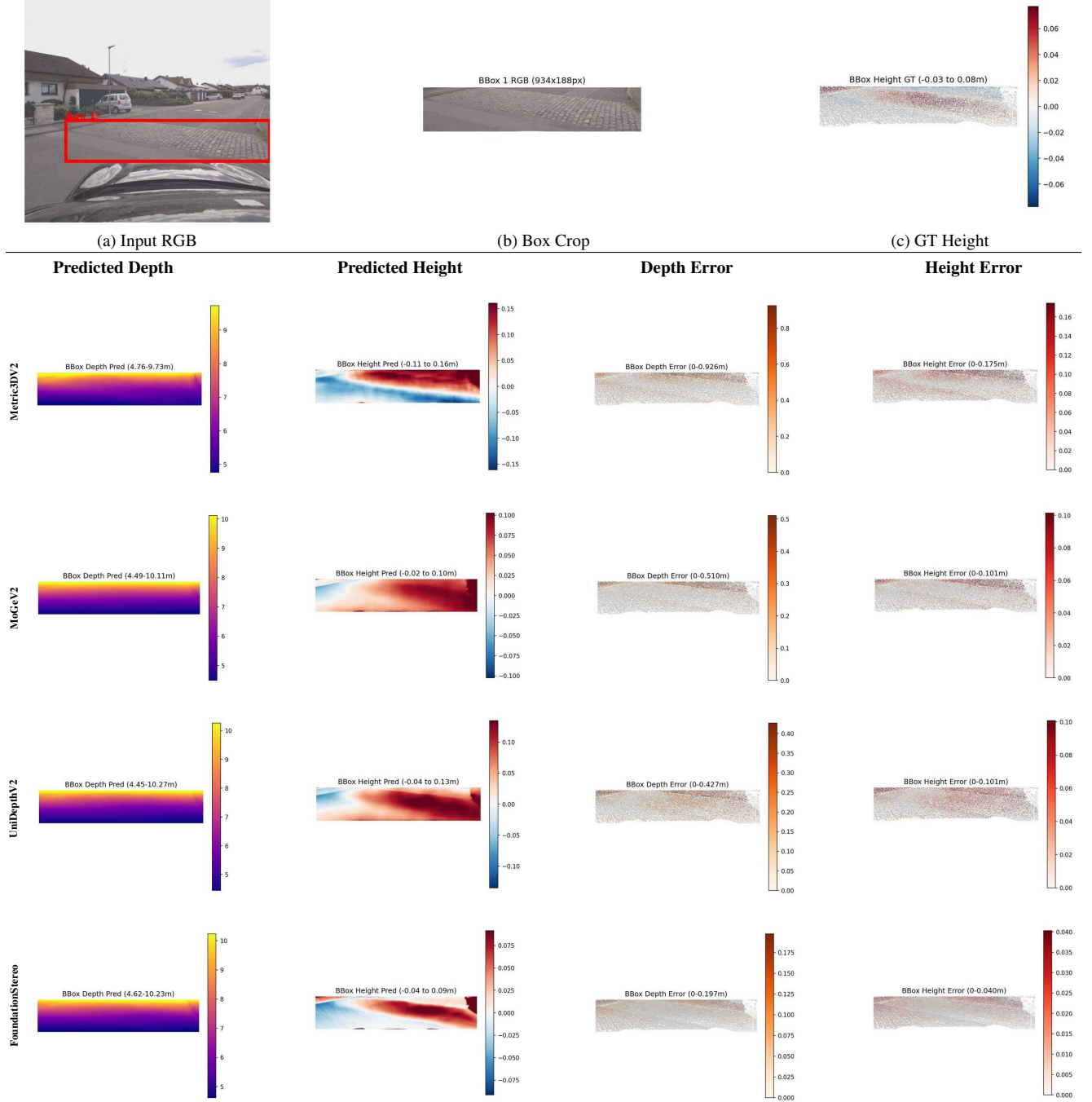


Figure 7. **Qualitative Comparison of a positive example of a speed bump.** Top: Input context and GT geometry. Bottom: Model predictions. Monocular baselines [2, 7, 9] (Rows 1–3). FoundationStereo [10] (Bottom Row).

Negative Example

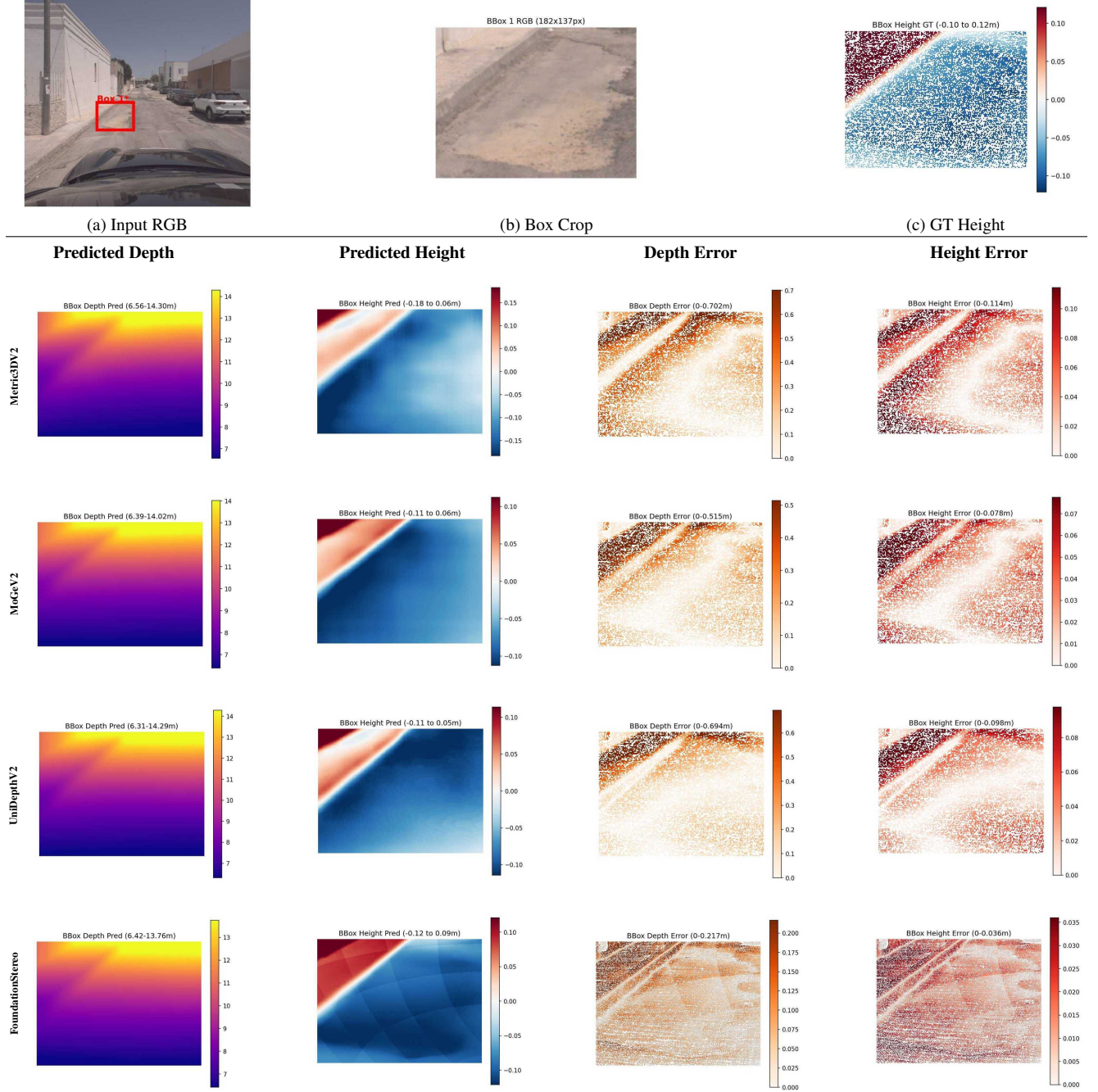


Figure 8. **Qualitative Comparison of a pothole example.** Top: Input context and GT geometry. Bottom: Model predictions. Monocular baselines [2, 7, 9] (Rows 1–3). FoundationStereo [10] (Bottom Row) accurately recovers the geometry.

Positive Road Irregularity

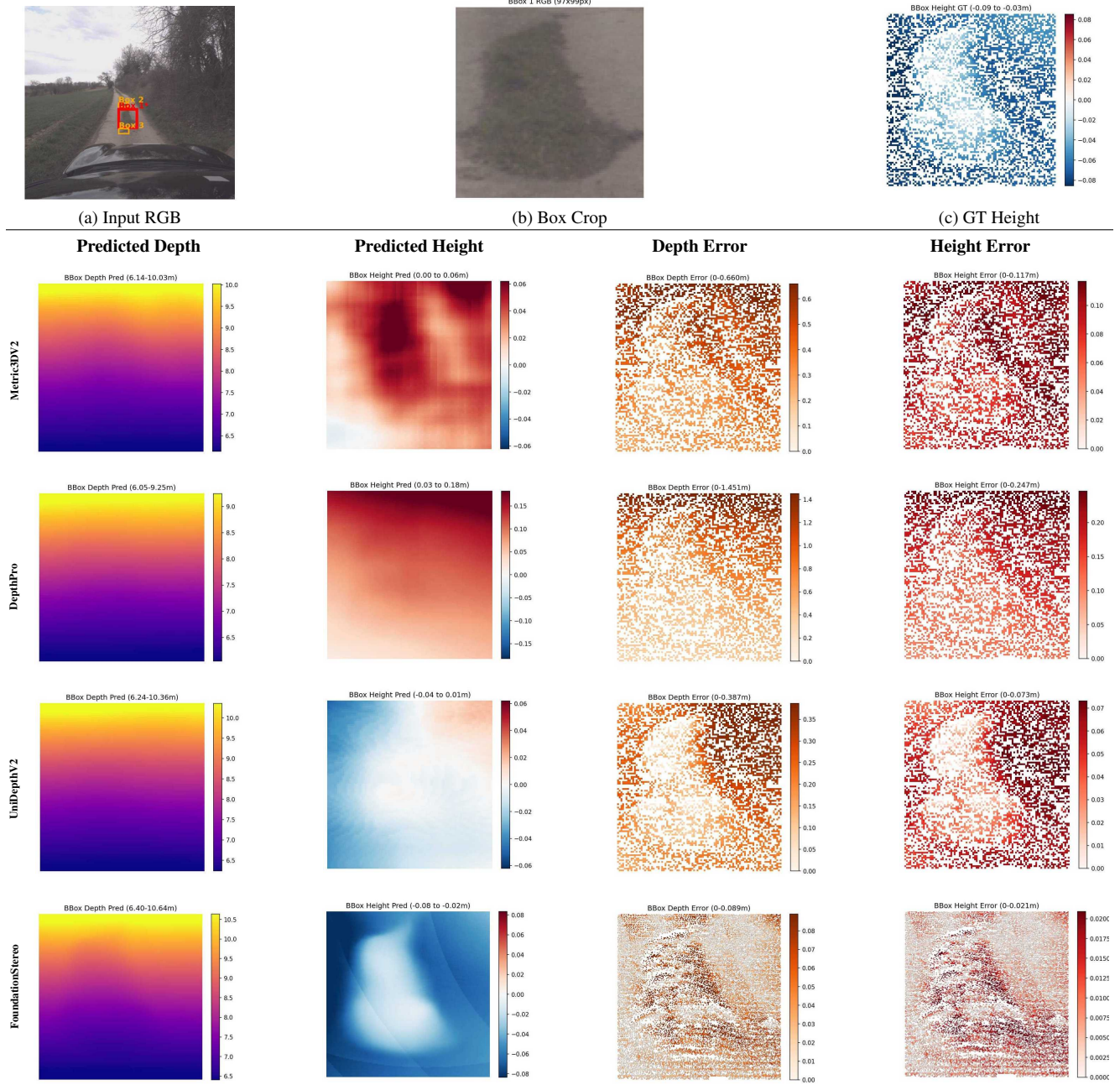


Figure 9. **Qualitative Comparison of a positive road irregularity.** Top: Input context and GT geometry. Bottom: Model predictions. Monocular baselines [1, 2, 7] (Rows 1–3). FoundationStereo [10] (Bottom Row).

Positive Road Irregularity

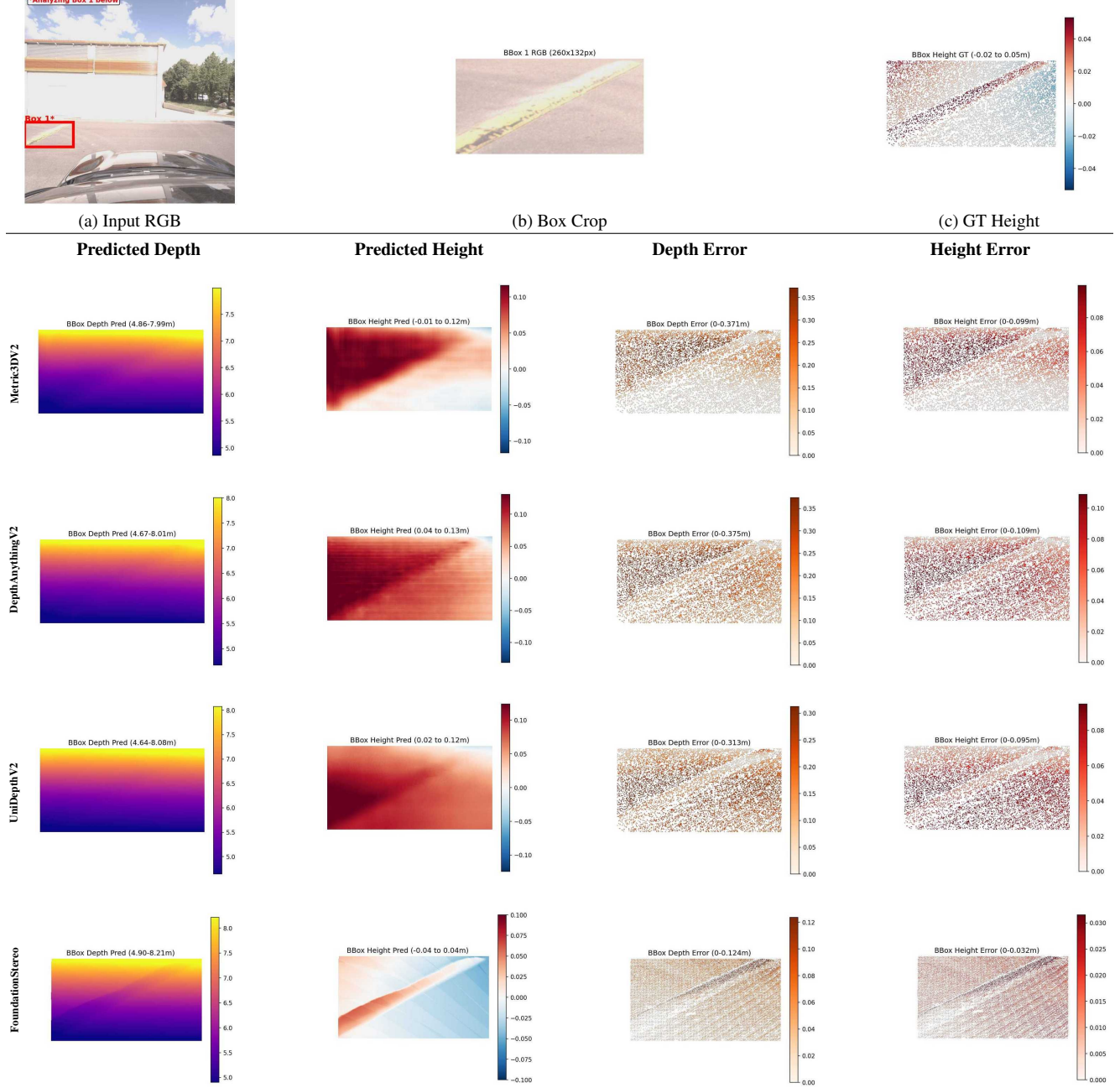


Figure 10. Qualitative Comparison of a positive road irregularity. Top: Input context and GT geometry. Bottom: Model predictions. Monocular baselines [2, 7, 12] (Rows 1–3). FoundationStereo [10] (Bottom Row).

References

- [1] Alexey Bochkovskiy, Amaël Delaunoy, Hugo Germain, Marcel Santos, Yichao Zhou, Stephan Richter, and Vladlen Koltun. Depth pro: Sharp monocular metric depth in less than a second. In *The Thirteenth International Conference on Learning Representations*, 2025. [9](#), [13](#)
- [2] 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. [9](#), [11](#), [12](#), [13](#), [14](#)
- [3] HumanSignal. labeling. GitHub repository. Archived 2024-02-29. Original work by Tzutalin. [5](#)
- [4] Glenn Jocher, Jing Qiu, and Ayush Chaurasia. Ultralytics YOLOv8, 2025. Version 8.3.105. [5](#)
- [5] Sagi Katz, Ayellet Tal, and Ronen Basri. Direct visibility of point sets. In *ACM SIGGRAPH 2007 Papers*, page 24–es, New York, NY, USA, 2007. Association for Computing Machinery. [4](#)
- [6] Frank Neuhaus, Tilman Koß, Robert Kohnen, and Dietrich Paulus. Mc2slam: Real-time inertial lidar odometry using two-scan motion compensation. In *German Conference on Pattern Recognition*, pages 60–72. Springer, 2018. [1](#)
- [7] Luigi Piccinelli, Christos Sakaridis, Yung-Hsu Yang, Mattia Segu, Siyuan Li, Wim Abbeloos, and Luc Van Gool. Unidepthv2: Universal monocular metric depth estimation made simpler. *arXiv preprint arXiv:2502.20110*, 2025. [8](#), [9](#), [11](#), [12](#), [13](#), [14](#)
- [8] Ruicheng Wang, Sicheng Xu, Cassie Dai, Jianfeng Xiang, Yu Deng, Xin Tong, and Jiaolong Yang. Moge: Unlocking accurate monocular geometry estimation for open-domain images with optimal training supervision. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025. [8](#)
- [9] Ruicheng Wang, Sicheng Xu, Yue Dong, Yu Deng, Jianfeng Xiang, Zelong Lv, Guangzhong Sun, Xin Tong, and Jiaolong Yang. Moge-2: Accurate monocular geometry with metric scale and sharp details. *arXiv preprint arXiv:2507.02546*, 2025. [8](#), [9](#), [11](#), [12](#)
- [10] Bowen Wen, Matthew Trepte, Joseph Aribido, Jan Kautz, Orazio Gallo, and Stan Birchfield. Foundationstereo: Zero-shot stereo matching. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025. [9](#), [11](#), [12](#), [13](#), [14](#)
- [11] Lihe Yang, Bingyi Kang, Zilong Huang, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything: Unleashing the power of large-scale unlabeled data. In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, 2024. [4](#)
- [12] Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything v2. *Advances in Neural Information Processing Systems*, 37, 2024. [9](#), [14](#)
- [13] Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. Open3d: A modern library for 3d data processing. *arXiv preprint arXiv:1801.09847*, 2018. [4](#)