RePLAy: Remove Projective LiDAR Depthmap Artifacts via Exploiting Epipolar Geometry

Shengjie Zhu*, Girish Chandar Ganesan*, Abhinav Kumar, and Xiaoming Liu

Department of Computer Science and Engineering,
Michigan State University, East Lansing, MI, USA, 48824
[zhusheng, ganesang, kumarab6, liuxm]@msu.edu
https://shngjz.github.io/RePLAy/

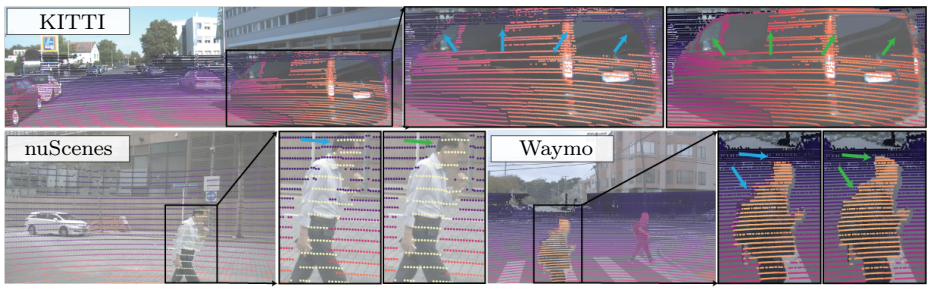


Fig. 1: Remove Projective LiDAR Depthmap Artifacts. The LiDAR projected depthmap has misalignment with the RGB camera. The background scans highlighted by blue arrows incorrectly overlay the foreground such as cars and pedestrians. The artifacts persist unattended in most AV datasets including KITTI-360 [41], nuScenes [10], Waymo [56], and DDAD [24]. We propose an easy-to-use, parameter-free, and analytical solution to remove the projective artifacts, shown by green arrows.

Abstract. 3D sensing is a fundamental task for Autonomous Vehicles. Its deployment often relies on aligned RGB cameras and LiDAR. Despite meticulous synchronization and calibration, systematic misalignment persists in LiDAR projected depthmap. This is due to the physical baseline distance between the two sensors. The artifact is often reflected as background LiDAR incorrectly projected onto the foreground, such as cars and pedestrians. The KITTI dataset uses stereo cameras as a heuristic solution to remove artifacts. However most AV datasets, including nuScenes, Waymo, and DDAD, lack stereo images, making the KITTI solution inapplicable. We propose *RePLAy*, a parameter-free analytical solution to remove the projective artifacts. We construct a binocular vision system between a hypothesized virtual LiDAR camera and the RGB camera. We then remove the projective artifacts by determining the epipolar occlusion with the proposed analytical solution. We show unanimous improvement in the State-of-The-Art (SoTA) monocular depth estimators and 3D object detectors with the artifacts-free depthmaps.

Keywords: LiDAR· Autonomous Driving · Epipolar Geometry

* Equal Contributions

(a) Autonomous Vehicles [10] **(b)** Robotics Machines [14] **(c)** Headset [26]

Fig. 2: LiDAR-RGB sensor sets have a physical baseline distance between the two sensors, resulting in Fig. 1 projective artifacts. **(a)** The background scanned by LiDAR incorrectly overlays on the foreground pedestrian observed by the RGB camera.

1 Introduction

LiDAR device scans the surrounding environment via emitting laser pulses into the 3D space. Its applications span across Autonomous Vehicles (AV) [40, 52, 56], AR/VR [27, 48], and Robotics [31]. Taking AV as an example, the LiDAR is synchronized and calibrated against the RGB camera. The correspondence between RGB pixels and LiDAR 3D points is determined by projecting the LiDAR 3D point onto the image plane, as shown with the Fig. 1 depthmaps.

Despite ideal synchronization and calibration, there still exists systematic misalignment between the RGB camera and LiDAR. Fig. 1 highlights the projective artifacts in LiDAR depthmap with blue arrows, where the background scans incorrectly overlay on the foreground objects.

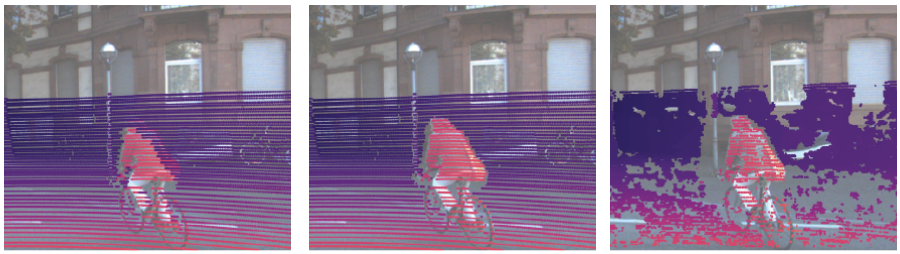
The projective artifacts arise from the physical baseline distance between the RGB camera and LiDAR. This physical limitation is ubiquitous among different LiDAR-RGB sensor sets used in Robotics, AR/VR, and AV tasks (Fig. 2). The baseline distance (Fig. 2a) renders certain 3D scans exclusively visible to LiDAR, rather than the RGB camera. For those scans, their correspondence with image pixels is absent, causing Fig. 1 projective artifacts.

The KITTI dataset [1] provides a heuristic solution to remove the projective artifacts, which we show in Fig. 3. However, the method requires stereo sensors, making it inapplicable to other AV datasets. Thus, the projective artifacts persist **unattended** in most AV datasets. Tab. 1 summarizes these datasets.

We propose RePLAy: a parameter-free, non-learning, and analytical solution to resolve the projective artifacts. RePLAy only requires the relative pose between the LiDAR scanner and the RGB camera, *i.e.*, their extrinsic matrix or calibration matrix. It applies to all AV datasets surveyed in Tab. 1 and diverse sensor settings included in Fig. 2. RePLAy does not even require RGB images, hence being intact at the asynchronization noise between the RGB camera and LiDAR. Fig. 3 provides a qualitative comparison with KITTI solution.

RePLAy first adopts an auto-calibration algorithm to calibrate a hypothesized virtual camera at the origin of the LiDAR 3D point cloud. The virtual LiDAR camera formulates a stereo system with the RGB camera. We attribute the projective artifact to the epipolar occlusion in stereo vision. We propose an analytical solution for detecting epipolar occlusion using epipolar geometry.

LiDAR projected depthmap is essential for diverse downstream applications. We select two to verify our effectiveness: monocular depth estimation (MonoDE) [5, 50, 59, 70] and monocular 3D object detection (Mono3DOD) that uses auxiliary depthmap [51, 63, 64, 66]. We benchmark their performance trained



(a) Raw Depthmap **(b)** RePLAy **(c)** KITTI [57] with Stereo

Fig. 3: Qualitative comparison with KITTI semi-dense depthmap. RePLAy only requires the LiDAR calibration matrix. KITTI [57] uses multi-frame fusion with the stereo camera to remove projective artifacts. The higher density of KITTI depthmap is from the additional sensor and temporal fusion.

Table 1: LiDAR Depthmap Projective Artifacts exist in most Autonomous Vehicle Datasets. KITTI uses stereo images as a heuristic solution.

	KITTI [1]	KITTI-360 [41]	nuScenes [10]	Waymo [56]	DDAD [24]	ApolloScape [30]	Argoverse [62]
Artifacts	No	Yes	Yes	Yes	Yes	Yes	Yes

using LiDAR depthmap cleaned with multiple algorithms and RePLAy on AV datasets. We demonstrate **unanimous** improvement in State-of-The-Art (SoTA) methods [5,29,51] on both tasks. We additionally release the processed depthmap of five major AV datasets to benefit the community. Our contributions are:

- ✓ We utilize epipolar geometry to remove the projective artifacts in the LiDAR depthmap.
- ✓ We demonstrate unanimous improvement in the SoTA monocular depth and monocular 3D object detection methods.
- ✓ We provide processed depthmap in KITTI [1], KITTI-360 [41], nuScenes [10], Waymo [56], and DDAD [24] datasets.

2 Related Work

LiDAR and Camera Misalignment. Multiple sources of misalignment exist between LiDAR and RGB camera. Foggy, rainy, and snowy weather negatively impacts both LiDAR [16, 37, 47] and RGB cameras [18–20, 54]. The slight drift of LiDAR and camera accumulates during vehicle operation. [3, 39, 67] investigate the robustness to the spatial misalignment. [67] benchmarks the temporal asynchronization between LiDAR and RGB camera. The projective artifact is another misalignment that universally exists in multiple AV datasets, as shown in Tab. 1. Our solution complements prior study [67] by addressing it.

Heuristic Solution for Projective Artifacts. KITTI [1, 57] dataset uses the stereo image as a heuristic solution. [57] produces a stereo reconstruction through semi-global matching [28]. They additionally augment the stereo results with 11 neighboring temporal frames. The projective artifacts are removed via

Algorithm 1 RePLAy: Remove Projective Artifacts

Input: LiDAR Point Cloud $\mathcal{V}$, Calibration Matrix $\mathbf{P} = \begin{bmatrix}\mathbf{R}\ \mathbf{t}\end{bmatrix}$, Camera Intrinsic $\mathbf{K}_r$
Output: Occluded Pixels $\mathcal{O}$

Auto-Calibration ▷ Sec. 3.1
Set Virtual LiDAR Camera $\mathbf{K}_l$
Produce Virtual LiDAR Depthmap $\mathbf{D}_l$
Densify Virtual LiDAR Depthmap $\overline{\mathbf{D}}_l$
for $\mathbf{v} \in \mathcal{V}$ **do**
    $\mathbf{p}_1,\ d_{\mathbf{p}_1} \longleftarrow \pi(\mathbf{R}\mathbf{v} \mid \mathbf{K}_l)$
    for $\mathbf{p}_2 \in \mathcal{L}_{\mathbf{p}_1}$ **do** ▷ Eq. (14) and Fig. 8
        $d_{\mathbf{p}_2} \longleftarrow$ bilinear interpolate $\overline{\mathbf{D}}_l$ at pixel location $\mathbf{p}_2$ ▷ Fig. 8
        if $g(\mathbf{p}_1, \mathbf{p}_2, d_{\mathbf{p}_1}, d_{\mathbf{p}_2} \mid \mathbf{K}_l, \mathbf{K}_r, \begin{bmatrix}\mathbf{I}\ \mathbf{t}\end{bmatrix}) \leq 0$ **then** ▷ Eq. (15)
            $\{\mathbf{p}_1\} \cup \mathcal{O}$
            break

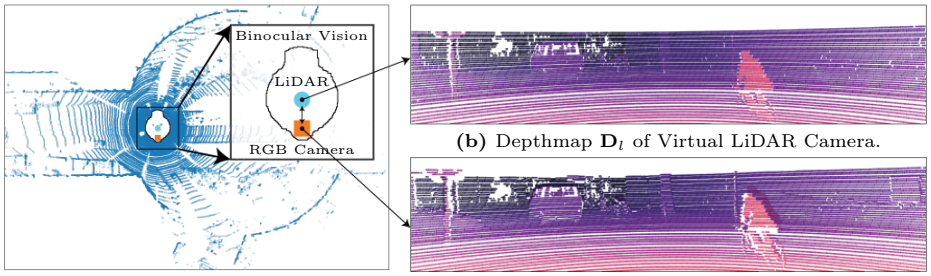
consistency verification between the stereo depthmap and LiDAR depthmap. Compared to ours, depthmap from [57] has improved density. However, the requirement of stereo sensors limits its application, as summarized in Tab. 1. Additionally, [57] has a complicated pipeline, and is therefore, not used even in the KITTI-360 [41] dataset, which shares a similar setup.

Stereo Occlusion Detection. Stereo co-planar cameras are classically formulated as the half-occlusion problem in stereo vision, *i.e.*, the area occluded in one camera of a stereo vision system. There has been extensive research on detecting the half-occlusion region [2, 45, 58, 65]. Several works [6, 7, 32, 60, 72] jointly optimize occlusion and disparity. Others [17, 55, 61] apply left-right consistency check. The area receives different estimations from left and right views are excluded. RePLAy focusses on setups without stereo but where LiDAR provides rough 3D structures. Compared to the half-occlusion baseline which proposes a heuristic solution, RePLAy is an analytical solution.

Downstream Tasks using LiDAR Depthmap. Multiple downstream tasks use LiDAR projected depthmaps. *E.g.*, MonoDE methods [4, 5, 38, 68, 71] use depthmap as the GT supervision. Mono3DOD methods [36, 43, 63, 64, 66] use depthmap for supervising auxiliary task along with 3D object detection. When deployed to AV datasets other than KITTI, the aforementioned methods [42, 69] have diminished performance due to the projective artifacts. RePLAy improves their performance by resolving the projective artifacts in LiDAR depthmap.

3 RePLAy

We now describe RePLAy in Algorithm 1. Sec. 3.1 attributes the projective artifacts to the epipolar occlusion induced by the binocular vision system between the virtual LiDAR camera and RGB camera. Sec. 3.2 outlines our analytical solution. Sec. 3.3 hold further discussions.



(a) LiDAR Point Cloud Bird-Eye View

(b) Depthmap $\mathbf{D}_l$ of Virtual LiDAR Camera.

(c) Depthmap $\mathbf{D}_r$ of RGB Camera.

Fig. 4: Virtual LiDAR camera and RGB Binocular System. In (a), we plot the bird-eye view of the point cloud $\mathcal{V}$ from LiDAR. A binocular system is set between virtual LiDAR (epipolar left) and RGB camera (epipolar right). Subfigure (b) plots the depthmap $\mathbf{D}_l$ from LiDAR view. As stated in Lemma 1, depthmap $\mathbf{D}_l$ does not have projective artifacts. Compare (b) and (c), the artifacts arise when transfer from LiDAR to RGB camera, due to the epipolar occlusion among the binocular system. We formally define the projective artifacts (or epipolar occlusion) in Eq. (7).

We execute Sec. 3.1 auto-calibration over the entire dataset. We initialize LiDAR intrinsic $\mathbf{K}_l$ the same as the RGB camera $\mathbf{K}_r$ with slight augmentation to ensure a sufficient Field-of-View.

3.1 LiDAR Depthmap Projective Artifacts as Epipolar Occlusion

Fig. 2 demonstrates the baseline distance between various LiDAR and RGB camera sensor sets. Fig. 2a shows that the background tree is solely visible to LiDAR. Intuitively, the background tree is occluded by the foreground pedestrian in a binocular vision system of the virtual LiDAR camera and RGB camera.

Virtual LiDAR Camera. LiDAR produces a point cloud $\mathcal{V}$ by emitting laser pulses into 3D space. Mathematically, we assume the LiDAR point cloud $\mathcal{V}$ as a set of 3D rays originating from the coordinate origin:

$$\mathcal{V} = \{\mathbf{v}_i\} = \{\mathbf{0} + \bar{\mathbf{r}}_i \cdot t_i \mid i \in \mathbb{Z},\ t > 0,\ \bar{\mathbf{r}}_i \in \mathbb{R}^3\}, \tag{1}$$

where $\mathbf{v}_i$ is a 3D point. $\bar{\mathbf{r}}_i$ is a normalized direction vector, and t_i is the euclidean distance between the 3D point and LiDAR. If we position a virtual LiDAR camera with arbitrary resolution and intrinsic $\mathbf{K}_l$ at the LiDAR's location, then:

Lemma 1. *The depthmap of the virtual LiDAR camera is free of projective artifacts, i.e., no two scanned points* $\mathbf{v}_1$ *and* $\mathbf{v}_2$ *overlap after projection.*

$$\forall \mathbf{v}_1, \mathbf{v}_2 \in \mathcal{V},\ \|\pi(\mathbf{v}_1 \mid \mathbf{K}_l) - \pi(\mathbf{v}_2 \mid \mathbf{K}_l)\|_2 \neq 0, \tag{2}$$

where $\pi(\cdot)$ stands for the projection process. See proof in Supp. (Appendix A). We visualize the artifacts-free depthmap $\mathbf{D}_l$ from the virtual LiDAR camera in Fig. 4b.

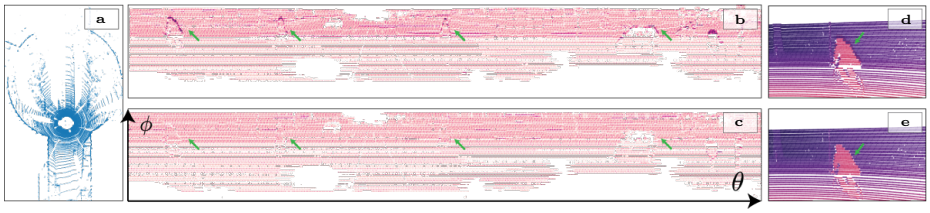


Fig. 5: Auto-Calibration. The LiDAR point cloud $\mathcal{V}$ in (a) is mapped to spherical coordinate $\theta - \phi$ with Eq. (3). We use a binary matrix $\mathbf{S}$ to record the unique pixel locations $\mathcal{S}$ after rasterization. Plots (b) and (c) are matrix $\mathbf{S}$ before and after auto-calibration where duplication (marked by arrows) is minimized with Eq. (5) loss L. Plots (d) and (e) contrast the virtual LiDAR depthmap $\mathbf{D}_l$ before and after auto-calibration where Lemma 1 is fulfilled.

Auto-Calibration. In practice, Lemma 1 may not hold in a realistic LiDAR point cloud due to possible bias in the point cloud coordinate origin $\mathbf{0}$. Fig. 5d, an example from the KITTI dataset shows persistent projective artifacts in the depthmap $\mathbf{D}_l$ when without any treatment. We perform auto-calibration to fulfill Lemma 1 by re-adjusting the point cloud origin. Fig. 4b and Fig. 5e show artifacts-free depthmap after auto-calibration.

Auto-calibration aims to resolve the artifacts in the virtual LiDAR camera depthmap $\mathbf{D}_l$. We convert each LiDAR point cloud to the spherical coordinate system. This hypothesizes an omnidirectional image $\mathbf{S}$ of the LiDAR point cloud. Denote the origin to estimate as $\mathbf{o}$. Set the new 3D point $\mathbf{u} = \mathbf{v} + \mathbf{o}$, we have the polar angle θ and azimuthal angle ϕ after applying a rasterization operation $[\cdot]$:

$$\theta = [\arccos(u_x/(u_x^2 + u_z^2))],\ \phi = [\arccos(u_y/(u_x^2 + u_y^2 + u_z^2))],\ [\theta\ \phi] = f(\mathbf{v} + \mathbf{o}). \quad (3)$$

Intuitively, function $f(\cdot)$ converts a 3D point $\mathbf{u} \in \mathcal{V}$ to a discrete 2D coordinate in $\mathbf{S}$. We denote the set of unique coordinates as $\mathcal{S}$:

$$\mathcal{S} = \{f(\mathbf{v} + \mathbf{o}) \mid \mathbf{v} \in \mathcal{V}\}. \quad (4)$$

We implement Eq. (4) with a binary matrix $\mathbf{S}$. Its elements are 1 at each discrete coordinate and 0 otherwise. The size of $\mathcal{S}$ equals to summarizing the matrix $\mathbf{S}$. Artifacts become duplication in set $\mathcal{S}$. Removing artifacts is to minimize duplication. The loss L is then defined as:

$$L = \|\mathcal{V}\| - \|\mathcal{S}\|. \quad (5)$$

The loss L is not differentiable. In each iteration, we approximate its gradient by ablating origin $\mathbf{o}$ with a small deviation and perform coordinate descent.

Virtual LiDAR Camera and RGB Binocular System. In Fig. 4, we set up a binocular vision system, denoting the virtual LiDAR camera and RGB camera as the epipolar left and right camera respectively. Correspondingly, set their intrinsic as $\mathbf{K}_l$ and $\mathbf{K}_r$, and projected depthmap as $\mathbf{D}_l$ and $\mathbf{D}_r$. Thus the problem of removing the projective artifacts is translated to detecting the epipolar occlusion between the virtual LiDAR camera and RGB camera.

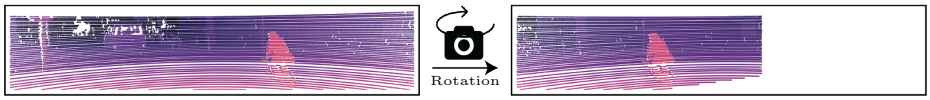


Fig. 6: Epipolar Occlusion with Pure Rotation Movement. We apply a rotation transformation along the x-z axis in the camera coordinate system. As indicated by Lemma 2, a pure rotation movement does not lead to projective artifacts.

Projective Artifacts are Epipolar Occlusion. We ensure that our virtual LiDAR has a larger field of view than the RGB camera by adjusting the resolution of the virtual LiDAR camera. Denote the relative movement between the LiDAR and RGB camera as $\mathbf{P} = \begin{bmatrix}\mathbf{R}\ \mathbf{t}\end{bmatrix}$, where the auto-calibrated center $\mathbf{o}$ is integrated into $\mathbf{t}$. Denote the projected pixels on epipolar left (LiDAR) and epipolar right (RGB) as $\mathbf{p}$ and $\mathbf{q}$, and their pixel depths as d and d'. We have:

$$\mathbf{p} = \pi(\mathbf{v} \mid \mathbf{K}_l), \quad \mathbf{q} = \pi(\mathbf{R}\mathbf{v} + \mathbf{t} \mid \mathbf{K}_r). \tag{6}$$

We then formally define the epipolar occlusion $\mathcal{O}$ on RGB view as:

$$\mathcal{O} = \{\mathbf{q}_1 \mid \forall\ \mathbf{q}_1, \mathbf{q}_2 \in \mathcal{Q},\ d'_{\mathbf{q}_1} > d'_{\mathbf{q}_2},\ \|\mathbf{q}_1 - \mathbf{q}_2\|_2 = 0\}. \tag{7}$$

Eq. (7) simply suggests if two pixels in RGB camera overlap with each other, then the one $\mathbf{q}_1$ with larger depth is occluded by the one with smaller depth $\mathbf{q}_2$. Occlusion detection finds all occluded pixels $\mathbf{q}_1$, *i.e.*, the set $\mathcal{O}$.

3.2 Epipolar Occlusion under Rotation and Translation

Suppose the relative camera motion among the Fig. 4a binocular system is an arbitrary camera pose $\mathbf{P} = \begin{bmatrix}\mathbf{R}\ \mathbf{t}\end{bmatrix}$. Set $\mathbf{I}$ as the identity matrix. RePLAy decomposes $\mathbf{P}$ into a sequential **pure rotation** and **pure translation** movement:

$$\mathbf{P}_1 = \begin{bmatrix}\mathbf{R}\ \mathbf{0}\end{bmatrix},\ \mathbf{P}_2 = \begin{bmatrix}\mathbf{I}\ \mathbf{t}\end{bmatrix},\ \mathbf{P} = \mathbf{P}_2\mathbf{P}_1. \tag{8}$$

We outline pure rotation movement in Lemma 2 and Fig. 6. The pure translation movement is illustrated in Lemma 3 and Fig. 7. Without loss of generality, we assume the epipoplar left and right cameras share the same intrinsic, *i.e.*, $\mathbf{K}_l = \mathbf{K}_r$. When with different intrinsics, we apply a resize and crop operation defined by $\mathbf{A} = \mathbf{K}_l\mathbf{K}_r^{-1}$ to unify the intrinsic.

Lemma 2. ***Pure rotation movement*** *does not cause epipolar occlusion.*

We know from epipolar geometry [25] that pure rotation induces homography between two views. In other words, there is a one-to-one mapping between pixels before and after rotation. See Supp for more information. Fig. 6 illustrates Lemma 2 with a visual example.

Lemma 3. ***Pure translation movement*** *causes epipolar occlusion along the direction of the epipolar line.*

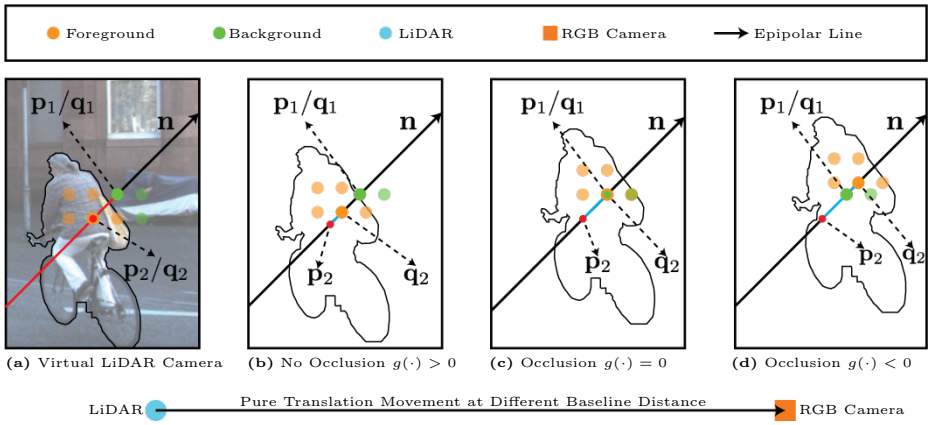


Fig. 7: Epipoplar Occlusion with Pure Translation Movement. A background pixel $\mathbf{p}_1$ is gradually occluded by a foreground pixel $\mathbf{p}_2$ under pure translation. We formulate three binocular systems at different baseline distances. For simplicity, we set the depth of $\mathbf{p}_1$ to infinity, making its 2D location static. Direction vector $\mathbf{n}$ is set to point from $\mathbf{p}$ to $\mathbf{q}$. The scalar t is marked by blue line segment. From Lemma 3, **(a)** The pixel $\mathbf{p}_1$ is only occluded by pixel $\mathbf{p}_2$ at the reverse direction of epipolar line (red line segment). **(b)** $\mathbf{p}_1$ is not occluded. **(c)** $\mathbf{p}_1$ is occluded following Eq. (7). **(d)** $\mathbf{p}_1$ is considered occluded following Eq. (17). Intuitively, the occluded pixel $\mathbf{p}_1$ (in a) is "caught up" (in c) and "surpassed" (in d) by $\mathbf{p}_2$ while moving along the epipolar line.

Proof. As pure rotation does not cause epipoplar occlusion, we apply the translation movement over the artifacts-free depthmap after the pure rotation. With slight abuse of the notation, we set the pixels after pure rotation movement as $\mathbf{p}$ and $\mathbf{q}$. The epipolar line $\mathbf{l}_\mathbf{p}$ on the right image from the left image pixel $\mathbf{p}$ is:

$$\mathbf{l}_\mathbf{p} = \mathbf{K}^{-\intercal}[\mathbf{t}_\times]\mathbf{R}\mathbf{K}^{-1}\mathbf{p} = \mathbf{K}^{-\intercal}[\mathbf{t}_\times]\mathbf{K}^{-1}\mathbf{p}, \tag{9}$$

where $[\mathbf{t}_\times]$ is the cross product in matrix form [71]. Due to a pure translation movement, the rotation $\mathbf{R}$ is removed as being an identity matrix. Further, the left and right epipolar lines overlap with each other in pure translation:

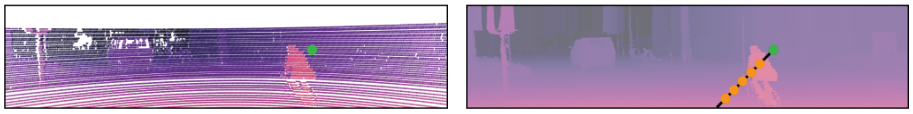
$$\mathbf{l}_\mathbf{q} = -\mathbf{K}^{-\intercal}[\mathbf{t}_\times]\mathbf{K}^{-1}\mathbf{q},\ \mathbf{p}^\intercal\mathbf{l}_\mathbf{q} = 0,\ \mathbf{p}^\intercal\mathbf{l}_\mathbf{p} = 0. \tag{10}$$

Also shown in Fig. 7, pixel $\mathbf{q}$ is hence a function of pixel $\mathbf{p}$:

$$\mathbf{q} = \mathbf{p} + t \cdot \mathbf{n},\ t = \|\mathbf{q} - \mathbf{p}\|_2 = \mathbf{n}^\intercal(\mathbf{q} - \mathbf{p}),\ \mathbf{n} = (\mathbf{q} - \mathbf{p})/\sqrt{\|\mathbf{q} - \mathbf{p}\|_2^2}, \tag{11}$$

where t is the moving distance, similar to the disparity in stereo vision. We define the vector $\mathbf{n}$ pointed from $\mathbf{p}$ to $\mathbf{q}$ as the direction of the epipoplar line $\mathbf{l}$. $\mathbf{p}$ and $\mathbf{q}$ are related by the equation $d'\mathbf{q} = d\mathbf{K}_r\mathbf{R}\mathbf{K}_l^{-1}\mathbf{p} + \mathbf{K}_r\mathbf{t}$. Expanding under pure translation movement gives us:

$$t^2 = (p_x - q_x)^2 + (p_y - q_y)^2 = \frac{(x + z \cdot q_x)^2 + (y + z \cdot q_y)^2}{(d' - z)^2} = \frac{(x + z \cdot q_x)^2 + (y + z \cdot q_y)^2}{d^2}. \tag{12}$$



(a) Virtual LiDAR Depthmap $\mathbf{D}_l$ **(b)** Densified Virtual LiDAR Depthmap $\overline{\mathbf{D}}_l$

Fig. 8: Per-pixel Occlusion Examination. To overcome LiDAR sparsity, for a query pixel, we verify inspection pixels along epipolar line $\mathcal{L}_{\mathbf{p}_1}$ following Eq. (17) to determine occlusion. Depth $\overline{\mathbf{D}}_l$ is densified with nearest-neighbor interpolation.

Please see the derivation in Supp. (Appendix B). Note, we assume pixels $\mathbf{p}$ and $\mathbf{q}$ are respectively visible to the left and right camera. This suggests $d > 0$ and $d = d' - z > 0$. Eq. (12) suggests moving distance t^2 **monotonously** decreases as the projected depth d' increases. It fits the intuition where a foreground object has a larger movement than a background object. From Eq. (7), when with occlusion, we have $\mathbf{q}_1 = \mathbf{q}_2$, and $d_{\mathbf{q}_1} < d_{\mathbf{q}_2}$. Combine Eq. (11) and Eq. (12):

$$\mathbf{q}_1 = \mathbf{q}_2, \text{ and } d_{\mathbf{q}_1} < d_{\mathbf{q}_2} \longrightarrow t_1^2 > t_2^2 \longrightarrow \mathbf{n}^\intercal(\mathbf{p}_1 - \mathbf{p}_2) < 0. \tag{13}$$

From Eq. (13), if pixel $\mathbf{q}_1$ is occluded by $\mathbf{q}_2$ in the right view, pixel $\mathbf{q}_2$ must move a longer distance along the epipolar line than $\mathbf{q}_1$. This suggests, in the left view, $\mathbf{p}_2$ resides at the "back" of $\mathbf{p}_1$ *w.r.t.* the moving direction. In other words, $\mathbf{p}_1$ is only occluded by other pixels along the reverse direction of epipolar line.

Epipolar Occlusion Judgement. Combine Lemma 2 and 3, a pixel $\mathbf{p}_1$ in LiDAR view is occluded by other pixels $\mathbf{p}_2$ along the reverse of epipolar line direction. For pixel $\mathbf{p}_1$, the set of inspection pixels is:

$$\mathcal{L}_{\mathbf{p}_1} = \{\mathbf{p}_2 \mid \mathbf{p}_2 = \mathbf{p}_1 - \mathbf{n} \cdot k \cdot \Delta t,\ k \in \mathbb{Z}\}. \tag{14}$$

k is an integer. At occlusion, we have $\mathbf{q}_1 = \mathbf{q}_2$, combined with Eq. (11):

$$\mathbf{q}_1 = \mathbf{q}_2 \longrightarrow \mathbf{p}_1 + t_1\mathbf{n} = \mathbf{p}_2 + t_2\mathbf{n} \longrightarrow \mathbf{n}^\intercal(\mathbf{p}_1 - \mathbf{p}_2) + t_1 - t_2 = 0. \tag{15}$$

We use Fig. 7 to explain Eq. (15). In Fig. 7a, the pixel $\mathbf{p}_1$ and $\mathbf{p}_2$ has an initial distance $\mathbf{n}^\intercal(\mathbf{p}_1 - \mathbf{p}_2)$. In Fig. 7c, after translation movement, each pixel moves a distance t_1 and t_2 along the epipolar line. Occlusion happens when pixel $\mathbf{p}_2$ overlaps pixel $\mathbf{p}_1$. We summarize Eq. (15) into a function $g(\cdot)$ as:

$$\lambda \leq g(\mathbf{p}_1, \mathbf{p}_2, d_{\mathbf{p}_1}, d_{\mathbf{p}_2} \mid \mathbf{K}, \mathbf{P}) \leq 0, \tag{16}$$

where the distance t_1 and t_2 are computed using intrinsic, camera motion, and depth values. In Fig. 7b, when $g(\cdot) > 0$, there is no occlusion. In Fig. 7c, when $g(\cdot) = 0$, two pixels overlap and occlusion happens. However, this is hard due to LiDAR sparsity, rasterization, and other noise sources. Hence, we relax with a threshold λ. In practice, we set λ to $-\infty$. This suggests the pixel $\mathbf{p}_1$ is occluded if any pixel $\mathbf{p}_2$ in the epipolar line "surpasses" it in the RGB view, as shown in Fig. 7d. Formally, we consider pixel $\mathbf{p}_1$ occluded as:

$$\mathcal{O} = \{\mathbf{p}_1 \mid \exists\ \mathbf{p_2} \in \mathcal{L}_{\mathbf{p}_1}, g(\mathbf{p}_1, \mathbf{p}_2, d_{\mathbf{p}_1}, d_{\mathbf{p}_2} \mid \mathbf{K}, \mathbf{P}) < 0\}. \tag{17}$$

Eq. (17) is also a necessary condition of epipolar occlusion. Due to the LiDAR sparsity, we apply a nearest-neighbor interpolation over virtual LiDAR depthmap $\mathbf{D}_l$ to acquire $d_{\mathbf{p}_2}$ whenever required, as shown in Fig. 8.

3.3 Discussions

Half-Occlusion (Depth Buffering) Baseline. Sec. 2 shows that half-occlusion algorithms solve projective artifacts in stereo vision, *i.e.*, stereo/epipolar occlusion, without knowing LiDAR point cloud. Despite a different setting, they provide a baseline solution. When preparing RGB depthmap $\mathbf{D}_r$, only pixels with smaller depth are preserved if multiple 3D points $\mathbf{v}$ have duplicate 2D locations after rasterization. The raw LiDAR depthmap uses half-occlusion. So, we refer Half-Occlusion as "Raw" following [23].

Does RePLAy simply densify the sparse depthmap to remove occlusion? No. Densification alone is insufficient to remove occlusion. RePLAy enables robust densification by formulating a stereo system with an *auto-calibration* procedure to remove occlusion.

Universal Existence of Projective Artifacts. Fig. 2 pictorially depicts that projective artifacts exist under all LiDAR and camera sensor settings, while Fig. 7 shows that these artifacts scale up with a larger baseline distance between sensors. We now show that this indeed is the case. The artifact width distribution plot on the outdoor KITTI and the indoor NYUv2 [53] datasets are shown in Fig. 10 in the Supp. The LiDAR and RGB camera in KITTI have a baseline distance of $0.4m$ [22]. We set the baseline distance to $0.1m$ following the AR headset configuration of Fig. 2. Despite the indoor image having a lesser resolution than outdoor (480×640 compared to 375×1242), we observe a similar distribution. Detailed definitions and evaluation protocols are in Supp. The analysis confirms that projective artifacts exist in both indoor and outdoor datasets.

4 Experiments

We benchmark RePLAy on depthmap quality, MonoDE and Mono3DOD tasks. Sec. 4.1 assesses depthmap quality, while Sec. 4.2 and Sec. 4.3. report MonoDE and Mono3DOD results respectively.

Datasets. We experiment on KITTI [21], nuScenes [10] and Waymo [56] datasets. KITTI provides stereo ground truth while the others do not.

Data Splits. We use the following splits of the above datasets:

- *KITTI Eigen* [15]: 23,158 training and 652 validation samples of raw KITTI.
- *KITTI Stereo* [46]: 200 training and 200 testing samples out of which 145 training scenes are mapped to the raw KITTI dataset.
- *nuScenes Frontal Val*: 28,130 training and 2,436 random validation samples (from 6,019 samples) from front camera and top LiDAR.
- *Waymo Frontal Val*: 52,386 training and 40,839 validation samples from the front camera and top LiDAR. We construct its training set by sampling every third frame from the training sequences as in [34].

Table 2: Depthmap Quality Assessment with baselines and RePLAy. We adopt KITTI semi-dense depthmap and KITTI Stereo disparity map as the reference GT. [Key: **Best**, **Second Best**, Mod. = Modified]

GT	Eval Region	Depthmap	$\delta_{0.5}\uparrow$	$\delta_1\uparrow$	$\delta_2\uparrow$	RMS ↓	$\text{RMS}_{\log}$ ↓	ARel ↓	SRel ↓	$\log_{10}$ ↓	$\text{SI}_{\log}$ ↓
Semi-Dense	All	Raw	0.973	0.983	0.990	1.935	0.087	0.029	0.332	0.010	8.660
		Mod. Half-Occ	**0.983**	**0.991**	**0.996**	**1.347**	**0.057**	**0.019**	**0.113**	**0.007**	**5.712**
		RePLAy	**0.985**	**0.994**	**0.998**	**1.241**	**0.046**	**0.016**	**0.063**	**0.007**	**4.592**
Stereo	All	Raw	0.950	0.962	0.970	4.198	0.169	0.083	1.604	0.024	16.691
		Mod. Half-Occ	**0.971**	**0.982**	**0.988**	**2.541**	**0.105**	**0.047**	**0.546**	**0.016**	**10.340**
		RePLAy	**0.975**	**0.986**	**0.991**	**2.224**	**0.086**	**0.034**	**0.354**	**0.014**	**8.362**
	Foreground	Raw	0.859	0.871	0.892	8.726	0.317	0.228	5.974	0.058	29.772
		Mod. Half-Occ	**0.919**	**0.931**	**0.949**	**5.285**	**0.199**	**0.154**	**2.366**	**0.033**	**19.250**
		RePLAy	**0.925**	**0.937**	**0.956**	**4.837**	**0.179**	**0.100**	**1.929**	**0.030**	**17.319**
	Background	Raw	0.967	0.980	0.986	2.708	0.113	0.052	0.774	0.017	11.037
		Mod. Half-Occ	**0.980**	**0.991**	**0.995**	**1.701**	**0.070**	**0.032**	**0.210**	**0.013**	**6.638**
		RePLAy	**0.982**	**0.993**	**0.998**	**1.439**	**0.052**	**0.027**	**0.085**	**0.012**	**4.722**

Table 3: Depthmap Quality Assessment on Removed Artifacts. RePLAy outperforms the Mod. half-occlusion baseline. For removed artifacts, a worse performance suggests a better algorithm. [Key: **Best**]

GT	Eval Region	Depthmap	$\delta_{0.5}\downarrow$	$\delta_1\downarrow$	$\delta_2\downarrow$	RMS ↑	$\text{RMS}_{\log}$ ↑	ARel ↑	SRel ↑	$\log_{10}$ ↑
Semi-Dense	All	Mod. Half-Occ	0.814	0.846	0.893	5.850	0.275	0.196	3.762	0.053
		RePLAy	**0.274**	**0.363**	**0.548**	**12.084**	**0.587**	**0.778**	**15.851**	**0.202**

- *KITTI Detection Val*: 3,748 training and 3,762 validation images from the left camera for the Mono3DOD task.

Ground Truth (GT) Depthmaps. We use the following as GT depthmaps:

- *KITTI Eigen*: We use the semi-dense depthmap available in KITTI.
- *KITTI Stereo*: We generate depthmaps from GT disparity maps, which are denser than semi-dense depthmaps.
- *nuScenes* and *Waymo Frontal Val*: We generate GT depthmaps by projecting only those LiDAR points present inside GT 3D bounding boxes. To prepare a reference depthmap, we follow [49] to filter LiDAR point cloud with GT 3D bounding boxes labels. Correspondingly, the evaluation is confined to the bounding box area, as only depth labels in that region are retained.

4.1 Depthmap Quality Assessment

Baselines. We compare RePLAy to raw depthmap, which is the half-occlusion (depth buffering) method and a modified half-occlusion method. The modified half-occlusion modifies the half-occlusion method (Sec. 3.3) by back-projecting the estimated dense virtual depthmap ($\overline{\mathbf{D}}_l$) onto the RGB camera and only preserving pixels in $\mathbf{D}_r$ that are less than their back-projected values.

Results. Tab. 2 compares the RePLAy depthmap with these baselines processed depthmaps. Tab. 2 verifies that RePLAy depthmap outperforms the raw and modified half-occlusion baselines. RePLAy depthmap agrees more with semi-dense or disparity depthmap after removing the projective artifacts. Fig. 3 shows qualitative comparisons. We believe projective artifacts persist in the half-occlusion method because of the LiDAR sparsity, leading to inferior depthmaps.

Table 4: Auto-Calibration Ablation on Depthmap Quality Assessment with semi-dense GT depthmap. [Key: **Best**]

Depthmap	AutoCalib	$\delta_{0.5}\uparrow$	$\delta_1\uparrow$	$\delta_2\uparrow$	RMS ↓	$\text{RMS}_{\log}$ ↓	ARel ↓	SRel ↓	$\log_{10}$ ↓	$\text{SI}_{\log}$ ↓
RePLAy	×	0.981	0.990	0.995	1.473	0.057	0.019	0.111	0.007	5.680
	✓	**0.985**	**0.994**	**0.998**	**1.241**	**0.046**	**0.016**	**0.063**	**0.007**	**4.592**

Table 5: MonoDE Results on KITTI Stereo. ZoeDepth [5] trained with RePLAy depthmaps outperforms raw and semi-dense [57] depthmaps on foreground. [Key: **Best**, **Second Best**].

Eval Region	Depthmap	$\delta_1\uparrow$	$\delta_2\uparrow$	$\delta_3\uparrow$	RMS ↓	$\text{RMS}_{\log}$ ↓	ARel ↓	SRel ↓	$\log_{10}$ ↓	$\text{SI}_{\log}$ ↓
All	Raw	0.955	0.982	0.992	2.647	0.116	0.067	0.420	0.026	11.324
	Semi-Dense	**0.977**	**0.994**	**0.997**	**2.274**	**0.082**	**0.043**	**0.201**	**0.018**	**7.971**
	RePLAy	**0.974**	**0.992**	**0.996**	**2.288**	**0.089**	**0.049**	**0.248**	**0.020**	**8.588**
Foreground	Raw	0.886	0.947	0.980	3.692	0.180	0.127	1.112	0.022	16.518
	Semi-Dense	**0.976**	**0.990**	**0.994**	**2.093**	**0.091**	**0.056**	**0.344**	**0.022**	**8.342**
	RePLAy	**0.945**	**0.978**	**0.989**	**2.939**	**0.128**	**0.081**	**0.642**	**0.029**	**11.798**
Background	Raw	0.973	0.993	0.998	2.294	0.084	0.048	0.188	0.020	7.981
	Semi-Dense	**0.978**	**0.996**	**0.998**	**2.283**	**0.076**	**0.040**	**0.167**	**0.017**	**7.239**
	RePLAy	**0.981**	**0.997**	**0.999**	**2.084**	**0.070**	**0.039**	**0.138**	**0.017**	**6.681**

Foreground Improvement. Tab. 2 rows 5 and 6 shows a significant improvement in foreground depthmap quality. On metric $\delta_{0.5}$, we reduce the errors by $46.8\% = 1 - \frac{1-0.925}{1-0.859}$. This aligns with our observation that projective artifacts mostly occur over foreground objects, confirming Fig. 1.

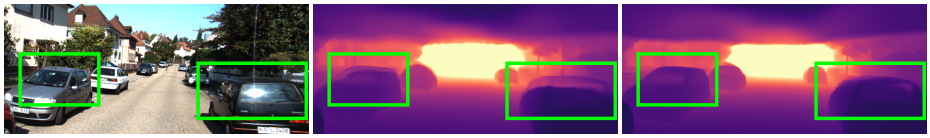
Comparison on Removed Artifacts. We next compare RePLAy on removed artifacts in Tab. 3. Tab. 3 shows that that in-addition to removing some artifacts, the modified half-occlusion also removes some non-artifacts (good points). Ideally, the removed points should be artifacts; therefore, the best algorithm should perform the worst on the quality assessment of removed points and hence the reversal in the directional indicator for Tab. 3. Qualitative comparison between raw, modified half-occlusion and RePLAy is provided in Supp. (Fig. 11).

Auto-Calibration Ablation. We next conduct ablations of the Auto-Calibration module (Sec. 3.1) in Tab. 4. Tab. 4 confirms the effectiveness of Auto-Calibration module in RePLAy, since Auto-Calibration provides 15% and 19% relative improvement in ARel and $\text{SI}_{\log}$ metrics respectively.

4.2 MonoDE: Monocular Depth Estimation

Baselines. We compare RePLAy to raw and Semi-Dense methods. We use ZoeDepth [5] as the depth model due to its superior generalization.

KITTI Stereo and Eigen Results. We train ZoeDepth [5] from scratch using depthmap with and without applying RePLAy on KITTI Eigen split. We evaluate on KITTI Stereo and Eigen split in Tabs. 5 and 6 respectively. Tab. 5 shows that ZoeDepth trained with RePLAy depthmap significantly outperforms the baselines. We observe despite semi-dense GT [57] using additional stereo cameras, RePLAy outperforms Semi-Dense especially in the background. This



(a) RGB Input **(b)** Raw depthmaps **(c)** RePLAy depthmaps

Fig. 9: MonoDE Qualitative Results on KITTI Eigen dataset. ZoeDepth [5] trained with RePLAy depthmaps (c) outperforms the one with raw depthmaps (b), especially on foreground.

Table 6: MonoDE Results. ZoeDepth trained with RePLAy outperforms the baselines in foreground regions of all datasets. [Key: **Best**]

Data Split	Eval Region	Depthmap	$\delta_1\uparrow$	$\delta_2\uparrow$	$\delta_3\uparrow$	RMS ↓	$\text{RMS}_{\log}$ ↓	ARel ↓	SRel ↓	$\log_{10}$ ↓	$\text{SI}_{\log}$ ↓
KITTI Eigen	All	Raw	0.869	0.959	0.987	2.911	0.163	0.124	0.606	0.046	13.501
		RePLAy	**0.927**	**0.985**	**0.997**	**2.185**	**0.110**	**0.082**	**0.287**	**0.033**	**8.674**
nuScenes Val	Foreground	Raw	0.971	0.987	0.993	1.347	0.075	0.049	0.245	0.018	6.339
		RePLAy	**0.981**	**0.993**	**0.996**	**1.150**	**0.059**	**0.039**	**0.137**	**0.016**	**4.905**
Waymo Val	Foreground	Raw	0.941	0.988	0.996	3.468	0.111	0.079	0.487	0.032	9.116
		RePLAy	**0.962**	**0.993**	**0.998**	**3.057**	**0.091**	**0.063**	**0.367**	**0.026**	**7.322**

Table 7: MonoDE Analysis to Environmental Conditions on nuScenes Val split. ZoeDepth trained with RePLAy depthmaps shows consistent improvements on foreground under all environmental conditions. [Key: **Best**]

Condition	Samples	Depthmap	$\delta_1\uparrow$	$\delta_2\uparrow$	$\delta_3\uparrow$	RMS ↓	$\text{RMS}_{\log}$ ↓	ARel ↓	SRel ↓	$\log_{10}$ ↓	$\text{SI}_{\log}$ ↓
Rain	864	Raw	0.984	0.991	0.995	1.338	0.058	0.040	0.163	0.016	4.859
		RePLAy	**0.989**	**0.993**	**0.995**	**1.222**	**0.050**	**0.034**	**0.127**	**0.015**	**4.110**
Night	533	Raw	0.965	0.982	0.990	1.347	0.089	0.059	0.335	0.021	7.481
		RePLAy	**0.978**	**0.991**	**0.995**	**1.063**	**0.067**	**0.045**	**0.172**	**0.017**	**5.374**
Sunny	1039	Raw	0.964	0.987	0.993	1.352	0.081	0.053	0.267	0.019	6.985
		RePLAy	**0.976**	**0.993**	**0.996**	**1.134**	**0.062**	**0.040**	**0.128**	**0.016**	**5.324**
All Tab. 6	2436	Raw	0.971	0.987	0.993	1.347	0.075	0.049	0.245	0.018	6.339
		RePLAy	**0.981**	**0.993**	**0.996**	**1.150**	**0.059**	**0.039**	**0.137**	**0.016**	**4.905**

may be due to [57] aggressively removing the background pixels around the foreground object boundaries. Fig. 9 shows this result visually. ZoeDepth trained with the raw depthmap learns strip-shaped artifacts (green boxes).

Foreground Improvement. We additionally ablate depth performance on foreground, background, and entire image using the GT segmentation mask in Tab. 5. Tab. 5 shows that ZoeDepth trained with RePLAy depthmap outperforms in the foreground area, which agrees with the conclusion of Sec. 4.1.

nuScenes and Waymo Frontal Val Results. We benchmark MonoDE on two datasets without stereo sensors: nuScenes [10] and Waymo [56] in Tab. 6. Tab. 6 confirms that RePLAy achieves significant improvements on these datasets.

Effect of Environmental Conditions. We next analyze the effect of environmental conditions on the nuScenes Val split in Tab. 7. We categorize the nuScenes [10] val set into rainy, night, and sunny following their description tag (nuScenes-devkit GitHub #124). Tab. 7 shows that RePLAy is robust across multiple conditions.

4.3 Mono3DOD: Monocular 3D Object Detection

We finally verify the impact of RePLAy on the Mono3DOD task.

Baseline. We compare RePLAy to raw method. We take SoTA 3D detectors, CaDDN [51] and MonoDTR [29], to showcase the impact of cleaning projected LiDAR depthmap with RePLAy, since these detectors use depthmap as an auxiliary training task. The two methods cover both convolutional and transformer architectures, making our benchmarking comprehensive. However, these two SoTA detectors use depth supervision in training differently.

CaDDN. CaDDN uses the densified raw depthmaps of [33]. For CaDDN experiments, we apply RePLAy to the projected depthmaps and then densify those maps using the same depth completion algorithm [33] and parameters of [51].

MonoDTR. MonoDTR [29] precomputes their GT depthmaps by 4× downsampling raw projected depthmaps. For MonoDTR experiments, we clean the projected depthmaps and then 4× downsampling.

Results. Tab. 8 shows results of retraining both detectors with our depthmaps on KITTI Val split. Cleaned depthmaps improves the performance of CaDDN and MonoDTR on most of the metrics. The biggest gains appear in the Easy category. This is expected as Easy cars are closer to the ego camera and receive the maximum number of LiDAR points. Hence, improving the projected depthmap by RePLAy benefits the Easy category more.

Table 8: KITTI Val Mono3DOD Results on Car category. Using RePLAy depthmaps in training boosts the Mono3DOD performance. [Key: **Best**, **Second Best**, * = Reported, †= Retrained, Mod= Moderate.]

Method	Depthmap	$IoU_{3D} > 0.7$ AP $_{3D\|R_{40}}$ [%]↑		
		Easy	Mod	Hard
M3D-RPN [8]	–	14.53	11.07	8.65
MonoPair [12]	–	16.28	12.30	10.42
MonoDLE [44]	–	17.45	13.66	11.68
Kinematic [9]	–	19.76	14.10	10.47
GrooMeD [35]	–	19.67	14.32	11.27
MonoRUn [11]	–	20.02	14.65	12.61
D4LCN [13]	Raw*	22.32	16.20	12.30
DFR-Net [73]	Raw*	24.81	17.78	14.41
CaDDN [51]	Raw†	**21.60**	**15.67**	**13.24**
	RePLAy	**22.88**	**15.70**	**13.18**
MonoDTR [29]	Raw†	24.50	**18.66**	**15.69**
	RePLAy	**26.15**	**18.77**	**15.69**

5 Conclusion

We propose an analytical solution to remove projective artifacts in LiDAR depthmap. Our parameter-free method, RePLAy, applies to diverse LiDAR and RGB camera sensor settings. RePLAy only requires the relative position between LiDAR and RGB camera. We verify its benefits with **unanimous** improvements on SoTA MonoDE and Mono3DOD methods. We release processed depthmaps of five major AV datasets to benefit the community.

Limitation. We use nearest-neighbor interpolation to densify virtual LiDAR depthmaps. Advanced strategies, *e.g.*, depth completion, may improve accuracy.

References

1. The KITTI Vision Benchmark Suite. http://www.cvlibs.net/datasets/kitti/eval_object.php?obj_benchmark=3d (2022), accessed: 2022-07-03 2, 3
2. Arbelaez, P., Maire, M., Fowlkes, C., Malik, J.: Contour detection and hierarchical image segmentation. TPAMI (2010) 4
3. Bai, X., Hu, Z., Zhu, X., Huang, Q., Chen, Y., Fu, H., Tai, C.L.: Transfusion: Robust lidar-camera fusion for 3D object detection with transformers. In: CVPR (2022) 3
4. Bhat, S.F., Alhashim, I., Wonka, P.: Adabins: Depth estimation using adaptive bins. In: CVPR (2021) 4
5. Bhat, S.F., Birkl, R., Wofk, D., Wonka, P., Müller, M.: ZoeDepth: Zero-shot transfer by combining relative and metric depth. arXiv preprint arXiv:2302.12288 (2023) 2, 3, 4, 12, 13
6. Birchfield, S., Tomasi, C.: Depth discontinuities by pixel-to-pixel stereo. IJCV (1999) 4
7. Bobick, A.F., Intille, S.S.: Large occlusion stereo. IJCV (1999) 4
8. Brazil, G., Liu, X.: M3D-RPN: Monocular 3D region proposal network for object detection. In: ICCV (2019) 14
9. Brazil, G., Pons-Moll, G., Liu, X., Schiele, B.: Kinematic 3D object detection in monocular video. In: ECCV (2020) 14
10. Caesar, H., Bankiti, V., Lang, A., Vora, S., Liong, V., Xu, Q., Krishnan, A., Pan, Y., Baldan, G., Beijbom, O.: nuScenes: A multimodal dataset for autonomous driving. In: CVPR (2020) 1, 2, 3, 10, 13
11. Chen, H., Huang, Y., Tian, W., Gao, Z., Xiong, L.: MonoRun: Monocular 3D object detection by reconstruction and uncertainty propagation. In: CVPR (2021) 14
12. Chen, Y., Tai, L., Sun, K., Li, M.: MonoPair: Monocular 3D object detection using pairwise spatial relationships. In: CVPR (2020) 14
13. Ding, M., Huo, Y., Yi, H., Wang, Z., Shi, J., Lu, Z., Luo, P.: Learning depth-guided convolutions for monocular 3D object detection. In: CVPR Workshops (2020) 14
14. Ebadi, K., Bernreiter, L., Biggie, H., Catt, G., Chang, Y., Chatterjee, A., Denniston, C.E., Deschênes, S.P., Harlow, K., Khattak, S., et al.: Present and future of slam in extreme underground environments. arXiv preprint arXiv:2208.01787 (2022) 2
15. Eigen, D., Puhrsch, C., Fergus, R.: Depth map prediction from a single image using a multi-scale deep network. NeurIPS (2014) 10
16. Filgueira, A., González-Jorge, H., Lagüela, S., Díaz-Vilariño, L., Arias, P.: Quantifying the influence of rain in lidar performance. Measurement (2017) 3
17. Fua, P.: A parallel stereo algorithm that produces dense depth maps and preserves image features. Machine vision and applications (1993) 4
18. Garg, K., Nayar, S.K.: When does a camera see rain? In: ICCV (2005) 3
19. Garg, K., Nayar, S.K.: Vision and rain. IJCV (2007) 3
20. Garvelmann, J., Pohl, S., Weiler, M.: From observation to the quantification of snow processes with a time-lapse camera network. Hydrology and Earth System Sciences (2013) 3
21. Geiger, A., Lenz, P., Stiller, C., Urtasun, R.: Vision meets robotics: The KITTI dataset. IJRR (2013) 10
22. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? the KITTI vision benchmark suite. In: CVPR (2012) 10

23. Godard, C., Mac Aodha, O., Firman, M., Brostow, G.J.: Digging into self-supervised monocular depth estimation. In: ICCV (2019) 10
24. Guizilini, V., Ambrus, R., Pillai, S., Raventos, A., Gaidon, A.: 3D packing for self-supervised monocular depth estimation. In: CVPR (2020) 1, 3
25. Hartley, R., Zisserman, A.: Multiple view geometry in computer vision. Cambridge university press (2003) 7
26. Herskovitz, J., Wu, J., White, S., Pavel, A., Reyes, G., Guo, A., Bigham, J.P.: Making mobile augmented reality applications accessible. In: SIGACCESS (2020) 2
27. Hillmann, C., Hillmann, C.: Comparing the gear vr, oculus go, and oculus quest. Unreal for Mobile and Standalone VR: Create Professional VR Apps Without Coding (2019) 2
28. Hirschmuller, H.: Stereo processing by semiglobal matching and mutual information. TPAMI (2007) 3
29. Huang, K.C., Wu, T.H., Su, H.T., Hsu, W.: MonoDTR: Monocular 3D object detection with depth-aware transformer. In: CVPR (2022) 3, 14
30. Huang, X., Cheng, X., Geng, Q., Cao, B., Zhou, D., Wang, P., Lin, Y., Yang, R.: The apolloscape dataset for autonomous driving. In: CVPR (2018) 3
31. Hutabarat, D., Rivai, M., Purwanto, D., Hutomo, H.: Lidar-based obstacle avoidance for the autonomous mobile robot. In: 2019 12th International Conference on Information & Communication Technology and System (ICTS). pp. 197–202. IEEE (2019) 2
32. Ishikawa, H., Geiger, D.: Occlusions, discontinuities, and epipolar lines in stereo. In: ECCV (1998) 4
33. Ku, J., Harakeh, A., Waslander, S.: In defense of classical image processing: Fast depth completion on the CPU. In: CRV (2018) 14
34. Kumar, A., Brazil, G., Corona, E., Parchami, A., Liu, X.: DEVIANT: Depth Equivariant Network for Monocular 3D Object Detection. In: ECCV (2022) 10
35. Kumar, A., Brazil, G., Liu, X.: GrooMeD-NMS: Grouped mathematically differentiable nms for monocular 3D object detection. In: CVPR (2021) 14
36. Kumar, A., Guo, Y., Huang, X., Ren, L., Liu, X.: SeaBird: Segmentation in Bird's View with Dice Loss Improves Monocular 3D Detection of Large Objects. In: CVPR (2024) 4
37. Kutila, M., Pyykönen, P., Holzhüter, H., Colomb, M., Duthon, P.: Automotive lidar performance verification in fog and rain. In: ITSC (2018) 3
38. Lee, J.H., Han, M.K., Ko, D.W., Suh, I.H.: From big to small: Multi-scale local planar guidance for monocular depth estimation. arXiv preprint arXiv:1907.10326 (2019) 4
39. Li, Y., Yu, A.W., Meng, T., Caine, B., Ngiam, J., Peng, D., Shen, J., Lu, Y., Zhou, D., Le, Q.V., et al.: Deepfusion: Lidar-camera deep fusion for multi-modal 3d object detection. In: CVPR (2022) 3
40. Li, Y., Ibanez-Guzman, J.: LiDAR for autonomous driving: The principles, challenges, and trends for automotive lidar and perception systems. IEEE Signal Processing Magazine (2020) 2
41. Liao, Y., Xie, J., Geiger, A.: KITTI-360: A novel dataset and benchmarks for urban scene understanding in 2D and 3D. TPAMI (2022) 1, 3, 4
42. Lin, J.T., Dai, D., Van Gool, L.: Depth estimation from monocular images and sparse radar data. In: IROS (2020) 4
43. Liu, F., Liu, X.: Voxel-based 3d detection and reconstruction of multiple objects from a single image. In: NeurIPS (2021) 4

44. Ma, X., Zhang, Y., Xu, D., Zhou, D., Yi, S., Li, H., Ouyang, W.: Delving into localization errors for monocular 3D object detection. In: CVPR (2021) 14
45. Martin, D., Fowlkes, C., Malik, J.: Learning to detect natural image boundaries using local brightness, color, and texture cues. TPAMI (2004) 4
46. Menze, M., Geiger, A.: Object scene flow for autonomous vehicles. In: CVPR (2015) 10
47. Michaud, S., Lalonde, J.F., Giguere, P.: Towards characterizing the behavior of lidars in snowy conditions. In: IROS (2015) 3
48. Milanović, V., Kasturi, A., Yang, J., Hu, F.: A fast single-pixel laser imager for VR/AR headset tracking. In: MOEMS and Miniaturized Systems XVI (2017) 2
49. Park, D., Ambrus, R., Guizilini, V., Li, J., Gaidon, A.: Is pseudo-lidar needed for monocular 3d object detection? In: ICCV (2021) 11
50. Piccinelli, L., Sakaridis, C., Yu, F.: iDisc: Internal discretization for monocular depth estimation. In: CVPR (2023) 2
51. Reading, C., Harakeh, A., Chae, J., Waslander, S.: Categorical depth distribution network for monocular 3D object detection. In: CVPR (2021) 2, 3, 14
52. Royo, S., Ballesta-Garcia, M.: An overview of LiDAR imaging systems for autonomous vehicles. Applied sciences (2019) 2
53. Silberman, N., Hoiem, D., Kohli, P., Fergus, R.: Indoor segmentation and support inference from rgbd images. ECCV (2012) 10
54. Spinneker, R., Koch, C., Park, S.B., Yoon, J.J.: Fast fog detection for camera based advanced driver assistance systems. In: ITSC (2014) 3
55. Sun, J., Li, Y., Kang, S.B., Shum, H.Y.: Symmetric stereo matching for occlusion handling. In: CVPR (2005) 4
56. Sun, P., Kretzschmar, H., Dotiwalla, X., Chouard, A., Patnaik, V., Tsui, P., Guo, J., Zhou, Y., Chai, Y., Caine, B., Vasudevan, V., Han, W., Ngiam, J., Zhao, H., Timofeev, A., Ettinger, S., Krivokon, M., Gao, A., Joshi, A., Zhang, Y., Shlens, J., Chen, Z., Anguelov, D.: Scalability in perception for autonomous driving: Waymo open dataset. In: CVPR (2020) 1, 2, 3, 10, 13
57. Uhrig, J., Schneider, N., Schneider, L., Franke, U., Brox, T., Geiger, A.: Sparsity invariant cnns. In: 3DV (2017) 3, 4, 12, 13
58. Wang, J., Zickler, T.: Local detection of stereo occlusion boundaries. In: CVPR (2019) 4
59. Wei, Y., Zhao, L., Zheng, W., Zhu, Z., Rao, Y., Huang, G., Lu, J., Zhou, J.: Surrounddepth: Entangling surrounding views for self-supervised multi-camera depth estimation. In: CoRL (2023) 2
60. Wei, Y., Quan, L.: Asymmetrical occlusion handling using graph cut for multi-view stereo. In: CVPR (2005) 4
61. Weng, J., Ahuja, N., Huang, T.S.: Two-view matching. In: ICCV (1988) 4
62. Wilson, B., Qi, W., Agarwal, T., Lambert, J., Singh, J., Khandelwal, S., Pan, B., Kumar, R., Hartnett, A., Pontes, J.K., et al.: Argoverse 2: Next generation datasets for self-driving perception and forecasting. arXiv preprint arXiv:2301.00493 (2023) 3
63. Wu, H., Wen, C., Li, W., Li, X., Yang, R., Wang, C.: Transformation-equivariant 3D object detection for autonomous driving. In: AAAI (2023) 2, 4
64. Wu, H., Wen, C., Shi, S., Li, X., Wang, C.: Virtual sparse convolution for multimodal 3D object detection. In: CVPR (2023) 2, 4
65. Xie, S., Tu, Z.: Holistically-nested edge detection. In: ICCV (2015) 4
66. Yang, Z., Chen, J., Miao, Z., Li, W., Zhu, X., Zhang, L.: Deepinteraction: 3D object detection via modality interaction. NeurIPS (2022) 2, 4

67. Yu, K., Tao, T., Xie, H., Lin, Z., Liang, T., Wang, B., Chen, P., Hao, D., Wang, Y., Liang, X.: Benchmarking the robustness of lidar-camera fusion for 3D object detection. In: CVPR (2023) 3
68. Yuan, W., Gu, X., Dai, Z., Zhu, S., Tan, P.: Neural window fully-connected crfs for monocular depth estimation. In: CVPR (2022) 4
69. Zheng, K., Li, S., Qin, K., Li, Z., Zhao, Y., Peng, Z., Cheng, H.: Depth estimation via sparse radar prior and driving scene semantics. In: ACCV (2022) 4
70. Zhu, S., Brazil, G., Liu, X.: The edge of depth: Explicit constraints between segmentation and depth. In: CVPR (2020) 2
71. Zhu, S., Liu, X.: Lighteddepth: Video depth estimation in light of limited inference view angles. In: CVPR (2023) 4, 8
72. Zitnick, L., Kanade, T.: A cooperative algorithm for stereo matching and occlusion detection. TPAMI (2000) 4
73. Zou, Z., Ye, X., Du, L., Cheng, X., Tan, X., Zhang, L., Feng, J., Xue, X., Ding, E.: The devil is in the task: Exploiting reciprocal appearance-localization features for monocular 3D object detection. In: ICCV (2021) 14