Image–LiDAR Fusion for Maritime Perception with SimpleBEV-XS

Method, Data, and Updated Results

Nader Nemati

September 8, 2025

Abstract

This report documents an efficient image–LiDAR fusion system for maritime perception built on a simplified BEV encoder (SimpleBEV-XS). The pipeline time-aligns monocular RGB frames with forward LiDAR, rasterizes point clouds into a fixed-field BEV histogram, and trains an efficient fusion head that estimates occupancy/obstacle likelihood over water. Implementation and experiments are conducted on the open-source *Puhang maritime dataset*. With the updated rendering and operating point, the fused head attains per-panel IoUs of **0.93 / 0.90 / 0.94 / 0.94** at thr = 0.60 (mean $\overline{\text{IoU}}$ = **0.9275**) and produces cleaner, more coherent maps with fewer water false positives. Across the sweep, the best epoch reaches **mIoU** 0.866 at thr = 0.75; the exported checkpoint in `metrics.json` reports **mIoU** 0.837 at thr = 0.95. The approach is intentionally calibration-lean and follows the design spirit of *Simple-BEV*, favoring simple lifting and reproducible training choices over heavy machinery, adapted here to maritime sensing with LiDAR fusion.[1]

1 Motivation and Scope

Maritime scenes mix large, texture-poor backgrounds (open water) with small or reflective targets (buoys, small craft, pier edges). In these conditions, monocular appearance is depth-ambiguous and LiDAR returns can be sparse or specular. Projecting both modalities into a common bird's-eye view (BEV) creates a geometric canvas where fusion is straightforward and decisions are easier to interpret. The goal is a clean, fast, and reproducible system that engineers can iterate on: the code path is short, data transformations are explicit, and evaluation is traceable. The design draws on *Simple-BEV* to keep lifting and BEV modeling simple while emphasizing choices that matter in practice for this domain.

2 Data and Preprocessing

Experiments are implemented on the open-source *Puhang maritime dataset*. RGB frames are paired with forward LiDAR using filename timestamps with a constant camera–LiDAR offset and a ±800 ms tolerance. For supervision and auxiliary input, LiDAR point clouds are cropped to a forward metric window and accumulated into a fixed-resolution BEV histogram (hits per cell) with $\log(1+x)$ normalization. In the absence of extrinsics, image features are lifted to BEV by adaptive pooling; when `calib.json` is available, the pipeline switches to a parameter-free voxel sampler (bilinear sampling) in the Simple-BEV style for geometrically grounded lifting. The raster window, axis transforms (`swap_xy`, `flip_x`), and visualization-only blur/dilate are made explicit for reproducibility. For convenience, the repository keeps a compact directory skeleton (e.g., `datasets/nuscenes_toy/{images,lidar}`) to standardize scripts and I/O, while the actual content and splits are drawn from the Puhang data.

[1] Harley et al., "Simple-BEV: What Really Matters for Multi-Sensor BEV Perception?" arXiv:2206.07959 (2022). https://arxiv.org/abs/2206.07959

Table 1: Rasterization and view configuration.

Parameter	Value
Image size (H×W)	256×448
BEV size (H×W)	128×128
Meters (forward × lateral)	$40 \times 30\,\mathrm{m}$
Height band (z)	$[-3, 3]\,\mathrm{m}$
Front FOV (viz)	120°
Unit scale	meters (1.0)
Axis transforms	`swap_xy`, `flip_x`
Viz-only blur/dilate	enabled

3 Model

The network operates in BEV after lifting. An RGB backbone with four stride-2 stages encodes the image; features are lifted to BEV by adaptive pooling or, when calibration is present, a parameter-free voxel sampler that bilinearly samples image features at projected voxel centers and reduces over height. In parallel, the LiDAR stream is the 1-channel BEV histogram. Each stream is compressed and processed by shallow residual blocks in BEV, then concatenated and decoded by a fusion head. The model produces three logits: a fused map (primary output), a camera-only map, and an aux-only (LiDAR) map. Training uses binary cross-entropy with positive-class reweighting plus a soft Dice term; optimization is AdamW (learning rate 5×10^{-4}, weight decay 10^{-2}) with optional mixed precision. To improve robustness, the implementation supports auxiliary sensor dropout, additive noise on the aux input, and occasional gradient detachment of aux features in the fused pass so the fused head remains useful even when the auxiliary signal is degraded. At evaluation, the sigmoid threshold is swept and the maximizer of mIoU is reported as `best_thr`; in addition, micro-IoU, precision/recall, ROC–AUC, latency, and TP/FP/FN are logged for traceability.

4 Results

Rendering the fused head at thr = 0.60 increases IoU across the qualitative panels and improves the visual quality of the maps: dock and edge contours are more coherent, spurious horizontal streaks are suppressed, camera-only responses are less scattered, and the aux-only rings hug the LiDAR ground truth more tightly. Earlier panels were rendered at thr = 0.95; although lowering a threshold often boosts recall, here the IoU also increases, suggesting stronger logits rather than a looser cutoff. For strict comparisons, render both runs at the same threshold or compare each at its `best_thr`.

Table 2: Per-panel IoU for fused maps at thr = 0.60.

Panel	0	1	2	3
IoU	**0.93**	**0.90**	**0.94**	**0.94**

Mean $\overline{\mathrm{IoU}} = \mathbf{0.9275}$.

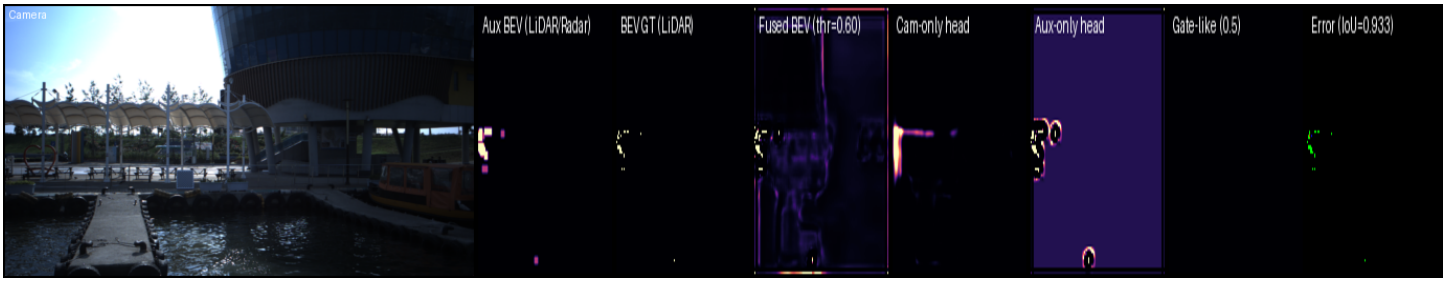
Across the training sweep, the fused head peaks at mIoU 0.8663 at thr = 0.75 (precision 0.8723, recall 0.9919, ROC–AUC 0.9936, latency ~ 236.7 ms). The exported checkpoint in `outputs/rebalanced_fusion/metrics.json` records mIoU 0.8370 at `best_thr` 0.95, with micro-IoU 0.8367, precision 0.8425, recall 0.9919, ROC–AUC 0.9956, and latency 267 ms. For transparency, the corresponding pixel-level counts are: Fusion (TP= 123, FP= 23, FN= 1), Camera (TP= 43, FP= 24, FN= 81), Aux (TP= 124, FP= 99, FN= 0).

5 Discussion

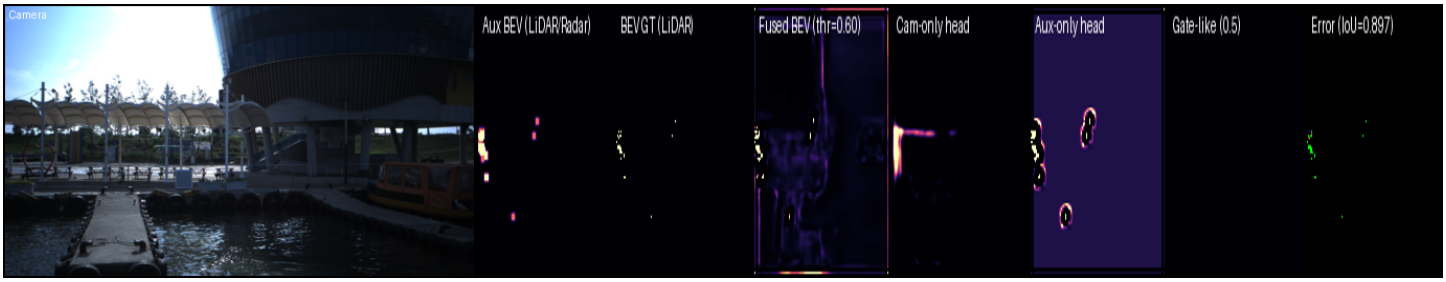
A BEV histogram provides a stable scaffold in low-texture maritime scenes, and a parameter-free lifter (after *Simple-BEV*) keeps the camera path simple and reproducible. Operating without precise extrinsics is practical for field data: adaptive pooling is a strong default, and voxel sampling becomes a drop-in upgrade once calibration is available. The end-to-end system is intentionally streamlined (four CNN stages and shallow BEV trunks), which keeps latency in the $\sim$ 235–270 ms range on our setup and makes ablations frictionless. The fused head shows high recall and strong ROC–AUC, suggesting a stable decision surface; scaling the dataset should mainly trade a small amount of recall for precision and allow a stricter operating point without sacrificing coverage.

6 Qualitative Panels

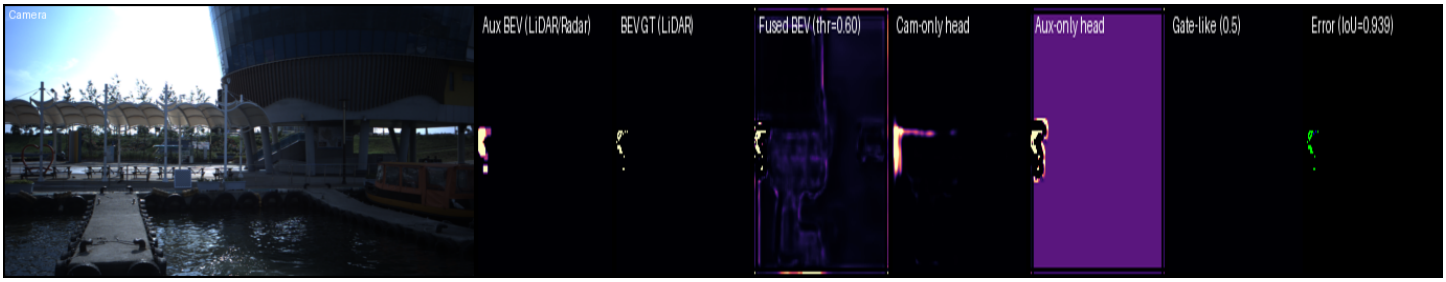
To prevent figures from drifting to the end and creating large blank spaces, the document uses the `float` and `placeins` packages and the `[H]` spec for here-placement. The graphics path points to the repository's visualization folder, so the figure includes remain short. The panels below are rendered at thr = 0.60.



(a) Panel 0 (IoU $\approx$ 0.93).

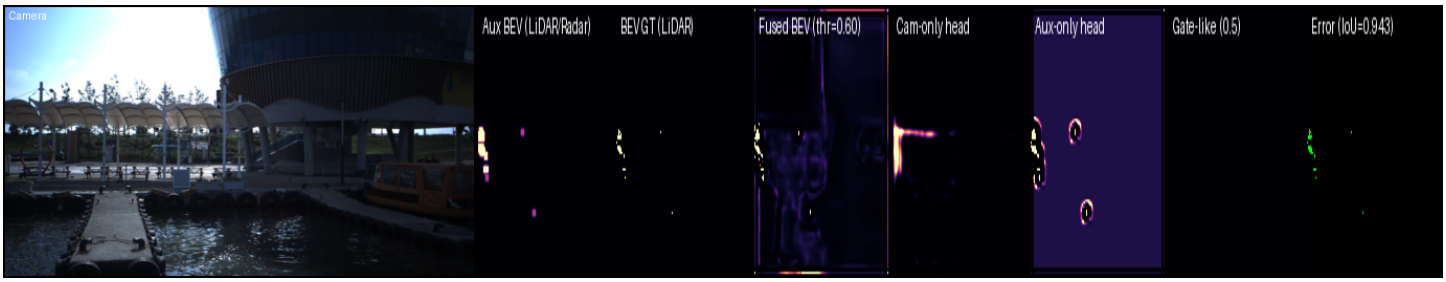


(b) Panel 1 (IoU $\approx$ 0.90).



(c) Panel 2 (IoU $\approx$ 0.94).

Figure 1: Updated validation panels (fused head at thr = 0.60).



(a) Panel 3 (IoU ≈ 0.94).

Figure 2: Updated validation panels (fused head at thr $= 0.60$).

7 Reproduction Snippets

Pairing.

```
python scripts/make_maritime_pairs.py \
  --img_dir ./datasets/nuscenes_toy/images \
  --lidar_dir ~/data/.../lidar_front/points \
  --out_csv datasets/maritime_pairs.csv \
  --max_ms 800 \
  --write_npy --npy_dir datasets/nuscenes_toy/lidar \
  --unit_scale 1.0 --swap_xy --flip_x
```

Training.

```
python -m src.models.simplebev_xs \
  --root ./datasets --out ./outputs/rebalanced_fusion \
  --epochs 50 --batch 4 \
  --img_h 256 --img_w 448 \
  --bev_h 128 --bev_w 128 \
  --meters_x 40 --meters_y 30 \
  --unit_scale 1.0 \
  --z_min -3 --z_max 3 \
  --fov_deg 120 \
  --swap_xy --flip_x \
  --blur_ksize 3 --dilate_ks 3 \
  --viz_p_low 0.5 --viz_p_high 99.5 --viz_gamma 0.6 --viz_colormap
```

References

[1] A. W. Harley, Z. Fang, J. Li, R. Ambrus, and K. Fragkiadaki, "Simple-BEV: What Really Matters for Multi-Sensor BEV Perception?" *arXiv preprint* arXiv:2206.07959, 2022. Available at https://arxiv.org/abs/2206.07959.