Iterative Camera-LiDAR Extrinsic Optimization via Surrogate Diffusion

Ni Ou[1], Zhuo Chen[2], Xinru Zhang[3] and Junzheng Wang[1]

1. School of Automation, Beijing Institute of Technology, Beijing, 100081, China
2. Department of Engineering, King's College London, Strand, London, WC2R 2LS, United Kingdom
3. School of Integrated Circuits and Electronics, Beijing Institute of Technology, Beijing, 100081, China

IROS HANGZHOU 2025

Introduction

Single-Step Model → Multi-Step Model

Naive Iteration: In each iteration, the input to the next iteration is generated by left-multiplying the model output with the input of the current iteration.
Multi-Range iteration: Similar to straightforward iteration, except that a different model is employed at each iteration, specialized for a specific error range.
Internal: Iterative updates of the extrinsics are performed within an *internal* recurrent architecture (e.g., LSTM or GRU), which exploits sequential dependencies.
Diffusion: The diffusion process progressively adds noise to the ground-truth extrinsic matrix to generate an initial noisy estimate, while the denoising process reverses this procedure through a surrogate calibration model.

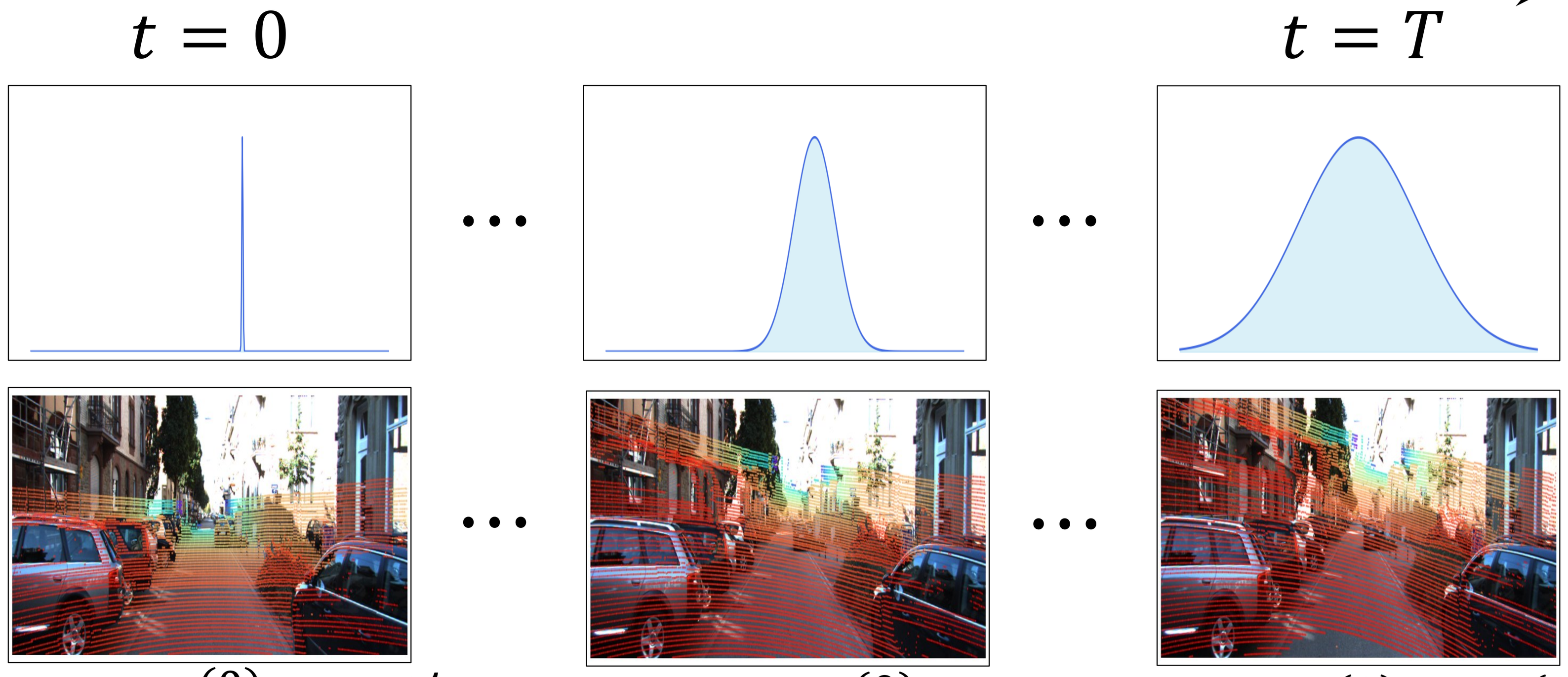
Comparison of Iterative Models

	versatility	accuracy	cost
Naive	Yes	Low	Low
Multi-Range	Yes	High	High
Internal	No	High	Low
Diffusion	Yes	High	Low

Methodology

Noise-Adding: $\boldsymbol{x}_t = \boldsymbol{q}(\boldsymbol{x}_t|\boldsymbol{x}_0,\boldsymbol{\epsilon}), \boldsymbol{\epsilon}\sim\mathcal{N}(0,\boldsymbol{E})$

$t = 0$ … $t = T$



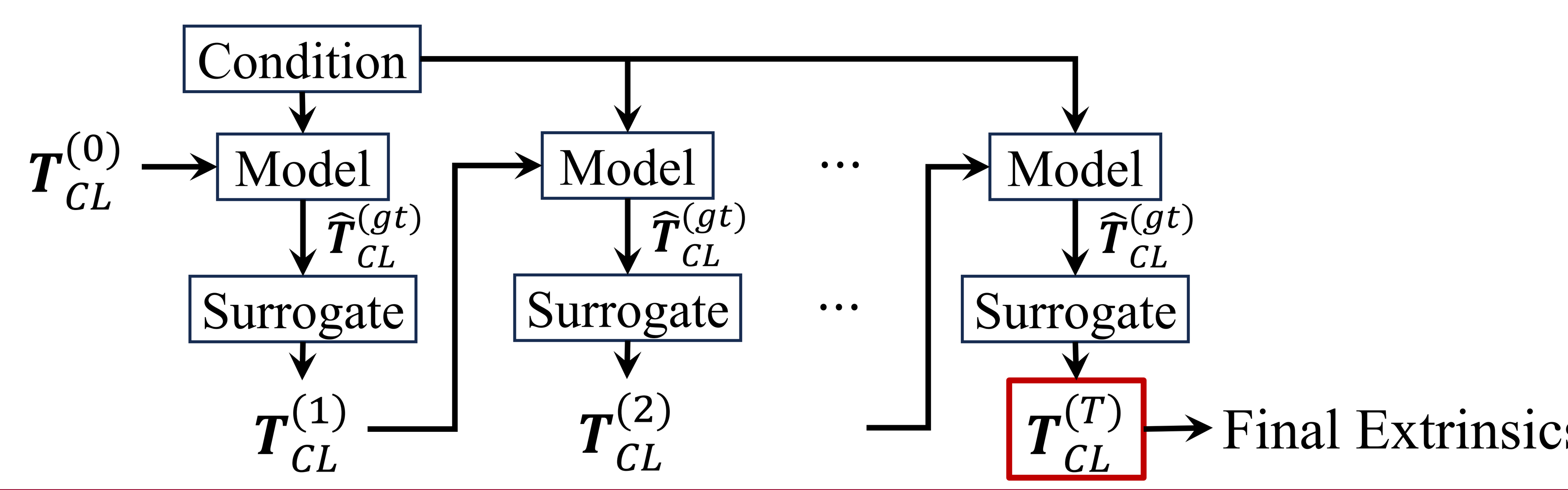
$\mathcal{G}(\boldsymbol{x}_0)\boldsymbol{T}^{(0)}_{CL} = \boldsymbol{T}^{gt}_{CL}$ $\quad$ $\mathcal{G}(\boldsymbol{x}_t)\boldsymbol{T}^{(0)}_{CL}$ $\quad$ $\mathcal{G}(\boldsymbol{x}_T)\boldsymbol{T}^{(0)}_{CL} = \boldsymbol{T}^{(0)}_{CL}$

Denoising: $\boldsymbol{x}_{t-1} = \boldsymbol{q}(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t,\hat{\boldsymbol{x}}_0),\quad \boldsymbol{x}_T = 0$

Condition ($\boldsymbol{C}$): image, point cloud, camera intrinsics

$$\begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \Rightarrow \text{Surrogate:} \begin{cases} \mathcal{G}(\hat{\boldsymbol{x}}_0)\boldsymbol{T}^{(0)}_{CL} = \hat{\boldsymbol{T}}^{(gt)}_{CL} \Rightarrow \hat{\boldsymbol{x}}_0 \\ \boldsymbol{x}_t \sim \mathcal{N}(\sqrt{\bar{\alpha}_t}\hat{\boldsymbol{x}}_0, (1-\bar{\alpha}_t)\boldsymbol{I}) \\ \boldsymbol{x}_{t-1} = \boldsymbol{q}(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t,\hat{\boldsymbol{x}}_0) \\ \boldsymbol{T}^{(i)}_{CL} = \mathcal{G}(\boldsymbol{x}_{T-i})\boldsymbol{T}^{(0)}_{CL} \end{cases}$$

Model: $\hat{\boldsymbol{T}}^{(gt)}_{CL} = \mathcal{G}\left(D_\theta\left(C, \boldsymbol{T}^{(i)}_{CL}\right)\right)\boldsymbol{T}^{(i)}_{CL}$



Results

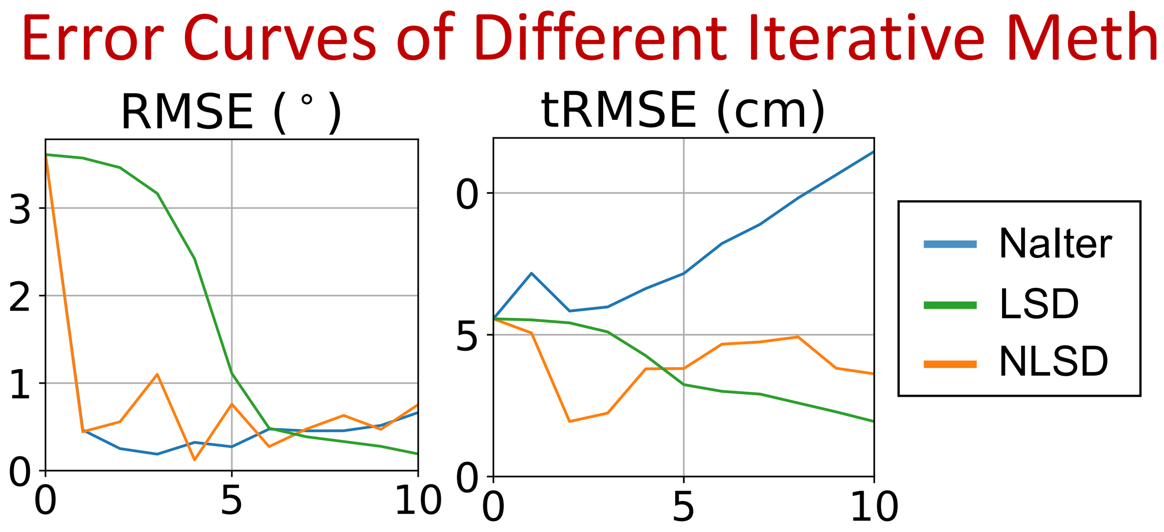
Robustness: Models × Iterative Methods

Method	3°3cm↑	5°5cm↑	ρ%↑
CalibNet (Single)	23.19%	49.37%	N/A
RGGNet (Single)	22.04%	43.53%	N/A
LCCNet (Single)	23.88%	48.47%	N/A
LCCRAFT (Single)	26.38%	47.33%	N/A
CalibNet + NaIter	12.50%	32.75%	2.98%
RGGNet + NaIter	19.65%	39.90%	8.55%
LCCNet + NaIter	13.28%	34.58%	4.74%
LCCRAFT + NaIter	10.39%	27.45%	4.75%
CalibNet + NLSD	23.46%	47.96%	7.66%
RGGNet + NLSD	20.67%	43.04%	6.19%
LCCNet + NLSD	26.15%	48.94%	7.15%
LCCRAFT + NLSD	26.29%	46.74%	7.16%
CalibNet + LSD	**24.39%**	**49.52%**	**38.62%**
RGGNet + LSD	**22.24%**	**44.09%**	**38.86%**
LCCNet + LSD	**26.27%**	**50.14%**	**45.54%**
LCCRAFT + LSD	**27.90%**	**49.96%**	**47.61%**

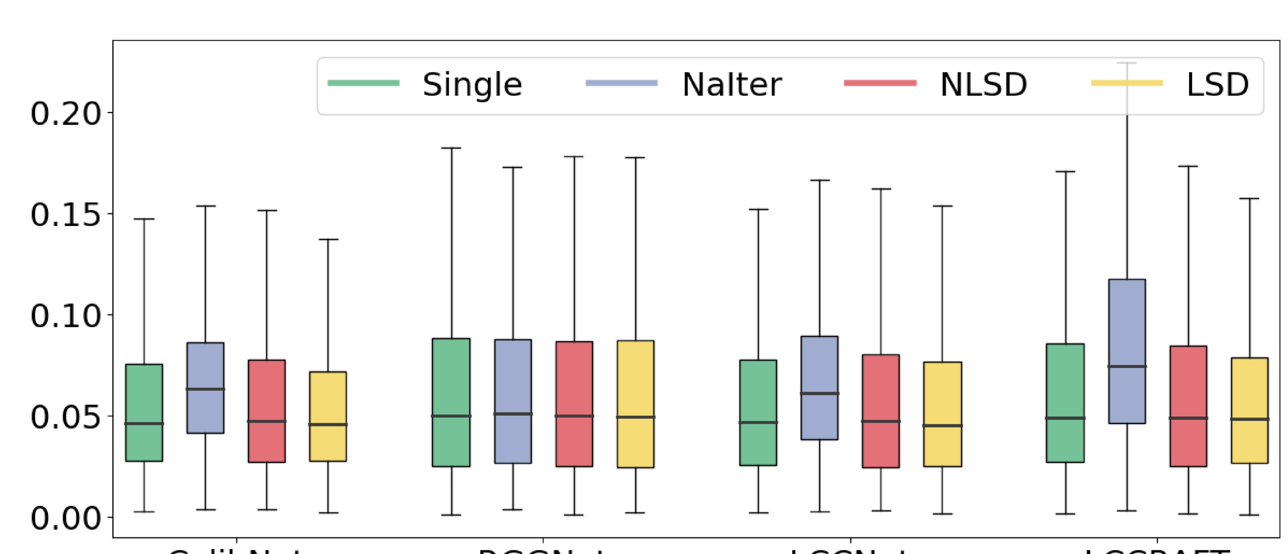
Inference Speed (ms/batch)

Method	Single↓	NaIter↓	NLSD↓	LSD↓
CalibNet	40.67	198.67	226.01	235.11
RGGNet	52.53	321.16	348.10	356.91
LCCNet	65.36	448.07	475.99	483.28
LCCRAFT	381.76	3002.66	3024.40	3097.26

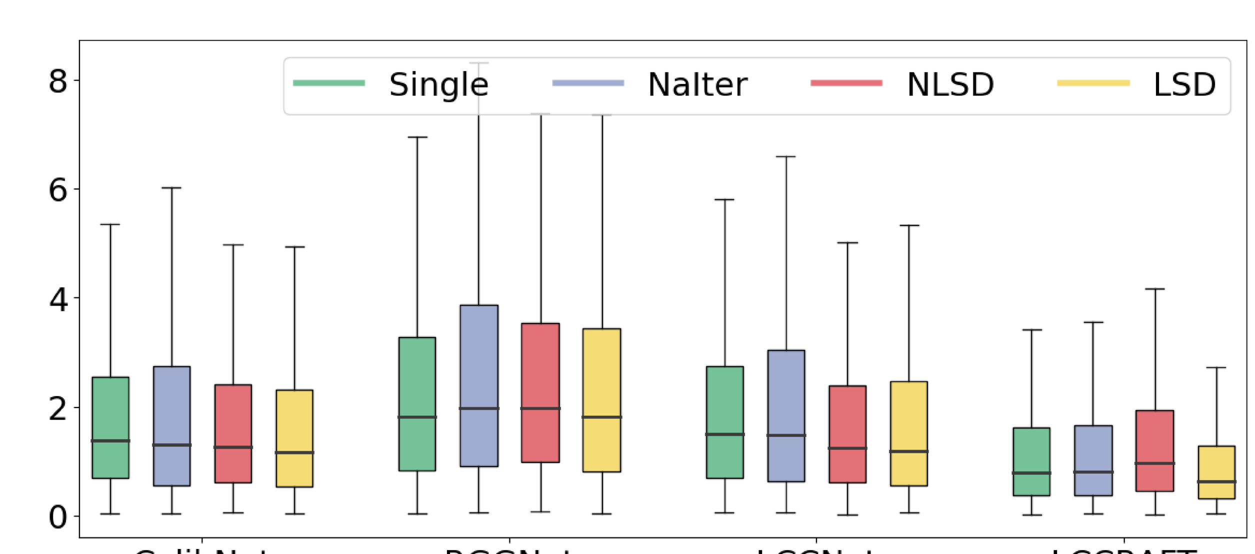
Error Curves of Different Iterative Methods

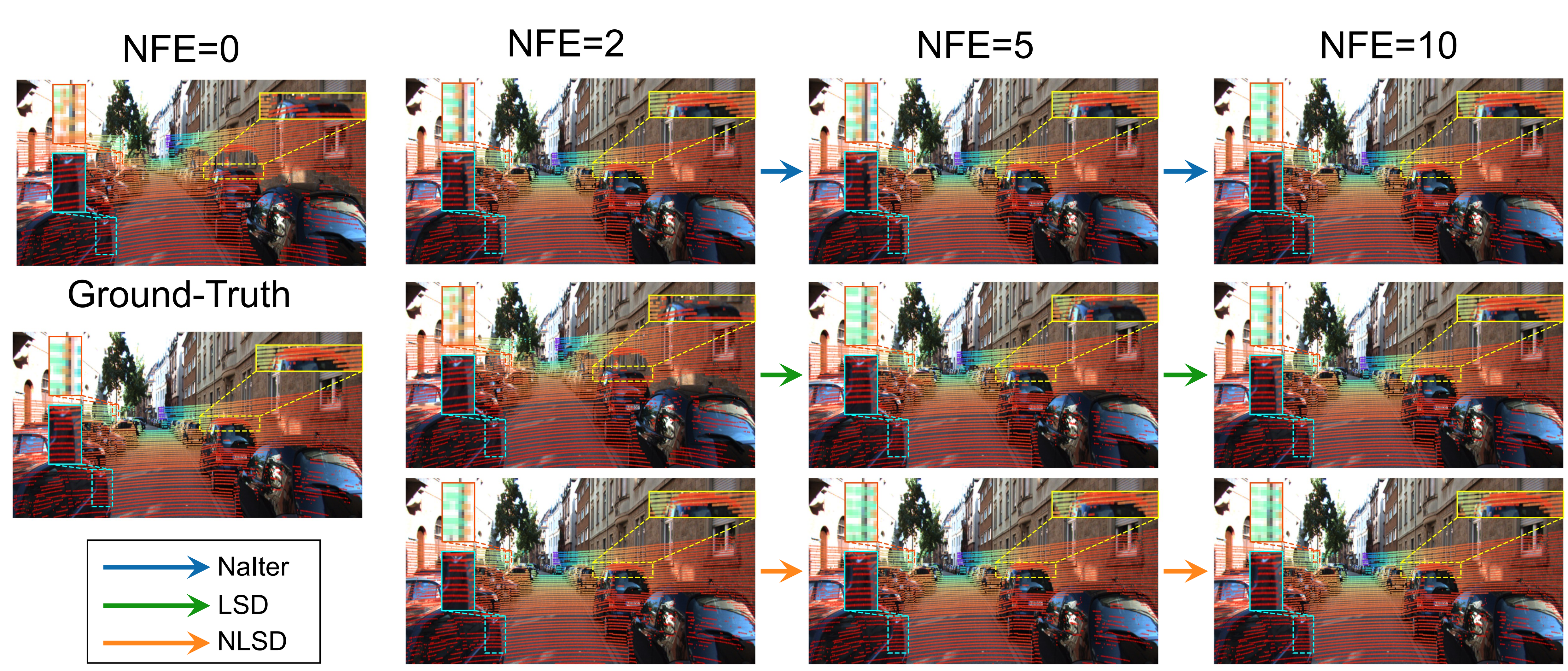


RMSE: Models × Iterative Methods



tMSE: Models × Iterative Methods





Coclusion

We proposed a Linear Surrogate Diffusion (LSD) model for denoiser-agnostic iterative camera-LiDAR calibration. Experiments on the KITTI dataset illustrate that LSD outperforms other single-model iterative methods in terms of accuracy, robustness and stability.