IDP: 3D MOT using Neural Radiance Fields

Student: Burak Cuhadar

Student Number: 03720534

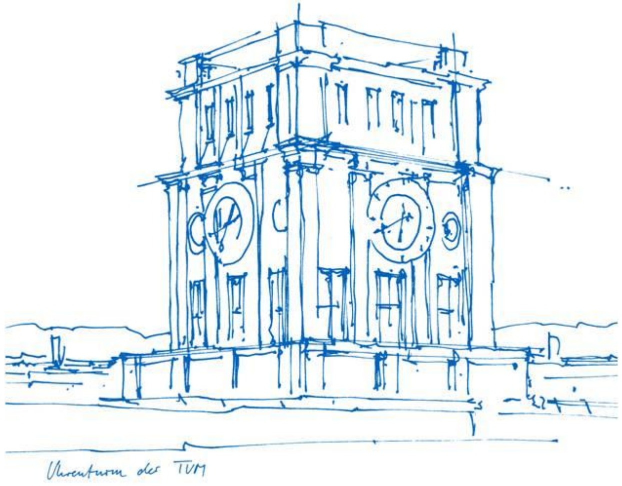
Advisor: Mariia Gladkova

Technische Universität München

TUM School of Computation, Information and Technology

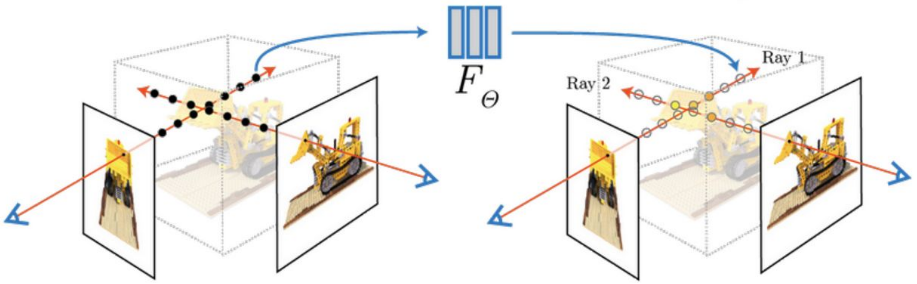
Chair of Computer Vision

Munich, 21. March 2024

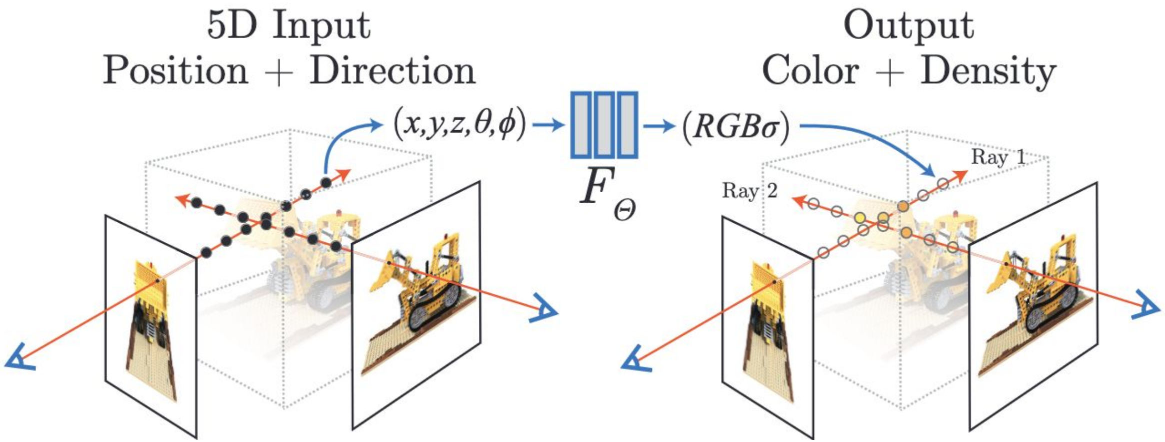


Motivation

- Success of Radiance Fields methods: NeRFs, 3D Gaussian Splatting etc.
- Static scene assumption
- Dynamic scenes, individual objects



Related Work: NeRF

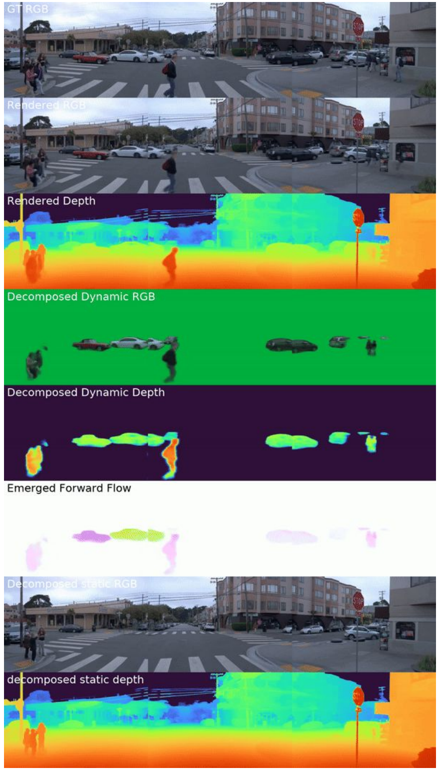


$$C(\mathbf{r}) = \int_{t_n}^{t_f} T(t)\sigma(\mathbf{r}(t))\mathbf{c}(\mathbf{r}(t),\mathbf{d})dt\,, \text{ where } T(t) = \exp\left(-\int_{t_n}^{t}\sigma(\mathbf{r}(s))ds\right)$$

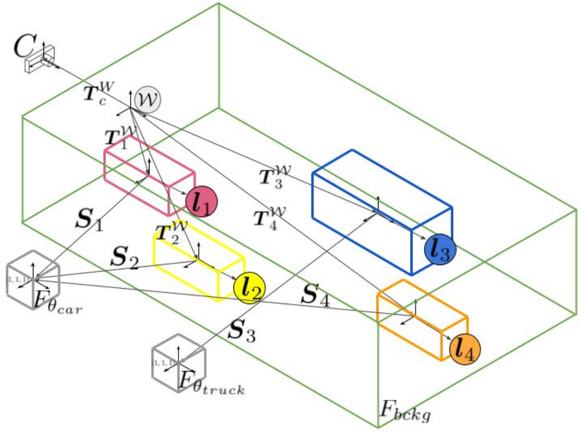
Related Work: NeRFs with Deformation Field

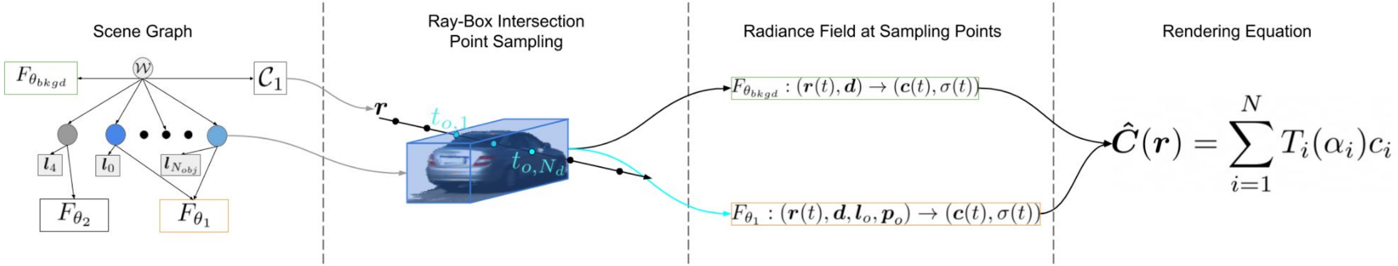
- works on monocular camera
- Deformation Field
- D-NeRF, Nerfies, HyperNeRF

Related Work: NeRFs for Scene Decomposition

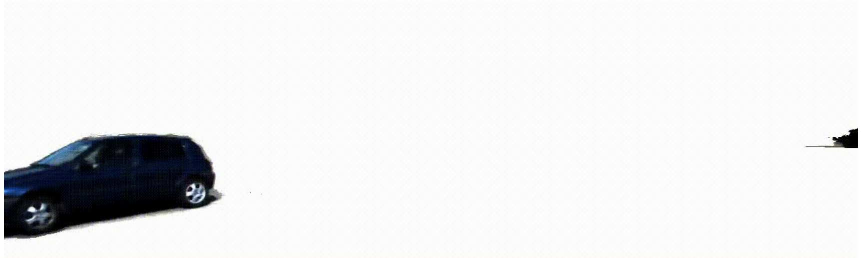


Related Work: Neural Scene Graphs

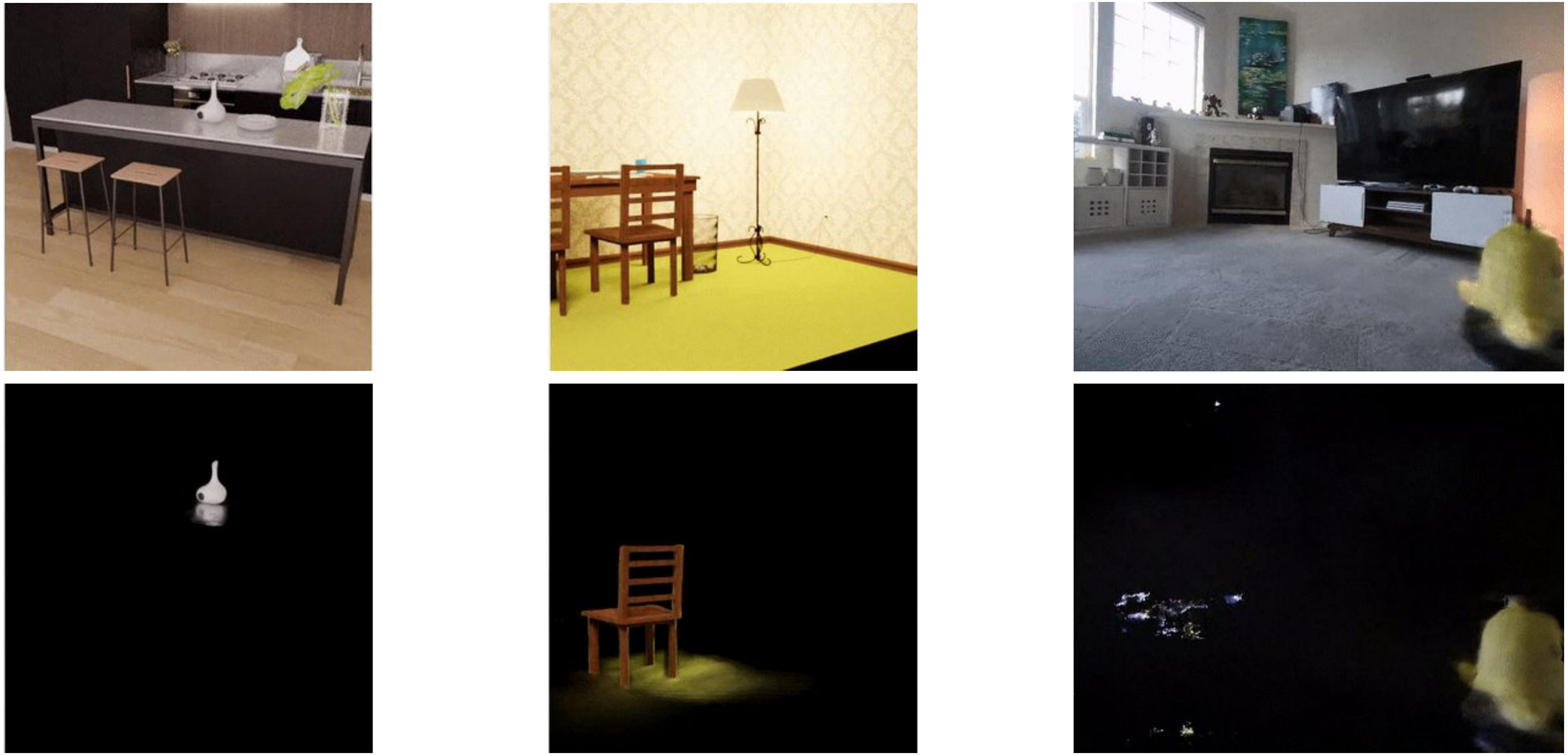




Related Work: Neural Scene Graphs



Related Work: STaR

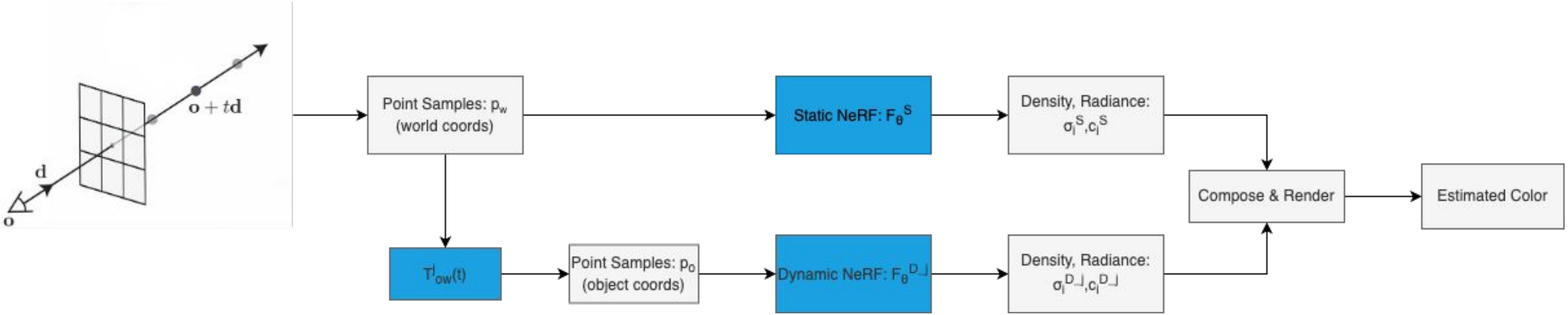


Related Work

	Dynamic Scenes	Scene Decomposition	Object Tracking
NeRFs with deformation field: D-NeRF, Nerfies, HyperNeRF	✓		
NeRFs for Scene Decomposition: D2NeRF, EmerNeRF	✓	✓	
Neural Scene Graph	✓	✓	✓ (uses off-the-shelf 3D Tracker)
STaR	✓	✓	✓

- Ours:
    - Rigid object tracking
    - Supports multiple objects
    - Decomposes each object individually implicitly

Method



$$p_o = T^j_{ow}(t) * p_w,\ where\ T^j_{ow}(t) \in SE(3)$$

Method: Composition

$$\hat{C}(\mathbf{r}) = \sum_{i=1}^{N} T_i(\alpha_i^S \mathbf{c}_i^S + \sum_{j=1}^{V} \alpha_i^{D_j} \mathbf{c}_i^{D_j})$$

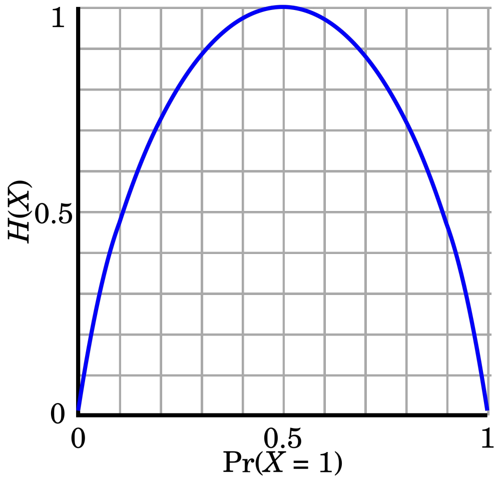
$$\text{where } T_i = \exp\left(-\sum_{j=1}^{i-1}(\sigma_j^S + \sum_{k=1}^{V} \sigma_j^{D_k})(s_{j+1} - s_j)\right)$$

$$\alpha_i^S = 1 - \exp(-\sigma_i^S(s_{i+1} - s_i)),$$
$$\alpha_i^{D_j} = 1 - \exp(-\sigma_i^{D_j}(s_{i+1} - s_i))$$

Method: Loss

$$\mathcal{L} = \mathcal{L}_{RGB} + \beta\mathcal{L}_{transparency} + \gamma\mathcal{L}_{decomposition} + \eta\mathcal{L}_{static} + \lambda\mathcal{L}_{ray}$$

- $\mathcal{L}_{transparency} = \sum_{i=1}^{M}(\mathcal{H}(\alpha_i^S) + \sum_{j=1}^{V}\mathcal{H}(\alpha_i^{D_j}))$
- $\mathcal{L}_{decomposition} = \left(\overline{\alpha}_i^S \log \overline{\alpha}_i^S + \overline{\alpha}_i^D \log \overline{\alpha}_i^D\right)\left(\alpha_i^S + \alpha_i^D\right)$ where $\overline{\alpha}_i^S = \alpha_i^S / (\alpha_i^S + \alpha_i^D)$



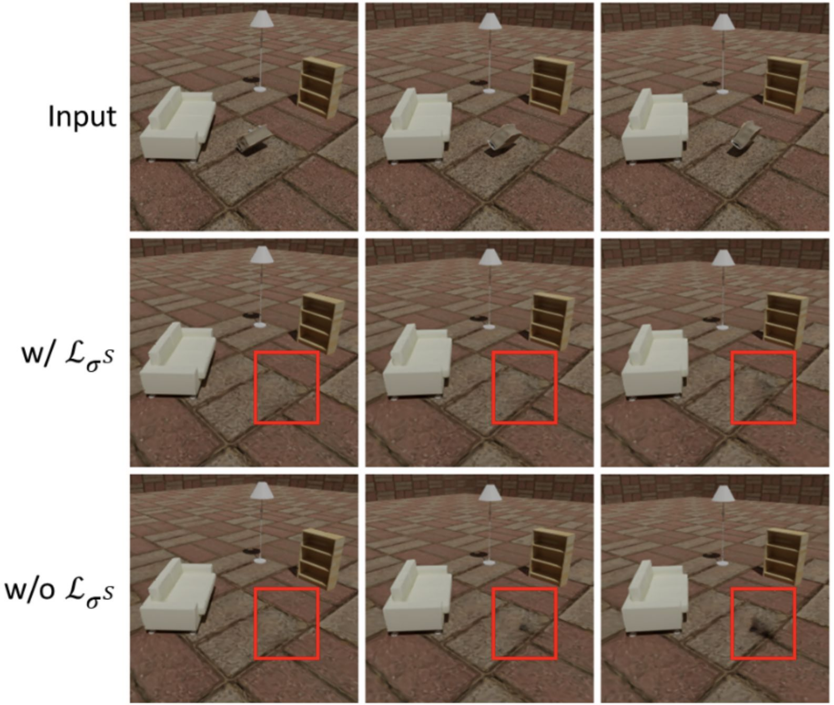
Method: Loss (from D2NeRF)

$$\mathcal{L}_{static} = -\sum_{i=1}^{N} p\left(\mathbf{r}_i\right) \log p\left(\mathbf{r}_i\right)$$

$$\text{,where } p\left(\mathbf{r}_i\right) = \frac{\alpha_i}{\sum_j \alpha_j} = \frac{1 - \exp\left(-\sigma_i \delta_i\right)}{\sum_j 1 - \exp\left(-\sigma_j \delta_j\right)}$$

$$\mathcal{L}_{ray}^{j}\left(\mathbf{r}\right) = \max_{t \in [t_n, t_f]} w^j\left(\mathbf{r}(t)\right)$$

$$\text{,where } w^j\left(\mathbf{x}\right) = \frac{\sigma^{D_j}\left(\mathbf{x}\right)}{\sum_i \sigma^{D_i}\left(\mathbf{x}\right) + \sigma^S(\mathbf{x})} \in [0, 1]$$



Method: Optimization

- Rigid pose optimization with PyPose library
- Pose initialization
    - Translation noise ~ $N(0,1)$
    - Rotation noise around y-axis ~ $N(\pi/32, \pi/16)$
- 3-stage optimization:
    - Appearance Initialization
        - until MSE loss of 9e-4
    - Optimization for the first k frames
        - until MSE loss of 1e-3
    - Online training
        - until MSE loss of 1e-3 and minimum 70k iterations

Dataset

- synthetic dataset using CARLA
- 50 views for training, 6 for validation, 12 for testing
- two datasets: one-vehicle(16 frames) and two-vehicle(12 frames)

Experiments: Novel-View Synthesis

	Sequence	One-vehicle			Two-vehicle		
	Metric	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
Composition	NeRF-time	**26.29**	0.869	0.321	**23.73**	**0.833**	**0.313**
	STaR [24]	25.98	0.871	0.312	23.40	0.818	0.333
	Ours	26.23	**0.874**	**0.306**	23.65	0.829	0.314
	Sequence	One-vehicle			Two-vehicle		
	Metric	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
Static	NeRF-time	**26.55**	0.871	0.316	**23.81**	**0.834**	**0.307**
	STaR [24]	26.32	0.873	0.306	23.65	0.821	0.322
	Ours	26.43	**0.875**	**0.302**	23.75	0.830	0.308
	Sequence	One-vehicle			Two-vehicle		
	Metric	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
Dynamic	NeRF-time	17.64	0.596	**0.004**	**19.62**	0.659	**0.006**
	STaR [24]	17.14	0.583	0.055	15.98	0.492	0.056
	Ours	**18.58**	**0.665**	0.033	19.31	**0.661**	0.045

Experiments: Novel-View Synthesis

Ground-Truth

Ours

Nerf-Time

Experiments: Novel-View Synthesis

Ground-Truth

Ours

Nerf-Time

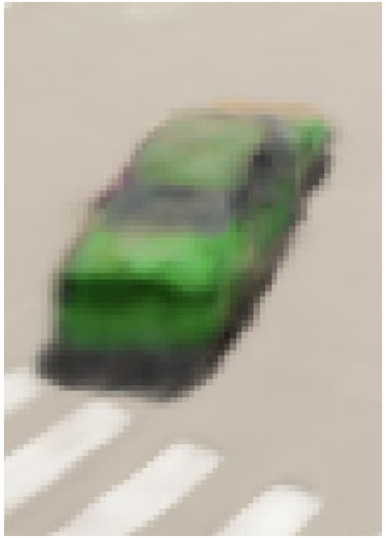
Experiments: Novel-View Synthesis

Ground-Truth | Ours | Nerf-Time | STaR

Experiments: Novel-View Synthesis

Ground-Truth

Ours

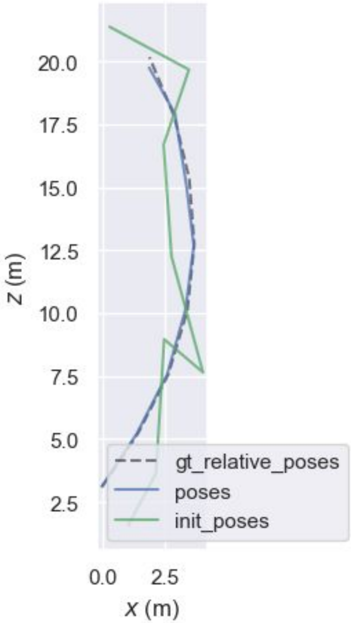


Nerf-Time

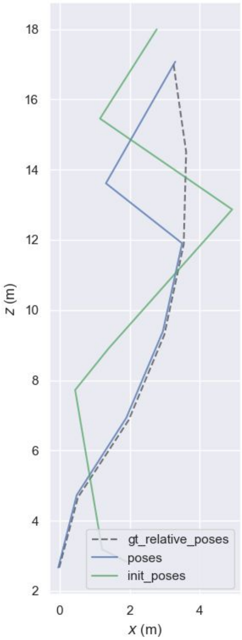
STaR

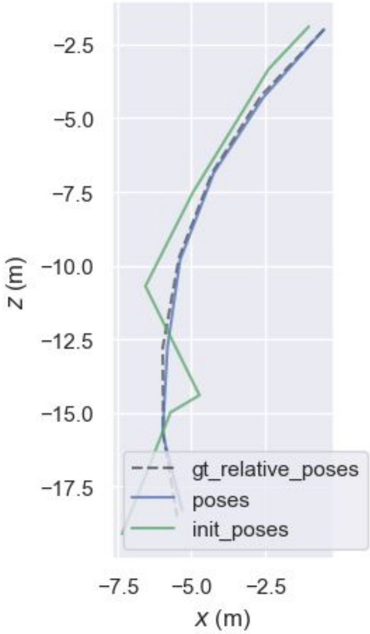
Experiments: Pose Estimation

One-vehicle Dataset



Two-vehicle Dataset



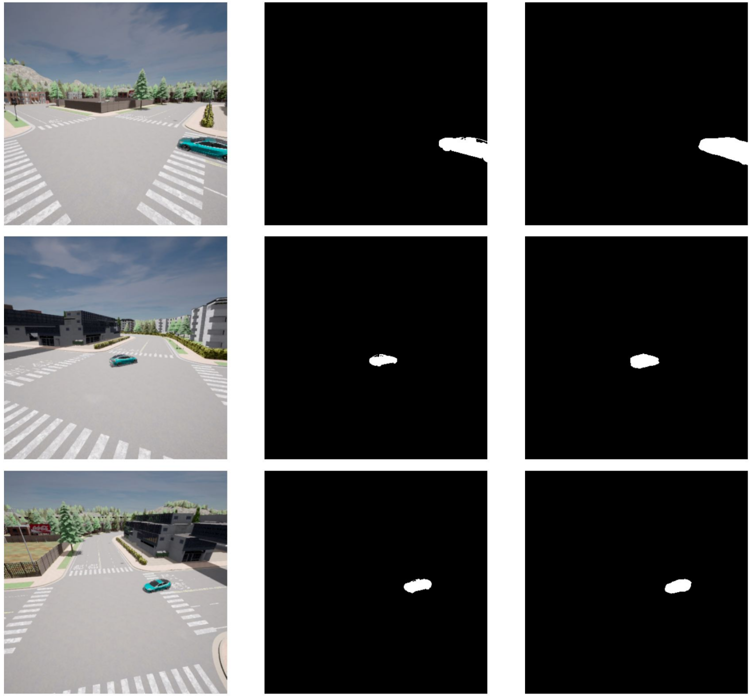


Experiments: Pose Estimation

	One-vehicle	Two-vehicle		
		Mean	First car	Second car
ATE	0.146	0.424	0.540	0.308
RPE	0.182	0.769	1.271	0.267

	Mean	First Vehicle	Second Vehicle
3D IOU:	0.924	0.932	0.917

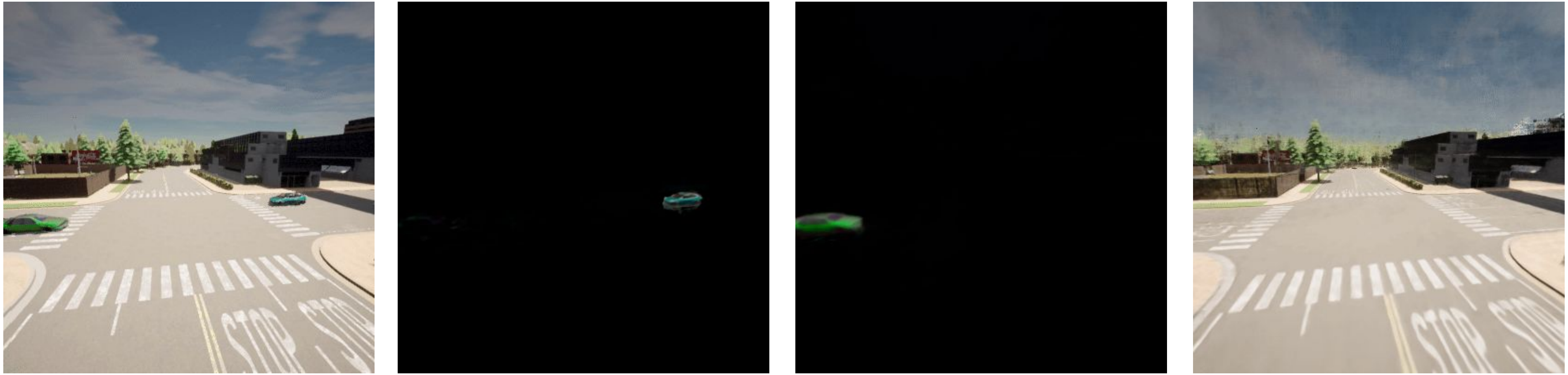
Experiments: Object Decomposition



G.T. RGB G.T. Mask Estimated Mask

	One-vehicle	Two-vehicle
2D IOU	0.79	0.67

Experiments: Object Decomposition



Experiments: Ablation Study

	Metric	PSNR ↑	SSIM ↑	LPIPS ↓
Composition	Ours(only entropy reg.)	23.40	0.818	0.333
	Ours(entropy + dynamic reg.)	23.39	0.818	0.332
	Ours(entropy + ray reg.)	23.51	0.824	0.320
	Ours(entropy + static reg.)	23.62	0.828	0.316
	Ours	**23.65**	**0.829**	**0.314**
	Metric	PSNR ↑	SSIM ↑	LPIPS ↓
Static	Ours(only entropy reg.)	23.65	0.821	0.322
	Ours(entropy + dynamic reg.)	23.63	0.821	0.321
	Ours(entropy + ray reg.)	23.69	0.826	0.312
	Ours(entropy + static reg.)	23.71	0.829	0.310
	Ours	**23.75**	**0.830**	**0.308**
	Metric	PSNR ↑	SSIM ↑	LPIPS ↓
Dynamic	Ours(only entropy reg.)	15.98	0.492	0.056
	Ours(entropy + dynamic reg.)	16.07	0.498	0.056
	Ours(entropy + ray reg.)	17.33	0.575	**0.042**
	Ours(entropy + static reg.)	19.29	0.658	0.050
	Ours	**19.31**	**0.661**	0.045

Experiments: Ablation Study

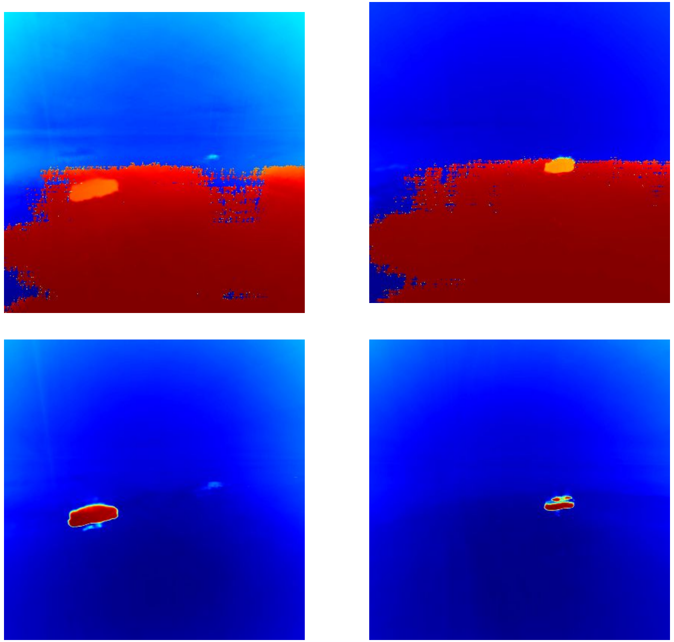
w/o static reg.

with static reg.

Ground-Truth RGB

farther

closer



w/o ray reg.

with ray reg.

Conclusion

- Adapted NeRF for dynamic scenes with rigid objects
- Limitations:
  - longer sequences
  - fixed number of objects
- Future work:
  - real-world datasets
  - ego-vehicle camera
  - adapt to changing number of vehicles