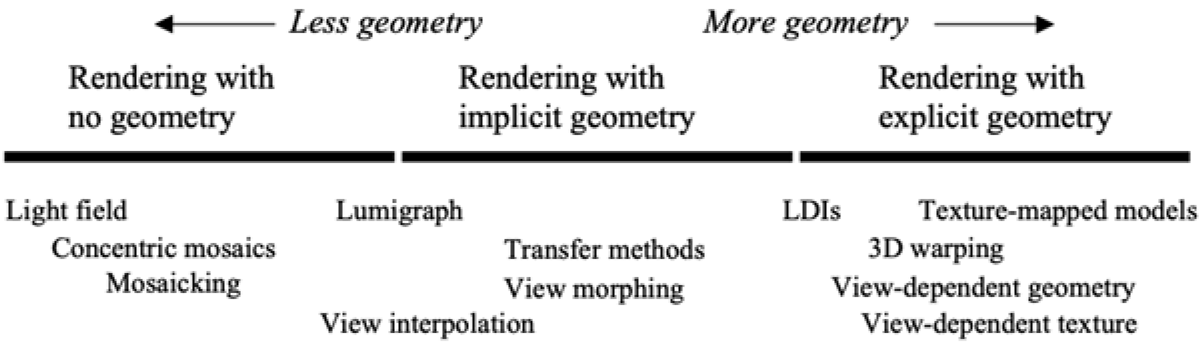
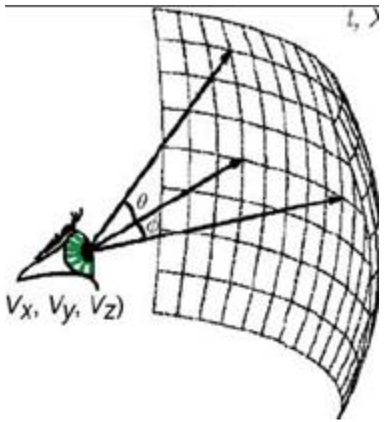
DynIBaR Neural Dynamic Image-Based Rendering

Zhengqi Li, Qianqian Wang, Forrester Cole, Richard Tucker, Noah Snavely
CVPR 2023, Award Candidate

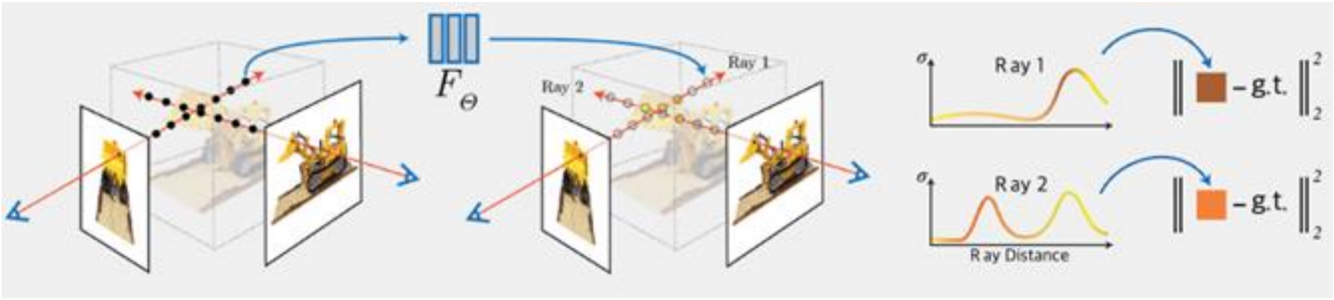
Weekly Meeting - 2023-07-28
KAIST Geometric AI Lab - Jaihoon Kim

Novel view synthesis

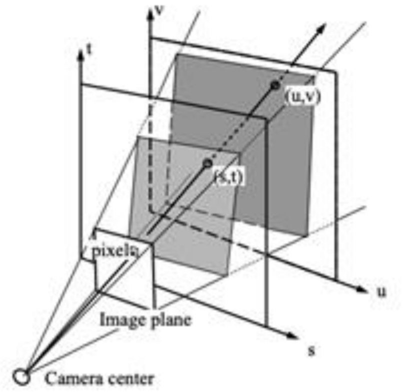




Plenoptic function



NeRF



Lumigraph

Mildenhall, Ben, et al. "Nerf: Representing scenes as neural radiance fields for view synthesis." Communications of the ACM 65.1 (2021): 99-106.
Gortler, Steven J., et al. "The lumigraph." Proceedings of the 23rd annual conference on Computer graphics and interactive techniques. 1996.

What about dynamic scenes ?

DNeRF

NSFF

HyperNeRF

Pumarola, Albert, et al. "D-nerf: Neural radiance fields for dynamic scenes." *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021.
Li, Zhengqi, et al. "Neural scene flow fields for space-time view synthesis of dynamic scenes." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2021.
Park, Keunhong, et al. "Hypernerf: A higher-dimensional representation for topologically varying neural radiance fields." *arXiv preprint arXiv:2106.13228* (2021).

DynlBaR teaser

Problem Definition

Objective

Synthesize novel view image from a dynamic video with

i) long time duration

ii) unbounded scene

iii) complex camera trajectories and scene motion

Input

Front-facing dynamic scene videos with synchronized multi-view cameras

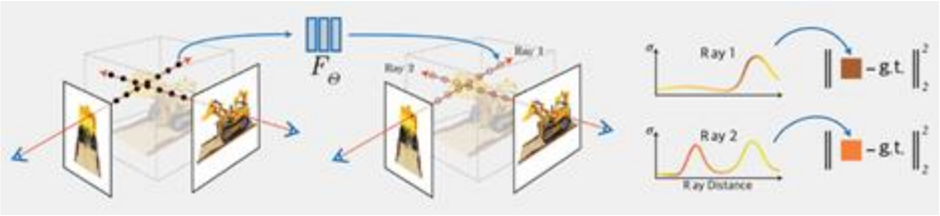
- N frames with camera poses

Output

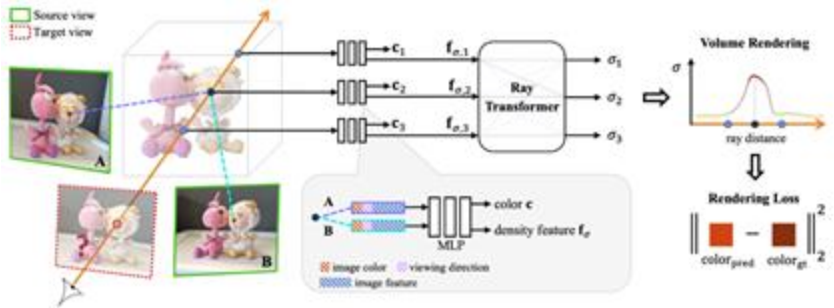
High quality, spatiotemporal consistent image at an arbitrary sampled pose and time

Related Work (i)
- **Novel view synthesis**

Extend static scene reconstruction ideas to dynamic scenes
- NeRF: Volume rendering from a 3D scene encoded in a MLP
- IBRNet: Combines classical IBR method with volume rendering



NeRF



IBRNet

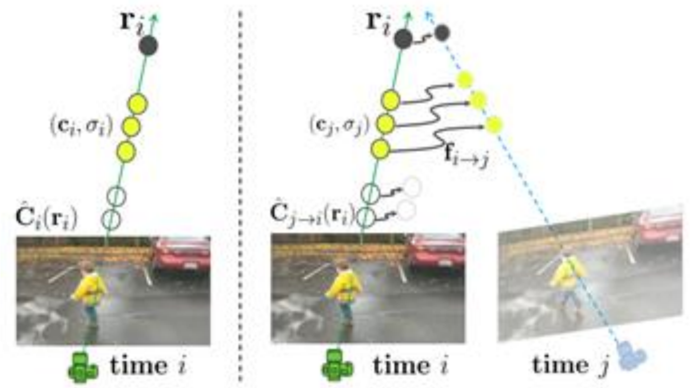
Wang, Qianqian, et al. "Ibrnet: Learning multi-view image-based rendering." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2021.

Related Work (ii)

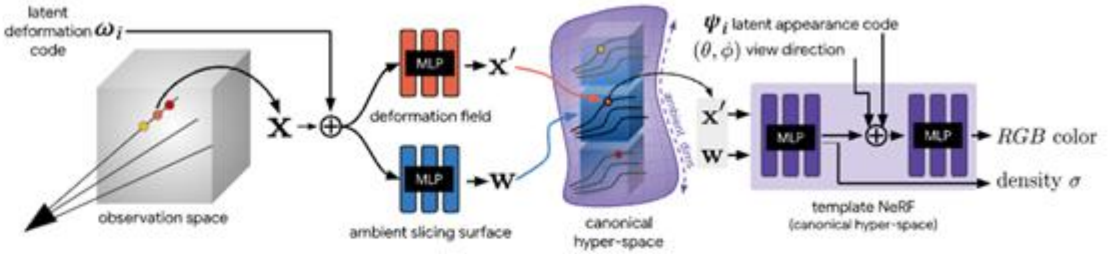
- **3D dynamic scene reconstruction**

Previous works fails to synthesize highly dynamic scenes with complex camera trajectories

- NSFF: Motion prediction cannot scale out to videos with long sequences
- HyperNeRF: Canonicalization is confined to object centric scene with controlled camera poses



NSFF



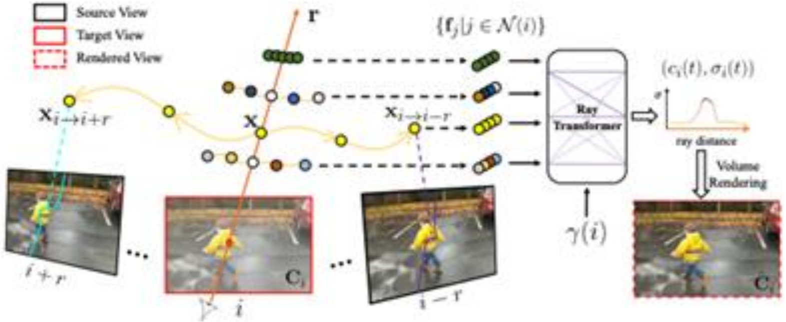
HyperNeRF

Proposed Method (i)

- **Motion-adjusted feature aggregation**

Dynamic scene

→ Epipolar constraint is violated



Proposed Method (i)

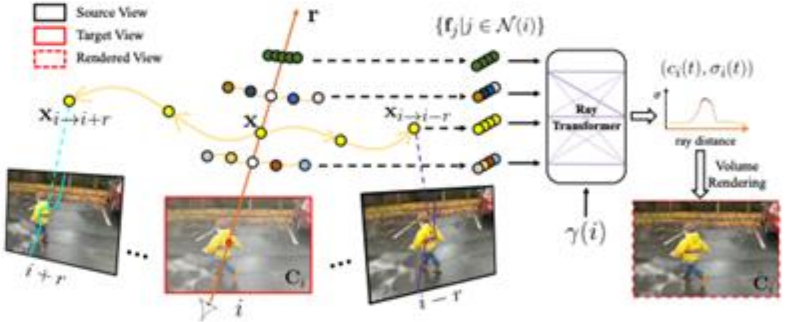
- **Motion-adjusted feature aggregation**

Dynamic scene

→ Epipolar constraint is violated

Estimate a scene flow: High computation (NSFF)

→ Learn motion trajectories using trainable basis functions



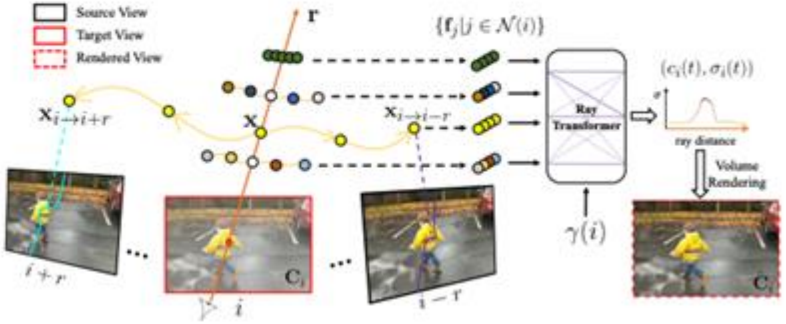
Proposed Method (i)

- **Motion-adjusted feature aggregation**

Dynamic scene
→ Epipolar constraint is violated

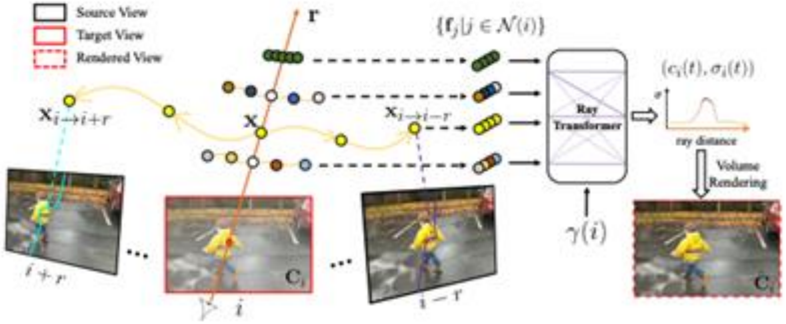
Estimate a scene flow: High computation (NSFF)
→ Learn motion trajectories using trainable basis functions

Jointly optimize
Basis coefficient: $\{\phi_i^l(\mathbf{x})\}_{l=1}^{L} = G_{\mathrm{MT}}(\gamma(\mathbf{x}), \gamma(i))$ $\phi_i^l \in \mathcal{R}^3$
Global motion basis (DCT) : $\{h_i^l\}_{l=1}^{L}$ $h_i^l \in \mathcal{R}.$



Proposed Method (i)

- **Motion-adjusted feature aggregation**



Dynamic scene

→ Epipolar constraint is violated

Estimate a scene flow: High computation (NSFF)

→ Learn motion trajectories using trainable basis functions

Jointly optimize

Basis coefficient: $\{\phi_i^l(\mathbf{x})\}_{l=1}^{L} = G_{\mathrm{MT}}(\gamma(\mathbf{x}), \gamma(i))$ $\quad \phi_i^l \in \mathcal{R}^3$

Global motion basis (DCT) : $\{h_i^l\}_{l=1}^{L}$ $\quad h_i^l \in \mathcal{R}$.

Motion trajectory $\Gamma_{\mathbf{x},i}(j) = \sum_{l=1}^{L} h_j^l \phi_i^l(\mathbf{x})$

Relative displacement is defined as $\Delta_{\mathbf{x},i}(j) = \Gamma_{\mathbf{x},i}(j) - \Gamma_{\mathbf{x},i}(i)$

Proposed Method (ii)

- **Cross-time rendering for temporal consistency**

Using naive photometric loss fails temporal consistency

→ Render a view at time i via some nearby time j

Proposed Method (ii)

- **Cross-time rendering for temporal consistency**

Using naive photometric loss fails temporal consistency

→ Render a view at time i via some nearby time j

Given $\mathbf{x}_{i\to j}$

Query a new trajectory $\{\phi_j^l(\mathbf{x}_{i\to j}\}_{l=1}^L = G_{\mathrm{MT}}(\mathbf{x}_{i\to j}, \gamma(j))$ $(\mathbf{x}_{i\to j})_{j\to k}$

Proposed Method (ii)

- **Cross-time rendering for temporal consistency**

Using naive photometric loss fails temporal consistency

→ Render a view at time i via some nearby time j

Given $\mathbf{x}_{i\to j}$

Query a new trajectory $\{\tilde{\phi}_j^l(\mathbf{x}_{i\to j})\}_{l=1}^L = G_{\mathrm{MT}}(\mathbf{x}_{i\to j}, \gamma(j))$ $(\mathbf{x}_{i\to j})_{j\to k}$

Render image at j using images k in the temporal window at j timestep

Volume render $(\mathbf{c}_j, \sigma_j)$ to from a color $\hat{\mathbf{C}}_{j\to i}$

Proposed Method (ii)

- **Cross-time rendering for temporal consistency**

Using naive photometric loss fails temporal consistency

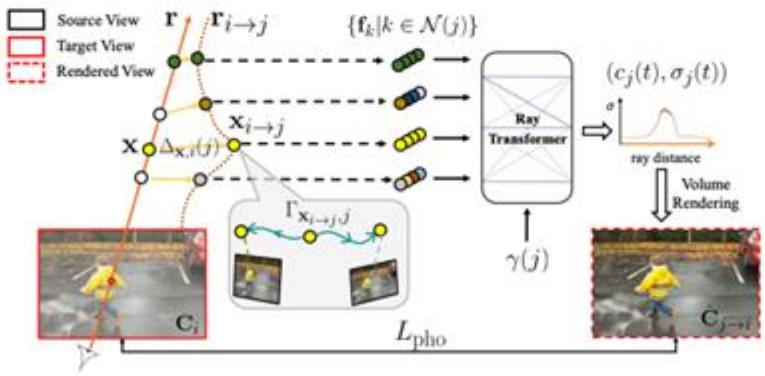
→ Render a view at time i via some nearby time j

Given $\mathbf{x}_{i\to j}$

Query a new trajectory $\{\phi_j^l(\mathbf{x}_{i\to j})\}_{l=1}^{L} = G_{\mathrm{MT}}(\mathbf{x}_{i\to j}, \gamma(j))$ $\quad (\mathbf{x}_{i\to j})_{j\to k}$

Render image at j using images k in the temporal window at j timestep

Volume render $(\mathbf{c}_j, \sigma_j)$ to from a color $\hat{\mathbf{C}}_{j\to i}$



Proposed Method (ii)

- **Cross-time rendering for temporal consistency**

Using naive photometric loss fails temporal consistency

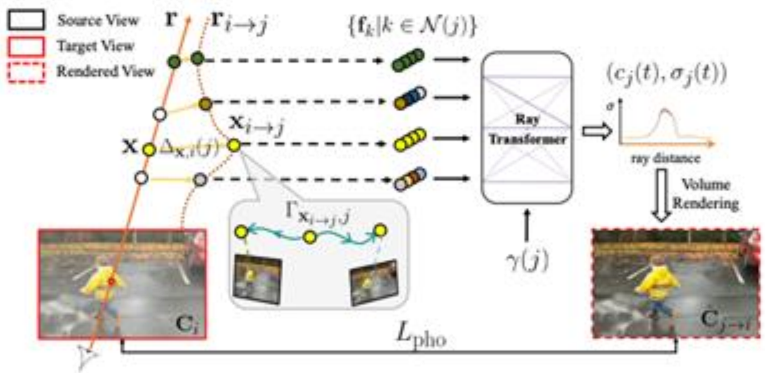
→ Render a view at time i via some nearby time j

Given $\mathbf{x}_{i\to j}$

Query a new trajectory $\{\phi^l_j(\mathbf{x}_{i\to j}\}_{l=1}^{L} = G_{\mathrm{MT}}(\mathbf{x}_{i\to j}, \gamma(j))$ $(\mathbf{x}_{i\to j})_{j\to k}$

Render image at j using images k in the temporal window at j timestep

Volume render $(\mathbf{c}_j, \sigma_j)$ to from a color $\hat{\mathbf{C}}_{j\to i}$



$$\mathcal{L}_{\mathrm{pho}} = \sum_{\mathbf{r}} \sum_{j\in\mathcal{N}(i)} \hat{\mathbf{W}}_{j\to i}(\mathbf{r})\rho(\mathbf{C}_i(\mathbf{r}), \hat{\mathbf{C}}_{j\to i}(\mathbf{r}))$$

Motion disocclusion weight

$$\hat{\mathbf{W}}_{j\to i}(\mathbf{r}) = 1 - \int_{t_n}^{t_f} w_{i\to j}\mathbf{r}(t)\, dt$$

$$w_{i\to j}(\mathbf{r}(t)) = T_i(\mathbf{r}(t))\alpha(\sigma_i(\mathbf{r}(t))) - T_j(\mathbf{r}(t))\alpha(\sigma_j(\mathbf{r}(t)))$$

$\rho(\mathbf{C}_i(\mathbf{r}), \hat{\mathbf{C}}_{j\to i}(\mathbf{r}))$ Charbonnier loss $\rho(x) = \sqrt{x^2 + \varepsilon^2}$

Proposed Method (iii)

- **Combining static and dynamic models**

Small temporal window is not suitable for large camera variations

→ Model the entire into two scenes, time-varying and time-invariant model

$$\mathcal{L}_{\text{pho}} = \sum_{\mathbf{r}} \sum_{j \in \mathcal{N}(i)} \hat{\mathbf{W}}_{j \to i}(\mathbf{r}) \rho(\mathbf{C}_i(\mathbf{r}), \hat{\mathbf{C}}^{\text{full}}_{j \to i}(\mathbf{r}))$$

Proposed Method (iii)

- **Combining static and dynamic models**

Small temporal window is not suitable for large camera variations

→ Model the entire into two scenes, time-varying and time-invariant model

$$\mathcal{L}_{\text{pho}} = \sum_{\mathbf{r}} \sum_{j \in \mathcal{N}(i)} \hat{\mathbf{W}}_{j \to i}(\mathbf{r}) \rho(\mathbf{C}_i(\mathbf{r}), \hat{\mathbf{C}}^{\text{full}}_{j \to i}(\mathbf{r}))$$

Separate static and dynamic components

→ Train a lightweight segmentation model to separate static and dynamic parts

$\hat{\mathbf{B}}_i^{\text{dy}}, \boldsymbol{\alpha}_i^{\text{dy}}, \boldsymbol{\beta}_i^{\text{dy}} = D(I_i)$ opacity map $\boldsymbol{\alpha}_i^{\text{dy}}$, confidence map $\boldsymbol{\beta}_i^{\text{dy}}$, and RGB image $\hat{\mathbf{B}}_i^{\text{dy}}$

$$\hat{\mathbf{B}}_i^{\text{full}}(\mathbf{r}) = \boldsymbol{\alpha}_i^{\text{dy}}(\mathbf{r})\hat{\mathbf{B}}_i^{\text{dy}}(\mathbf{r}) + (1 - \boldsymbol{\alpha}_i^{\text{dy}}(\mathbf{r}))\hat{\mathbf{B}}^{\text{st}}(\mathbf{r})$$

Proposed Method (iii)

- **Combining static and dynamic models**

Small temporal window is not suitable for large camera variations

→ Model the entire into two scenes, time-varying and time-invariant model

$$\mathcal{L}_{\text{pho}} = \sum_{\mathbf{r}} \sum_{j \in \mathcal{N}(i)} \hat{\mathbf{W}}_{j \to i}(\mathbf{r}) \rho(\mathbf{C}_i(\mathbf{r}), \hat{\mathbf{C}}^{\text{full}}_{j \to i}(\mathbf{r}))$$

Separate static and dynamic components

→ Train a lightweight segmentation model to separate static and dynamic parts

$\hat{\mathbf{B}}_i^{\text{dy}}, \boldsymbol{\alpha}_i^{\text{dy}}, \boldsymbol{\beta}_i^{\text{dy}} = D(I_i)$ opacity map $\boldsymbol{\alpha}_i^{\text{dy}}$, confidence map $\boldsymbol{\beta}_i^{\text{dy}}$, and RGB image $\hat{\mathbf{B}}_i^{\text{dy}}$

$$\hat{\mathbf{B}}_i^{\text{full}}(\mathbf{r}) = \boldsymbol{\alpha}_i^{\text{dy}}(\mathbf{r})\hat{\mathbf{B}}_i^{\text{dy}}(\mathbf{r}) + (1 - \boldsymbol{\alpha}_i^{\text{dy}}(\mathbf{r}))\hat{\mathbf{B}}^{\text{st}}(\mathbf{r})$$

Proposed Method (iii)

- **Combining static and dynamic models**

$\hat{\mathbf{B}}_i^{\text{full}}$ $\boldsymbol{\alpha}_i^{\text{dy}} \odot \hat{\mathbf{B}}_i^{\text{dy}}$

Small temporal window is not suitable for large camera variations

→ Model the entire into two scenes, time-varying and time-invariant model

$$\mathcal{L}_{\text{pho}} = \sum_{\mathbf{r}} \sum_{j \in \mathcal{N}(i)} \hat{\mathbf{W}}_{j \to i}(\mathbf{r}) \rho(\mathbf{C}_i(\mathbf{r}), \hat{\mathbf{C}}^{\text{full}}_{j \to i}(\mathbf{r}))$$

Separate static and dynamic components

→ Train a lightweight segmentation model to separate static and dynamic parts

$\hat{\mathbf{B}}_i^{\text{dy}}, \boldsymbol{\alpha}_i^{\text{dy}}, \boldsymbol{\beta}_i^{\text{dy}} = D(I_i)$ opacity map $\boldsymbol{\alpha}_i^{\text{dy}}$, confidence map $\boldsymbol{\beta}_i^{\text{dy}}$, and RGB image $\hat{\mathbf{B}}_i^{\text{dy}}$

$$\hat{\mathbf{B}}_i^{\text{full}}(\mathbf{r}) = \boldsymbol{\alpha}_i^{\text{dy}}(\mathbf{r}) \hat{\mathbf{B}}_i^{\text{dy}}(\mathbf{r}) + (1 - \boldsymbol{\alpha}_i^{\text{dy}}(\mathbf{r})) \hat{\mathbf{B}}^{\text{st}}(\mathbf{r})$$

$$\mathcal{L}_{\text{seg}} = \sum_{\mathbf{r}} \log \left(\boldsymbol{\beta}_i^{\text{dy}}(\mathbf{r}) + \frac{\|\hat{\mathbf{B}}_i^{\text{full}}(\mathbf{r}) - \mathbf{C}_i(\mathbf{r})\|^2}{\boldsymbol{\beta}_i^{\text{dy}}(\mathbf{r})} \right)$$

Cauchy distribution to take heteroscedastic aleatoric uncertainty of pixels

$$f(x; x_0, \gamma) = \frac{1}{\pi\gamma \left[1 + \left(\frac{x - x_0}{\gamma}\right)^2\right]} = \frac{1}{\pi} \left[\frac{\gamma}{(x - x_0)^2 + \gamma^2}\right]$$

Proposed Method (iii)

- Combining static and dynamic models

$\hat{\mathbf{B}}_i^{\text{full}}$ $\quad$ $\boldsymbol{\alpha}_i^{\text{dy}} \odot \hat{\mathbf{B}}_i^{\text{dy}}$

Small temporal window is not suitable for large camera variations

→ Model the entire into two scenes, time-varying and time-invariant model

$$\mathcal{L}_{\text{pho}} = \sum_{\mathbf{r}} \sum_{j \in \mathcal{N}(i)} \hat{\mathbf{W}}_{j \to i}(\mathbf{r}) \rho(\mathbf{C}_i(\mathbf{r}), \hat{\mathbf{C}}^{\text{full}}_{j \to i}(\mathbf{r}))$$

Separate static and dynamic components

→ Train a lightweight segmentation model to separate static and dynamic parts

$\hat{\mathbf{B}}_i^{\text{dy}}, \boldsymbol{\alpha}_i^{\text{dy}}, \boldsymbol{\beta}_i^{\text{dy}} = D(I_i)$ opacity map $\boldsymbol{\alpha}_i^{\text{dy}}$, confidence map $\boldsymbol{\beta}_i^{\text{dy}}$, and RGB image $\hat{\mathbf{B}}_i^{\text{dy}}$

$$\hat{\mathbf{B}}_i^{\text{full}}(\mathbf{r}) = \boldsymbol{\alpha}_i^{\text{dy}}(\mathbf{r}) \hat{\mathbf{B}}_i^{\text{dy}}(\mathbf{r}) + (1 - \boldsymbol{\alpha}_i^{\text{dy}}(\mathbf{r})) \hat{\mathbf{B}}^{\text{st}}(\mathbf{r})$$

$$\mathcal{L}_{\text{seg}} = \sum_{\mathbf{r}} \log \left(\boldsymbol{\beta}_i^{\text{dy}}(\mathbf{r}) + \frac{||\hat{\mathbf{B}}_i^{\text{full}}(\mathbf{r}) - \mathbf{C}_i(\mathbf{r})||^2}{\boldsymbol{\beta}_i^{\text{dy}}(\mathbf{r})} \right)$$

Cauchy distribution to take heteroscedastic aleatoric uncertainty of pixels

$$\mathcal{L}_{\text{mask}} = \sum_{\mathbf{r}} (1 - M_i)(\mathbf{r}) \rho(\hat{\mathbf{C}}^{\text{st}}(\mathbf{r}), \mathbf{C}_i(\mathbf{r})) + \sum_{\mathbf{r}} M_i(\mathbf{r}) \rho(\hat{\mathbf{C}}_i^{\text{dy}}(\mathbf{r}), \mathbf{C}_i(\mathbf{r}))$$

$$f(x; x_0, \gamma) = \frac{1}{\pi\gamma \left[1 + \left(\frac{x - x_0}{\gamma}\right)^2\right]} = \frac{1}{\pi} \left[\frac{\gamma}{(x - x_0)^2 + \gamma^2}\right]$$

Proposed Method (iv)

- **Regularization**

Monocular reconstruction of complex dynamic scenes is highly ill-posed

→ Additional regularization terms

$$\mathcal{L} = \mathcal{L}_{\text{pho}} + \mathcal{L}_{\text{mask}} + \mathcal{L}_{\text{reg}}$$
$$\mathcal{L}_{\text{reg}} = \mathcal{L}_{\text{data}} + \mathcal{L}_{\text{MT}} + \mathcal{L}_{\text{cpt}}$$

$\mathcal{L}_{\text{data}}$: Monocular depth and optical flow L1 loss with pretrained model

$\mathcal{L}_{\text{MT}}$: Cycle-consistent motion trajectory regularization

$$\mathcal{L}_{\text{cycle}} = \sum_{\mathbf{x}} \sum_{j \in \mathcal{N}(i)} w_{i \to j}(\mathbf{x}) ||\Delta_{\mathbf{x},i}(j) + \Delta_{\mathbf{x}_{i \to j},j}(i)||_1$$
$$w_{i \to j}(\mathbf{x}) = 1 - |T_i(\mathbf{x})\alpha(\sigma_i(\mathbf{x})) - T_j(\mathbf{x})\alpha(\sigma_j(\mathbf{x}))|$$

$$\begin{aligned}\mathcal{L}_{\text{sm}} = & \sum_{j \in \mathcal{N}(i)} \sum_{t \in [t_n, t_f]} ||\Delta_{\mathbf{r}(t),i}(j) - \Delta_{\mathbf{r}(t+1),i}(j)||_1 \\ & + ||\Delta_{\mathbf{r}(t),j}(j+1) - \Delta_{\mathbf{r}(t),j+1}(j+2)||_1 \\ & + \sum_{j \in \mathcal{N}(i)} \sum_{t \in [t_n, t_f]} ||\Delta_{\mathbf{r}(t),i}(j)||_1\end{aligned}$$

$\mathcal{L}_{\text{cpt}}$: Compactness prior

$$\mathcal{L}_{\text{etp}} = \sum_{\mathbf{r}} -R(\mathbf{r}) \log(R(\mathbf{r})) - (1 - R(\mathbf{r})) \log(1 - R(\mathbf{r}))$$
$$R(\mathbf{r}) = \frac{\hat{W}^{\text{dy}}(\mathbf{r})}{\hat{W}^{\text{dy}}(\mathbf{r}) + \hat{W}^{\text{st}}(\mathbf{r})}$$

Evaluation: Quantitative

Dataset: Nvidia Dynamic Scene Dataset, UCSD Dynamic Scenes Dataset

Methods	Full			Dynamic Only		
	SSIM↑	PSNR↑	LPIPS↓	SSIM↑	PSNR↑	LPIPS↓
Nerfies [49]	0.609	20.64	0.204	0.455	17.35	0.258
HyperNeRF [50]	0.654	20.90	0.182	0.446	17.56	0.242
DVS [19]	0.921	27.44	0.070	0.778	22.63	<u>0.144</u>
NSFF [35]	<u>0.927</u>	<u>28.90</u>	<u>0.062</u>	<u>0.783</u>	<u>23.08</u>	0.159
Ours	**0.957**	**30.86**	**0.027**	**0.824**	**24.24**	**0.062**

Table 1. **Quantitative evaluation on the Nvidia dataset [75].**

Methods	Full			Dynamic Only		
	SSIM↑	PSNR↑	LPIPS↓	SSIM↑	PSNR↑	LPIPS↓
Nerfies [49]	0.823	24.32	0.096	0.595	18.45	0.234
HyperNeRF [50]	0.859	25.10	0.095	0.618	19.26	0.212
DVS [19]	0.943	30.64	0.075	<u>0.866</u>	<u>26.57</u>	<u>0.096</u>
NSFF [35]	<u>0.952</u>	<u>31.75</u>	<u>0.034</u>	0.851	25.83	0.115
Ours	**0.983**	**36.47**	**0.014**	**0.909**	**28.01**	**0.042**

Table 2. **Quantitative evaluation on the UCSD dataset [37].**

Evaluation: Qualitative

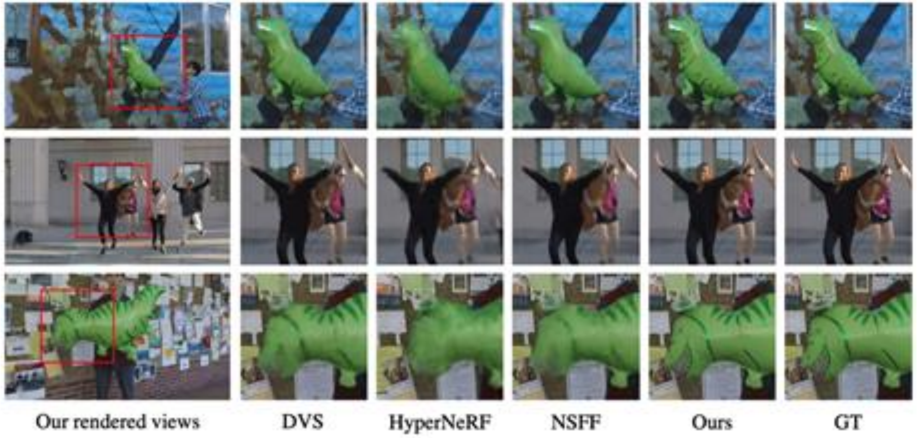


Figure 6. **Qualitative comparisons on the Nvidia dataset [75].**

Figure 7. **Qualitative comparisons on the UCSD dataset [37].**

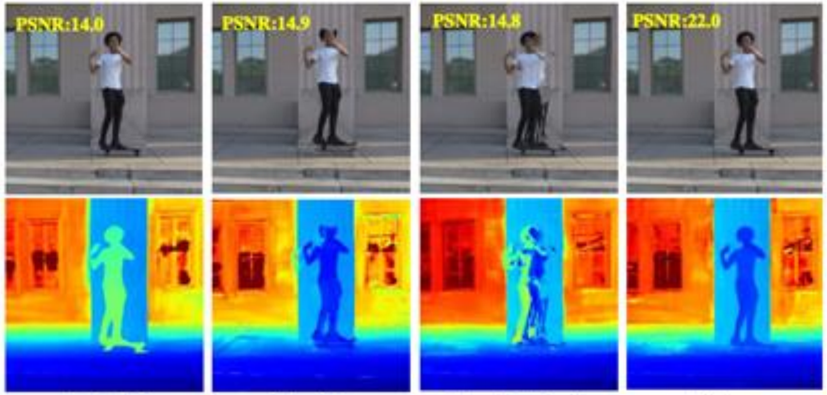
Figure 8. **Qualitative comparisons on in-the-wild videos.** We show results on 10-second videos of complex dynamic scenes. The leftmost column shows the start and end frames of each video; on the right we show novel views at intermediate times rendered from our approach and prior state-of-the-art methods [19, 35, 50].

Evaluation: Ablation study

Methods	Full			Dynamic Only		
	SSIM↑	PSNR↑	LPIPS↓	SSIM↑	PSNR↑	LPIPS↓
A) [70]+time	0.905	25.33	0.081	0.683	20.09	0.122
B) w/o TC	0.911	27.57	0.074	0.751	22.16	0.104
C) w/ SF	0.935	29.42	0.035	0.797	22.41	0.095
D) w/ M-SF	0.947	29.59	0.033	0.814	22.97	0.084
E) w/o static rep.	0.919	28.19	0.047	**0.840**	24.01	0.071
F) w/o $\mathcal{L}_{mask}$	0.930	29.95	0.036	0.835	**24.30**	**0.063**
G) w/o $\mathcal{L}_{reg}$	0.921	29.46	0.042	0.795	22.19	0.080
Full	**0.957**	**30.77**	**0.028**	<u>0.837</u>	<u>24.27</u>	<u>0.066</u>

Table 3. **Ablation study on the Nvidia Dataset.** See Sec. 5.2 for detailed descriptions of each configuration.

A) NSFF w/ extra time embedding
B) Without temporal consistency loss
C) With scene flow model
D) Multiple scene flow model
E) Without mask loss
F) Without regularization loss



(a) w/o TC (b) w/ SF (c) w/ DCT basis (d) Ours

Figure 4. **Qualitative ablations.** From left to right, we show rendered novel views (top) and depths (bottom) from our system (a) without enforcing temporal consistency, (b) aggregating image features with scene flow fields instead of motion trajectories, (c) representing motion trajectory with a fixed DCT basis instead of a learned one, and (d) with full configuration. Simpler configurations significantly degrade rendering quality as indicated by PSNR calculated over the regions of moving objects.

Discussion & Limitations

Figure 10. **Limitations.** Our method might fail to model moving thin objects such as moving leash (left). Our method can fail to render dynamic contents only visible in distant frames (middle). The rendered static content can be unrealistic or blank if insufficient source views feature are aggregated for a given pixel (right).

Conclusion

- Represented a dynamic scene within a IBRNet framework
- Motion trajectory prediction is superior to flow field
- Cross-time rendering for temporal consistency

- Not generalizable
- Fails to render where source views are sparse
- Rendering quality is dependent on choice of source views