DynaMoDe-NeRF: Motion-aware Deblurring Neural Radiance Field for Dynamic Scenes

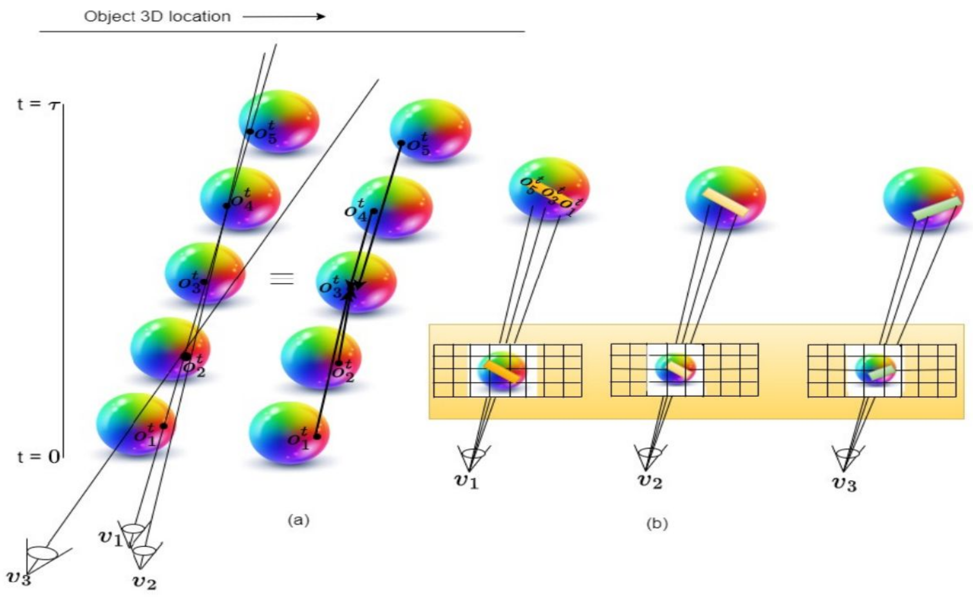
Ashish Kumar, A.N. Rajagopalan
Image Processing and Computer Vision (IPCV) Lab
Indian Institute of Technology, Madras,India

Motivation

- Neural Radiance Field (NeRF): Implicit scene representation
- Most prior works assume ideal conditions of artifact-free visual inputs
- Real scenarios: artifacts such as object motion blur for static cameras
- Some prior works address:
  - Camera motion blur
  - Defocus blur
  - Camera and defocus blur
- Object motion blur
  - Complex interaction between object motion, camera pose and the depth
  - DynaMoDe-NeRF: grounded on the physics of object motion blur formation

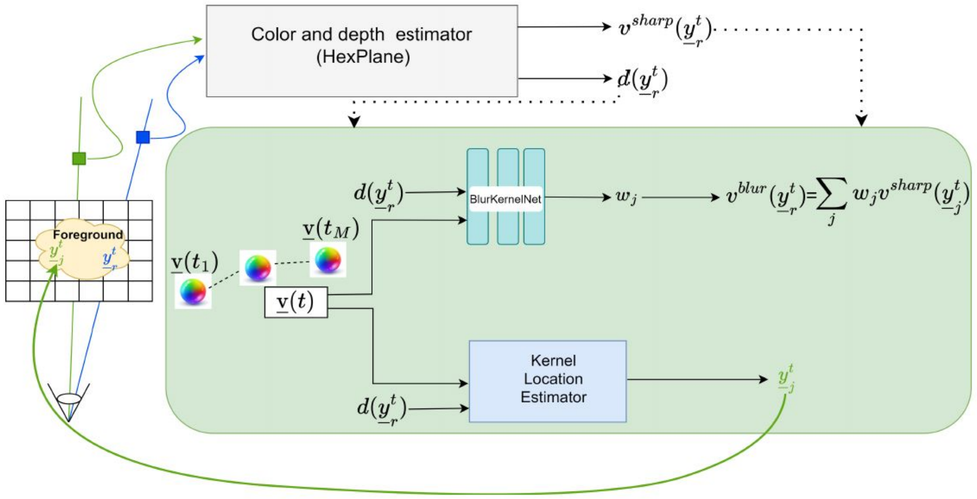
Object Motion Blur Formation

- Each view captures distinct 2D motion paths (yellow, (b)) and observes different object points.
- Despite view differences, object 3D motion is consistent in a shared reference frame, with multiple 3D point mapping to a single pixel location during exposure, governed by object velocity.
- Similar poses can produce varied blur due to depth



Contributions

- DynaMoDe-NeRF proposes to analytically correlate object 3D motion, camera pose and depth with the observed blur locations.
- Proposes BlurKernelNet that learns kernel weights in a velocity and depth-aware manner by encoding depth and motion dependencies of blur.
- Introduce a 3D motion smoothness regularizer that respects temporally smooth transition of motion.


Color and depth estimator (HexPlane)
$v^{sharp}(\underline{y}_r^t)$
$d(\underline{y}_r^t)$
Foreground
y_j^t
$\underline{y}_r^t$
$\mathbf{v}(t_1)$
$\mathbf{v}(t_M)$
$\mathbf{v}(t)$
BlurKernelNet
w_j
$v^{blur}(\underline{y}_r^t)=\sum_j w_j v^{sharp}(\underline{y}_j^t)$
Kernel Location Estimator
$\underline{y}_j^t$

Kernel Location Estimator

- Sharp: Result of one (3D point) to one (pixel) mapping (approximately)
- Blur: Result of many to one mapping.
- Explicit: if all 3D points contributing to a single pixel is known, the blur intensity is weighted average of their color.

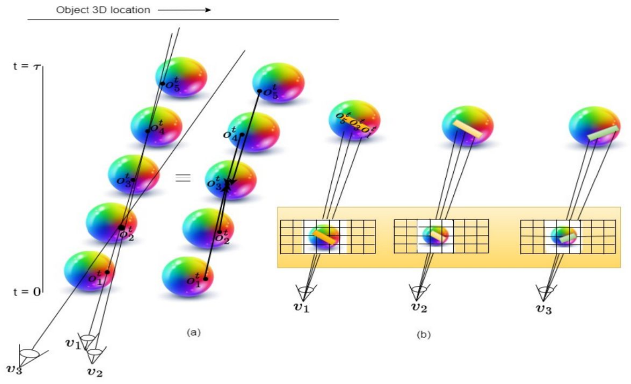
$$v^{\text{blur}}(\underline{y}_r^t) = \sum_{\underline{y}_j^t \in \mathcal{N}(\underline{y}_r^t)} w_j \cdot v^{\text{sharp}}(\underline{y}_j^t)$$

- Kernel locations

$$\underline{o}_j^t = \underline{o}_r^t + (j - (m+1)) \frac{\tau}{2m} \underline{\mathrm{v}}(t)$$

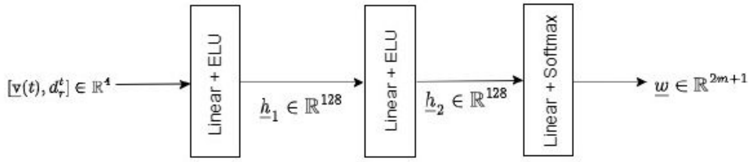
$$\underline{y}_j^t = \Pi^{-1}\left(P^v \Pi(\underline{o}_r^t) + (j - (m+1)) \frac{\tau}{2m} P^v \tilde{\Pi}(\underline{\mathrm{v}})\right)$$

$$\underline{y}_j^t = \Pi^{-1}\left(d(\underline{\mathrm{y}}_r^t) \Pi(\underline{y}_r^t) + (j - (m+1)) \frac{\tau}{2m} P_v \tilde{\Pi}(\underline{\mathrm{v}})\right)$$



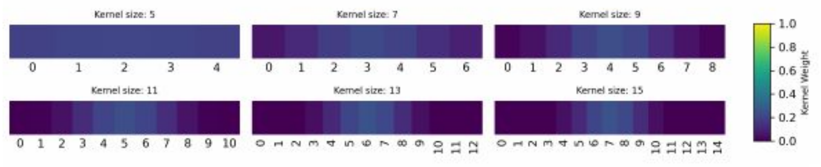
Kernel Weight Estimation (BlurKernel-Net)

- Control velocities: Linearly interpolate for intermediate time

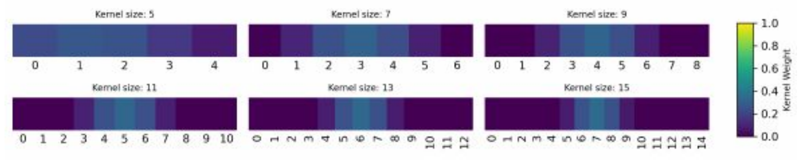


$[\underline{v}(t), d_r^t] \in \mathbb{R}^4$
Linear + ELU
$\underline{h}_1 \in \mathbb{R}^{128}$
Linear + ELU
$\underline{h}_2 \in \mathbb{R}^{128}$
Linear + Softmax
$\underline{w} \in \mathbb{R}^{2m+1}$

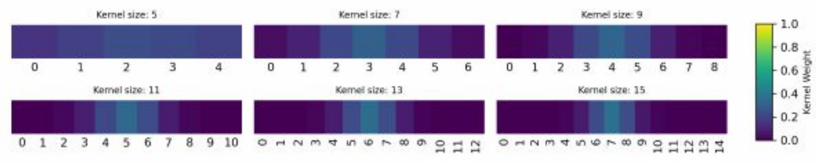
- Non-dependence of weights on kernel size



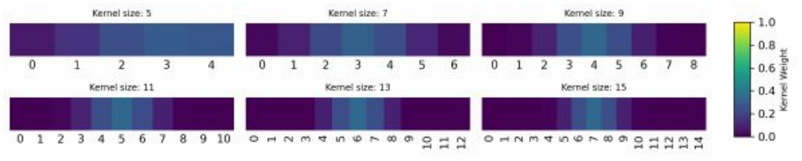
Lychee



Monkey



Cube



Sphere

Losses and Regularizers

- Photometric loss

$$\mathcal{L}_{\text{photo}} = \frac{1}{B}\sum_{r=1}^{B}\left\| v^{\text{blur}}(\underline{y}_r^t) - c(\underline{y}_r^t)\right\|^2$$

- 3D direction regularizer

$$\mathcal{L}_{\text{3dd}} = \frac{1}{L-1}\sum_{t=1}^{L-1}\left(\frac{\underline{\mathrm{v}}(t)}{||\underline{\mathrm{v}}(t)||}\odot\frac{\underline{\mathrm{v}}(t+1)}{||\underline{\mathrm{v}}(t+1)||} - 1\right)^2$$

- 3D velocity magnitude regularizer

$$\mathcal{L}_{3dm} = \frac{1}{2mB}\sum_{r=1}^{B}\sum_{j=1}^{2m+1}\text{ReLU}\left(\left|\underline{y}_r^t - \underline{y}_j^t\right| - t_h\right)$$

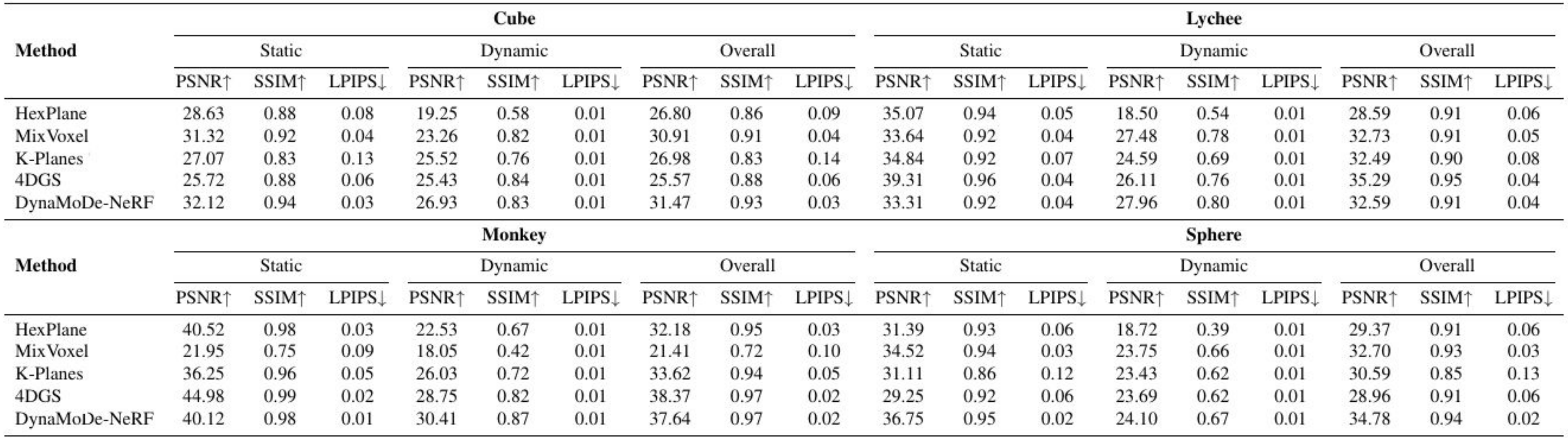
- 2D velocity regularizer

$$\mathcal{L}_{\text{2dv}} = |\underline{\mathrm{v}}_{mask} - T(\underline{y}_r^t, d(\underline{y}_r^t); P^v)\underline{\mathrm{v}}(t)|$$

- Kernel regularizer

$$\mathcal{L}_{\text{kernel}} = \|\nabla\underline{w}\|^2, \quad \text{where} \quad \nabla\underline{w} = (w_{j+1} - w_j)^2$$

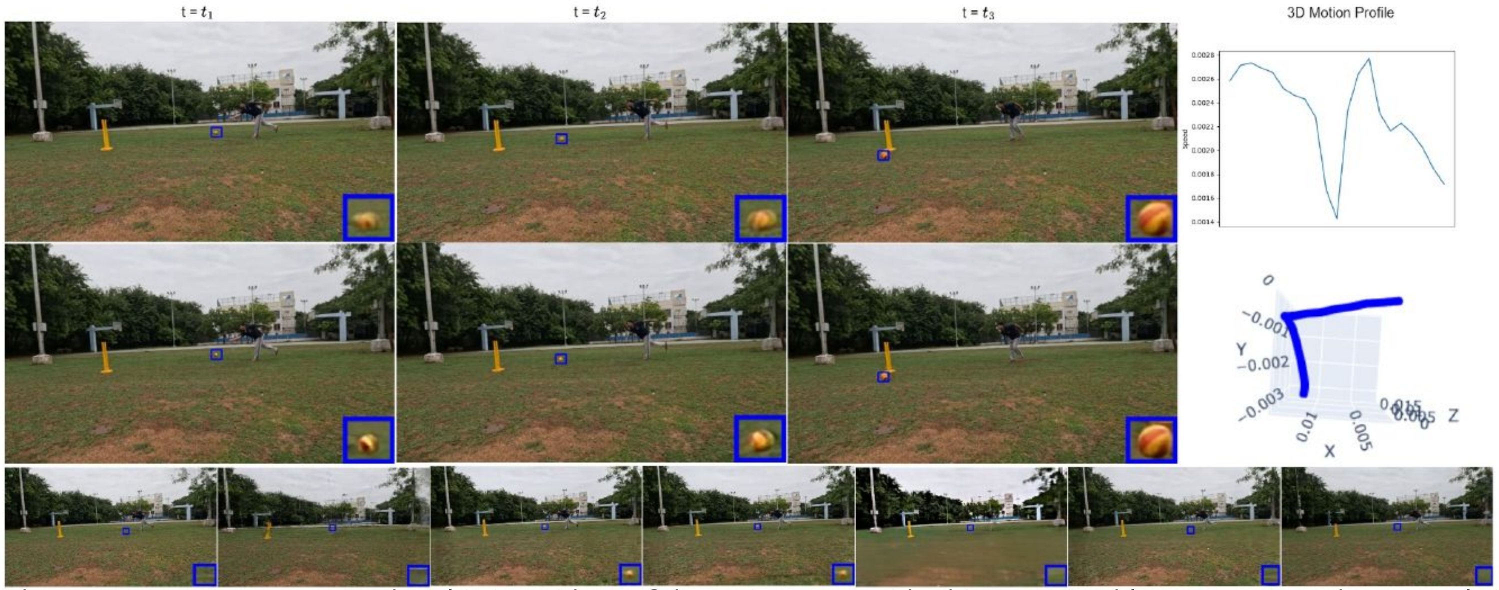
Result



Method	Cube									Lychee								
	Static			Dynamic			Overall			Static			Dynamic			Overall		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
HexPlane	28.63	0.88	0.08	19.25	0.58	0.01	26.80	0.86	0.09	35.07	0.94	0.05	18.50	0.54	0.01	28.59	0.91	0.06
MixVoxel	31.32	0.92	0.04	23.26	0.82	0.01	30.91	0.91	0.04	33.64	0.92	0.04	27.48	0.78	0.01	32.73	0.91	0.05
K-Planes	27.07	0.83	0.13	25.52	0.76	0.01	26.98	0.83	0.14	34.84	0.92	0.07	24.59	0.69	0.01	32.49	0.90	0.08
4DGS	25.72	0.88	0.06	25.43	0.84	0.01	25.57	0.88	0.06	39.31	0.96	0.04	26.11	0.76	0.01	35.29	0.95	0.04
DynaMoDe-NeRF	32.12	0.94	0.03	26.93	0.83	0.01	31.47	0.93	0.03	33.31	0.92	0.04	27.96	0.80	0.01	32.59	0.91	0.04
Method	Monkey									Sphere								
	Static			Dynamic			Overall			Static			Dynamic			Overall		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
HexPlane	40.52	0.98	0.03	22.53	0.67	0.01	32.18	0.95	0.03	31.39	0.93	0.06	18.72	0.39	0.01	29.37	0.91	0.06
MixVoxel	21.95	0.75	0.09	18.05	0.42	0.01	21.41	0.72	0.10	34.52	0.94	0.03	23.75	0.66	0.01	32.70	0.93	0.03
K-Planes	36.25	0.96	0.05	26.03	0.72	0.01	33.62	0.94	0.05	31.11	0.86	0.12	23.43	0.62	0.01	30.59	0.85	0.13
4DGS	44.98	0.99	0.02	28.75	0.82	0.01	38.37	0.97	0.02	29.25	0.92	0.06	23.69	0.62	0.01	28.96	0.91	0.06
DynaMoDe-NeRF	40.12	0.98	0.01	30.41	0.87	0.01	37.64	0.97	0.02	36.75	0.95	0.02	24.10	0.67	0.01	34.78	0.94	0.02

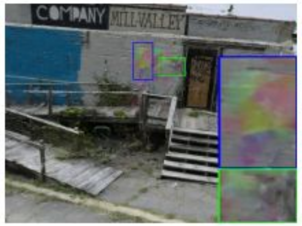
Comparisons across videos (Cube, Lychee, Monkey, and Sphere): Static (non-moving regions), Dynamic (moving foreground), Overall (entire image). The performance of our method is consistently superior in dynamic regions.

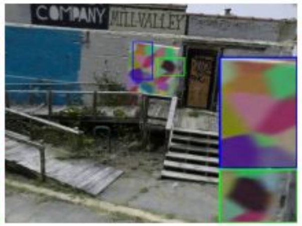
Result

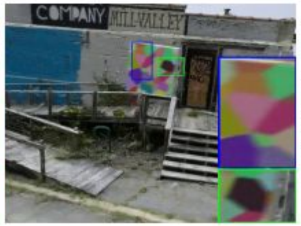


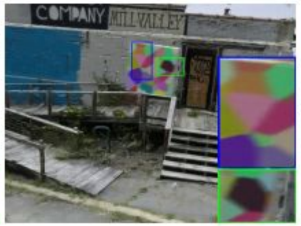
Our proposed DynaMoDe-NeRF uses posed multi-view videos of dynamic scenes with object motion blur to generate sharp novel views and recover 3D motion (velocity and trajectory). The first row shows blurry frames from a test viewpoint, while the second row shows corresponding rendered sharp novel views at three time instants. Our method effectively handles varying blur levels, from minor (column 3 at t_3) to significant (columns 1 and 2). The last column illustrates the recovered 3D motion (speed and 3D trajectory) effectively capturing the physics. Competing works, Hexplane, Mixvoxel, Deblur-NeRF, DP-NeRF, DyBluRF, GS4D, and KPlanes (from left to right in the last row), fail to synthesize novel sharp views in a robust manner (shown for time t1). Some miss the moving object completely.

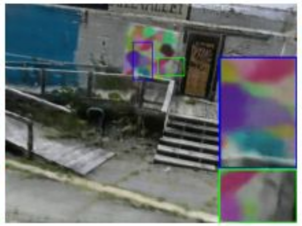
Result

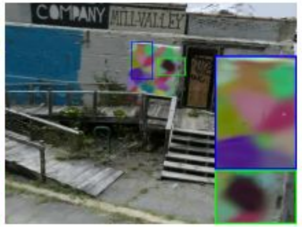

(a) HexPlane

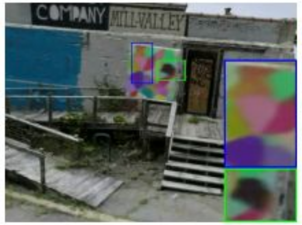

(b) MixVoxel

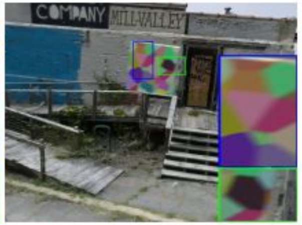

(c) Deblur-NeRF

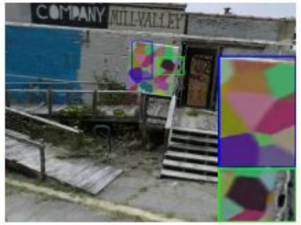

(d) DP-NeRF

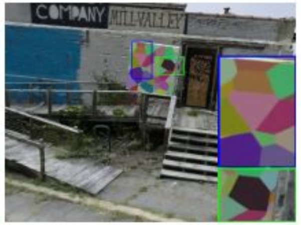

(e) DyBluRF

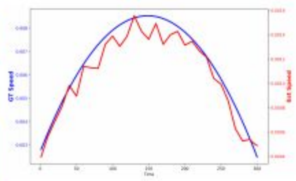

(f) GS4D

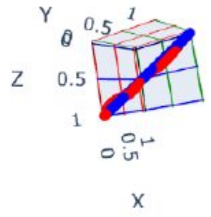

(g) KPlanes


(h) Blur GT


(i) Ours


(j) Sharp GT

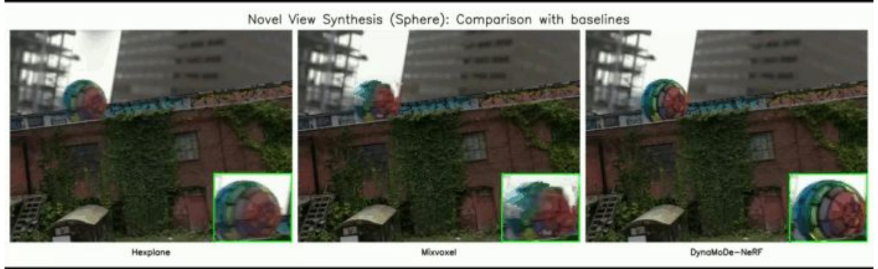

(k) Speed (CS: 0.99)

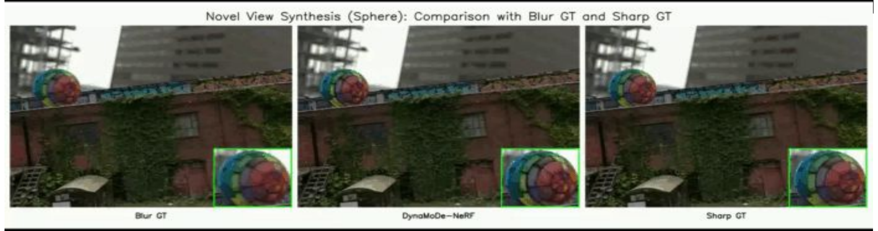


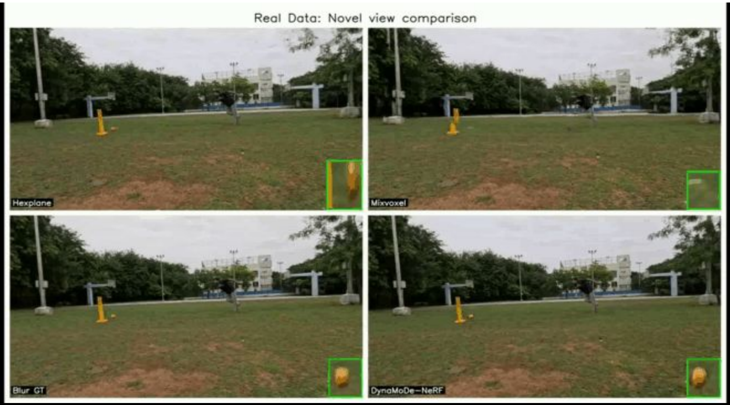
(l) Trajectory (CS:0.99)

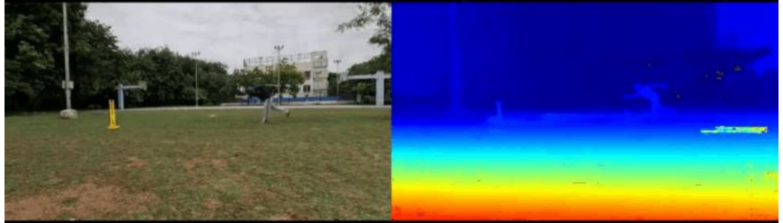
We present and compare results on an unseen view for a single time instance (Cube video). The Blur GT is an unused blurred image shown for qualitative validation of the estimated sharp novel view alongside its blurry and sharp ground truth counterparts. We compare the estimated vτ (in red) with the ground truth speed and 3D trajectory (in blue), achieving a cosine similarity (CS) of over 0.98 for complex speeds and above 0.81 for complex trajectories.

Result


Novel View Synthesis (Sphere): Comparison with baselines
Hexplane
Mixvoxel
DynaMoDe–NeRF


Novel View Synthesis (Sphere): Comparison with Blur GT and Sharp GT
Blur GT
DynaMoDe–NeRF
Sharp GT


Real Data: Novel view comparison
Hexplane
Mixvoxel
Blur GT
DynaMoDe–NeRF



Conclusions

- ❏ Deal with localized object motion blur
- ❏ Grounded on the physics
- ❏ Handles temporally varying blur
- ❏ Recovers object 3D motion profile
- ❏ BlurKernel-Net: Non-dependence of weights on kernel size