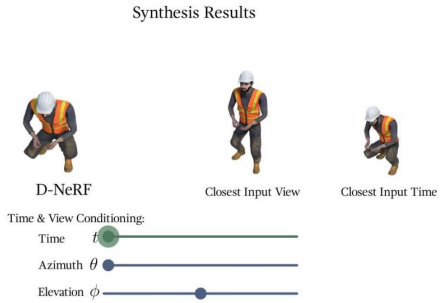
NeRFs for Deformable Objects

2022.05.23

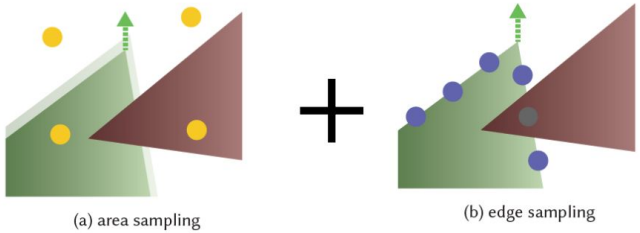
20213392 Jaehoon Yoo

Recap

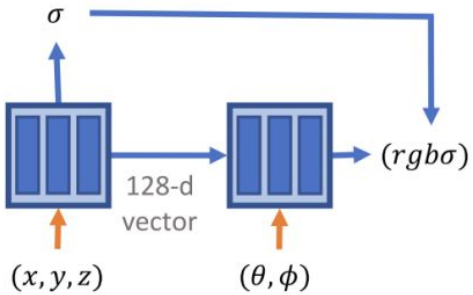
D-NeRF



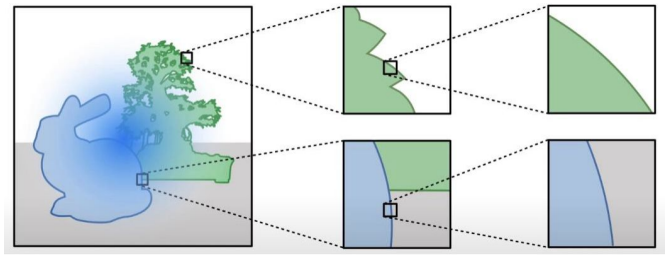
Diff. MCRT thruogh Edge Sampling



FastNeRF



Reparameterizing Discontinuous Integrands for Diff. Rendering



Nerfies: Deformable Neural Radiance Fields

Park et al. ICCV. 2021

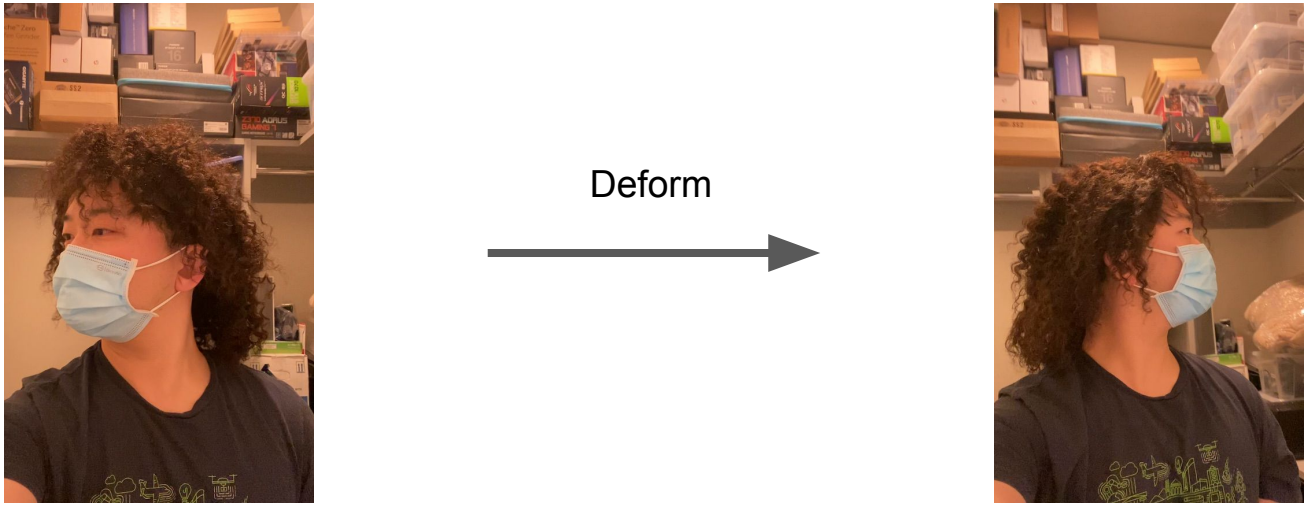
Nerfies: Goal

- Reconstructing deformable scenes using photos/videos captured casually from mobile devices.

(a) casual capture (b) input images (c) nerfie novel views (d) novel view depth

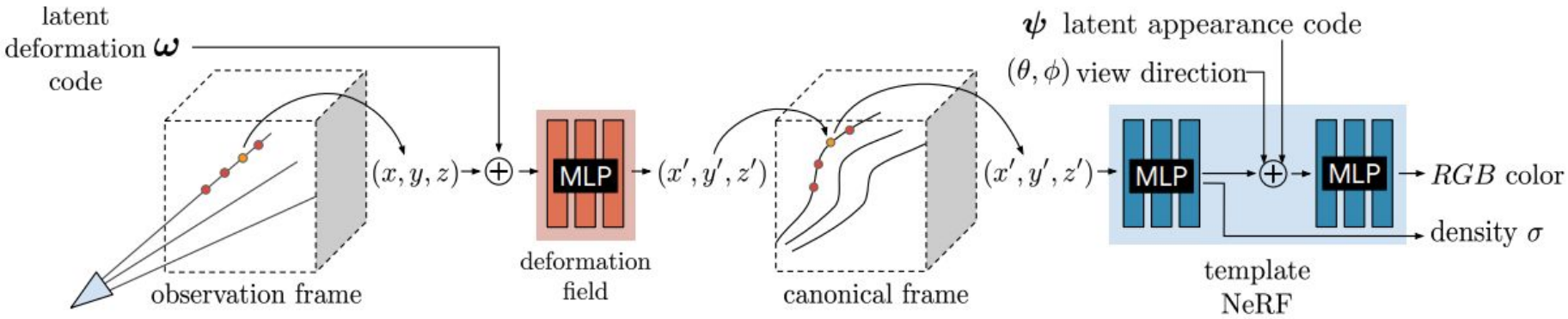
Nerfies: Challenge

- While capturing the scene with mobile devices, the scene can be deformed.



Nerfies: Solution

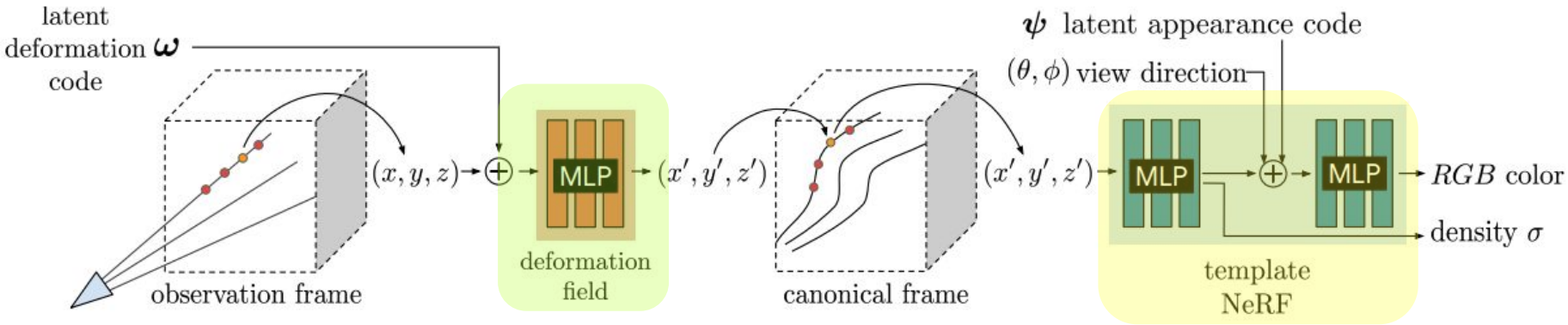
- Nerfies model the canonical frame of the scene through the deformation field network.
- Additional regularization terms are utilized to optimize the deformation field.
  - Elastic regularization
  - Background regularization
  - Coarse-to-fine annealing



Nerfies: Neural Deformation Fields

- Nerfies G extends a neural deformation field T in front of the NeRF F.
- For each frame i, the appearance code ψ_i and the deformation code ω_i is learned.

$$G(\mathbf{x}, \mathbf{d}, \boldsymbol{\psi}_i, \boldsymbol{\omega}_i) = F\left(T(\mathbf{x}, \boldsymbol{\omega}_i), \mathbf{d}, \boldsymbol{\psi}_i\right)$$



Nerfies: Neural Deformation Fields

- Nerfies utilized SE(3) (rotation + screw motion) as a deformation function T.
- MLP W inferences the SE(3) parameters (r; v) from the position and code.

$$W : (\mathbf{x}, \boldsymbol{\omega}_i) \rightarrow (\mathbf{r}, \mathbf{v})$$

$$T(\mathbf{x}, \omega_i) = e^{\mathbf{r}}\mathbf{x} + \mathbf{G}\mathbf{v}$$

Rotation Matrix

Screw Matrix

Nerfies: Elastic Regularization

- The deformation field can be ambiguous in some cases.
- To alleviate this issue, we regularize the singular value of the Jacobian of the deformation field T.
- This regularization makes the deformation to be elastic.

$$\mathbf{J}_T = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$$

$$L_{\text{elastic}}(\mathbf{x}) = \|\log \mathbf{\Sigma} - \log \mathbf{I}\|_F^2 = \|\log \mathbf{\Sigma}\|_F^2$$

Nerfies: Background Regularization

- This regularization term constrains the background to be static.
- x_ks are a static 3D points on the background.

$$L_{\text{bg}} = \frac{1}{K} \sum_{k=1}^{K} \left\| T(\mathbf{x}_k) - \mathbf{x}_k \right\|_2$$

Nerfies: Coarse-to-Fine Deformation Regularization

- Regularize the positional encoding to let the model learn coarse shapes first.

$$w_j(\alpha) = \frac{(1 - \cos(\pi \operatorname{clamp}(\alpha - j, 0, 1))}{2}$$

$$\gamma_\alpha(\mathbf{x}) = (\mathbf{x}, \cdots, w_k(\alpha)\sin(2^k\pi\mathbf{x}), w_k(\alpha)\cos(2^k\pi\mathbf{x}), \cdots)$$

Nerfies: Results

https://nerfies.github.io/

Nerfies: Results

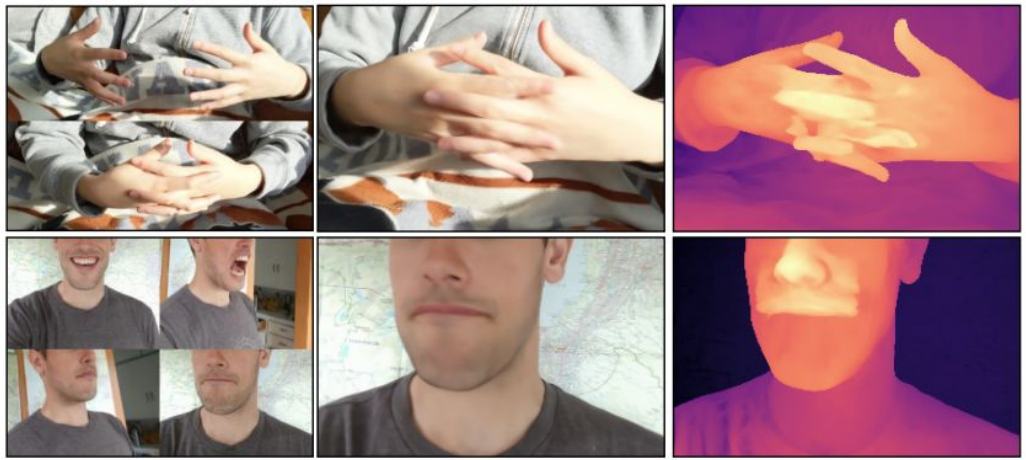
- Elastic regularization alleviates the distortion of surfaces.

Nerfies: Results

	Quasi-Static														Dynamic									
	Glasses (78 images)		Beanie (74 images)		Curls (57 images)		Kitchen (40 images)		Lamp (55 images)		Toby Sit (308 images)		Mean		Drinking (193 images)		Tail (238 images)		Badminton (356 images)		Broom (197 images)		Mean	
	PSNR↑	LPIPS↓	PSNR↑	LPIPS↓	PSNR↑	LPIPS↓	PSNR↑	LPIPS↓	PSNR↑	LPIPS↓	PSNR↑	LPIPS↓	PSNR↑	LPIPS↓	PSNR↑	LPIPS↓	PSNR↑	LPIPS↓	PSNR↑	LPIPS↓	PSNR↑	LPIPS↓	PSNR↑	LPIPS↓
NeRF [39]	18.1	.474	16.8	.583	14.4	.616	19.1	.434	17.4	.444	22.8	.463	18.1	.502	18.6	.397	23.0	.571	18.8	.392	21.0	.667	20.3	.506
NeRF + latent	19.5	.463	19.5	.535	17.3	.539	20.1	.403	18.9	.386	19.4	.385	19.1	.452	21.9	.233	24.9	.404	20.0	.308	21.9	.576	22.2	.380
Neural Volumes [31]	15.4	.616	15.7	.595	15.2	.588	16.2	.569	13.8	.533	13.7	.473	15.0	.562	16.2	.198	18.5	.559	13.1	.516	16.1	.544	16.0	.454
NSFF†	19.6	.407	21.5	.402	18.0	.432	21.4	.317	20.5	.239	26.9	.208	21.3	.334	27.7	.0803	30.6	.245	21.7	.205	28.2	.202	27.1	.183
$\gamma(t)$ + Trans† [29]	22.2	.354	20.8	.471	20.7	.426	22.5	.344	21.9	.283	25.3	.420	22.2	.383	23.7	.151	27.2	.391	22.9	.221	23.4	.627	24.3	.347
Ours ($\lambda = 0.01$)	23.4	.305	22.2	.391	24.6	.319	23.9	.280	23.6	.232	22.9	.159	23.4	.281	22.4	.0872	23.9	.161	22.4	.130	21.5	.245	22.5	.156
Ours ($\lambda = 0.001$)	24.2	.307	23.2	.391	24.9	.312	23.5	.279	23.7	.230	22.8	.174	23.7	.282	21.8	.0962	23.6	.175	22.1	.132	21.0	.270	22.1	.168
No elastic	23.1	.317	24.2	.382	24.1	.322	22.9	.290	23.7	.230	23.0	.257	23.5	.300	22.2	.0863	23.7	.174	22.0	.132	20.9	.287	22.2	.170
No coarse-to-fine	23.8	.312	21.9	.408	24.5	.321	24.0	.277	22.8	.242	22.7	.244	23.3	.301	22.3	.0960	24.3	.257	21.8	.151	21.9	.406	22.6	.228
No SE3	23.5	.314	21.9	.401	24.5	.317	23.7	.282	22.7	.235	22.9	.206	23.2	.293	22.4	.0867	23.5	.191	21.2	.156	20.9	.276	22.0	.177
Ours (base)	24.0	.319	20.9	.456	23.5	.345	22.4	.323	22.1	.254	22.7	.184	22.6	.314	22.6	.127	24.3	.298	21.1	.173	22.1	.503	22.5	.275
No BG Loss	22.3	.317	21.5	.395	20.1	.371	22.5	.290	20.3	.260	22.3	.145	21.5	.296	22.3	.0856	23.5	.210	20.4	.161	20.9	.330	21.8	.196

Nerfies: Limitations

- Cannot model the topological variation.
- Hard to model rapid motions.
- Optimizing SE(3) is still non-convex problem due to the ambiguity.



HyperNeRF: A Higher-Dimensional Representation for Topologically Varying Neural Radiance Fields

Park et al. ACM Trans. Graph. 2021

HyperNeRF: Problem & Solution

- Alleviate the Nerfies' lack of modeling topological variation by utilizing the hyperspace.

(a) Input Images

(b) Nerfies

$(\mathbf{c}, \sigma) = F(x, y, z)$

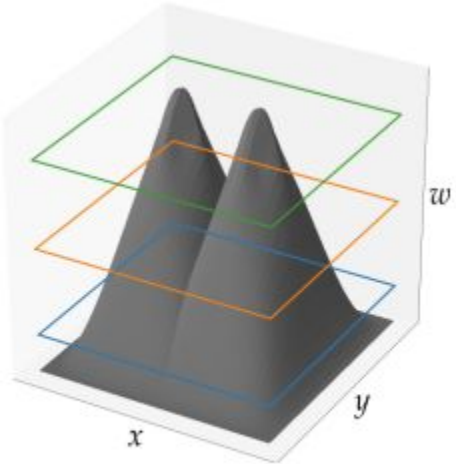
(c) HyperNeRF

$(\mathbf{c}, \sigma) = F(x, y, z, w_1, w_2)$

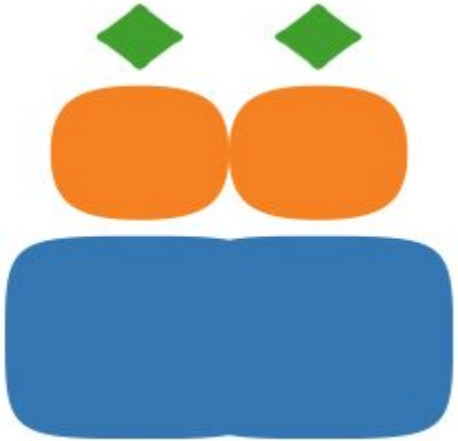
(d) Hyper-Space

HyperNeRF: Level Set Method

- Topologically varying shapes on N dimension space can be seen as slicing the M dimensional shape with a plane. (M > N).



(a) 3D Auxiliary Function

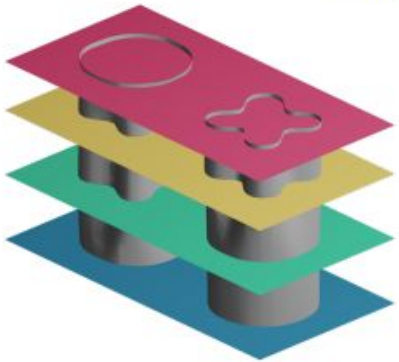


(b) 2D Sliced Shapes

HyperNeRF: Level Set Method

- Similar to the level set method, HyperNeRF models the 3+W dimensional shape and slice the hyper-shape with W dimensional cut.
- We call the W dimensions as ambient dimensions.

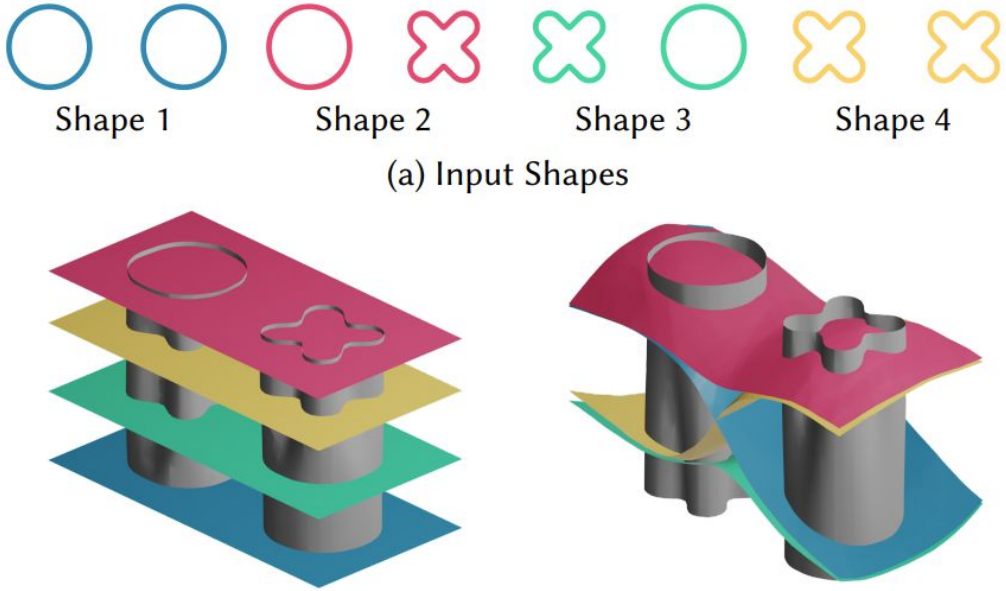
Shapes from Slices



Shape on Hyperspace

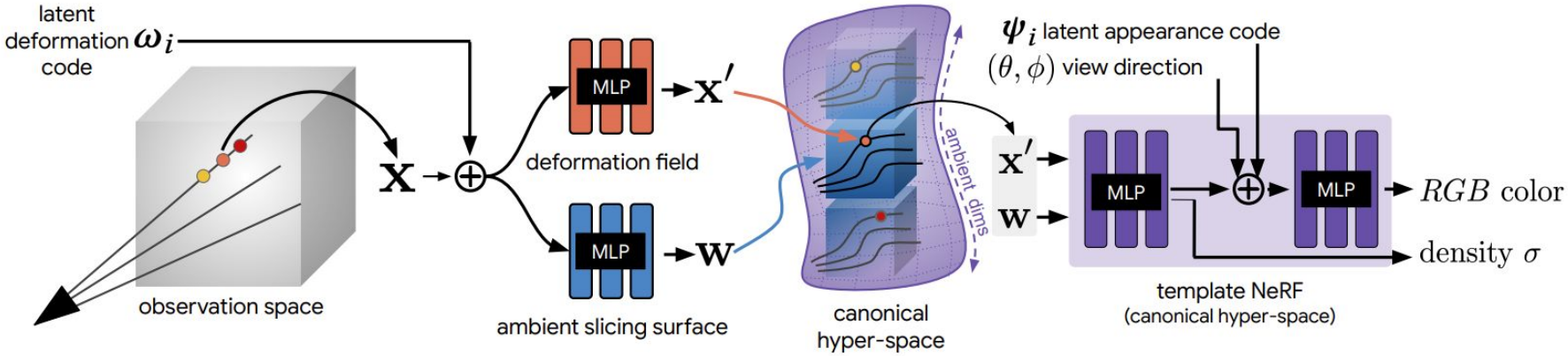
HyperNeRF: Deformable Slicing

- By deforming the slicing surface, the shape on the hyper space can be even simpler.



HyperNeRF:

- Deformation field models topology preserved motions, while ambient slicing surface models topology varying motions.



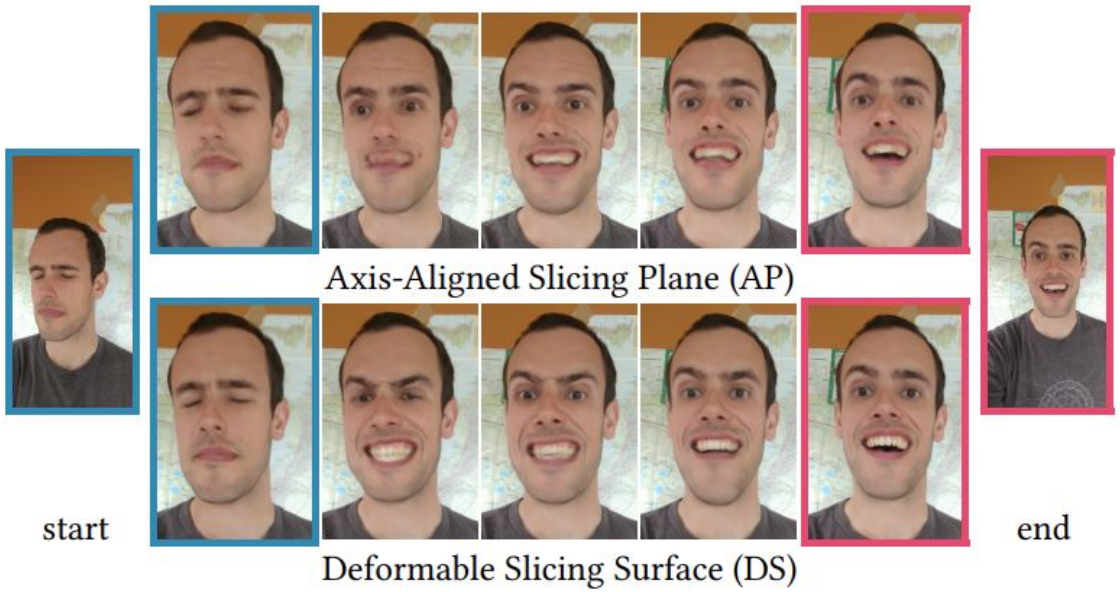
HyperNeRF: Implementation

- HyperNeRF is implemented by adding the ambient slicing surface H to the Nerfies.
- Axis-aligned Slicing Plane can be implemented by adopting a function H which is not depends on the position x.

$$\mathbf{x}' = T(\mathbf{x}, \boldsymbol{\omega}_i),$$
$$\mathbf{w} = H(\mathbf{x}, \boldsymbol{\omega}_i),$$
$$(\mathbf{c}, \sigma) = F(\mathbf{x}', \mathbf{w}, \mathbf{d}, \boldsymbol{\psi}_i)$$

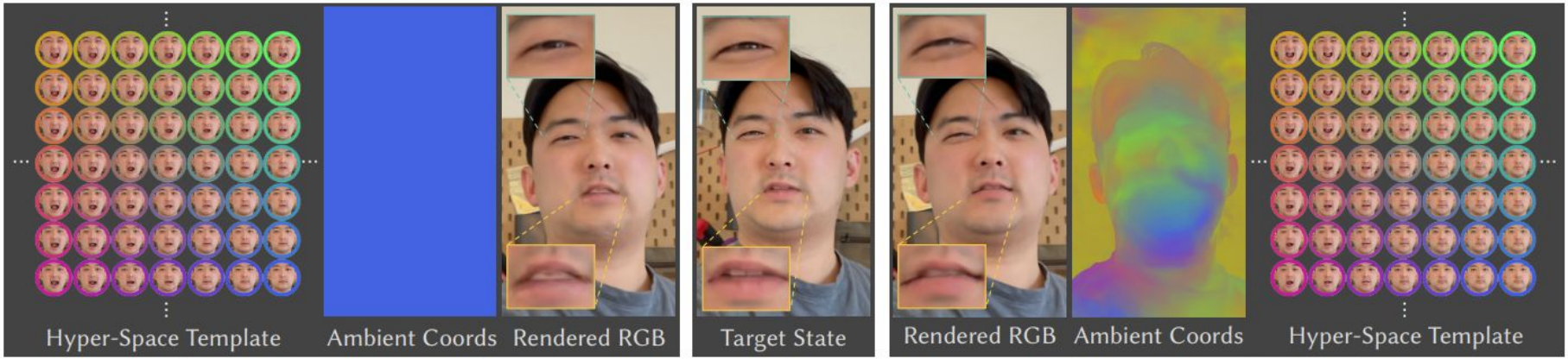
HyperNeRF: Results

- Deformable Slicing is more expressive than Axis-Aligned Slicing.



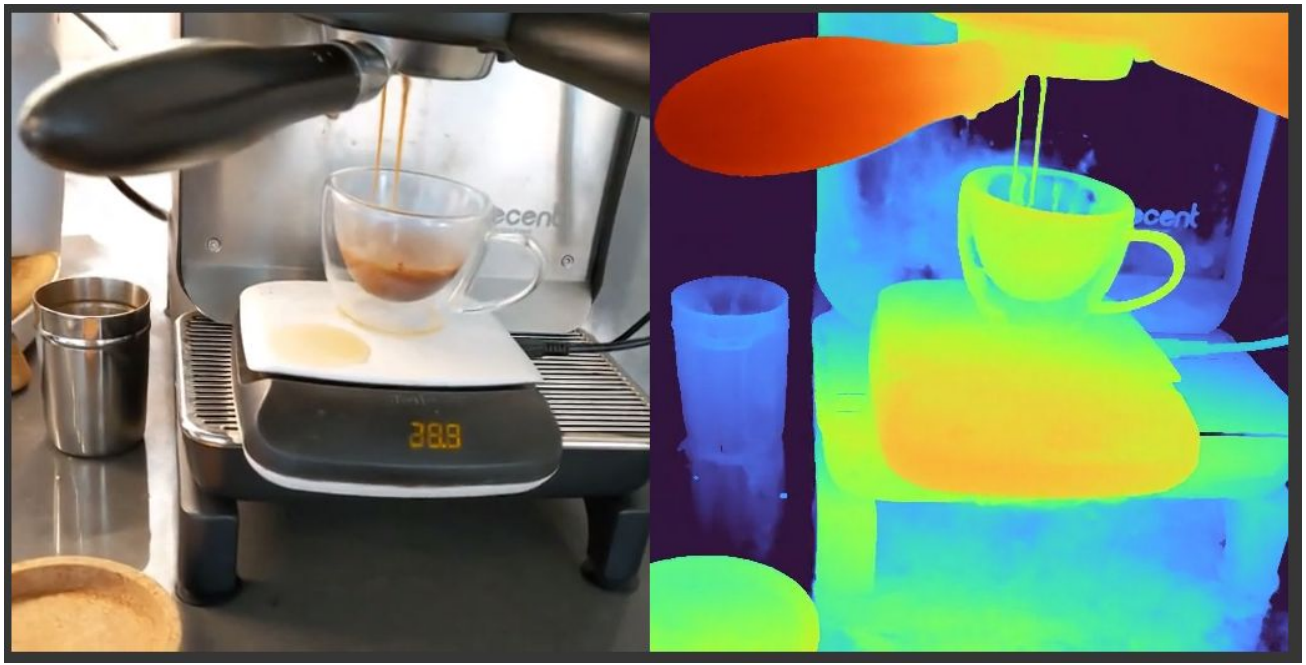
HyperNeRF: Results

- Deformable Slicing is more expressive than Axis-Aligned Slicing.



(a) Axis-Aligned Slicing Plane (AP) (b) (c) Deformable Slicing Surface (DS)

HyperNeRF: Results



https://hypernerf.github.io/

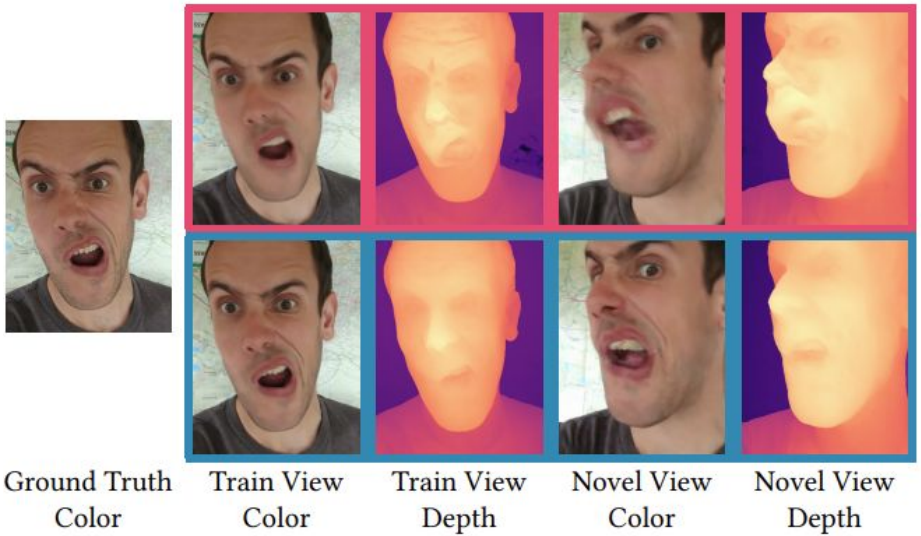
HyperNeRF: Results

- HyperNeRF outperforms Nerfies on dynamic sequences.

	Broom (197 images)			3D printer (207 images)			Chicken (164 images)			Expressions (259 images)			Peel Banana (513 images)			Mean		
	PSNR↑	MS-SSIM↑	LPIPS↓	PSNR↑	MS-SSIM↑	LPIPS↓	PSNR↑	MS-SSIM↑	LPIPS↓	PSNR↑	MS-SSIM↑	LPIPS↓	PSNR↑	MS-SSIM↑	LPIPS↓	PSNR↑	MS-SSIM↑	LPIPS↓
NeRF [Mildenhall et al. 2020]	19.9	.653	.692	20.7	.780	.357	19.9	.777	.325	20.1	.697	.394	20.0	.769	.352	20.1	.735	.424
NV [Lombardi et al. 2019]	17.7	.623	.360	16.2	.665	.330	17.6	.615	.336	14.6	.672	.276	15.9	.380	.413	16.4	.591	.343
NSFF [Li et al. 2020]†	26.1	.871	.284	27.7	.947	.125	26.9	.944	.106	26.7	.922	.157	24.6	.902	.198	26.4	.917	.174
Nerfies [Park et al. 2020]	19.2	.567	.325	20.6	.830	.108	26.7	.943	.0777	21.8	.802	.150	22.4	.872	.147	22.1	.803	.162
Nerfies (w/o elastic)	19.4	.581	.323	20.2	.820	.115	26.0	.935	.0837	21.8	.800	.149	21.7	.852	.157	21.8	.798	.165
Hyper-NeRF (DS)	19.3	.591	.296	20.0	.821	.111	26.9	.948	.0787	21.6	.800	.148	23.3	.896	.133	22.2	.811	.153
Hyper-NeRF (DS, w/ elastic)	19.5	.605	.277	20.2	.823	.109	27.5	.954	.0756	21.9	.806	.144	22.7	.882	.133	22.3	.814	.148
Hyper-NeRF (AP)	19.6	.596	.319	20.0	.814	.131	27.2	.950	.0941	22.2	.817	.149	22.4	.874	.142	22.2	.810	.167
Hyper-NeRF (w/o deform)	20.6	.714	.613	21.4	.846	.212	27.6	.950	.108	22.0	.793	.196	24.3	.914	.170	23.2	.843	.260

HyperNeRF: Results

- HyperNeRF outperforms Nerfies on dynamic sequences.



HyperNeRF: Limitations

- Cannot capture the rapid motions.
- Cannot utilize domain specific priors

Q&A