INVERSE
GRAPHICS

MODULE 5:

Motion and Objectness

Dynamic Neural Scene Representations & Physics

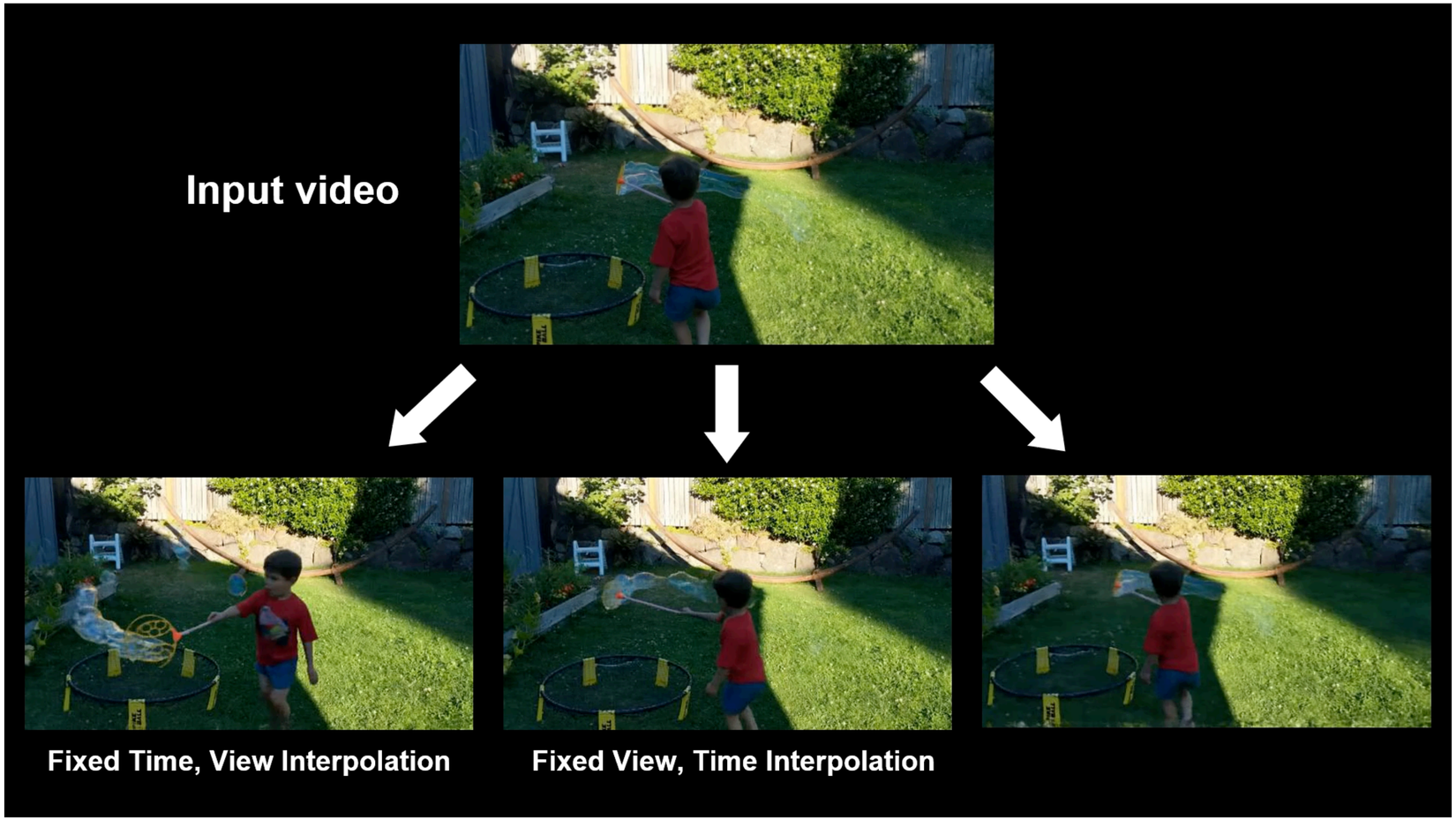


Image Courtesy: Neural Scene Flow Fields for Space-Time View Synthesis of Dynamic Scenes. *Li et. al.*

Why?	**Our world is not static, but dynamic. We can reason about dynamic processes (rigid and non-rigid), such as tracking a car over time or realizing when something is deforming. We need mathematical tools for this.**
What you'll learn.	**How to reconstruct dynamic scenes. Shortcomings and open research directions.**

Dynamic Neural Scene Representations & Physics

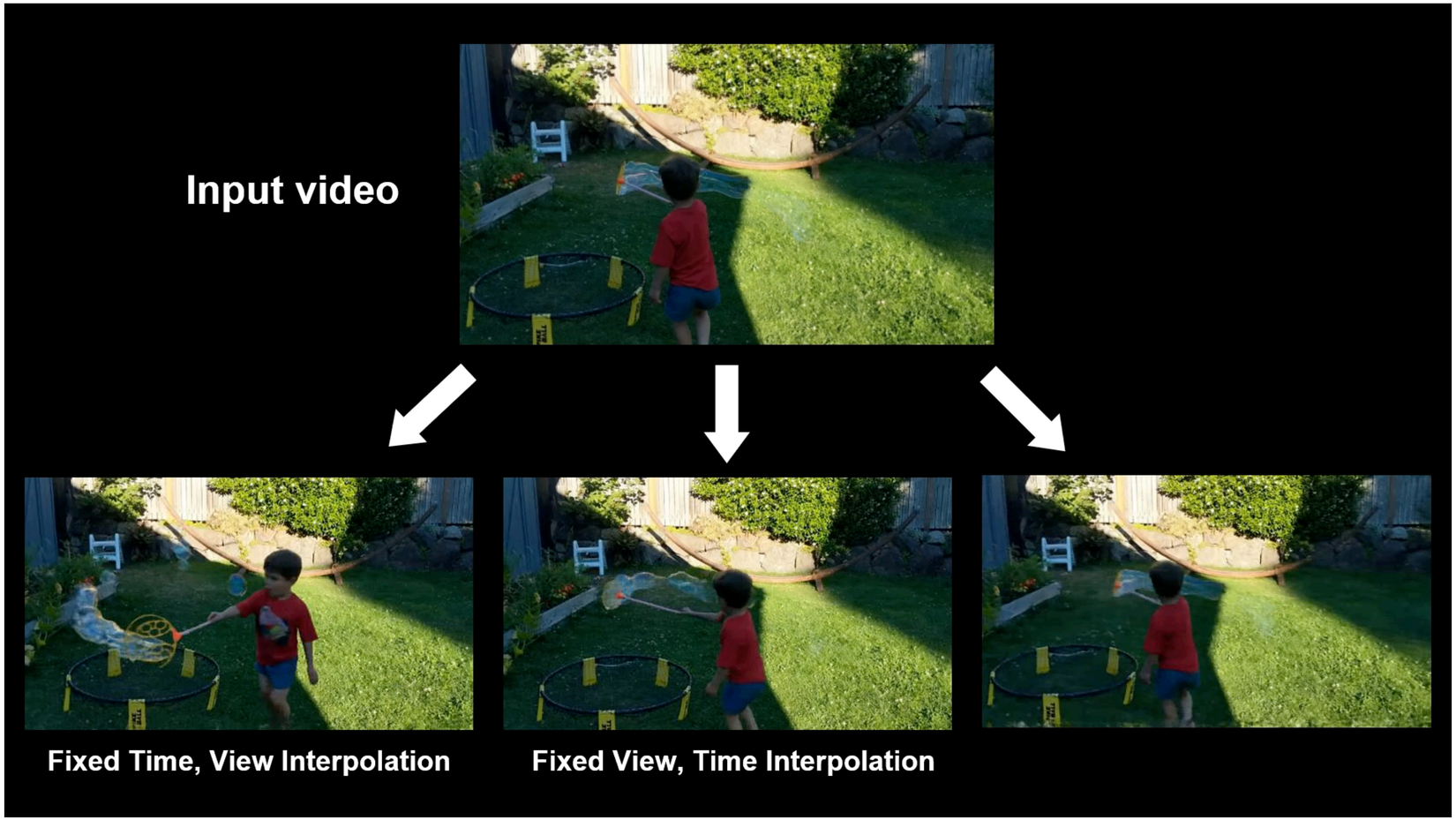
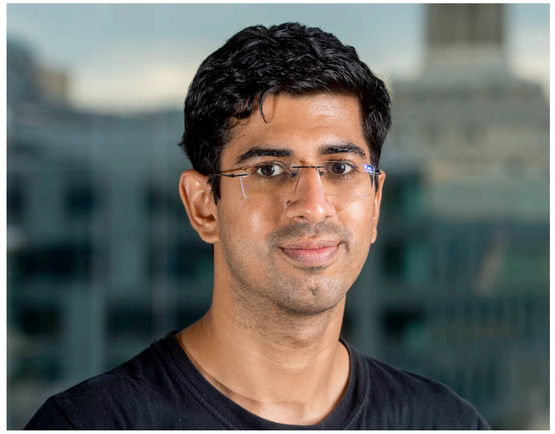


Image Courtesy: Neural Scene Flow Fields for Space-Time View Synthesis of Dynamic Scenes. *Li et. al.*

Why?	**Our world is not static, but dynamic. We can reason about dynamic processes (rigid and non-rigid), such as tracking a car over time or realizing when something is deforming. We need mathematical tools for this.**
What you'll learn.	**How to reconstruct dynamic scenes. Shortcomings and open research directions.**

Many Slides Adapted From:



CMU 16-889: Learning for 3D Vision

Shubham Tulsiani

We live in a Dynamic World

We live in a Dynamic World

Slide curtesy of Shubham Tulsiani, CMU 16-889: Learning for 3D Vision

We live in a Dynamic World

We live in a Dynamic World

We live in a Dynamic World

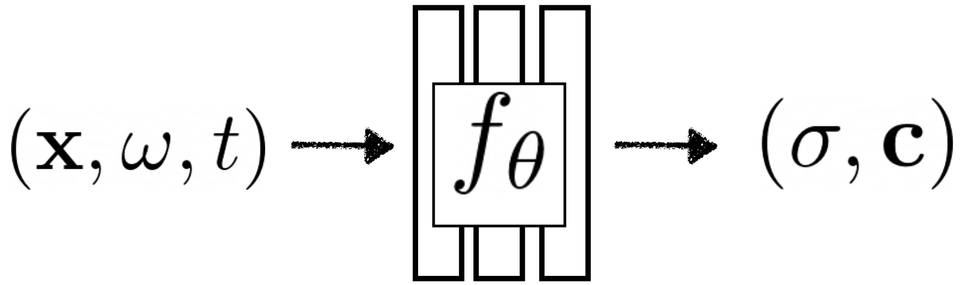
(and other more ‘useful’ situations also involve a dynamic structure)

Slide curtesy of Shubham Tulsiani, CMU 16-889: Learning for 3D Vision

Towards Modeling a Dynamic World

Idea 1: Just add time as an additional variable!

Towards Modeling a Dynamic World



Idea 1: Just add time as an additional variable!

Towards Modeling a Dynamic World

$$(\mathbf{x}, \omega, t) \rightarrow f_\theta \rightarrow (\sigma, \mathbf{c})$$

Idea 1: Just add time as an additional variable!

Slide curtesy of Shubham Tulsiani, CMU 16-889: Learning for 3D Vision

Towards Modeling a Dynamic World

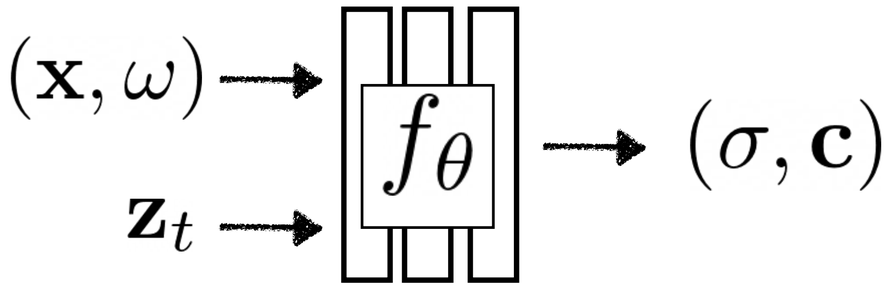
$$(\mathbf{x}, \omega, t) \rightarrow \boxed{f_\theta} \rightarrow (\sigma, \mathbf{c})$$

Idea 1: Just add time as an additional variable!

Temporally far does not imply structurally far

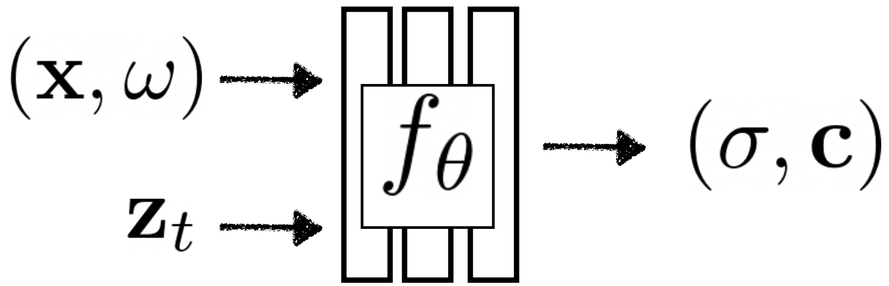
Slide curtesy of Shubham Tulsiani, CMU 16-889: Learning for 3D Vision

Towards Modeling a Dynamic World

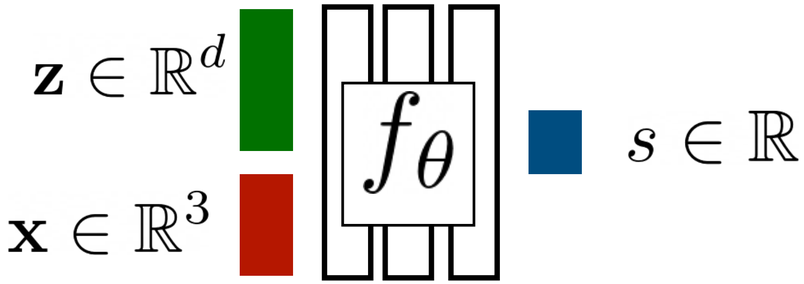


Idea 1 (alternative implementation): Time-based latent variable

Towards Modeling a Dynamic World



Idea 1 (alternative implementation): Time-based latent variable

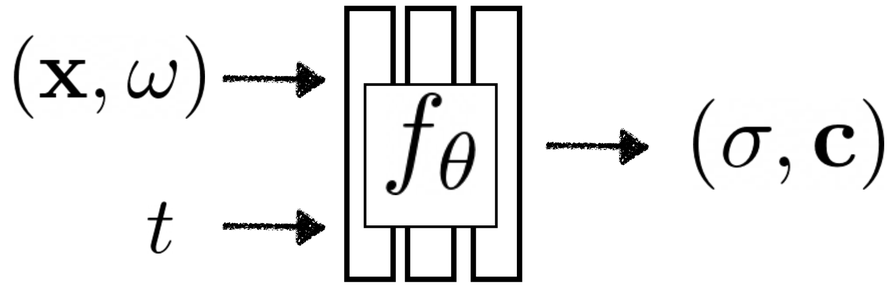


$$f_\theta : \mathbb{R}^3 \times \mathbb{R}^d \rightarrow \mathbb{R}$$

Generalizable Signed Distance Field:
(latent code, position) -> (distance)

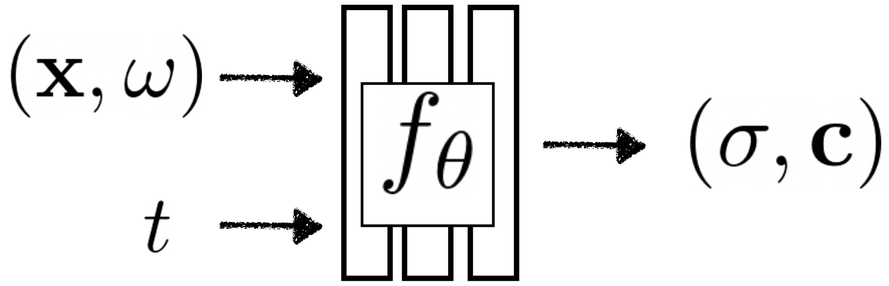
Slide curtesy of Shubham Tulsiani, CMU 16-889: Learning for 3D Vision

Towards Modeling a Dynamic World



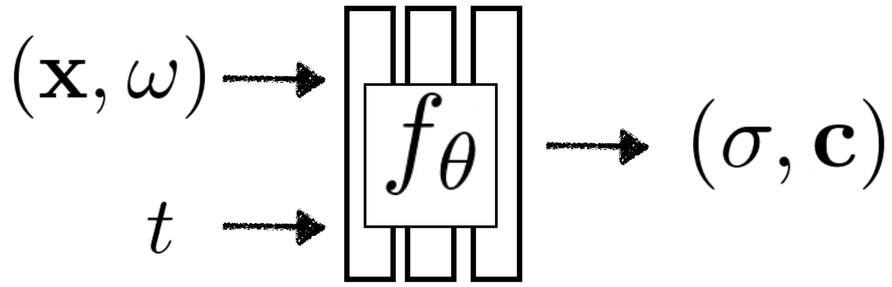
Slide curtesy of Shubham Tulsiani, CMU 16-889: Learning for 3D Vision

Towards Modeling a Dynamic World



Idea 1: Time as additional input — f(· , · , **t**) is the scene at time **t**

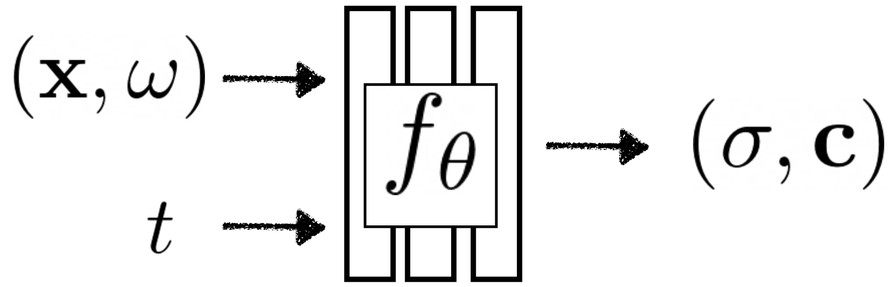
Towards Modeling a Dynamic World



Idea 1: Time as additional input — f(· , · , **t**) is the scene at time **t**

Would typically require multi-view data at each instance

Towards Modeling a Dynamic World



Idea 1: Time as additional input — f(· , · , **t**) is the scene at time **t**

Would typically require multi-view data at each instance

No explicit constraints that f(· , · , **t**) and f(· , · , **t'**) represent a common scene

Slide curtesy of Shubham Tulsiani, CMU 16-889: Learning for 3D Vision

What makes a scene persistent?

t=0

t=100

What makes a scene persistent?

t=0

t=100

The stuff at t=0 is the ‘same’ is t=100, but possibly at a different location

What makes a scene persistent?

The stuff at t=0 is the ‘same’ is t=100, but possibly at a different location

What makes a scene persistent?

Today's Goals

What makes a scene persistent?

Today's Goals

1. Formalize correspondence across time as 'scene flow', and see representative prediction methods

What makes a scene persistent?

Today's Goals

1. Formalize correspondence across time as 'scene flow', and see representative prediction methods

2. Temporal consistency via Neural scene flows

What makes a scene persistent?

Today's Goals

1. Formalize correspondence across time as 'scene flow', and see representative prediction methods
2. Temporal consistency via Neural scene flows
3. Beyond temporal smoothness: from scene flow to canonical flow

Correspondence via Flow Fields

Optical Flow

Given a pixel in frame 1, where does it go to in frame 2?

Image Courtesy: RAFT. *Teed et. al.*

Correspondence via Flow Fields

Optical Flow

Given a pixel in frame 1, where does it go to in frame 2?

$\mathbb{R}^2 \rightarrow \mathbb{R}^2$ A 2D flow field

Image Courtesy: RAFT. *Teed et. al.*

Inverse flow

Optical Flow

Given a pixel in frame **2, where does it come from frame 1**?

Image Courtesy: RAFT. *Teed et. al.*

Inverse flow

Optical Flow

Given a pixel in frame **2, where does it come from frame 1**?

$\mathbb{R}^2 \rightarrow \mathbb{R}^2$ A 2D flow field

Image Courtesy: RAFT. *Teed et. al.*

Inverse flow

Optical Flow

Given a pixel in frame **2, where does it come from frame 1**?

$\mathbb{R}^2 \rightarrow \mathbb{R}^2$ A 2D flow field

An image:

$$\mathcal{I} : \mathbb{R}^2 \rightarrow \mathbb{R}^3; \quad \mathcal{I}(\mathbf{x}) = \mathbf{c}$$

A 2D flow field (optical flow):

$$\Phi(\mathbf{x}) : \mathbb{R}^2 \rightarrow \mathbb{R}^2; \quad \Phi(\mathbf{x}) = \Delta\mathbf{x}$$

Can be used to “warp” image:

$$\hat{\mathcal{I}}(\mathbf{x}) = \mathcal{I}(\mathbf{x} + \Phi(\mathbf{x}))$$

Image Courtesy: RAFT. *Teed et. al.*

What if want to warp over multiple time steps?

What if want to warp over multiple time steps?

What if want to warp over multiple time steps?

$\mathcal{I}_0(\mathbf{x})$

What if want to warp over multiple time steps?

$\mathcal{I}_0(\mathbf{x})$  $\mathcal{I}_1(\mathbf{x}) = \mathcal{I}_0(\mathbf{x} + \Phi_{10}(\mathbf{x}))$

What if want to warp over multiple time steps?

$$\mathcal{I}_0(\mathbf{x}) \Rightarrow \mathcal{I}_1(\mathbf{x}) = \mathcal{I}_0(\mathbf{x} + \Phi_{10}(\mathbf{x})) \Rightarrow \mathcal{I}_2(\mathbf{x}) = \mathcal{I}_1(\mathbf{x} + \Phi_{21}(\mathbf{x}))$$

$$= \mathcal{I}_0(\mathbf{x} + \Phi_{21}(\mathbf{x}) +$$

What if want to warp over multiple time steps?

$$\mathcal{I}_0(\mathbf{x}) \Rightarrow \mathcal{I}_1(\mathbf{x}) = \mathcal{I}_0(\mathbf{x} + \Phi_{10}(\mathbf{x})) \Rightarrow \mathcal{I}_2(\mathbf{x}) = \mathcal{I}_1(\mathbf{x} + \Phi_{21}(\mathbf{x}))$$
$$= \mathcal{I}_0(\mathbf{x} + \Phi_{21}(\mathbf{x}) + \Phi_{10}(\mathbf{x} + \Phi_{21}(\mathbf{x})))$$

Computing Optical Flow: Correspondence Volumes

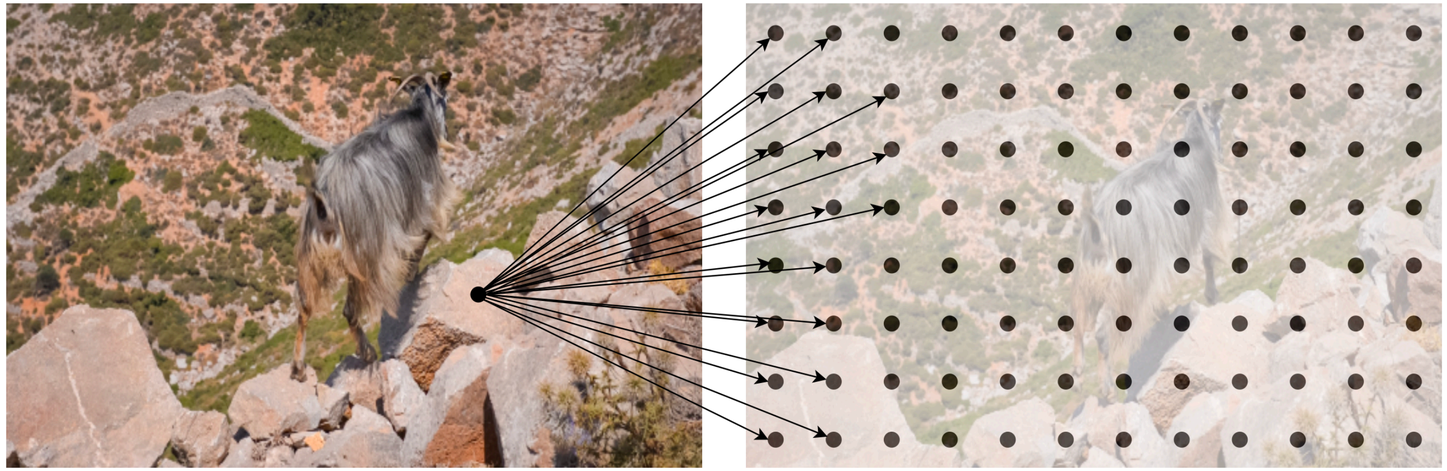
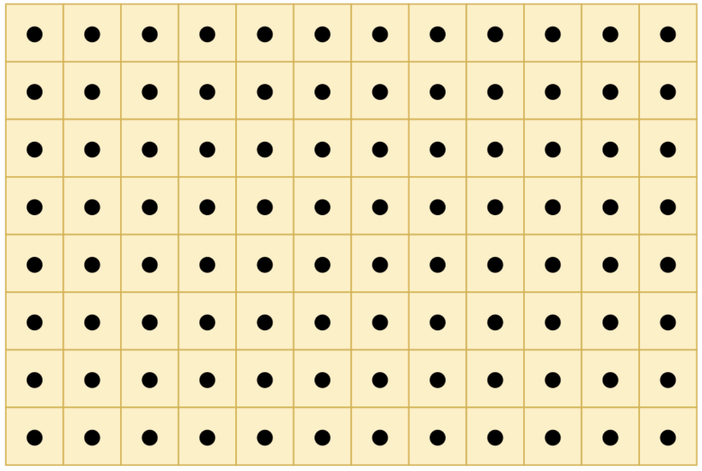


Image 1

Image 2



$\mathbf{C}^1 \in H \times W \times H \times W$

Image Courtesy: RAFT. *Teed et. al.*

Computing Optical Flow: Correspondence Volumes

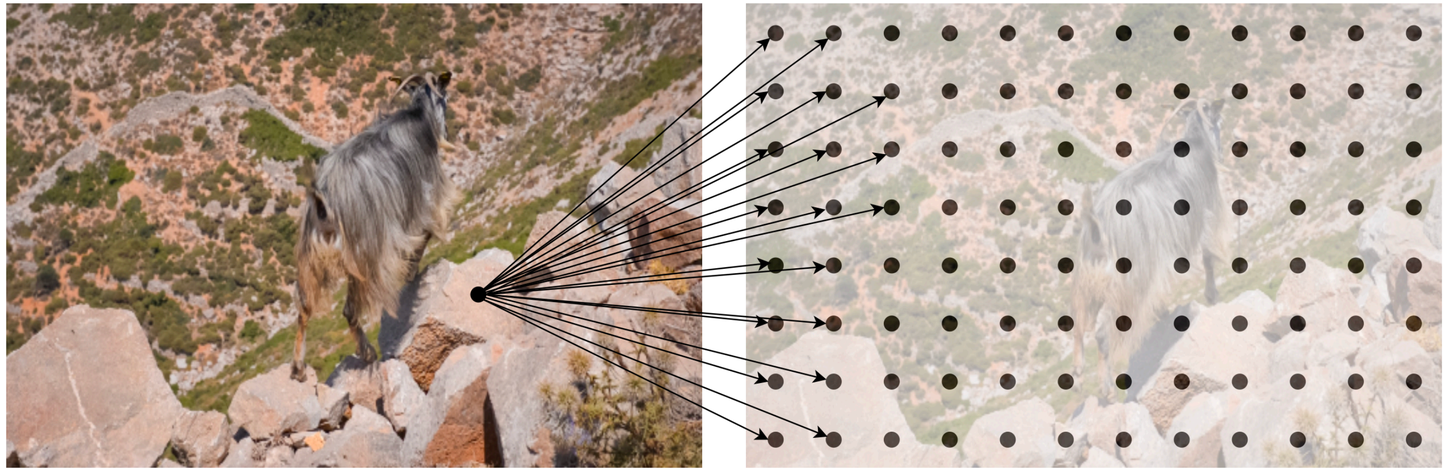
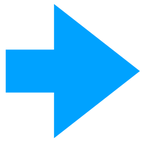
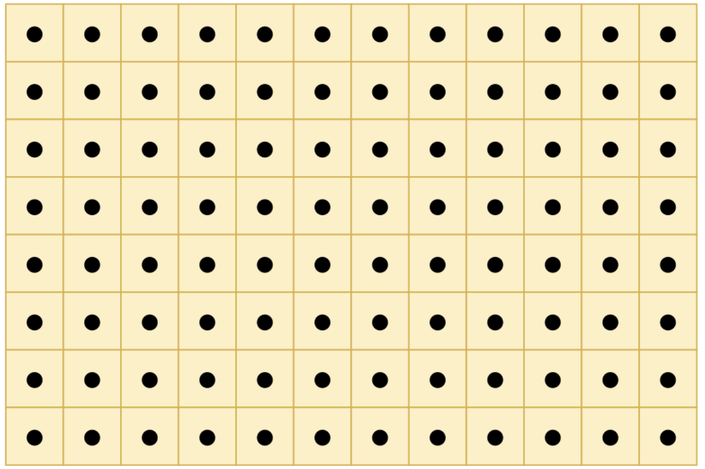


Image 1

Image 2





$\mathbf{C}^1 \in H \times W \times H \times W$

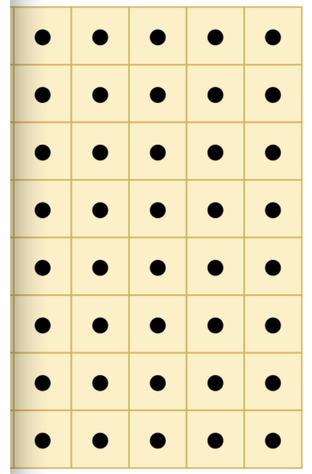
Idea: Correlate every pixel in image 1 with every pixel in image 2. Optical flow can now be computed as a smoothed minimization problem on this cost volume.

Image Courtesy: RAFT. *Teed et. al.*

Computing Optical Flow: Correspondence Volumes

Ima

Blackboard: 2D Example



$\times H \times W$

Idea: Correlate every pixel in image 1 with every pixel in image 2. Optical flow can now be computed as a smoothed minimization problem on this cost volume.

Image Courtesy: RAFT. *Teed et. al.*

Computing Optical Flow: Correspondence Volumes

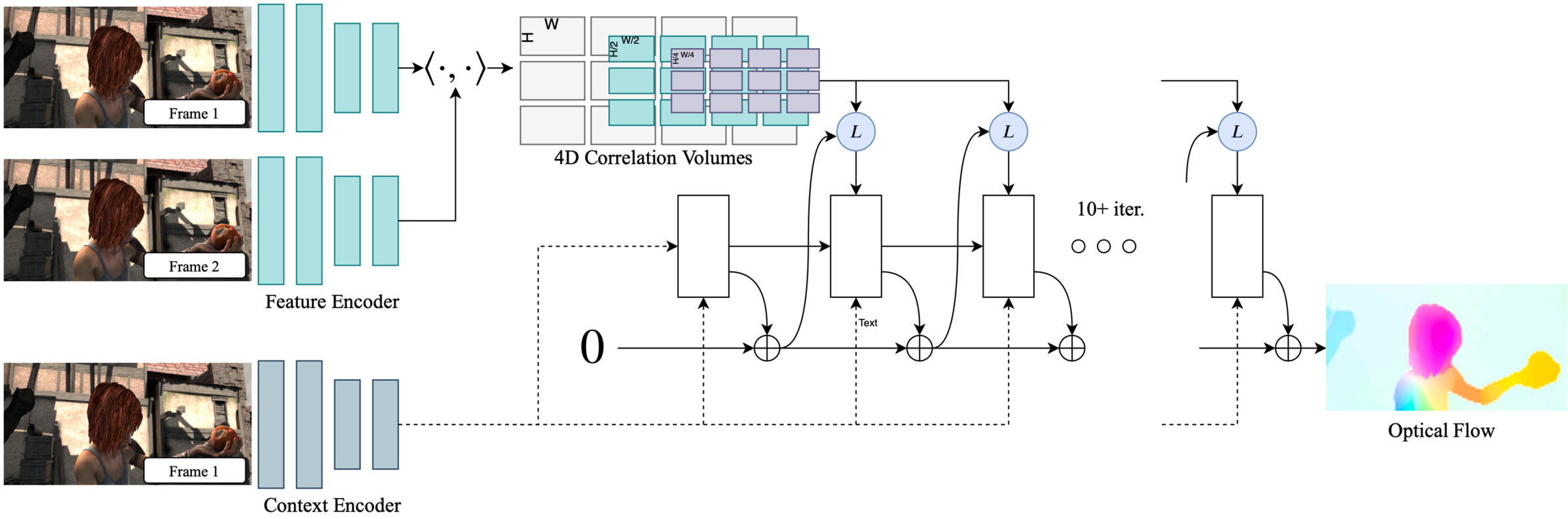


Image Courtesy: RAFT. *Teed et. al.*

Computing Optical Flow: Correspondence Volumes

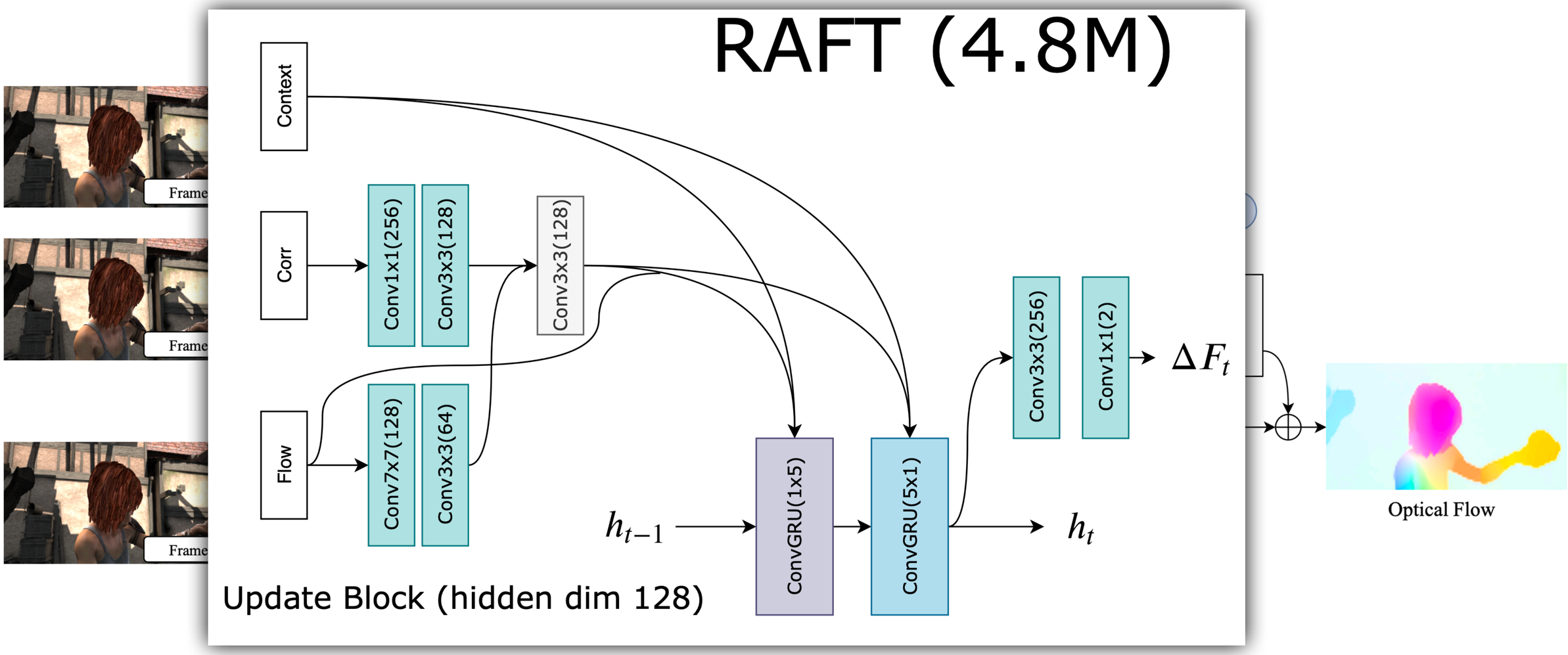


Image Courtesy: RAFT. *Teed et. al.*

Correspondence via Flow Fields

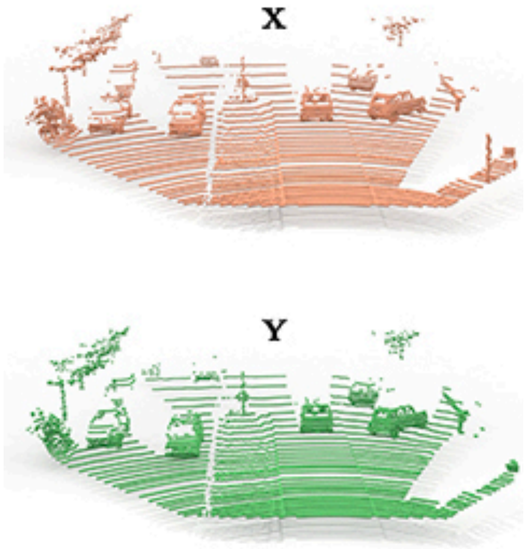
Optical Flow

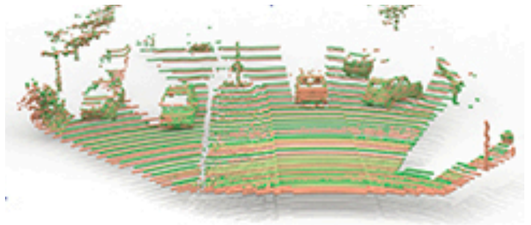
Given a pixel in frame 1, where does it go to in frame 2?

$\mathbb{R}^2 \rightarrow \mathbb{R}^2$ A 2D flow field

Correspondence via Flow Fields

Optical Flow





Per-point scene flow

Given a pixel in frame 1, where does it go to in frame 2?

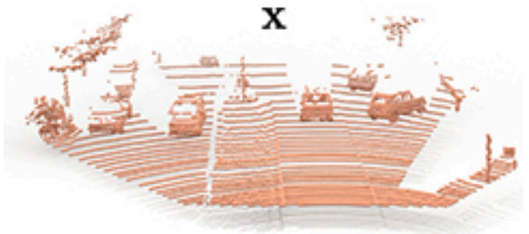
$\mathbb{R}^2 \rightarrow \mathbb{R}^2$ A 2D flow field

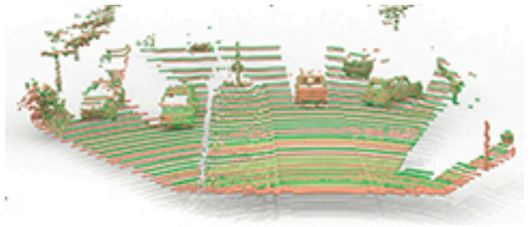
Correspondence via Flow Fields

Optical Flow

Given a pixel in frame 1, where does it go to in frame 2?

$\mathbb{R}^2 \rightarrow \mathbb{R}^2$ A 2D flow field





Per-point scene flow

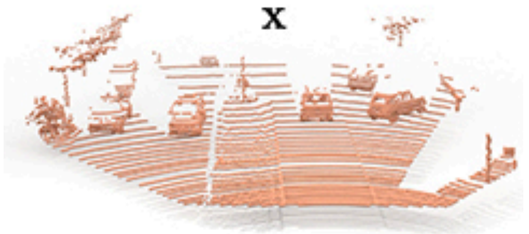
Given a 3D point at time t1, where does it go at time t2?

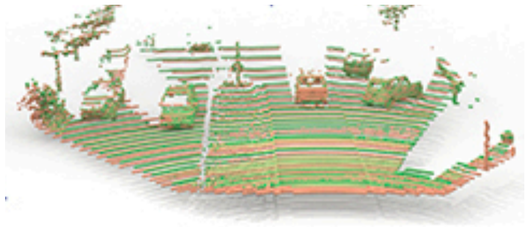
Correspondence via Flow Fields

Optical Flow

Given a pixel in frame 1, where does it go to in frame 2?

$\mathbb{R}^2 \rightarrow \mathbb{R}^2$ A 2D flow field





Per-point scene flow

Given a 3D point at time t1, where does it go at time t2?

$\mathbb{R}^3 \rightarrow \mathbb{R}^3$ A 3D flow field

3D Scene Flow

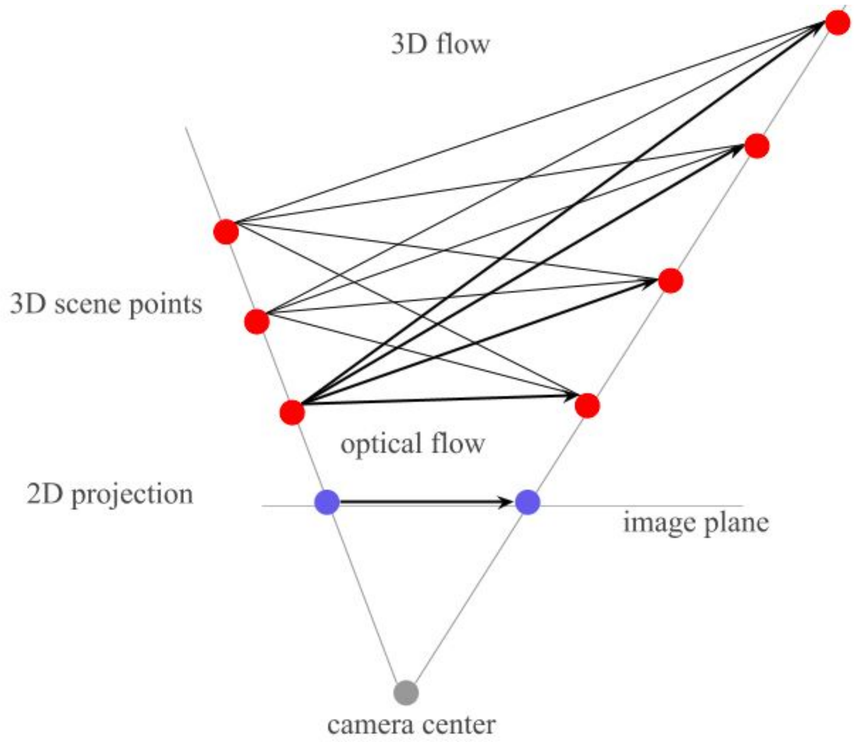
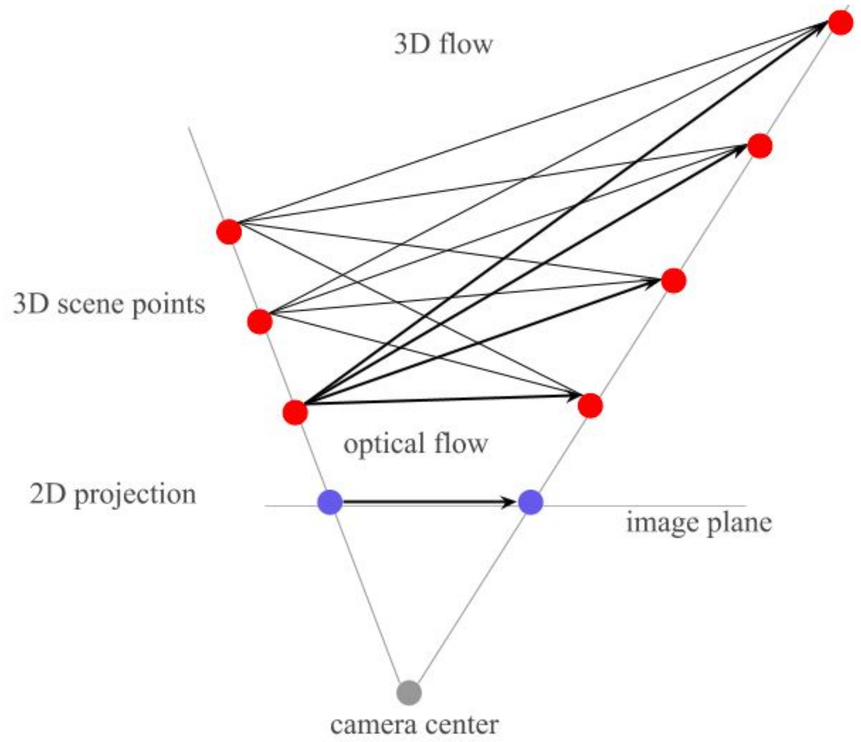


Image Courtesy: Gengshan Yang

Slide curtesy of Shubham Tulsiani, CMU 16-889: Learning for 3D Vision

3D Scene Flow



Observation: 2D flow constrains possible 3D flow (for visible points), and optical flow can be computed given 3D scene flow plus depth

Image Courtesy: Gengshan Yang

3D Scene Flow

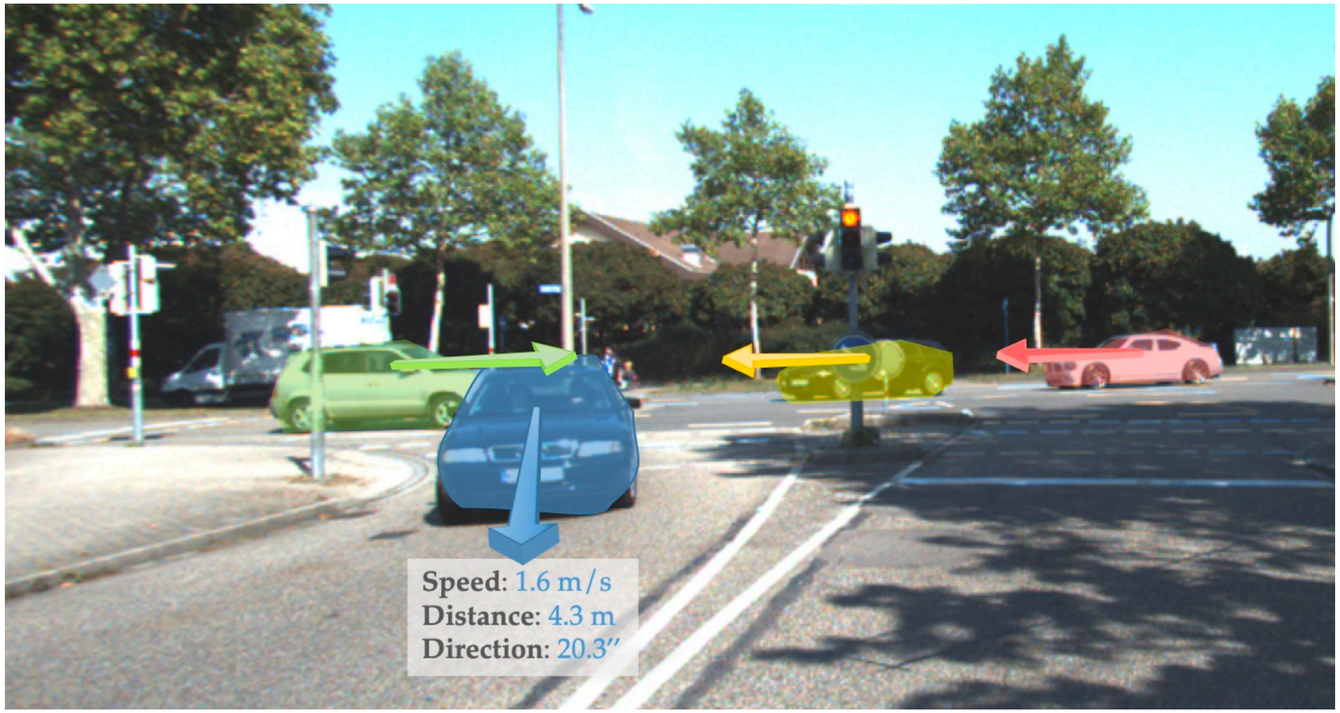
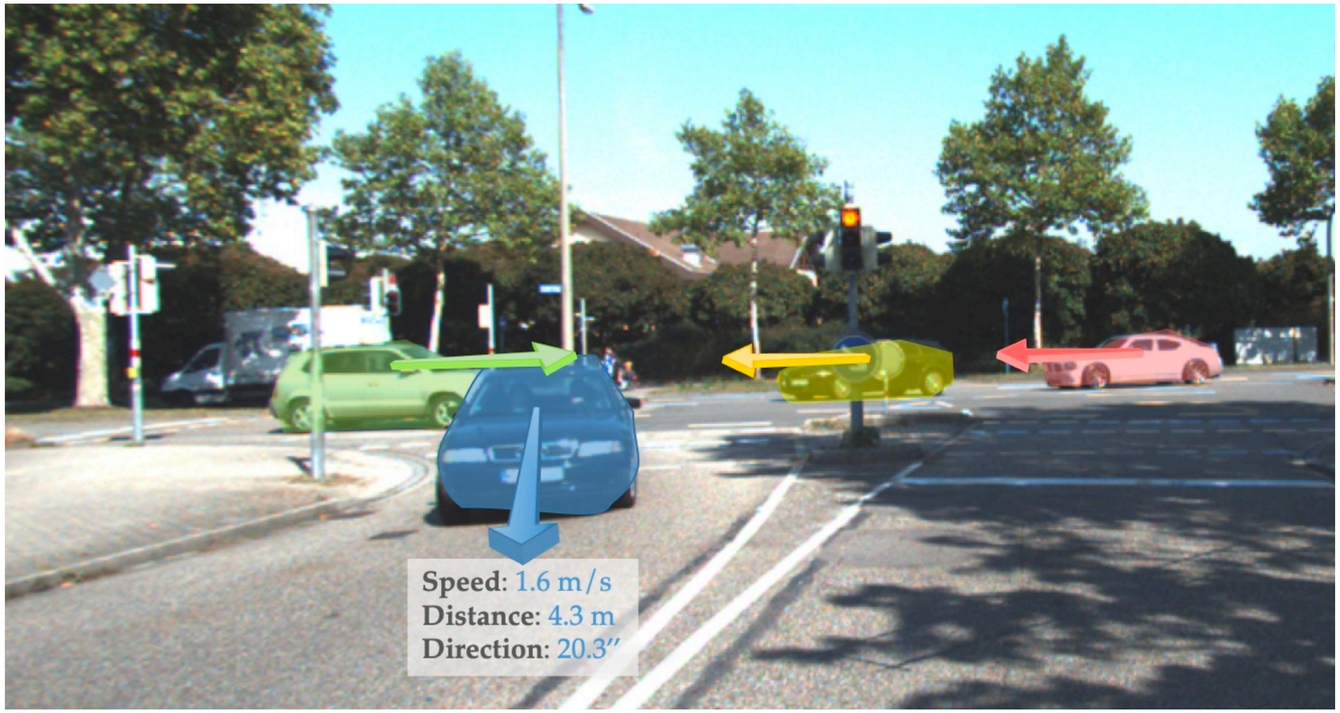


Image Courtesy: Zhe Cao

3D Scene Flow



Observation: 3D flow typically comprised of piecewise rigid motions

Image Courtesy: Zhe Cao

Parametrizing Scene Flow Fields

Image Courtesy: Zhe Cao

Parametrizing Scene Flow Fields

For an object rotating around its centre:

$$\mathbf{x}_{t+1} = R(\mathbf{x}_t - \bar{\mathbf{x}}) + \bar{\mathbf{x}}$$

Image Courtesy: Zhe Cao

Parametrizing Scene Flow Fields

For an object rotating around its centre:

$$\mathbf{x}_{t+1} = R(\mathbf{x}_t - \bar{\mathbf{x}}) + \bar{\mathbf{x}}$$

Option 1: Predict a per-point translation

(requires different predictions for different points)

$$\mathbf{\Phi}(\mathbf{x}) : \mathbb{R}^2 \rightarrow \mathbb{R}^2; \quad \mathbf{\Phi}(\mathbf{x}) = \Delta\mathbf{x}$$

Image Courtesy: Zhe Cao

Parametrizing Scene Flow Fields

For an object rotating around its centre:

$$\mathbf{x}_{t+1} = R(\mathbf{x}_t - \bar{\mathbf{x}}) + \bar{\mathbf{x}}$$

Option 1: Predict a per-point translation

(requires different predictions for different points)

$$\mathbf{\Phi}(\mathbf{x}) : \mathbb{R}^2 \rightarrow \mathbb{R}^2; \quad \mathbf{\Phi}(\mathbf{x}) = \Delta\mathbf{x}$$

Image Courtesy: Zhe Cao

Option 2: Predict a per-point translation **and** rotation

(allows **same** prediction for **all** points within a rigid object)

$$\mathbf{\Phi}(\mathbf{x}) : \mathbb{R}^2 \rightarrow SE(3); \quad \mathbf{\Phi}(\mathbf{x}) = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ 0 & 1 \end{bmatrix}$$

Predicting 3D Scene Flow

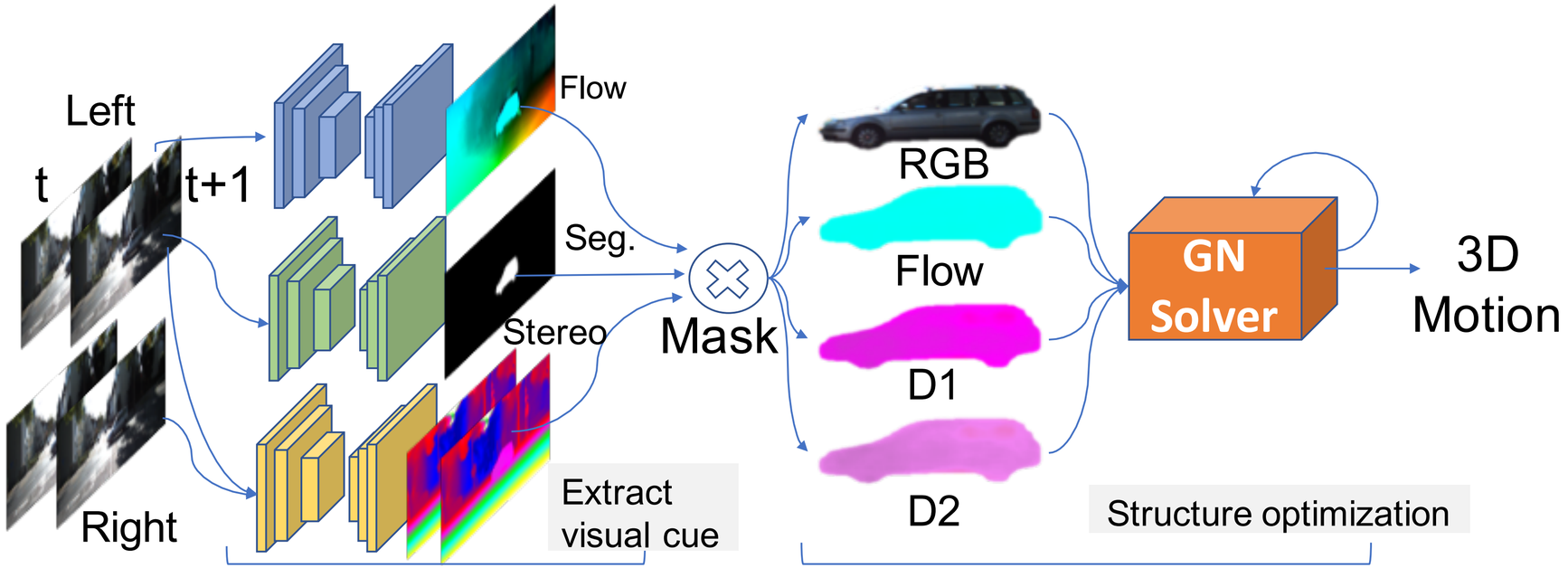


Figure 2: **Overview of our approach:** Given two consecutive stereo images, we first estimate the flow, stereo, and segmentation (Sec. 3.1). The visual cues of each instance are then encoded as energy functions (Sec. 3.2) and passed into the Gaussian-Newton (GN) solver to find the best 3D rigid motion (Sec. 3.3). The GN solver is unrolled as a recurrent network.

Image Courtesy: Deep Rigid Instance Scene Flow. Ma et. al.

Predicting 3D Scene Flow

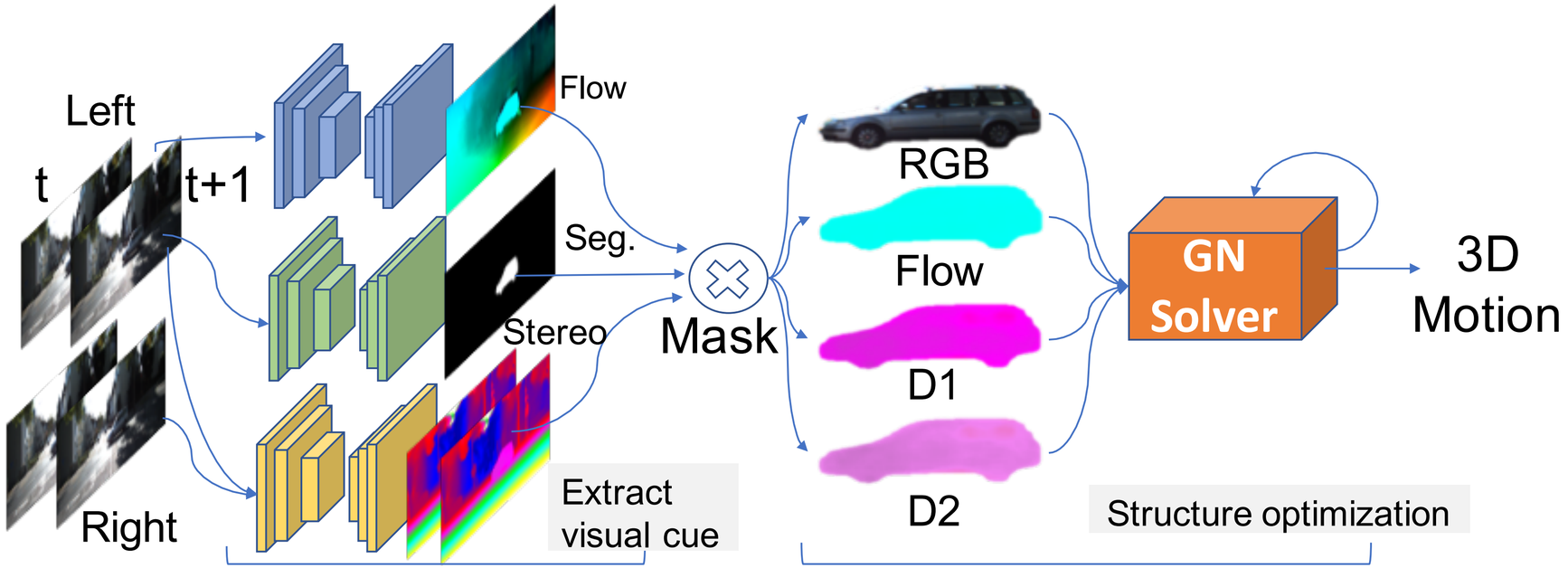


Figure 2: **Overview of our approach:** Given two consecutive stereo images, we first estimate the flow, stereo, and segmentation (Sec. 3.1). The visual cues of each instance are then encoded as energy functions (Sec. 3.2) and passed into the Gaussian-Newton (GN) solver to find the best 3D rigid motion (Sec. 3.3). The GN solver is unrolled as a recurrent network.

Estimate a per-instance 3D motion consistent with predicted depths and flow

Image Courtesy: Deep Rigid Instance Scene Flow. Ma et. al.

Predicting 3D Scene Flow

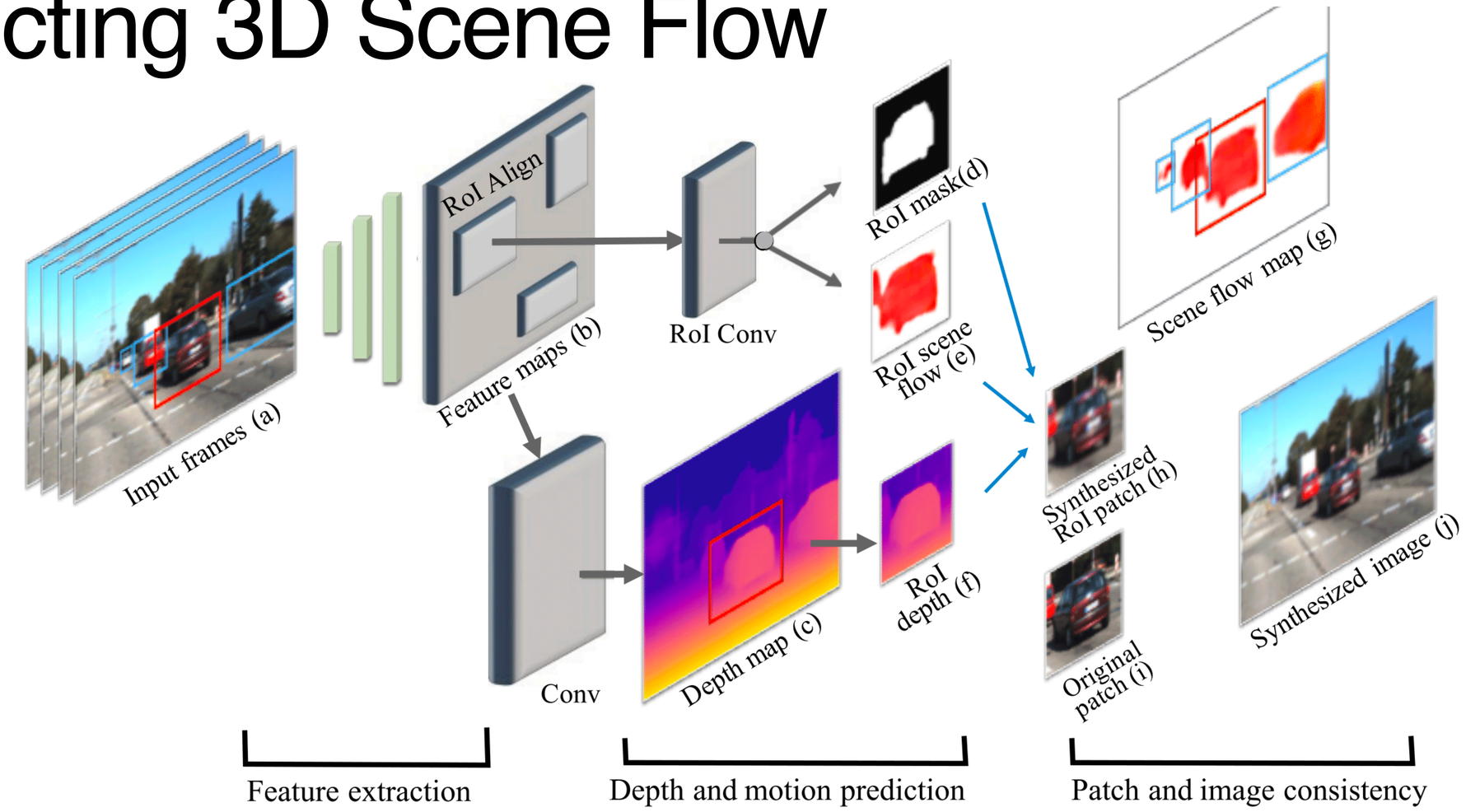
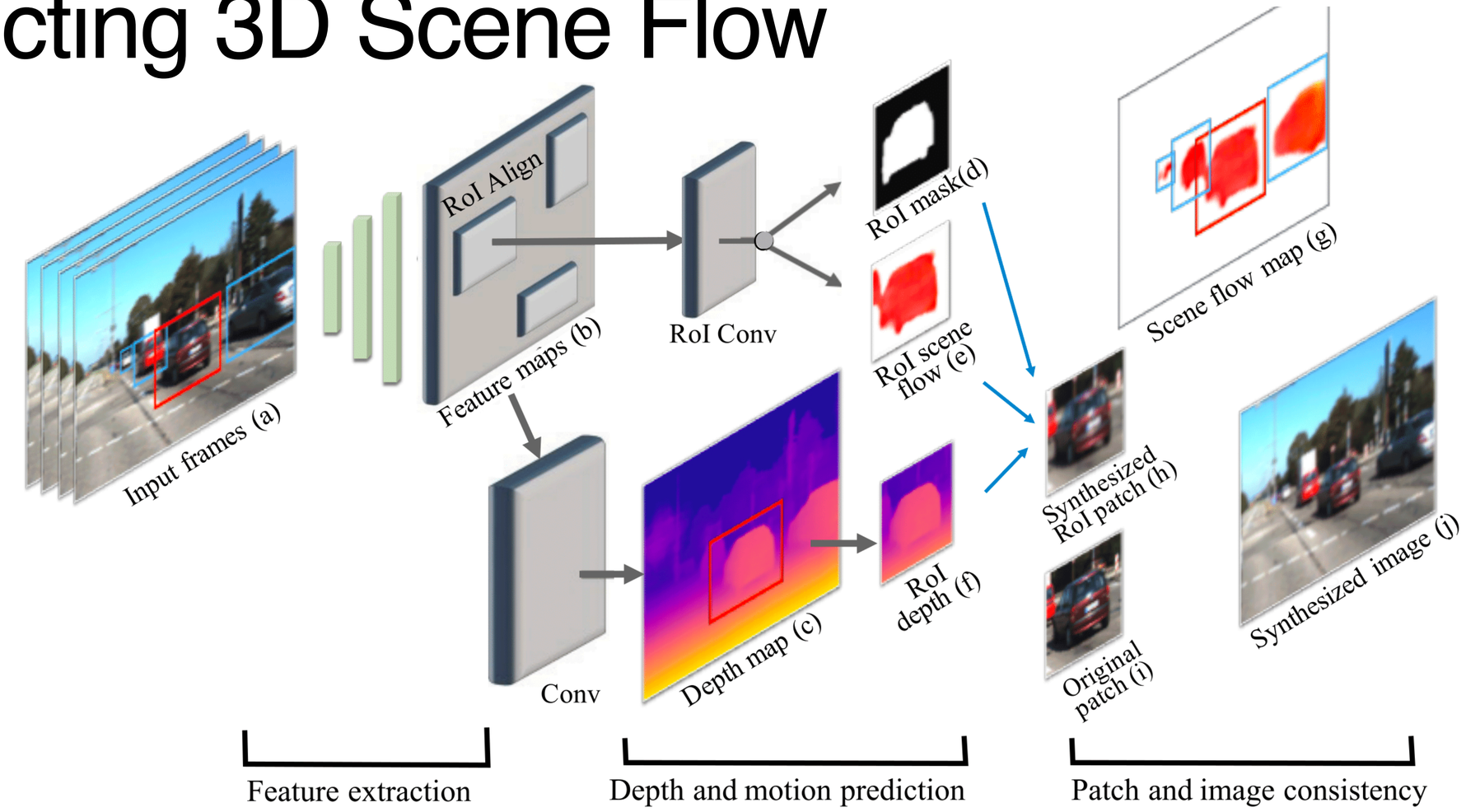


Image Courtesy: Learning Independent Object Motion from Unlabelled Stereoscopic Videos. Cao et. al.

Predicting 3D Scene Flow



Self-supervised learning of depth and scene flow

Image Courtesy: Learning Independent Object Motion from Unlabelled Stereoscopic Videos. Cao et. al.

Predicting 3D Scene Flow

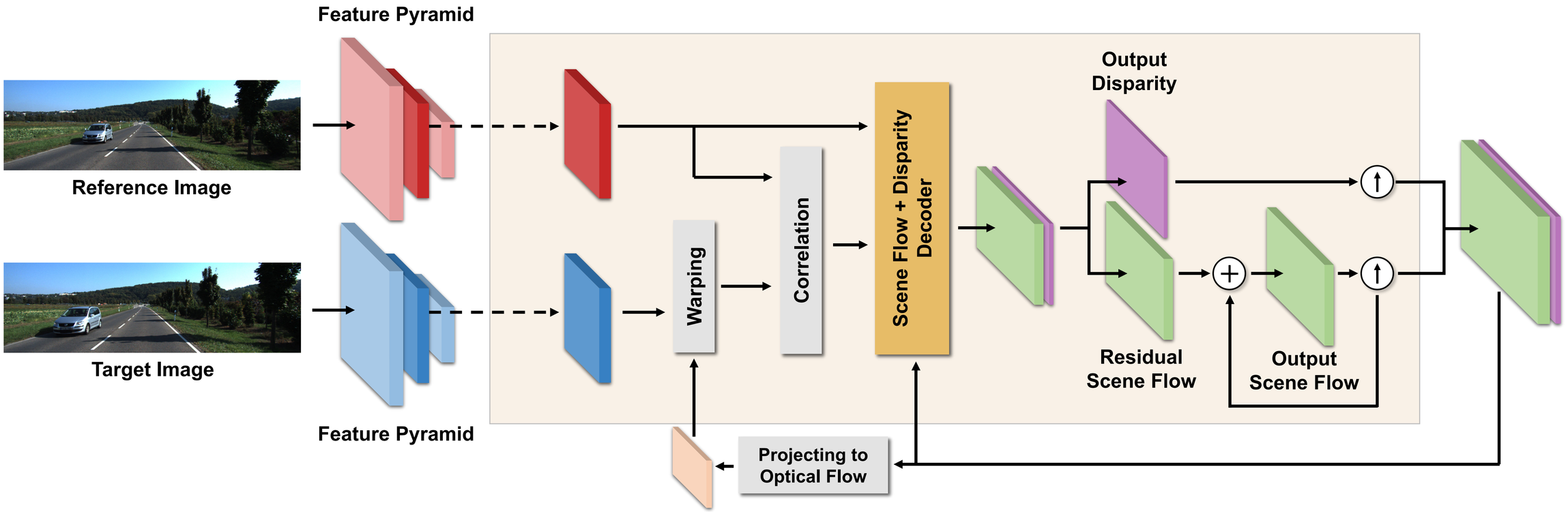
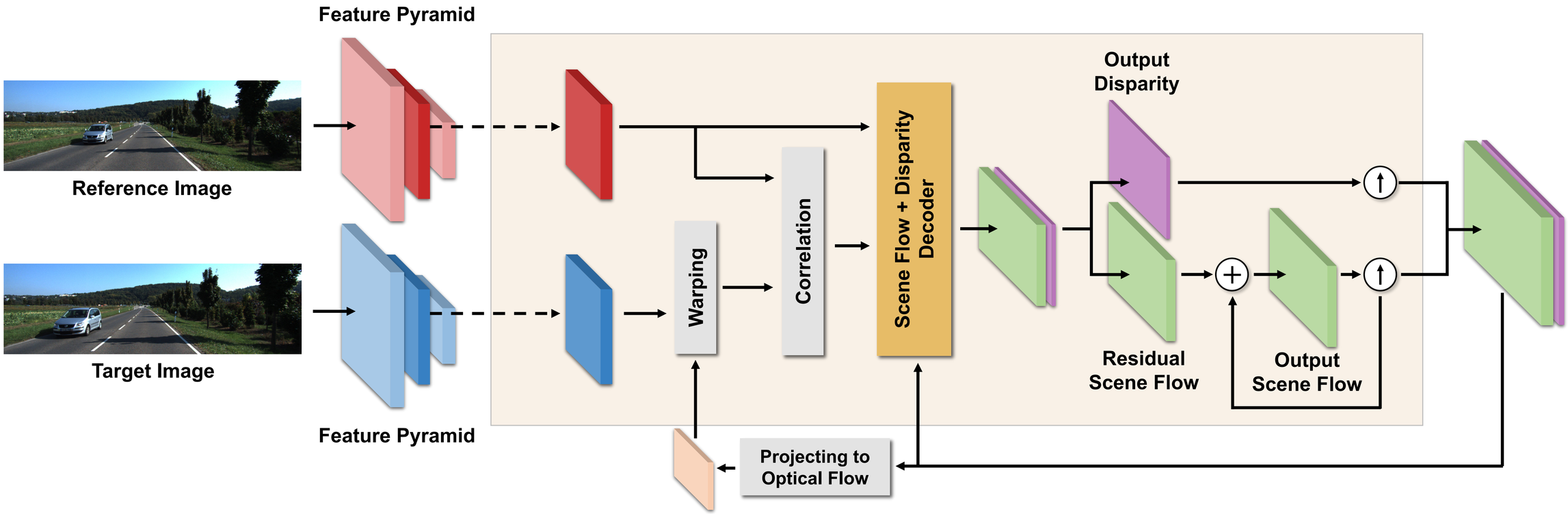


Image Courtesy: Self-Supervised Monocular Scene Flow Estimation. Hur et. al.

Predicting 3D Scene Flow



Self-supervised learning of depth and scene flow

Image Courtesy: Self-Supervised Monocular Scene Flow Estimation. Hur et. al.

Predicting 3D Scene Flow

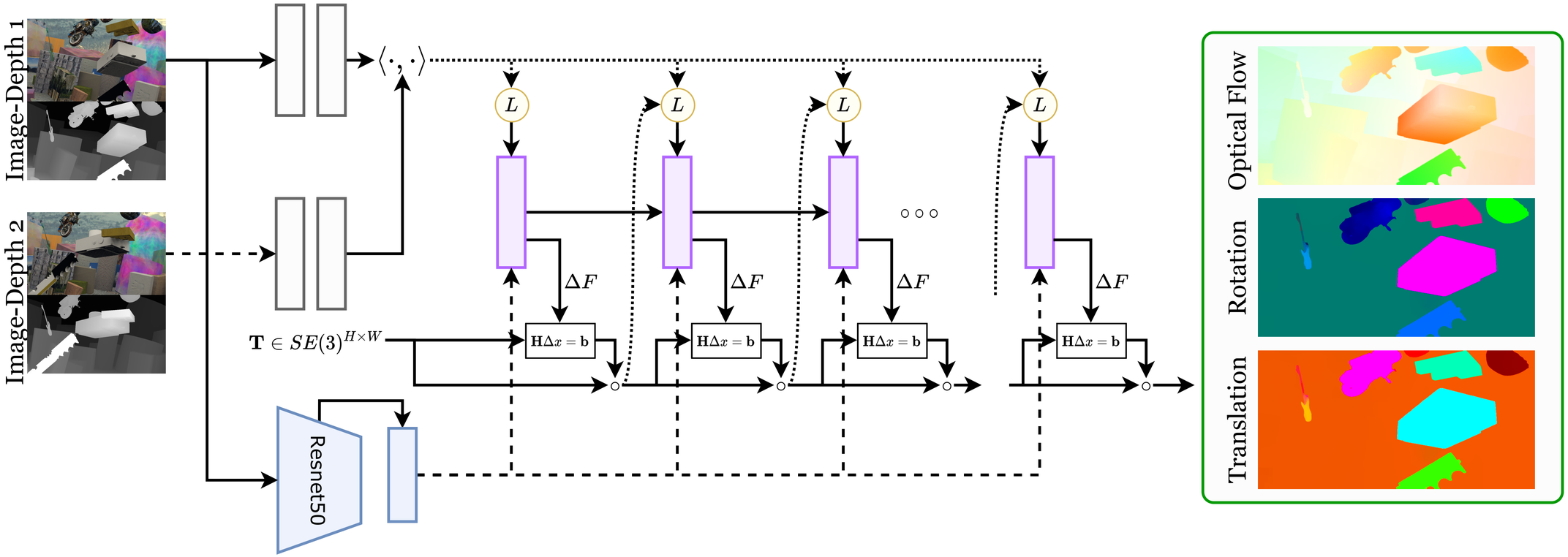
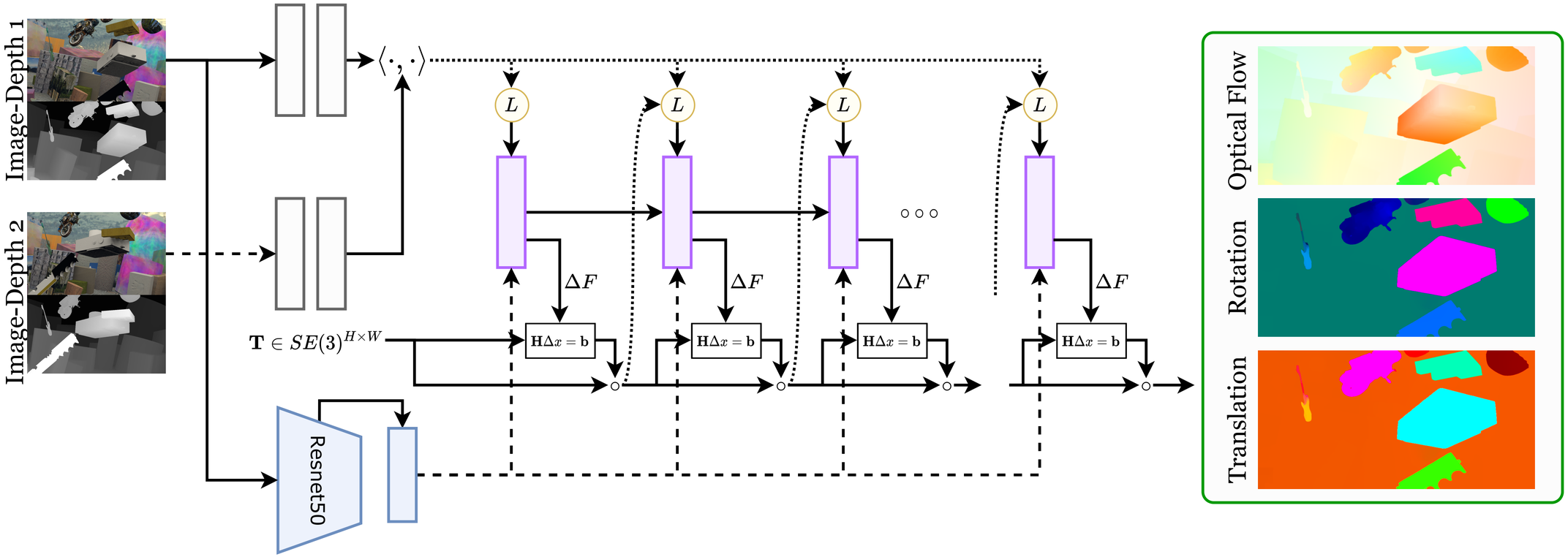


Image Courtesy: RAFT-3D: Scene Flow using Rigid-Motion Embeddings. Teed et. al.

Slide curtesy of Shubham Tulsiani, CMU 16-889: Learning for 3D Vision

Predicting 3D Scene Flow



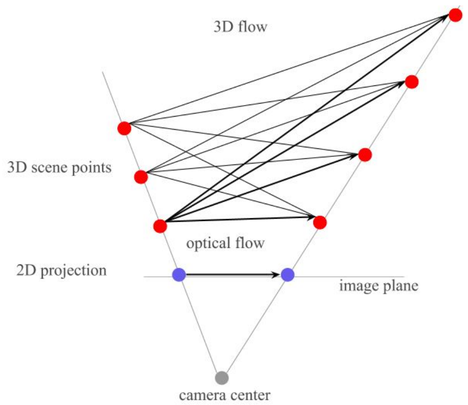
Iteratively update pixelwise motion fields in context of a 2D cost volume while enforcing that nearby pixels have a similar motion

Image Courtesy: RAFT-3D: Scene Flow using Rigid-Motion Embeddings. Teed et. al.

3D Scene Flow: Some takeaways

3D Scene Flow: Some takeaways

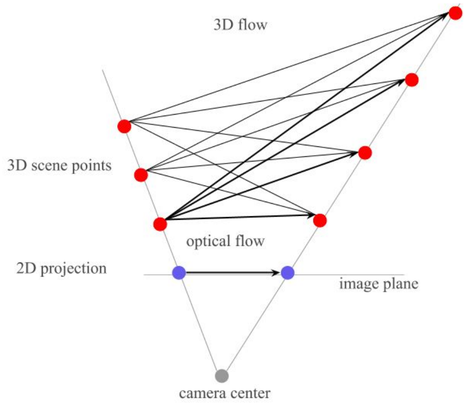
Scene flow allows predicting optical flow



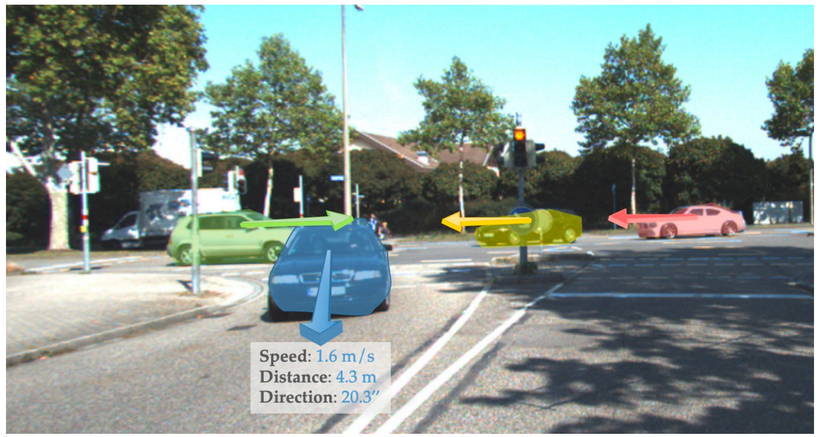
Slide curtesy of Shubham Tulsiani, CMU 16-889: Learning for 3D Vision

3D Scene Flow: Some takeaways

Scene flow allows predicting optical flow



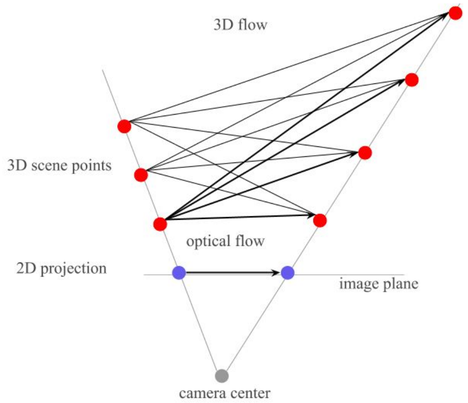
Typical motions are piecewise rigid



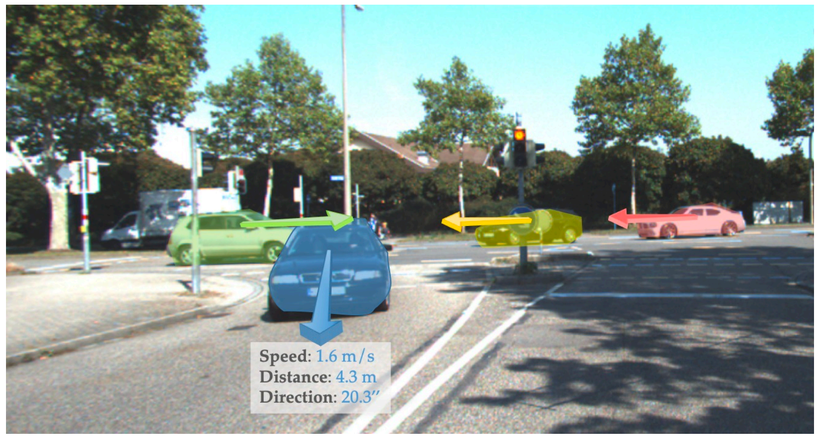
Slide curtesy of Shubham Tulsiani, CMU 16-889: Learning for 3D Vision

3D Scene Flow: Some takeaways

Scene flow allows predicting optical flow



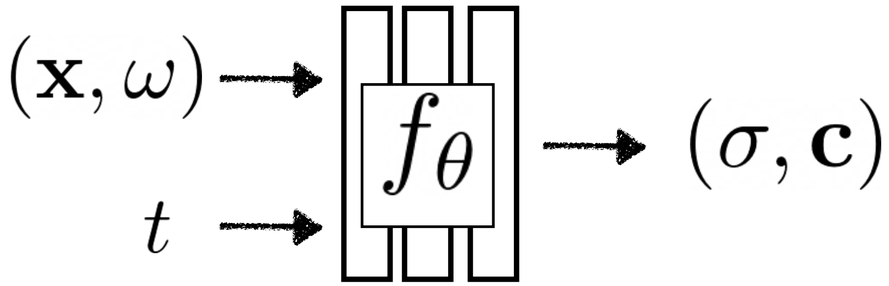
Typical motions are piecewise rigid



Possible parametrization as pointwise SE3 field (instead of pointwise translation)

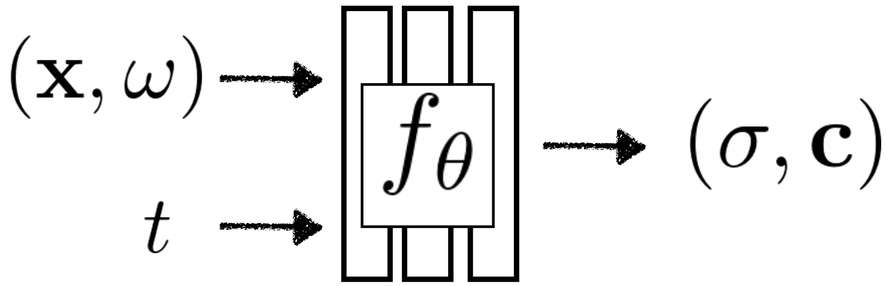
Slide curtesy of Shubham Tulsiani, CMU 16-889: Learning for 3D Vision

Towards Modeling a Dynamic World



Slide curtesy of Shubham Tulsiani, CMU 16-889: Learning for 3D Vision

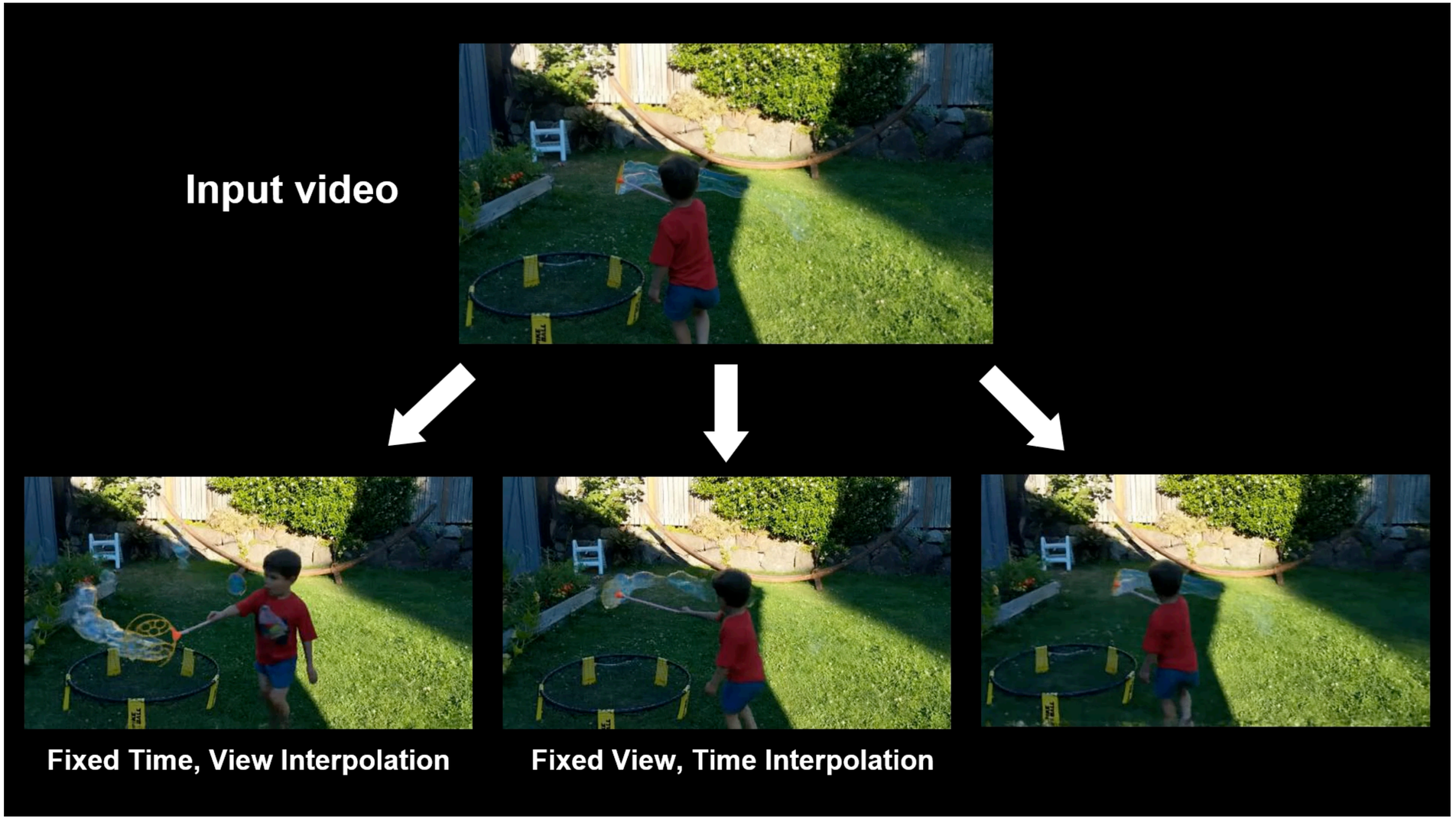
Towards Modeling a Dynamic World



Can we add explicit constraints regarding scene persistence?

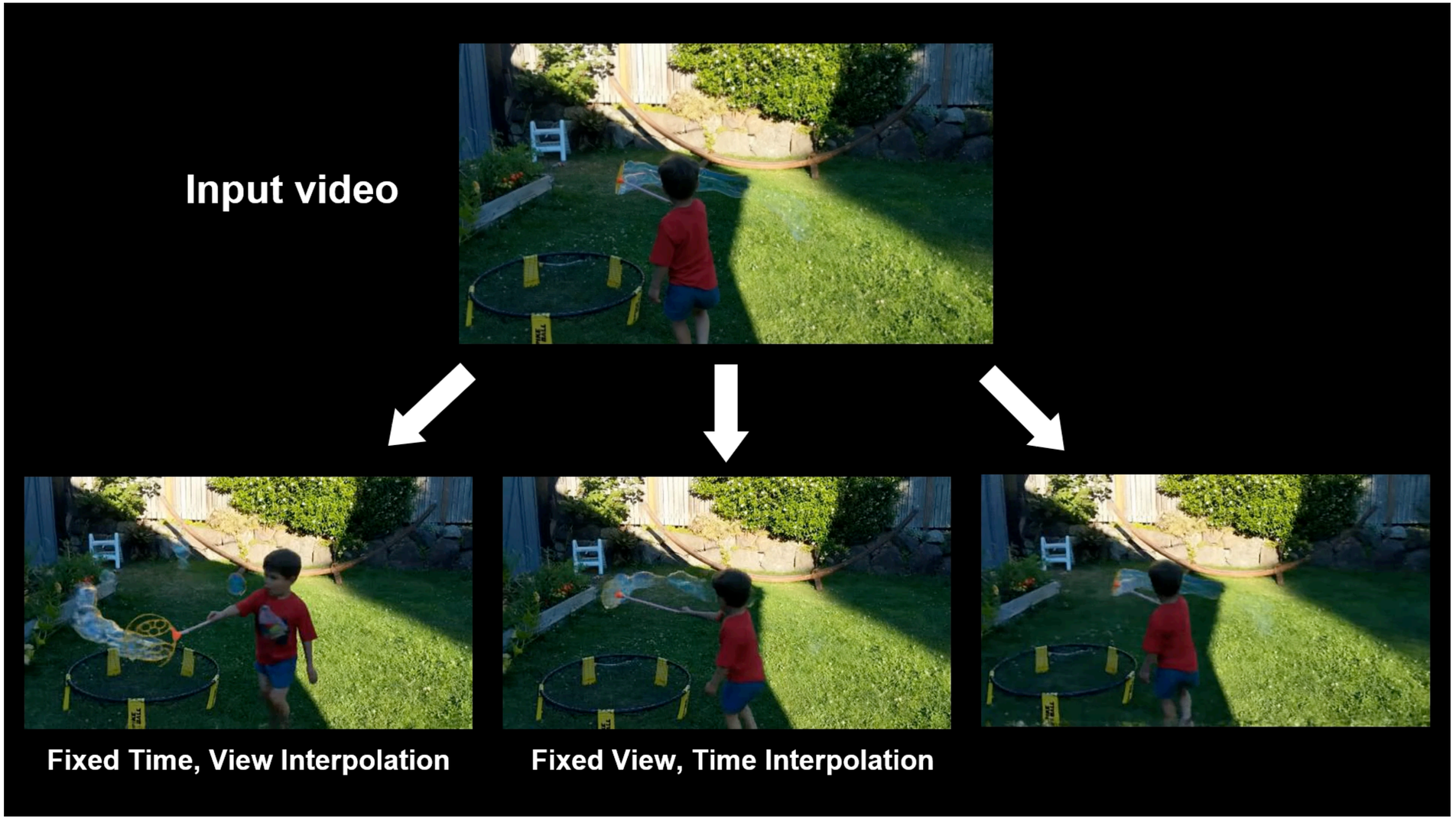
Slide curtesy of Shubham Tulsiani, CMU 16-889: Learning for 3D Vision

Towards Modeling a Dynamic World



Video Courtesy: Neural Scene Flow Fields for Space-Time View Synthesis of Dynamic Scenes. *Li et. al.*

Towards Modeling a Dynamic World



Video Courtesy: Neural Scene Flow Fields for Space-Time View Synthesis of Dynamic Scenes. *Li et. al.*

Neural Scene Flow Fields

$$(\mathbf{c}_i, \sigma_i, \mathcal{F}_i, \mathcal{W}_i) = F_\Theta^{\text{dy}}(\mathbf{x}, \mathbf{d}, i)$$

Neural Scene Flow Fields for Space-Time View Synthesis of Dynamic Scenes. *Li et. al.*

Neural Scene Flow Fields

$$(\mathbf{c}_i, \sigma_i, \mathcal{F}_i, \mathcal{W}_i) = F_\Theta^{\text{dy}}(\mathbf{x}, \mathbf{d}, i)$$

Basically, a time-dependent NeRF representation (***i*** is indexing time)

Neural Scene Flow Fields for Space-Time View Synthesis of Dynamic Scenes. *Li et. al.*

Neural Scene Flow Fields

$$(\mathbf{c}_i, \sigma_i, \mathcal{F}_i, \mathcal{W}_i) = F_\Theta^{\text{dy}}(\mathbf{x}, \mathbf{d}, i)$$

$\mathcal{F}_i$ indicates the 3D flow to next/previous frame for point **x**

Image Courtesy: Neural Scene Flow Fields for Space-Time View Synthesis of Dynamic Scenes. *Li et. al.*

Neural Scene Flow Fields

$$(\mathbf{c}_i, \sigma_i, \mathcal{F}_i, \mathcal{W}_i) = F_\Theta^{\text{dy}}(\mathbf{x}, \mathbf{d}, i)$$

$\mathcal{F}_i$ indicates the 3D flow to next/previous frame for point **x**

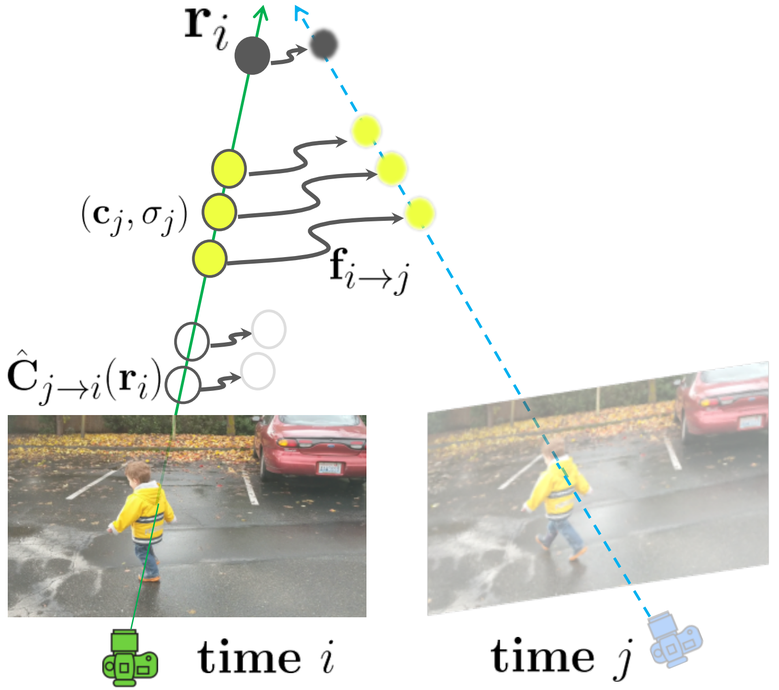


Image Courtesy: Neural Scene Flow Fields for Space-Time View Synthesis of Dynamic Scenes. *Li et. al.*

Enforcing Temporal Persistence

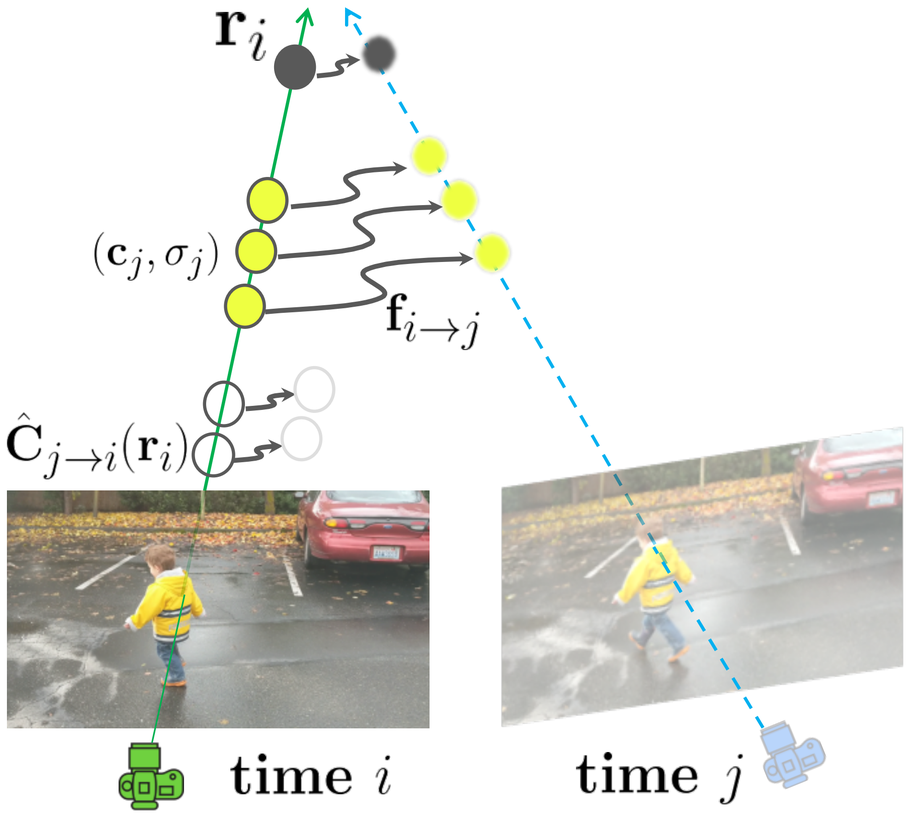
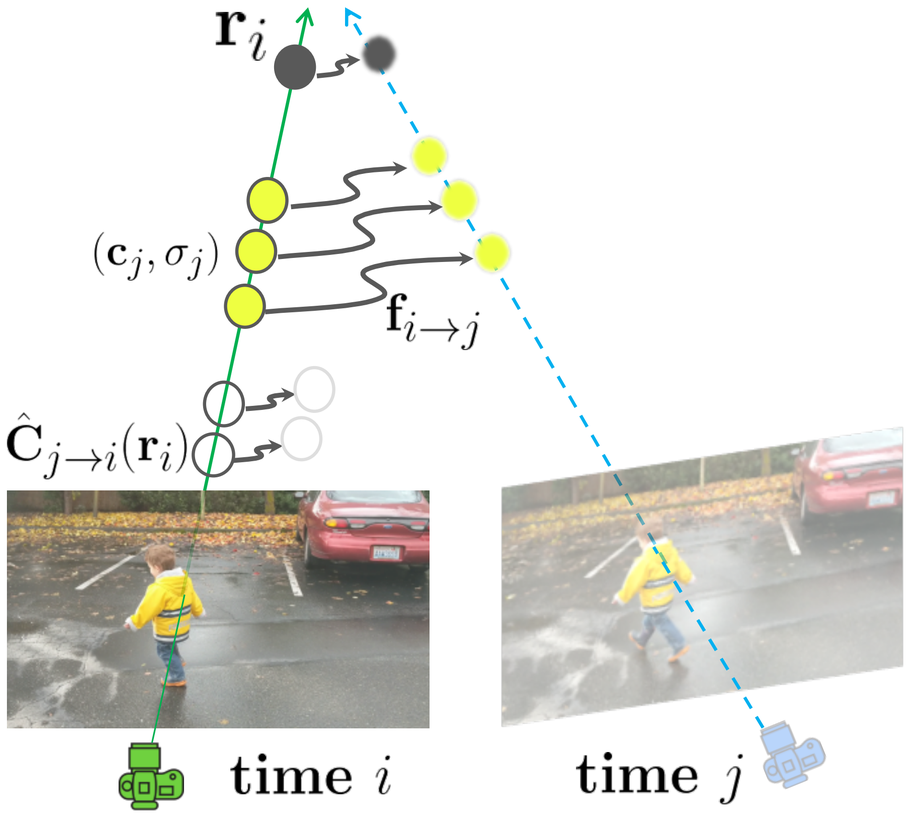


Image Courtesy: Neural Scene Flow Fields for Space-Time View Synthesis of Dynamic Scenes. *Li et. al.*

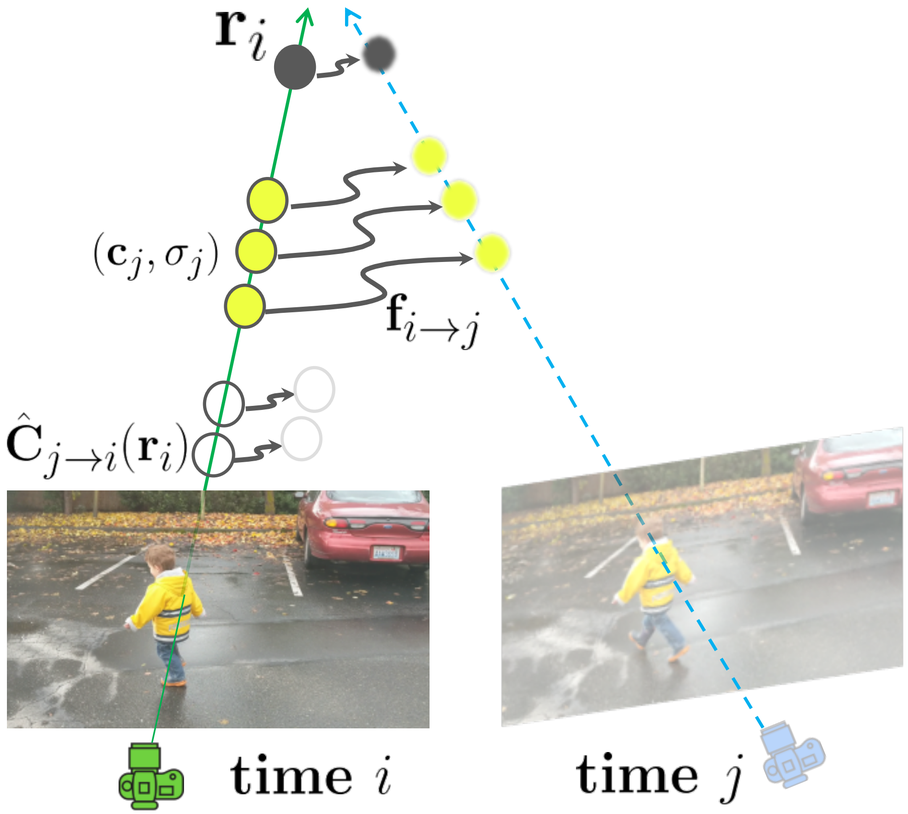
Enforcing Temporal Persistence



Rendering for a ray ***r*** at time ***i*** using the NeRF for time ***j***

Image Courtesy: Neural Scene Flow Fields for Space-Time View Synthesis of Dynamic Scenes. *Li et. al.*

Enforcing Temporal Persistence



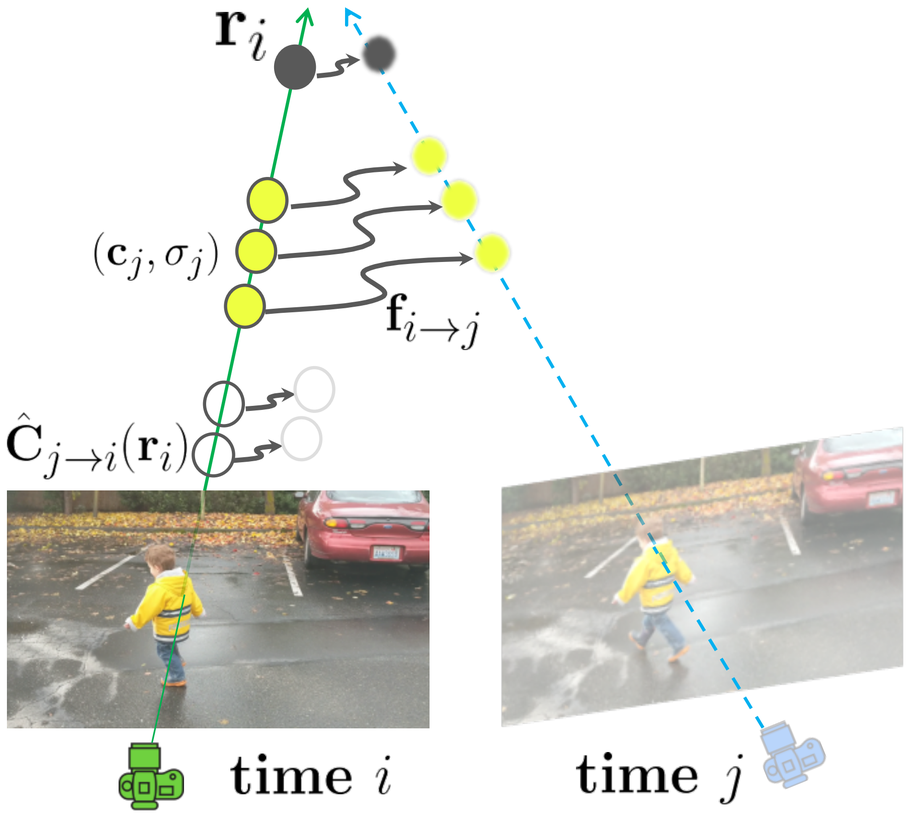
$$\hat{\mathbf{C}}_{j \to i}(\mathbf{r}_i) = \int_{t_n}^{t_f} T_j(t)\, \sigma_j(\mathbf{r}_{i \to j}(t))\, \mathbf{c}_j(\mathbf{r}_{i \to j}(t), \mathbf{d}_i) dt$$

$$\text{where } \mathbf{r}_{i \to j}(t) = \mathbf{r}_i(t) + \mathbf{f}_{i \to j}(\mathbf{r}_i(t)).$$

Rendering for a ray ***r*** at time ***i*** using the NeRF for time ***j***

Image Courtesy: Neural Scene Flow Fields for Space-Time View Synthesis of Dynamic Scenes. *Li et. al.*

Enforcing Temporal Persistence



$$\hat{\mathbf{C}}_{j \to i}(\mathbf{r}_i) = \int_{t_n}^{t_f} T_j(t)\, \sigma_j(\mathbf{r}_{i \to j}(t))\, \mathbf{c}_j(\mathbf{r}_{i \to j}(t), \mathbf{d}_i) dt$$

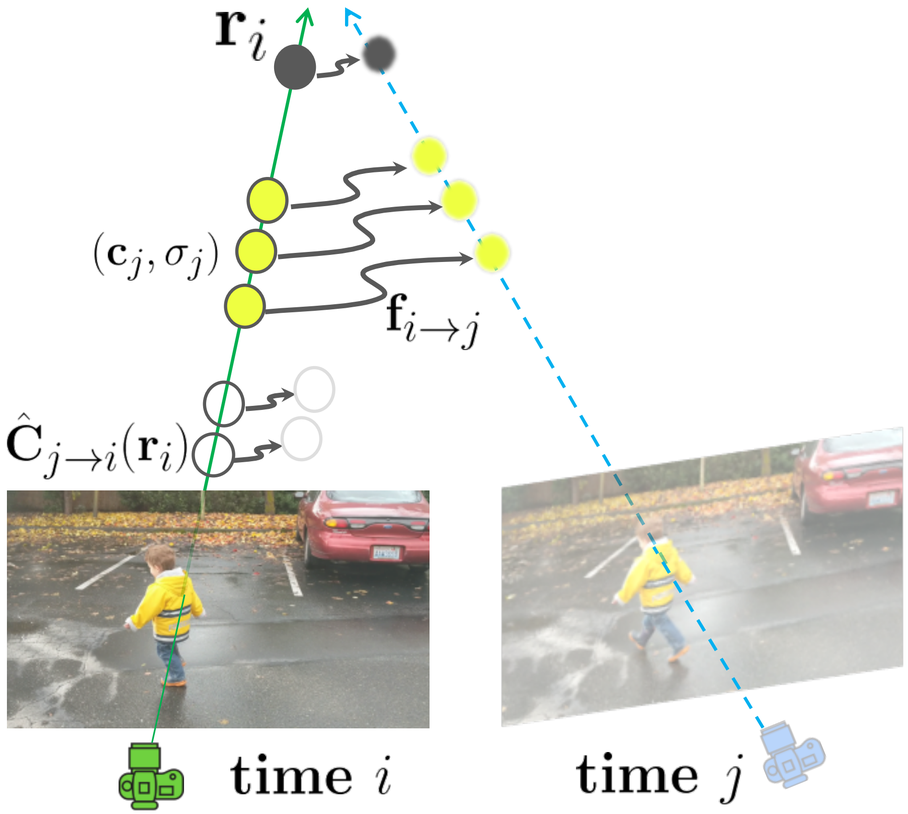
$$\text{where } \mathbf{r}_{i \to j}(t) = \mathbf{r}_i(t) + \mathbf{f}_{i \to j}(\mathbf{r}_i(t)).$$

$$\mathcal{L}_{\text{pho}} = \sum_{\mathbf{r}_i} \sum_{j \in \mathcal{N}(i)} ||\hat{\mathbf{C}}_{j \to i}(\mathbf{r}_i) - \mathbf{C}_i(\mathbf{r}_i)||_2^2$$

This should match the observed image at time ***i***

Image Courtesy: Neural Scene Flow Fields for Space-Time View Synthesis of Dynamic Scenes. *Li et. al.*

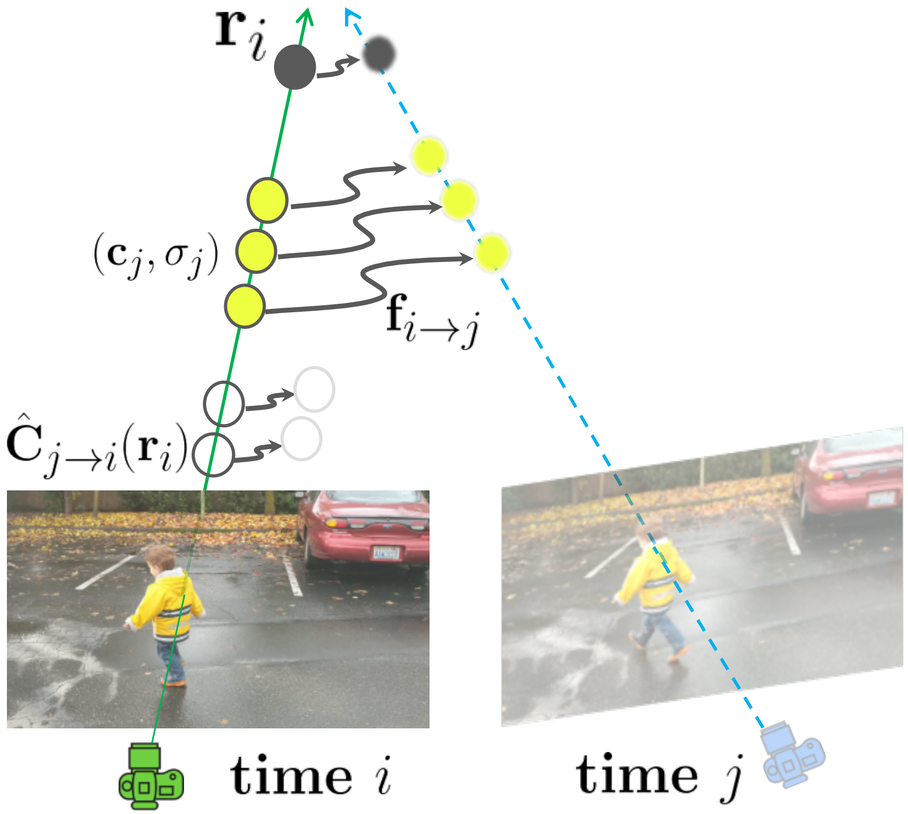
Scene Flow Fields: additional regularizations



$$\mathcal{L}_{\text{cyc}} = \sum_{\mathbf{x}_i} \sum_{j \in i \pm 1} ||\mathbf{f}_{i \to j}(\mathbf{x}_i) + \mathbf{f}_{j \to i}(\mathbf{x}_{i \to j})||_1$$

Image Courtesy: Neural Scene Flow Fields for Space-Time View Synthesis of Dynamic Scenes. *Li et. al.*

Scene Flow Fields: additional regularizations

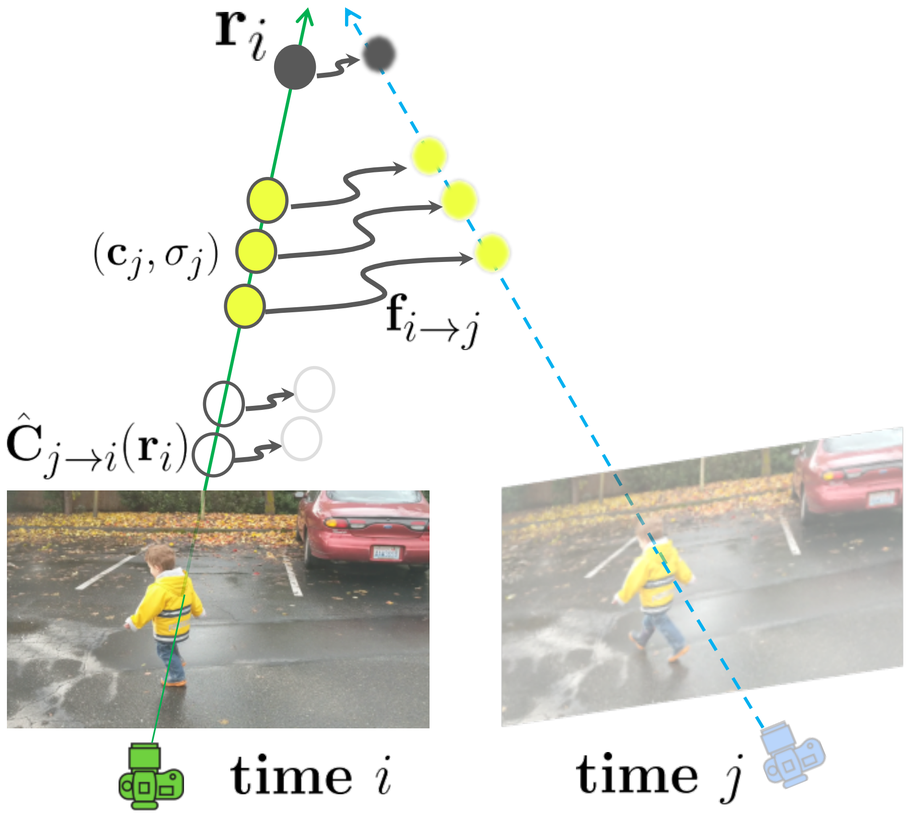


$$\mathcal{L}_{\text{cyc}} = \sum_{\mathbf{x}_i} \sum_{j \in i \pm 1} ||\mathbf{f}_{i \to j}(\mathbf{x}_i) + \mathbf{f}_{j \to i}(\mathbf{x}_{i \to j})||_1$$

The scene flow should be cycle consistent

Image Courtesy: Neural Scene Flow Fields for Space-Time View Synthesis of Dynamic Scenes. *Li et. al.*

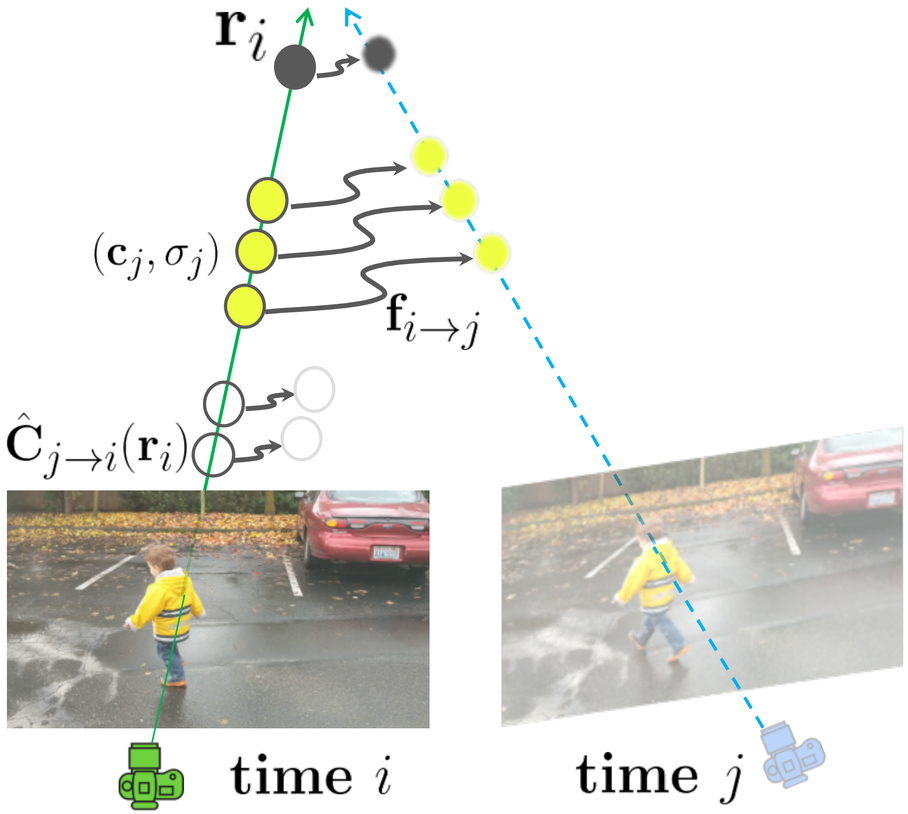
Scene Flow Fields: additional regularizations



$$\mathcal{L}_{\text{geo}} = \sum_{\mathbf{r}_i} \sum_{j \in \{i \pm 1\}} ||\hat{\mathbf{p}}_{i \to j}(\mathbf{r}_i) - \mathbf{p}_{i \to j}(\mathbf{r}_i))||_1$$

Image Courtesy: Neural Scene Flow Fields for Space-Time View Synthesis of Dynamic Scenes. *Li et. al.*

Scene Flow Fields: additional regularizations



$$\mathcal{L}_{\text{geo}} = \sum_{\mathbf{r}_i} \sum_{j \in \{i \pm 1\}} ||\hat{\mathbf{p}}_{i \to j}(\mathbf{r}_i) - \mathbf{p}_{i \to j}(\mathbf{r}_i))||_1$$

The scene flow match with off-the-shelf optical flow

Image Courtesy: Neural Scene Flow Fields for Space-Time View Synthesis of Dynamic Scenes. *Li et. al.*

Neural Scene Flow Fields: Some other ‘tricks’

$$(\mathbf{c}_i, \sigma_i, \mathcal{F}_i, \mathcal{W}_i) = F_\Theta^{\text{dy}}(\mathbf{x}, \mathbf{d}, i)$$

W as ‘confidence’ — used as weight in objectives, encouraged to be 1

Image Courtesy: Neural Scene Flow Fields for Space-Time View Synthesis of Dynamic Scenes. *Li et. al.*

Neural Scene Flow Fields: Some other ‘tricks’

$$(\mathbf{c}_i, \sigma_i, \mathcal{F}_i, \mathcal{W}_i) = F_\Theta^{\text{dy}}(\mathbf{x}, \mathbf{d}, i)$$

W as ‘confidence’ — used as weight in objectives, encouraged to be 1

Use off-the-shelf depth prediction to inform geometry (rendered depth = predicted depth)

Image Courtesy: Neural Scene Flow Fields for Space-Time View Synthesis of Dynamic Scenes. *Li et. al.*

Neural Scene Flow Fields: Some other ‘tricks’

$$(\mathbf{c}_i, \sigma_i, \mathcal{F}_i, \mathcal{W}_i) = F_\Theta^{\text{dy}}(\mathbf{x}, \mathbf{d}, i)$$

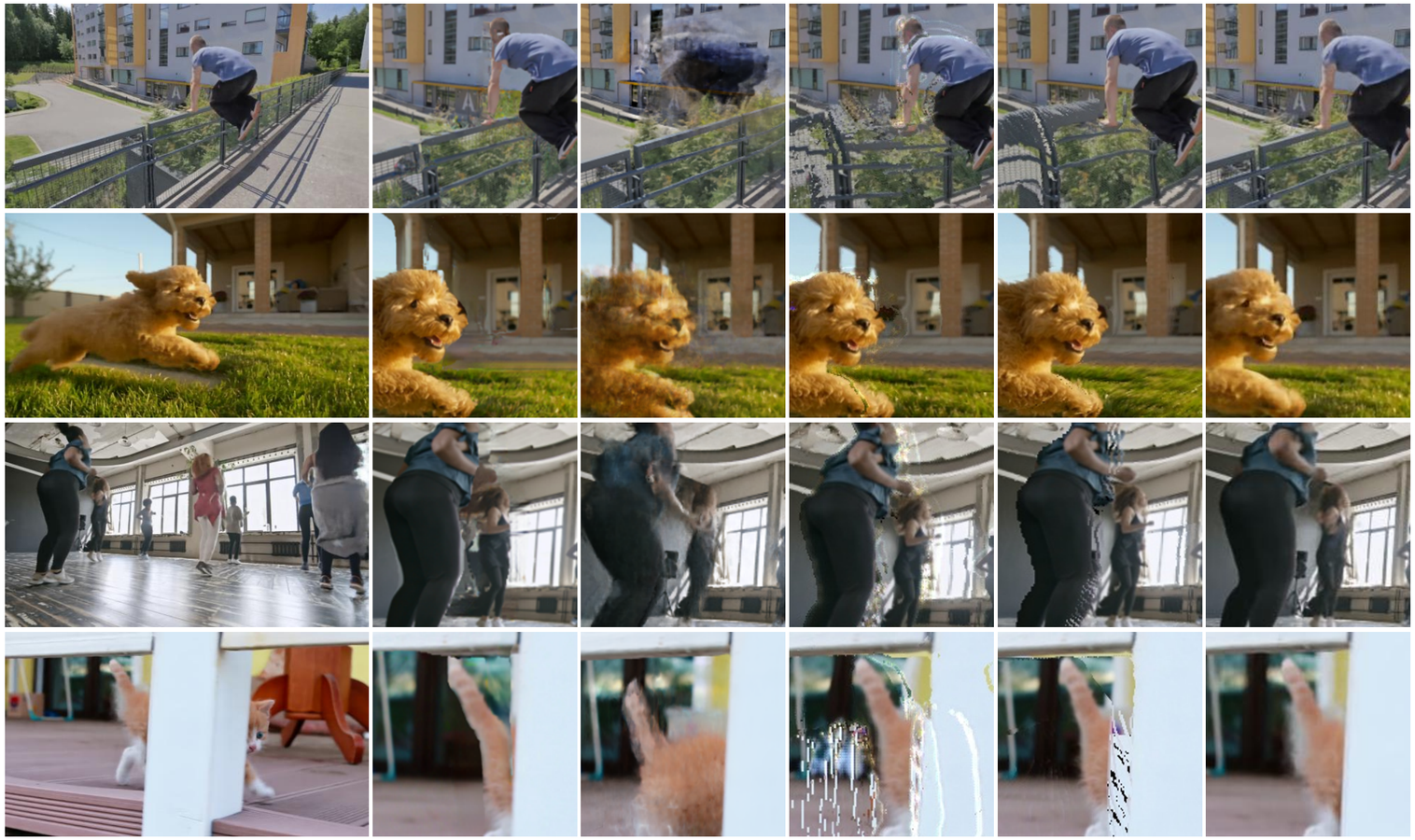
W as ‘confidence’ — used as weight in objectives, encouraged to be 1

Use off-the-shelf depth prediction to inform geometry (rendered depth = predicted depth)

Additionally learn a ‘static’ scene model

Image Courtesy: Neural Scene Flow Fields for Space-Time View Synthesis of Dynamic Scenes. *Li et. al.*

Neural Scene Flow Fields



Our rendered views | 3D Photo [64] | NeRF [46] | Yoon *et al.* [81] | Luo *et al.* [40] | Ours

Figure 9: **Qualitative comparisons on monocular video clips.** When compared to baselines, our approach more correctly synthesizes hidden content in disocclusions (shown in the last three rows), and locations with complex scene structure such as the fence in the first row.

Image Courtesy: Neural Scene Flow Fields for Space-Time View Synthesis of Dynamic Scenes. *Li et. al.*

Neural Scene Flow Fields

Methods	Dynamic Only			Full		
	SSIM (↑)	PSNR (↑)	LPIPS (↓)	SSIM (↑)	PSNR (↑)	LPIPS (↓)
NeRF [46]	0.522	16.74	0.328	0.862	24.29	0.113
[64] + [52]	0.490	16.97	0.216	0.616	19.43	0.217
[81] + [52]	0.498	16.85	0.201	0.748	21.55	0.134
Ours (w/o static)	<u>0.720</u>	<u>21.51</u>	<u>0.149</u>	<u>0.875</u>	<u>26.35</u>	<u>0.090</u>
Ours (w/ static)	**0.724**	**21.58**	**0.143**	**0.892**	**27.38**	**0.066**

Table 2: **Quantitative evaluation of novel view and time synthesis.** See Sec. 4.2 for a description of the baselines.

Significant improvements in modeling dynamic aspects over NeRF

Image Courtesy: Neural Scene Flow Fields for Space-Time View Synthesis of Dynamic Scenes. *Li et. al.*

Neural Scene Flow Fields

Methods	Dynamic Only			Full		
	SSIM (↑)	PSNR (↑)	LPIPS (↓)	SSIM (↑)	PSNR (↑)	LPIPS (↓)
NeRF (w/ time)	0.630	18.89	0.159	0.875	24.33	0.081
w/o $\mathcal{L}_z$	0.710	19.66	0.132	0.882	25.16	0.078
w/o $\mathcal{L}_{\text{geo}}$	0.713	19.74	0.139	0.885	25.19	0.079
w/o $\mathcal{L}_{\text{cyc}}$	0.731	20.52	0.115	0.890	26.15	0.072
w/o $\mathcal{L}_{\text{reg}}$	0.751	21.22	0.110	0.895	26.67	0.074
w/o $\mathcal{W}_i$	0.754	21.31	0.112	0.894	26.23	0.074
w/o static	**0.760**	<u>21.88</u>	<u>0.108</u>	<u>0.906</u>	<u>26.95</u>	<u>0.071</u>
Full (w/ static)	<u>0.758</u>	**21.91**	**0.097**	**0.928**	**28.19**	**0.045**

Table 3: **Ablation study on the Dynamic Scenes dataset.** See Sec. 4.2 for detailed descriptions of each of the ablations.

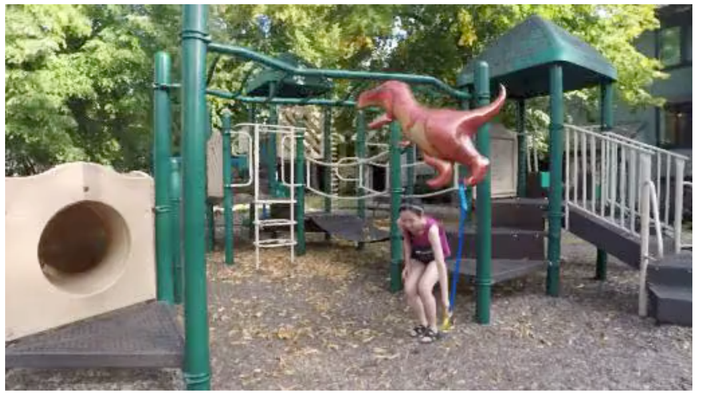
Image Courtesy: Neural Scene Flow Fields for Space-Time View Synthesis of Dynamic Scenes. *Li et. al.*

Neural Scene Flow Fields

Input

Fixed time, changing view

Fixed view, changing time



Spacetime interpolation

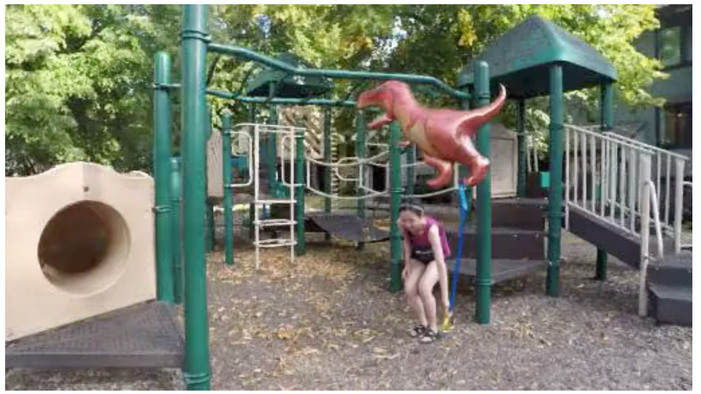
Image Courtesy: Neural Scene Flow Fields for Space-Time View Synthesis of Dynamic Scenes. *Li et. al.*

Neural Scene Flow Fields

Input

Fixed time, changing view

Fixed view, changing time



Spacetime interpolation

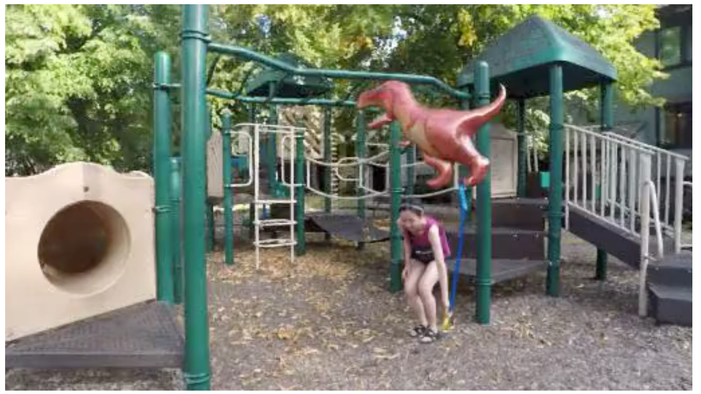
Image Courtesy: Neural Scene Flow Fields for Space-Time View Synthesis of Dynamic Scenes. *Li et. al.*

Neural Scene Flow Fields

Input

Fixed time, changing view

Fixed view, changing time



Spacetime interpolation

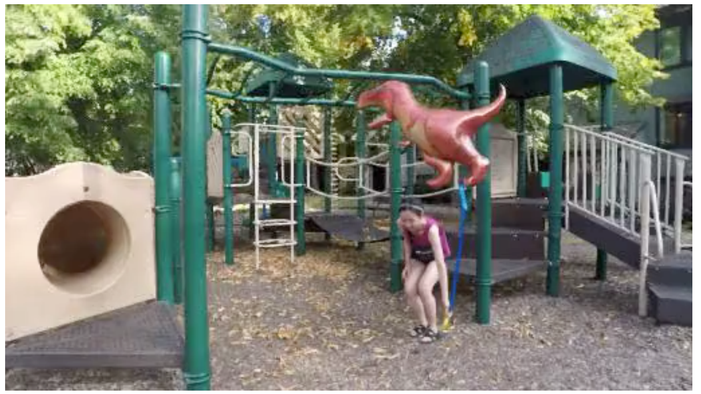
Image Courtesy: Neural Scene Flow Fields for Space-Time View Synthesis of Dynamic Scenes. *Li et. al.*

Neural Scene Flow Fields

Input

Fixed time, changing view

Fixed view, changing time



Spacetime interpolation

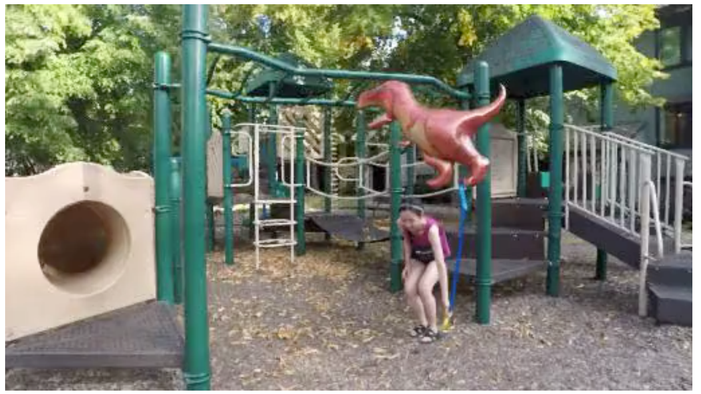
Image Courtesy: Neural Scene Flow Fields for Space-Time View Synthesis of Dynamic Scenes. *Li et. al.*

Neural Scene Flow Fields

Input

Fixed time, changing view

Fixed view, changing time



Spacetime interpolation

Image Courtesy: Neural Scene Flow Fields for Space-Time View Synthesis of Dynamic Scenes. *Li et. al.*

Neural Scene Flow Fields

Input

Fixed time, changing view

Fixed view, changing time

Spacetime interpolation

Image Courtesy: Neural Scene Flow Fields for Space-Time View Synthesis of Dynamic Scenes. *Li et. al.*

Neural Scene Flow Fields

Input

Fixed time, changing view

Fixed view, changing time

Spacetime interpolation

Image Courtesy: Neural Scene Flow Fields for Space-Time View Synthesis of Dynamic Scenes. *Li et. al.*

Neural Scene Flow Fields

Input

Fixed time, changing view

Fixed view, changing time

Spacetime interpolation

Image Courtesy: Neural Scene Flow Fields for Space-Time View Synthesis of Dynamic Scenes. *Li et. al.*

Neural Scene Flow Fields

Input

Fixed time, changing view

Fixed view, changing time

Spacetime interpolation

Image Courtesy: Neural Scene Flow Fields for Space-Time View Synthesis of Dynamic Scenes. *Li et. al.*

Neural Scene Flow Fields

Input

Fixed time, changing view

Fixed view, changing time

Spacetime interpolation

Image Courtesy: Neural Scene Flow Fields for Space-Time View Synthesis of Dynamic Scenes. *Li et. al.*

Neural Scene Flow Fields

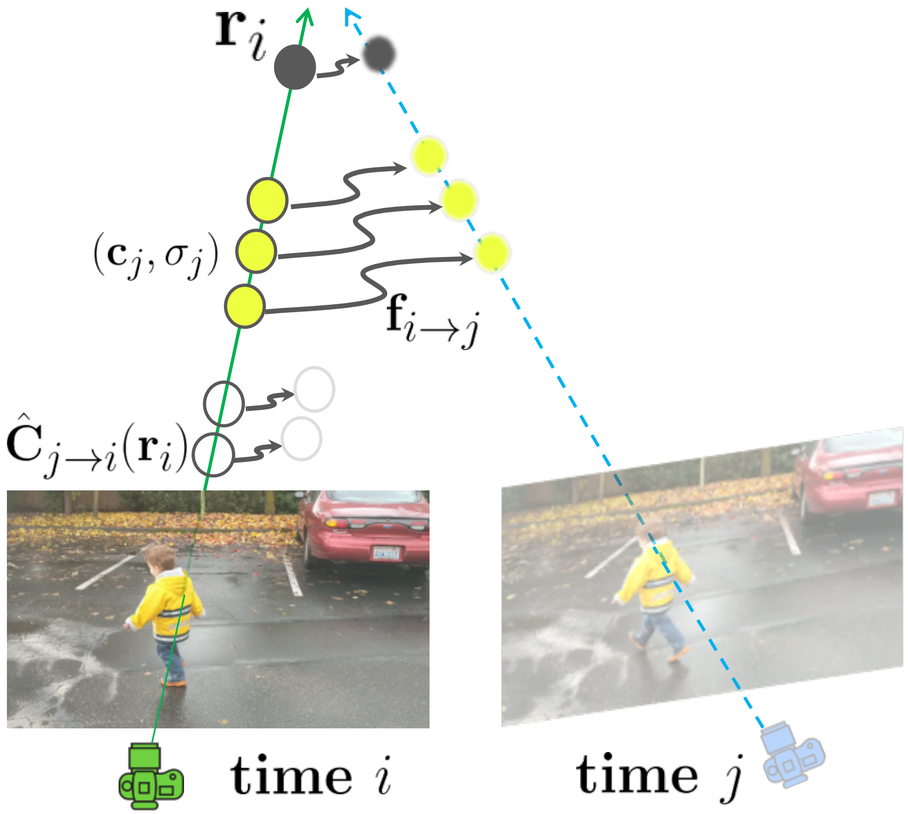
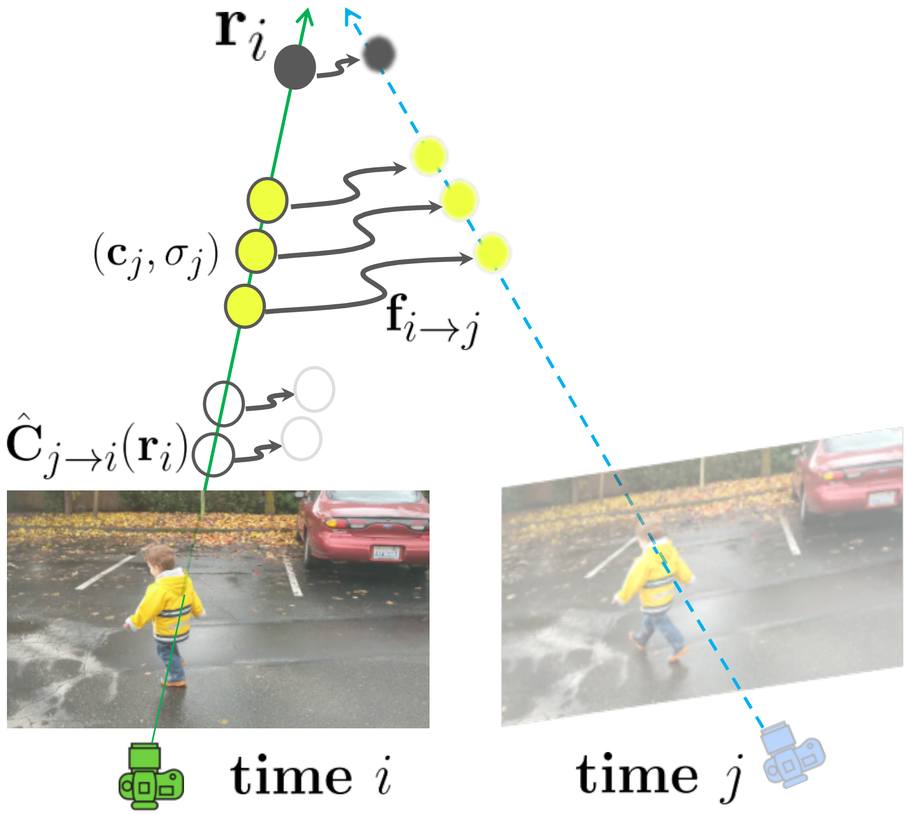


Image Courtesy: Neural Scene Flow Fields for Space-Time View Synthesis of Dynamic Scenes. *Li et. al.*

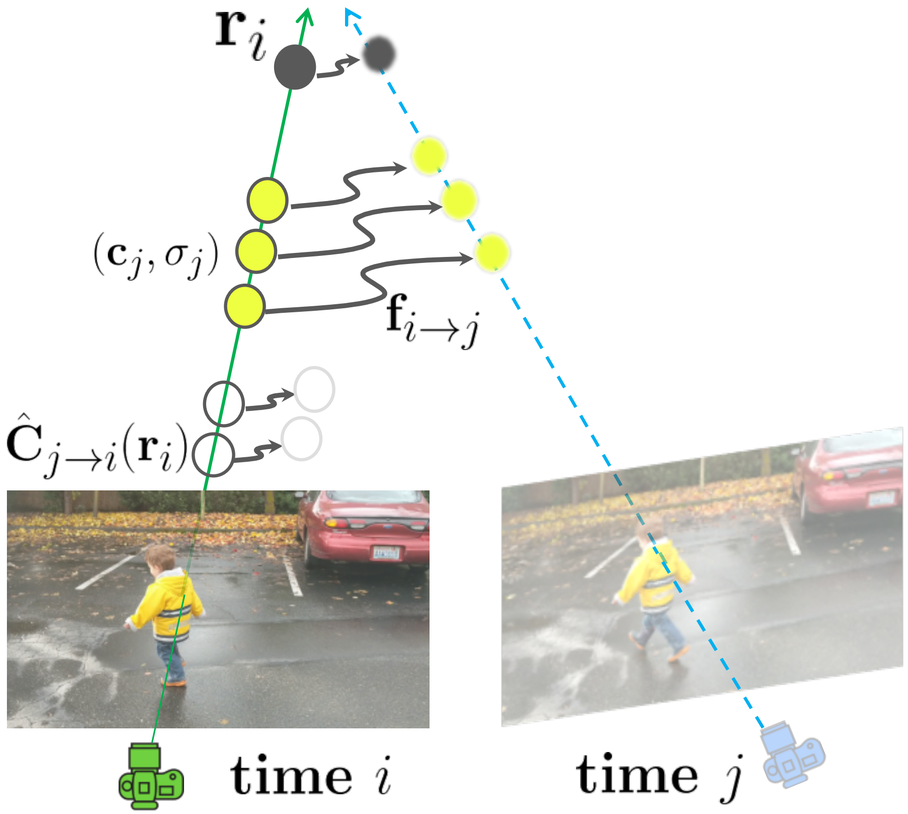
Neural Scene Flow Fields



Only model flow w.r.t neighboring frames

Image Courtesy: Neural Scene Flow Fields for Space-Time View Synthesis of Dynamic Scenes. *Li et. al.*

Neural Scene Flow Fields

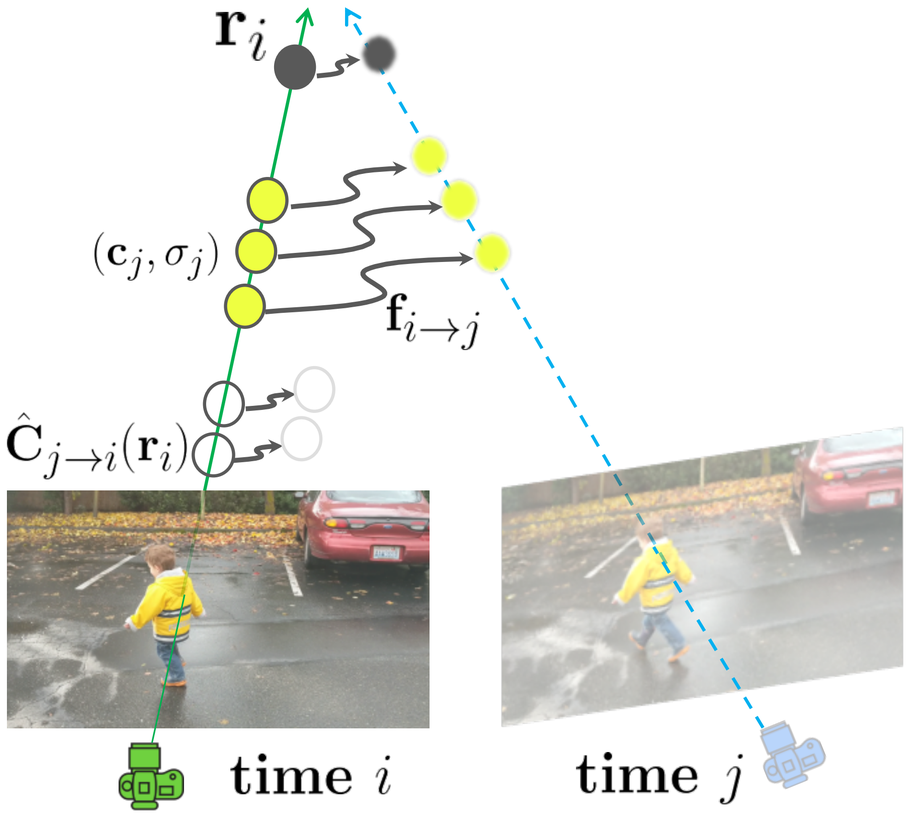


Only model flow w.r.t neighboring frames

No guarantees for persistence over long horizons

Image Courtesy: Neural Scene Flow Fields for Space-Time View Synthesis of Dynamic Scenes. *Li et. al.*

Neural Scene Flow Fields



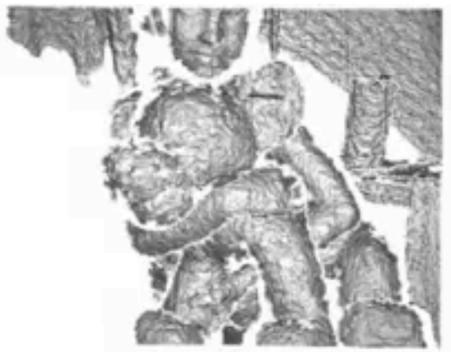
Only model flow w.r.t neighboring frames

No guarantees for persistence over long horizons

Drift: long-range correspondences maybe inaccurate

Image Courtesy: Neural Scene Flow Fields for Space-Time View Synthesis of Dynamic Scenes. *Li et. al.*

Another way to remove degree of freedom: ‘Canonical’ Representations



Live Input Depth Map

Live Model Output

Live RGB Image (unused)

Canonical Model Reconstruction

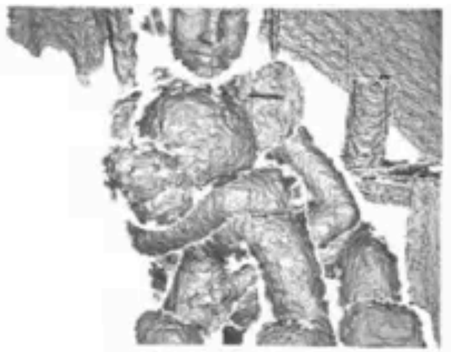
Warped Model

(Relative camera motion removed)

Video credits: DynamicFusion: Reconstruction and Tracking of Non-rigid Scenes in Real-Time.
Newcombe et. al

Key idea: Scene at at time is deformation of a “canonical” scene

Another way to remove degree of freedom: ‘Canonical’ Representations



Live Input Depth Map

Live Model Output

Live RGB Image (unused)

Canonical Model Reconstruction

Warped Model

(Relative camera motion removed)

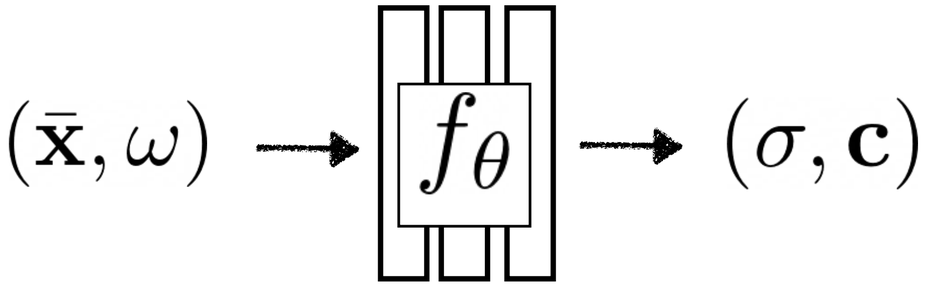
Video credits: DynamicFusion: Reconstruction and Tracking of Non-rigid Scenes in Real-Time.
Newcombe et. al

Key idea: Scene at at time is deformation of a “canonical” scene

Towards Neural Canonical Representations

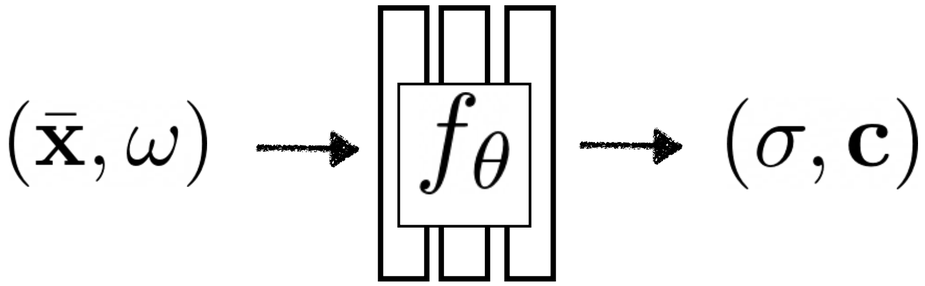
Slide curtesy of Shubham Tulsiani, CMU 16-889: Learning for 3D Vision

Towards Neural Canonical Representations

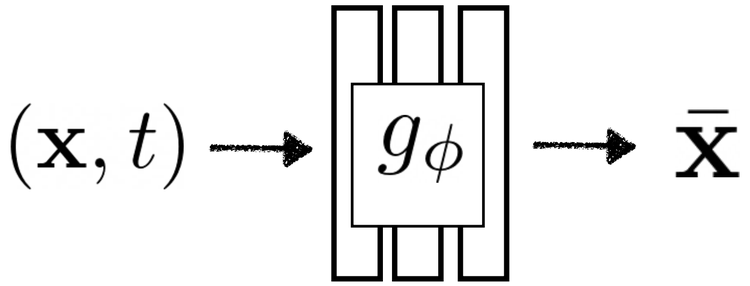


Model a single (canonical)
scene representation

Towards Neural Canonical Representations



Model a single (canonical) scene representation



Predict flow to the canonical scene at each ***t***

Towards Neural Canonical Representations

Non-Rigid Neural Radiance Fields: Reconstruction and Novel View Synthesis of a Dynamic Scene From Monocular Video

Edgar Tretschk[1] Ayush Tewari[1] Vladislav Golyanik[1] Michael Zollhoefer[2] Christoph Lassner[2] Christian Theobalt[1]

[1]Max Planck Institute for Informatics, Saarland Informatics Campus [2]Facebook Reality Labs

(Accepted at ICCV 2021)

Nerfies: Deformable Neural Radiance Fields

Keunhong Park[1], Utkarsh Sinha[2], Jonathan T. Barron[2], Sofien Bouaziz[2], Dan B Goldman[2], Steven M. Seitz[1,2], Ricardo Martin-Brualla[2]

[1]University of Washington, [2]Google Research

D-NeRF: Neural Radiance Fields for Dynamic Scenes

Albert Pumarola, Enric Corona, Gerard Pons-Moll, Francesc Moreno-Noguer

Slide curtesy of Shubham Tulsiani, CMU 16-889: Learning for 3D Vision

Towards Neural Canonical Representations

Non-Rigid Neural Radiance Fields: Reconstruction and Novel View Synthesis of a Dynamic Scene From Monocular Video

Edgar Tretschk[1] Ayush Tewari[1] Vladislav Golyanik[1] Michael Zollhoefer[2] Christoph Lassner[2] Christian Theobalt[1]

[1]Max Planck Institute for Informatics, Saarland Informatics Campus [2]Facebook Reality Labs

(Accepted at ICCV 2021)

Nerfies: Deformable Neural Radiance Fields

Keunhong Park[1], Utkarsh Sinha[2], Jonathan T. Barron[2], Sofien Bouaziz[2], Dan B Goldman[2], Steven M. Seitz[1,2], Ricardo Martin-Brualla[2]

[1]University of Washington, [2]Google Research

D-NeRF: Neural Radiance Fields for Dynamic Scenes

Albert Pumarola, Enric Corona, Gerard Pons-Moll, Francesc Moreno-Noguer

Several concurrent works with instantiations of the same base idea!

Slide curtesy of Shubham Tulsiani, CMU 16-889: Learning for 3D Vision

Neural Canonical Representations

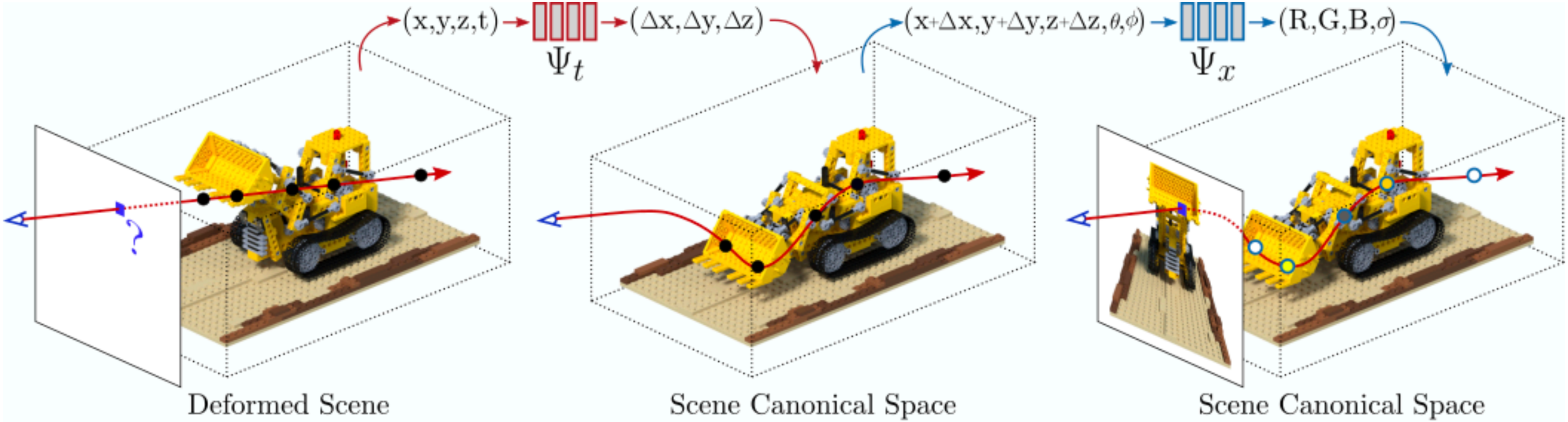
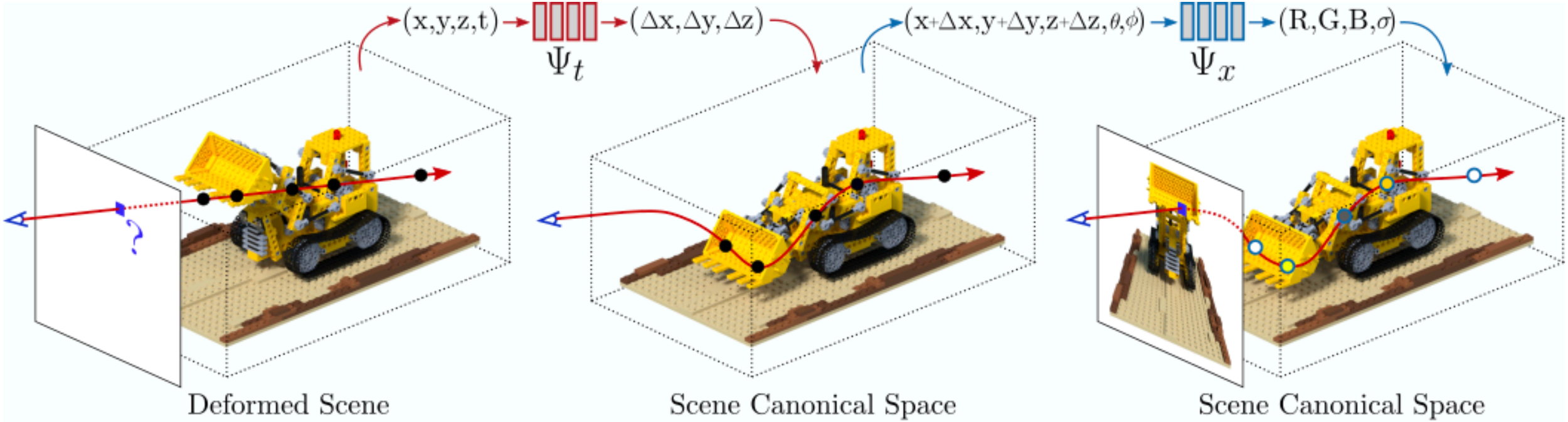


Image Courtesy: D-NeRF: Neural Radiance Fields for Dynamic Scenes. *Pumarola et. al.*

Slide curtesy of Shubham Tulsiani, CMU 16-889: Learning for 3D Vision

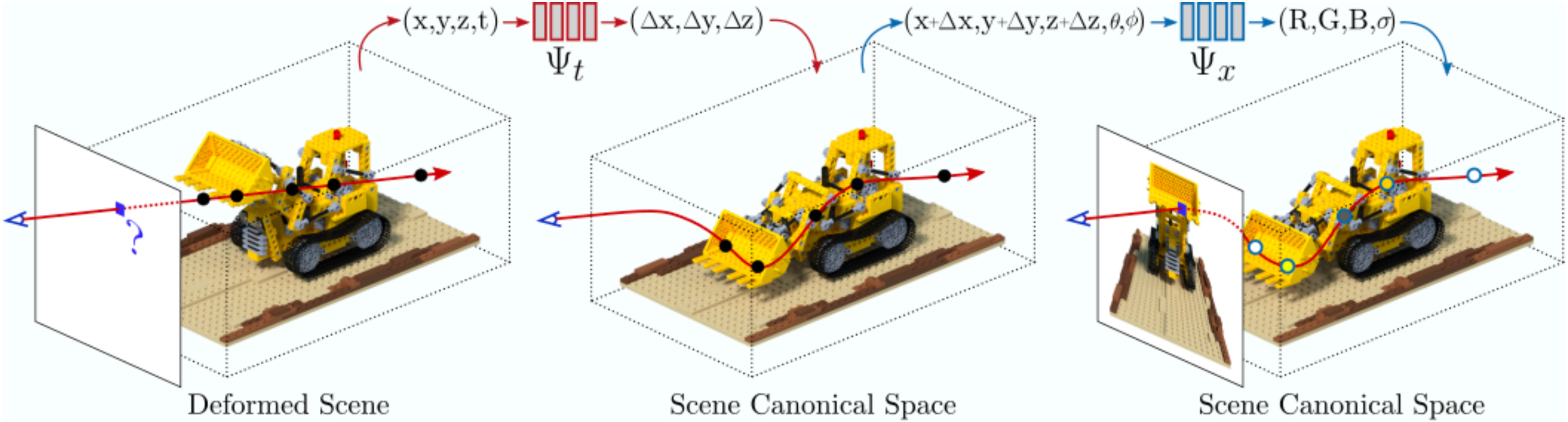
Neural Canonical Representations



Learn two networks — one for flow, one for canonical representation

Image Courtesy: D-NeRF: Neural Radiance Fields for Dynamic Scenes. *Pumarola et. al.*

Neural Canonical Representations

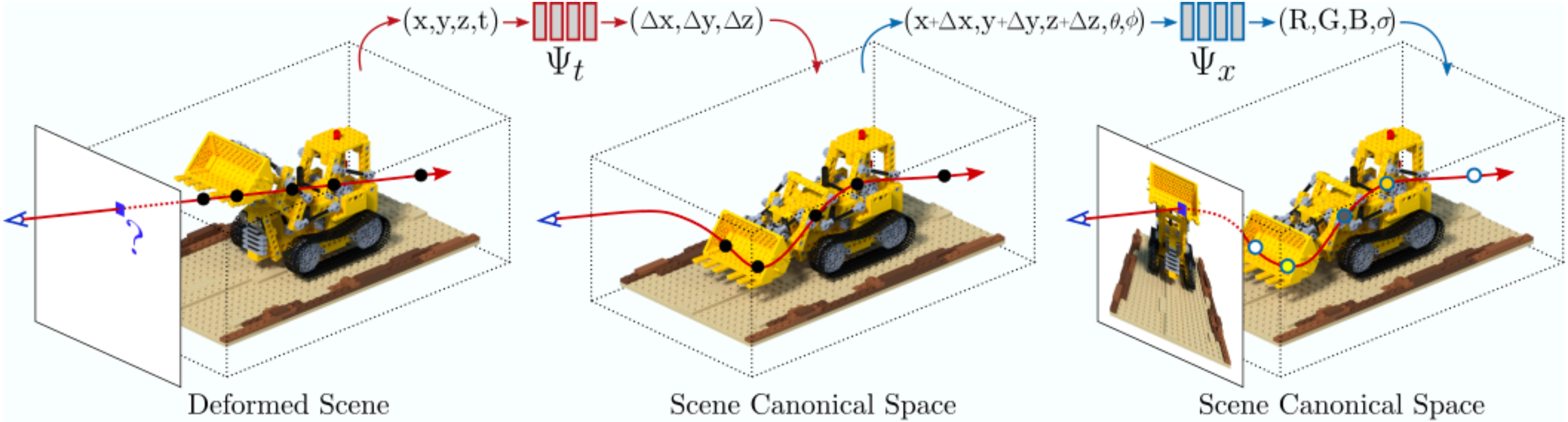


Learn two networks — one for flow, one for canonical representation

Learn both using only a rendering objective!

Image Courtesy: D-NeRF: Neural Radiance Fields for Dynamic Scenes. *Pumarola et. al.*

Neural Canonical Representations



Learn two networks — one for flow, one for canonical representation

Learn both using only a rendering objective!

Minor technical concern: why not transform the ray direction using deformation?

Image Courtesy: D-NeRF: Neural Radiance Fields for Dynamic Scenes. *Pumarola et. al.*

Neural Canonical Representations

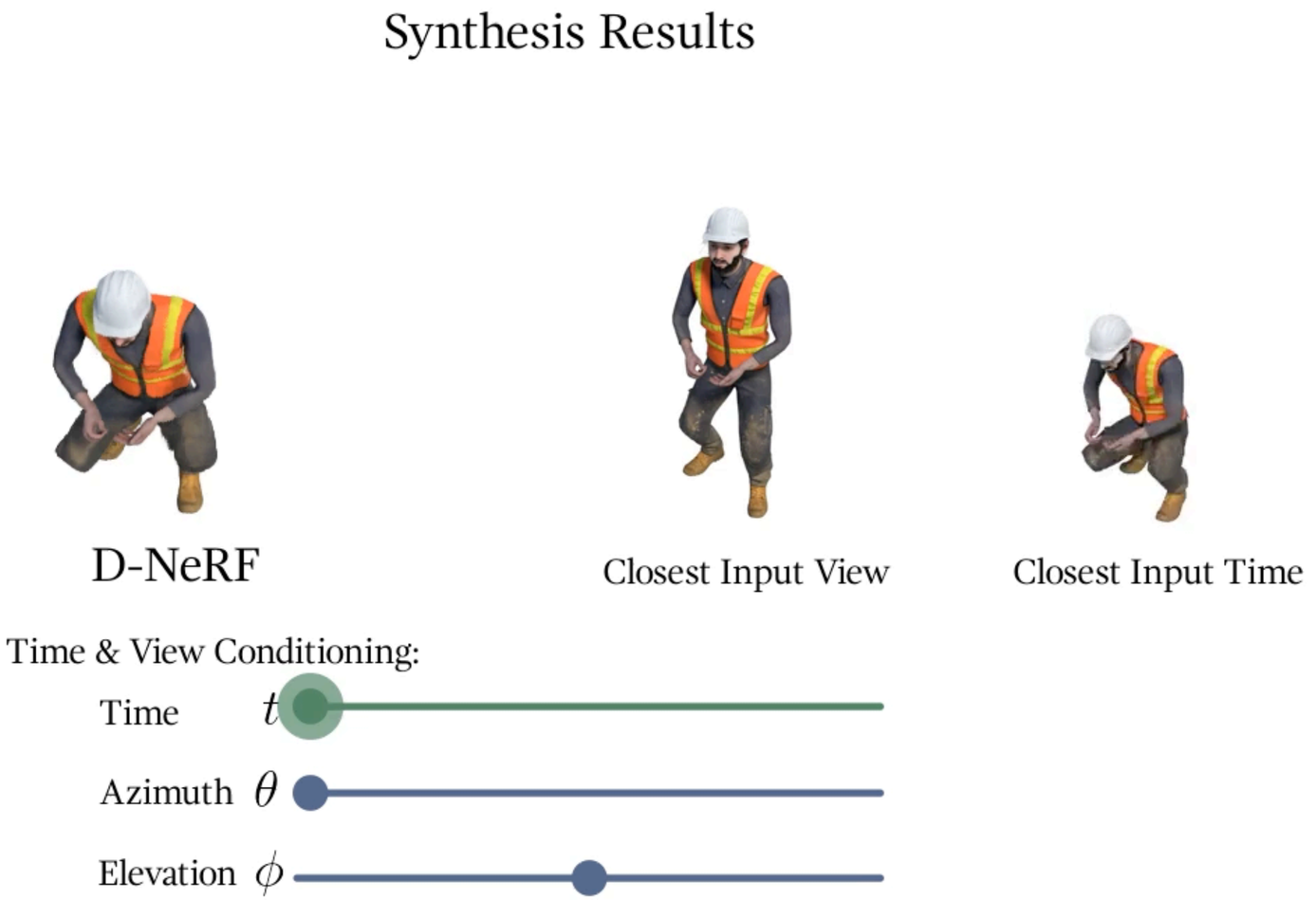


Image Courtesy: D-NeRF: Neural Radiance Fields for Dynamic Scenes. *Pumarola et. al.*

Neural Canonical Representations

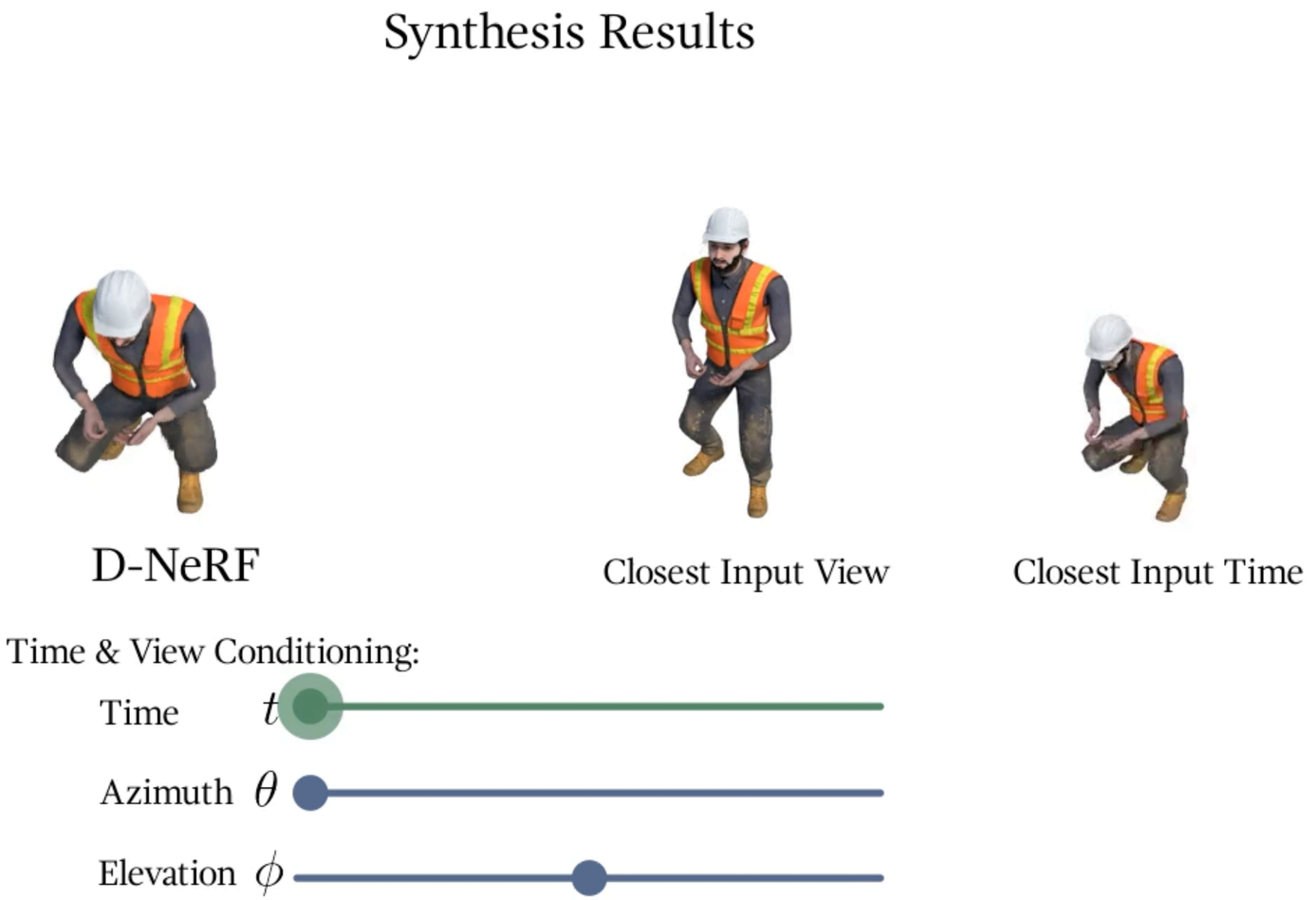
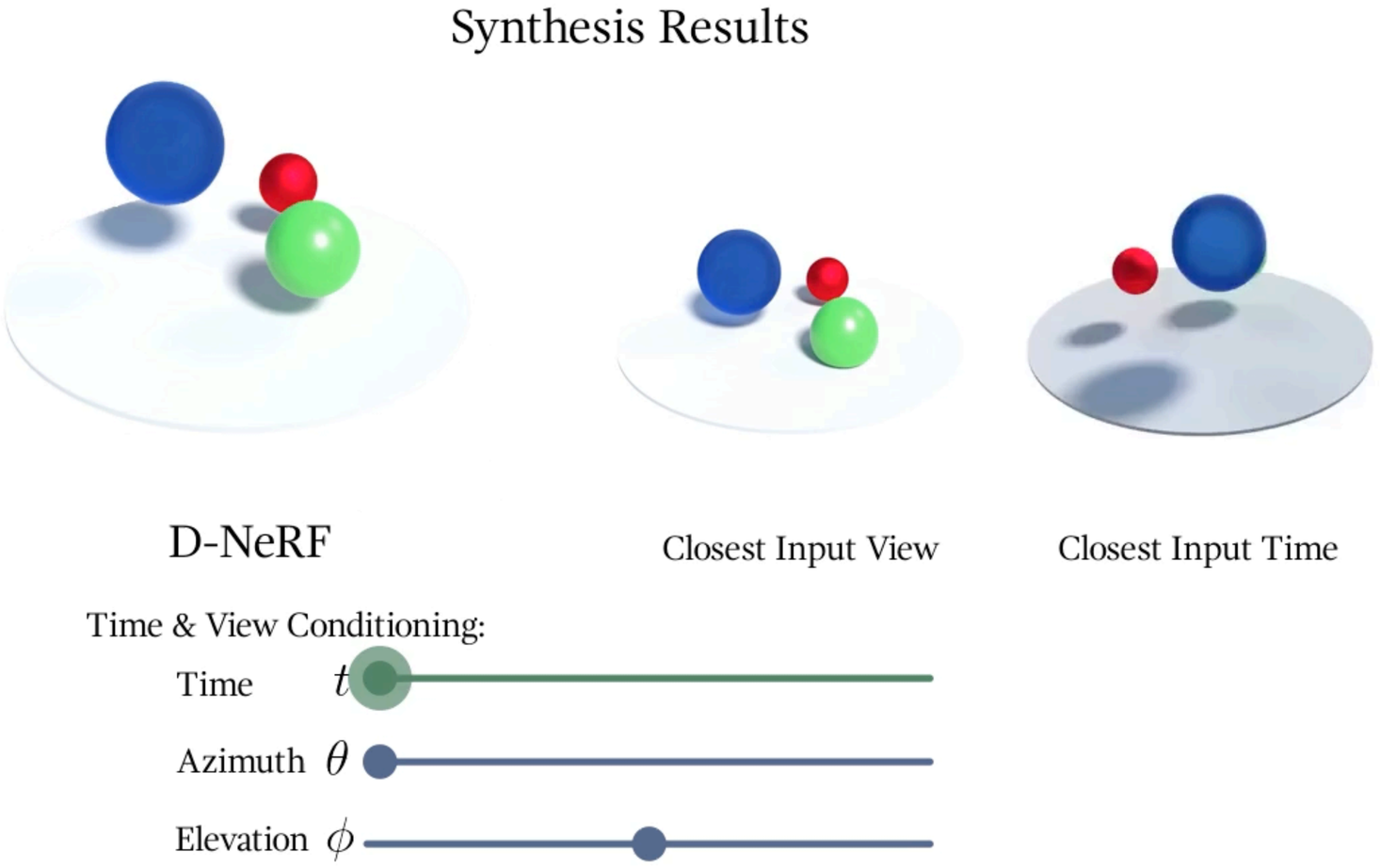


Image Courtesy: D-NeRF: Neural Radiance Fields for Dynamic Scenes. *Pumarola et. al.*

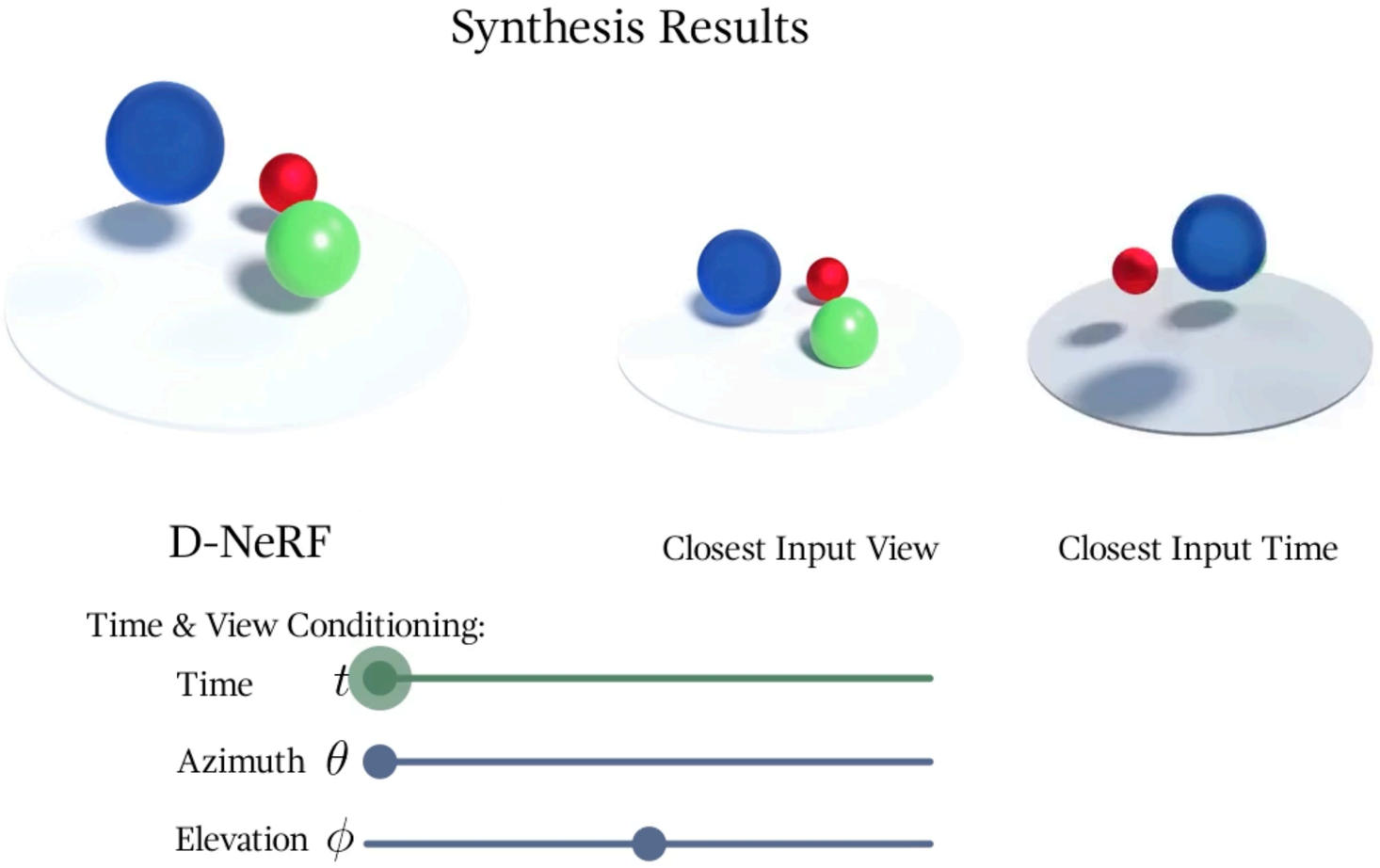
Neural Canonical Representations



Randomly sampled view at each time step (unrealistic as implies a teleporting camera!)

Image Courtesy: D-NeRF: Neural Radiance Fields for Dynamic Scenes. *Pumarola et. al.*

Neural Canonical Representations



Randomly sampled view at each time step (unrealistic as implies a teleporting camera!)

Image Courtesy: D-NeRF: Neural Radiance Fields for Dynamic Scenes. *Pumarola et. al.*

Neural Canonical Representations

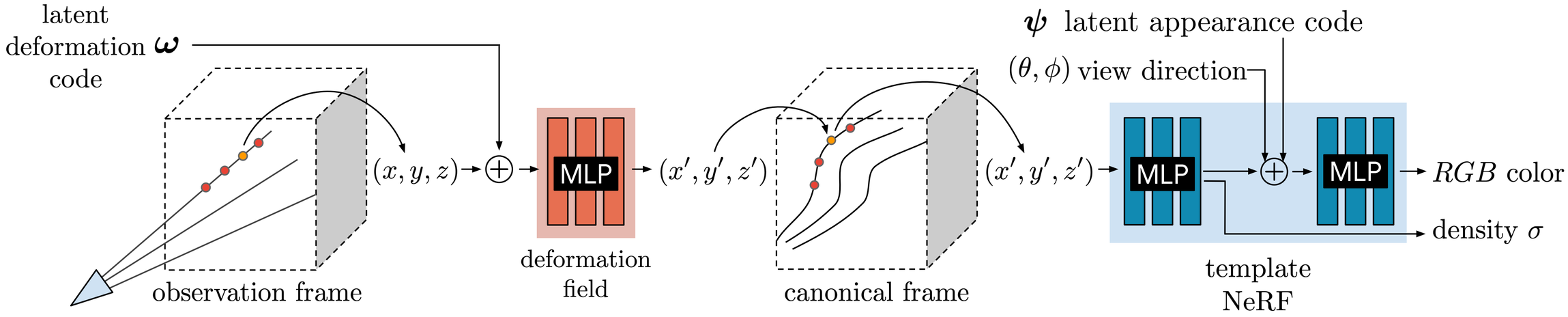
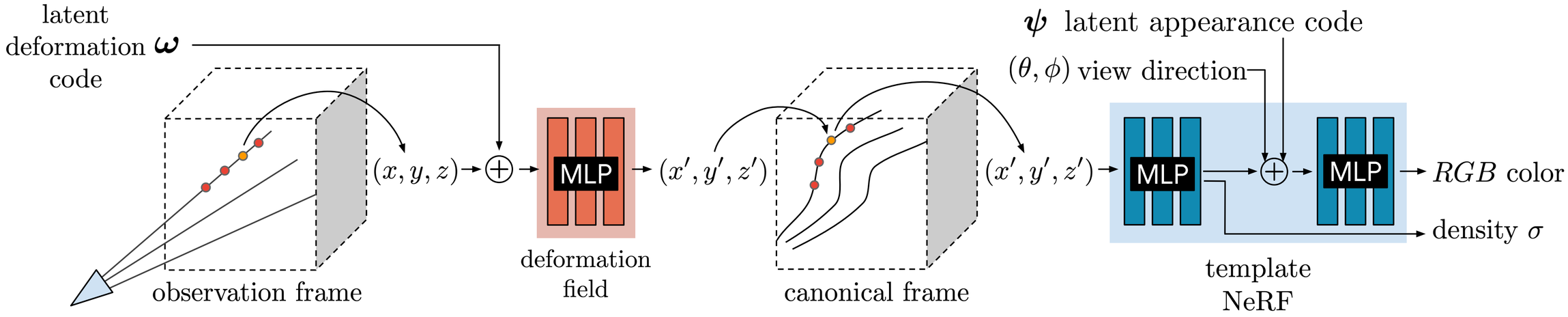


Image Courtesy: Nerfies: Deformable Neural Radiance Fields. *Park et. al.*

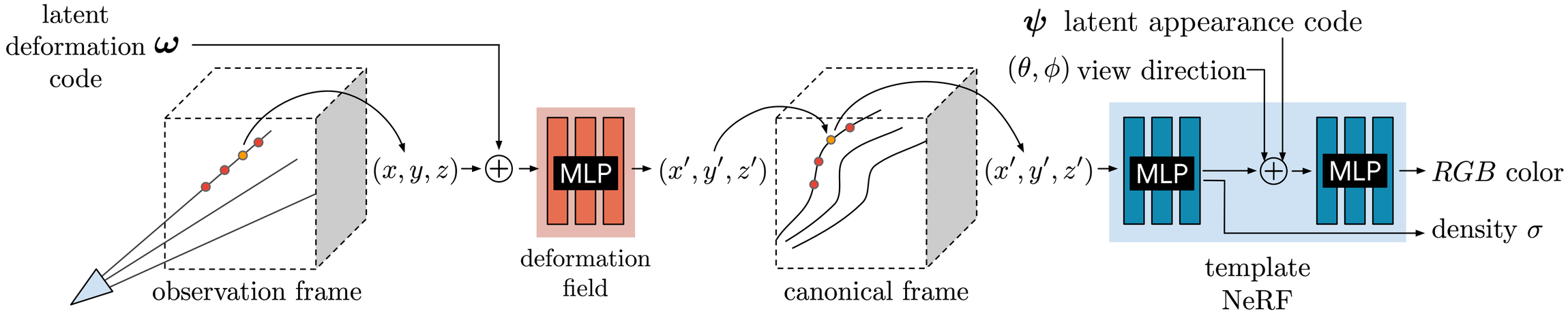
Neural Canonical Representations



$$\text{SE(3) field } W : (\mathbf{x}, \boldsymbol{\omega}_i) \rightarrow \text{SE(3)}$$

Image Courtesy: Nerfies: Deformable Neural Radiance Fields. *Park et. al.*

Neural Canonical Representations

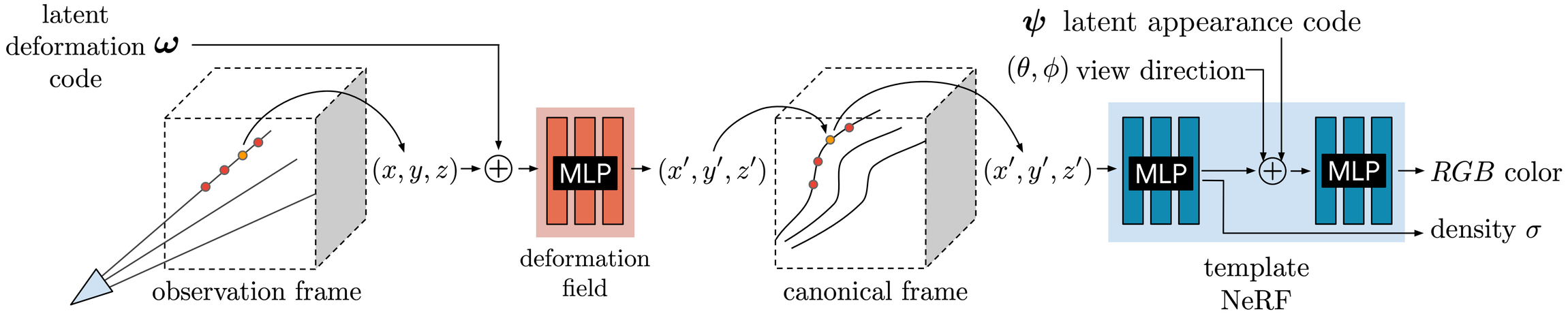


$$\text{SE(3) field } W : (\mathbf{x}, \boldsymbol{\omega}_i) \rightarrow \text{SE(3)}$$

Optimizable latent code as input for deformation prediction (instead of ***t***)

Image Courtesy: Nerfies: Deformable Neural Radiance Fields. *Park et. al.*

Neural Canonical Representations



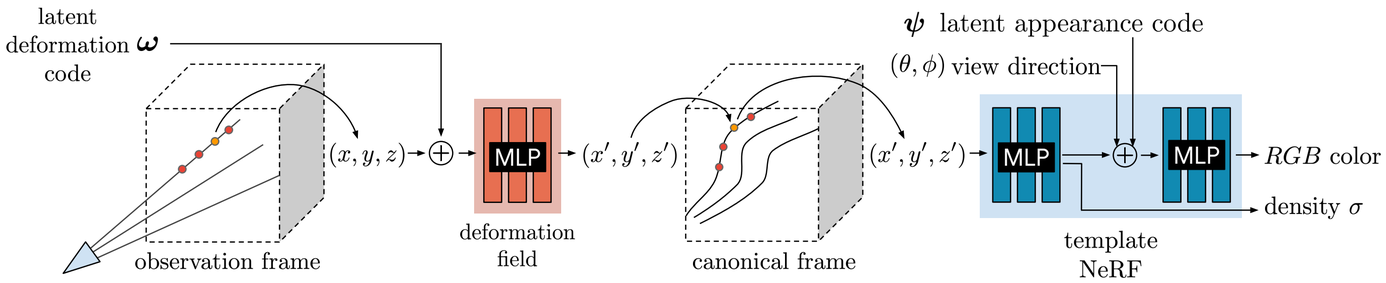
$$\text{SE(3) field } W : (\mathbf{x}, \boldsymbol{\omega}_i) \rightarrow \text{SE(3)}$$

Optimizable latent code as input for deformation prediction (instead of ***t***)

Additionally, an appearance latent code to allow lighting etc. changes (inspired by 'NeRF in the wild')

Image Courtesy: Nerfies: Deformable Neural Radiance Fields. *Park et. al.*

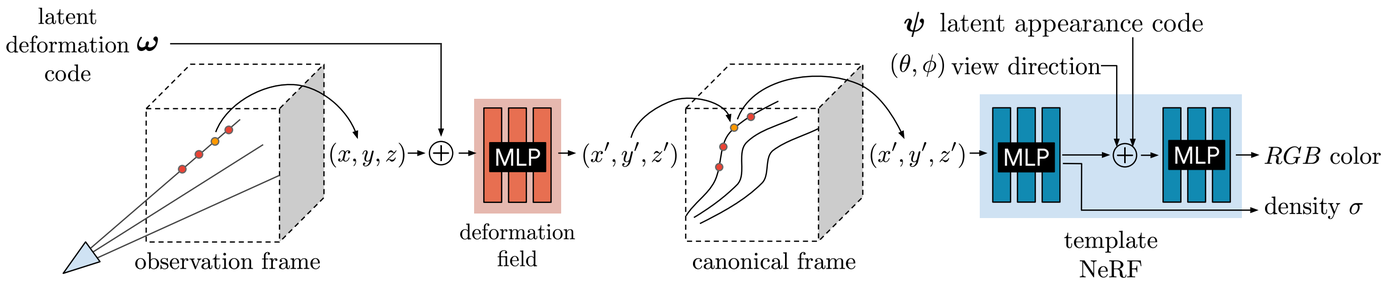
Neural Canonical Representations



Many regularizations in addition to data terms

Image Courtesy: Nerfies: Deformable Neural Radiance Fields. *Park et. al.*

Neural Canonical Representations

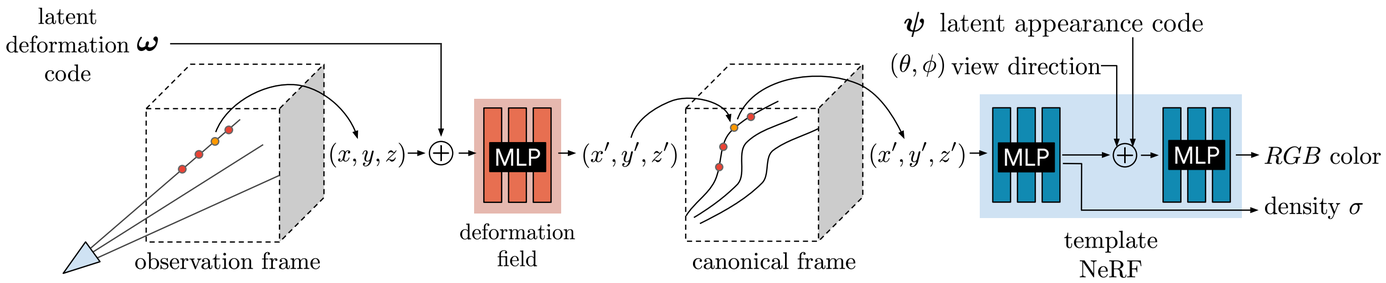


Many regularizations in addition to data terms

$$\mathbf{J}_T(\mathbf{x}) \longrightarrow \mathbf{J}_T = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T \longrightarrow L_{\text{elastic}}(\mathbf{x}) = \|\log \boldsymbol{\Sigma} - \log \mathbf{I}\|_F^2 = \|\log \boldsymbol{\Sigma}\|_F^2$$

Image Courtesy: Nerfies: Deformable Neural Radiance Fields. *Park et. al.*

Neural Canonical Representations



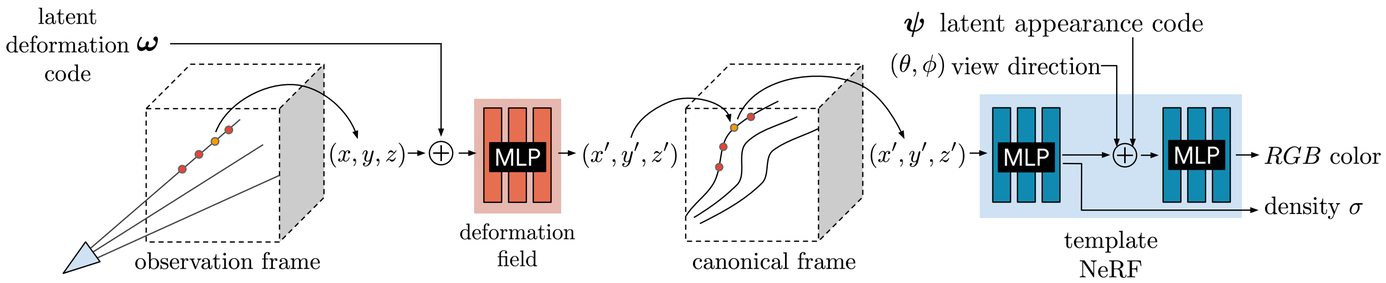
Many regularizations in addition to data terms

$$\mathbf{J}_T(\mathbf{x}) \longrightarrow \mathbf{J}_T = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T \longrightarrow L_{\text{elastic}}(\mathbf{x}) = \|\log \boldsymbol{\Sigma} - \log \mathbf{I}\|_F^2 = \|\log \boldsymbol{\Sigma}\|_F^2$$

Locally rigid transformation (accounting for the spatially varying SE3 field)

Image Courtesy: Nerfies: Deformable Neural Radiance Fields. *Park et. al.*

Neural Canonical Representations



Many regularizations in addition to data terms

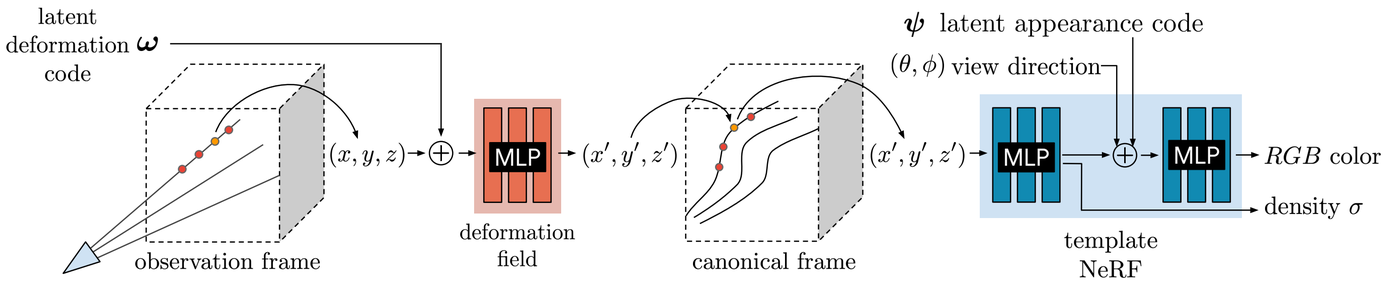
$$\mathbf{J}_T(\mathbf{x}) \longrightarrow \mathbf{J}_T = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T \longrightarrow L_{\text{elastic}}(\mathbf{x}) = \|\log \boldsymbol{\Sigma} - \log \mathbf{I}\|_F^2 = \|\log \boldsymbol{\Sigma}\|_F^2$$

Locally rigid transformation (accounting for the spatially varying SE3 field)

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

Image Courtesy: Nerfies: Deformable Neural Radiance Fields. *Park et. al.*

Neural Canonical Representations



Many regularizations in addition to data terms

$$\mathbf{J}_T(\mathbf{x}) \longrightarrow \mathbf{J}_T = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T \longrightarrow L_{\text{elastic}}(\mathbf{x}) = \|\log \boldsymbol{\Sigma} - \log \mathbf{I}\|_F^2 = \|\log \boldsymbol{\Sigma}\|_F^2$$

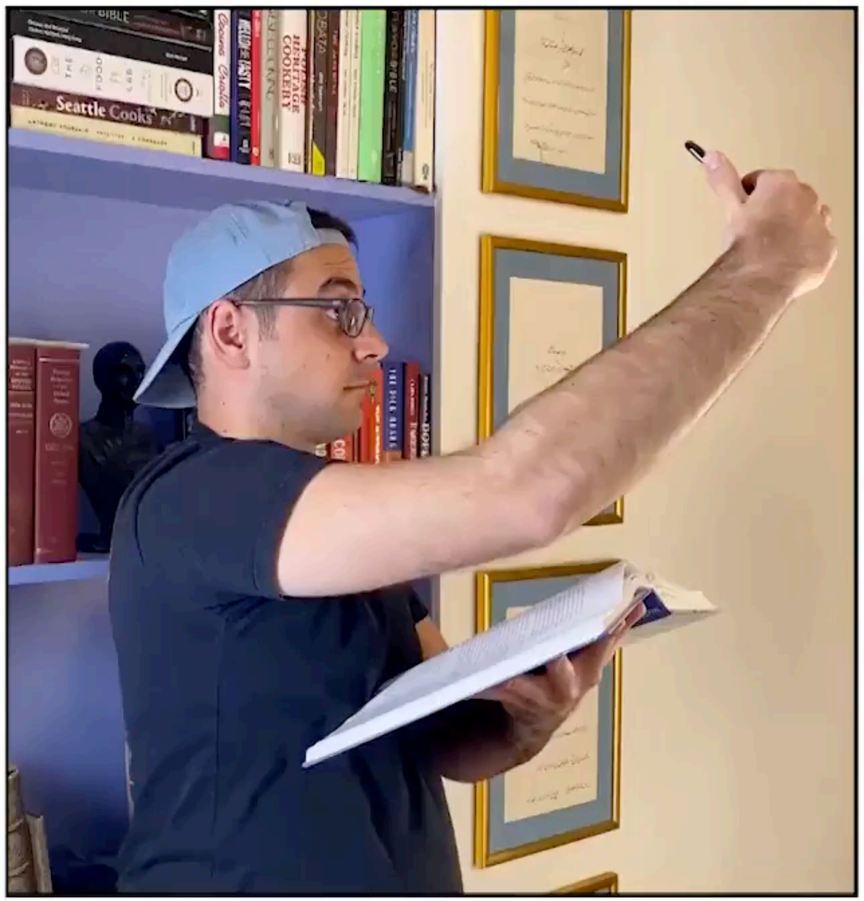
Locally rigid transformation (accounting for the spatially varying SE3 field)

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

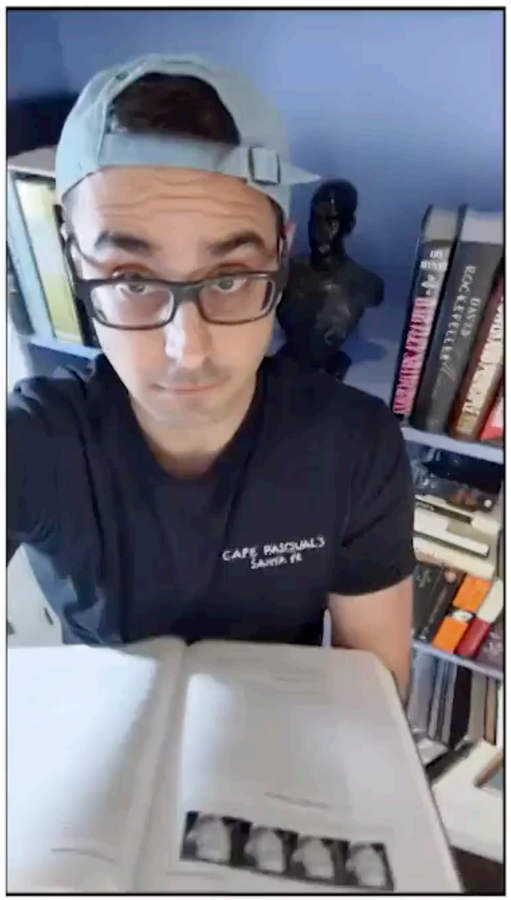
Points in background should be static

Image Courtesy: Nerfies: Deformable Neural Radiance Fields. *Park et. al.*

Neural Canonical Representations

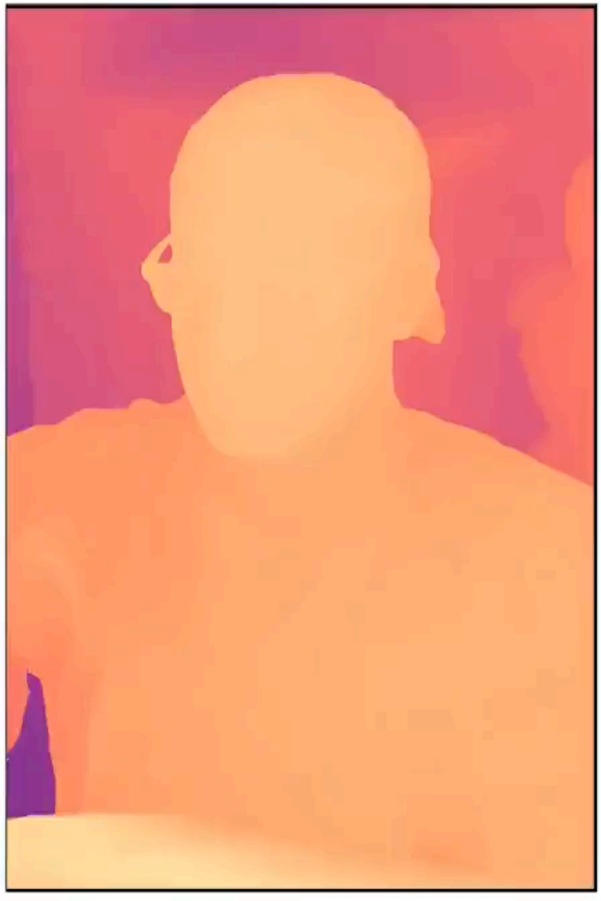


(a) Capture Process



(b) Input

(c) Nerfie

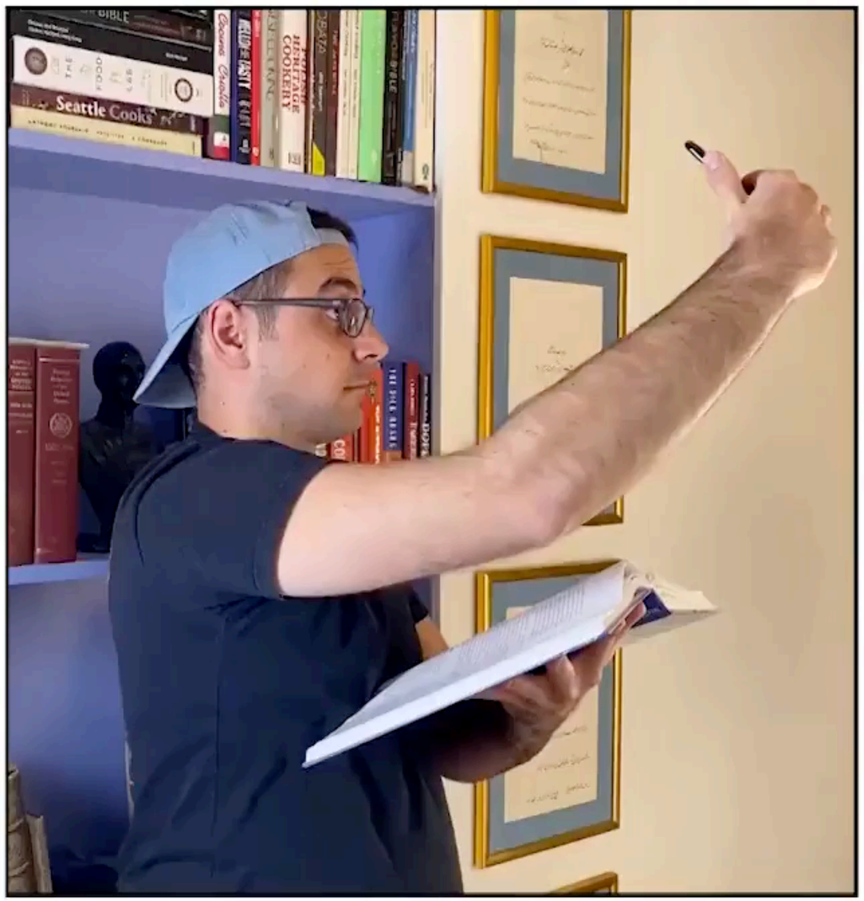


(d) Nerfie Depth

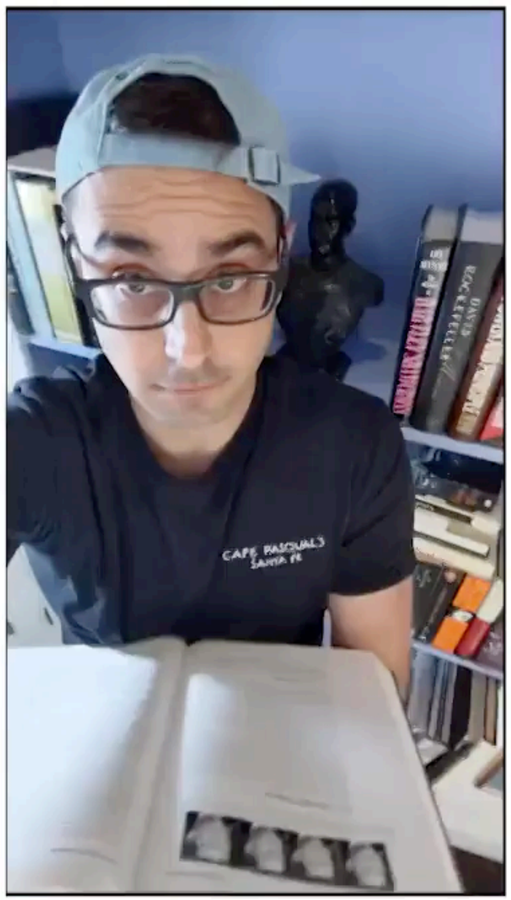
Image Courtesy: Nerfies: Deformable Neural Radiance Fields. *Park et. al.*

Slide curtesy of Shubham Tulsiani, CMU 16-889: Learning for 3D Vision

Neural Canonical Representations

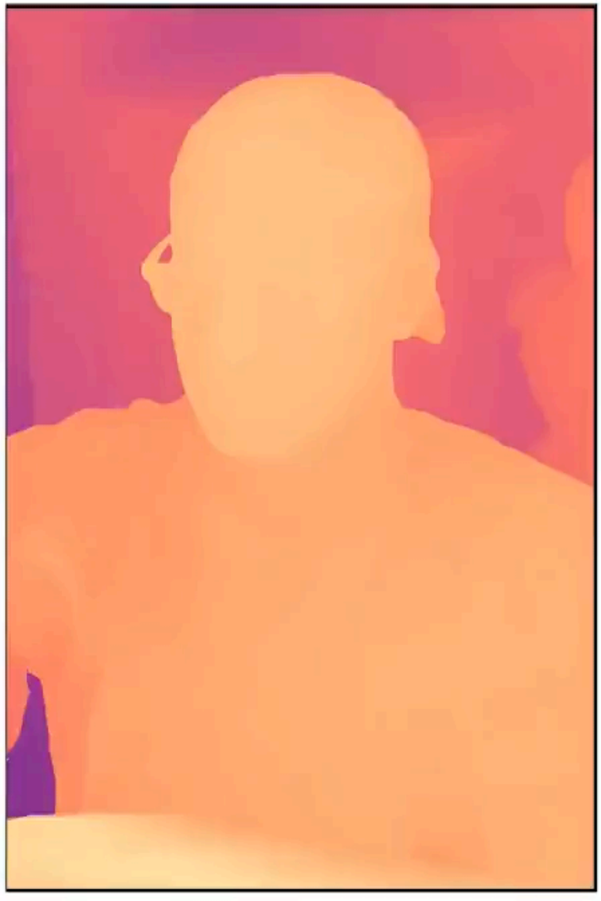


(a) Capture Process



(b) Input

(c) Nerfie



(d) Nerfie Depth

Image Courtesy: Nerfies: Deformable Neural Radiance Fields. *Park et. al.*

Slide curtesy of Shubham Tulsiani, CMU 16-889: Learning for 3D Vision

Neural Canonical Representations

Input Video | Novel View Color | Novel View Depth

Image Courtesy: Nerfies: Deformable Neural Radiance Fields. *Park et. al.*

Slide curtesy of Shubham Tulsiani, CMU 16-889: Learning for 3D Vision

Neural Canonical Representations

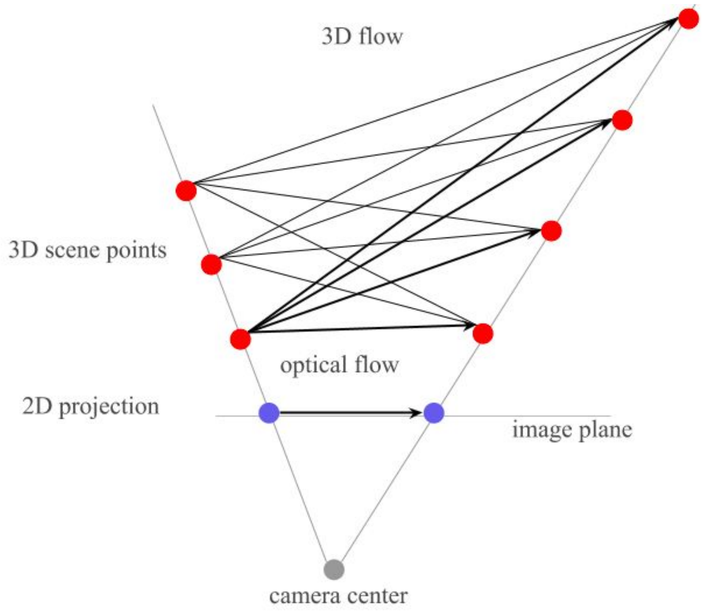
Input Video | Novel View Color | Novel View Depth

Image Courtesy: Nerfies: Deformable Neural Radiance Fields. *Park et. al.*

Slide curtesy of Shubham Tulsiani, CMU 16-889: Learning for 3D Vision

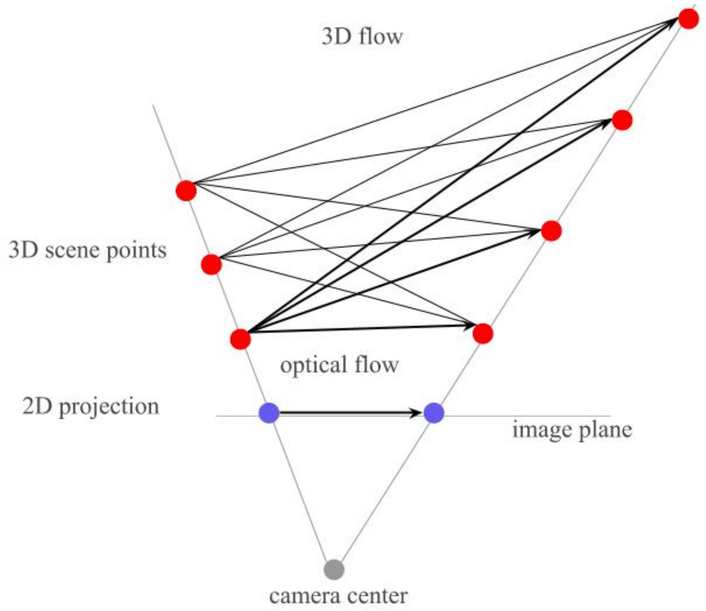
Summary

Summary

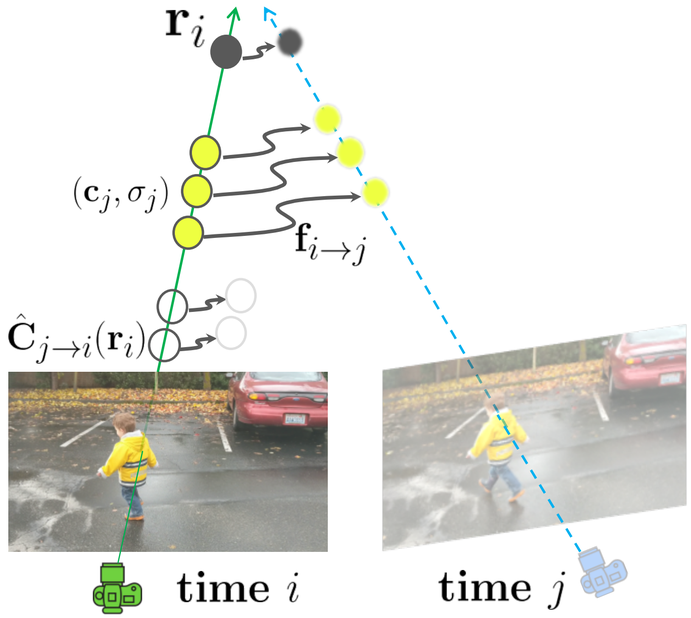


Scene Flow for 3D Correspondence across time

Summary

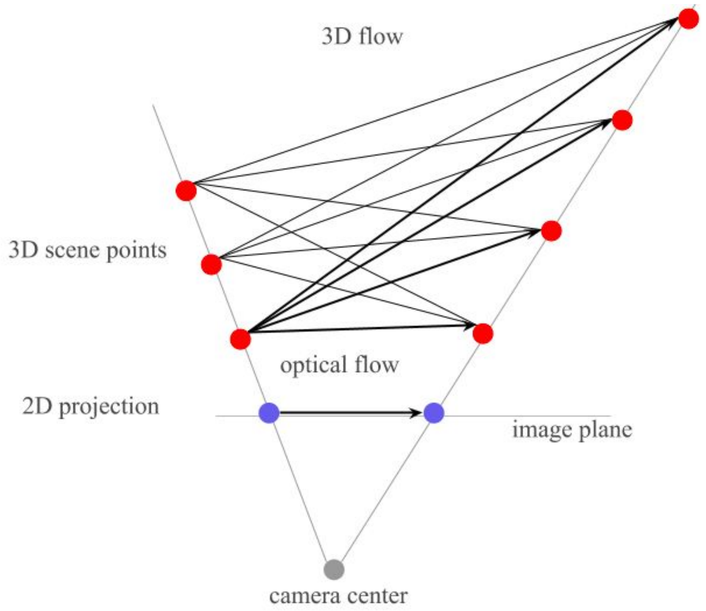


Scene Flow for 3D Correspondence across time

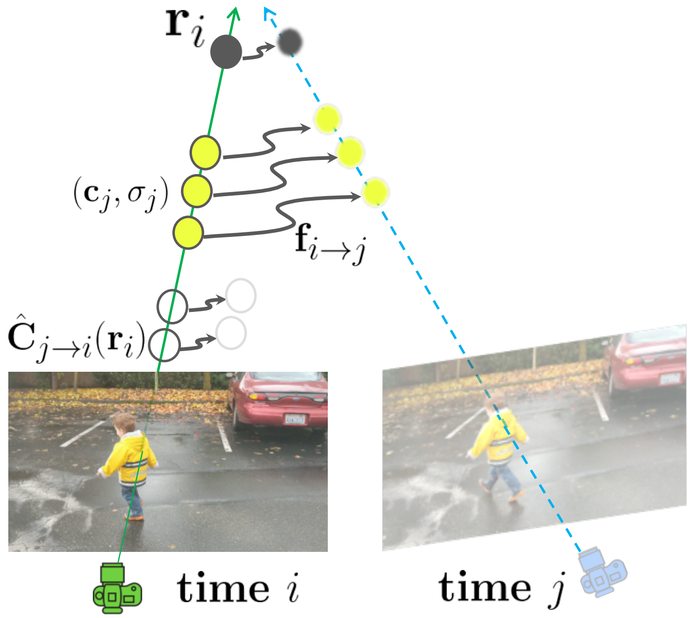


Joint Inference of Scene Flow and per-time Neural Representations

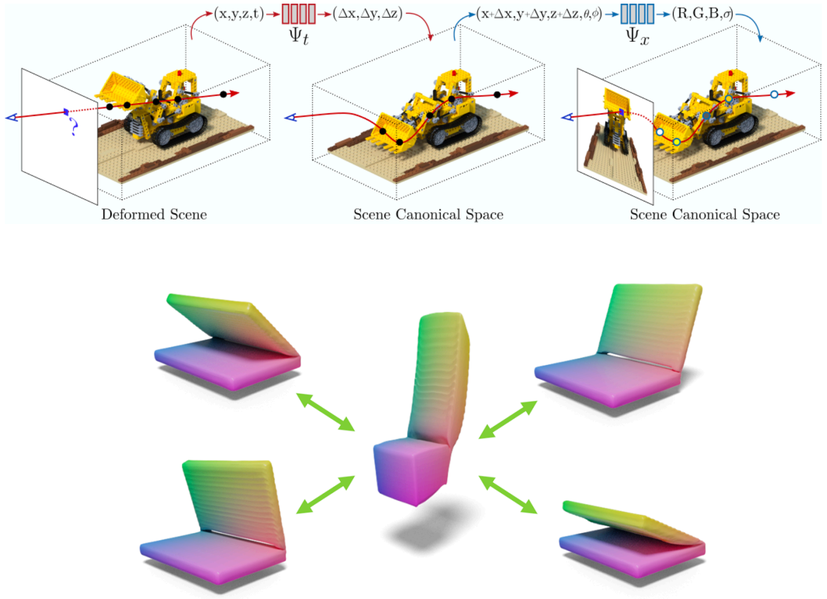
Summary



Scene Flow for 3D Correspondence across time

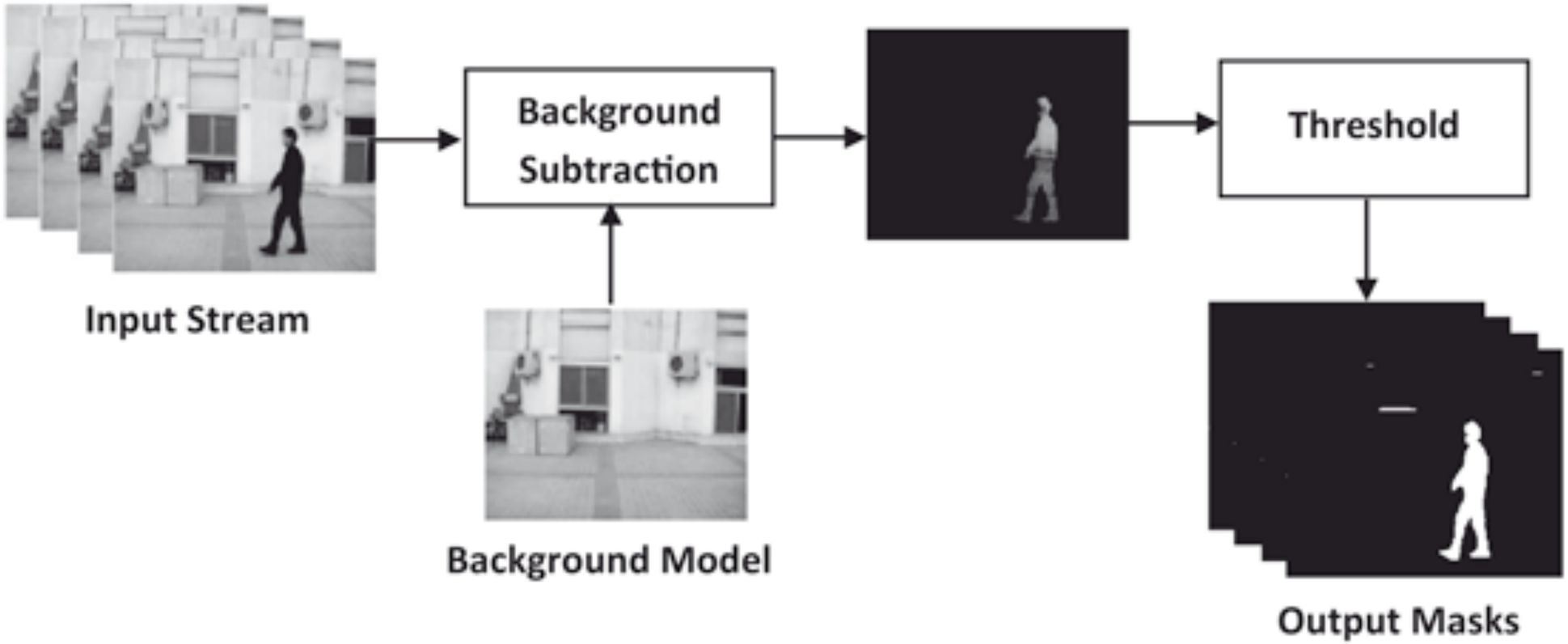


Joint Inference of Scene Flow and per-time Neural Representations



Joint Inference of Scene Flow and per-time Neural Representations

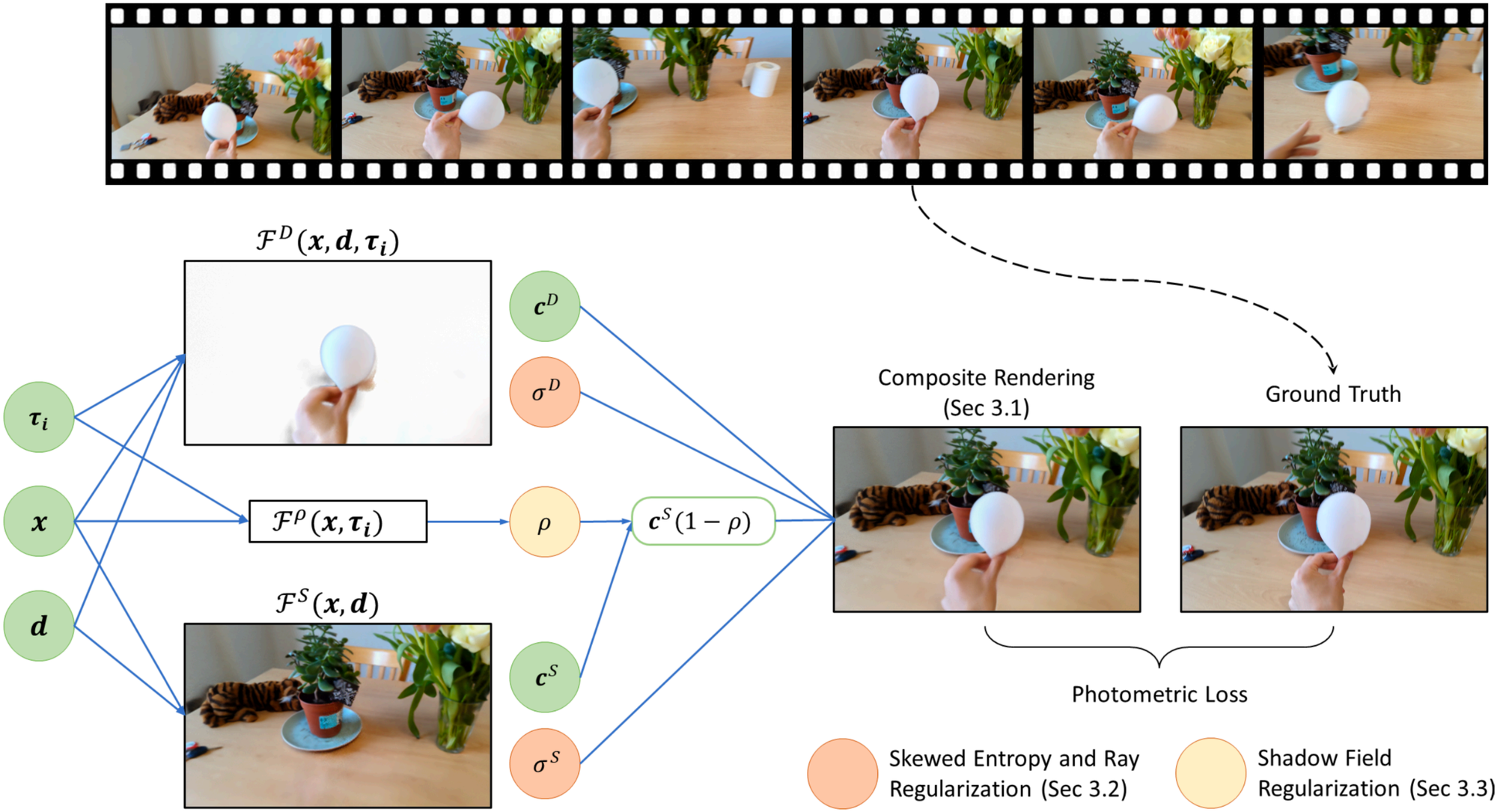
Decoupling Static and Dynamic Scene Parts



Classic CV Concept: Background Subtraction

Image source: https://media.springernature.com/original/springer-static/image/chp%3A10.1007%2F978-3-319-07386-6_3/MediaObjects/321700_1_En_3_Fig1_HTML.gif

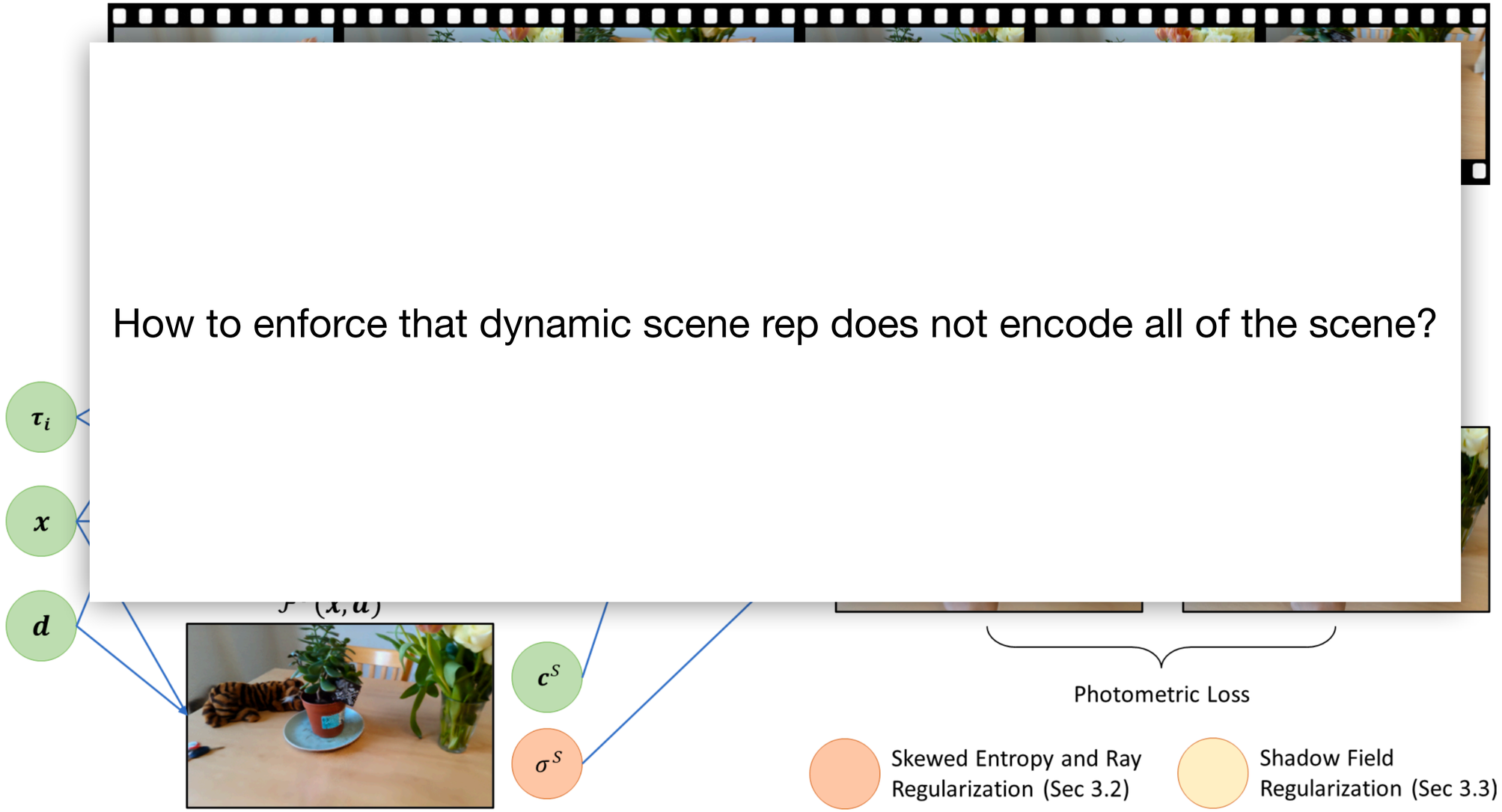
Decoupling Static and Dynamic Scene Parts



D2NeRF: Self-Supervised Decoupling of Dynamic and Static Objects from a Monocular Video, Wu et al., NeurIPS 2022

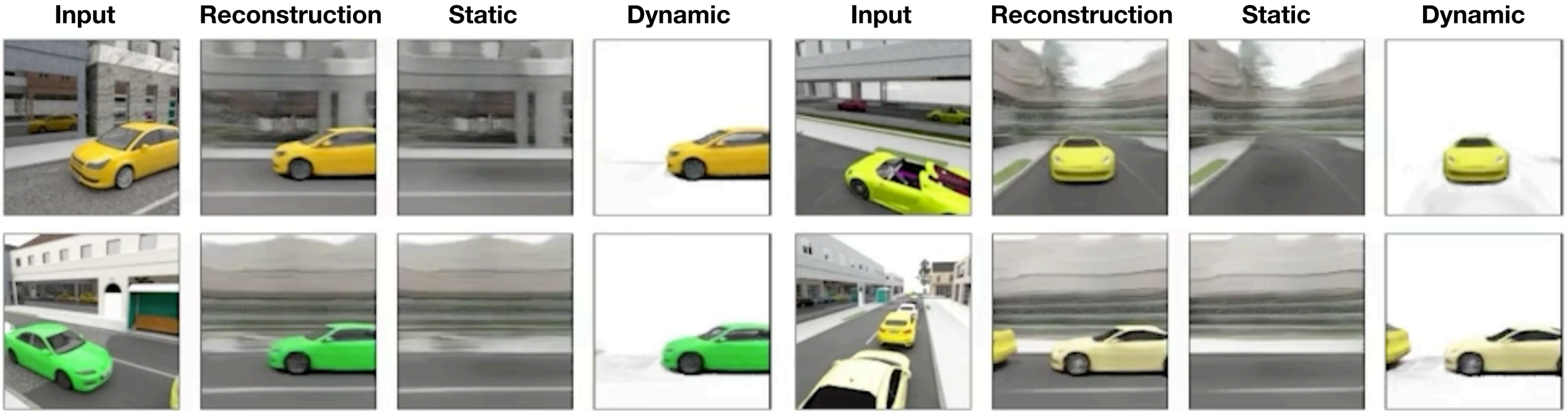
Decoupling Static and Dynamic Scene Parts

How to enforce that dynamic scene rep does not encode all of the scene?



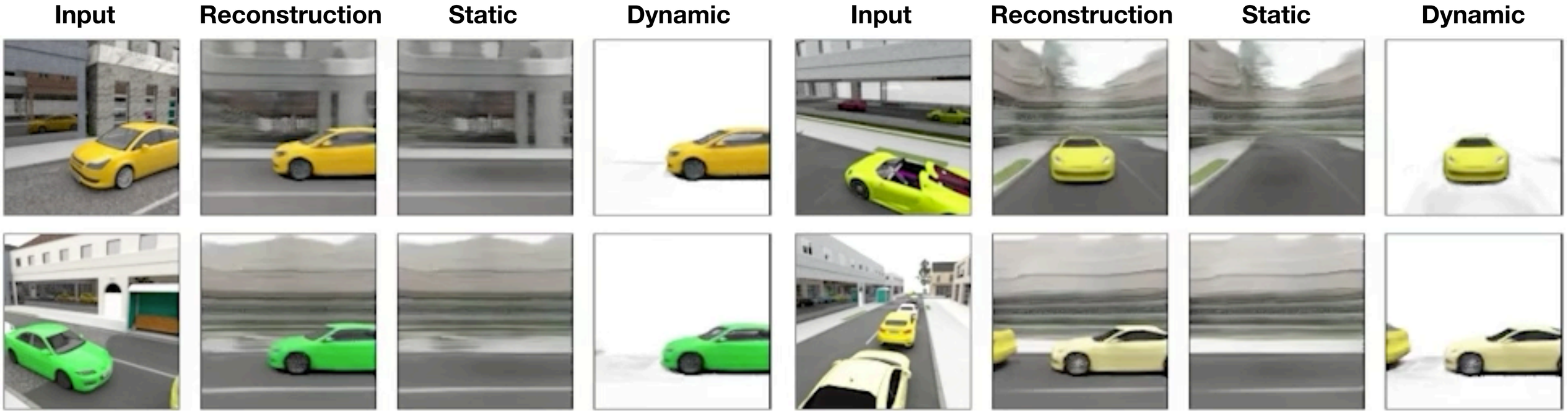
D2NeRF: Self-Supervised Decoupling of Dynamic and Static Objects from a Monocular Video, Wu et al., NeurIPS 2022

Static-dynamic disentanglement from a **single** image!



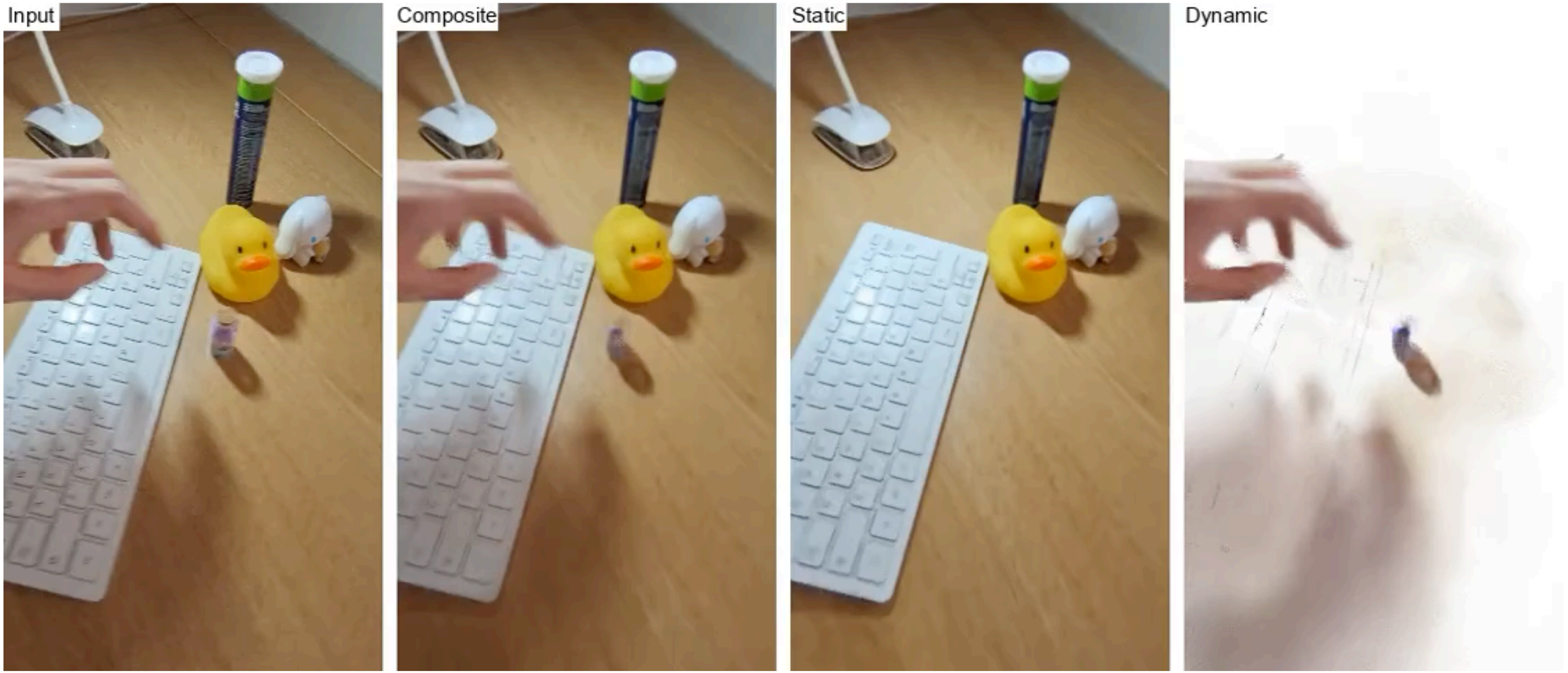
Seeing 3D Objects in a Single Image via Self-Supervised Static-Dynamic Disentanglement, Sharma et al. 2022

Static-dynamic disentanglement from a **single** image!



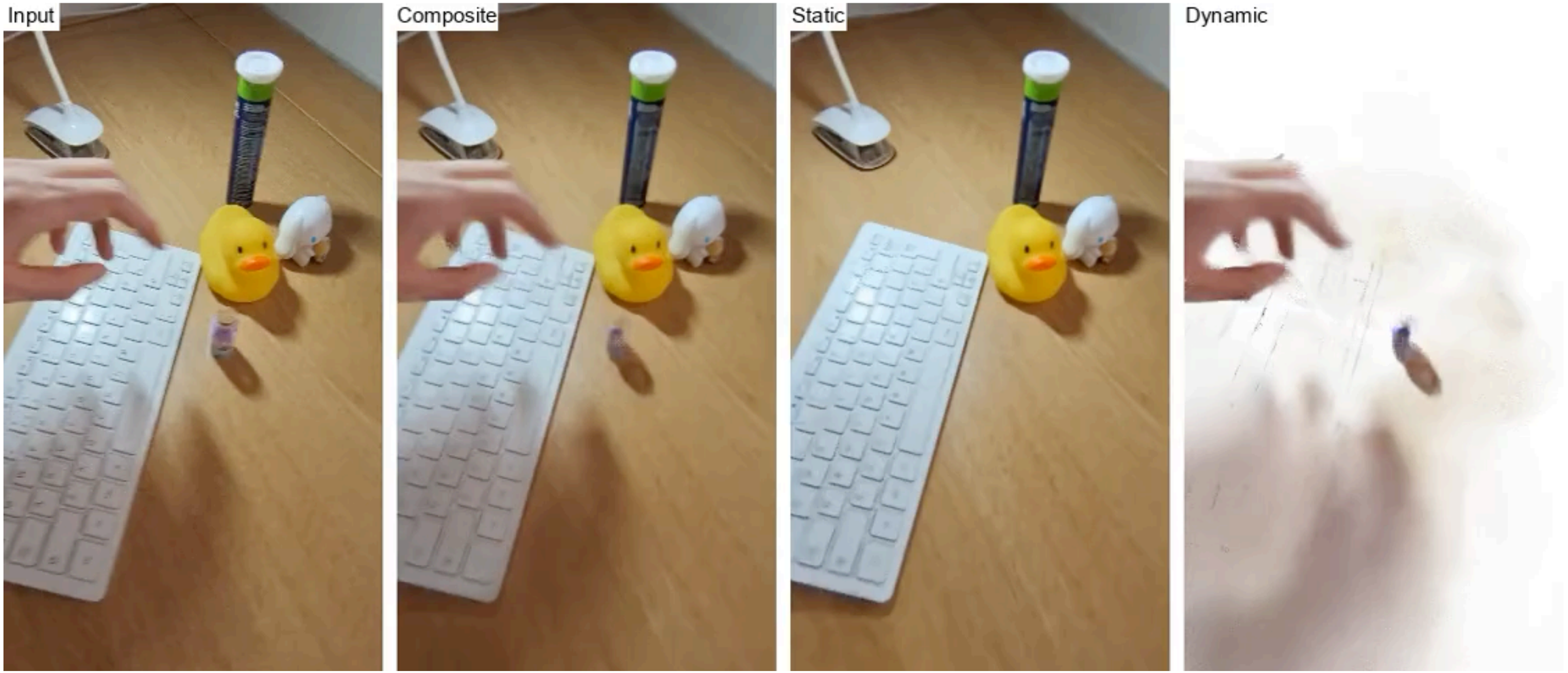
Seeing 3D Objects in a Single Image via Self-Supervised Static-Dynamic Disentanglement, Sharma et al. 2022

Decoupling Static and Dynamic Scene Parts



D2NeRF: Self-Supervised Decoupling of Dynamic and Static Objects from a Monocular Video, Wu et al., NeurIPS 2022

Decoupling Static and Dynamic Scene Parts



D2NeRF: Self-Supervised Decoupling of Dynamic and Static Objects from a Monocular Video, Wu et al., NeurIPS 2022

Summary

- In general, very much an unsolved problem.
- Cannot generate spatio-temporal novel views from monocular video reliably: lots of scene-specific hyperparameter tuning necessary.
- Heavily under-determined: Potentially require more prior-based inference methods.