MonST3R: A Simple Approach for Estimating Geometry in the Presence of Motion

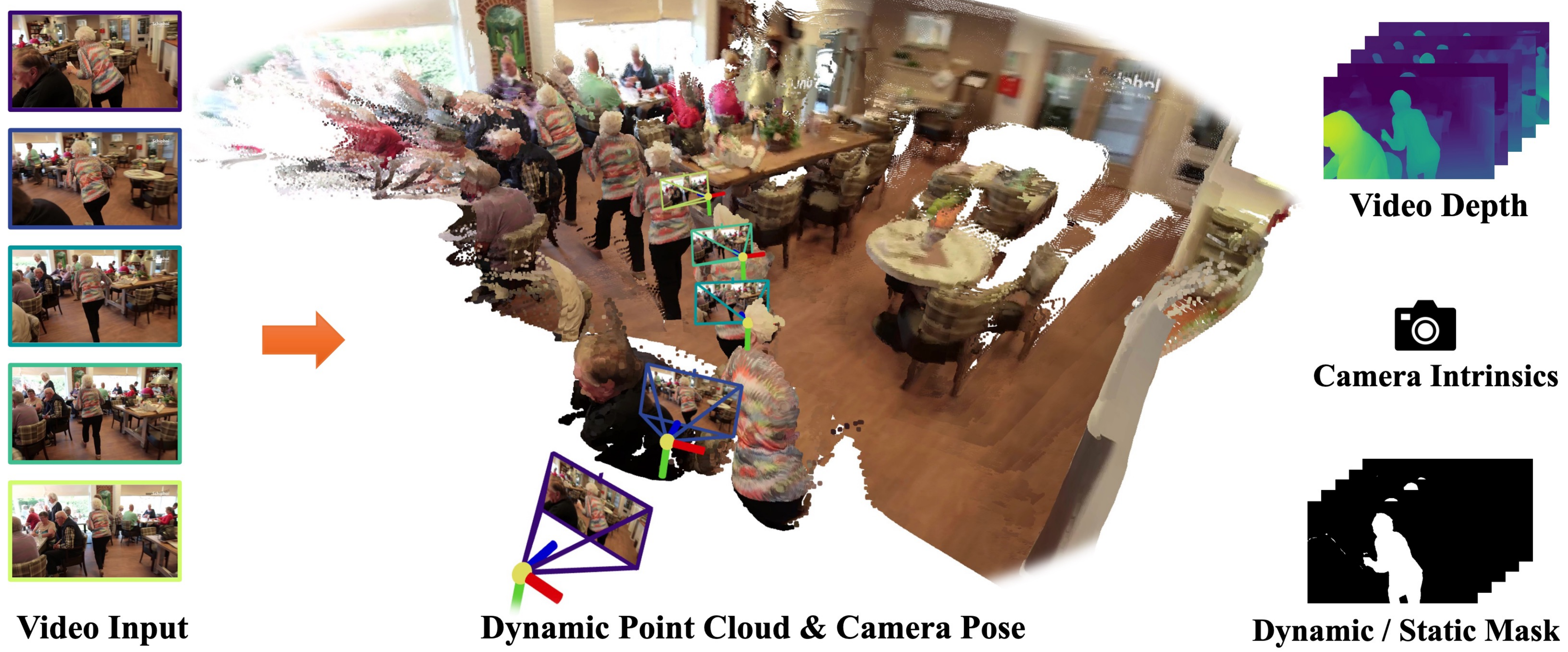
Junyi Zhang [1], Charles Herrmann [2,+], Junhwa Hur [2], Varun Jampani [3], Trevor Darrell [1], Forrester Cole [2], Deqing Sun [2,*], Ming-Hsuan Yang [2,4,*]
[1] UC Berkeley, [2] Google DeepMind, [3] Stability AI, [4] UC Merced

Code & Results

Introduction

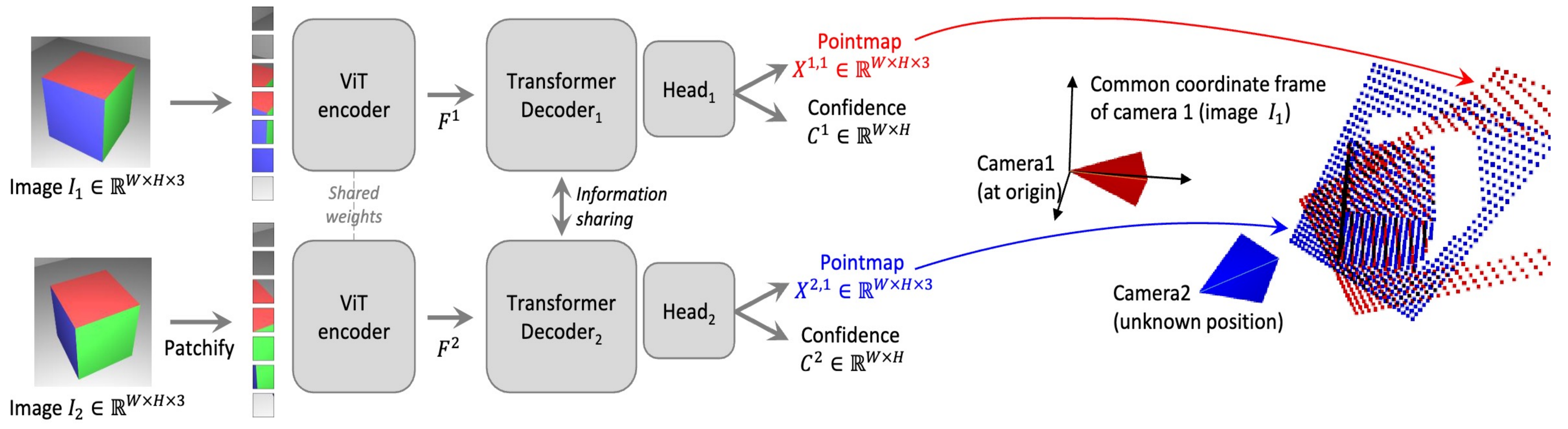
- **Task: estimating global geometry** given a casually-captured **monocular video** of **dynamic scenes,** in a primarily **feed-forward** manner
- **Existing methods** rely on multi-stage pipelines or global optimizations that decompose the problem into subtasks, complex and prone to errors
- **How:** we take a geometry-first approach that directly estimates per-timestep geometry of dynamic scenes
- **Key insight:** by simply estimating a pointmap for each timestep, we adapt DUSt3R's representation, previously used for static scenes, to dynamic scenes.
- **Challenge:** despite the scarcity of training data, we show that by posing the problem as a fine-tuning task, strategically training the model on limited data can surprisingly enable it to handle dynamics

Overview



Given a video of dynamic scene, MonST3R processes it to produce a time-varying **dynamic point cloud**, along with per-frame camera poses and intrinsics, in a predominantly **feed-forward** manner

Pointmap Representation of DUSt3R



Given **two frames**, DUSt3R estimates two corresponding pointmaps (xyz coordinates for each pixel), **aligned in** the **camera coordinate** system of **the first frame;** from which, camera intrinsics, pose, and depth can be derived

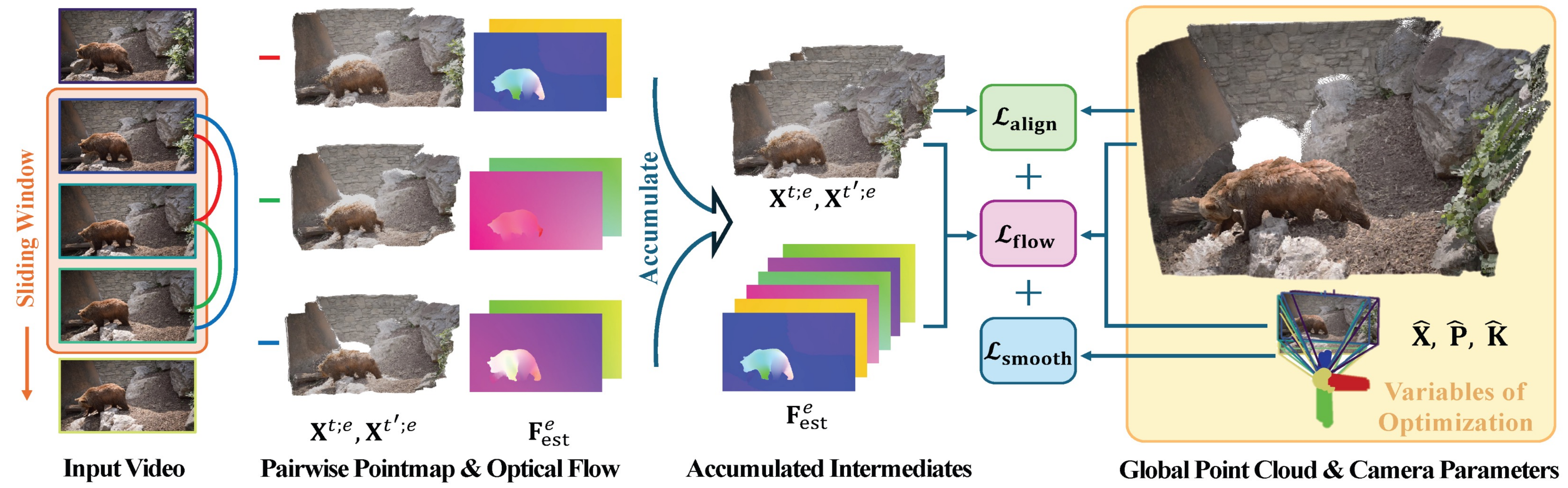
No constraint on dynamic/static scenes in the representation! But how does the model actually work for dynamic scenes? **->**

Limitation of DUSt3R on Dynamic Scenes

As this is mainly a **data issue**, we propose a simple approach to adapt DUSt3R to dynamic scenes, by **fine-tuning** on a small set of dynamic videos, which surprisingly works well

Dynamic Global Point Cloud

for video input, aggregate pairwise results to build global point cloud with global alignment



Quantitative & Qualitative results

Table 1: Video depth evaluation

Alignment	Category	Method	Sintel Abs Rel ↓	Sintel δ<1.25 ↑	Bonn Abs Rel ↓	Bonn δ<1.25 ↑	KITTI Abs Rel ↓	KITTI δ<1.25 ↑
Per-sequence scale & shift	Single-frame depth	Marigold	0.532	51.5	0.091	93.1	0.149	79.6
		Depth-Anything-V2	0.367	55.4	0.106	92.1	0.140	80.4
	Video depth	NVDS	0.408	48.3	0.167	76.6	0.253	58.8
		ChronoDepth	0.687	48.6	0.100	91.1	0.167	75.9
		DepthCrafter (Sep. 2024)	**0.292**	**69.7**	0.075	**97.1**	0.110	88.1
	Joint video depth & pose	Robust-CVD	0.703	47.8	-	-	-	-
		CasualSAM	0.387	54.7	0.169	73.7	0.246	62.2
		MonST3R	0.335	58.5	**0.063**	96.4	**0.104**	**89.5**
Per-sequence scale	Video depth	DepthCrafter (Sep. 2024)	0.692	53.5	0.217	57.6	0.141	81.8
	Joint depth & pose	**MonST3R**	**0.345**	**56.2**	**0.065**	**96.3**	**0.106**	**89.3**

Table 2: Camera pose estimation

Category	Method	Sintel ATE ↓	Sintel RPE trans ↓	Sintel RPE rot ↓	TUM-dynamics ATE ↓	TUM-dynamics RPE trans ↓	TUM-dynamics RPE rot ↓	ScanNet (static) ATE ↓	ScanNet (static) RPE trans ↓	ScanNet (static) RPE rot ↓
Pose only	DROID-SLAM*	0.175	0.084	1.912	-	-	-	-	-	-
	DPVO*	0.115	0.072	1.975	-	-	-	-	-	-
	ParticleSfM	0.129	**0.031**	**0.535**	-	-	-	0.136	0.023	0.836
	LEAP-VO*	**0.089**	0.066	1.250	0.068	**0.008**	1.686	0.070	0.018	**0.535**
Joint depth & pose	Robust-CVD	0.360	0.154	3.443	0.153	0.026	3.528	0.227	0.064	7.374
	CasualSAM	0.141	0.035	0.615	0.071	0.010	1.712	0.158	0.034	1.618
	DUSt3R w/ mask†	0.417	0.250	5.796	0.083	0.017	3.567	0.081	0.028	0.784
	MonST3R	0.108	0.042	0.732	**0.063**	0.009	**1.217**	**0.068**	**0.017**	0.545

* requires ground truth camera intrinsics as input, † unable to estimate the depth of foreground object.