Supplementary Material: *MoSca*: Dynamic Gaussian Fusion from Casual Videos via 4D Motion Scaffolds

Jiahui Lei[1] Yijia Weng[2] Adam W. Harley[2] Leonidas Guibas[2] Kostas Daniilidis[1,3]
[1] University of Pennsylvania [2] Stanford University [3] Archimedes, Athena RC
{leijh, kostas}@cis.upenn.edu, {yijiaw, aharley, guibas}@cs.stanford.edu

S.1. Code

We release the full implementation and hyper-parameters at https://www.cis.upenn.edu/~leijh/projects/mosca.

S.2. More Results

Please check our supplemental video for more results.

S.3. Detailed Metrics

We report detailed per-scene metrics in Tab. S.5, Tab. S.6 and Tab. S.7.

S.4. More Ablation

S.4.1. Running Time and More specs

We report more detailed specs, per-component running time, and peak GPU VRAM consumption from the raw video frames on DyCheck in Tab S.1. Note that *MoSca* can run on a reasonably limited hardware setup in relatively efficient fitting

S.4.2. Robustness

We manually add Gaussian noise to simulate the failure of one or multiple 2D priors. In Tab. S.2, the noise magnitude is indicated as STD of pixels for RAFT and TAP and cm for Depths (as D $x$cm). We also randomly drop $x\%$ of visible slots on a track to simulate losing tracks. Note that all noise is independent across time, making these simulated failures more challenging than real cases since 2D models tend not to make such independently incorrect predictions. From Tab. S.2 we do observe a performance decrease, but MoSca still converges and behaves reasonably. Among the 3 priors, it turns out that the TAP model is the pivot.

S.4.3. Ablation on Geometry Regularization

"no geometric optimization stage" in Tab. 5 in our main paper refers to completely skipping stage C in Fig. 2 and using trivial initialization while still enforcing the ARAP and Vel geo-regularizers in photometric optimization. We further study the effect of the geo-regularizers separately in Tab. S.3 ($10\times$ means increase the loss weight by 10), where we observe the significance of ARAP loss and the geometric optimization (step-C) phase, and a performance decrease in LPIPS when removing the Vel loss.

S.4.4. Different Priors and Camera Pose

How will different 2D foundational models affect the camera pose accuracy? We provide additional studies on different priors, replacing the UniDepth and SpaTracker used in the main paper with older methods ZoeDepth and Bootstapir or newer ones Metric3D and CoTracker-v3 in Tab. S.4, where we observe relatively consistent performance that matches or outperforms SOTA.

S.5. Camera Solver Additional Details

As mentioned in Section 3.2.2 of the main paper, we provide more details about the camera initialization process here.

S.5.1. Field of View (FOV) Initialization

The first step in the background stage is to initialize the projection matrix. We simplify the camera projection model to have only one degree of freedom—the Field of View (FOV), corresponding to a shared focal length. Because there is only one degree of freedom, we can perform a linear search to find the initial FOV that best explains the estimated tracking and depth.

Given an FOV and any two time frames, we use the depth to unproject the co-visible pixels along the static 2D trajec-

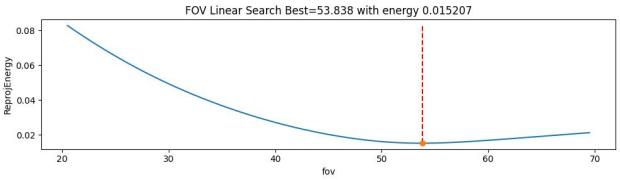

Figure S.1. Energy landscape during FOV enumeration

Table S.1. More details of the pipeline, extending Tab.7 in the main paper

Running time, VRAM and Model detials					
#FG_GS	106596	#FG_GS/#Nodes	46.1	skin corr mag mean	0.007
#Nodes	3176.7	#FG_GS/#Frames	241.5	skin corr mag std	0.026
#BG_GS	311457	#ALL_GS/#Frames	1062.2	skin corr mag max	1.179
prep dep time	1.8 min	fit ba time	2.5 min	prep peak VRAM	8.1 G
prep flow time	8.7 min	fit geo time	5.6 min	fit ba peak VRAM	6.6 G
prep tap time	7.4 min	fit photo time	38.6 min	fit geo peak VRAM	12.0 G
total prep time	18.0 min	total train time	46.7 min	fit photo peak VRAM	9.3 G
Ours **total**	64.7 min	Shape-of-Motion **train**	∼2 hrs	inference VRAM	1.3 G
RoDyNerf **train**	∼28 hrs	GS. Marbles **train**	∼5-9 hrs	Test FPS	37.8

Table S.2. Robustness on Dycheck SPIN scene

	RAFT 0pix				RAFT 5pix			
mPSNR↑	D 0cm	D 5cm	D 10cm	D 15cm	D 0cm	D 5cm	D 10cm	D 15cm
TAP 0pix 0%	20.79	20.59	20.30	19.84	20.26	20.16	19.93	18.00
TAP 5pix 5%	19.78	19.80	19.38	19.38	19.42	19.37	19.26	17.69
TAP 10pix 10%	18.70	18.18	18.20	17.36	18.47	18.08	18.10	16.88
TAP 20pix 25%	16.51	16.54	16.49	16.31	16.52	16.58	16.46	16.38
	RAFT 10pix				RAFT 20pix			
mPSNR↑	D 0cm	D 5cm	D 10cm	D 15cm	D 0cm	D 5cm	D 10cm	D 15cm
TAP 0pix 0%	20.15	18.10	19.97	17.96	20.12	20.15	19.96	17.98
TAP 5pix 5%	19.31	19.43	19.22	17.67	19.42	19.39	19.23	17.68
TAP 10pix 10%	18.49	18.10	18.13	17.03	18.48	19.39	18.12	17.00
TAP 20pix 25%	16.52	16.62	16.48	16.46	16.53	16.56	16.49	16.39

Table S.3. Effect of geometric regularizers on DyCheck

DyCheck	mPSNR↑	mSSIM↑	mLPIPS↓
full (main paper)	19.32	0.706	0.264
10× ARAP	19.31	0.709	0.259
10× Vel	19.12	0.702	0.267
No ARAP	18.84	0.695	0.287
No Vel	19.32	0.705	0.272
No geo (Tab.5)	18.85	0.693	0.287
No geo&ARAP	18.82	0.690	0.293
No geo&Vel	18.82	0.691	0.295
No geo&ARAP&Vel	18.76	0.687	0.300

Table S.4. Camera results under different 2D foundational models

TUM dataset	ATE↓	RPE trans↓	RPE rot↓
zoedepth bootstapir	0.038	0.012	0.445
zoedepth cotracker	0.039	0.013	0.455
zoedepth spatracker	0.038	0.012	0.445
unidepth bootstapir	0.034	0.012	0.425
unidepth cotracker	0.044	0.014	0.453
unidepth spatracker (paper)	0.031	0.011	0.426
metric3d bootstapir	0.035	0.012	0.428
metric3d cotracker	0.034	0.012	0.428
metric3d spatracker	0.034	0.012	0.423

tories between these frames. We then analytically solve a $\mathrm{SIM}(3)$ Procrustes problem to find the best relative pose between the two cameras. Using this pose, we compute the reprojection error between the depths.

After computing all possible time pairs for a fixed FOV, we construct a reprojection energy for it. Therefore, we can perform a linear search over all possible FOVs to find the optimal FOV that minimizes this energy. Note that all the above operations can be efficiently computed in parallel, allowing the overall enumeration to be completed within seconds. Figure S.1 shows an example of such an enumeration on a DyCheck [1] iPhone sequence. This algorithm provides an initial FOV estimate, which will be further optimized in the global bundle adjustment (BA). If this smart initial guess fails, we fall back to a predefined starting FOV value.

S.5.2. Smoothness Regularization

Real-world camera trajectories are usually smooth. Therefore, in practice, in addition to the bundle adjustment loss defined in Eq. 6 and Eq. 7 of the main paper, we add a smoothness regularization loss:

$$\mathcal{L}_{\text{cam-smooth}} = \lambda_R \| \log\left(\mathbf{R}_{t+1}^{-1}\mathbf{R}_t\right) \|_F + \lambda_t \|\mathbf{t}_{t+1} - \mathbf{t}_t\|, \quad (1)$$

where $\mathbf{R}$ and $\mathbf{t}$ are the rotation and translation components of the camera pose $\mathbf{W}$ at each time step. Another practical trick we use to enhance smoothness is parameterizing

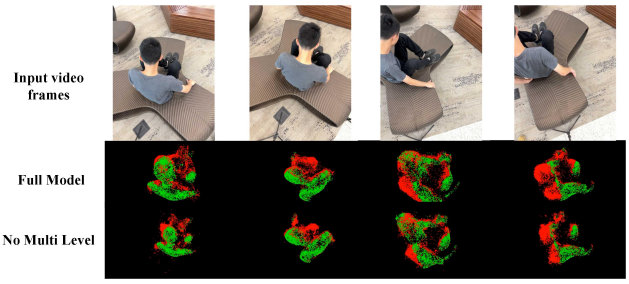

Figure S.2. The first row shows input views with challenging truncation and occlusion, where we cannot trivially back-project the 2D tracker. These challenging areas justify the necessity of the geometry optimization stage in Section 3.2.3. The second and third rows visualize the scaffold nodes after geometric optimization with the ARAP loss, where the green nodes are visible from the input view and the red nodes are invisible and solved by the optimization.

the camera poses as a list of incremental (delta) poses between neighboring frames during the early stages of the BA optimization.

S.6. Additional Details about the Scaffold

As mentioned in Section 3.2.3 of the main paper, we provide more details about the motion scaffold here.

S.6.1. Resampling of Nodes

When inferring the foundational 2D long-term tracker, we sample much more densely (e.g., around $30k$ queries) across all frames of the video because we do not know the accurate dynamic foreground mask and the foreground 3D motion at the very beginning. Some trackers, like CoTracker [2], depend on dense sampling for optimal performance. However, the number of our MoSca nodes is usually far less than $30k$, so we need to resample nodes from the denser raw tracker-queried trajectories.

Given many lifted 3D curves, we use the curve distance defined in Eq. 2 to compute a dense distance matrix. We sort all these curves by the number of time steps they have been visible. Using this order, we iterate over all curves and include a curve in the subsampled set if it is farther than a threshold unit from any already included curves. This spatial unit controls the overall density of the nodes and is empirically set to a predefined fixed value. Depending on the scene, the number of subsampled nodes is usually around 100 to 4000, compared to the raw $30k$ tracker output.

S.6.2. Multi-Level Topology

As shown in our ablation study in Sec. 4.3, the multi-level topology is critical when using ARAP regularization, especially when the object is largely truncated and occluded. Figure S.2 provides more examples. Note that without regularizing on a multi-level topology, the invisible occluded nodes will not be optimized to reasonable configurations. To augment the topology $\mathcal{E}$ defined in Eq.2, we simply subsample the nodes using different spatial units with the resampling algorithm introduced above in SectionS.6.1. The multi-level ARAP loss is applied to the neighborhood pair of nodes (edges) between coarse and denser levels. All level neighboring pairs are jointly united to form the augmented multi-level topology $\tilde{\mathcal{E}}$ for computing the ARAP losses in Eq. 9. Note that the multi-level topology is only used to compute ARAP losses and is not involved in the skinning.

References

[1] Hang Gao, Ruilong Li, Shubham Tulsiani, Bryan Russell, and Angjoo Kanazawa. Monocular dynamic view synthesis: A reality check. *Advances in Neural Information Processing Systems*, 35:33768–33780, 2022. 2, 4

[2] Nikita Karaev, Ignacio Rocco, Benjamin Graham, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. Cotracker: It is better to track together. *arXiv preprint arXiv:2307.07635*, 2023. 3

[3] Jae Shin Yoon, Kihwan Kim, Orazio Gallo, Hyun Soo Park, and Jan Kautz. Novel view synthesis of dynamic scenes with globally coherent depths from a monocular camera. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5336–5345, 2020. 5

Table S.5. Per-scene metrics for DyCheck [1] in the setting of with camera pose.

	Apple			Block			paper-windmill			space-out		
	mPSNR↑	mSSIM↑	mLPIPS↓	mPSNR↑	mSSIM↑	mLPIPS↓	mPSNR↑	mSSIM↑	mLPIPS↓	mPSNR↑	mSSIM↑	mLPIPS↓
T-NeRF	17.43	0.728	0.508	17.52	0.669	0.346	17.55	0.367	0.258	17.71	0.591	0.377
NSFF	16.47	0.754	0.414	14.71	0.606	0.438	14.94	0.272	0.348	17.65	0.636	0.341
Nerfies	17.54	0.750	0.478	16.61	0.639	0.389	17.34	0.378	0.211	17.79	0.622	0.303
HyperNeRF	17.64	0.743	0.478	17.54	0.670	0.331	17.38	0.382	0.209	17.93	0.605	0.320
PGDVS	16.66	0.721	0.411	16.38	0.601	0.293	17.19	0.386	0.277	16.49	0.592	0.326
DyPoint	17.78	0.743	-	17.67	0.667	-	17.32	0.366	-	17.78	0.603	-
DpDy	-	0.735	0.596	-	0.630	0.478	-	0.387	0.447	-	0.622	0.457
Dyn. gaussians	7.65	-	0.766	7.55	-	0.684	6.24	-	0.729	6.79	-	0.733
4D GS	15.41	-	0.456	11.28	-	0.633	15.60	-	0.297	14.60	-	0.372
Gaussian Marbles	17.7	-	0.492	17.42	-	0.384	17.04	-	0.394	15.94	-	0.435
DyBluRF	18.00	0.737	0.488	17.47	0.665	0.349	18.19	0.405	0.301	18.83	0.643	0.326
CTNeRF	19.53	0.691	-	19.74	0.626	-	17.66	0.346	-	18.11	0.601	-
D-NPC	18.79	0.763	0.414	16.38	0.649	0.320	17.99	0.409	0.212	16.44	0.616	0.346
Shape-of-Motion	18.57	0.771	0.341	17.41	0.644	0.323	18.14	0.415	0.225	16.85	0.601	0.324
Ours	19.40	0.810	0.340	18.06	0.680	0.330	22.34	0.740	0.150	20.48	0.660	0.260
	spin			teddy			wheel			AVE		
	mPSNR↑	mSSIM↑	mLPIPS↓	mPSNR↑	mSSIM↑	mLPIPS↓	mPSNR↑	mSSIM↑	mLPIPS↓	mPSNR↑	mSSIM↑	mLPIPS↓
T-NeRF	19.16	0.567	0.443	13.71	0.570	0.429	15.65	0.548	0.292	16.96	0.577	0.379
NSFF	17.26	0.540	0.371	12.59	0.537	0.527	14.59	0.511	0.331	15.46	0.551	0.396
Nerfies	18.38	0.585	0.309	13.65	0.557	0.372	13.82	0.458	0.310	16.45	0.570	0.339
HyperNeRF	19.20	0.561	0.325	13.97	0.568	0.350	13.99	0.455	0.310	16.81	0.569	0.332
PGDVS	18.49	0.590	0.247	13.29	0.516	0.399	12.68	0.429	0.429	15.88	0.548	0.340
DyPoint	19.04	0.564	-	13.95	0.551	-	14.72	0.515	-	16.89	0.573	-
DpDy	-	0.500	0.571	-	0.531	0.562	-	0.511	0.504	-	0.559	0.516
Dyn. gaussians	8.08	-	0.651	7.41	-	0.690	7.28	-	0.593	7.29	-	0.692
4D GS	14.42	-	0.339	12.36	-	0.466	11.79	-	0.436	13.64	-	0.428
Gaussian Marbles	18.88	-	0.428	13.95	-	0.442	16.14	-	0.351	16.72	-	0.418
DyBluRF	18.20	0.541	0.400	14.61	0.572	0.425	16.26	0.575	0.325	17.37	0.591	0.373
CTNeRF	19.79	0.516	-	14.51	0.509	-	14.48	0.430	-	17.69	0.531	-
D-NPC	18.48	0.565	0.284	13.70	0.559	0.401	16.10	0.618	0.253	16.84	0.597	0.319
Shape-of-Motion	19.35	0.582	0.247	13.69	0.542	0.380	17.21	0.628	0.230	17.32	0.598	0.296
Ours	21.31	0.750	0.190	15.47	0.620	0.350	18.17	0.680	0.230	19.32	0.706	0.264

Table S.6. Per-scene metrics for DyCheck [1] in the setting of NO camera pose.

	Apple			Block			paper-windmill			space-out		
	mPSNR↑	mSSIM↑	mLPIPS↓	mPSNR↑	mSSIM↑	mLPIPS↓	mPSNR↑	mSSIM↑	mLPIPS↓	mPSNR↑	mSSIM↑	mLPIPS↓
RobustDynrf	18.73	0.722	0.552	18.73	0.634	0.513	16.71	0.321	0.482	18.56	0.594	0.413
Dyn. gaussians	7.96	-	0.775	7.13	-	0.737	6.732	-	0.736	7.42	-	0.698
4D GS	14.44	-	0.716	12.30	-	0.706	14.46	-	0.790	14.93	-	0.640
Gaussian Marbles	16.50	-	0.499	16.11	-	0.363	16.19	-	0.454	15.97	-	0.437
Ours COLFREE	15.87	0.700	0.510	18.25	0.670	0.320	21.40	0.650	0.170	22.40	0.750	0.200
Ours COLFREE(w. focal)	17.21	0.740	0.430	18.12	0.670	0.320	21.47	0.660	0.170	22.44	0.740	0.200
	spin			teddy			wheel			AVE		
	mPSNR↑	mSSIM↑	mLPIPS↓	mPSNR↑	mSSIM↑	mLPIPS↓	mPSNR↑	mSSIM↑	mLPIPS↓	mPSNR↑	mSSIM↑	mLPIPS↓
RobustDynrf	17.41	0.484	0.570	14.33	0.536	0.613	15.20	0.449	0.478	17.10	0.534	0.517
Dyn. gaussians	9.15	-	0.635	7.75	-	0.709	7.03	-	0.641	7.60	-	0.704
4D GS	12.77	-	0.697	11.86	-	0.729	10.99	-	0.803	13.11	-	0.726
Gaussian Marbles	17.51	-	0.424	13.68	-	0.443	14.58	-	0.389	15.79	-	0.430
Ours COLFREE	20.67	0.670	0.230	15.62	0.630	0.350	17.68	0.660	0.240	18.84	0.676	0.289
Ours COLFREE(w. focal)	20.83	0.690	0.230	15.46	0.620	0.360	17.58	0.660	0.240	19.02	0.683	0.279

Table S.7. Per-scene metrics for NVIDIA [3].

PSNR LPIPS	Jumping		Skating		Truck		Umbrella		Balloon1		Balloon2		Playground		AVE	
D-NeRF	22.36	0.193	22.48	0.323	24.10	0.145	21.47	0.264	19.06	0.259	20.76	0.277	20.18	0.164	21.49	0.232
NR-NeRF	20.09	0.287	23.95	0.227	19.33	0.446	19.63	0.421	17.39	0.348	22.41	0.213	15.06	0.317	19.69	0.323
TiNeuVox	20.81	0.247	23.32	0.152	23.86	0.173	20.00	0.355	17.30	0.353	19.06	0.279	13.84	0.437	19.74	0.285
HyperNeRF	18.34	0.302	21.97	0.183	20.61	0.205	18.59	0.443	13.96	0.530	16.57	0.411	13.17	0.495	17.60	0.367
NSFF	24.65	0.151	29.29	0.129	25.96	0.167	22.97	0.295	21.96	0.215	24.27	0.222	21.22	0.212	24.33	0.199
DynNeRF	24.68	0.090	32.66	0.035	28.56	0.082	23.26	0.137	22.36	0.104	27.06	0.049	24.15	0.080	26.10	0.082
MonoNeRF	24.26	0.091	32.06	0.044	27.56	0.115	23.62	0.180	21.89	0.129	27.36	0.052	22.61	0.130	25.62	0.106
4DGS	21.93	0.269	24.84	0.174	23.02	0.175	21.83	0.213	21.32	0.185	18.81	0.178	18.40	0.196	21.45	0.199
Casual-FVS	23.45	0.100	29.98	0.045	25.22	0.090	23.24	0.096	23.75	0.079	24.15	0.081	22.19	0.074	24.57	0.081
CTNeRF	24.35	0.094	33.51	0.034	28.27	0.084	23.48	0.129	22.19	0.111	26.86	0.048	24.28	0.077	26.13	0.082
DynPoint	24.69	0.097	31.34	0.045	29.30	0.061	24.59	0.086	22.77	0.099	27.63	0.049	25.37	0.039	26.53	0.068
D-NPC	24.51	0.116	30.22	0.061	28.92	0.084	24.15	0.136	22.26	0.161	25.84	0.107	23.59	0.099	25.64	0.109
RoDynRF	25.66	0.071	28.68	0.040	29.13	0.063	24.26	0.089	22.37	0.103	26.19	0.054	24.96	0.048	25.89	0.067
RoDynRF w/o-COL	24.27	0.100	28.71	0.046	28.85	0.066	23.25	0.104	21.81	0.122	25.58	0.064	25.20	0.052	25.38	0.079
Gaussian Marbles	19.61	0.180	24.24	0.091	27.18	0.060	23.76	0.123	23.65	0.072	21.60	0.142	16.21	0.235	22.32	0.129
ours	25.01	0.090	33.41	0.030	27.83	0.080	25.17	0.090	23.58	0.100	27.80	0.050	24.25	0.050	26.72	0.070
ours w/o-COL	25.07	0.080	32.25	0.050	27.75	0.080	25.42	0.090	23.54	0.100	27.66	0.060	24.12	0.050	26.54	0.073