Marginalized Bundle Adjustment: Multi-View Camera Pose from Monocular Depth Estimates

Supplementary Material

Dataset	Max Graph Size		Max Runtime	
	Images	Pairs	Ours	COLMAP
ScanNet [14]	391	43,964	2.98h	0.91h
ETH3D [59]	76	3,142	41min	9min
IMC2021 [6]	25	600	14min	3min
7-Scenes [60]	8,000	564,418	3.12h	13.52h
Wayspots [9]	1,157	1,333,196	4.72h	Crash

Table 9. **Max Pose Graph and its Runtime** in each benchmarked dataset. We run experiments on ScanNet, ETH3D, and IMC2021 with single V100 GPU, and on 7-Scenes and Wayspots with eight.

6. Extended Methodology

Compare Surrogate Loss to MAGSAC. We rewrite the surrogate loss Eq. (10) into an integration form:

$$\mathcal{L}_{\text{MBA}} = \frac{1}{\|R\|} \sum_{i,j,k} -F(r_{i,j,k}) \cdot \mathbb{1}[r_{i,j,k} < \tau_{\max}] \\ = -\int_{0}^{\tau_{\max}} F(r) \cdot p(r)\ dr. \tag{14}$$

The formulation in Eq. (14) resembles the scoring function presented by MAGSAC [2]. In Eq. (14), we marginalize the residual probability density function $p(r)$ with its empirical CDF function $F(r)$. MAGSAC [2] instead replaces the empirical CDF function $F(r)$ with a fixed chi-square distribution derived from assumptions on the distribution of residual errors for inliers and outliers. Despite their similarity, Eq. (10) only serves as a surrogate forward loss for our BA objective. We compare RANSAC performance using the proposed scoring function Eq. (8) against MAGSAC in Tab. 8. Our more generalized scoring function demonstrates comparable performance. Please refer to Sec. 7.1 for details of the experiments.

7. Extended Experiments

7.1. Two-View RANSAC

We follow the RoMa [20] evaluation protocol to benchmark RANSAC-based essential matrix estimation. Results on the MegaDepth-1500 [38] and ScanNet-1500 [15] benchmarks are reported in Tab. 8. Replacing RoMa's default RANSAC, we evaluate MAGSAC++ and our method under identical budgets, tuning both over the same inlier-threshold grid and sampling the same number of correspondences. For both methods, we sample 1 000 correspondences and sweep the inlier threshold over the grid $\{0.02, 0.05, 0.1, 0.2, 0.3, 0.5, 0.8, 1.0, 1.2, 1.5, 1.7, 2.0\}$ in normalized pixel coordinates. Our method uses a GPU-parallelized estimator: we always randomly compute 64 minimal solutions in parallel and retain the one that maximizes the scoring function in Eq. (9). We use OpenCV's MAGSAC++ implementation. Both MAGSAC++ and our method substantially outperform standard RANSAC, and our alternative scoring function Eq. (9) achieves performance on par with MAGSAC++. These results indicate that our RANSAC-motivated scoring function applies to two-view essential matrix estimation, beyond the multi-view pose setting.

Ablation	Method	RRA@5°	RTA@5°
Stages	Initialization	62.6	41.0
	Coarse	97.1	87.9
	Fine	97.3	90.2
Density	$\kappa = 30$	91.7	78.6
	$\kappa = 200$	97.3	90.2
	$\kappa = 500$	99.3	92.0
Losses	Soft L1	87.5	73.9
	Cauchy	86.2	75.2
	Tukey	63.5	40.7
	L2	79.4	66.0
	MBA (ours)	97.3	90.2

Table 10. **Ablations** on ETH3D dataset.

Type	Method	ETH3D Dataset		
		AUC@1°	AUC@3°	AUC@5°
Detector-Based	SIFT+NN + COLMAP [56] CVPR'16	26.71	38.86	42.14
	SIFT + NN + PixSfM [41] ICCV'21	26.94	39.01	42.19
	D2Net + NN + PixSfM [41]ICCV'21	34.50	49.77	53.58
	R2D2 + NN + PixSfM [41] ICCV'21	43.58	62.09	66.89
	SP + SG + PixSfM [41] ICCV'21	50.82	68.52	72.86
Detector-Free	LoFTR + PixSfM [41] ICCV'21	54.35	73.97	78.86
	LoFTR + DF-SfM [28] CVPR'24	59.12	75.59	79.53
	AspanTrans. + DF-SfM [28]CVPR'24	57.23	73.71	77.70
	MatchFormer + DF-SfM [28]CVPR'24	56.70	73.00	76.84
Dense Matching	DKM + Dense-SfM [32]CVPR'25	59.04	77.73	82.20
	RoMa + Dense-SfM [32]CVPR'25	**60.92**	**78.41**	**82.63**
Deep-based	VGG-SfM [73]CVPR'24	(compared in Tab. 2)		
Point-Based	Mast3r-SfM [18]arXiv'24	35.85	58.46	65.03
	Dense-SfM + Mast3r-SfM [18]arXiv'24	37.50	59.18	65.48
MDE	SfMfM (Ours)	27.72	63.35	74.02

Table 11. **Structure-from-Motion** on ETH3D [58, 59] dataset in metric AUC at multiple thresholds following DF-SfM.

7.2. FastMap Benchmark Comparison

Following FastMap [36], we evaluate our method on a comprehensive set of large-scale real-world datasets that cover diverse camera trajectory patterns and scene complexities. The evaluation includes eight datasets: Mip-NeRF360 [4], Tanks and Temples [30], NeRF-OSR [52],

		ATE↓				RTA@3↑				AUC-R&T @ 3 ↑				RTA@1↑				AUC-R&T @ 1 ↑			
	n_imgs	MBA (Ours)	FastMap [36]	GLOMAP [47]	COLMAP [56]	MBA (Ours)	FastMap [36]	GLOMAP [47]	COLMAP [56]	MBA (Ours)	FastMap [36]	GLOMAP [47]	COLMAP [56]	MBA (Ours)	FastMap [36]	GLOMAP [47]	COLMAP [56]	MBA (Ours)	FastMap [36]	GLOMAP [47]	COLMAP [56]
mipnerf360 (9)	215.6	5.0e-4	4.2e-4	3.3e-5	5.8e-5	99.6	99.9	100.0	100.0	85.5	97.4	98.2	97.2	87.3	99.8	100.0	99.7	66.1	92.4	94.6	91.9
tnt_advanced (6)	337.8	1.7e-2	6.4e-3	1.2e-2	1.2e-3	59.9	71.4	79.1	98.5	34.7	42.6	75.3	94.8	25.2	42.3	77.5	97.0	12.5	16.7	69.8	90.0
tnt_intermediate (8)	268.6	5.7e-3	7.8e-5	1.9e-5	2.6e-4	89.8	99.9	100.0	99.8	61.7	94.1	99.0	98.9	62.8	99.3	99.9	99.5	35.4	83.1	96.9	97.3
tnt_training (7)	470.1	3.8e-3	3.0e-3	1.1e-2	3.0e-4	88.8	87.8	88.7	99.9	63.2	77.2	87.9	99.5	63.6	82.1	88.6	99.9	31.9	60.5	86.3	98.7
nerf_osr (8)	402.8	1.4e-3	1.6e-3	1.1e-3	1.3e-3	89.8	91.7	92.0	92.1	69.3	70.9	71.9	71.7	54.4	71.1	71.9	71.7	35.0	43.2	45.2	44.7
drone_deploy (9)	524.7	2.4e-3	4.9e-3	4.3e-3	2.0e-3	89.6	97.9	98.2	91.3	71.8	79.2	81.1	65.2	72.4	89.6	91.5	73.5	46.6	50.4	53.5	40.2
urban_scene (3)	3824	8.1e-5	1.7e-5	1.4e-5	1.4e-5	99.0	99.9	99.9	100.0	85.9	95.3	97.0	97.0	94.1	99.5	99.6	99.6	63.2	86.3	91.2	91.3
mill19_building	1920	5.1e-5	3.0e-4	1.3e-2	1.9e-5	99.6	99.9	0.1	99.9	93.5	95.5	0.0	95.6	95.9	99.3	0.0	99.3	81.8	87.0	0.0	87.4
mill19_rubble	1657	4.3e-5	3.6e-5	6.4e-5	3.4e-5	99.9	99.9	99.8	99.9	95.8	93.6	94.5	94.6	98.4	98.6	98.6	98.7	87.8	81.6	84.7	84.8
eyeful_apartment	3804	3.6e-3	2.8e-3	9.4e-3	2.2e-3	57.8	86.8	75.0	90.2	34.5	45.5	50.5	62.0	21.2	51.1	61.3	71.7	8.1	6.4	18.2	21.9
eyeful_kitchen	6042	9.7e-4	3.1e-3	7.4e-3	-	72.3	85.0	59.9	-	46.0	38.1	41.2	-	33.7	46.7	51.7	-	13.4	4.6	14.4	-

Table 12. **Pose accuracy comparison** on MipNeRF360 [4], Tanks and Temples [30], NeRF-OSR [52], DroneDeploy [50], Urbanscene3D [39], Mill-19 [70], and Eyeful Tower [82]. We follow the evaluation protocol established by FastMap [36] for reporting pose accuracy metrics on these datasets. Note, [36] uses a different COLMAP groundtruth to Tab. 4. For datasets with multiple scenes, we denote the average number of images as `dataset-name(#scenes)`. Results are listed separately for each scene in Mill-19 and Eyeful Tower. Metrics are color-coded with dark green for best performance and light green for competitive performance. Red denotes complete failures and gray indicates timeout (did not finish in one week).

DroneDeploy [50], Urbanscene3D [39], Mill-19 [70], and Eyeful Tower [82], with scene sizes ranging from approximately 200 to 6,000 images per scene. We compare against FastMap, GLOMAP [47], and COLMAP [56] using standard pose accuracy metrics including ATE, RTA@δ, and AUC-R&T@δ at multiple thresholds. The results are presented in Tab. 12, where we report per-dataset averages for multi-scene datasets and individual results for Mill-19 and Eyeful Tower scenes.

7.3. Ablations

Runtime. Our work primarily addresses the challenge of applying pre-trained monocular depth estimators (MDE) to multi-view pose estimation. As a result, computational efficiency has not been a primary focus of this work. In particular, we used first-order gradient descent for optimization, which can be less efficient than second-order methods. In Tab. 9, we present a runtime comparison with COLMAP. We report only the Bundle Adjustment time, excluding any preprocessing overhead. Overall, our method running 50k iterations is approximately 2–4× slower than COLMAP. The use of first-order optimization facilitates scaling up to substantially larger problem set which is nontrivial to achieve with second-order optimization. Notably, COLMAP crashes on the Wayspots dataset. In Tab. 10, we ablate the number of iterations, only running 5k steps with a sophisticated optimization scheme. (Detailed in paragraph Optimization Strategy). Yet, this requires dataset-specific engineering, hurting the generalization capability of our method. We leave a more thorough investigation into computational efficiency for future work.

Number of sampled pixels (κ). Our method requires a certain level of sampling density to achieve the desired level of accuracy, as shown in Tab. 10 on ETH3D. With insufficient sampling ($\kappa = 30$), the results significantly decrease to 91.7% RRA and 78.6% RTA at 5°. Our default configuration ($\kappa = 200$) achieves 98.0% RRA and 91.3% RTA. Further increasing the sampling density to $\kappa = 500$ yields additional minor improvements, reaching 99.3% RRA and 92.0% RTA, slightly outperforming the numbers reported in the main paper (97.3% RRA and 90.2% RTA at 5° in Tab. 2).

Loss Function Comparison. We ablate different losses in Tab. 10 for their effectiveness for monocular depth-based pose estimation. Our marginalized Bundle-Adjustment (MBA) approach significantly outperforms traditional loss functions, achieving 97.3% RRA and 90.2% RTA. Among conventional losses, the soft L1 (87.5% RRA, 73.9% RTA) and Cauchy (86.2% RRA, 75.2% RTA) show moderate performance, while the l2 loss performs worst (79.4% RRA, 66.0% RTA). These findings highlight the importance of robust loss functions that can effectively handle noise in monocular depth estimates.

7.4. Extended ScanNet and 7-Scenes Results

We also report results on the ScanNet dataset using calibrated cameras (*i.e.*, known intrinsics) in Tab. 14. We further present the detailed per-sequence performance on the 7-Scenes dataset in Tab. 15.

7.5. Evaluation Protocols

ETH3D Dataset in Tab. 2. The ETH3D consists of 13 multi-view scenes containing up to 76 high-resolution photographs per scene. We evaluate on ETH3D dataset following two evaluation protocols. Tab. 2 follows MASt3R-SfM [18] to report *Relative Rotation Accuracy* (RRA@τ) and *Relative Translation Accuracy* (RTA@τ), which measure the percentage of image pairs whose estimated relative pose errors fall below a threshold $\tau = 5°$. For each image pair (i, j) with valid ground-truth poses, the rotation error is computed as the angular difference between the estimated and ground-truth relative rotations, *i.e.*, the angle of ${R_{ij}^{\text{gt}}}^{-1} R_{ij}^{\text{est}}$, while the translation error is the angle between the normalized translation directions t_{ij}^{gt} and t_{ij}^{est}. These errors are aggregated over all possible image pairs within each scene, yielding one RRA and one RTA score per scene. The final reported scores are obtained by averag-

Scene	COLMAP [56]		ACE-Zero [10]		FlowMap [62]		VGGSfM [73]		DF-SfM [28]		MASt3R-SfM [18]		Ours / DUSt3R [79]		Ours / ZoeDepth [5]		Ours / UniDepth [48]	
	RRA	RTA	RRA	RTA	RRA	RTA	RRA	RTA	RRA	RTA	RRA	RTA	RRA	RTA	RRA	RTA	RRA	RTA
courtyard	56.3	60.0	4.0	1.9	7.5	3.6	50.5	51.2	80.7	74.8	89.8	64.4	94.7	94.7	94.7	94.5	94.7	94.4
delivery area	34.0	28.1	27.4	1.9	29.4	23.8	22.0	19.6	82.5	82.0	83.1	81.8	83.1	83.0	87.8	82.0	83.1	83.1
electro	53.3	48.5	16.9	7.9	2.5	1.2	79.9	58.6	82.8	81.2	100.0	95.5	95.6	78.2	91.9	78.5	93.0	77.2
facade	92.2	90.0	74.5	64.1	15.7	16.8	57.5	48.7	80.9	82.6	74.3	75.3	100.0	99.2	100.0	97.4	80.9	86.0
kicker	87.3	86.2	26.2	16.8	1.5	1.5	100.0	97.8	93.5	91.0	100.0	100.0	100.0	98.9	100.0	98.5	100.0	98.0
meadow	0.9	0.9	3.8	0.9	3.8	2.9	100.0	96.2	56.2	58.1	58.1	58.1	100.0	58.1	45.7	33.3	100.0	56.7
office	36.9	32.3	0.9	0.0	0.9	1.5	64.9	42.1	71.1	54.5	100.0	98.5	100.0	86.2	100.0	85.7	100.0	86.5
pipes	30.8	28.6	9.9	1.1	6.6	12.1	100.0	97.8	72.5	61.5	100.0	100.0	100.0	96.7	100.0	94.5	100.0	97.8
playground	17.2	18.1	3.8	2.6	2.6	2.8	37.3	40.8	70.5	70.1	100.0	93.6	94.7	93.8	100.0	96.5	100.0	99.2
relief	16.8	16.8	16.8	17.0	6.9	7.7	59.6	57.9	32.9	32.9	34.2	40.2	100.0	98.9	100.0	97.4	100.0	99.6
relief 2	11.8	11.8	7.3	5.6	8.4	2.8	69.9	70.3	40.9	39.1	57.4	76.1	100.0	98.9	100.0	98.6	100.0	99.8
terrace	100.0	100.0	5.5	2.0	33.2	24.1	38.7	29.6	100.0	99.6	100.0	100.0	100.0	100.0	100.0	94.5	100.0	98.6
terrains	100.0	99.5	15.8	4.5	12.3	13.8	70.4	54.9	100.0	91.9	58.2	52.5	100.0	95.4	100.0	93.4	100.0	95.2
Average	49.0	47.8	16.4	9.7	10.1	8.8	65.4	58.9	74.2	70.7	81.2	79.7	**97.3**	**90.2**	93.9	88.1	96.3	**90.2**

Table 13. **Structure-from-Motion** ablation with other Monocular Depth Models ZoeDepth and UniDepth on ETH3D dataset [58, 59].

			Calibrated			Uncalibrated		
Method	Depth	Corres.	Acc@3°	Acc@5°	Acc@10°	Acc@3°	Acc@5°	Acc@10°
COLMAP [56]	-	SuperPoint [17]	0.398	0.589	0.783	0.342	0.505	0.670
SfMfM (Ours)	ZoeDepth [5]	RoMa [20]	0.396	0.614	0.823	0.372	0.586	0.811
	DUSt3R [79]	RoMa [20]	0.426	0.631	0.830	0.403	**0.615**	0.820
	UniDepth [48]	RoMa [20]	0.432	0.636	0.833	**0.407**	0.612	**0.823**
	DUSt3R [79]	MASt3R [34]	0.432	0.639	0.837	0.384	0.596	0.811
	UniDepth [48]	MASt3R [34]	**0.439**	**0.645**	**0.841**	0.393	0.598	0.817

Table 14. **Extended Structure-from-Motion Results** on the ScanNet dataset [14], with calibrated and uncalibrated cases.

ing the per-scene RRA and RTA across all 13 scenes.

IMC2021 in Tab. 3 We follow DF-SfM [28] in evaluating the Area Under the Curve (AUC) of relative pose accuracy at multiple thresholds. For each image pair with ground-truth camera poses, we compute the rotation error as the angular difference (in degrees) between the estimated and ground-truth relative rotations, and the translation error as the angle between the corresponding normalized translation directions. The pose error is defined as the maximum of the rotation and translation errors. The AUC at threshold τ is defined as the area under the cumulative distribution function (CDF) of pose errors up to τ, normalized by τ, where CDF(e) denotes the proportion of image pairs with pose error less than e degrees. On the IMC dataset, the AUC metric is computed globally across all valid image pairs, without per-scene aggregation. Following standard practice, we report AUC at thresholds: 3°, 5°, and 10°.

ScanNet Dataset. Similarly, for the ScanNet dataset, we define the pose error as the angular error given by the maximum of the rotation and translation angular errors. For each image pair with ground-truth poses, the rotation error is computed as the angular difference between the estimated and ground-truth relative rotations, and the translation error is defined as the angle between the corresponding normalized translation vectors. We report the *Accuracy* (ACC@τ), defined as the percentage of image pairs with pose error less than a given threshold τ. The metric is computed over all visible frame pairs within each scene and averaged to produce a per-scene ACC score. The final result is average across all scenes. For a fair comparison with COLMAP, we only report scores over the subset of frame pairs for which COLMAP successfully returns a pose estimate.

FastMap Benchmark. Following FastMap [36], we report standard pose accuracy metrics including Absolute Translation Error (ATE), Relative Translation Accuracy (RTA@δ), and Area Under the Curve for Rotation and Translation (AUC-R&T@δ) at thresholds of 1° and 3°. For each image pair with valid ground-truth poses, the rotation error is computed as the angular difference between estimated and ground-truth relative rotations, and the translation error as the angle between normalized translation directions. ATE measures the absolute translation error magnitude. RTA@δ reports the percentage of image pairs with both rotation and translation errors below threshold δ. AUC-R&T@δ computes the area under the accuracy curve up to threshold δ. For multi-scene datasets, we report per-dataset averages, while Mill-19 and Eyeful Tower results are listed per scene.

7-Scenes Dataset. For the 7-Scenes dataset [60], we follow DUSt3R [79] on the standard evaluation protocol by computing the median rotation and translation errors across all test frames. The translation error is measured as the Euclidean distance (in centimeters) between the predicted and ground-truth camera positions, while the rotation error is computed as the angular difference (in degrees) between the predicted and ground-truth orientations. These median errors provide a robust summary of pose estimation accuracy in the presence of outliers and are reported for each scene individually. The final scores are obtained by averaging the

Category	Method	Chess	Fire	Heads	Office	Pumpkin	Kitchen	Stairs	Average
FM	AS [55]PAMI'16	4/1.96	3/1.53	2/1.45	9/3.61	8/3.10	7/3.37	3/2.22	5.1/2.46
	HLoc [53]CVPR'19	2/0.79	2/0.87	2/0.92	3/0.91	5/1.12	4/1.25	6/1.62	3.4/1.07
E2E	SC-wLS [81]ECCV'22	3/0.76	5/1.09	3/1.92	6/0.86	8/1.27	9/1.43	12/2.80	6.6/1.45
	NeuMaps [65]CVPR'23	2/0.81	3/1.11	2/1.17	3/0.98	4/1.11	4/1.33	4/1.12	3.1/1.09
	PixLoc [54]CVPR'21	2/0.80	2/0.73	1/0.82	3/0.82	4/1.21	3/1.20	5/1.30	2.9/0.98
SCR	ACE [9]CVPR'23	1.9/0.7	1.9/0.9	0.9/0.6	2.7/0.8	4.2/1.1	4.2/1.3	3.9/1.1	2.8/0.93
	DSAC* [8]PAMI'22	1.9/1.11	1.9/1.24	1.1/1.82	2.6/1.18	4.2/1.41	3.0/1.70	4.2/1.42	2.7/1.41
	HSCNet [37]CVPR'20	2/0.7	2/0.9	1/0.9	3/0.8	4/1.0	4/1.2	3/0.8	2.7/0.90
	HSCNet++ [78]IJCV'24	2/0.63	2/0.79	1/0.8	2/0.65	3/0.85	3/1.09	3/0.83	2.29/0.81
APR	Direct-PN [11]3DV'21	10/3.52	27/8.66	17/13.1	16/5.96	19/3.85	22/5.13	32/10.6	20/7.26
	DFNet [12]ECCV'22	3/1.15	9/3.71	8/6.08	7/2.14	10/2.76	9/2.87	11/5.58	8/3.47
	MAREPO [13]CVPR'24	2.1/1.24	2.3/1.39	1.8/2.03	2.8/1.26	3.5/1.48	4.2/1.71	5.6/1.67	3.2/1.54
MDE	SfMfM (Ours)	2.2/0.77	1.9/0.80	1.1/0.80	3.0/0.91	4.3/1.04	3.7/1.32	2.7/0.78	2.7/0.92

Table 15. **Extended Camera Relocalization Results** on the 7-Scenes dataset [60], with per-scene performance.

per-scene median errors across all seven scenes.

Wayspots Dataset. For the Wayspots dataset [9], we follow the official evaluation protocol by measuring the accuracy of absolute camera pose predictions at multiple error thresholds. Specifically, the predicted pose is considered correct if its translation error is below 10 cm and its rotation error is below 5°, computed with respect to the ground-truth camera pose. For each test sequence, we report the percentage of query images that meet this criterion. The final performance is obtained by averaging the per-sequence accuracies across all test scenes.

7.6. Extended Implementation Details

In both coarse and fine stages, we optimize with Adam [29] for $50,000$ iterations at a learning rate of 1e-3. Within each pair of frames, we sample $\kappa = 200$ pixels. For coarse BA objective Eq. (13), we set maximum logged residual value $\bar{\tau}_{\text{max}} = 10$. For fine BA objective Eq. (5), we set $\tau_{\text{max}} = 20$. We parameterize camera poses following SPARF [69], where rotations are represented using a 6-DoF continuous representation and translations are encoded as 3-DoF vectors. We include image pairs where at least $\nu \geq 15\%$ of the pixels are co-visible. During preprocessing, we sample correspondences only from dense regions where the confidence score exceeds a threshold of $\chi > 0.2$. To compute the probability density function (PDF) and cumulative distribution function (CDF) using a histogram-based kernel density estimation (KDE) algorithm, we use a 1×100 histogram vector. For multi-GPU parallelization, we adopt different strategies in the coarse and fine stages. In the coarse stage, we distribute multiple complete sub-graphs (as shown in Fig. 2) across different GPUs. In the fine stage, we randomly assign frame pairs to different GPUs for processing. We observe that the intrinsic parameters typically converge more slowly than the others; therefore, we increase their learning rate by a factor of 50, *i.e.*, the intrinsic parameters use a learning rate of 5e-2.

Two-View Pose Initialization. We initialize the two-view pose sequentially. First, we estimate the essential matrix using the five-point algorithm [35] within a RANSAC loop on normalized image coordinates, and decompose it to recover the rotation $\mathbf{R}$ and the unit translation direction $\hat{\mathbf{t}}$ with $\|\hat{\mathbf{t}}\| = 1$. Next, we resolve the absolute translation scale using the monocular depth map $\mathbf{D}_i$ of the source frame I_i. For each pixel $p \in I_i$ with depth $d(p) = \mathbf{D}_i[p]$, we use the projection operator $\pi_{i\to j}$ to project it to the target frame I_j for a candidate scale $s > 0$. The projection $\pi_{i\to j}(d(p), s)$ depends on the camera intrinsics $\mathbf{K}_i$, $\mathbf{K}_j$ and the relative pose $\mathbf{R}, s\hat{\mathbf{t}}$. We choose $s(p)$ that minimizes the distance between the projected point $\pi_{i\to j}(d(p), s)$ and the corresponding pixel $v \in I_j$ along the epipolar line defined by (I_i, I_j). Repeating this process for all valid pixels yields a set of per-pixel scale estimates $\{s(p)\}$. The median of these estimates is taken as the final scale $s^{\star}$, forming the initialized two-view pose $\big(\mathbf{R},\ s^{\star}\hat{\mathbf{t}}\big)$.

Camera Intrinsic Initialization. In uncalibrated settings, we initialize camera intrinsics using the DUSt3R [79] pointmap. First, we convert the pointmap into an incidence field following WildCamera [86]: for each pixel $\mathbf{p}$ with corresponding 3D point (x, y, z), we compute the incidence (ray-direction) vector $(x/z, y/z, 1)$. This incidence map encodes the incoming camera-ray direction for each pixel, which, under the pinhole model, depends solely on camera intrinsics and pixel coordinates. We then apply WildCamera's RANSAC-based calibration procedure to recover the 1 DoF intrinsic. We repeat the intrinsic initialization process for each frame. When shared intrinsics are assumed across the collection, we set the initial focal length to the median value across all frames.