Turntable-Constrained Camera Pose Estimation

Supplementary Material

1. Experiment Details

1.1. Synthetic Data

All experiments are conducted on a controlled synthetic dataset generated using a unified parametric orbiting-camera model.

Camera Intrinsics A calibrated pinhole camera with fixed intrinsics is used:

$$\mathbf{K} = \begin{bmatrix} 800 & 0 & 320 \\ 0 & 820 & 240 \\ 0 & 0 & 1 \end{bmatrix}.$$

Camera Motion Model We generate a turntable trajectory with a fixed rotation axis $\mathbf{a} = [0, 0, 1]^\top$ and orbit radius $r = 2.5$. The camera center follows

$$\mathbf{C}(\theta) = r\big(\cos\theta\,\mathbf{b} + \sin\theta\,\mathbf{c}\big),$$

where $(\mathbf{b}, \mathbf{c})$ is an orthonormal basis of the plane orthogonal to $\mathbf{a}$, so the orbit is centered at the world origin. Camera orientation is

$$\mathbf{R}_{cw}(\theta) = \mathrm{Rodrigues}(\mathbf{a}, \theta),$$

hence the initial pose at $\theta = 0$ is identity.

Temporal Sampling We sample $m = 12$ views. Incremental angles are i.i.d.

$$\Delta\theta_i \sim \mathcal{U}(6^\circ, 14^\circ), \quad i = 1, \ldots, m-1,$$

and absolute angles are accumulated as

$$\theta_0 = 0, \quad \theta_i = \sum_{k=1}^{i} \Delta\theta_k.$$

Ground-truth adjacent relative rotations are obtained from consecutive absolute poses.

Scene Geometry We sample $N = 300$ 3D points:

$$x \sim \mathcal{U}(-1.2, 1.2), \quad y \sim \mathcal{U}(-1.0, 1.0), \quad z \sim \mathcal{U}(2.0, 6.0).$$

Image Measurements Each point is projected into every view using $(\mathbf{K}, \mathbf{R}_{cw}, \mathbf{C})$, then perturbed by zero-mean Gaussian pixel noise with standard deviation σ. For stress tests, $\sigma \in \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$. Full point correspondences are retained across all frames.

1.2. Evaluation Metrics

We evaluate rotation accuracy for all experiments, and additionally report trajectory-level metrics when the method induces a camera trajectory. The main tables use slightly different subsets of metrics for rendered-image and synthetic experiments, matching the quantities that are reliably observable in each setting.

Pair Coverage For rendered-image experiments, not all methods recover valid motion for every adjacent image pair. We therefore report *pair coverage*, defined as

$$\mathrm{PairCoverage} = \frac{\#\{\text{adjacent pairs with valid estimate}\}}{\#\{\text{total adjacent pairs}\}}.$$

Higher is better.

Relative Rotation Error For an adjacent pair (i, j), let $R^\star_{ij}$ denote the ground-truth relative rotation and $\hat{R}_{ij}$ the estimate. We measure relative rotation error by the geodesic distance on $\mathrm{SO}(3)$:

$$e^{\mathrm{rel}}_{ij} = \angle\big(\hat{R}_{ij}(R^\star_{ij})^\top\big).$$

For the rendered-image table, we summarize these errors using AUC@5, AUC@10, and AUC@20, i.e. the area under the cumulative error curve up to thresholds 5°, 10°, and 20°, respectively. Higher is better. For the synthetic table, we report the mean relative rotation error in degrees. Lower is better.

Absolute Rotation Error Let $\hat{R}_i$ and $R^\star_i$ denote the estimated and ground-truth absolute rotations for frame i. We report the mean absolute rotation error

$$\frac{1}{m}\sum_{i=1}^{m} \angle\big(\hat{R}_i(R^\star_i)^\top\big),$$

in degrees. Lower is better.

Trajectory Metrics When a method yields a full turntable trajectory, we additionally evaluate the recovered camera path using trajectory-aware metrics.

First, we report the mean translation-direction relative pose error (RPE), computed from consecutive camera displacements:

$$e^{\mathrm{dir}}_i = \arccos\left(\frac{\langle \hat{\mathbf{t}}_i, \mathbf{t}^\star_i \rangle}{\|\hat{\mathbf{t}}_i\|\,\|\mathbf{t}^\star_i\|}\right),$$

where $\hat{\mathbf{t}}_i$ and $\mathbf{t}_i^\star$ are the estimated and ground-truth inter-frame translation vectors after alignment. We report the mean of e_i^{dir} in degrees. Lower is better.

Second, to capture whether the recovered orbit sweeps the correct angular extent, we report the *trajectory coverage error*, defined as the absolute difference between the estimated and ground-truth angular sweep of the camera centers on the turntable plane. Lower is better.

Third, we report the *normalized loop-closure error*, i.e. the discrepancy between the estimated and ground-truth start-to-end closure distance, normalized by the trajectory scale (radius). This penalizes trajectories that fail to close correctly even if they are globally aligned. Lower is better.

1.3. Details of Rendered Turntable Experiments

3D Models. We use five publicly available Creative Commons models obtained from Sketchfab:

- **Shiba**: https://skfb.ly/6WxVW
- **Car**: https://skfb.ly/6sUFy
- **Aeroplane**: https://skfb.ly/6QXS8
- **Laptop**: https://skfb.ly/6RVFt
- **Coffee Maker**: https://skfb.ly/o9PYE

These objects were selected to represent a range of geometric properties, including curved organic shapes (Shiba), rigid mechanical structures (Car, Aeroplane), and partially symmetric industrial objects (Laptop, Coffee Maker).

Figure 3 shows the five models used in our experiments.

Turntable Capture Setup. Each model is placed at the centre of a virtual turntable within Blender. The object rotates about a fixed vertical axis while the camera remains static in the world frame. This configuration produces the same single-axis motion geometry described in Section 3.

We generate image sequences using the BlenderNeRF [17] rendering pipeline. For each object, the turntable animation produces a sequence of **100 images** uniformly sampling a full revolution. The camera intrinsics remain fixed across all views.

Studio Lighting Simulation. To emulate the controlled illumination conditions typical of studio object capture, we illuminate the scene using three diffuse light sources positioned around the object. The lighting configuration is designed to produce approximately uniform illumination while minimising harsh shadows and directional texture cues. This setting intentionally reduces background features and emphasises the object itself, reproducing the feature-poor conditions often encountered in turntable capture scenarios.

1.4. Implementation Details

Experiments were run on an Intel(R) Core(TM) Ultra 7 165U CPU (1.70 GHz) with 16 GB RAM, without GPU acceleration.

Feature Matching. Adjacent-frame correspondences are obtained using ORB features (up to 3000 features by default) and brute-force Hamming matching with cross-check. Pixel coordinates are then back-projected into calibrated normalized image coordinates using the known intrinsic matrix $\mathbf{K}$.

Essential Matrix Estimation. For each adjacent pair we estimate the essential matrix using `cv2.findEssentialMat` with RANSAC (reprojection threshold 1 px, confidence 0.999). We optionally refit $\mathbf{E}$ via the eight-point algorithm using only the inlier correspondences. Relative pose $(\mathbf{R}, \mathbf{t})$ is recovered via cheirality-based selection among the four possible decompositions of $\mathbf{E}$.

Axis Prior. For real sequences, the turntable axis $\hat{\mathbf{a}}$ is pre-computed offline from COLMAP absolute rotations. Specifically, we compute the mean of the per-pair rotation axes, with signs aligned to maintain consistent orientation. The resulting unit vector is stored alongside each dataset and used by all axis-aware methods during inference.

Orbit-Constrained Optimization (A++, A+++). Both methods parameterize the orbit center as

$$\rho = \alpha\hat{\mathbf{b}} + \beta\hat{\mathbf{c}}, \tag{20}$$

where $(\hat{\mathbf{b}}, \hat{\mathbf{c}})$ is an orthonormal basis of the plane perpendicular to $\hat{\mathbf{a}}$.

The A++ formulation jointly optimizes $(\alpha, \beta, \Delta\theta_1, \ldots, \Delta\theta_m)$ by minimizing either Frobenius-norm residuals of $\mathbf{E}$ or Sampson distances. We optionally use a Huber loss (scale 1.0) and an ℓ_2 regularizer on ρ.

The A+++ formulation instead optimizes absolute frame angles $(\theta_1, \ldots, \theta_m)$ directly with a soft prior

$$\lambda_\theta \|\theta_i - \theta_i^0\|^2, \tag{21}$$

where $\lambda_\theta = 10^{-2}$ by default and θ_i^0 denotes the naive initialization.

Both optimization problems are solved using `scipy.optimize.least_squares` with the trust-region reflective (TRF) solver. The maximum number of function evaluations is set to 400 for A++ and 500 for A+++.

Initialization. The initialization (α^0, β^0) for A++ is obtained via a linear null-space solve from the observed $\hat{\mathbf{E}}$ matrices and the initial angle estimates. For A+++, we use a simple default initialization $(\alpha_0, \beta_0) = (1.0, 0.0)$.

Synthetic Data. Synthetic scenes consist of $N = 300$ three-dimensional points uniformly sampled in

$$[-1.2, 1.2] \times [-1.0, 1.0] \times [2.0, 6.0] \text{ meters.}$$

Camera poses follow an exact turntable model with orbit radius $r = 2.5$ m. Per-step angles are drawn independently from $\mathrm{Uniform}(6^\circ, 14^\circ)$ unless otherwise specified in ablation experiments. Pixel noise is modeled as additive i.i.d. Gaussian noise with $\sigma = 0.7$ px. Each noise level is evaluated over 20 independent trials.

Rotation Averaging (Method B). Starting from the A++ pairwise rotations, we perform chordal rotation averaging with iterative reweighting for 120 iterations.

1.5. Prior Experiments: Failures of SfM on Turntable

We show the details of the COLMAP runs used to generate the failure examples shown in the main paper (Sec. 3.3). All experiments are performed on sequences of 100 images rendered from the Blender turntable dataset described in Sec. 5.2. Camera intrinsics are known and fixed across all views.

COLMAP reconstruction pipeline For each sequence we execute the following commands.

```
colmap feature_extractor \
  --database_path $DB/db.db \
  --image_path $DB/images \
  --ImageReader.camera_model PINHOLE \
  --ImageReader.single_camera 1 \
  --ImageReader.camera_params "INTRINSICS"

colmap exhaustive_matcher \
  --database_path $DB/db.db

colmap mapper \
  --database_path $DB/db.db \
  --image_path $DB/images \
  --output_path $DB/sparse \
  --Mapper.ba_refine_focal_length 0 \
  --Mapper.ba_refine_principal_point 0

colmap model_converter \
  --input_path $DB/sparse/0 \
  --output_path $DB/sparse/0 \
  --output_type TXT
```

The `feature_extractor` stage detects ORB-like local features and stores them in the database. We assume a single shared calibrated camera with known intrinsic parameters. The `exhaustive_matcher` then computes pairwise feature matches between all image pairs.

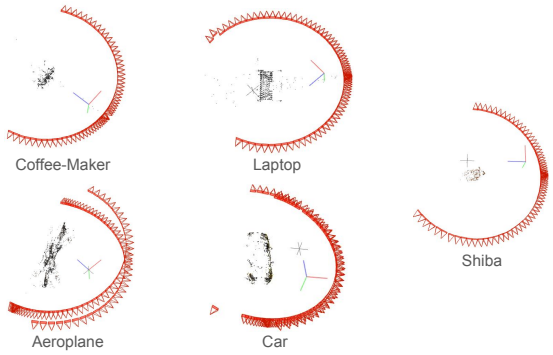

Figure 4. COLMAP reconstructions for the five turntable sequences. Red pyramids indicate estimated camera poses and black points denote the reconstructed sparse structure. Although the ground-truth motion follows a single circular orbit, the recovered trajectories often deviate from this geometry, forming partial arcs or drifting away from a common plane. This illustrates how generic SfM pipelines can produce physically inconsistent trajectories when the single-axis motion prior is not enforced.

The incremental SfM reconstruction is performed with `colmap mapper`. Since the camera intrinsics are known, we disable bundle-adjustment refinement of the focal length and principal point. The resulting sparse reconstruction is finally converted to a text representation for inspection.

Results Figure 4 shows the recovered camera trajectories for the five turntable sequences. Although the ground-truth motion follows a single circular orbit, the reconstructed camera centres frequently deviate from this structure. In several cases the trajectory forms partial arcs or drifts away from a common orbit plane, indicating that the estimated poses are not consistent with a shared rotation axis.

Figure 5 shows a case where COLMAP successfully recovers the orbit trajectory. Here the camera centres form a coherent circular ring, demonstrating that generic SfM can succeed when feature correspondences are sufficiently distinctive. However, the failures in Fig. 4 illustrate that this behaviour is not guaranteed even when the true motion exactly satisfies the single-axis model.

2. Additional Experiments

This section presents additional ablations that help interpret the behaviour of the proposed turntable-aware estimators. Rather than introducing new components, these experiments aim to clarify (i) when sequence-level constraints provide the largest benefit, (ii) how robust the methods are to reduced geometric or temporal evidence, and (iii) whether the strongest sequence-level variant is limited by model expressiveness, initialization, or optimization. Unless otherwise stated, all experiments follow the same datasets, evaluation metrics, and pipelines used in the main

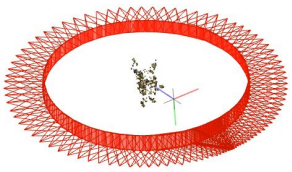

Figure 5. Example where COLMAP successfully reconstructs the turntable trajectory. Left: rendered input views of the object. Right: recovered camera poses forming a coherent circular orbit around the object. This example shows that generic SfM can recover the correct motion when feature correspondences are sufficiently distinctive, although the failures in Fig. 4 demonstrate that such success is not guaranteed.

paper. Synthetic experiments use the controlled single-axis simulator, while rendered-image experiments use the rendered turntable sequences with the same feature extraction and matching pipeline as in the main text.

2.1. Sequence Length and Temporal Sparsity

A central premise of this work is that single-axis motion introduces a *sequence-level* constraint: all camera poses share a common rotation axis and are linked through a single orbit geometry. If this inductive bias is correct, methods that explicitly enforce the shared-axis model should benefit more from additional temporal evidence than unconstrained pairwise estimators.

Synthetic images: sequence length. We first test this hypothesis in the controlled synthetic setting by varying the number of views in the sequence,

$$m \in \{6, 8, 12, 24, 50, 100\},$$

while keeping the random seeds, scene generation, and noise trials fixed. This isolates the effect of trajectory length from other confounding factors. Under this setup, unconstrained pairwise methods are expected to saturate once sufficient views are available for reliable local estimation, whereas constrained estimators can continue to accumulate evidence as additional views reinforce the shared-axis structure.

Real images: temporal sparsity proxy. In rendered-image sequences, varying the total trajectory length alone is less controlled. Instead, we simulate reduced temporal

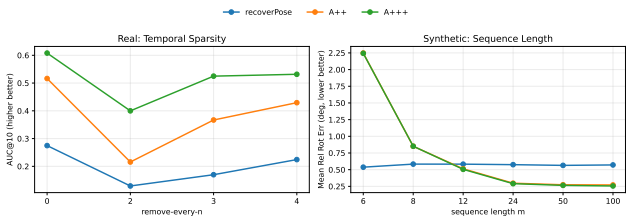

Figure 6. Effect of temporal evidence on pose estimation. Left: rendered-image experiment with increasing temporal sparsity. Right: synthetic experiment with increasing sequence length. Pairwise baselines show limited improvement as additional views become available, whereas turntable-aware estimators benefit more from denser temporal support, particularly when global sequence-level constraints are enforced.

evidence through temporal subsampling, progressively increasing the spacing between retained frames. This preserves the same scenes, camera model, and evaluation pipeline while weakening the temporal support available to the estimator.

Results. The results are summarised in Fig. 6. For synthetic sequences, pairwise baselines show only modest improvement as the number of views increases, whereas constrained variants—especially the globally refined model—benefit more substantially from longer trajectories. This behaviour supports the interpretation that the turntable prior becomes increasingly informative when evidence is aggregated across the entire sequence rather than estimated independently for each pair.

A similar pattern is observed on real images. As temporal sampling becomes sparser, generic baselines degrade more noticeably, while turntable-aware estimators remain comparatively stable. Because the shared-axis model couples all views through a common low-dimensional motion family, it continues to regularise the solution even when intermediate observations are removed.

2.2. Correspondence Support

We next study robustness as the amount of available correspondence evidence decreases. Both experiments probe the same underlying factor—the strength of geometric constraints derived from feature correspondences—but control it differently depending on the data regime.

Synthetic control. In the synthetic setting, the correspondence count can be controlled directly by varying the number of tracked points per trial,

$$N \in \{300, 150, 75\},$$

while keeping all other factors fixed. This provides a clean test of how estimation accuracy changes as geometric sup-

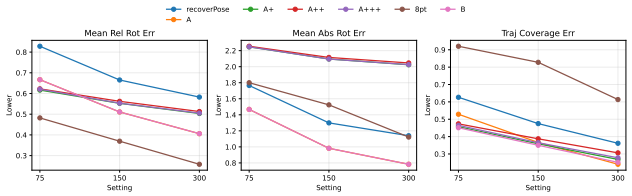

Figure 7. Synthetic correspondence-support ablation. The number of tracked correspondences is varied while all other factors remain fixed. All methods degrade as geometric evidence weakens, but turntable-aware estimators remain more stable than pairwise baselines, particularly in trajectory-level metrics.

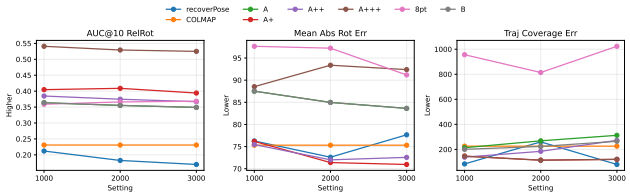

Figure 8. Rendered-image correspondence-support ablation using feature budget as a proxy. The number of detected features is varied while keeping the scene and evaluation pipeline fixed. Reducing the feature budget weakens geometric evidence and degrades performance for all methods. However, turntable-aware estimators remain comparatively stable, particularly in trajectory-level metrics, indicating that the shared-axis constraint helps regularise estimation when local correspondence support is limited.

port weakens.

Figure 7 shows that all methods degrade as the number of correspondences decreases, reflecting the increasing instability of pairwise geometry estimation. However, the constrained variants degrade more gracefully than unconstrained pairwise baselines, particularly in trajectory-level metrics. This behaviour is consistent with the intuition that the turntable prior reduces the effective solution space: even when individual pairs provide weaker evidence, the shared-axis constraint couples the sequence and suppresses geometrically plausible but physically inconsistent solutions.

Rendered-image proxy. On real images, correspondence support cannot be controlled directly. Instead, we vary the feature detection budget, which indirectly affects the number of usable matches. This provides a practical proxy for reducing correspondence density while keeping the scenes, camera model, and evaluation pipeline fixed.

Figure 8 shows a similar trend to the synthetic experiment. As the feature budget decreases, performance degrades for all methods due to weaker geometric evidence. However, the constrained variants remain comparatively stable, particularly in trajectory-level metrics. In contrast, pairwise baselines exhibit larger fluctuations as the available matches become sparse. This behaviour suggests that the shared-axis constraint helps stabilise estimation by coupling information across the sequence, partially compensating for weaker local correspondence support.

2.3. Oracle Gap and Initialization Ceiling

To distinguish between modelling limitations and initialization limitations, we compare the practical variants against oracle-initialized counterparts on the rendered-image benchmark. The oracle settings replace the estimated axis, or both the axis and angular variables, with ground-truth values before running the corresponding downstream stage. For the rendered-image benchmark, the ground-truth axis is obtained from the provided absolute camera poses in `transforms_test.json` (by NerfBlender [17]): we form adjacent relative rotations, extract their rotation axes, align their signs consistently, and average them to obtain a single shared turntable axis. When oracle angular variables are used, the corresponding ground-truth angular increments are computed by measuring each adjacent ground-truth relative rotation about this ground-truth axis.

The results in Table 3 reveal a highly uneven oracle gap. The difference between A^+ and its oracle-axis counterpart is small, suggesting that axis estimation alone is not the dominant source of error. In contrast, providing both the correct axis and rotation angles dramatically improves performance. This indicates that the main limitation of the practical pipeline is not the expressive power of the turntable-constrained model, but the quality of the initialization supplied to the coupled optimisation. Once the optimisation starts sufficiently close to the correct basin, the structured model is able to recover nearly perfect trajectories.

2.4. Sensitivity to Estimation Presets

We evaluate four estimator presets to check whether conclusions depend on a single tuning choice. **Vanilla** keeps permissive gating (minimal filtering), **Loose** applies mild pair filtering, **Baseline** uses moderate filtering/regularization, and **Strict** applies the strongest filtering. These presets mainly control which adjacent pairs are retained for downstream estimation.

Across presets, the absolute numbers change, but the key trend is consistent: stricter gating can reduce error on retained pairs while reducing usable pair coverage. Hence, we interpret this study as a coverage–accuracy trade-off rather than a different method ranking under fundamentally new conditions. The main results therefore report *vanilla* as the default operating point, which preserves high coverage while maintaining stable trajectory behavior.

Table 3. Rendered-image oracle gap ablation, macro-averaged over datasets. Oracle variants use ground-truth axis and/or rotation angles during initialization. Bold indicates the best overall result, and underlined values denote the best non-oracle baseline.

Method	AUC@10	Mean Abs Rot Err	Traj Cov Err	Loop Err/R
A+	0.386	<u>74.86</u>	115.52	<u>1.439</u>
A++	0.361	76.30	241.96	1.650
A+++	<u>0.532</u>	83.66	<u>114.16</u>	1.447
A+ oracle-axis	0.386	75.18	118.01	1.482
A++ oracle-axis	0.362	76.03	237.80	1.674
A++ oracle-axis+theta	**0.996**	**0.18**	**9.38**	**0.158**

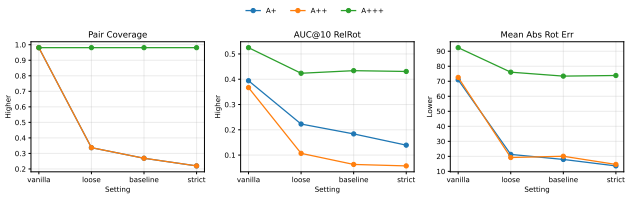

Figure 9. Rendered-image preset sensitivity for turntable-aware variants (**A+**, **A++**, **A+++**). From left to right: pair coverage, AUC@10, and mean absolute rotation error. Changing presets primarily changes gating strictness: stricter settings keep fewer pairs, which can lower error on retained pairs but usually reduce overall coverage.

Figure 10. Initialisation and optimisation sensitivity of the global orbit refinement (A^{+++}). Axis, angle, and orbit parameters are perturbed with increasing jitter magnitude, and the optimisation budget is varied independently. Across all sweeps, the evaluation metrics remain nearly unchanged, indicating that the optimisation is locally robust and not budget-limited.

2.5. Robustness of Global Orbit Refinement

The strongest sequence-level variant, A^{+++}, performs joint optimisation over a low-dimensional orbit parameterisation. A natural question is whether its performance is limited by the quality of the initialisation or by insufficient optimisation effort.

To evaluate this, we conduct two complementary sweeps. First, we perturb the initial parameters of the orbit model, modifying one variable family at a time: the shared rotation axis, the per-view rotation angles, or the orbit vector. For each parameter type we apply increasing perturbations with magnitudes

$$\{0, 0.5, 1, 2, 5\}.$$

Second, we vary the optimisation budget to test whether additional iterations improve the final solution.

Figure 10 summarises the results. Across all perturbation types and magnitudes, the evaluation metrics change only marginally. Neither axis perturbations nor moderate angle or orbit perturbations produce meaningful degradation in AUC@10, mean rotation error, or trajectory coverage error. Similarly, increasing the optimisation budget has little effect on the final metrics, indicating that the optimisation converges quickly and does not benefit from additional iterations.

Taken together, these results suggest that the refinement stage is locally stable within the tested perturbation range and is not limited by optimisation budget. Instead, the main challenge appears to be obtaining an initialisation that lies sufficiently close to the correct basin of attraction. This interpretation is consistent with the oracle-gap analysis in the previous section.

2.6. Baseline Regime Ablation

We next examine how the effectiveness of the turntable prior depends on the baseline regime. In the synthetic benchmark, we vary the range of adjacent rotation increments to generate small-, medium-, and large-baseline settings:

$$U(2, 6), \qquad U(6, 14), \qquad U(12, 24).$$

Figure 11 summarises the results. When the angular increments are small, pairwise geometry becomes more ambiguous and generic baselines degrade significantly. In this regime, the turntable-aware estimators benefit most from the shared-axis constraint, which injects global geometric structure when local evidence is weak. As the baseline increases, relative pose estimation becomes easier for all methods, and the advantage of stronger sequence-level constraints gradually narrows.

2.7. Full table for Rendered Turntable Experiments

Table 4 reports the full per-dataset results for the rendered turntable experiments presented in the main paper. In addi-

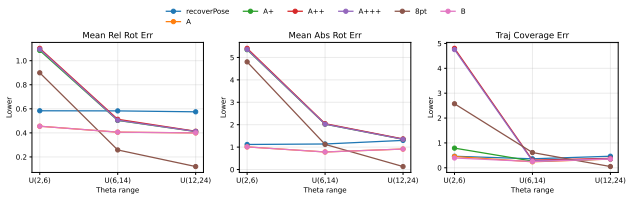

Figure 11. Angular increment range ablation on synthetic data. The range of adjacent rotation increments controls the effective baseline between views. Small-baseline regimes make pairwise geometry more ambiguous, where turntable-aware estimators provide the largest benefit. As the baseline increases, generic methods improve and the performance gap between constrained and unconstrained approaches becomes smaller.

tion to the averaged results shown in the main text, we provide metrics for each object sequence, including pair coverage, rotation accuracy, and trajectory-level errors. This detailed breakdown helps illustrate how the different estimators behave across objects with varying geometry and symmetry.

Dataset	Method	Cov.	AUC_{10} RelRot	RelRot	CovErr	RPE_t^{dir}	LoopErr
	A	**1.000**	0.444	6.812	127.670	50.372	1.759
	A+	**1.000**	0.470	6.541	156.350	64.436	1.653
	8pt	**1.000**	0.366	43.216	1666.226	68.791	0.607
	recoverPose	0.924	0.186	11.038	**27.494**	**6.917**	0.404
aeroplane	COLMAP	**1.000**	0.199	15.147	360.000	79.913	**0.000**
	A++	**1.000**	0.448	6.741	163.884	97.241	1.592
	A+++	**1.000**	**0.584**	**4.740**	155.826	69.821	1.704
	B	**1.000**	0.444	6.812	149.621	75.250	1.801
	C	**1.000**	0.177	12.193	180.513	21.527	1.881
	A	**1.000**	0.383	7.504	104.821	41.605	1.686
	A+	**1.000**	0.417	7.154	135.452	53.318	1.827
	8pt	**1.000**	0.327	28.480	1443.009	51.958	0.015
	recoverPose	0.909	0.139	11.365	**20.503**	**5.821**	0.307
car	COLMAP	**1.000**	0.200	15.864	360.000	74.357	**0.000**
	A++	**1.000**	0.401	7.316	154.590	83.011	1.726
	A+++	**1.000**	**0.634**	**3.710**	137.981	61.165	1.846
	B	**1.000**	0.383	7.504	139.542	63.610	1.841
	C	**1.000**	0.166	12.873	171.264	29.757	1.732
	A	**1.000**	0.332	9.473	**68.465**	**14.056**	1.019
	A+	**1.000**	0.353	9.034	96.905	23.232	1.476
	8pt	**1.000**	0.359	18.877	322.952	41.334	**0.266**
	recoverPose	0.788	0.141	11.693	69.202	16.754	1.093
coffee_maker	COLMAP	0.682	0.304	8.581	114.599	85.515	1.602
	A++	**1.000**	0.339	9.238	120.851	65.091	1.764
	A+++	**1.000**	**0.601**	**4.300**	86.727	44.928	1.379
	B	**1.000**	0.332	9.473	97.658	23.595	1.491
	C	**1.000**	0.337	13.622	941.868	99.459	0.570
	A	**0.924**	0.165	23.643	898.685	49.519	1.273
	A+	**0.924**	0.255	17.203	**65.972**	73.768	1.187
	8pt	**0.924**	**0.401**	12.183	925.599	116.409	0.981
	recoverPose	0.909	0.243	12.976	240.691	70.025	**0.627**
laptop	COLMAP	0.818	0.254	**12.041**	86.056	53.353	1.472
	A++	**0.924**	0.201	20.891	609.081	40.327	1.622
	A+++	**0.924**	0.287	17.536	68.749	82.272	1.201
	B	**0.924**	0.165	23.643	671.247	**30.915**	0.722
	C	**0.924**	0.140	16.278	517.161	57.988	0.709
	A	**1.000**	0.404	6.991	136.270	56.210	1.850
	A+	**1.000**	0.420	6.602	156.773	82.242	1.672
	8pt	**1.000**	0.380	8.006	61.679	**19.218**	0.814
	recoverPose	0.788	0.111	13.053	**30.451**	19.898	0.333
shiba	COLMAP	0.788	0.271	11.074	97.462	62.366	1.587
	A++	**1.000**	0.418	6.569	162.351	76.507	1.531
	A+++	**1.000**	**0.583**	**4.222**	155.566	87.185	1.706
	B	**1.000**	0.404	6.991	157.285	82.149	1.674
	C	**1.000**	0.206	15.359	376.005	63.242	**0.097**
	A	**0.985**	0.345	10.885	267.182	42.352	1.518
	A+	**0.985**	0.383	9.307	122.290	59.399	1.563
	8pt	**0.985**	0.367	22.152	883.893	59.542	**0.537**
	recoverPose	0.864	0.164	12.025	**77.668**	**23.883**	0.553
Avg	COLMAP	0.858	0.245	12.541	203.623	71.101	0.932
	A++	**0.985**	0.361	10.151	242.151	72.436	1.647
	A+++	**0.985**	**0.538**	**6.902**	120.970	69.074	1.567
	B	**0.985**	0.345	10.885	243.071	55.104	1.506
	C	**0.985**	0.205	14.065	437.362	54.395	0.998

Table 4. Per-dataset results and overall average on rendered-image SfO benchmarks. Higher is better for coverage and AUC; lower is better for errors. Coverage error is computed on absolute angular sweep (direction-agnostic).

References

[1] Eric Brachmann, Alexander Krull, Sebastian Nowozin, Jamie Shotton, Frank Michel, Stefan Gumhold, and Carsten Rother. Dsac-differentiable ransac for camera localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6684–6692, 2017. 1

[2] Ethan Elms, Yasir Latif, Tae Ha Park, and Tat-Jun Chin. Event-based structure-from-orbit. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19541–19550, 2024. arXiv:2405.06216. 1, 2, 3

[3] Andrew W. Fitzgibbon, Geoff Cross, and Andrew Zisserman. Automatic 3D model construction for turn-table sequences. In *3D Structure from Multiple Images of Large-Scale Environments*, pages 155–170. Springer, 1998. 1, 2, 3

[4] Vincent Frémont and Ryad Chellali. Direct camera calibration using two concentric circles from a single view. In *Proceedings of the 12th International Conference on Artificial Reality and Telexistence (ICAT)*, 2002. December 4–6, Tokyo, Japan. 2

[5] Vincent Frémont and Ryad Chellali. Turntable-based 3D object reconstruction. In *Proceedings of the 2004 IEEE Conference on Cybernetics and Intelligent Systems (CIS)*, pages 1277–1282, 2004. 2

[6] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003. 1

[7] Richard Hartley, Jochen Trumpf, Yuchao Dai, and Hongdong Li. Rotation averaging. *International Journal of Computer Vision*, 103(3):267–305, 2013. 2, 3, 5

[8] Richard I Hartley. In defense of the eight-point algorithm. *IEEE Transactions on pattern analysis and machine intelligence*, 19(6):580–593, 1997. 4

[9] Joao F Henriques and Andrea Vedaldi. Mapnet: An allocentric spatial memory for mapping environments. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8476–8484, 2018. 1

[10] Guang Jiang, Hung-Tat Tsui, Long Quan, and Andrew Zisserman. Single axis geometry by fitting conics. In *Computer Vision – ECCV 2002*, pages 537–550. Springer, 2002. 2

[11] Guang Jiang, Hung-Tat Tsui, Long Quan, and Andrew Zisserman. Geometry of single axis motions using conic fitting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(10):1343–1348, 2003. 2

[12] Sing Bing Kang. Quasi-euclidean recovery from unknown but complete orbital motion. Technical Report CRL 97/10, Digital Equipment Corporation, Cambridge Research Laboratory, 1997. 2

[13] Alex Kendall, Matthew Grimes, and Roberto Cipolla. Posenet: A convolutional network for real-time 6-dof camera relocalization. In *Proceedings of the IEEE international conference on computer vision*, pages 2938–2946, 2015. 1

[14] Vincent Lepetit, Francesc Moreno-Noguer, and Pascal Fua. Ep n p: An accurate o (n) solution to the p n p problem. *International journal of computer vision*, 81(2):155–166, 2009. 1

[15] David Nistér. An efficient solution to the five-point relative pose problem. *IEEE transactions on pattern analysis and machine intelligence*, 26(6):756–770, 2004. 1

[16] Onur Özyeşil, Vladislav Voroninski, Ronen Basri, and Amit Singer. A survey of structure from motion. *Acta Numerica*, 26:305–364, 2017. 2

[17] Maxime Raafat. Blendernerf, 2024. 2, 5

[18] Harpreet S. Sawhney, John Oliensis, and Allen R. Hanson. Image description and 3-D reconstruction from image trajectories of rotational motion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(9):885–898, 1993. 2

[19] Johannes L. Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4104–4113, 2016. 2

[20] Richard Szeliski. Shape from rotation. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 625–631, 1991. 2