Seeing Down the Line: Endoscopic Reconstruction with Centerline Constraints

Andrea Dunn Beltran[1] ASDUNNBE@CS.UNC.EDU
[1] *The University of North Carolina, Department of Computer Science, Chapel Hill, NC, USA*

Romain Hardy[2] ROMAIN_HARDY@FAS.HARVARD.EDU
Pranav Rajpurkar[2] PRANAV_RAJPURKAR@HMS.HARVARD.EDU
[2] *Harvard University, Department of Biomedical Informatics, Boston, MA, USA*

Editors: Under Review for MIDL 2026

Abstract

Colonoscopy remains the gold standard for colorectal cancer screening, but there is still no real-time, geometry-aware way to quantify which parts of the colon have been inspected during a procedure. We revisit 3D Gaussian endoscopic reconstruction as a representation and geometry problem rather than a new network design. Assuming known camera poses and off-the-shelf depth or photometric supervision, we add a simple centerline-based coordinate system and priors on top of an existing Gaussian mapping backbone. From the noisy pose stream we maintain an online centerline and Bishop frame, assign each Gaussian tubular coordinates (s, r, θ), and use these coordinates both to regularize the map toward a hollow tube and to accumulate coverage statistics in colon-intrinsic space. On long C3VD phantom colonoscopy sequences, this lightweight modification achieves Chamfer distance comparable to or better than an endoscopy-specific 3D Gaussian SLAM baseline while running at frame rates close to MonoGS and yielding improved rendering quality, with negligible additional computation. At the same time, the same representation produces unrolled colon views and segment-wise coverage summaries essentially "for free", making centerline-aware Gaussian mapping a practical drop-in component for future real-time quality monitoring tools in colonoscopy.

Keywords: Endoscopy, 3D Recontruction, Coverage

1. Introduction

Colorectal cancer is a leading cause of cancer-related death, yet it is highly preventable when precancerous polyps are detected and removed early (Siegel et al., 2024). Colonoscopy is the gold standard for screening and therapy (Rex et al., 2015), but tandem studies still report substantial adenoma miss rates, especially for flat and serrated lesions hidden behind folds (Zhao et al., 2019). Current quality indicators such as adenoma detection rate and withdrawal time have improved practice (Kaminski et al., 2010), but they are coarse and retrospective: they do not say which parts of the colon were actually inspected and how well during a particular procedure.

At the same time, 3D reconstruction methods for endoscopy have matured. Recent 3D Gaussian SLAM systems can build detailed maps from monocular endoscopic video (Matsuki et al., 2024; Wang et al., 2024), and they are increasingly considered as building blocks

for navigation and documentation. However, most systems are optimized either for general scenes or for short surgical clips, and treat the reconstruction as an end in itself. They do not explicitly encode colon anatomy or provide real-time coverage summaries that match how endoscopists report procedures (segment, insertion depth, circumferential position). In practice, the cost of adding such structure is a concern: clinical systems will favor small, robust changes over entirely new networks.

We revisit endoscopic 3D Gaussian mapping from this angle. Assuming that camera poses are provided by an external tracker or SLAM system and that depth or photometric supervision is available, we ask: *How much benefit can we obtain from a minimal geometric modification to an existing system?* Our answer is to keep the backbone unchanged and add a simple centerline-based coordinate system and priors. From the noisy pose stream we maintain an online colon centerline and Bishop frame, assign each Gaussian tubular coordinates (s, r, θ), and use these coordinates to (i) regularize the Gaussians toward a hollow tube around the centerline and (ii) accumulate online coverage statistics in colon-intrinsic space. This requires some additional geometry and bookkeeping, but only modest extra computation.

On long C3VD phantom colonoscopy sequences (Bobrow et al., 2023), this lightweight modification yields a useful trade-off: it matches or improves the Chamfer distance of EndoGSLAM (Wang et al., 2024), runs at effective frame rates close to a MonoGS-style mapper (Matsuki et al., 2024), and provides better held-out renderings, while simultaneously producing unrolled colon views and segment-wise coverage summaries "for free." We do not claim a fundamentally new representation or learning paradigm; rather, we show that making colon anatomy a first-class constraint in an otherwise standard Gaussian mapping pipeline gives clinically relevant outputs at minimal incremental cost.

Our contributions are:

- **Centerline-aware Gaussian mapping.** We extend a standard 3D Gaussian mapper with an online colon centerline and Bishop frame, assigning each Gaussian tubular coordinates without changing the underlying rendering or optimization machinery.
- **Cheap colon-specific priors and coverage.** Using these coordinates, we add simple tube and smoothness priors that keep Gaussians near the mucosal surface and accumulate per-segment coverage statistics in real time, yielding unrolled coverage maps essentially for free.
- **Evaluation on long, physician recorded sequences.** On C3VD, our method achieves EndoGSLAM level Chamfer distance with MonoGS-like frame rates and better rendering quality, while providing online coverage maps and segment summaries not available from either baseline.

2. Related work

Per-frame assistance and quality assessment. Deep learning has enabled real-time computer-aided detection and diagnosis in colonoscopy, with systems that highlight polyps, classify lesion types, and estimate per-frame quality scores (**??**). Large datasets such as HyperKvasir (Borgli et al., 2020) have supported multi-class lesion detection and automated quality assessment, including withdrawal speed and bowel preparation (**?**). These approaches operate mainly in image space, reasoning over individual frames or short clips.

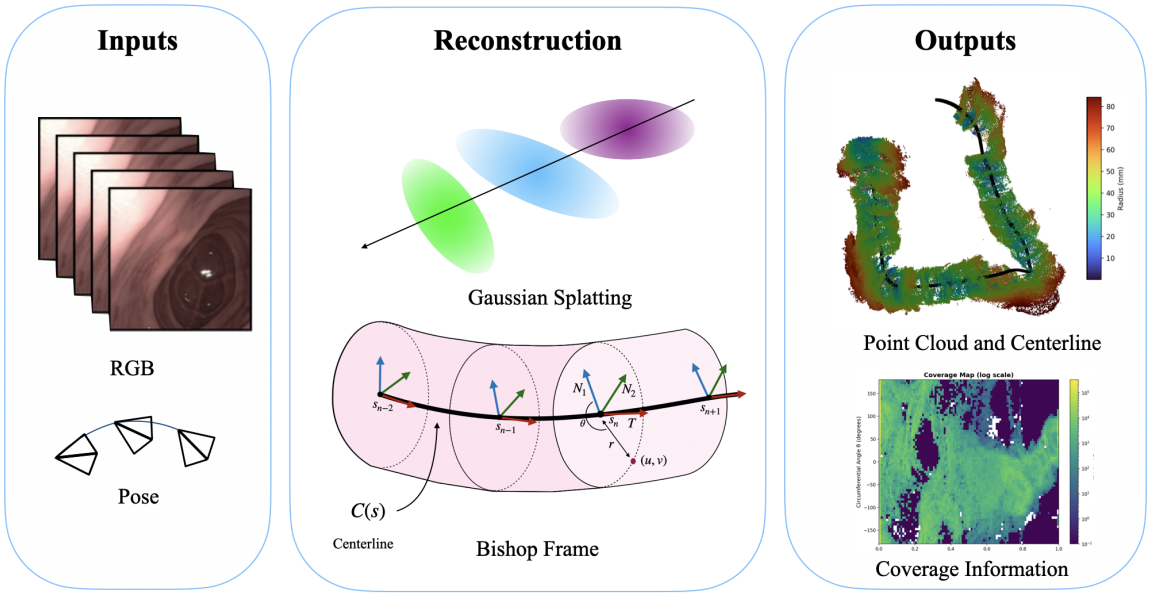

Figure 1: Overview of our centerline-aware 3D Gaussian mapping pipeline. **Inputs:** monocular colonoscopy RGB frames and externally provided poses. **Reconstruction:** a thin geometric layer around a standard 3D Gaussian mapper maintains an online centerline $C(s)$ and Bishop frame, assigns tubular coordinates (s, r, θ), and updates keyframes and coverage counters in colon-intrinsic space. **Outputs:** a 3D Gaussian reconstruction with centerline and online coverage maps in (s, θ), providing segment-wise coverage summaries with minimal extra computation.

They do not maintain a persistent 3D representation of the colon or provide explicit, geometry-based coverage measures.

Dense SLAM and endoscopic reconstruction. Dense SLAM has evolved from volumetric fusion to neural implicit and 3D Gaussian representations (Zhu et al., 2022; **?**; Keetha et al., 2024). Endoscopic variants adapt these ideas to deformable and specular anatomy. RNNSLAM couples a recurrent depth-and-pose network with a SLAM backend for colon reconstruction (Ma et al., 2021), while EndoGSLAM integrates 3D Gaussian splatting into endoscopic surgery, demonstrating real-time tracking and dense mapping (Wang et al., 2024). The C3VD dataset provides phantom colonoscopy videos with depth and ground-truth geometry, enabling quantitative evaluation (Bobrow et al., 2023). These systems focus on geometry and tracking accuracy; they typically treat the colon as a generic scene and do not organize the map in colon-intrinsic coordinates or expose coverage metrics as first-class outputs. Our work builds directly on this line, but holds pose fixed and instead modifies the representation and loss to be colon-aware.

Tubular and anatomical priors. Tubular priors are widely used to model elongated anatomical structures such as vessels and airways (Bauer and Bischof, 2009; Chlebiej et al., 2023). For the colon, prior work has explored digital "unrolling" for visualization and

registration (Rossides et al., 2021) and tubular non-rigid structure-from-motion models for colonoscopic video (Sengupta et al., 2021; Floor et al., 2022). These methods show the value of encoding tube-like anatomy, but are typically offline and not framed as small modifications to existing real-time reconstruction pipelines. In contrast, we integrate a simple tubular coordinate system and associated priors directly into a Gaussian mapper, and show that this modest geometric addition is enough to match strong baselines in geometry while providing online coverage information at negligible extra cost.

3. Method

Our only modification to a standard 3D Gaussian mapper is to wrap it in a colon-intrinsic coordinate system based on an online centerline and its Bishop frame. This provides a natural parameter s for insertion depth, tubular coordinates (s, r, θ) for each Gaussian, and cheap coverage statistics in the same space. All rendering, networks, and optimizers are unchanged from a MonoGS-style backbone; implementation details and hyperparameters are given in Appendix A. Our key intuition is that this thin geometric layer is sufficient to unlock both better use of Gaussian capacity and coverage-aware outputs at minimal additional cost.

3.1. Gaussian map and reconstruction loss

Let $\{T_t\}_{t=1}^{N} \subset SE(3)$ denote per-frame camera poses and $\{I_t\}_{t=1}^{N}$ the corresponding RGB images. We maintain a set of 3D Gaussian primitives $\{g_i\}_{i=1}^{M}$, where each g_i has mean $\mu_i \in \mathbb{R}^3$, anisotropic covariance S_i, color, and opacity. A differentiable Gaussian rasterizer projects $\{g_i\}$ into keyframe views T_k to produce rendered images $\hat{I}_k$ and, where available, depth maps $\hat{D}_k$. Using observed keyframe RGB I_k and depth D_k, we adopt standard photometric and depth reconstruction losses

$$\mathcal{L}_{\text{photo}} = \sum_{k \in \mathcal{K}} \sum_{p \in \Omega_k} \left\| I_k(p) - \hat{I}_k(p) \right\|_1, \qquad \mathcal{L}_{\text{depth}} = \sum_{k \in \mathcal{K}} \sum_{p \in \Omega_k} \left\| D_k(p) - \hat{D}_k(p) \right\|_1, \tag{1}$$

where $\mathcal{K}$ is the set of keyframes and Ω_k the valid pixels. We do not introduce new networks or detectors; all supervision comes from existing tracking and depth components.

Front-end / back-end separation. The system is split into a light *front-end* and a heavier *back-end*:

- **Front-end.** Processes every frame, maintains an online centerline $C(s)$, its Bishop frame, an arc-length-based keyframe schedule, and low-cost coverage counters in colon coordinates. This runs at the native video frame rate and can provide insertion depth and coarse coverage in real time.
- **Back-end.** Operates only on keyframes. Given the current centerline, it assigns tubular coordinates to Gaussians, evaluates colon priors, and updates the Gaussian map using the combined loss. This mirrors the mapping component of MonoGS-style 3D Gaussian reconstruction, but treats poses as fixed and injects colon-aware geometric constraints.

This separation keeps the extra computation marginal while allowing the Gaussian optimizer to focus on fewer, well-chosen views.

3.2. Online centerline and Bishop frame

Our goal is to convert noisy colonoscopy motion into a smooth, 1D backbone that approximates the scope path and serves as a clinically meaningful coordinate for insertion depth and segment location.

Backbone construction. We represent the exploration path by a centerline curve $C(s) \in \mathbb{R}^3$ parameterized by arc length s. Online, we maintain a sparse set of *backbone points* $\{b_k\}_{k=1}^{K}$ derived from camera centers x_t. A new backbone point is added when (i) the translational motion since b_K exceeds $d_{\min}$, (ii) the motion direction does not exceed a maximum bend angle, and (iii) the candidate is not within d_{loop} of non-neighbor backbone points (loop avoidance). This filters small jitter while following the exploration path; full thresholds are listed in Appendix A.

B-spline smoothing and arc length. Given $\{b_k\}$, we fit a cubic B-spline $\tilde{C}(u)$, $u \in [0, 1]$, and sample it at $\{u_m\}_{m=0}^{M}$. We compute cumulative distances $s_0 = 0$, $s_m = s_{m-1} + \|\tilde{C}(u_m) - \tilde{C}(u_{m-1})\|_2$, store positions $C_m = \tilde{C}(u_m)$ and arc-lengths s_m, and query intermediate points by linear interpolation in s. The coordinate s is therefore a smoothed approximation of physical insertion depth, more useful for documentation and coverage than frame index or raw Euclidean distance.

Bishop frame. To define a stable tubular coordinate system we compute a Bishop frame $\{T(s), N_1(s), N_2(s)\}$ along C. For each sample C_m we estimate the tangent T_m by finite differences and propagate an orthonormal pair of normals $(N_{1,m}, N_{2,m})$ using discrete parallel transport, i.e. the minimal rotation taking T_{m-1} to T_m followed by re-orthogonalization. This yields an orthonormal basis with minimal twist at each s_m, and is numerically stable even in nearly straight segments where a Frenet frame would be ill-conditioned.

3.3. Colon-intrinsic coordinates

The Bishop frame turns the colon into a tubular coordinate system. Given a 3D point x (e.g. a Gaussian center μ_i), we first find its closest point on the centerline by minimizing $\|x - C(s)\|_2$ over sampled $\{s_m\}$ and, optionally, refining with a 1D line search, obtaining $s^\star$ and $C^\star = C(s^\star)$. We then query the Bishop frame at $s^\star$, $T^\star, N_1^\star, N_2^\star$, and express the offset $\Delta x = x - C^\star$ as

$$u = \Delta x \cdot N_1^\star, \quad v = \Delta x \cdot N_2^\star, \quad \ell = \Delta x \cdot T^\star.$$

The radial distance and circumferential angle are

$$r(x) = \sqrt{u^2 + v^2}, \qquad \theta(x) = \operatorname{atan2}(v, u),$$

so that $(s^\star, r(x), \theta(x))$ defines colon-intrinsic coordinates for x. We use these coordinates for keyframe selection, colon-aware priors, and coverage accumulation. Clinically, $s^\star$ aligns with insertion depth and segment, while $\theta(x)$ captures circumferential location on the mucosal surface.

3.4. Arc-length-based keyframing

Standard keyframe selection thresholds on Euclidean camera motion or image overlap, which oversamples straight segments and undersamples tight bends in a tubular organ. We instead select keyframes by distance traveled along the centerline.

At frame t with camera center x_t, we estimate the arc-length increment Δs_t since frame $t-1$: if the centerline already covers both positions, we project x_{t-1} and x_t onto C to obtain s_{t-1} and s_t and set $\Delta s_t = |s_t - s_{t-1}|$; otherwise, while the centerline is still growing, we approximate forward progress using the tangent at the endpoint and the signed projection of $(x_t - x_{t-1})$ onto that tangent, clipped at zero. We maintain an accumulator $A_t = A_{t-1} + \Delta s_t$ with $A_0 = 0$ and create a new keyframe when standard appearance- and time-based criteria are met and $A_t \geq \tau_{\mathrm{KF}}$, where τ_{KF} is the desired spacing in millimeters. The accumulator is then reset. This yields approximately uniform keyframe density in s and automatically densifies keyframes in high-curvature segments where coverage is more challenging.

3.5. Colon-aware priors and total loss

Once tubular coordinates are defined, we use them to regularize the reconstruction toward an anatomically plausible hollow tube with a smooth backbone. This ties the Gaussian map to colon geometry and reduces the search space for the optimizer.

Radial tube prior. The colon wall is approximately tubular around the centerline at a characteristic radius R_{wall}. For each Gaussian i with center x_i we compute its radius $r_i = r(x_i)$ and penalize deviations from R_{wall} via

$$\mathcal{L}_{\mathrm{radial}} = \sum_i w_i \, \rho\big((r_i - R_{\mathrm{wall}})^2\big), \tag{2}$$

where $\rho(\cdot)$ is a robust penalty and w_i is an optional weight (e.g. down-weighting low-coverage or highly specular regions). This encourages Gaussians to concentrate near the mucosal surface instead of filling the lumen interior.

Centerline smoothness prior. To avoid spurious kinks from noisy motion, we regularize the discrete second derivative of the sampled centerline positions $\{C_m\}$:

$$\mathcal{L}_{\mathrm{curv}} = \sum_{m=1}^{M-1} \big\| C_{m+1} - 2C_m + C_{m-1} \big\|_2^2. \tag{3}$$

This penalizes rapid curvature changes while allowing anatomically plausible bending and yields a reliable backbone for reporting and unrolling.

Total loss. The back-end minimizes a combination of photometric, geometric, and colon-aware terms:

$$\mathcal{L} = \mathcal{L}_{\mathrm{photo}} + \mathcal{L}_{\mathrm{depth}} + \lambda_{\mathrm{radial}} \mathcal{L}_{\mathrm{radial}} + \lambda_{\mathrm{curv}} \mathcal{L}_{\mathrm{curv}}, \tag{4}$$

with scalar weights $\lambda_{\mathrm{radial}}$ and λ_{curv}. The same tubular coordinates used to define these priors are also used to compute unrolled coverage maps and segment-wise coverage statistics in Section 4, so the representation that improves geometry and efficiency also directly exposes clinically meaningful readouts.

4. Experiments

We evaluate whether a thin layer of colon-aware geometry on top of a standard 3D Gaussian mapper can (i) match the geometric accuracy of an endoscopy-specific 3DGS baseline, (ii) retain the frame rates of a MonoGS-style mapper, and (iii) expose useful online coverage information essentially for free.

4.1. Datasets and protocol

We use the four screening colonoscopy videos from the C3VD phantom dataset recorded by practicing gastroenterologists (**?**). Each sequence contains 4,700–5,500 frames and is accompanied by ground truth camera poses, RGB video, and a watertight CAD mesh of the colon mold. We use per-frame monocular depth predictions from (Hardy et al., 2025) as supervision for all methods and resize images to 384×384.

To ensure a fair and tractable comparison, all methods are evaluated under the same "every-other-frame" protocol, yielding an effective input rate of 15 fps. Every 8th processed frame is held out as a validation frame; the remainder are used for optimization. We run per-scene optimization over the entire sequence and evaluate online: at each processed frame, the current map is used to render all held-out frames seen so far. Additional low-level details (batch sizes, optimizer settings) are given in Appendix B.

4.2. Baselines

We compare against two 3D Gaussian mapping back-ends:

EndoGSLAM. EndoGSLAM is an endoscopy-specific 3D Gaussian SLAM system originally evaluated on short, robot-acquired C3VD sequences (Wang et al., 2024). For our long phantom sequences we disable tracking and pose refinement and feed ground-truth poses, so only the mapping component is active.

MonoGS-style mapper. The MonoGS baseline follows a standard 3D Gaussian mapping pipeline without colon-specific structure (**?**). It uses similar depth supervision and rendering losses as our method, but does not estimate a centerline, does not use tubular coordinates, and employs a conventional frame-based keyframe policy.

All methods use exactly the same inputs (RGB, predicted depth, ground-truth poses) and run on the same GPU.

4.3. Metrics

Reconstruction quality. For each held-out frame k, we render $\hat{I}_k$ from its ground-truth pose and compute PSNR and SSIM with respect to the observed RGB image I_k. For geometry we compute a symmetric Chamfer distance (CD) between the reconstructed surface and the phantom mesh: we sample point clouds from both, compute nearest-neighbour squared distances in each direction, and average over points restricted to a fixed radial band around the centerline to ignore distant background.

Runtime and memory. Effective FPS is defined as $\text{FPS} = N/t$, where N is the number of processed frames (including non-keyframes) and t is the total wall-clock time for the sequence, including all components of each method. We also report the number of active Gaussians at the end of optimization.

Table 1: **Quantitative comparison on C3VD phantom sequences** (ground-truth poses). Values are mean ± standard deviation over four sequences. Higher is better for PSNR, SSIM, FPS; lower is better for Chamfer distance (CD). Per-sequence scores are provided in Appendix D.

Method	PSNR ↑	SSIM ↑	FPS ↑	CD ↓	# points (M)
EndoGSLAM	11.32 ± 0.24	0.346 ± 0.025	1.08 ± 0.18	6.61 ± 1.35	3.11 ± 0.36
MonoGS	11.26 ± 0.66	0.320 ± 0.053	8.20 ± 0.67	7.91 ± 0.56	0.59 ± 0.07
Ours	11.56 ± 0.92	0.335 ± 0.057	6.73 ± 0.43	5.73 ± 0.58	1.14 ± 0.21

Coverage. Using tubular coordinates (s, r, θ), we maintain online coverage statistics for each centerline segment and circumferential bin: (i) a scalar coverage score per segment (fraction of time the segment is within a viewing cone from the active camera) and (ii) a histogram of Gaussian counts over θ ("quadrants"). These metrics are updated in real time as s grows and are later compared to a visibility oracle derived from the phantom mesh (Appendix C).

4.4. Geometry–speed trade-off

Table 1 summarizes reconstruction quality, runtime, and model size, averaged over the four C3VD sequences; per-sequence results are given in Appendix D.

Our centerline-aware mapper matches or slightly improves PSNR and SSIM relative to both baselines while achieving a Chamfer distance lower than both: we reach EndoGSLAM-level CD despite using fewer Gaussians, and improve CD by 2.2 mm on average over the MonoGS baseline. At the same time, our effective FPS is close to that of MonoGS and approximately 6× higher than EndoGSLAM, even though we maintain an online centerline, Bishop frame, and coverage counters. This supports our claim that a thin geometric layer can recover much of the geometry that EndoGSLAM obtains from a heavier mapping stack while retaining MonoGS-like frame rates.

4.5. Qualitative geometry comparison

Figure 2 compares the reconstructed colon geometry on a representative C3VD sequence. We show the ground-truth phantom mesh alongside point clouds from our centerline-aware mapper, EndoGSLAM, and the MonoGS-style baseline, annotated with Chamfer distance (CD), number of active Gaussians, and effective FPS.

MonoGS achieves the highest FPS but allocates many Gaussians throughout the lumen, producing a thick, irregular tube and higher CD. EndoGSLAM concentrates points more tightly on the wall but at the cost of $\sim 3\times$ more Gaussians and substantially lower FPS. Our method forms a thin, continuous tubular shell that more closely matches the phantom geometry while using fewer Gaussians than EndoGSLAM and running at near-MonoGS frame rates. This visual comparison agrees with the quantitative trade-off in Table 1: a small amount of colon-aware geometry suffices to recover EndoGSLAM-level accuracy without sacrificing the efficiency of a MonoGS-style mapper.

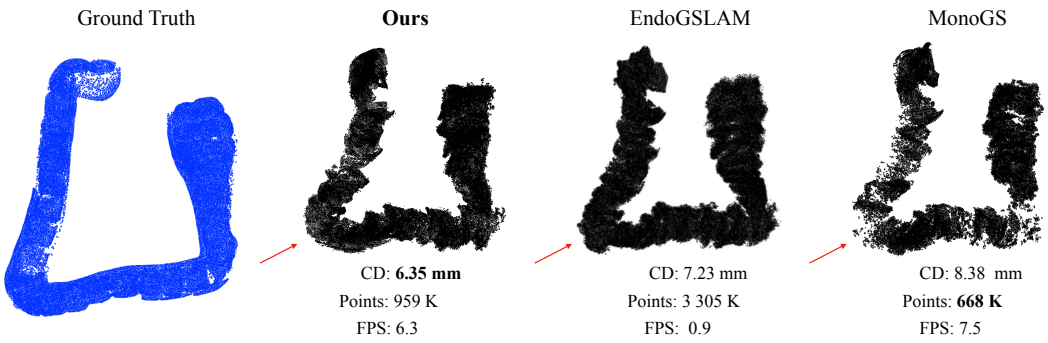

Figure 2: Geometry–speed trade-off on a C3VD phantom sequence. From left to right: ground-truth colon mesh and reconstructions from our centerline-aware mapper, EndoGSLAM, and a MonoGS-style mapper. Our method attains lower hamfer distance than both baselines, uses roughly 3× fewer Gaussians than EndoGSLAM, and maintains near-MonoGS frame rates, yielding a thin tubular wall with fewer interior splats.

4.6. Coverage in colon coordinates

A key advantage of expressing the map in tubular coordinates is that coverage can be computed online in the same representation. Figure 3 visualizes our coverage output for one phantom sequence.

Panel (a) shows the colon unrolled into (s, θ) with coverage encoded as a heatmap. Gaps or cold regions correspond to stretches that were rarely viewed with favourable distance and angle; in the phantom sequences these often occur around sharp bends and short withdrawal bursts. Panel (b) aggregates this into simple segment-wise summaries for during or after a procedure, highlighting under-inspected segments. Panel (c) uses the same coordinates to report how Gaussians (and therefore map capacity) are distributed over circumferential angle. If a segment's Gaussians are heavily concentrated in one quadrant, it means the camera spent most of its time looking along that wall; the opposite wall may have received little attention, even if overall time in that segment was adequate.

These coverage statistics are updated continuously as the centerline grows, with negligible additional cost: they reuse the same projections needed to render keyframes and require only per-segment, per-quadrant counters. In Appendix C we compare our online coverage scores to an oracle based on the phantom mesh and ground-truth poses and find good agreement, supporting their use as geometry-aware quality indicators rather than purely heuristic visualizations.

5. Conclusions and limitations

We have shown that a thin layer of colon-aware geometry on top of a standard 3D Gaussian mapper is enough to change the geometry–speed–utility trade-off. By organizing the map around an online centerline and tubular coordinates, we match or improve the Chamfer

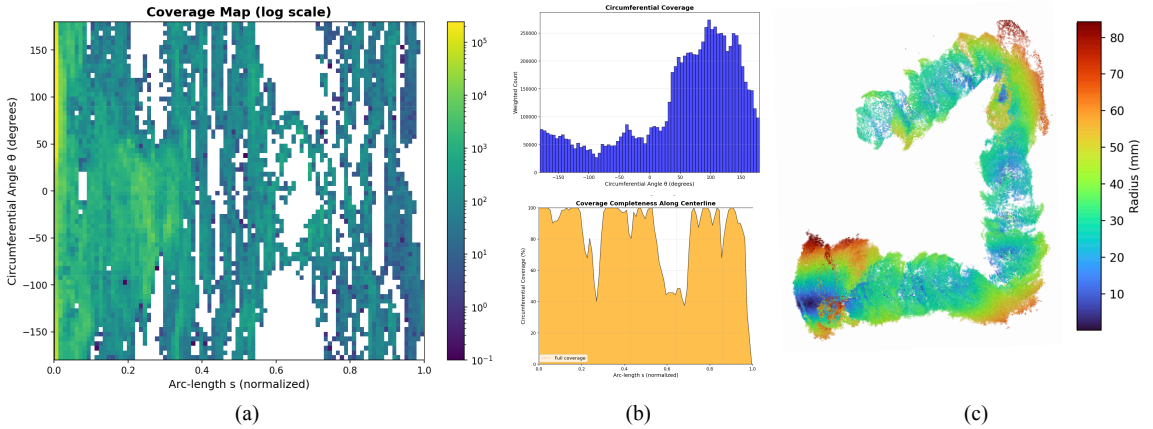

Figure 3: **Online coverage from tubular coordinates on a phantom sequence.** (a) Unrolled coverage map in (s, θ): horizontal axis is arc length along the centerline, vertical axis is circumferential angle θ, and color denotes our per-bin coverage score. Vertical lines mark anatomical segments of the phantom. (b) Segment-wise coverage summaries: top, bar plot of the fraction of surface area in each segment exceeding a minimum coverage threshold; bottom, coverage fraction as a function of s. (c) Circumferential balance: histogram of Gaussian counts over θ, aggregated along the entire sequence (left) and for two example segments (right). Strong asymmetries indicate that the camera spent most of the time looking at one wall, suggesting that the opposite wall may warrant closer inspection.

distance of an endoscopy-specific 3DGS baseline while running at frame rates close to a MonoGS-style mapper on long C3VD phantom sequences. At the same time, the same representation yields unrolled colon views, segment-wise coverage summaries, and simple circumferential balance metrics essentially "for free," without new networks or changes to the underlying Gaussian rasterizer.

Our method has several limitations. First, it assumes externally provided camera poses, which are not yet routine in clinical colonoscopy; in practice, these would need to come from a robust SLAM or robotic tracking system, and our current implementation does not correct global drift or miscalibration. Second, although the additional overhead of maintaining a centerline, Bishop frame, and coverage counters is modest, the overall pipeline remains GPU-intensive and inherits the memory and compute demands of 3D Gaussian mapping. Third, our evaluation is restricted to silicone phantom data without annotated lesions, so we cannot yet assess lesion-level endpoints or the impact of coverage metrics on real withdrawal quality. Despite these constraints, the results suggest that expressing 3DGS reconstructions in colon-intrinsic coordinates is a practical way to obtain both better use of model capacity and clinically relevant coverage outputs at minimal additional cost. Future work includes learning centerlines directly from image features, integrating additional cues such as near-field lighting and multi-view depth, extending the formulation to joint pose and centerline refinement, and validating coverage-based metrics in in vivo colonoscopy cohorts.

References

Christian Bauer and Horst Bischof. Segmentation of interwoven 3D tubular tree structures in volumetric data. *Medical Image Analysis*, 13(1):172–184, 2009. doi: 10.1016/j.media.2008.06.006. URL https://www.sciencedirect.com/science/article/pii/S1361841509001406.

Taylor L. Bobrow, Mayank Golhar, Rohan Vijayan, Venkata S. Akshintala, Juan R. Garcia, and Nicholas J. Durr. Colonoscopy 3d video dataset with paired depth from 2d–3d registration. *Medical Image Analysis*, 90:102956, 2023. ISSN 1361-8415. doi: 10.1016/j.media.2023.102956. URL https://durrlab.github.io/C3VD/.

Håvard D. Borgli et al. Hyperkvasir, a comprehensive multi-class image and video dataset for gastrointestinal endoscopy. *Scientific Data*, 7(283), 2020. doi: 10.1038/s41597-020-00622-y.

Mirosław Chlebiej et al. Customizable tubular model for n-furcating blood vessels and its application to 3d reconstruction of the cerebrovascular system. *Computer Methods and Programs in Biomedicine*, 234:107490, 2023. doi: 10.1016/j.cmpb.2023.107490. URL https://pmc.ncbi.nlm.nih.gov/articles/PMC10182136/.

Petter A. Floor et al. 3d reconstruction of the human colon from capsule endoscope video. In *Computer Vision in Clinical Surgery (CVCS) Workshop*, 2022. URL https://ceur-ws.org/Vol-3271/Paper2_CVCS2022.pdf.

Romain Hardy, Tyler Berzin, and Pranav Rajpurkar. Coloncrafter: A depth estimation model for colonoscopy videos using diffusion priors. *arXiv preprint arXiv:2509.13525*, 2025.

Michal F. Kaminski et al. Quality indicators for colonoscopy and the risk of interval cancer. *New England Journal of Medicine*, 362(19):1795–1803, 2010. doi: 10.1056/NEJMoa0907667.

Nikhil Keetha, Jay Karhade, Krishna Murthy Jatavallabhula, Gengshan Yang, Sebastian Scherer, Deva Ramanan, and Jonathon Luiten. Splatam: Splat, track & map 3d gaussians for dense rgb-d slam. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. URL https://spla-tam.github.io/.

Ruibin Ma, Rui Wang, Yubo Zhang, Stephen Pizer, Stephanie K. McGill, J. Keith Rosenman, and Jan-Michael Frahm. RNNSLAM: Reconstructing the 3d colon to visualize missing regions during a colonoscopy. *Medical Image Analysis*, 72:102100, 2021. ISSN 1361-8415. doi: 10.1016/j.media.2021.102100. URL https://www.sciencedirect.com/science/article/pii/S1361841521001468.

Hidenobu Matsuki, Riku Murai, Paul H. J. Kelly, and Andrew J. Davison. Gaussian splatting slam. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 21620–21630, 2024. URL https://openaccess.thecvf.com/content/CVPR2024/html/Matsuki_Gaussian_Splatting_SLAM_CVPR_2024_paper.html.

Douglas K. Rex et al. Quality indicators for colonoscopy. *Gastrointestinal Endoscopy*, 81 (1):31–53, 2015. doi: 10.1016/j.gie.2014.07.058.

Christos Rossides et al. 3d cyclorama for digital unrolling and visualization of tubular colon samples. *Scientific Reports*, 11:13519, 2021. doi: 10.1038/s41598-021-93184-x. URL https://www.nature.com/articles/s41598-021-93184-x.

Arnav Sengupta, Adrien Bartoli, Olivier Colliot, and Sid Ahmed Akkouche. Colonoscopic 3d reconstruction by tubular non-rigid structure-from-motion. In *Information Processing in Computer-Assisted Interventions (IPCAI)*, pages 1–11. Springer, 2021. doi: 10.1007/978-3-030-79984-5_1. URL https://encov.ip.uca.fr/publications/pubfiles/2021_Sengupta_etal_IPCAI_tubular.pdf.

Rebecca L. Siegel et al. Colorectal cancer statistics, 2024. *CA: A Cancer Journal for Clinicians*, 74(x):x–x, 2024. doi: 10.3322/caac.XXXX.

Kailing Wang, Chen Yang, Yuehao Wang, Sikuang Li, Yan Wang, Qi Dou, Xiaokang Yang, and Wei Shen. EndoGSLAM: Real-time dense reconstruction and tracking in endoscopic surgeries using gaussian splatting. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, Lecture Notes in Computer Science. Springer, 2024. URL https://arxiv.org/abs/2403.15124.

Shu Zhang Zhao et al. Magnitude, risk factors, and factors associated with adenoma miss rate of tandem colonoscopy: A systematic review and meta-analysis. *Gastroenterology*, 156(6):1661–1674, 2019. doi: 10.1053/j.gastro.2019.01.260.

Zihan Zhu, Songyou Peng, Viktor Larsson, Yifan Wang, Qiang Wang, Zhuwen Li, Martin R. Oswald, and Marc Pollefeys. NICE-SLAM: Neural implicit scalable encoding for slam. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12776–12786, 2022. doi: 10.1109/CVPR52688.2022.01245. URL https://arxiv.org/abs/2112.12130.

Appendix A. Implementation details

Backbone and optimization. Our implementation builds on a MonoGS-style 3D Gaussian mapping backbone with the same differentiable rasterizer and optimizer. Unless otherwise noted, we use the same hyperparameters for all sequences: Adam with a fixed learning rate, per-primitive opacity and covariance regularizers, and the depth and photometric losses described in Section 3.1. We process frames in temporal order and run a fixed number of gradient steps per keyframe; non-keyframes are used only to update the centerline, Bishop frame, and coverage counters.

Centerline update and thresholds. The backbone points used to define the centerline are updated online as in Section 3.2. We trigger a new backbone point when the camera center has moved at least $d_{\min}$ along the trajectory, the incremental bend angle is below a maximum $\theta_{\max}$, and the candidate is more than d_{loop} from non-neighbour points (loop avoidance). In practice, $d_{\min}$ is on the order of a few millimeters so that backbone points

are more sparsely spaced than individual frames but more densely than typical colonic segments, and $\theta_{\max}$ prevents implausibly sharp turns. The centerline is refit as a cubic B-spline whenever a new backbone point is added; this update is inexpensive compared to Gaussian optimization.

Keyframing and continuous coverage updates. At each frame we update the arc-length accumulator A_t and, when a keyframe is triggered, we (i) add the current frame to the optimization set, (ii) freeze the current coverage statistics for that frame, and (iii) reset A_t. Coverage counters are updated for every frame, not just keyframes: for the current camera pose we identify the closest centerline sample and increment coverage scores for bins within a small arc-length neighbourhood of this sample and within a viewing cone of the camera ray. This allows coverage maps to be displayed continuously during the procedure without waiting for optimization to converge.

Appendix B. Chamfer distance and alignment

Although both the reconstruction and the phantom CAD mesh are expressed in the same physical units (millimeters), there are small residual calibration mismatches between the coordinate frames. Before computing surface error, we therefore perform a rigid alignment with Iterative Closest Point (ICP), and then measure a one-sided distance from the reconstructed surface to the subset of the phantom surface that is actually seen by the cameras.

Rigid alignment. For each sequence we first extract a reconstructed surface point cloud $\mathcal{P}_{\text{traj}} = \{x_i\}$ from the Gaussian map (e.g. by sampling points from the splats or rendering a dense depth map and back-projecting valid pixels). We also sample points $\mathcal{P}_{\text{obj}} = \{y_j\}$ from the phantom OBJ mesh. We run rigid ICP (point-to-plane) to find a transformation $T_{\text{ICP}} \in SE(3)$ that aligns the reconstructed surface to the phantom:

$$T_{\text{ICP}} = \arg\min_{T \in SE(3)} \sum_i \operatorname{dist}\big(Tx_i,\, \mathcal{P}_{\text{obj}}\big)^2.$$

All reconstruction points are then transformed by T_{ICP}. Camera poses are updated consistently so that rays still intersect the aligned phantom mesh.

Visible phantom surface. We are interested in the part of the phantom surface that is actually observable given the camera trajectories and intrinsics, rather than the entire CAD mold. To approximate this, for each sequence we cast rays from the aligned cameras through the image plane and intersect them with the phantom mesh: for every pixel (u, v) in each frame with known intrinsics K and pose T_t, we:

1. Back-project (u, v) to a 3D ray in camera coordinates using K^{-1},
2. Transform the ray into world coordinates using T_t,
3. Compute the first intersection of this ray with the phantom mesh.

Each successful ray–mesh intersection yields a point z_k on the phantom surface. Collecting all such intersections over the sequence gives a set $\mathcal{P}_{\text{hit}} = \{z_k\}$ of "hit" points: the subset of the phantom that is actually seen by the cameras under the given trajectory and field of view.

One-sided surface error (traj → hit obj). Our Chamfer distance (CD) is then defined as a one-sided RMS distance from the reconstructed surface to the visible phantom surface:

$$\mathrm{CD} = \sqrt{\frac{1}{|\mathcal{P}_{\mathrm{traj}}|} \sum_{x_i \in \mathcal{P}_{\mathrm{traj}}} \min_{z_k \in \mathcal{P}_{\mathrm{hit}}} \|x_i - z_k\|_2^2}.$$

Here $\mathcal{P}_{\mathrm{traj}}$ denotes the aligned reconstruction points (our "traj" set) and $\mathcal{P}_{\mathrm{hit}}$ denotes the set of phantom points that were actually hit by at least one camera ray (the "hit obj" set). We do not compute the reverse direction (obj → traj), because parts of the phantom mesh may never be visible given the particular trajectory; including them would penalize methods for failing to reconstruct unseen surfaces.

All CD values reported in the main paper and Appendix are in millimeters and use this one-sided traj→hit-obj definition, with identical raycasting and sampling settings for all methods.

Appendix C. Per-sequence quantitative results

Table 2 reports per-sequence PSNR, SSIM, FPS, Chamfer distance, and number of active Gaussians for all methods.

Table 2: Per-sequence results on C3VD phantom data (ground-truth poses).

Method	Sequence	Pose	PSNR ↑	SSIM ↑	FPS ↑	CD ↓	# points
EndoGSLAM	v1	GT	11.1376	0.3579	1.295	6.6143	2 579 787
EndoGSLAM	v2	GT	11.3462	0.3679	0.941	7.6801	3 306 894
EndoGSLAM	v3	GT	11.7037	0.3538	0.915	7.2696	3 305 803
EndoGSLAM	v4	GT	11.0808	0.3023	1.154	9.7394	3 261 161
MonoGS	v1	GT	12.1283	0.3980	9.113	7.7577	624 033
MonoGS	v2	GT	10.8580	0.3067	8.262	7.1817	520 626
MonoGS	v3	GT	11.4013	0.2915	7.581	8.3797	668 002
MonoGS	v4	GT	10.6372	0.2823	7.835	8.3044	545 336
Ours	v1	GT	12.6561	0.3770	7.279	5.6035	993 965
Ours	v2	GT	10.9343	0.3272	6.239	4.9906	1 430 093
Ours	v3	GT	11.9787	0.3786	6.693	6.3479	959 687
Ours	v4	GT	10.6890	0.2586	6.701	5.9614	1 163 906

Appendix D. Additional qualitative results and robustness to pose noise

Figure 4 shows additional qualitative reconstructions across all four phantom sequences, complementing the single-sequence comparison in Figure 2.

To illustrate the behaviour of the centerline under pose noise, we add synthetic perturbations to the ground-truth trajectories and recompute the centerline.

These experiments support the intended use case: the centerline acts as a low-pass filter over camera motion and is robust to local jitter, but it does not correct gross drift or miscalibration, which must be handled by the upstream tracking system.

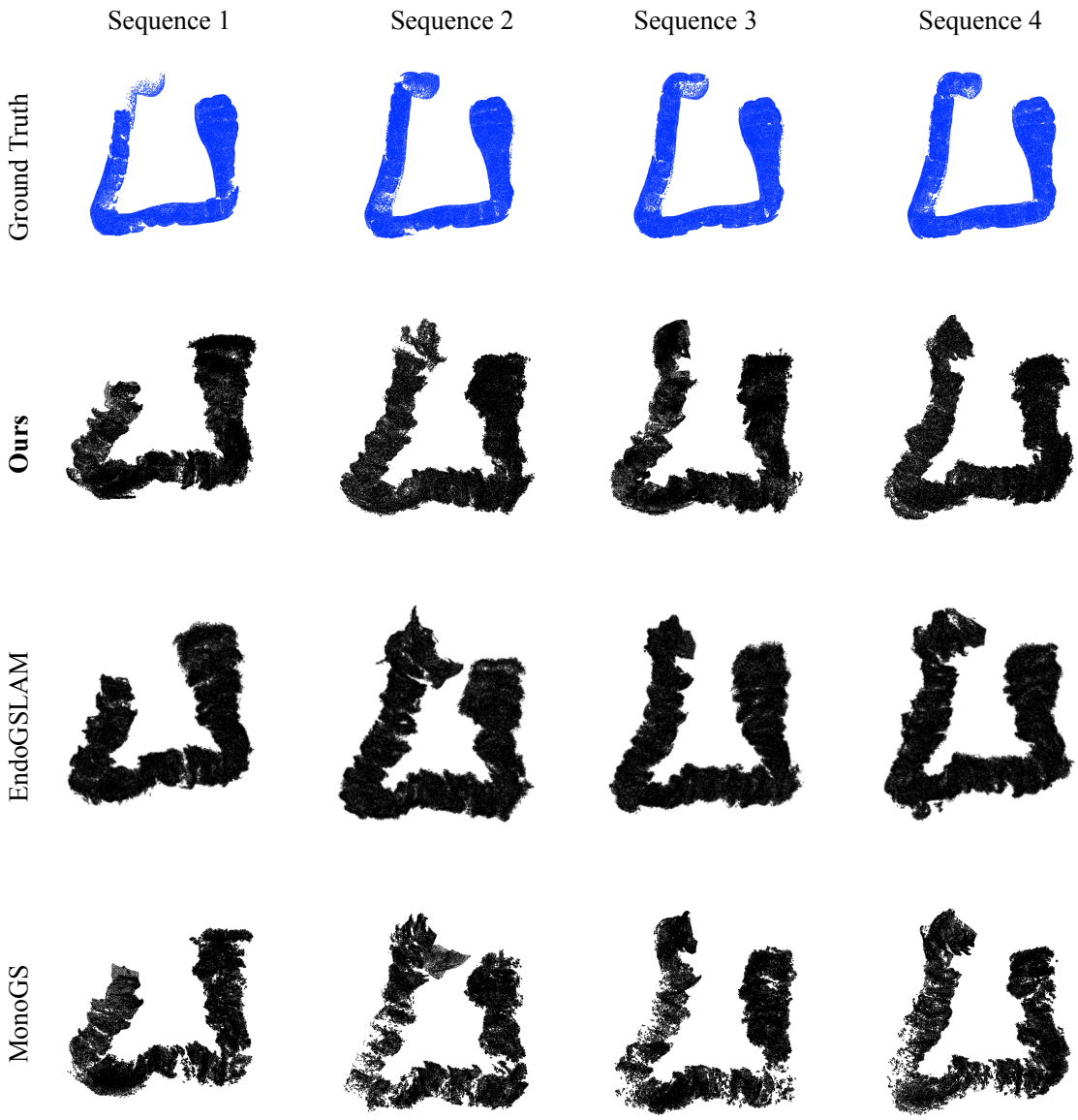

Figure 4: Additional qualitative reconstructions on all four C3VD phantom sequences. For each sequence we show the reconstructed Gaussians from our method (colored by radial distance from the centerline) and the corresponding ground-truth mesh outline. Across sequences, Gaussians are consistently concentrated in a thin band around the colon wall, with few interior points, illustrating the effect of the tubular prior and centerline-aware keyframing.

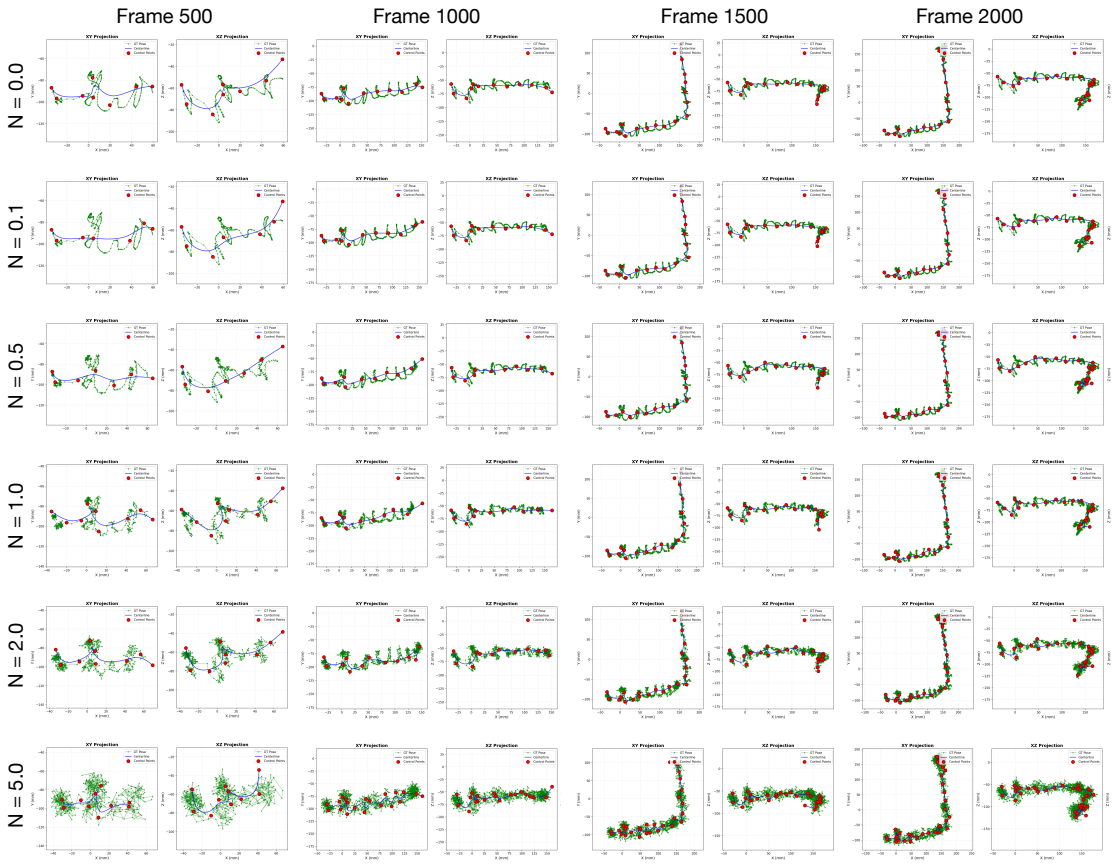

Figure 5: Effect of pose noise on the estimated centerline. Left: ground-truth camera trajectory and corresponding centerline. Middle: trajectory with added high-frequency local jitter; the B-spline centerline remains smooth and close to the original, indicating robustness to local pose noise. Right: trajectory with a slowly varying global bias; the centerline is displaced accordingly, since our method does not attempt to correct systematic tracking drift. In practice, this means that our representation can tolerate realistic local noise from a tracker but still depends on a globally reasonable pose estimate, as expected for a mapping-only system.

Appendix E. Bishop frame construction (details)

For completeness we summarize the discrete Bishop-frame computation used in our implementation; the main text gives only the high-level description.

Given sampled centerline points $\{C_m\}_{m=0}^{M}$ with arc-length parameters $\{s_m\}$, we estimate unit tangents T_m by centered finite differences

$$T_m = \frac{C_{m+1} - C_{m-1}}{\|C_{m+1} - C_{m-1}\|_2}, \qquad m = 1, \ldots, M-1,$$

and one-sided differences at the endpoints. At $m = 0$ we choose an initial normal $N_{1,0}$ orthogonal to T_0 by crossing with a fixed reference axis,

$$a_{\text{ref}} = \begin{cases} (0,1,0)^\top, & \text{if } |T_0 \cdot (0,0,1)^\top| > 0.9, \\ (0,0,1)^\top, & \text{otherwise,} \end{cases}$$

$$N_{1,0} = \frac{T_0 \times a_{\text{ref}}}{\|T_0 \times a_{\text{ref}}\|_2}, \qquad N_{2,0} = \frac{T_0 \times N_{1,0}}{\|T_0 \times N_{1,0}\|_2}.$$

For $m = 1, \ldots, M$ we propagate the normals by discrete parallel transport. Let

$$a_m = T_{m-1} \times T_m, \qquad \theta_m = \operatorname{atan2}(\|a_m\|_2,\, T_{m-1} \cdot T_m),$$

and define the unit axis $\hat{a}_m = a_m / \|a_m\|_2$ when $\|a_m\|_2 > \epsilon$ and $\hat{a}_m = T_m$ otherwise. We then rotate the previous normal $N_{1,m-1}$ by Rodrigues' formula

$$R(\hat{a}_m, \theta_m) = I + \sin\theta_m [\hat{a}_m]_\times + (1 - \cos\theta_m)[\hat{a}_m]_\times^2,$$

and set

$$N_{1,m} = R(\hat{a}_m, \theta_m)\, N_{1,m-1}, \qquad N_{2,m} = \frac{T_m \times N_{1,m}}{\|T_m \times N_{1,m}\|_2}.$$

This discrete Bishop frame $\{T_m, N_{1,m}, N_{2,m}\}$ satisfies orthonormality up to numerical precision and minimizes twist along the curve, avoiding the instability of Frenet frames in nearly straight segments. In practice we re-orthogonalize the triplet by a single Gram–Schmidt step every few samples; this has negligible cost compared to Gaussian rendering and optimization.