Online 3D reconstruction and dense tracking in endoscopic videos

Michel Hayoz[1], Christopher Hahne[1], Thomas Kurmann[3], Max Allan[3], Guido Beldi[2] and Daniel Candinas[2] and Pablo Márquez-Neila[1], and Raphael Sznitman[1]

[1] ARTORG Center, University of Bern, Switzerland
[2] Dept. of Visceral Surgery and Medicine, Inselspital, Switzerland
[3] Applied Research, Intuitive Surgical, USA
michel.hayoz@unibe.ch

Abstract. 3D scene reconstruction from stereo endoscopic video data is crucial for advancing surgical interventions. In this work, we present an online framework for online, dense 3D scene reconstruction and tracking, aimed at enhancing surgical scene understanding and assisting interventions. Our method dynamically extends a canonical scene representation using Gaussian splatting, while modeling tissue deformations through a sparse set of control points. We introduce an efficient online fitting algorithm that optimizes the scene parameters, enabling consistent tracking and accurate reconstruction. Through experiments on the StereoMIS dataset, we demonstrate the effectiveness of our approach, outperforming state-of-the-art tracking methods and achieving comparable performance to offline reconstruction techniques. Our work enables various downstream applications thus contributing to advancing the capabilities of surgical assistance systems.

Keywords: stereo endoscopy, 3D reconstruction, online dense tracking

1 Introduction

3D scene reconstruction represents a fundamental challenge for surgical scene understanding [2,8,12,11]. The ability to infer accurate 3D geometry from endoscopic image data would have numerous important downstream tasks, such as retrospective analysis for surgical training, integrated virtual overlays of preoperative image data, and augmented surgical robotics. As such, the need for methods that yield real-time and consistent 3D estimates of the surgical site is paramount for the next generation of surgical assistant tools.

Recent years have seen a variety of highly promising methods for 3D reconstruction of surgical scenes. Primarily driven by advances in neural rendering, these have shown an impressive ability to reconstruct dynamically deforming surgical scenes [15,18,17,16,9]. Unfortunately, they are limited by offline processing or lack physical constraints to control tissue deformations. Other approaches originating from natural images, such as video-based tracking [5,19] and multi-camera reconstruction [10], have also shown great potential, while their refinement to surgical scenes is not always evident. The recently introduced benchmark for soft-tissue trackers in robotic surgery [3] will also provide interesting algorithmic developments despite its focus on sparse-point tracking and not dense tracking – the latter being essential for most downstream applications.

In this work, however, we focus on online 3D reconstructions from stereo endoscopic video data. Drawing from recent developments in Gaussian splatting [6], which allows for fast reconstruction and rendering, we propose a novel framework by way of dense point tracking in the endoscopic scene. Unlike traditional methods that assume a fixed topology at initialization [10,15], our approach dynamically initializes a set of Gaussian models and updates as new scene parts become visible over time. We incorporate per-Gaussian learning rate modulation to ensure accurate optimization while retaining information from past frames. Additionally, we integrate optical flow motion initialization to ease convergence in single-camera settings. Finally, we parametrize deformations using a small set of control points distributed on the scene surface and proportional to scene complexity. The reduced number of control points, allowed by using Gaussian kernel interpolation in the deformation fields, results in faster fitting times, simpler geometric priors, and quicker tracking. To validate our method, we evaluate it on the publicly available StereoMIS dataset. We outperform state-of-the-art tracking algorithms and demonstrate comparable performance to offline 3D reconstruction methods [4].

2 Method

Our scene reconstruction framework is designed to densely track surface points in a video sequence (Fig. 1). Each frame t of the video consists of tuple $\mathbf{f}_t = (\mathbf{i}_t, \mathbf{d}_t, \mathbf{P}_t)$ containing the RGB image, the depth map, and the camera pose, respectively. Our tracking method models the scene using a combination of static Gaussian splatting and non-rigid deformations. The parameters of the model are optimized in an online manner based on photometric, geometric, and physical constraints by minimizing the reconstruction error between each video frame $\mathbf{f}_t$ and the synthetically generated frame $\hat{\mathbf{f}}_t$. The subsequent sections describe the scene representation model and the online fitting algorithm.

2.1 Scene Model

Canonical and non-rigid scenes. The rigid component of the scene, known as *canonical scene*, is modeled via Gaussian splatting [6] using a collection $\mathcal{G} = \{g_i\}_{i=1}^{G}$ of 3D colored Gaussians. A colored Gaussian is defined by a tuple g_i containing the position $\boldsymbol{\mu}_i \in \mathbb{R}^3$, the scale $\mathbf{s}_i \in \mathbb{R}^3$, the orientation $\mathbf{q}_i \in \mathbb{H}$,

[4] Code: `https://github.com/mhayoz/online_endo_track`, data:`https://zenodo.org/records/10867949`

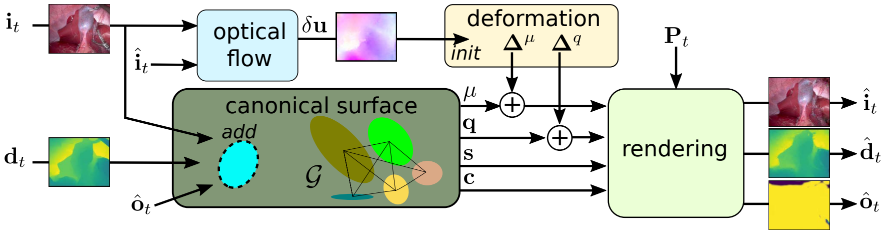

Fig. 1: Overview of our proposed scene reconstruction and dense tracking method.

and the color $\mathbf{c}_i \in \mathbb{R}^3$ of the Gaussian. The covariance of the Gaussian can be trivially computed from its orientation and scale as $\boldsymbol{\Sigma} = R(\mathbf{q})\,\mathrm{diag}(\mathbf{s})^2 R(\mathbf{q})^T$, where $R(\mathbf{q})$ is the rotation matrix corresponding to the quaternion $\mathbf{q}$.

Tissue deformations are an integral part of surgical scenes. We model these by warping the canonical surface with a translation vector field $\boldsymbol{\mathit{\Delta}}^{\mu}\colon \mathbb{R}^3 \to \mathbb{R}^3$ and a rotation vector field $\boldsymbol{\mathit{\Delta}}^{q}\colon \mathbb{R}^3 \to \mathbb{H}$. These warps act additively over the locations and orientations of the Gaussians of the canonical scene. For the Gaussian g_i, its warped location and orientation are then $\boldsymbol{\mu}_i' = \boldsymbol{\mu}_i + \boldsymbol{\mathit{\Delta}}^{\mu}(\boldsymbol{\mu}_i)$, and $\mathbf{q}_i' = \mathbf{q}_i + \boldsymbol{\mathit{\Delta}}^{q}(\boldsymbol{\mu}_i)$, respectively. Note that the scales of the Gaussians are not warped, as we expect nearly isometric tissue deformation.

The deformation fields are modelled with a collection $\mathcal{K} = \{(\mathbf{p}_k, \delta\boldsymbol{\mu}_k, \delta\mathbf{q}_k)\}_{k=1}^{K}$ of control points. Each control point is a tuple containing its position $\mathbf{p}_k \in \mathbb{R}^3$, a translation offset $\delta\boldsymbol{\mu}_k \in \mathbb{R}^3$, and an orientation offset $\delta\mathbf{q}_k \in \mathbb{H}$, which serve as the parameters of the translation and the orientation fields. Similar to [13], the translation field at the location $\mathbf{x} \in \mathbb{R}^3$ is defined as a weighted average,

$$\boldsymbol{\mathit{\Delta}}^{\mu}(\mathbf{x}) = \frac{1}{\sum_{k=1}^{K} w(\mathbf{x}, \mathbf{p}_k)} \sum_{k=1}^{K} w(\mathbf{x}, \mathbf{p}_k)\,\delta\boldsymbol{\mu}_k, \tag{1}$$

where the Gaussian kernel, $w(\mathbf{x}_1, \mathbf{x}_2) = \exp\left(-\gamma \left\|\mathbf{x}_1 - \mathbf{x}_2\right\|_2^2\right)$, is used to measure the contribution of control point k to location $\mathbf{x}$, and the hyperparameter γ determines the rate of decay in the influence of the control points over distance. The orientation field is defined similarly using the orientation offsets of the control points.

Rendering. Following [6], the image rendering function r_{image} takes a collection of Gaussians $\mathcal{G}$ and the camera pose $\mathbf{P}$ and computes the color of the pixel at location $\mathbf{u} \in \mathbb{R}^2$ as,

$$r_{\text{image}}(\mathbf{u}; \mathcal{G}, \mathbf{P}) = \sum_{i=1}^{G} \mathbf{c}_{\omega[i]} \cdot \alpha_{\omega[i]} \prod_{i'=1}^{i-1} \left(1 - \alpha_{\omega[i']}\right), \tag{2}$$

where sequence ω contains the indices of the Gaussians sorted by depth from the camera. The factor α_i is the opacity corresponding to Gaussian g_i at pixel $\mathbf{u}$

after projection onto the screen space. Details on computing ω and Gaussian projection can be found in [6]. Analog to image rendering, the depth rendering function replaces color with depth, while the opacity rendering function replaces color with a constant value of 1. We obtain the synthetic color image $\hat{\mathbf{i}}(\mathcal{G}, \mathbf{P})$, depth $\hat{\mathbf{d}}(\mathcal{G}, \mathbf{P})$ and opacity $\hat{\mathbf{o}}(\mathcal{G}, \mathbf{P})$ by evaluating the corresponding rendering functions at all image pixels $\{\mathbf{u}_p\}_{p=1}^{P}$.

2.2 Model fitting

Model fitting works in an online manner. For each new frame at time t, we find the model parameters $\mathcal{G}_t$ and $\mathcal{K}_t$ that minimize the differences between the measured images $\mathbf{i}_t, \mathbf{d}_t$ and their predicted counterparts $\hat{\mathbf{i}}_t, \hat{\mathbf{d}}_t$. Our optimization approach proceeds in three steps: first, the canonical scene is updated with new Gaussians covering previously unseen areas of the scene; second, control points $\mathcal{K}_t$ are initialized using optical flow; third, the parameters in $\mathcal{G}_t$ and $\mathcal{K}_t$ are jointly optimized to minimize the reconstruction error.

Canonical scene extension: As the camera moves throughout the sequence, establishing a canonical scene at the beginning of the sequence, as is done in [10], is not feasible. Instead, we progressively expand the canonical scene by adding new Gaussians to cover new regions of the scene as they appear. We identify these regions by finding pixels with low opacity in the opacity image $\hat{\mathbf{o}}(\boldsymbol{\Delta}(\mathcal{G}_{t-1}, \mathcal{K}_{t-1}), \mathbf{P}_t)$, and add a new Gaussian to each pixel with an opacity smaller than 0.95. The parameters of the new Gaussian are set according to the color and coordinates of the corresponding pixel $\mathbf{u}$. Its position is found by back projection to the 3D scene, $\boldsymbol{\mu} = \pi_{3D}(\mathbf{u}; \mathbf{d}_t, \mathbf{P}_t)$, and its scale is set to the distance between $\boldsymbol{\mu}$ and to the position $\boldsymbol{\mu}_j$ of the nearest Gaussian.

Control point initialization: The control points of the deformation fields are placed in the locations of a random subset of P_t Gaussians serving as *anchor Gaussians*, $\mathbf{p}_k = \boldsymbol{\mu}_{\sigma_t[k]}, \quad k \in \{1, \ldots, K_t\}$, where σ_t contains the indices of the *anchor Gaussians* obtained as K_t random elements sampled without replacement from the Gaussian index set $\{1, \ldots, G_t\}$. The number of control points (and anchor Gaussians) K_t is set to a fraction of the number of Gaussians, $K_t = \frac{G_t}{k}$. We set $k = 64$ in all our experiments.

The translation offsets $\delta\boldsymbol{\mu}_k$ of the control points are initialized using optical flow. RAFT [14] calculates optical flow between the synthetic image $\hat{\mathbf{i}}(\mathcal{G}_t, \mathbf{P}_t)$ and the frame image $\mathbf{i}_t$, yielding screen-space offsets that are projected back to the 3D space. The control point offsets $\delta\boldsymbol{\mu}_k$ are initialized to minimize the difference between the modeled offsets and those computed with optical flow. This optimization, detailed in the supplementary material, efficiently finds a closed-form solution via least squares.

Energy minimization: In the last stage, the initialized parameters in $\mathcal{G}_t$ and $\mathcal{K}_t$ (except positions $\{\mathbf{p}_k\}$) are updated to minimize the energy function,

$$\underset{\mathcal{G}_t, \mathcal{K}_t}{\operatorname{argmin}} E_{\text{external}} + E_{\text{internal}}, \tag{3}$$

that describes the quality of the model fit as a combination of an external energy and an internal energy. The external energy penalizes the deviations between the observed image and depth and their synthetic counterparts, measured with the standard MSE,

$$E_{\text{external}} = \lambda_1 \left\| \mathbf{i}_t - \hat{\mathbf{i}}_t \right\|_2^2 + \lambda_2 \left\| \mathbf{d}_t - \hat{\mathbf{d}}_t \right\|_2^2 . \tag{4}$$

The synthetic image $\hat{\mathbf{i}}_t = \hat{\mathbf{i}}(\boldsymbol{\Delta}(\mathcal{G}_t, \mathcal{K}_t), \mathbf{P}_t)$ and depth $\hat{\mathbf{d}}_t = \hat{\mathbf{d}}(\boldsymbol{\Delta}(\mathcal{G}_t, \mathcal{K}_t), \mathbf{P}_t)$ are rendered after warping the canonical scene with the deformation fields. The internal energy

$$E_{\text{internal}} = \lambda_3 E_{\text{rigidloc}} + \lambda_4 E_{\text{rigidrot}} + \lambda_4 E_{\text{iso}} + \lambda_5 E_{\text{visible}} \tag{5}$$

incorporates geometric and physical priors. Unlike previous methods, these priors are applied only to pairs of neighbor anchor Gaussians rather than all Gaussian pairs. Rigidity terms penalize changes in the relative positions and orientations of neighboring anchor Gaussians,

$$E_{\text{rigidloc}} = \frac{1}{4K_t} \sum_{i \in \sigma_t} \sum_{j \in \mathcal{N}_i^4} w(\boldsymbol{\mu}_{i,t}, \boldsymbol{\mu}_{j,t}) \left\| (\boldsymbol{\mu}'_{j,t-1} - \boldsymbol{\mu}'_{i,t-1}) - (\boldsymbol{\mu}'_{j,t} - \boldsymbol{\mu}'_{i,t}) \right\|_2^2 , \tag{6}$$

$$E_{\text{rigidrot}} = \frac{1}{4K_t} \sum_{i \in \sigma_t} \sum_{j \in \mathcal{N}_i^4} w(\boldsymbol{\mu}_{i,t}, \boldsymbol{\mu}_{j,t}) \left\| (\mathbf{q}'_{j,t-1} \mathbf{q}'^{-1}_{i,t-1}) - (\mathbf{q}'_{j,t} \mathbf{q}'^{-1}_{i,t}) \right\|_2^2 , \tag{7}$$

where σ_t contains the indices of the anchor Gaussians and $\mathcal{N}_i^4$ contains the 4 nearest anchor Gaussians to i. Similarly, the isometry term encourages the translation field to produce nearly isometric deformations,

$$E_{\text{iso}} = \frac{1}{4K_t} \sum_{i \in \sigma_t} \sum_{j \in \mathcal{N}_i^4} w(\boldsymbol{\mu}_{i,t}, \boldsymbol{\mu}_{j,t}) \left| \|(\boldsymbol{\mu}_{j,t} - \boldsymbol{\mu}_{i,t})\|_2^2 - \|(\boldsymbol{\mu}'_{j,t} - \boldsymbol{\mu}'_{i,t})\|_2^2 \right| . \tag{8}$$

Finally, the visibility term

$$E_{\text{visible}} = \frac{1}{\sum_{k=1}^{K_t} \mathbb{I}(\mathbf{p}_k; \mathbf{P}_t)} \sum_{k=1}^{K_t} \mathbb{I}(\mathbf{p}_k; \mathbf{P}_t) \left\| \delta \boldsymbol{\mu}_k \right\|_2^2 \tag{9}$$

penalizes deformations of the scene parts that do not project to the image, preventing drift. The invisibility predicate $\mathbb{I}(\mathbf{p}; \mathbf{P})$ is 1 if the point $\mathbf{p}$ is not visible on the screen, and 0 otherwise.

Gradient modulation: To prevent drifting in the canonical scene, we gradually slow down the updates to the parameters of its Gaussians. To this end, we count the number of times that each Gaussian i has been updated, denoted v_i, and compute the modulation factor $\rho_i = 2\,(1 - \text{sigm}(c_1 v_i - c_2))$. The gradients of the energy with respect to the parameters of the Gaussian i are multiplied by this factor before applying the optimizer update rule.

Tracking: Our method enables tracking any surface point of the scene. Given a surface point $\mathbf{x}$ at time t, tracking starts by approximating it with its closest

Gaussian i in $\mathcal{G}_t$. For subsequent frames, its 3D position is given by $\boldsymbol{\mu}'_i$. When the point to track is given as a screen-space 2D point $\mathbf{u}$, we first find its corresponding surface point $\mathbf{x} = \pi_{3D}(\mathbf{u}; \mathbf{d}_t, \mathbf{P}_t)$ and proceed as before.

3 Experimental setup and results

Datasets: We evaluated our method on the StereoMIS dataset [12] and selected subsequences of 200 frames with 10 frames per second with resolution 512x640 pixels. All sequences contain challenging scenes with breathing motions, tissue deformations, and occlusions. In each frame, we manually annotated 3 to 4 distinct landmarks to evaluate the tracking. We cannot benchmark our method on the SurgT-Challenge dataset [3] because it does not feature camera poses, an essential input to our method.

Baselines: PIPS++ [19] is a SOTA 2D long-term tracking approach. Due to memory constraints, we partition the sequences into chunks of 50 frames. We then initialize tracked points using the last estimated locations from the preceding chunk and link all estimates to form complete trajectories. To simulate dense tracking, we estimate a uniform grid of 2048 points defined in the initial frame and linearly interpolate trajectories of the evaluation points. We mask instrument areas in the input RGB images by filling them with black.

Similar to top-performing techniques in the SurgT-Challenge [3], we employ frame-to-frame optical flow to estimate dense point tracking. Specifically, we utilize RAFT [14] as a SOTA optical flow estimation method. We set the optical flow to zero for pixels occupied by surgical instruments.

Implementation Details: We optimize the canonical scene for 1000 iterations on the first frame, setting the deformation fields to zero. For each subsequent frame, we optimize for 100 iterations using Adam [7]. We run our code on an NVIDIA RTX3090 GPU, resulting in an average processing time of 2 seconds per frame. We infer depth from stereo RGB images using the stereo disparity estimated by RAFT [14] and mask surgical instrument pixels in Eq. 4, with masks inferred using DeepLabv3+ [4] trained on EndoVis2018 segmentation dataset [1]. Due to the random sampling of anchor Gaussians, we run each experiment 100 times and report the average.

Metrics: Following [5], we utilize the median trajectory error (MTE), the average position accuracy δ_{avg}, and the survival rate, as measures for accuracy and robustness in tracking. We do not assess point tracking accuracy in 3D due to the lack of reliable ground-truth depth estimates. Similarly, we only visually assess the 3D reconstruction due to having only a single endoscope and thus no hold-out views.

3.1 Results

Tab. 1 presents the point tracking results on the StereoMIS dataset. Our method consistently outperforms the baselines across most cases and, on average, for all three evaluation metrics. This demonstrates the accuracy captured by MTE and

Metric	Method	P1_1	P2_0	P2_1	P3_1	P3_2	H1_1	H3_1	mean
MTE ↓ (px)	PIPS++[19]	67.30	10.40	317.40	5.50	21.60	129.20	10.40	80.26
	RAFT[14]	42.72	83.86	197.67	10.98	18.31	126.18	21.66	71.63
	ours	**21.04**	**7.91**	**14.29**	**4.14**	**14.20**	**10.51**	**8.60**	**11.53**
δ_{avg} ↑ (%)	PIPS++[19]	**42.90**	66.20	31.20	77.80	81.10	33.90	66.20	57.04
	RAFT[14]	41.62	37.07	34.04	67.02	77.80	29.56	66.67	50.54
	ours	32.36	**66.89**	**61.14**	**80.46**	**83.67**	**54.66**	**68.50**	**63.95**
survival ↑ (%)	PIPS++[19]	37.00	**100.00**	57.30	93.80	82.20	62.40	**100.00**	76.10
	RAFT[14]	50.33	53.67	57.33	89.33	82.25	59.50	80.63	67.58
	ours	**70.90**	**100.00**	**87.45**	**100.00**	**84.20**	**87.67**	91.13	**88.77**

Table 1: Experimental results on StereoMIS dataset.

δ_{avg} and the robustness of our method indicated by the survival rate. Our method achieves a 100.0% survival rate in two cases, indicating successful tracking of all points until the end of the sequences without failure. All methods accurately track points embedded on textured surfaces, as illustrated in Fig. 2 (top). They also handle breathing motion and tool-induced deformation, which validates the physical constraints imposed by our method.

Our method handles occlusions and remains robust against motion blur caused by rapid camera movement, as illustrated in Fig. 2 (middle) at $t = 12$. In contrast, PIPS++ fails to track points with occlusions lasting longer than its chunk size, yet our method handles arbitrarily long occlusions, as depicted in Fig. 2 (bottom), where all points are tracked even after more than 100 frame occlusions between $t = 20$ and $t = 112$.

In some cases, our method struggles to capture tissue deformations accurately after long occlusions in regions with repetitive or no texture. This is attributed to unobserved tissue deformations, causing discrepancies between the estimated and actual deformations, leading to incorrect convergence, as illustrated in Fig.2 (middle) at $t = 83$ and $t = 157$. Note, our method is not intrinsically limited to short sequences but tracking in a real surgical scenario may pose unaddressed challenges for long-term tracking and reconstruction of large scenes.

Comparison against offline reconstruction methods Tab. 2 provides an additional comparison against state-of-the-art offline endoscopic scene reconstruction methods [15,18]. Our method outperforms EndoNerf and achieves comparable results to EndoSurf while being online and using only a fraction of the processing time. Example images and implementation details for [15,18] are provided in the supplementary material.

Ablation study We present an ablation study in Tab. 3. *sparse* refers to the deformation representation using a sparse set of control points, whereas *dense* explicitly represents the deformation for each Gaussian as in [10]. The most significant elements include the isometry energy E_{iso}, the visible energy E_{visible}, and the sparse deformation representation. While less critical, optical flow initialization and local-rigid energies still enhance effectiveness and robustness, which

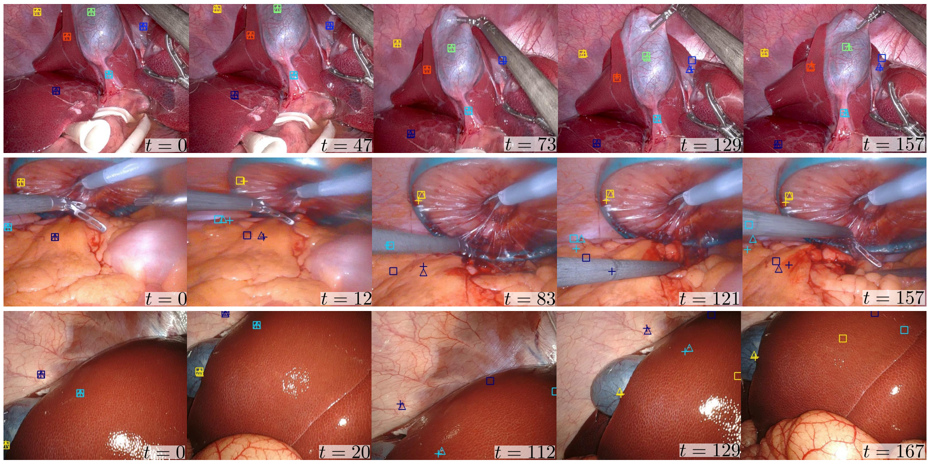

Fig. 2: 2D point tracking over time results. Annotated ground-truth points are marked with triangles, PIPS++ with squares, and **ours** with crosses.

Method	online	processing time	MTE ↓(px)	δ_{avg} ↑ (%)	survival ↑ (%)
EndoNerf [15]	✗	7h	50.70±63.05	31.22±17.17	22.54±22.05
EndoSurf [18]	✗	11h	**7.78±4.89**	**67.87±10.41**	87.88±14.49
ours	✓	7min	11.53±5.51	63.95±17.23	**88.77±10.00**

Table 2: Comparison to offline 3D reconstruction methods on StereoMIS. Metrics are reported as the mean ± std over all sequences.

is evident in the increased standard deviation of metrics when omitted from optimization.

Downstream application - 3D semantic segmentation: Our method achieves dense point tracking and coherent 3D scene reconstruction, facilitating downstream tasks like 3D semantic segmentation. We use the segmentation network defined in section 3 to infer semantic classes for each pixel, assigning them to Gaussians upon creation. Once assigned, semantics remain unchanged, enabling straightforward projection and propagation in 3D, as shown in Fig. 3. Visualizations for all scenes are available in the supplementary materials.

4 Conclusion

We proposed a framework for online 3D scene reconstruction and dense tracking from stereo endoscopic video. To achieve this, we represent the scene as a collection of Gaussians that dynamically extend as the scene is explored and model tissue deformations through a sparse set of control points with physical priors. Through point tracking evaluation on the StereoMIS dataset, we validate the physical priors and demonstrate consistent online 3D reconstruction capability, outperforming state-of-the-art video tracking methods. We show the practical-

Description	E_{rigid}	E_{iso}	E_{visible}	sparse	flow	MTE ↓ (mm)	δ_{avg} ↑(%)	survival ↑ (%)
ours	✓	✓	✓	✓	✓	**11.53±5.51**	**63.95±17.23**	**88.77±10.00**
w/o local-rigid	✗	✓	✓	✓	✓	12.26±8.31	61.72±19.06	84.50±12.12
w/o iso loss	✓	✗	✓	✓	✓	17.18±9.99	55.68±18.17	78.18±18.91
w/o inv. loss	✓	✓	✗	✓	✓	17.55±11.23	53.36±19.26	82.63±19.83
dense	✓	✓	✓	✗	✓	16.69±8.69	49.82±17.13	83.48±14.91
w/o flow	✓	✓	✓	✓	✗	13.00±6.99	58.36±17.47	86.41± 16.40

Table 3: Ablation study on StereoMIS. Metrics are reported as the mean ± std over all sequences.

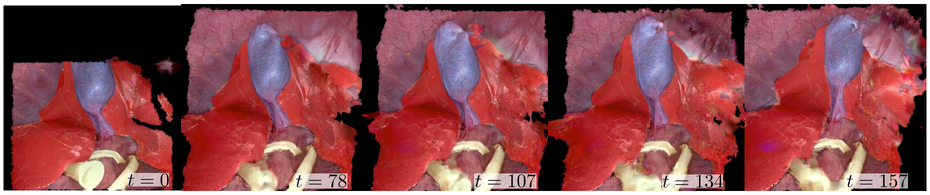

Fig. 3: 3D semantic segmentation as a downstream application. Semantic classes are overlayed: gall-bladder (purple), liver (red) and plastic tubes (yellow).

ity of our framework on 3D semantic segmentation as a downstream application, highlighting its potential in surgical training, augmented reality overlays, and robotic assistance. Future efforts should focus on enhancing speed to achieve real-time processing, extending the method to long sequences and validating its robustness in real-world scenarios.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Allan, M., Kondo, S., Bodenstedt, S., Leger, S., Kadkhodamohammadi, R., Luengo, I., Fuentes, F., Flouty, E., Mohammed, A., Pedersen, M., et al.: 2018 robotic scene segmentation challenge. arXiv preprint arXiv:2001.11190 (2020)
2. Bodenstedt, S., Allan, M., Agustinos, A., Du, X., Garcia-Peraza-Herrera, L., Kenngott, H., Kurmann, T., Müller-Stich, B., Ourselin, S., Pakhomov, D., Sznitman, R., Teichmann, M., Thoma, M., Vercauteren, T., Voros, S., Wagner, M., Wochner, P., Maier-Hein, L., Stoyanov, D., Speidel, S.: Comparative evaluation of instrument segmentation and tracking methods in minimally invasive surgery. arXiv preprint arXiv:1805.02475 (2018)
3. Cartucho, J., Weld, A., Tukra, S., Xu, H., Matsuzaki, H., Ishikawa, T., Kwon, M., Jang, Y.E., Kim, K.J., Lee, G., Bai, B., Kahrs, L.A., Boecking, L., Allmendinger, S., Müller, L., Zhang, Y., Jin, Y., Bano, S., Vasconcelos, F., Reiter, W., Hajek, J., Silva, B., Lima, E., Vilaça, J.L., Queirós, S., Giannarou, S.: Surgt challenge: Benchmark of soft-tissue trackers for robotic surgery. Medical Image Analysis **91**, 102985 (2024). `https://doi.org/10.1016/j.media.2023.102985`

4. Chen, L.C., Zhu, Y., Papandreou, G., Schroff, F., Adam, H.: Encoder-decoder with atrous separable convolution for semantic image segmentation. In: European Conference on Computer Vision (ECCV). pp. 833–851 (2018). https://doi.org/10.1007/978-3-030-01234-2_49
5. Doersch, C., Gupta, A., Markeeva, L., Recasens, A., Smaira, L., Aytar, Y., Carreira, J., Zisserman, A., Yang, Y.: Tap-vid: A benchmark for tracking any point in a video. In: Advances in Neural Information Processing Systems. vol. 35, pp. 13610–13626 (2022)
6. Kerbl, B., Kopanas, G., Leimkühler, T., Drettakis, G.: 3d gaussian splatting for real-time radiance field rendering. ACM Transactions on Graphics **42**(4) (2023)
7. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. International Conference on Learning Representations (ICLR) (2015)
8. Kurmann, T., Marquez-Neila, P., Allan, M., Wolf, S., Sznitman, R.: Mask then classify: multi-instance segmentation for surgical instruments. International Journal of Computer Assisted Radiology and Surgery (IJCARS) p. 1227–1236 (2021). https://doi.org/10.1007/s11548-021-02404-2
9. Lin, S., Miao, A.J., Lu, J., Yu, S., Chiu, Z.Y., Richter, F., Yip, M.C.: Semantic-super: A semantic-aware surgical perception framework for endoscopic tissue identification, reconstruction, and tracking. In: IEEE International Conference on Robotics and Automation (ICRA) (2023). https://doi.org/10.1109/ICRA48891.2023.10160746
10. Luiten, J., Kopanas, G., Leibe, B., Ramanan, D.: Dynamic 3d gaussians: Tracking by persistent dynamic view synthesis. In: International Conference on 3D Vision (3DV) (2024)
11. Maier-Hein, L., Eisenmann, M., Sarikaya, D., März, K., Collins, T., Malpani, A., Fallert, J., Feussner, H., Giannarou, S., Mascagni, P., Nakawala, H., Park, A., Pugh, C., Stoyanov, D., Vedula, S.S., Cleary, K., Fichtinger, G., Forestier, G., Gibaud, B., Grantcharov, T., Hashizume, M., Heckmann-Nötzel, D., Kenngott, H.G., Kikinis, R., Mündermann, L., Navab, N., Onogur, S., Roß, T., Sznitman, R., Taylor, R.H., Tizabi, M.D., Wagner, M., Hager, G.D., Neumuth, T., Padoy, N., Collins, J., Gockel, I., Goedeke, J., Hashimoto, D.A., Joyeux, L., Lam, K., Leff, D.R., Madani, A., Marcus, H.J., Meireles, O., Seitel, A., Teber, D., Ückert, F., Müller-Stich, B.P., Jannin, P., Speidel, S.: Surgical data science – from concepts toward clinical translation. Medical Image Analysis **76** (2022). https://doi.org/10.1016/j.media.2021.102306
12. Michel, H., Hahne, C., Gallardo, M., Candinas, D., Kurmann, T., Allan, M., Sznitman, R.: Learning how to robustly estimate camera pose in endoscopic videos. International Journal of Computer Assisted Radiology and Surgery (IJCARS) (2023). https://doi.org/10.1007/s11548-023-02919-w
13. Newcombe, R.A., Fox, D., Seitz, S.M.: Dynamicfusion: Reconstruction and tracking of non-rigid scenes in real-time. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 343–352 (2015). https://doi.org/10.1109/CVPR.2015.7298631
14. Teed, Z., Deng, J.: RAFT: Recurrent All-Pairs Field Transforms for Optical Flow. In: European Conference on Computer Vision (ECCV). pp. 402–419 (2020). https://doi.org/10.1007/978-3-030-58536-5-24
15. Wang, Y., Long, Y., Fan, S.H., Dou, Q.: Neural rendering for stereo 3d reconstruction of deformable tissues in robotic surgery. In: International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI). pp. 431–441 (2022). https://doi.org/10.1007/978-3-031-16449-1_41

16. Yang, C., Wang, K., Wang, Y., Dou, Q., Yang, X., Shen, W.: Efficient deformable tissue reconstruction via orthogonal neural plane. IEEE transactions on medical imaging (TMI) (2024). https://doi.org/10.1109/TMI.2024.3388559
17. Yang, C., Wang, K., Wang, Y., Yang, X., Shen, W.: Neural lerplane representations for fast 4d reconstruction of deformable tissues. In: Medical Image Computing and Computer Assisted Intervention (MICCAI) (2023). https://doi.org/10.1007/978-3-031-43996-4_5
18. Zha, R., Cheng, X., Li, H., Harandi, M., Ge, Z.: Endosurf: Neural surface reconstruction of deformable tissues with stereo endoscope videos. In: Medical Image Computing and Computer Assisted Intervention (MICCAI). pp. 13–23 (2023). https://doi.org/10.1007/978-3-031-43996-4_2
19. Zheng, Y., Harley, A.W., Shen, B., Wetzstein, G., Guibas, L.J.: Pointodyssey: A large-scale synthetic dataset for long-term point tracking. In: Internation Conference on Computer Vision (ICCV) (2023)