LiteTracker: Leveraging Temporal Causality for Accurate Low-latency Tissue Tracking

Mert Asim Karaoglu[1,2], Wenbo Ji[1,2], Ahmed Abbas[1], Nassir Navab[2], Benjamin Busam[2], and Alexander Ladikos[1]

[1] ImFusion GmbH, Munich, Germany
karaoglu@imfusion.com
[2] Computer Aided Medical Procedures, Technische Universität München, Germany

Abstract. Tissue tracking plays a critical role in various surgical navigation and extended reality (XR) applications. While current methods trained on large synthetic datasets achieve high tracking accuracy and generalize well to endoscopic scenes, their runtime performances fail to meet the low-latency requirements necessary for real-time surgical applications. To address this limitation, we propose LiteTracker, a low-latency method for tissue tracking in endoscopic video streams. LiteTracker builds on a state-of-the-art long-term point tracking method, and introduces a set of training-free runtime optimizations. These optimizations enable online, frame-by-frame tracking by leveraging a temporal memory buffer for efficient feature re-use and utilizing prior motion for accurate track initialization. LiteTracker demonstrates significant runtime improvements being around 7× faster than its predecessor and 2× than the state-of-the-art. Beyond its primary focus on efficiency, LiteTracker delivers high-accuracy tracking and occlusion prediction, performing competitively on both the STIR and SuPer datasets. We believe LiteTracker is an important step toward low-latency tissue tracking for real-time surgical applications in the operating room. Our code is publicly available at https://github.com/ImFusionGmbH/lite-tracker.

Keywords: Tissue Tracking · Point Tracking · Endoscopy

1 Introduction

Accurate tissue tracking is an essential component for a wide range of surgical robotics and extended reality (XR) applications [20,23,12,22]. Endoscopic videos present a unique set of challenges for this task including strong non-rigid deformations, self-occlusions, viewpoint changes, abrupt camera motions, and further occlusions caused by surgical tools [10]. In addition to aforementioned challenges, employed algorithms must meet strict low-latency requirements due to the real-time demands of intraoperative applications.

Prior methods widely explore frame-to-frame motion for tissue tracking. Sparse [17,18] and dense [8] optical and scene flow-based methods have been developed to track tissue motion in 2D and 3D. While these approaches effectively capture short-term motion, their restricted temporal context prevents

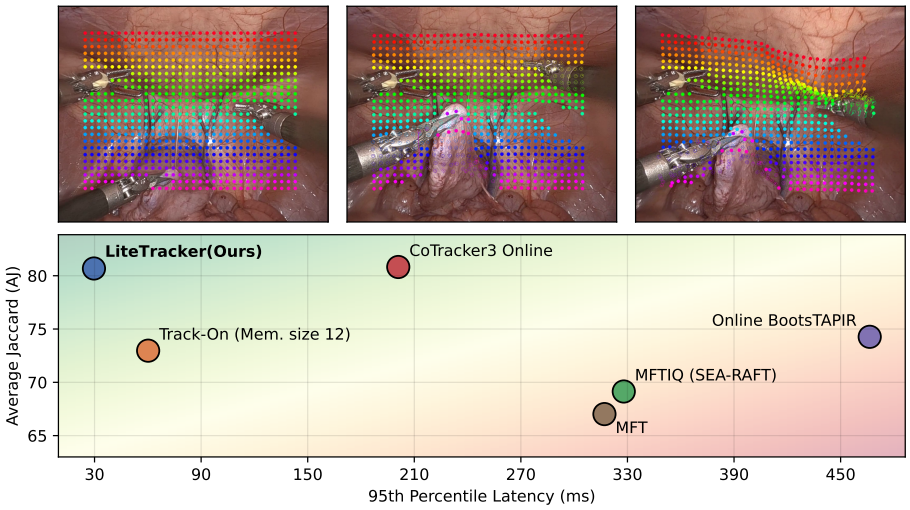

Fig. 1. (Top) Demonstration of LiteTracker on a video from STIR Challenge 2024 dataset [16]. (Bottom) Latency and tracking accuracy comparison. Average Jaccard (AJ) metric is computed on SuPer dataset [11]. Latencies showcase the 95th percentile of inference step measurements for 1,024 points initalized at the first frame of a video with 615 frames. LiteTracker is approximately 2× faster than the fastest evaluated method, Track-On [1], and 7× than its predecessor, CoTracker3 [9], while exhibiting close to state-of-the-art tracking accuracy.

occlusion handling and makes them susceptible to drift over long sequences. This issue becomes particularly pronounced in endoscopic videos, where occlusions, viewpoint changes, and rapid camera motions frequently disrupt tracking [2].

Building on the foundations of Particle Videos [15], recent years have seen extensive research into long-term point tracking using both end-to-end [6,5,9,1] and hybrid [13,21,2] approaches. Compared to frame-to-frame methods, these models leverage a longer temporal context, improving tracking accuracy, enhancing robustness against drift, and handling occlusions more effectively. Trained on large realistic synthetic datasets and, in certain cases, fine-tuned on real world videos with weakly supervised training [5,9,2], they show successful generalization to endoscopic scenes [23,2]. However, their low inference speed hinder their suitability for real-time applications.

To this end, we introduce LiteTracker, a low-latency tissue tracking method tailored for real-time surgical applications. Building upon CoTracker3 [9], a state-of-the-art long-term point tracking approach, LiteTracker adopts a point-based representation and incorporates a set of training-free runtime optimizations.

These enhancements enable LiteTracker to be approximately 7 times faster than its predecessor [9] and 2 times than the fastest method evaluated in our experiments [1], as shown in Fig. 1.

In addition to its efficiency gains, LiteTracker demonstrates strong tracking and occlusion prediction performance, ranking competitively on both the STIR Challenge 2024 [16] and SuPer [11] datasets.

Our contributions are summarized as follows:

1. A temporal memory buffer for caching previously computed features, enabling efficient per-frame predictions.
2. Exponential moving average flow initialization for fast and robust tracking convergence without iterative updates.
3. A low-latency, state-of-the-art tissue tracking method designed for integration as a building block into real-time surgical applications.

2 Methodology

We build LiteTracker using CoTracker3 [9] as our foundation due to its high tracking accuracy and strong generalizibility to endoscopic scenes. In this section, we first give an overview of the CoTracker3 architecture and next, introduce our method, LiteTracker.

2.1 Overview of CoTracker3

CoTracker3's [9] online model offers an end-to-end, transformer-based architecture that processes a video stream in a sliding-window fashion. Given a video stream $(I_t)_{t=1}^{T}$ consisting of T frames, and a set of query points Q containing queries $q = (t^q, x_{t^q}, y_{t^q})$ initialized at frame t^q, CoTracker3 predicts the visibility score $V_t \in [0, 1]$, confidence score $C_t \in [0, 1]$ and location $P_t = (x_t, y_t) \in \mathbb{R}^2$ for each query q for all frames $t > t_q$ coming after their initialization.

For new frames, the initial location, visibility and confidence scores (P_t^{init}, V_t^{init}, C_t^{init}) are set to the estimated values of the previous frame.

The model operates in a sliding-window fashion, with a window size, T_W, of 16 frames and stride, S, of 8 frames. The tracking algorithm begins with a 2D convolutional neural network (CNN) to extract multi-scale feature maps $\boldsymbol{\Phi}_t$ of each frame in the window, spatially downsampled by factors of $\{4, 8, 12, 16\}$ w.r.t. the input resolution. These feature maps are used to extract multi-scale patch features $\phi_t^s \in \mathbb{R}^{7\times7\times d}$ around point locations. In the case of a query point, patch features $\phi_{t^q}^s$ are extracted around the query location P_{t^q} from $\boldsymbol{\Phi}_{t^q}$. These multi-scale query features are compared with the point features constructing a 4D correlation tensor [3], which is then passed through a multi-layer perceptron (MLP) to generate correlation features. The correlation features, current point estimates, confidence and visibility scores along with spatio-temporal positional encoding is passed to a transformer-based iterative refinement module. This module predicts the updates for confidence ΔC_t, visibility ΔV_t, and location ΔP_t for all frames in the window. In the next refinement iteration, the updated location estimates are used to re-extract the patch features ϕ_t^s, and the correlation features are recomputed. This process is repeated, as defined in the original implementation, $L := 6$ times, with the final estimates serving as the predictions.

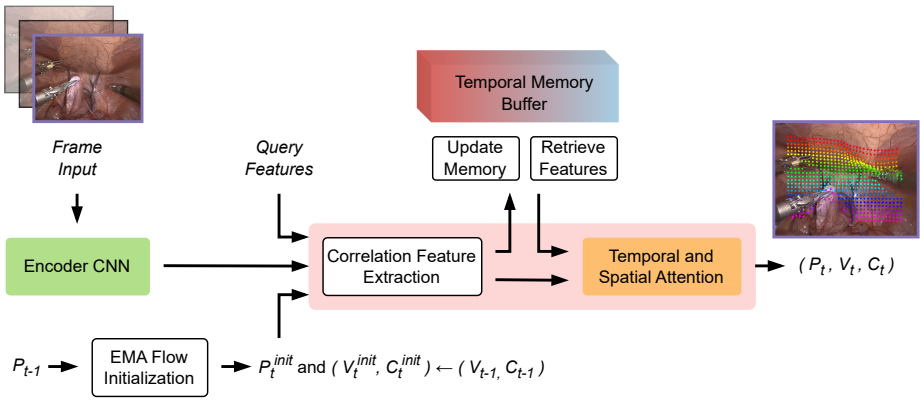

Fig. 2. LiteTracker's architecture. Given a video stream and a set of query points, we extract feature maps for each new frame and compute correlation features between the queries and points initialized with exponential moving average flow (EMA flow). We store these correlation features in a temporal memory buffer for efficient re-use on subsequent frames. Utilizing a transformer we propagate spatio-temporal information via attention mechanism to yield new point locations, visibility and confidence scores.

2.2 LiteTracker: An Efficient Solution

Despite its strong tracking performance, CoTracker3 [9] exhibits several efficiency limitations that hinder its deployment in real-time surgical applications. These bottlenecks arise primarily from its reliance on sliding-window processing and the iterative refinement module, both of which introduce computational overhead and latency. To this end, we introduce LiteTracker. LiteTracker proposes a set of training-free runtime optimizations, see Fig. 2, to enable low-latency, accurate tissue tracking.

Frame-by-frame Processing with Temporal Memory CoTracker3 [9] operates in a sliding-window manner requiring accumulation of a stride ($S := 8$) of frames before performing the next prediction. A straight-forward way to perform frame-by-frame prediction is to set the stride S to 1 instead of 8. This choice however, introduces a computational overhead since the processing of $T_W - S$ many frames need to be repeated multiple times. To tackle this challenge, we propose maintaining a temporal memory buffer to cache prior computations. Concretely, we cache the correlation features since their computation is relatively expensive, involving pair-wise similarity measurement of multi-scale patch features. To this end we maintain a ring buffer, with a capacity T_B of 16 frames, adhering to the original window size of CoTracker3. The ring buffer operates in a first-in, first-out (FIFO) manner where the newly processed frames overwrite the oldest ones. Our temporal memory buffer is expressly designed to allow the introduction of new query points at any time during tracking. When additional points are queried in later frames, a set of proxy feature vectors are appended to the buffer.

Table 1. Comparison of long-term point tracking methods on STIR Challenge 2024 [16] and SuPer [11] datasets. δ^{avg} and AJ indicate respectively the average tracking accuracy and average Jaccard metrics across a set of pixel thresholds and OA indicates the occlusion accuracy metric. "*" designates that for that method, its score on the STIR dataset are taken from the official pre- and post-challenge submission results. Latencies showcase the 95th percentile of inference step (frame or window) measurements for 1,024 points initalized at the first frame of a video of 615 frames on an NVIDIA RTX 3090.

Model	Input	STIR	SuPer		Latency (ms) ↓
		δ^{avg} ↑	AJ ↑	OA ↑	
CoTracker3 Online [9]	Window	75.24	**80.82**	<u>96.96</u>	200.98
Online BootsTAPIR [5]	Frame	68.59	74.27	93.63	466.38
MFT* [13]	Frame	**77.62**	67.02	83.73	317.05
A-MFST* [2]	Frame	58.59	–	–	–
MFTIQ (SEA-RAFT)* [21]	Frame	76.82	69.14	83.73	327.93
MFTIQ (ROMA)* [21]	Frame	<u>77.22</u>	66.61	83.73	4355.11
Track-On (Mem. size 12) [1]	Frame	72.74	72.97	86.86	<u>60.18</u>
Track-On (Mem. size 48) [1]	Frame	74.44	70.22	83.92	74.80
LiteTracker (Ours)	Frame	75.81	<u>80.68</u>	**97.45**	**29.67**

The transformer architecture dynamically masks these proxies during temporal and spatial attention, ensuring that they do not influence existing tokens while preserving the causality of the pipeline.

Track Initialization with Exponential Moving Average Flow Initializing the points on newly arrived frames by their prior locations, CoTracker3 originally performs multiple iterative refinement updates which includes computationally expensive components. Since these iterations are executed sequentially, they impose a linear increase in runtime. Inspired by prior works in optical flow [24], we introduce a simple yet effective, training-free strategy to improve the convergence of the iterative refinement module by providing a more accurate location initialization. We achieve this using an exponential moving average flow (EMA flow) scheme, F_t, formulated as

$$F_t = \alpha(P_{t-1} - P_{t-2}) + (1 - \alpha)F_{t-1}, \tag{1}$$

which is used to initialize the location for new frame as

$$P_t^{\text{init}} = P_{t-1} + F_t. \tag{2}$$

We empirically set the temporal smoothing factor α, to 0.8. Exponential moving average flow initialization enables fast and robust trajectory extrapolation, allowing accurate location estimation in a single pass (i.e., $L := 1$) of the refinement module.

3 Experiments

3.1 Implementation Details and Experimental Setup

In LiteTracker, we do not perform any training and directly use the pretrained model weights from CoTracker3 online [9], which were obtained by training on TAP-Vid Kubric [4] and then fine-tuning on a curated set of unlabeled real videos through weak supervision.

We conduct all experiments on a single NVIDIA RTX 3090 GPU using PyTorch [14], with half-precision enabled for all methods from the literature where recommended by the authors. For a fair comparison, we empirically found that CoTracker3 achieves its highest performance on tissue tracking when the number of iterative refinement steps, L, is set to 4 instead of the original value of 6. Therefore, for CoTracker3, we set $L := 4$ in all evaluations. Following the previous benchmarks [4], we utilize the suggested input size of each method for inference, in the case of LiteTracker, 512×384, and rescale the estimated tracks to the benchmark's evaluation size.

3.2 Evaluation

Datasets We compare LiteTracker with the state-of-the-art long-term point tracking methods on two different tissue tracking benchmarks [16,11].

For the evaluation on STIR dataset [19], we use its subset utilized for STIR Challenge 2024 [16]. This dataset consists of 60 sequences of different lengths, the longest one being up to 4 minutes. It includes sparse annotations, labeling the initial and end locations of tracked point in the initial and the last frame. It utilizes the average tracking accuracy metric (δ^{avg}) computed at the original image size, 1280×1024, with accuracy thresholds set to $\{4, 8, 16, 32, 64\}$ pixels. Since it lacks the ground truth location and occlusion labels for the intermediate frames, this benchmark only measure the tracking performance.

We additionally evaluate LiteTracker on the SuPer [11] dataset which consists of a single video of 522 frames, with manually annotated location and occlusion labels of 20 points for every 10^{th} frame. Leveraging the ground truth occlusion labels, we employ the average Jaccard (AJ) and occlusion accuracy (OA) metrics to measure the tracking and visibility prediction performance. Following their original implementation [4], we compute AJ on the image size of 256×256 with accuracy thresholds set to $\{1, 2, 4, 8, 16\}$.

Results We present our results in Tab. 1. For the STIR dataset, we report the official pre- and post-challenge submissions, denoted with "*", as these methods did not have direct access to the dataset, which may have limited their ability to fine-tune their models for better performance. For the remaining methods, we use publicly available implementations where possible; otherwise, unavailable results are marked with "–".

As highlighted in Tab. 1, LiteTracker delivers competitive tracking performance on the STIR dataset, significantly outperforming the previous fastest

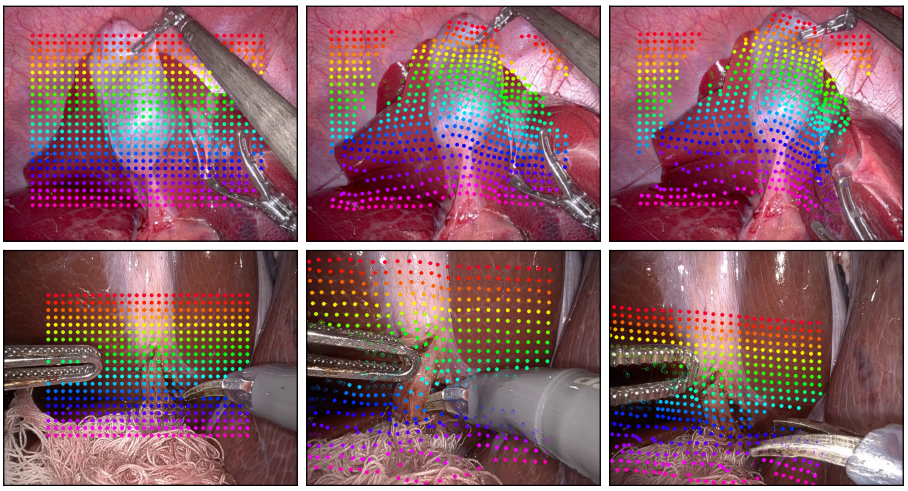

Fig. 3. Qualitative results on video samples from the STIR Challenge 2024 [16] (top) and StereoMIS [7] (bottom) datasets. LiteTracker shows high tissue-tracking accuracy and occlusion handling under challenging deformations, tool interactions and perspective changes.

state-of-the-art method that we evaluate on, Track-On [1]. On the SuPer dataset, LiteTracker ranks second in tracking accuracy, just behind CoTracker3 [9], while achieving the highest occlusion prediction score.

Despite its high accuracy, LiteTracker sets a new state-of-the-art in runtime efficiency, achieving an inference latency of 29.67 ms, measured as the 95th percentile over a 615-frame video tracking 1,024 points. This represents a substantial improvement over existing methods, running approximately 2 times faster than Track-On [1] and 7 times faster than its predecessor, CoTracker3 [9]. Moreover, when deploying CoTracker3 in a real-time application, the implicit latency introduced by frame accumulation for sliding-window processing must also be considered. For instance, with an input video stream running at a frame rate R of 30 Hz, this results in an additional delay of $(S-1) \times 1000 \div R$, (233.33 ms), leading to a total latency of 434.31 ms. This makes CoTracker3 approximately 16.6 times slower than our method.

Finally, Fig. 3 showcases LiteTracker's accurate tissue tracking under viewpoint changes, occlusions, and deformations on video samples from STIR Challenge 2024 [16] and StereoMIS [7] datasets.

Ablation Studies We conduct further experiments to analyze the impact of our contributions, providing their results in Fig. 4.

In our first study, we evaluate the EMA flow initialization's contribution to the overall tracking accuracy with respect to the varying number of refinement iterations. Our results, collected on the STIR Challenge 2024 dataset [16], show-

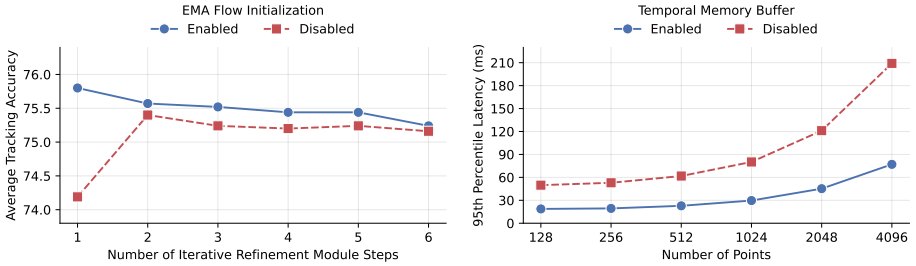

Fig. 4. Ablation studies. (Left) Exponential moving average flow (EMA flow) initialization improves tracking convergence, leading to its highest average tracking accuracy on the STIR Challenge 2024 dataset [16] with a single pass through the iterative refinement module. (Right) Temporal memory buffer improves runtime efficiency by approximately 2.7 times and provides low-latency inference during frame-by-frame tracking.

case that our EMA flow initialization already yields the model's highest tracking accuracy with a single pass through the iterative refinement module. We believe that our initialization scheme mimics temporal smoothness common to deformations in surgical scenes, there by providing a robust first estimate. Unlike CoTracker3, see Sec.3.1, further refinement iterations degrade the tracking accuracy of LiteTracker. We believe this can be attributed to the use of pre-trained weights.

In our second study, we analyze the effect of the temporal memory buffer on the runtime performance, tracking different number of points. Our results showcase that our caching mechanism consistently improves the runtime performance and results at approximately 2.7 times faster inference speeds. Furthermore, its compact size of the stored features results in negligible memory usage, 64 MB for 1,024 points, allowing them to be kept on the GPU for fast access.

4 Conclusion

In this paper, we propose LiteTracker, a low-latency tissue tracking method designed for real-time surgical applications. Building on CoTracker3, a state-of-the-art long-term point tracking method, LiteTracker incorporates a set of training-free runtime optimizations enabling efficient and accurate frame-by-frame tissue tracking. Our experimental results demonstrate that LiteTracker achieves substantial runtime improvements, running approximately 7× faster than its predecessor and 2× faster than the state-of-the-art, making it well-suited for real-time deployment. Furthermore, despite being designed for efficiency, LiteTracker maintains competitive tracking accuracy, coming close to the state-of-the-art on the SuPer and STIR datasets. We believe LiteTracker presents a significant step toward practical usage of low-latency, accurate tissue tracking and opens new avenues for future research utilizing it as a building block for real-time surgical applications.

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

References

1. Aydemir, G., Cai, X., Xie, W., Güney, F.: Track-on: Transformer-based online point tracking with memory. arXiv preprint arXiv:2501.18487 (2025)
2. Chen, Y., Wu, Z., Schmidt, A., Salcudean, S.E.: A-mfst: Adaptive multi-flow sparse tracker for real-time tissue tracking under occlusion. arXiv preprint arXiv:2410.19996 (2024)
3. Cho, S., Huang, J., Nam, J., An, H., Kim, S., Lee, J.Y.: Local all-pair correspondence for point tracking. In: European Conference on Computer Vision. pp. 306–325. Springer (2024)
4. 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. Advances in Neural Information Processing Systems **35**, 13610–13626 (2022)
5. Doersch, C., Luc, P., Yang, Y., Gokay, D., Koppula, S., Gupta, A., Heyward, J., Rocco, I., Goroshin, R., Carreira, J., et al.: Bootstap: Bootstrapped training for tracking-any-point. In: Proceedings of the Asian Conference on Computer Vision. pp. 3257–3274 (2024)
6. Harley, A.W., Fang, Z., Fragkiadaki, K.: Particle video revisited: Tracking through occlusions using point trajectories. In: European Conference on Computer Vision. pp. 59–75. Springer (2022)
7. Hayoz, M., 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 **18**(7), 1185–1192 (2023)
8. Ihler, S., Kuhnke, F., Laves, M.H., Ortmaier, T.: Self-supervised domain adaptation for patient-specific, real-time tissue tracking. In: Medical Image Computing and Computer Assisted Intervention–MICCAI 2020: 23rd International Conference, Lima, Peru, October 4–8, 2020, Proceedings, Part III 23. pp. 54–64. Springer (2020)
9. Karaev, N., Makarov, I., Wang, J., Neverova, N., Vedaldi, A., Rupprecht, C.: Cotracker3: Simpler and better point tracking by pseudo-labelling real videos. arXiv preprint arXiv:2410.11831 (2024)
10. Karaoglu, M.A., Markova, V., Navab, N., Busam, B., Ladikos, A.: Ride: Self-supervised learning of rotation-equivariant keypoint detection and invariant description for endoscopy. In: 2024 IEEE International Conference on Robotics and Automation (ICRA). pp. 10764–10771. IEEE (2024)
11. Li, Y., Richter, F., Lu, J., Funk, E.K., Orosco, R.K., Zhu, J., Yip, M.C.: Super: A surgical perception framework for endoscopic tissue manipulation with surgical robotics. IEEE Robotics and Automation Letters **5**(2), 2294–2301 (2020)
12. Liang, X., Liu, F., Zhang, Y., Li, Y., Lin, S., Yip, M.: Real-to-sim deformable object manipulation: Optimizing physics models with residual mappings for robotic surgery. In: 2024 IEEE International Conference on Robotics and Automation (ICRA). pp. 15471–15477. IEEE (2024)
13. Neoral, M., Šerỳch, J., Matas, J.: Mft: Long-term tracking of every pixel. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 6837–6847 (2024)

14. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al.: Pytorch: An imperative style, high-performance deep learning library. Advances in neural information processing systems **32** (2019)
15. Sand, P., Teller, S.: Particle video: Long-range motion estimation using point trajectories. International journal of computer vision **80**, 72–91 (2008)
16. Schmidt, A., Karaoglu, M.A., Sinha, S., Jang, M., Ha, H.G., Jung, K., Gu, K., Ullah, I., Lee, H., Šerỳch, J., et al.: Point tracking in surgery–the 2024 surgical tattoos in infrared (stir) challenge. arXiv preprint arXiv:2503.24306 (2025)
17. Schmidt, A., Mohareri, O., DiMaio, S., Salcudean, S.E.: Fast graph refinement and implicit neural representation for tissue tracking. In: 2022 International Conference on Robotics and Automation (ICRA). pp. 1281–1288. IEEE (2022)
18. Schmidt, A., Mohareri, O., DiMaio, S., Salcudean, S.E.: Sendd: Sparse efficient neural depth and deformation for tissue tracking. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. pp. 238–248. Springer (2023)
19. Schmidt, A., Mohareri, O., DiMaio, S., Salcudean, S.E.: Surgical tattoos in infrared: A dataset for quantifying tissue tracking and mapping. IEEE Transactions on Medical Imaging (2024)
20. Schmidt, A., Mohareri, O., DiMaio, S., Yip, M.C., Salcudean, S.E.: Tracking and mapping in medical computer vision: A review. Medical Image Analysis p. 103131 (2024)
21. Serych, J., Neoral, M., Matas, J.: Mftiq: Multi-flow tracker with independent matching quality estimation. arXiv preprint arXiv:2411.09551 (2024)
22. Shinde, N.U., Liang, X., Liu, F., Zhang, Y., Richter, F., Herbert, S., Yip, M.C.: Jiggle: An active sensing framework for boundary parameters estimation in deformable surgical environments. arXiv preprint arXiv:2405.09743 (2024)
23. Teufel, T., Shu, H., Soberanis-Mukul, R.D., Mangulabnan, J.E., Sahu, M., Vedula, S.S., Ishii, M., Hager, G., Taylor, R.H., Unberath, M.: Oneslam to map them all: a generalized approach to slam for monocular endoscopic imaging based on tracking any point. International Journal of Computer Assisted Radiology and Surgery pp. 1–8 (2024)
24. Wang, Y., Lipson, L., Deng, J.: Sea-raft: Simple, efficient, accurate raft for optical flow. In: European Conference on Computer Vision. pp. 36–54. Springer (2024)