Tracking Better, Tracking Longer: Automatic Keyframe Selection in Model-based Laparoscopic Augmented Reality

Kilian Chandelon[1,2*] and Adrien Bartoli[3,1,2]

[1*]EnCoV, Institut Pascal, UMR6602 CNRS, UCA, Clermont-Ferrand University Hospital, France.
[2]SurgAR - Surgical Augmented Reality, Clermont-Ferrand, 63000, France.
[3]Department of Clinical Research and Innovation, Clermont-Ferrand University Hospital, France.

*Corresponding author(s). E-mail(s): kilian.chandelon@gmail.com;
Contributing authors: adrien.bartoli@gmail.com;

Abstract

Purpose. We present a novel automatic system for markerless real-time augmented reality. Our system uses a dynamic keyframe database, which is required to track previously unseen or appearance-changing anatomical structures. Our main objective is to track the organ more accurately and over a longer time frame through the surgery. **Methods.** Our system works with an offline stage which constructs the initial keyframe database and an online stage which dynamically updates the database with new keyframes automatically selected from the video stream. We propose five keyframe selection criteria ensuring tracking stability and a database management scheme ensuring real-time performance. **Results.** Experimental results show that our automatic keyframe selection system based on a dynamic keyframe database outperforms the baseline system with a static keyframe database. An increase in number of tracked frames without requiring surgeon input is observed with an average improvement margin over the baseline of 11.9%. The frame rate is kept at the same values as the baseline, close to 50 FPS, and rendering remains smooth.

Conclusion. Our software-based tracking system copes with new viewpoints and appearance changes during surgery. It improves surgical organ tracking performance. Its criterion-based architecture allows a high degree of flexibility in the implementation, hence compatibility with various use cases.

Keywords: laparoscopy, keyframes, automatic selection, database management, model-based augmented reality, surgical navigation

1 Introduction

In laparoscopic surgery the surgeon uses small incisions and a camera to operate in the abdominal cavity. A major challenge is to accurately localise and visualise sub-surface anatomical structures, which can be alleviated by Computer-Assisted Surgery (CAS). A key tool in CAS is Augmented Reality (AR), which consists in displaying visual information such as the tumours available from preoperative data –MRI or CT– directly in the intraoperative view. Implementing AR efficiently requires markerless real-time organ tracking, which is yet unresolved in the general case and forms our main focus.

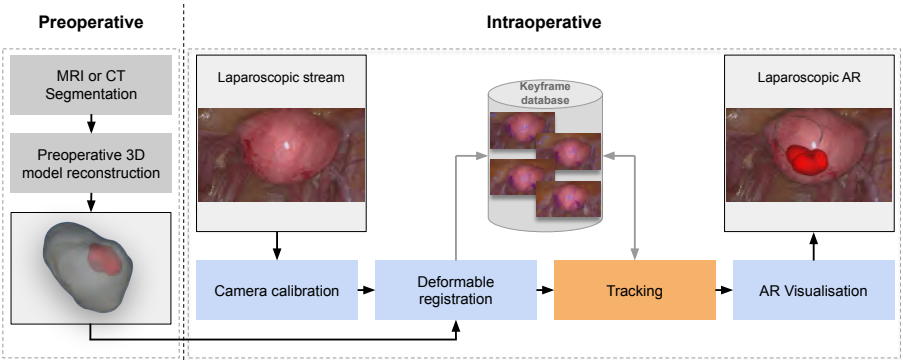

Fig. 1 Overview of the existing CAS-AR pipeline used in [1], exploiting a model-based tracking system. We use this pipeline and make a core technical contribution made to the tracking system.

Specifically, we endeavour to improve the state-of-the-art model-based tracking system already in use in several CAS-AR pipelines, such as [1]. We use the same CAS-AR pipeline as [1], which we reproduce in figure 1. The tracking system represents an essential part of this pipeline. It works by computing keypoint correspondences such as SIFT [2] between the current laparoscopic frame and a set of reference laparoscopic frames called keyframes, then computing camera pose from the correspondences. Because the keyframes are registered to preoperative 3D model in the prior steps of the CAS-AR pipeline, chaining the transformations then allows one to transfer information from the preoperative

3D model to the current frame, and fuse them to realise AR. The tracking system in [1] uses a database of keyframes constructed at the beginning of surgery. A major limitation is that the keyframe database is thus static, whereas new parts of the organ become visible and the organ appearance changes during surgery. This hinders the performance of organ tracking.

Our objective is to track the organ more accurately and on a longer timeframe through the surgery, by taking into account the greatest number of possible surgical events. We propose an improved tracking system with a dynamical keyframe database system, where the keyframes are automatically updated as and when needed. This is a challenging problem because the selection of keyframes is multi-criteria and critical, for adding the wrong keyframe causes tracking drift. The size of the database must be scalable, with the ability to forget keyframes, in order to maintain the real-time tracking performance.

Automatic keyframe selection is a problem encountered in many offline and real-time computer vision tasks, including camera pose estimation and 3D reconstruction with Simultaneous Localisation And Mapping (SLAM) [3, 4] or visual odometry [5]. The selected keyframes must cover the visible space but not be redundant. The keyframe system may be used with a vocabulary tree to organise the keypoint descriptors and speedup matching, as in [6, 7]. The core problem in keyframe selection is the definition of criteria to decide if the current frame can be appropriately considered to form a new keyframe. Existing criteria fail to address the complexity of the laparoscopic surgical environment. The problem of dynamic keyframe selection was overlooked in [1]. Our main contribution is a set of five criteria which, when combined, allows the tracking system to trigger the addition of a keyframe to the database.

2 Materials and Methods

2.1 *StaticDB*: baseline static keyframe database system

Our system is a re-implementation following the method and algorithm descriptions from [1]. Some slight implementation differences may thus exist. *StaticDB* has an offline and an online stage, as shown in figure 2. The *offline stage* constructs the static keyframe database. It assumes the organ remains rigid to a good extent during surgery. It uses Structure-from-Motion (SfM) [8, 9] to compute the keyframe poses and Iterative Closest Point (ICP) to compute the deformable transformation with respect to the preoperative 3D model. The keyframes are then preprocessed with resizing, channel selection and gaussian blurring, and keypoints extracted using POPSIFT [10], a GPU implementation of SIFT. Finally, the keypoint descriptors are stored in a multi-vector database.

The *online stage* processes the current frame of the video stream in real-time. The frame is first preprocessed similarly to the keyframes. Keypoints are then extracted with POPSIFT and matched to the keypoint database using Brute Force Matching (BFM). Finally, camera pose is computed from

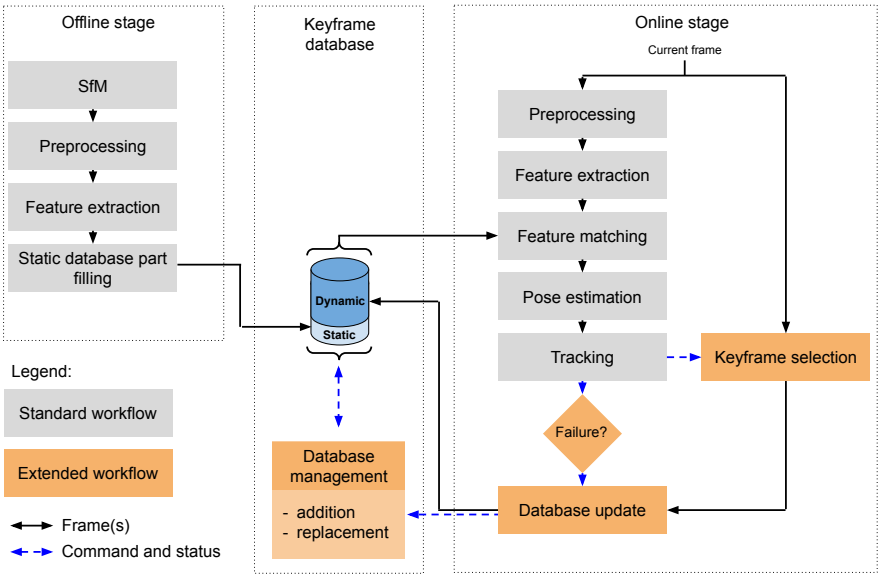

Fig. 2 Existing tracking system from [1] with a static keyframe database and its proposed extension with a dynamic keyframe database

the correspondences by a RANSAC-PnP, whose inliers are refined by a second PnP, and AR displayed.

2.2 *DynamicDB*: proposed dynamic keyframe database system

We extend *StaticDB* with a dynamic database part using two new steps: keyframe selection and database management. We keep the keyframe database's static part to ensure that the overall system does not drift by completely losing long-term focus on the organ.

The *keyframe selection step* selects new keyframes from the video stream. It tests the current frame with five necessary criteria. The first two criteria determine if the database lacks knowledge of the surgical field covered by the frame. C1 is passed if the number of keypoint matches to the keyframe database is lower than 50 and C2 is passed if the number of RANSAC-PnP pose estimation inliers is lower than 8. The last three criteria determine if the frame is good enough in terms of pose and focus. The pose quality involves two criteria. C3 is passed if the pose estimation root mean square reprojection residuals is lower than 0.4 % of the image diagonal and C4 is passed if camera jerk in normalised SfM units is lower than 1 s^{-3} (high jerk, hence high frequency, is typical of pose instability). The focus quality involves C5, which is passed if the frame Laplacian variance is lower than 2.9 % of the maximum image intensity. Each criterion acts on a specific parameter of the keyframe selection module. They

are not directly correlated and the thresholds they involve were thus chosen on an individual basis with regard to the expected results with empirical trials.

The selection of a keyframe is the result of a sequential validation of the above criteria. Being necessary, they are combined with the 'and' operator, making their order of test only important for algorithmic efficiency. Indeed, some criteria have a higher computation time, as shown in table 1. In our implementation, we thus check them in ascending order of computation time, *i.e.*, C1, C2, C3, C4 and lastly C5.

Table 1 Computational time of the five criteria used in our proposed dynamic keyframe database system named *DynamicDB*. The tests ran on a PC with Linux, CPU Intel i9-10900K and GPU Nvidia RTX 2080Ti. The average value, standard deviation σ, minimum and maximum computation time are given for each criteria.

	Computation time (μs)			
Criteria	Average	σ	Min	Max
C1	0.020	0.004	0.017	0.118
C2	0.021	0.003	0.016	0.038
C3	8.628	7.719	3.019	56.513
C4	15.046	4.852	10.977	80.017
C5	6799.952	353.936	6395.590	7664.350

Lastly, we add a keyframe to the database only if tracking failed for three consecutive frames, which was observed to reduce the number of undesired selections.

The *database management step* updates the dynamic database part with the selected keyframes so as to maintain real-time tracking performance. Obviously, the more keyframes, the slower the processing. We thus determine a maximal size N of the database a priori. This size depends on many factors, including the system implementation and the hardware capacity; we determine it experimentally. The database management step adds the selected frame and drops out a past keyframe if the database size exceeds N. The drop out criterion takes into account the amount of time which passed since a keyframe was last used, allowing the replacement of the keyframes which happen to no longer be relevant to tracking.

3 Experimental Results and Discussion

We have compared *DynamicDB* to our re-implementation of *StaticDB* on seven laparoscopic videos, showing the uterus with rigid and non-rigid movements with instruments generating smoke and bleeding. Our main evaluation criterion is the number of frames for which the uterus is tracked. We ran the tests on a PC with Linux, CPU Intel i9-10900K and GPU Nvidia RTX 2080Ti. We found the maximum size of the dynamic database to be $N = 30$ for a framerate greater than 25 fps.

The experimental results are shown in table 2. A general increase in number of tracked frames is observed with *DynamicDB*, with an average improvement margin over *StaticDB* of 11.19 %. The observed differences depend on the nature of the surgical scene and the movements of the organ. We have observed a stronger improvement when the organ deforms and a larger area is explored. As these experiments are retrospective, they do not take the user reaction to the tracking failure into account, which could lead to an even stronger increase. In other words, *DynamicDB* could modify the user's behaviour and lead to an even greater number of frames being tracked. As an example, a tracking failure, caused by a failure to match the current frame to the database, interrupts the AR display. Should they require AR, the user would react by naturally reverting to a spatial configuration for which AR was earlier shown. Furthermore, the improvement margin is also strongly linked to the static part of the keyframe database filled with SfM data: these keyframes are strong elements that cannot be modified and therefore impact the initialisation and the results.

Table 2 Comparison statistics in number of tracked frames.

Video index	Number of frames	Frames tracked with *StaticDB* (%)	Frames tracked with *DynamicDB* (%)	Difference (%)
1	1892	55.57	75.15	+19.58
2	1554	95.75	95.91	+00.19
3	2207	32.02	73.17	+41.15
4	1475	80.24	80.30	+00.06
5	5484	73.09	77.94	+04.84
6	0514	85.78	96.09	+10.31
7	3250	96.12	98.34	+02.22

Figure 3 gives details on video #3, where the laparoscope and the uterus are moved to extreme positions. *StaticDB* failed after 45 seconds while *DynamicDB* succeeded for all frames where the organ is sufficiently rigidly related to its model. Both methods are equivalent in terms of framerate, with averages of 47 fps and 46 fps respectively, well above the 25 fps of the video stream and we did not observe a visual difference in terms of rendering smoothness.

4 Conclusion

Our new tracking system shows that dynamical model update improves organ tracking. Specifically, it allows tracking to succeed in spite of new viewpoints and dynamic appearance changes. Our system is entirely software-based, it can thus easily integrate any CAS-AR suite and be extended with multiple criteria depending on the use case. We plan to improve our system with additional selection criteria, especially blur quantification to achieve invariance to organ texturing, and adding an explicit representation of the spatial coverage of

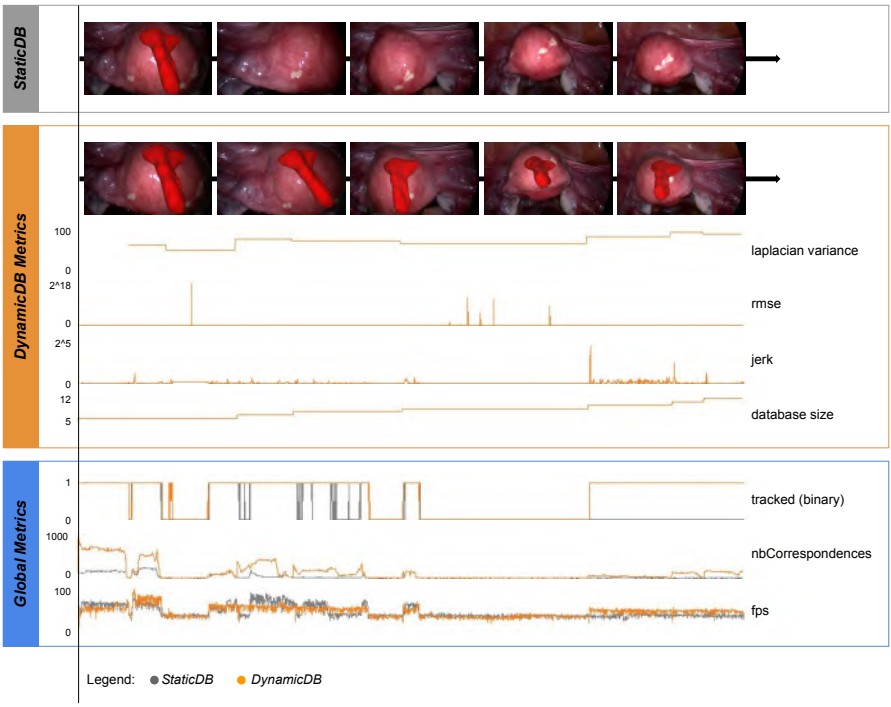

Fig. 3 Detailed comparison between *StaticDB* and *DynamicDB* for laparoscopic video #3. The top gray box shows AR results from *StaticDB* [1]. The middle orange box shows results from the proposed *DynamicDB*. It first shows the AR results, which are qualitatively successful and substantially more persistent than for *StaticDB*. It then shows the quantities involves in the proposed keyframe selection criteria and the keyframe database size. Finally, the bottom blue box shows global metrics comparing *StaticDB* and *DynamicDB*.

keyframes. Finally, we plan to evaluate our system further during in vivo AR-assisted laparoscopic surgery.

5 Disclosures and declarations

Conflict of Interest: The authors declare that they have no conflict of interest.

Ethical approval: All procedures performed in studies involving human participants were in accordance with the ethical standards of the institutional and/or national research committee and with the 1964 Helsinki declaration and its later amendments or comparable ethical standards. For this type of study formal consent is not required. This article does not contain any studies with animals performed by any of the authors.

Informed consent: Informed consent was obtained from all individual participants included in the study.

References

[1] Collins T, Pizarro D, Gasparini S, Bourdel N, Chauvet P, Canis M, Calvet L, Bartoli A (2021) Augmented Reality Guided Laparoscopic Surgery of the Uterus. IEEE Transactions on Medical Imaging, 40, 371-380. https://doi.org/10.1109/TMI.2020.3027442

[2] Lowe DG (2004) Distinctive Image Features from Scale-Invariant Keypoints. International Journal of Computer Vision, 60, 91-110. https://doi.org/10.1023/B:VISI.0000029664.99615.94

[3] Mahmoud N, Cirauqui I, Hostettler A, Doignon C, Soler L, Marescaux J, Montiel J.M (2016) ORBSLAM-Based Endoscope Tracking and 3D Reconstruction. ArXiv, abs/1608.08149. https://doi.org/10.1007/978-3-319-54057-3_7

[4] Lamarca J, Parashar S, Bartoli A, Montiel J.M (2021) DefSLAM: Tracking and Mapping of Deforming Scenes From Monocular Sequences. IEEE Transactions on Robotics, 37, 291-303. https://doi.org/10.1109/TRO.2020.3020739

[5] Recasens D, Lamarca J, F'acil J.M, Montiel J.M, Civera J (2021) Endo-Depth-and-Motion: Reconstruction and Tracking in Endoscopic Videos Using Depth Networks and Photometric Constraints. IEEE Robotics and Automation Letters, 6, 7225-7232. https://doi.org/10.1109/LRA.2021.3095528

[6] Dong Z, Zhang G, Jia J, Bao H (2009) Keyframe-based real-time camera tracking. 2009 IEEE 12th International Conference on Computer Vision, 1538-1545. https://doi.org/10.1109/ICCV.2009.5459273

[7] Dong Z, Zhang G, Jia J, Bao H (2014) Efficient keyframe-based real-time camera tracking. Comput. Vis. Image Underst., 118, 97-110. https://doi.org/10.1016/j.cviu.2013.08.005

[8] AliceVision "Meshroom: A 3D reconstruction software" (2018) [Online]. Available: https://github.com/alicevision/meshroom

[9] Griwodz C, Gasparini S, Calvet L, Gurdjos P, Castan F, Maujean B, De Lillo G, Lanthony Y (2021) AliceVision Meshroom: An open-source 3D reconstruction pipeline. Proceedings of the 12th ACM Multimedia Systems Conference. https://doi.org/10.1145/3458305.3478443

[10] Griwodz C, Calvet L, Halvorsen P (2018) Popsift: a faithful SIFT implementation for real-time applications. Proceedings of the 9th ACM Multimedia Systems Conference. https://doi.org/10.1145/3204949.3208136