Computers & Graphics 110 (2023) 125–140

Contents lists available at ScienceDirect

Computers & Graphics

journal homepage: www.elsevier.com/locate/cag

Technical Section

Mixed Reality Annotation of Robotic-Assisted Surgery videos with real- time tracking and stereo matching☆

Cristina Portalés [a], Jesús Gimeno [a], Antonio Salvador [b], Alfonso García-Fadrique [c], Sergio Casas-Yrurzum [a,*]

[a] Institute of Robotics and Information Technology and Communication (IRTIC), University of Valencia, Valencia, Spain
[b] General and Gastrointestinal Surgery. Fundación Investigación Consorcio Hospital General Universitario de Valencia (FIHGUV), Valencia, Spain
[c] General and Gastrointestinal Surgery. Valencian Institute of Oncology (IVO), Valencia, Spain

ARTICLE INFO

Article history:
Received 9 March 2022
Received in revised form 15 December 2022
Accepted 16 December 2022
Available online 23 December 2022

Keywords:
Mixed Reality
Annotation
Robotic-Assisted Surgery
Tracking
Stereo matching

ABSTRACT

Robotic-Assisted Surgery (RAS) is beginning to unlock its potential. However, despite the latest advances in RAS, the steep learning curve of RAS devices remains a problem. A common teaching resource in surgery is the use of videos of previous procedures, which in RAS are almost always stereoscopic. It is important to be able to add virtual annotations onto these videos so that certain elements of the surgical process are tracked and highlighted during the teaching session. Including virtual annotations in stereoscopic videos turns them into Mixed Reality (MR) experiences, in which tissues, tools and procedures are better observed. However, an MR-based annotation of objects requires tracking and some kind of depth estimation. For this reason, this paper proposes a real-time hybrid tracking–matching method for performing virtual annotations on RAS videos. The proposed method is hybrid because it combines tracking and stereo matching, avoiding the need to calculate the real depth of the pixels. The method was tested with six different state-of-the-art trackers and assessed with videos of a sigmoidectomy of a sigma neoplasia, performed with a Da Vinci® X surgical system. Objective assessment metrics are proposed, presented and calculated for the different solutions. The results show that the method can successfully annotate RAS videos in real-time. Of all the trackers tested for the presented method, the CSRT (Channel and Spatial Reliability Tracking) tracker seems to be the most reliable and robust in terms of tracking capabilities. In addition, in the absence of an absolute ground truth, an assessment with a domain expert using a novel continuous-rating method with an Oculus Quest 2 Virtual Reality device was performed, showing that the depth perception of the virtual annotations is good, despite the fact that no absolute depth values are calculated.

1. Introduction

Minimally invasive surgery (MIS) represents a milestone in the field of surgery. This type of surgical procedure is based on making small incisions in the patient's body, through which surgeons introduce different instruments and usually a laparoscopic camera. By avoiding large incisions, both the recovery time and infection risk are minimized. A step forward in MIS is Robotic-Assisted Surgery (RAS). In an RAS procedure, the surgeon does not directly handle the surgical tools, but rather controls robotic arms that are introduced into the patient's body through small incisions. RAS avoids the surgeon having to stand for a long time, as well as avoiding human hand tremor, and it offers surgeons the possibility of making movements that would be physically impossible if they had to hold the surgical material with their own hands. In addition, the surgeon acquires an enhanced perception of the surgical target, because stereoscopic cameras are usually employed with operating consoles that have stereoscopic viewers. Thus, the surgeon's movements become even more precise.

Despite the latest advances in the field of RAS, the steep learning curve of RAS devices still represents an obstacle to their evolution and deployment. Training in robotic surgery entails two fundamental aspects: skill and knowledge. Regarding the former, skill training is usually achieved using simulators [1,2], which allow the surgeon's skills to be polished without putting a patient at risk. Regarding the latter, a widely used resource is to record RAS procedures and then display them as teaching elements, allowing RAS trainees (future RAS-certified surgeons) to observe, from a first-person perspective, how to carry out RAS procedures. Since these types of surgeries are minimally invasive, and the patient's body is not exposed to permit the trainees to observe

☆ This article was recommended for publication by Anderson Maciel.
* Corresponding author.
E-mail address: Sergio.Casas@uv.es (S. Casas-Yrurzum).

https://doi.org/10.1016/j.cag.2022.12.006

the procedure live, videos are an extremely important teaching asset.

A very important factor in RAS is that the surgeon has a stereoscopic vision of the tissues; i.e., they perceive depth cues, unlike most non-robotic laparoscopic procedures which are usually mono. Thus, when using surgical videos for teaching RAS, these videos should be viewed by surgeons with a stereoscopic display in their training process, so that RAS trainees can observe the procedure as if they were performing it from the control console of the surgical robot itself. Otherwise, the teaching value of the video will be reduced. To this end, special devices such as Virtual Reality (VR) glasses or Head-Mounted Displays (HMD) are necessary, but no head tracking is necessary since the system displays stereoscopic videos.

On these stereoscopic videos of RAS procedures, it is important to be able to make virtual annotations in order to highlight certain aspects of the surgical process during the teaching session. Although these annotations could be fixed, static annotations have a reduced teaching value, so tracking-based dynamic annotations are preferable. The problem with these virtual annotations is that, as the video is stereoscopic, they need to be shown at the right depth, because if they are shown on the overlay plane, presence may be reduced due to accommodation problems, discomfort and dizziness. Therefore, in order to provide an augmented perception of the surgical process, the annotations should be spatially consistent with the rest of the elements shown in the video. This is the problem addressed by the MiRARAS project: *Mixed Reality Annotation for Robotic-Assisted Surgery*. This project researches methods and technologies to properly add virtual annotations onto videos of RAS procedures, using the Mixed Reality (MR) paradigm, by which real (tissues and surgical material) and virtual information (annotation) are properly blended.

In order to perform these virtual annotations, three seemingly different challenges need to be addressed: (i) the system needs to calculate the depth of the pixels shown in the stereoscopic video; (ii) the user (teacher) should be able to select any anatomical structure, and the system should be able to track it for the desired amount of time, so that an annotation can be correctly placed, highlighting the element of interest; (iii) the proper type of annotative element should be chosen and placed. The challenging nature of this problem is how to combine these three elements – depth estimation, tracking and annotation – so that meaningful annotations can be provided. If depth is not properly calculated, and/or the object is not properly tracked, the teaching value of the annotated video will be very much reduced. This process can be pre-calculated; i.e., the video can be annotated offline. Notwithstanding, it is much more useful for the teacher to be able to select an element of interest, choose an annotation and show it, properly tracked throughout the video, in real-time. Thus, we will focus on real-time solutions for this problem.

Annotation is an essential part of the MR paradigm, since MR applications allow the contextualization and placement of virtual information. Many previous MR works make use of annotations [3,4]. However, despite its importance, very few works have tried to focus or theorize on this subject, with the exception of [5], in which the authors propose and develop a taxonomy and a data model for AR annotation. The use of annotation in the surgical field is also not uncommon [6–10]. Some works have proposed annotation tools for endoscopic videos [11] without tracking. Others use tracking [12], or even tracking and object detection [13], but they do not work with stereoscopic videos nor do they show the annotations in MR.

Visual object tracking and stereo matching are two of the most recurrent research areas in computer vision. Despite recent progress, object tracking is still challenging due to occlusion, illumination variation, background clutter and other factors [14]. Currently, there is no single tracking algorithm that can handle all the factors in real-time [15]. In addition, an important problem with object tracking is that, although there are some evaluation tracking datasets and benchmarks, there is no standard method to evaluate trackers [14]. For this reason, we will explore the use of a new quality metric to assess the process, tailored to the stereoscopic domain.

In recent years, more and more researchers have been trying to explore Artificial Intelligence (AI) approaches [16–18], in order to improve tracking success and make them work well under sustained occlusion. The problem with these trackers is that separating the domain-specific information from the domain-independent information is very cumbersome. Thus, if the models are trained for outdoor tracking, as most models are, they will perform poorly in endoscopic scenarios. They also need to be trained with large image datasets, making them hard to adapt to specific scenarios. For this reason, our solution will not be based on AI approaches.

The use of tracking in endoscopic videos, MIS and RAS is also common. However, most of these works specialize in tracking either the surgeon's hands [19] or surgical tools [20,21]. Others are not constrained to these objects [22–24], but are not designed for stereoscopic videos and/or do not work in real time.

Stereo matching – the process of finding the pixels in a stereoscopic view that correspond to the same 3D point in the scene – and depth estimation represent two important topics in computer-assisted surgery. Although stereo matching and depth estimation are different problems, solving stereo matching often leads to depth estimation. Some depth estimation methods have also been proposed for MIS and RAS. However, the endoscopic domain is more challenging than outdoor scenes for a number of reasons: (i) the unfeasibility of providing a ground truth [25]; (ii) tissue deformation [26]; (iii) overexposure [27] and specular reflections [28]; (iv) small field of view [29,30]; (v) low contrast and textureless images; (vi) different brightness of the two stereo images. These problems make most traditional depth-estimation algorithms fail [31]. There are also depth estimation methods based on AI. Although some deep learning-based methods are promising, most of these algorithms assume no tissue deformation, no tool occlusion, a limited disparity range, or even no camera movement [32,33]. These solutions are also computationally expensive and/or require previous training for the specific endoscopic domain, which makes them as yet unsuitable for generic real-time RAS video annotation, which is the scope of our research.

Tracking and depth estimation are often treated separately. The same happens with the annotation problem. In fact, no previous frameworks for annotating stereoscopic videos – combining tracking and stereo matching – for MIS or RAS have been reported in the academic community. Thus, we believe our work is meaningful and novel. As will be explained later, our proposal is to treat these problems jointly in real-time, avoiding the need to obtain true depth estimations of the pixels of the RAS videos. As in [34–37] we will also compare different OpenCV trackers, although with different goals, scenarios, tracking methods and even evaluation procedures, since no previous works have been found exploring the use and comparison of these tracking methods for the stereoscopic RAS domain.

2. Materials and methods

The first and most obvious approach to this RAS-oriented annotation problem would be to treat it as two separate problems: (1) depth estimation – or even stereo-based reconstruction –; and (2) object tracking. With this approach, depth information would be obtained first from the stereo pairs of the video. Once depth

information is extracted for every pixel of the video, if an object can be properly identified and tracked, we could place a virtual annotation in the scene, in the correct place at the right depth, because we know how far the pixels of the object are from the camera. We can also track the object more easily, either in 2D or in 3D space.

However, both problems are computationally complex. Therefore, achieving a reliable solution for both of these problems that works in real-time is cumbersome. In addition, RAS images – and MIS images in general – are not always neat. These images are sometimes blurry because of occasional liquid splashes, smoke, occlusions with the laparoscopic tools and trocars, lack of contrast and poor focus. Thus, depth estimation is likely to fail and/or take too much time (or too much training data in the case of AI-based algorithms) to be obtained. If the desired RAS video annotation procedure is to be used by surgeons on average desktop computers, or laptops for teaching purposes, a lightweight reliable strategy is necessary.

In order to overcome these issues, we propose a different and simpler approach. This approach is based on considering both problems at once, since for the aims of this research it is not necessary to obtain a complete 3D reconstruction of the tissues, or even an estimation of the depth of all the pixels in the image. We also propose inverting the order of the steps, performing the tracking in 2D space first. Having a stereoscopic pair, we can use tracking and stereo matching methods to identify an object as it moves through time (with a tracking algorithm), but also as it "moves" (from the camera perspective) through space (with a stereo-matching method); i.e., we can first track the object as it moves from frame to frame as if the video stream were mono, and then we can perform localized stereo matching – only the region of interest (ROI) to be annotated is matched – within the stereoscopic pair. In general, epipolar lines need to be calculated, so that stereo matching is performed along them. However, it is common to keep the axes of stereoscopic cameras parallel [38] –since convergent toe-in stereo systems may cause the user discomfort [39]–. In these parallel setups, epipolar lines are horizontal, and vertical disparity should be zero. The final result of the process is the horizontal disparity of the tracked object.

Once we have the correct horizontal disparity of the ROI that we want to highlight, this information is enough to place a virtual annotation in this position. From this information we could potentially obtain the exact depth of the pixels if we have the intrinsic parameters of the camera, but this is not necessary. All we need to do is to place the annotation in the correct 2D position for both images of the stereoscopic pair. The position within the left image will be provided by the tracking algorithm. The position within the right image will be provided by a stereo matching procedure (matching, in the right image, the ROI found in the left image).

This method is fast, simple and can be implemented in web-based applications without the need to use GPU-based acceleration which may not be available to all users. The universality of access to the application is essential to achieve a useful and portable teaching tool for surgeons, who would like to have a simple-to-use method that requires no special hardware or software.

Although we could use a single tracking algorithm to track an object through time, and also through space, trackers are designed to analyze mono images that change over time. Therefore, feeding a tracker with both left and right images –alternately– of a stereo pair could be counterproductive. For this reason, we propose using a classical tracking algorithm to track the object – through time – only in the left image. Because of the high spatial correlation between the two images of a stereo pair, it is highly likely that the differences between the left and the right image would be small, so a stereo matching algorithm should be able to find the ROI in the right image with much more accuracy compared to using the tracker for the right image after using it for the left image of the same instant. This hybrid tracking–matching solution is much less likely to fail if the stereoscopic parallax is high, where tracking algorithms end up performing poorly if they are used for stereo matching. Preliminary tests have confirmed that a hybrid tracking–matching solution is much more stable than a tracking-only (interleaving left and right images and treating the video as a mono video for tracking purposes) approach. Moreover, by using stereo matching it is easier to ensure that the region found in the right image is the same size as the region in the left image.

In order to test our MR-oriented annotation method, we first prepared a Use Case using a video extracted from an RAS simulator. Then, we recorded two videos corresponding to two different real Use Cases, as will be explained later. Both of them were extracted from a real surgical procedure: a sigmoidectomy of a sigma neoplasia using a Da Vinci® X surgical robot. This surgical procedure was performed at the Hospital General de Valencia (Spain). As previously explained, we are interested in annotating these stereoscopic RAS videos in order to add suitable annotations to them and increase their didactic value.

3. Calculation

The proposed hybrid tracking–matching method can be summarized as follows:

Step 1- Load a stereoscopic RAS video.

Step 2- Select the first left frame of the video and allow the user to mark in it an ROI to track and annotate. The user can also optionally place a text over the ROI.

Step 3- Choose and initialize a tracker.

Step 4a- For each stereo pair, split the pair into left and right images.

Step 4b- For each left image, use the tracker to find the ROI.

Step 4c- For each stereo pair, use the tracked ROI of the left image to perform a stereo matching of the same region in the right image.

Step 4d- For each stereo pair, calculate horizontal and vertical disparity in order to analyze the success of the tracking procedure.

Step 4e- For each stereo pair, place the annotation at the corresponding left and right tracked locations of the ROI.

For the tracking step (step 4b), any tracking algorithm can be used. As will be explained later, the trackers implemented in OpenCV (Tracker class [40]) will be used in the assessment process. For the stereo matching algorithm (step 4c), we propose calculating the normalized correlation (*NC*) between the ROI from the left image and the right image, according to Eq. (1).

$$NC\,(x,y) = \frac{\sum_{i,j}\left(R\,(x+i,y+j)\cdot L_{ROI}\,(i,j)\right)}{\sqrt{\sum_{i,j}R(x+i,y+j)^2\cdot\sum_{i,j}L_{ROI}(i,j)^2}} \tag{1}$$

where:

R represents the right image, in which the ROI should be found/matched.

L_{ROI} represents the ROI (extracted from the left image) that we need to match in R.

x and y represent the coordinates of R at which the normalized correlation is calculated.

i and j represent the coordinates the algorithm uses to iterate through L_{ROI}.

The matched region in the right image will be the one that maximizes the value of *NC* with respect to the ROI of the left

image, and it is presumably the correct location to place the virtual element in the right image, whereas the correct location for the left image will be provided by the tracking algorithm applied to the stream of left images.

Algorithm 1 - Stereo_RAS_Annotation

Input:
A : *Annotation*
V : *StereoscopicVideo* // Unannotated video

Parameters:
annotationUpdateFrequency : *integer*

Output:
AV : *StereoscopicVideo* // Annotated video

Variables:
annotationCount, i, j : *integer*
F : *StereoscopicFrame*
L, R : *Frame*
ROI_L , ROI_R : *vector [ ] of BoundingBox*
T: *Tracker*

Algorithm:

```
AV = CreateStereoscopicVideo()
AV.fps = V.fps

annotationCount = AV.fps / annotationUpdateFrequency
F = GetStereoscopicFrame(V, 0)

(L, R) = SplitStereoscopicFrame(F)
ROI_L[0] = ChooseROI(L)

T = InitTracker()
i = 1
j = 0

while (i <= V.numFrames) do
{
    F = GetStereoscopicFrame (V, i)

    if (i mod annotationCount = 1) then
    {
        j = j + 1
        (L, R) = SplitStereoscopicFrame(F)
        ROI_L[j] = Track(T, L, ROI_L[j-1])
        ROI_R[j] = StereoMatch(R, ROI_L[j])
        CalculateAndPrintMetrics(ROI_L[j], ROI_R[j])
    }
    PlaceStereoAnnotation(F, A, ROI_L[j], ROI_R[j])
    AV.addFrame(F)

    i = i + 1
}
```

As we are interested in real-time solutions, this process can be further improved as, in order to perform proper annotation of objects, it is not always necessary to track the position of the object for every single frame of the stereoscopic video. Objects in RAS do not usually move fast. Thus, we can perform the tracking process skipping some frames. The lower the tracking update rate is, the less smooth the annotation will be. In addition, the lower the update rate is, the more probable it is that a quick displacement of the objects (or the camera) will cause a tracking failure. However, if we assume that objects do not move too fast, we could safely skip several frames of the video, keeping the annotation still for the frames we skip. This will likely increase the performance of the method without causing noticeable effects in the final annotation.

For optimization and practical purposes, other steps can be added, such as resizing the video or cropping the edges of the images, since some parts of RAS videos have no practical value. The final algorithm is shown in Algorithm 1, which is also summarized in Fig. 1.

As can be seen, the algorithm has only one intrinsic parameter: the annotation frequency, which defines how many frames per second are used to track the object. There could be additional parameters related to the tracking algorithm or the stereo matching procedure, but we will not address this question because it is beyond the scope of this study.

Since it is not possible to obtain an absolute ground truth for the objective evaluation of the performance of the algorithm, we collected the following datasets (for each annotated video):

-Tracker confidence for each frame (left image). This should be provided by the tracking algorithm. It is a number between 0 and 1.

-Stereo matching confidence for each frame (right image). This is the maximized value of *NC*. Its range is also from 0 to 1.

-Bounding boxes representing the ROI in both images of the stereo pair, as the desired object is tracked and matched for each frame.

From these datasets, we created the following metrics:

-*Tracking quality*: a heuristic measure to approximate the quality of the solution, which will be explained later.

-*Left-to-right horizontal disparity*: this measures the horizontal difference between the right and the left ROIs of the same stereoscopic pair. This should be small and change smoothly over time.

-*Left-to-right vertical disparity*: this measures the vertical difference between the right and the left ROIs of the same stereoscopic pair. In parallel stereo systems this should ideally be zero.

-*Left-to-right size disparity*: this measures the difference in square pixels between the size of the right and left ROIs of the same stereoscopic pair. This should ideally be zero.

-*Left temporal positional disparity*: this measures the number of pixels that the ROI of the left image moved in one iteration. This should be small and change smoothly over time.

-*Right temporal positional disparity*: this measures the number of pixels that the ROI of the right image moved in one iteration. This should also be small and change smoothly over time.

-*Left temporal size disparity*: this measures the number of square pixels that the area of the ROI of the left image changed in one iteration. This should be small or zero.

-*Right temporal size disparity*: this measures the number of square pixels that the area of the ROI of the right image changed in one iteration. This should be small or zero.

-*Total left motion*: this is the sum of the left temporal positional disparities over the duration of the video. It is expected that this quantity will be larger for poor trackers, since poor tracking often leads to jumps in the positions of the ROI.

-*Total right motion*: this is the sum of the right temporal positional disparities over the duration of the video. This should also be small.

-*Total left size change*: this is the sum of the left temporal sizes disparities over the duration of the video. This should ideally be zero.

-*Total right size change*: this is the sum of the right temporal sizes disparities over the duration of the video. This should ideally be zero.

These measurements were averaged (except from the last four metrics, which are already aggregated measurements) in order to obtain summarized numbers for each test. These averaged values will be used in the results section. However, it is important to

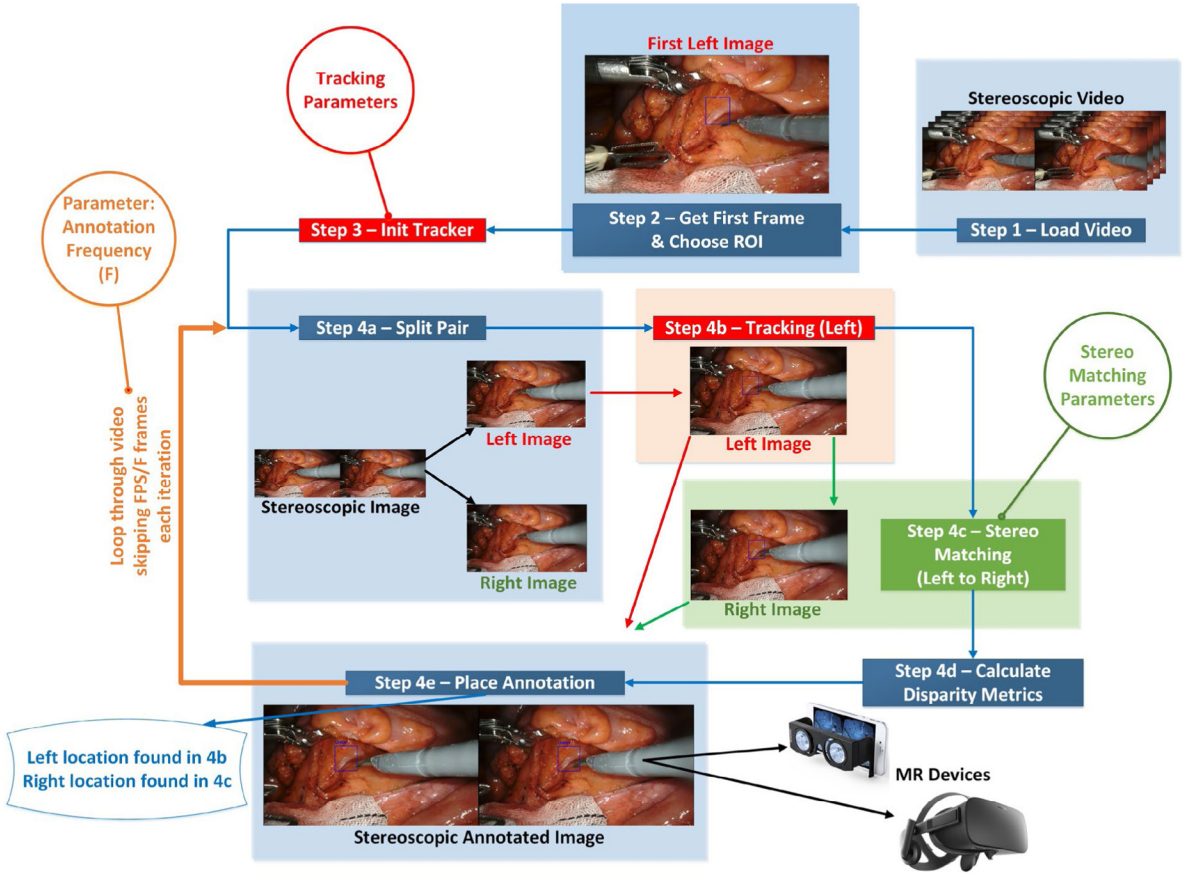

Fig. 1. Flowchart of the proposed method, showing steps 1 to 4.

note that, since disparities can be either positive or negative, we calculated the average (or sum) of the absolute values.

The first metric, tracking quality, is a heuristic measurement in the 0–6 range that we have developed in order to obtain a single number from each frame of the process. It assumes parallel stereo and its calculation obeys the following rules:

-Tracking quality value starts at 0 points.

-If vertical disparity is below a certain threshold, 2 points are added.

-If the x value of the left-to-right size disparity is below a certain threshold, 1 point is added.

-If the y value of the left-to-right size disparity is below a certain threshold, 1 point is added.

-If left temporal positional disparity is below a certain threshold, 1 point is added.

-If right temporal positional disparity is below a certain threshold, 1 point is added.

-The previous sum is multiplied by the tracking confidence.

The higher the tracking quality, the better, as happens with the values of tracking confidence and matching confidence.

4. Experiments and results

A threefold evaluation of the aforementioned method is proposed. First, an objective evaluation – based on the performance metrics explained earlier – was performed. The thresholds for the quality metric were setup at 2 pixels for the spatial disparities and 5 pixels for the temporal disparities. Next, we measured the computational load of the solution. Finally, a validation with a domain expert was also performed. To this end, a subjective but systematic rating procedure, which will be explained later, was used.

4.1. Experimental set-up

As aforementioned, three videos representing three different Use Cases were used:

-Use Case #0. In this Use Case, a RAS simulator developed by our research team was used in order to record a stereoscopic video and test our method under highly controlled conditions. The selected ROI will be easy to track with just some occlusions throughout the video. However, the video will be long enough to ensure that the method is stable. In this video, the user is trying to join the two ends of a severed artery. The ends are marked with a green and a blue dot, respectively. Fig. 2 shows this Use Case.

-Use Case #1. In this Use Case, the goal is to track the ureter during sigmoidectomy surgery. In this video, the ureter remains mostly visible – with fast and occasional occlusions caused by the Da Vinci® tools – and it is not manipulated, so it barely changes shape. The surgical importance of this annotation is to highlight the position of the ureter to remind surgeons that this small anatomical structure should be identified in this type of procedures in order to avoid unintentional damage to it. Ureteral injury, although rare in colorectal surgery, leads to severe complications. Thus, it is important to highlight this situation. Fig. 3 shows this Use Case.

-Use Case #2. In this Use Case, the goal is to highlight a colorectal anastomosis as it is performed during sigmoidectomy surgery. Anastomoses are common for digestive surgeons. However, from a computer vision point of view, tracking this process is really hard, because there are frequent occlusions and tissue deformations, which create constant changes in the region of interest. In addition, a circular stapling device suddenly appears

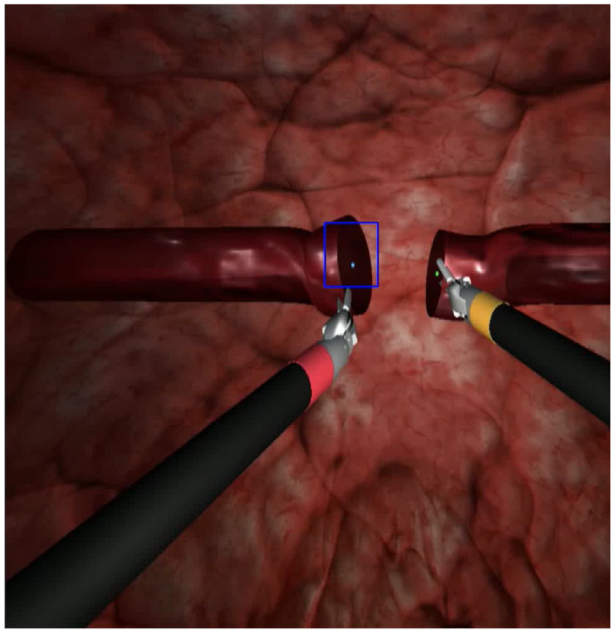

Fig. 2. First frame and ROI of Use Case #0 – Identifying the ends of a severed artery.

in the video after piercing the rectum, making the tracking quite challenging. Fig. 4 shows this Use Case.

The supplementary material contains the original unannotated videos. The videos from Use Cases #1 and #2 were captured with the laparoscopic parallel stereo camera of a Da Vinci® X surgical robot. Regarding the length of the annotations, we chose videos with durations longer than 30 s, because in preliminary tests with surgeons we could see that it would not be unusual to highlight and annotate moving elements for more than 15–20 s. In all cases, the testing method was similar. First, the domain expert – a surgeon with expertise in RAS – started the video to choose a region of interest and an annotation text. This ROI is intended to highlight a particular part of the surgical video, and represents the input of our annotation procedure. Once the ROI was chosen, six versions of the aforementioned hybrid tracking–matching approach were used and compared. Each version uses a different tracking algorithm. All of them use the same stereo matching algorithm explained earlier. The tracking algorithms used in these experiments are listed below, and their implementations can be found in the OpenCV library [40]:

-CSRT (Channel and Spatial Reliability Tracking) [41,42]
-KCF (Kernelized Correlation Filter) [43]
-Median Flow [44]
-MIL (Multiple Instance Learning) [45]
-MOSSE (Minimum Output Sum of Squared Error) [46]
-GOTURN (Generic Object Tracking Using Regression Networks) [17]

All these trackers were run using the same ROI, their default parameters in the OpenCV 4.5.3 implementation, the exact same videos and the same annotation update frequencies. Figs. 2, 3 and 4 show the first (left) image of the video and the ROI chosen by the domain expert for each Use Case.

We tested other trackers, such as TLD (Tracking, learning and detection) [47] and Boosting [48], but they performed very poorly for this problem.

4.2. Results

As aforementioned, a three-fold evaluation was performed. The first one is an objective assessment that includes a pseudo ground truth. The goal of this evaluation is to verify the feasibility of the proposed method in several scenarios and analyze how to choose trackers for this problem. The second one assesses the performance in order to verify the real-time capabilities of the solution. Finally, a subjective evaluation with a domain expert is presented in order to assess that the method is subjectively valid for its intended use.

4.2.1. Objective evaluation

First, we tested Use Case #0. We first used a RAS simulator for a number of reasons: (i) synthetic videos are perfectly aligned videos with zero vertical disparity; (ii) we can control the situation of the different objects (tissues, surgical instruments, etc.) so that we can decide when and how occlusions occur; (iii) unlike real RAS footage, the images from the simulator can be neat and clear; (iv) with a simulator we can control the stereoscopic parallax of the image. This is something that cannot be generally done with stereoscopic RAS cameras. The first row of Table 1 shows the parameters used to perform this test.

For this synthetic Use Case, only the CSRT method was used, since the goal is to validate the feasibility of using the proposed method. As seen in Table 2, the results report a good quality metric and high confidence values. It also reports small total motion values and only a slightly high vertical disparity (2.196) due to sporadic mistakes in the horizontal alignment. This value drops to 0.476 when no absolute values are used in the averaging process. This is acceptable, since the images should have zero vertical disparity. A look at the resulting video also shows that the tracker works well, following the highlighted element for the entire duration of the video.

Then, we tested Use Case #1. This Use Case is suitable to assess if our method is effective in a real case with surgical value. Table 1 shows the parameters used to perform this test. Table 3 provides a comparative summary of the performance of the aforementioned six tracking methods. We have also added a pseudo ground truth (PGT) solution, which is a solution created by the domain expert by a manual selection of the ROI over the video key frames, for both the left and the right images, keeping the size of the ROI fixed throughout the video. This is not really a ground truth, because the process is tedious and prone to errors, especially for the stereo matching, but it is a reasonable approximation. In this regard, we will process the PGT solution in a similar way to the rest of methods, i.e., computing the same metrics, and then we will compare the PGT metrics with the metrics of the methods. We also compute a metric of similarity between the PGT solution and the results obtained from the different versions of the proposed annotation solution. The metric we compute is Intersection over Union (IoU), as proposed in [49], which calculates the amount of relative overlapping between two solutions. IoU is proposed for mono. Thus, we extended this metric and calculated the IoU for both images of the stereo pair. The resulting value shown in Table 3 is the average of the two.

As seen in Table 3, the CSRT method outperforms the rest of trackers and offers a stable solution, which performs similar to that of the PGT. It provides the highest IoU, the highest quality metric, the highest tracking and matching confidence and the lowest total motion, both for the left and right channels. Indeed, the method is able to successfully track the object throughout the total length of the video, something that the rest of trackers are not able to do. KCF loses track when the camera is first moved quickly, which occurs around 6 s into the video. A similar situation occurs with Median Flow, although this tracker tries to recover the tracking, reporting an incorrect ROI for some time. The situation is not much better for MIL. In this case, the tracker does not lose track, but reports a ROI that is noticeably misplaced throughout much of the video. MIL is not able to identify that

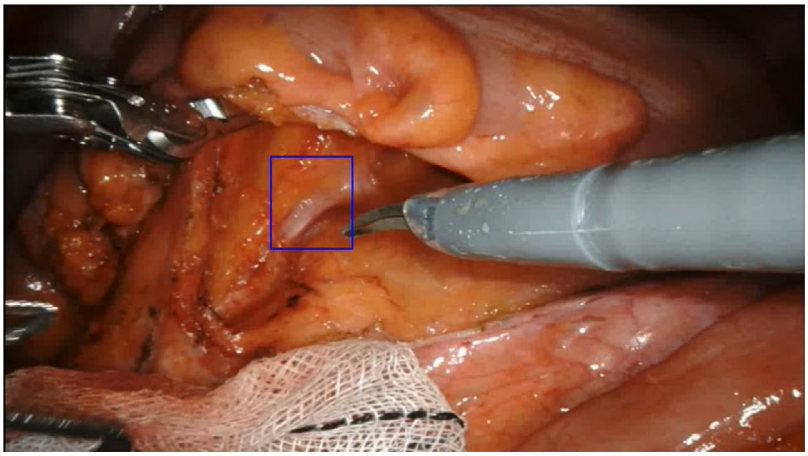

Fig. 3. First frame and ROI of Use Case #1 – Identifying the ureter in a sigmoidectomy.

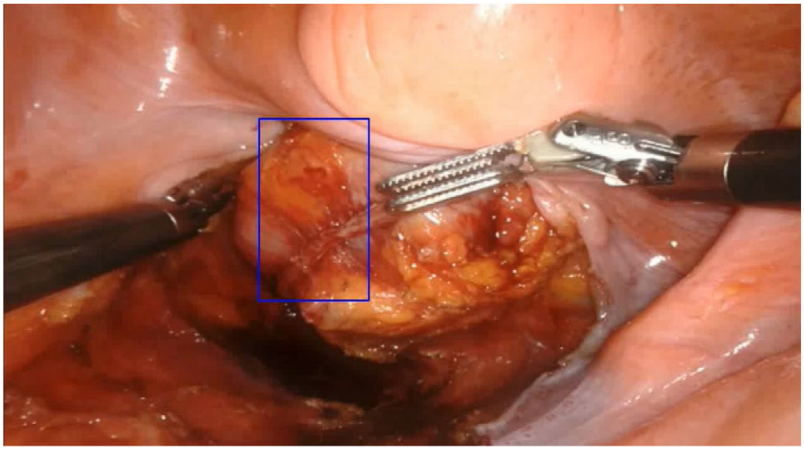

Fig. 4. First frame and ROI of Use Case #2 – Highlighting a colorectal anastomosis in a sigmoidectomy.

Table 1
Parameters of the three use cases.

Use case	Video duration [s]	Update interval [s]	Update frequency [Hz]	Resolution [pixels, fps]	Stereo resolution [pixels, fps]	ROI upper left corner [pixels]	ROI extent [pixels]
#0	72	0.1	10	1024 × 1024 @ 30 fps	2048 × 2048 @ 30 fps	(549, 357)	88 × 106
#1	30	0.1	10	1280 × 720 @ 25 fps	2560 × 720 @ 25 fps	(426, 243)	130 × 147
#2	60	0.1	10	1280 × 720 @ 25 fps	2560 × 720 @ 25 fps	(410, 190)	175 × 290

Table 2
Performance with the hybrid tracking–matching method using the CSRT tracker. Use Case #0.

Quality metric [0–6]	Average abs. vertical disparity [px]	Total motion (L) [px]	Total motion (R) [px]	Average tracker confidence (L)	Average matching confidence (R)
4.483	2.196	6,402.30	8,762.92	1.000	0.926

the tracking is wrong, and thus the confidence levels remain at maximum levels throughout the whole video, something that is misleading and causes the quality metric to be artificially high. A similar situation occurs with the MOSSE tracker, since the tracker is not reportedly failing, but the tracked ROI is wrong most of the time. In fact, the total motion values for this tracker are six times higher than with the CSRT tracker. GOTURN also performs poorly, as it loses track around 10 s into the video and tends to overexpand the ROI. None of the methods, with the exception of CSRT, obtain IoU values greater than 0.2, which shows that CSRT performs significantly better for this Use Case.

Figs. 5 and 6 show the evolution of both the quality metric and the vertical disparity over time. Red color means the value is lower than 4 for the quality metric, or greater than 2 for the absolute value of the vertical disparity. Fig. 7 shows how the ureter is properly tracked using the CSRT-based solution.

As seen in Fig. 6, a small (around 2 pixels) but systematic vertical disparity appears with the CSRT method. A similar situation occurs with MIL and MOSSE, whereas the other two methods offer almost zero average vertical disparity. Taking into account that only CSRT and – occasionally – MOSSE are able to track the ureter correctly, it is possible that the video presents a small vertical distortion. In order to confirm this, we calculated the

Table 3
Compared performance with the hybrid tracking–matching method. Use Case #1.

Tracking method	Quality metric [0–6]	Average abs. vertical disparity [px]	Total motion (L) [px]	Total motion (R) [px]	Average tracker confidence (L)	Average matching confidence (R)	IoU with respect to PGT
CSRT	5.750	2.003	1,000.01	1,138.82	1.000	0.998	0.688
KCF	1.117	0.383	1,307.68	1,435.57	0.205	0.204	0.176
MedFlow	2.136	0.516	4,707.18	4,866.98	0.383	0.381	0.138
MIL	4.824	2.681	2,351.39	2,592.33	1.000	0.987	0.154
MOSSE	5.497	1.859	5,736.32	5,956.95	0.976	0.967	0.178
GOTURN	3.971	2.144	4,979.49	5,156.63	1.000	0.982	0.096
PGT	5.731	2.040	875.70	940.54	–	–	1.000

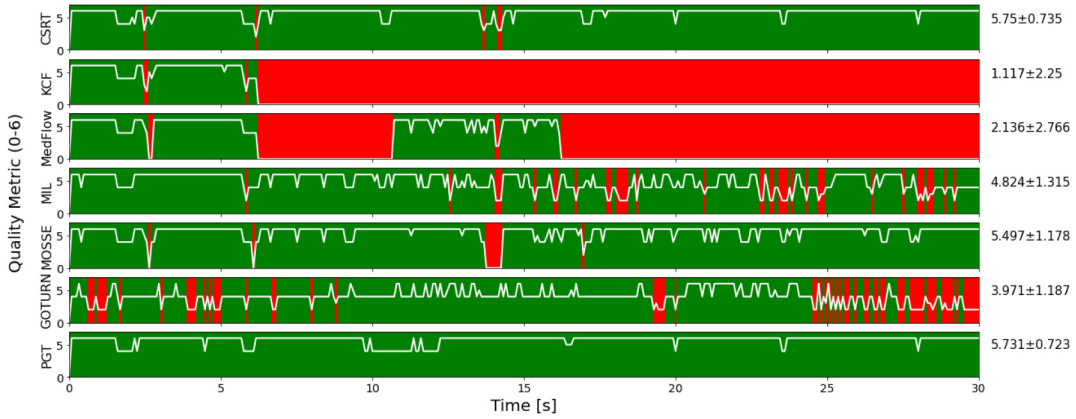

Fig. 5. Use Case #1 – Quality metric comparison (average ± standard deviation).

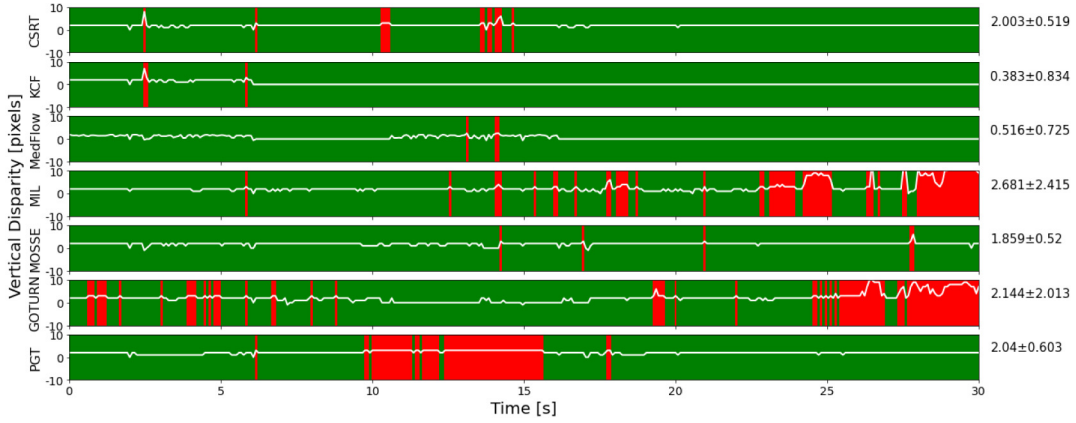

Fig. 6. Use Case #1 – Vertical disparity comparison (average ± standard deviation).

average vertical disparity (with no absolute values) for the CSRT tracker, and the value remained at 2.003. This means that the video of Use Case #1 likely contains a small vertical distortion caused by the Da Vinci® system. This is also confirmed by the value of the average absolute vertical disparity of the PGT, which is 2.04 and because it also occurs for Use Case #2. This, and the fact that the PGT calculation is not systematic and is prone to errors, explains the red zone of the PGT graph in Fig. 6.

Then, we tested Use Case #2. This Use Case entails a challenge for our tracking–matching method, since there are several occlusions, camera movements and also a lot of tissue deformation. Table 1 shows the parameters used for this Use Case. As can be seen, the same frequency as in Use Case #1 is used, but the video is twice as long.

As can be seen in Table 4, apparently only the MIL tracker works acceptably, as the IoU, the quality metric and the tracking and matching confidence values are the highest of all with this method. However, this is again misleading, since the truth is that the MIL tracker does not accurately follow the selected object. Instead, it drifts away and the result is offset with respect to the expected solution. The CSRT method, on the contrary, acknowledges a tracking failure after 15 s. Although it is able to properly track the object for the first 15 s, the value of the IoU and the quality metric are low, precisely because the tracker is able to detect a tracking failure and stops tracking, something that MIL does not do. KCF loses track after 8 s. Median Flow also loses track quickly, although it tries to recover the tracking, reporting many incorrect places, which leads to a high total motion value. A different situation occurs with the MOSSE-based method. This time, the method sometimes tracks the object well, and sometimes gets confused and tracks something else, such as the surgical tools. However, it is also capable of turning back to the right spot, which creates frequent jumps. These jumps are reflected in the total motion values, which are the highest of all. The quality metric and the confidence are high for this method. Finally, GOTURN proves to be even worse than in Use Case #1,

Table 4
Compared performance with the hybrid tracking–matching method. Use Case #2.

Tracking method	Quality metric [0–6]	Average abs. vertical disparity [px]	Total motion (L) [px]	Total motion (R) [px]	Average tracker confidence (L)	Average tracker confidence (R)	IoU with respect to PGT
CSRT	1.245	1.692	2,618.25	3,564.48	0.256	0.249	0.193
KCF	0.639	0.217	1,434.13	1,510.35	0.132	0.131	0.121
MedFlow	2.153	18.682	8,671.78	23,671.53	0.507	0.478	0.102
MIL	5.364	1.521	3,610.18	4,438.01	1.000	0.993	0.342
MOSSE	4.015	41.216	11,257.04	35,063.37	0.993	0.932	0.216
GOTURN	4.236	166.91	4,117.25	39,809.56	1.000	0.975	0.025
PGT	5.451	1.969	2,176.58	2,888.19	–	–	1.000

Table 5
Compared performance using CSRT with different update rates. Use Case #2.

Tracking method	Quality metric [0–6]	Average abs. vertical disparity [px]	Total motion (L) [px]	Total motion (R) [px]	Average tracker confidence (L)	Average tracker confidence (R)	IoU with respect to PGT
CSRT 10 Hz	1.245	1.692	2,618.25	3,564.48	0.256	0.249	0.193
CSRT 2 Hz	4.484	1.833	1,764.04	2,146.71	1.000	0.980	0.558
PGT	5.451	1.969	2,176.58	2,888.19	–	–	1.0

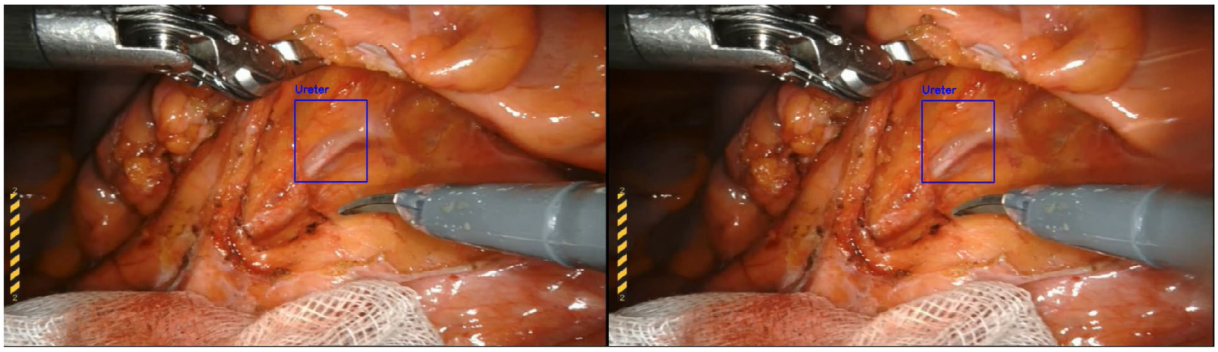

Fig. 7. A snapshot of Use Case #1 – Ureter properly tracked in both left and right images.

since the ROI drifts away from the target area from the beginning, and the size of the ROI keeps changing erratically.

Vertical disparity is only kept low for CSRT, KCF and MIL, although in the case of KCF this is because the tracking fails. This is also the reason why KCF obtains low values for total motion. All things considered, it seems that the best methods for this Use Case are the MOSSE and CSRT trackers, despite their IoU being around 0.2. The former is able to track the object sometimes, but jumps too much and presents an unacceptable vertical disparity for a parallel stereoscopic video (the second highest of all six trackers). The latter is able to properly track the object for a while, and obtains the closest results to the PGT in terms of disparity and motion. However, once the object is lost, it is never detected again. Fig. 8 shows two snapshots of the CSRT tracker tracking the ROI correctly.

After looking at the annotated videos and doing a series of quick tests, we noticed that by modifying the annotation update frequency the tracking success could change. Thus, we decided to increase the frequency. Surprisingly, there were very few improvements with this change, because most of the problems occur when occlusions happen. Sometimes, capturing more frames only makes things worse, as occlusions last longer and, thus, tracking success drops. Therefore, we decided instead to raise the annotation update interval. Indeed, with a lower update frequency the CSRT-based method performed extremely well, being able to track the object for the 60 s that the video lasted. The rest of the methods never achieved complete tracking of the object, neither when increasing nor decreasing the annotation frequency. Table 5 shows the difference between using a 10 Hz frequency (update interval of 0.1 s, as in Table 4) and a 2 Hz frequency (0.5 s). At 2 Hz, the CSRT tracker is able to track the object for the whole one-minute sequence, with substantial changes in the quality metric and in average confidence. Total motion is also reduced. Vertical disparity is just slightly augmented, because the longer the successful tracking time, the more likely a sporadic vertical mismatch will occur, something that is much less likely when tracking fails. The IoU metric also improves substantially. It does not reach higher values because the ROI of the PGT is fixed and the ROI of CSRT is not (in this test). Thus, the overlapping is smaller than expected. In addition, given the amount of tissue deformation, it is not easy to determine the correctness of a possible solution. As we can see, in this case the quality metric is a more reliable indicator than the IoU.

Figs. 9 and 10 show the evolution of both the quality metric and vertical disparity over time, including the two tests with the CSRT-based method (named CSRT10 and CSRT2 respectively).

4.2.2. Computation time

Although tracking quality is important, it is also necessary to evaluate the computational load of the proposed solutions. It is known that the selected tracking methods have different computational costs. However, it is important to measure how they behave in combination with stereo matching. To do so, we measured the computation time of the proposed hybrid solution under different circumstances. We assessed the three Use Cases and tested the performance of the annotation method for three different annotation update rates: 10 Hz (update the annotation 10 times per second), 2 Hz and 1 Hz. Tables 6–8 show the total computation time – and also the break down depicting the time to compute the tracking and the time to

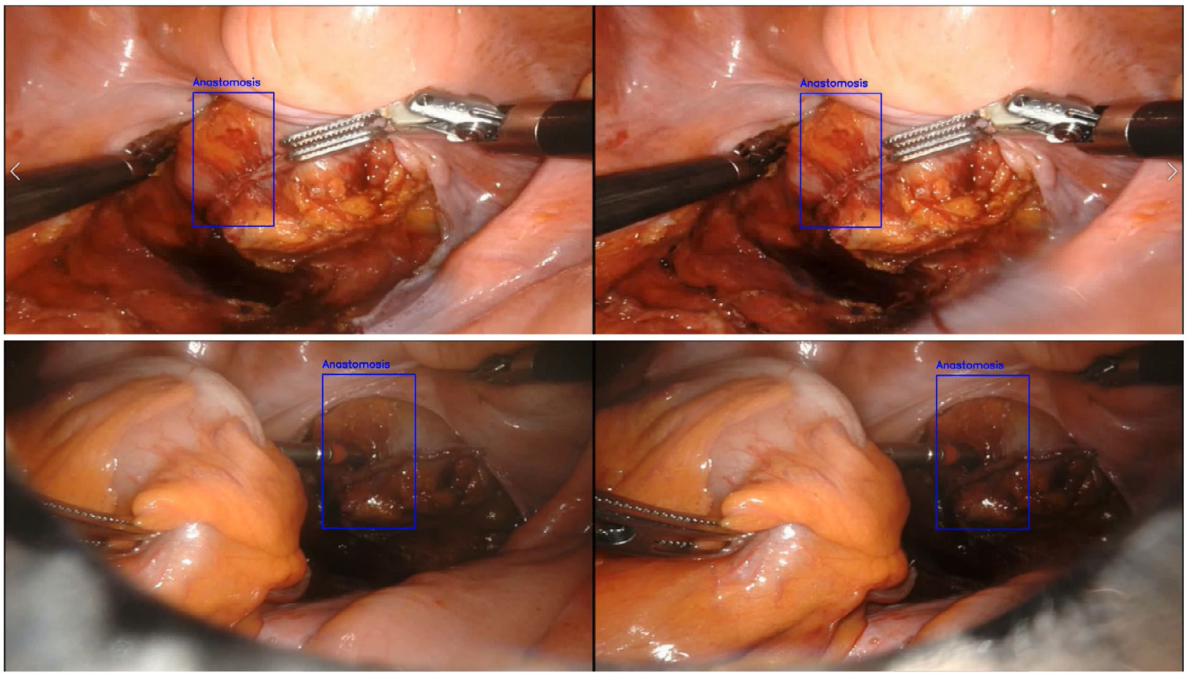

Fig. 8. Two snapshots of Use Case #2 — Anastomosis properly tracked in both left and right images.

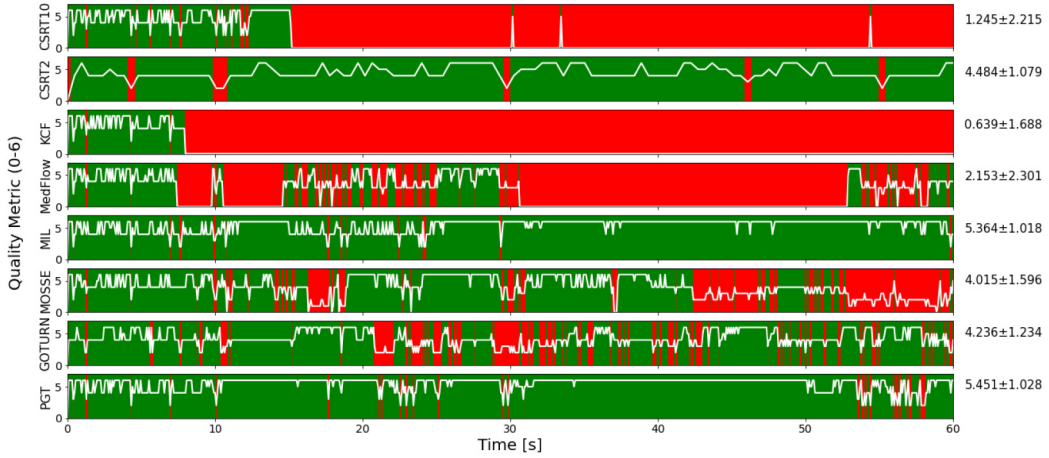

Fig. 9. Use Case #2 – Quality metric comparison (average ± standard deviation).

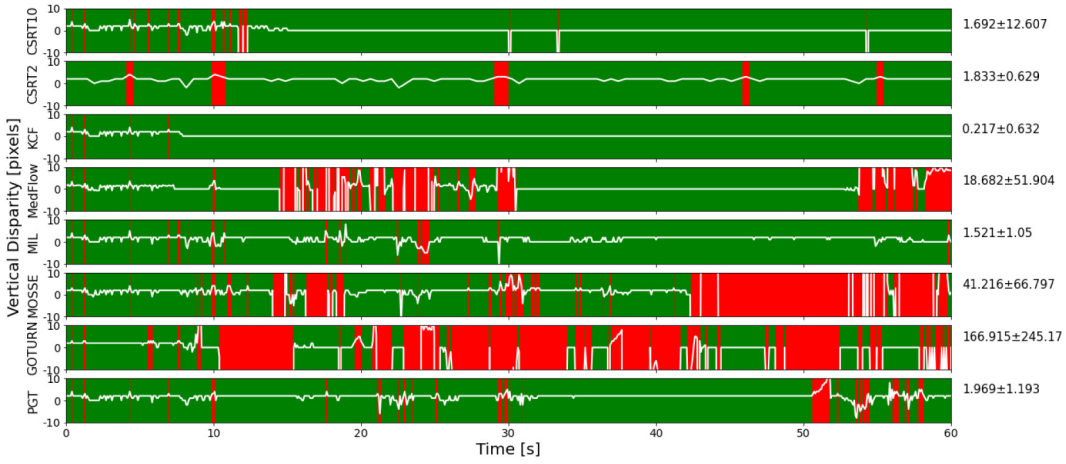

Fig. 10. Use Case #2 – Vertical disparity comparison (average ± standard deviation).

Table 6
Computational load (in seconds) for Use Case #0 (72 s).

Tracking method	Annotation frequency								
	1 Hz			2 Hz			10 Hz		
	Time [s]			Time [s]			Time [s]		
	Total	Tracking	Matching	Total	Tracking	Matching	Total	Tracking	Matching
CSRT	23.586	1.514	3.038	26.151	2.415	4.784	62.866	13.886	30.554
KCF	18.979	0.110	0.298	19.177	0.221	0.724	24.220	0.969	4.681
MedFlow	20.766	0.303	1.766	22.606	0.605	3.434	50.912	3.062	29.676
MIL	24.535	2.898	2.957	30.698	5.791	5.890	80.957	32.881	29.422
MOSSE	20.326	0.027	1.646	20.821	0.048	2.366	43.049	0.330	24.630
GOTURN	23.987	2.448	2.716	29.102	4.961	5.420	69.576	25.182	26.638

Table 7
Computational load (in seconds) for Use Case #1 (30 s).

Tracking method	Annotation frequency								
	1 Hz			2 Hz			10 Hz		
	Time [s]			Time [s]			Time [s]		
	Total	Tracking	Matching	Total	Tracking	Matching	Total	Tracking	Matching
CSRT	8.210	0.778	1.017	9.980	1.514	2.046	27.882	7.483	13.981
KCF	6.582	0.064	0.068	6.740	0.127	0.133	9.843	0.840	2.454
MedFlow	6.604	0.120	0.069	6.880	0.241	0.136	13.283	1.239	5.501
MIL	8.654	1.286	0.952	11.009	2.598	1.951	34.422	16.294	11.678
MOSSE	6.615	0.014	0.136	6.677	0.026	0.142	19.114	0.381	12.271
GOTURN	8.406	0.935	1.000	10.331	1.082	2.055	30.341	10.876	12.991

Table 8
Computational load (in seconds) for Use Case #2 (60 s).

Tracking method	Annotation frequency								
	1 Hz			2 Hz			10 Hz		
	Time [s]			Time [s]			Time [s]		
	Total	Tracking	Matching	Total	Tracking	Matching	Total	Tracking	Matching
CSRT	14.024	0.768	0.553	23.614	5.335	5.107	25.321	6.437	6.180
KCF	13.254	0.431	0.197	14.013	0.915	0.416	21.157	5.159	3.697
MedFlow	12.924	0.265	0.046	13.466	0.505	0.249	28.371	2.641	13.011
MIL	17.443	2.451	2.258	22.625	5.159	4.696	70.796	30.774	27.282
MOSSE	13.716	0.176	0.888	14.198	0.371	1.097	40.414	1.815	25.896
GOTURN	16.655	1.913	1.945	20.694	3.772	4.134	57.893	20.946	24.325

perform the stereo matching – for each Use Case and tracking method. Since each Use Case has a different video length, and different annotation frequencies were tested, it is important to take these two factors into account when comparing times. All the measurements were made with a desktop computer running an Intel Core i7-12700KF@3.6 Ghz processor with 32 Gb of RAM.

As can be seen, the cost per annotation stays fairly stable – per Use Case – for each tracking method. The differences found between different annotation frequencies are caused by the effect of the annotation frequency on the tracking success – the less success, the faster the methods are, since stereo matching is avoided if tracking fails –. With respect to the different trackers, the CSRT, GOTURN and MIL trackers are the most expensive in terms of computation time. However, the reason is not that they need more time but that other trackers lose tracking very often, reducing the computation time. This happens recurrently with the Median Flow and MOSSE trackers with low annotation frequencies. Thus, this efficiency is not particularly useful. In fact, one of the reasons that seems to make MIL and GOTURN the most expensive methods is that these trackers do not acknowledge the loss of tracking. In the case of CSRT, this tracker works well with low and high frequencies and tends to follow the object correctly, with a cost lower than GOTURN and MIL, which in any case is acceptable for this type of application.

The most important conclusion, however, is that in almost all cases, the total computational cost is lower than the length of the video. For instance, all but one method (MIL) can perform the annotation task in less than 72 s for Use Case #0, in less than 30 s for Use Case #1, and in less than 60 s for Use Case #2. Thus, the proposed method can work in real-time. Another important conclusion is that the time needed for tracking roughly equals the time needed for stereo matching in most cases.

4.2.3. Assessment by domain expert

As no absolute ground truth is possible for this particular domain and type of application, we also performed a validation with a domain expert. The domain expert is a gastrointestinal surgeon with RAS certification. He has performed dozens of robotic-assisted surgeries. He has also performed the kind of RAS procedure seen in the videos.

We were interested in the evolution of two quantities – tracking quality and depth perception – over time. Thus, we designed an assessment tool by which the domain expert could watch the annotated videos and at the same time perform a continuous rating of what he perceives. We asked him to watch each of the annotated videos (one video per tracking method, for each of the Use Cases) with VR glasses, while using our continuous rating tool. An Oculus Quest 2 [50] with the application Bigscreen [51] was used to show the stereoscopic annotated videos. Fig. 11 shows a picture of the evaluation process. The videos were shown in randomized order. No head tracking was used.

For the sake of simplicity, the rater was a keyboard-based application by which the domain expert could either press or release the space key. A key pressed means the rating is good. A key released means the rating is bad. This way, we can find out

Fig. 11. Domain expert assessing the stereoscopic RAS videos enhanced with MR annotations, using an Oculus Quest 2 device.

which parts of the video are poorly annotated, both in terms of depth perception and in terms of object tracking. The rater was applied twice: once for rating depth perception and then again for rating tracking perception, since we considered it to be too difficult for the expert to rate both features at the same time. To the best of our knowledge, this evaluation system is a new contribution in this area.

The annotated video included a blue rectangle highlighting the ROI and a text over the blue rectangle. Since the video is stereoscopic, we are assessing if the annotations are properly placed in both the left and right images. If the horizontal or the vertical disparity are wrong, the domain expert will not be able to perceive stereoscopic vision. In addition, if the tracking is wrong, he would complain and rate the target tracking poorly.

Figs. 12 and 13 show the average results provided by the domain expert and the evolution of the ratings over time. Depth ratings were conditionally averaged, i.e., only the values corresponding to a correctly tracked object were included in the average. Green means a positive evaluation, red means negative, and yellow means the evaluation is not applicable (this occurs when depth perception cannot be assessed because no annotation appears as a result of a tracking loss). GOTURN was excluded from the test due to its poor performance. Use Case #0 was also excluded from the subjective test since it does not show real footage.

As can be seen, although the CSRT-based solution seems to be the best of the tested methods in terms of tracking, it is not always the best in terms of depth perception; the domain expert felt that other methods, such as Median Flow, albeit being inferior in terms of tracking record, had a higher level of functionality in terms of depth perception. After some conversations with the expert, we reached the conclusion that the CSRT tracker tends to create size disparity. In other words, the annotated area grows and shrinks constantly. While this could be good for some applications, it seems counterproductive for this application. There are also small fluctuations in the centroid with this method.

Thus, we decided to perform a final test making a change in the CSRT parameters so that the auto-scale is disabled. Table 9 shows the results of assessing the CSRT method with a fixed ROI size. As can be seen, tracking accuracy decreases greatly using a fixed-sized ROI and therefore the small increase in depth perception is not relevant, as the method fails to track the object for most of the video. In this regard, it is important to point out that depth ratings are conditionally averaged, and therefore the average depth ratings of methods like Median Flow or MOSSE only include a small part of the video, whereas the average ratings of the depth perception with CSRT include most of the video because tracking does not fail as often. This partially explains the lower rating of CSRT in depth perception and can be clearly observed in Fig. 13.

5. Discussion

The results presented in the previous section suggest that the proposed method could be used for real-time annotation of different RAS videos and that it is able to work well even in challenging scenarios such as Use Case #2. The supplementary material contains the annotated videos. Nevertheless, 100% success is not guaranteed.

With respect to the objective comparative evaluation, the feasibility of the proposed method is demonstrated given the similarity between the metrics of the PGT and those of some versions of the proposed algorithm. In this regard, the most appropriate tracker seems to be the CSRT method, since it works well for Use Cases #0 and #1. It can also work extremely well for Use Case #2, although in this case, fine-tuning seems necessary. Nevertheless, this tracker presents one small problem: in the default implementation of this method with OpenCV, the size of the tracked ROI is not fixed and changes continuously, albeit in small amounts. When using these changing ROIs for stereoscopic annotation, the domain expert complained about this and verbalized that it was negative for depth perception. The tracker can be forced to work with a fixed-sized ROI. However, making the tracker work with a fixed size increases the likelihood that it would fail to track the object. A trade-off needs to be found in order to be able to use the method reliably with no fine-tuning of its parameters and no human supervision. In any case, a smoothing process, such as a weighted moving average could be applied to the size of the of calculated ROI, significantly reducing the importance of this problem. Another solution could be to calculate a variable size ROI but show a constant size ROI to the user.

In addition to the proposed hybrid tracking–matching solution and the comparison of several tracking algorithms for the RAS domain, this work also proposes a simple quality metric for assessing the solutions to this problem. Our quality metric seems to correlate well with IoU and with the domain expert if the tracker is able to report the events of tracking failure. It also works better than the IoU metric for some particular cases (for instance, CSRT2 vs CSRT10 in Use Case #2). In addition, it is also much easier to calculate than IoU, since IoU needs a PGT, which is very laborious to obtain for this problem. For some cases, such as Use Case #2, the amount of tissue deformation makes the PGT not only laborious but difficult to determine.

However, the main weakness of the proposed quality metric is precisely that it is based on tracking confidence reported by the tracking method. Therefore, we could argue that this proposed metric is useful if the tracker has the ability to be aware of the loss of tracking. In this regard, it is important to note that some trackers are more reliable than others in terms of reporting their tracking confidence. Indeed, some trackers report tracking failures when they cannot detect the object, as occurs with the CSRT tracker, while others do not, such as the MIL tracker. This makes an objective evaluation hard to perform. It is also worth noting that the quality metric is sensitive to frequency, but in a different way than other metrics are. It also uses thresholds for spatial and temporal disparities that could be modified. Further analyses in this regard may be of interest.

With respect to the performance of the proposed solution, the experiments suggest that MIL and GOTURN are the most expensive methods in terms of computation time, and they also do not track the ROI very well. CSRT comes in third place and its cost is higher than other methods, but those other methods (KCF and

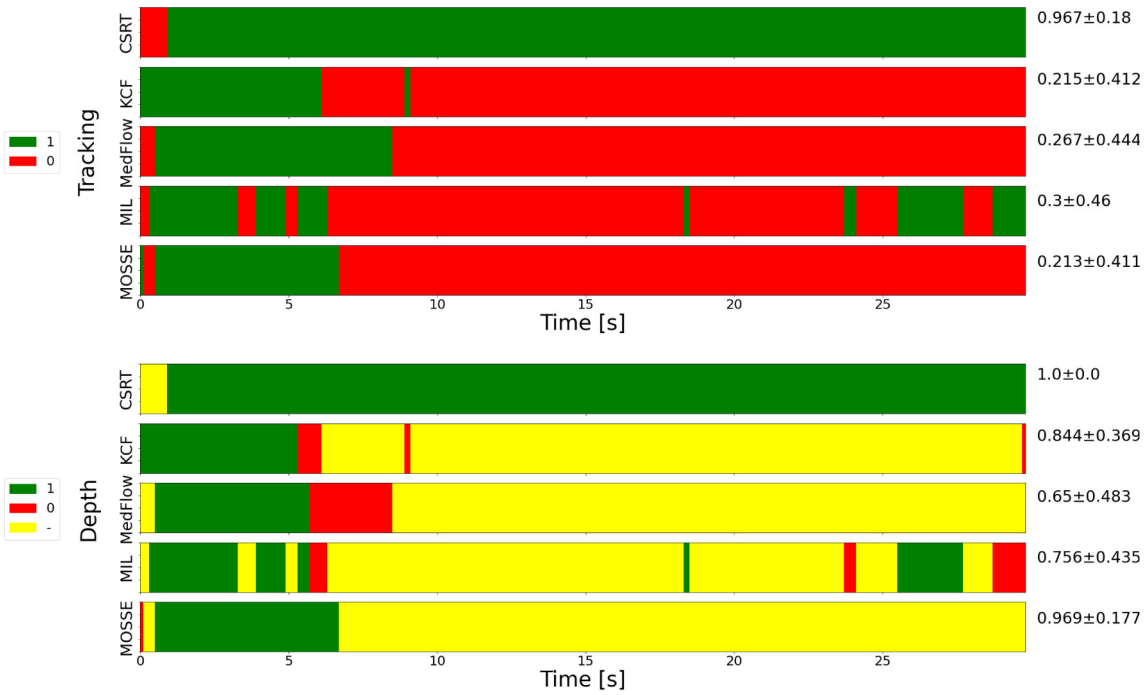

Fig. 12. Use Case #1 – Evaluation by domain expert for each tracking method (average ± standard deviation).

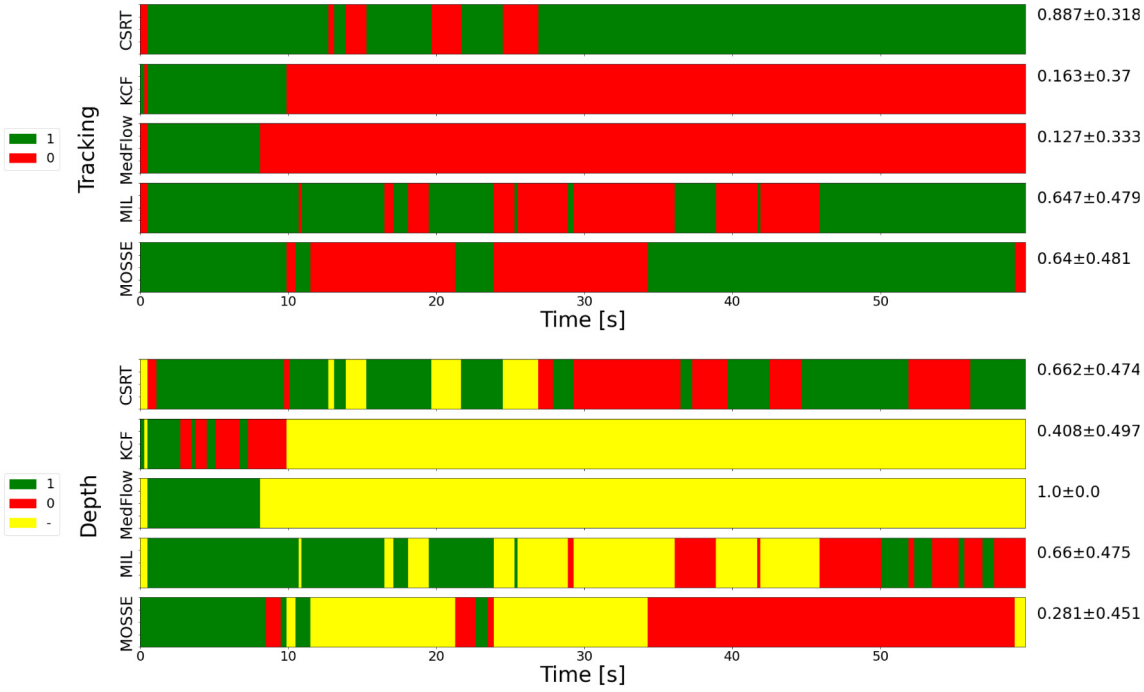

Fig. 13. Use Case #2 – Evaluation by domain expert for each tracking method (average ± standard deviation).

Table 9
Assessment by domain expert after setting CSRT method to avoid auto-scaling. Average results.

Tracking method	Use Case #1		Use Case #2	
	Depth	Target tracking	Depth	Target tracking
CSRT default	1.000	0.967	0.662	0.887
CSRT fixed size	0.938	0.967	0.717	0.279

Median Flow mainly) perform poorly and often lose track of the ROI. This reduces the computation time when tracking failure is acknowledged by the methods, since stereo matching is avoided if tracking fails. On the contrary, CSRT performs much better, but at a higher cost. Nevertheless, in the worst-case scenario, less than a tenth of a second is needed for each annotation with a

10 Hz annotation frequency using CSRT. This will allow real-time usage, since the measured time includes rendering. In addition, if we assume that annotations do not last long and the annotation frequency may be reduced to 5 Hz or even lower values, since objects in RAS do not normally move at extremely high speeds, we could further reduce the computation time.

In this regard, it is worth noting that the stereo matching process takes a lot of time and could be optimized. Currently, the stereo matching algorithm searches through the whole image. For parallel stereo images, like the ones tested in our experiments, only horizontal epipolar matching is necessary. This would allow us to restrict the search of the ROI to an epipolar band, optimizing the process substantially. However, in order to keep the method as general as possible we decided not to restrict the search to horizontal matching. This way, the method would also be potentially suitable for non-parallel stereo cameras in which vertical disparity is not always zero.

It is important to highlight that we intentionally used HD videos and a regular computer for the evaluation, in order to get results that are meaningful for real day-to-day use. We cannot expect that high-end hardware will be needed to apply our annotation method. Surgeons will probably use average computers for their training. Therefore, measuring performance with high-end computers using CUDA or other GPU-based accelerations – as other authors do [33] – is, in our opinion, not useful for this particular application, since we want to maximize the portability and universality of the solution. We are also using Python for similar reasons, as the application is still in a research phase. The use of C++ could further improve performance.

It is also important to highlight that for simplicity we used static texts, but dynamic texts with quantitative data – such as lesion sizes – could be used as well. The proposed method also allows several regions to be marked and tracked simultaneously.

With respect to the domain expert evaluation, the proposed continuous rating method seems to have worked well. It did provide results that match our experience and, most importantly, correlate with most of the objective metrics, with the caveats previously mentioned concerning the reported confidence provided by the tracking methods. Although this assessment is limited to one person and does not constitute a user study, the evaluation is systematic and it was performed by an expert, whose opinion is much more meaningful than that of a random group of people. In addition, by choosing a continuous rating, the final average rating is much more reliable than simply asking the domain expert to rate the whole experience with a single value, which is much harder. To the best of our knowledge, this is the first time this type of evaluation has been applied to this field. It is worth noting that although a $350 VR system was used for the assessment with the domain expert, cheap VR glasses/cardboards – some of them costing just a few US dollars – could also be used to visualize the annotated videos, making our solution easy to deploy in real teaching environments. The choice of a high-quality VR device responds to the need to offer the expert a configuration that is as immersive as possible, so that the strengths and weaknesses of the proposed solution appear as clearly as possible.

Overall, the results can be considered satisfactory. We have to take into account that this study is quite singular, because we are using tracking methods that have been mostly designed and tested to find outdoor objects, such as cars or people. Tissues behave differently. In addition, in RAS, occlusions with surgical tools are common, making the tracking process very challenging for short-term trackers, like the ones that were tested. The CSRT method, however, seems to deal well with short occlusions and is able to remember what it was trying to track even if occlusions occur. Nevertheless, these occlusions need to be short and partial, otherwise the method will fail unless a low annotation update frequency is used. The problem with lower frequencies is that they increase the chance that the tracking method will fail in non-occlusion cases.

However, it seems that real-time tracking is not enough to obtain a bulletproof unsupervised annotation method, since there are situations where the tracker gets lost because of occlusions and quick camera movements. These situations cannot be predicted beforehand, and it is impractical to fine-tune the parameters of the trackers or the annotation update frequency for each particular situation. A possible improvement could be testing in real-time the accuracy of the method to determine a dynamic optimized annotation frequency. However, other solutions, such as trackers based on machine learning may be necessary to offer a more robust solution, even if this is not a real-time one. These trackers should be trained with inputs that are similar to the ones that need to be annotated. Given the variety of surgical procedures, this is not easy. Solutions based on deep learning could be an alternative, although they take a great deal of time to be trained and are better for generic contexts. This explains why the GOTURN tracker fails for this problem, since this deep learning tracker is trained with generic objects and not with internal body tissues.

6. Conclusions and future work

Given the increasing importance of RAS within the surgical field, this paper presents a hybrid tracking–matching method for the creation of MR-based annotations in stereoscopic RAS videos with a teaching perspective. The method is hybrid because it uses a tracking algorithm for the temporal dimension and a matching algorithm for the spatial dimension (left-to-right stereo matching).

As demonstrated in the experiments performed, the method can work well with RAS videos, as it has been successfully used and assessed in two different real Use Cases and in one simulated case. The method is unsupervised and can be run in real-time with a regular computer. Therefore, it could be used by any RAS teacher in order to improve their teaching materials. This is the most important conclusion of this work.

Of all the trackers tested for the presented method, the CSRT tracker seems to be the most reliable and robust, since it is able to work under harsh conditions. It also does not need a high update frequency to work well. Thus, it seems a good tracker for this application, although it usually works better when the tracked ROI can change in size over time, something that may be counterproductive for the correct depth perception of the annotations. Indeed, a virtual annotation that is constantly changing its position or size over time is uncomfortable. In any case, this problem can be minimized by reducing the rate at which the annotations are calculated, or by smoothing out the solution using a weighted moving average. The reduction in the update rate of the annotation could also have a positive effect in avoiding loss of tracking due to occlusions, as one of our experiments has shown. Any combination of calculation frequency and annotation frequency is possible.

Validation by a domain expert also confirms that the proposed method is a suitable solution for this problem. The method works well, although it is true that the virtual annotations created with this algorithm will not be properly occluded if a real object is placed between the camera and the virtual annotation. The most important limitation of this method is that virtual objects are not occlusion-aware since we skip true depth estimation calculations. However, annotations of occluded objects are rare and not very useful, since annotations usually highlight visible objects. In fact, the domain expert did not complain about this at all. In addition, knowing the intrinsic parameters of the camera, the method would also be able to calculate the true depth of the ROI.

It is also true that sustained occlusions can make short term trackers fail, even the CSRT. Thus, for long-term tracking it is important to also explore methods that are either based on the use of AI or combine AI with traditional methods based on convolutional filters. Future work includes improving this method to better deal with occlusions and use depth estimation to make it occlusion-aware. It will also be interesting to try a different tracker, develop a tracker tailored for this problem, perform an ablation study of the parameters involved in the proposed annotation method or even compare tracked annotations to fixed annotations. We will also add the option to perform the stereo matching restricted to an epipolar band in order to allow further optimization at the cost of losing universality. Last but not least, the researchers plan to include this MR-based annotation method in a RAS video teaching application, so that we can test it under real conditions and perform usability studies with surgeons.

Funding

Sergio Casas, Jesús Gimeno and Alfonso García-Fadrique are supported by "Conselleria de Innovación, Universidades, Ciencia y Sociedad Digital", Spain of the government of the Valencian Autonomous Region (Generalitat Valenciana) of Spain, through project GV/2021/037 "Mixed Reality Annotation for Robotic-Assisted Surgery (MiRARAS)". Cristina Portalés is supported by the Spanish government postdoctoral grant Ramón y Cajal, Spain, under grant No. RYC2018-025009-I.

CRediT authorship contribution statement

Cristina Portalés: Writing – original draft, Writing – review & editing, Visualization, Investigation, Software. **Jesús Gimeno:** Conceptualization, Methodology, Resources, Software. **Antonio Salvador:** Data curation, Investigation, Resources. **Alfonso García-Fadrique:** Data curation, Validation, Resources. **Sergio Casas-Yrurzum:** Methodology, Investigation, Writing – original draft, Writing – review & editing, Supervision, Project administration, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Appendix A. Supplementary data: annotated and unannotated videos

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.cag.2022.12.006.

References

[1] Ferro M, Brunori D, Magistri F, Saiella L, Selvaggio M, Fontanelli GA. A portable da vinci simulator in virtual reality. In: 2019 third IEEE international conference on robotic computing. IEEE; 2019, p. 447–8.
[2] Schmidt MW, Köppinger KF, Fan C, Kowalewski K-F, Schmidt LP, Vey J, et al. Virtual reality simulation in robot-assisted surgery: Meta-analysis of skill transfer and predictability of skill. BJS Open 2021;5:zraa066.
[3] Cao D, Kim S. Augmented reality annotation for real-time collaboration system. J Korea Multim Soc 2020;23:483–9.
[4] García-Pereira I, Gimeno J, Portalés C, Vidal-González M, Morillo P. On the design of a mixed-reality annotations tool for the inspection of pre-fab buildings. The Eurographics Association; 2018, http://dx.doi.org/10.2312/ceig.20181157.
[5] García-Pereira I, Gimeno J, Morillo P, Casanova-Salas P. A taxonomy of augmented reality annotations. In: Presented at the 15th international conference on computer graphics theory and applications. 2020, p. 412–9.
[6] Al Hajj H, Lamard M, Conze P-H, Roychowdhury S, Hu X, Maršalkaite G, et al. CATARACTS: Challenge on automatic tool annotation for cataRACT surgery. Med Image Anal 2019;52:24–41.
[7] Andersen D, Popescu V, Cabrera ME, Shanghavi A, Mullis B, Marley S, et al. An augmented reality-based approach for surgical telementoring in austere environments. Mil Med 2017;182:310–5.
[8] Gasques D, Johnson JG, Sharkey T, Feng Y, Wang R, Xu ZR, et al. ARTEMIS: A collaborative mixed-reality system for immersive surgical telementoring. In: Proceedings of the 2021 CHI conference on human factors in computing systems. 2021, p. 1–14.
[9] Lecuyer G, Ragot M, Martin N, Launay L, Jannin P. Assisted phase and step annotation for surgical videos. Int J Comput Assist Radiol Surg 2020;1–8.
[10] Lin C, Andersen D, Popescu V, Rojas-Munoz E, Cabrera ME, Mullis B, et al. A first-person mentee second-person mentor AR interface for surgical telementoring. In: 2018 IEEE international symposium on mixed and augmented reality adjunct. IEEE; 2018, p. 3–8.
[11] Hudelist MA, Husslein H, Münzer B, Kletz S, Schoeffmann K. A tool to support surgical quality assessment. In: 2017 IEEE third international conference on multimedia big data. IEEE; 2017, p. 238–9.
[12] Oropesa I, Escamirosa FP, Sánchez-Margallo JA, Enciso S, Rodríguez-Vila B, Martínez AM, et al. Interpretation of motion analysis of laparoscopic instruments based on principal component analysis in box trainer settings. Surg Endosc 2018;32:3096–107.
[13] Nogueira-Rodríguez A, Domínguez-Carbajales R, Campos-Tato F, Herrero J, Puga M, Remedios D, et al. Real-time polyp detection model using convolutional neural networks. Neural Comput Appl 2021;1–22.
[14] Dardagan N, Brđanin A, Džigal D, Akagic A. Multiple object trackers in OpenCV: A benchmark. In: 2021 IEEE 30th international symposium on industrial electronics. IEEE; 2021, p. 1–6.
[15] Agrawal K, Lal R, Patil H, Kannaiyan S, Gupta D. DeepSCT: Deep learning based self correcting object tracking mechanism. In: 2021 national conference on communications (NCC). presented at the 2021 national conference on communications. Kanpur, India: IEEE; 2021, p. 1–6. http://dx.doi.org/10.1109/NCC52529.2021.9530080.
[16] Bertinetto L, Valmadre J, Henriques JF, Vedaldi A, Torr PH. Fully-convolutional siamese networks for object tracking. In: European conference on computer vision. Springer; 2016, p. 850–65.
[17] Held D, Thrun S, Savarese S. Learning to track at 100 fps with deep regression networks. In: European conference on computer vision. Springer; 2016, p. 749–65.
[18] Valmadre J, Bertinetto L, Henriques J, Vedaldi A, Torr PH. End-to-end representation learning for correlation filter based tracking. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2017, p. 2805–13.
[19] Zhang M, Cheng X, Copeland D, Desai A, Guan MY, Brat GA, et al. Using computer vision to automate hand detection and tracking of surgeon movements in videos of open surgery, vol. 10. 2020.
[20] Bouget D, Allan M, Stoyanov D, Jannin P. Vision-based and marker-less surgical tool detection and tracking: A review of the literature. Med Image Anal 2017;35:633–54.
[21] Qiu L, Li C, Ren H. Real-time surgical instrument tracking in robot-assisted surgery using multi-domain convolutional neural network. Healthcare Technol Lett 2019;6:159–64. http://dx.doi.org/10.1049/htl.2019.0068.
[22] Dakua SP, Abinahed J, Zakaria A, Balakrishnan S, Younes G, Navkar N, et al. Moving object tracking in clinical scenarios: Application to cardiac surgery and cerebral Aneurysm clipping. Int J CARS 2019;14:2165–76. http://dx.doi.org/10.1007/s11548-019-02030-z.
[23] Penza V, Du X, Stoyanov D, Forgione A, Mattos LS, De Momi E. Long Term Safety Area tracking (LT-SAT) with online failure detection and recovery for robotic minimally invasive surgery. Med Image Anal 2018;45:13–23. http://dx.doi.org/10.1016/j.media.2017.12.010.
[24] Ryu J, Moon Y, Choi J, Kim HC. A Kalman-filter-based common algorithm approach for object detection in surgery scene to assist surgeon's situation awareness in robot-assisted Laparoscopic surgery. J Healthcare Eng 2018;2018:1–11. http://dx.doi.org/10.1155/2018/8079713.
[25] Sharan L, Burger L, Kostiuchik G, Wolf I, Karck M, De Simone R, et al. Domain gap in adapting self-supervised depth estimation methods for stereo-endoscopy. Curr Dir Biomed Eng 2020;6.
[26] Liu X, Sinha A, Ishii M, Hager GD, Reiter A, Taylor RH, et al. Dense depth estimation in monocular endoscopy with self-supervised learning methods. IEEE Trans Med Imaging 2019;39:1438–47.
[27] Liu F, Jonmohamadi Y, Maicas G, Pandey AK, Carneiro G. Self-supervised depth estimation to regularise semantic segmentation in knee arthroscopy. In: International conference on medical image computing and computer-assisted intervention. Springer; 2020, p. 594–603.

[28] Huang B, Zheng J-Q, Nguyen A, Tuch D, Vyas K, Giannarou S, et al. Self-supervised generative adversarial network for depth estimation in laparoscopic images. In: International conference on medical image computing and computer-assisted intervention. Springer; 2021, p. 227–37.
[29] Chen L, Tang W, John NW, Wan TR, Zhang JJ. SLAM-based dense surface reconstruction in monocular minimally invasive surgery and its application to augmented reality. Comput Methods Programs Biomed 2018;158:135–46.
[30] Song J, Wang J, Zhao L, Huang S, Dissanayake G. Dynamic reconstruction of deformable soft-tissue with stereo scope in minimal invasive surgery. IEEE Robot Autom Lett 2017;3:155–62.
[31] Luo H, Wang C, Duan X, Liu H, Wang P, Hu Q, et al. Unsupervised learning of depth estimation from imperfect rectified stereo laparoscopic images. Comput Biol Med 2021;105109.
[32] Li Z, Liu X, Drenkow N, Ding A, Creighton FX, Taylor RH, et al. Revisiting stereo depth estimation from a sequence-to-sequence perspective with transformers. In: Proceedings of the IEEE/CVF international conference on computer vision. 2021, p. 6197–206.
[33] Long Y, Li Z, Yee CH, Ng CF, Taylor RH, Unberath M, et al. E-dssr: efficient dynamic surgical scene reconstruction with transformer-based stereoscopic depth perception. In: International conference on medical image computing and computer-assisted intervention. Springer; 2021, p. 415–25.
[34] Allebosch G, Van den Bossche S, Veelaert P, Philips W. Camera-based system for drafting detection while cycling. Sensors 2020;20(1241). http://dx.doi.org/10.3390/s20051241.
[35] Keh J, Cruz M, Rivera M, Jose JA, Sybingco E, Dadios E, et al. AutoTrack: Interactive visual object tracking for efficient object annotations. In: 2020 IEEE 12th international conference on humanoid, nanotechnology, information technology, communication and control, environment, and management (hnicem). presented at the 2020 ieee 12th international conference on humanoid, nanotechnology, information technology, communication and control, environment, and management. Manila, Philippines: IEEE; 2020, p. 1–4. http://dx.doi.org/10.1109/HNICEM51456.2020.9400096.
[36] Lehtola V, Huttunen H, Christophe F, Mikkonen T. Evaluation of visual tracking algorithms for embedded devices. In: Sharma P, Bianchi FM, editors. Image analysis. Lecture notes in computer science, Cham: Springer International Publishing; 2017, p. 88–97. http://dx.doi.org/10.1007/978-3-319-59126-1_8.
[37] Tannus J. Comparison of OpenCV tracking algorithms for a post-stroke rehabilitation exergame. In: 2020 22nd symposium on virtual and augmented reality (svr). presented at the 2020 22nd symposium on virtual and augmented reality. Porto de Galinhas, Brazil: IEEE; 2020, p. 272–6. http://dx.doi.org/10.1109/SVR51698.2020.00049.
[38] Avinash A, Abdelaal AE, Salcudean SE. Evaluation of increasing camera baseline on depth perception in surgical robotics. In: 2020 IEEE international conference on robotics and automation. IEEE; 2020, p. 5509–15.
[39] Doerner R, Steinicke F. Perceptual aspects of VR. In: Virtual and augmented reality (VR/AR). Springer; 2022, p. 39–70.
[40] Open Source Computer Vision. Opencv tracking API [WWW Document]. In: OpenCV tracking API. 2022, https://docs.opencv.org/4.5.4/d9/df8/group__tracking.html [Accessed 10 Oct 22].
[41] Lukežič A, Vojíř T, Čehovin L, Matas J, Kristan M. Discriminative correlation filter with channel and spatial reliability. Int J Comput Vis 2018;126:671–88. http://dx.doi.org/10.1007/s11263-017-1061-3.
[42] Lukezic A, Vojir T, ˇCehovin Zajc L, Matas J, Kristan M. Discriminative correlation filter with channel and spatial reliability. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2017, p. 6309–18.
[43] Henriques JF, Caseiro R, Martins P, Batista J. High-speed tracking with kernelized correlation filters. IEEE Trans Pattern Anal Mach Intell 2014;37:583–96.
[44] Kalal Z, Mikolajczyk K, Matas J. Forward-backward error: Automatic detection of tracking failures. In: 2010 20th international conference on pattern recognition. IEEE; 2010, p. 2756–9.
[45] Babenko B, Yang M-H, Belongie S. Visual tracking with online multiple instance learning. In: 2009 IEEE conference on computer vision and pattern recognition. IEEE; 2009, p. 983–90.
[46] Bolme DS, Beveridge JR, Draper BA, Lui YM. Visual object tracking using adaptive correlation filters. In: 2010 IEEE computer society conference on computer vision and pattern recognition. IEEE; 2010, p. 2544–50.
[47] Kalal Z, Mikolajczyk K, Matas J. Tracking-learning-detection. IEEE Trans Pattern Anal Mach Intell 2011;34:1409–22.
[48] Grabner H, Grabner M, Bischof H. Real-time tracking via on-line boosting. In: Bmvc. Citeseer; 2006, p. 6.
[49] Shvets AA, Rakhlin A, Kalinin AA, Iglovikov VI. Automatic instrument segmentation in robot-assisted surgery using deep learning. In: 2018 17th IEEE international conference on machine learning and applications. IEEE; 2018, p. 624–8.
[50] Meta. Oculus quest 2 [WWW Document]. In: Meta quest, Vol. 2. 2021, https://www.oculus.com/quest-2/ [Accessed 22 Dec 21].
[51] Bigscreen Inc. Bigscreen - your ultimate virtual reality hangout [WWW Document]. 2022, https://www.bigscreenvr.com/ [accessed 10 Oct 2022].