Hybrid Tracking Module for Real-Time Tool Tracking for an Autonomous Exoscope

Elisa Iovene, *Graduate Student Member, IEEE,* Diego Cattaneo, Junling Fu, *Graduate Student Member, IEEE,* Giancarlo Ferrigno, *Senior Member, IEEE,* and Elena De Momi, *Senior Member, IEEE*

***Abstract*—Exoscopes have emerged as a promising visual solution within the field of microneurosurgery. However, manual repositioning poses a challenge causing interruptions that disrupt the surgical flow. Thus, the need for hands-free exoscope control arises. This paper introduces a position-based visual-servoing control approach, comprising a detection module, a hybrid tracking module, and a control module that adjusts a robotic camera holder to follow a surgical tool. The hybrid module was integrated to track and predict the surgical tool's future position to minimize system latency. The proposed system is composed of a 7 Degree-of-Freedom robotic manipulator with an eye-in-hand stereo camera. A comparative analysis with three alternative approaches (Convolutional Neural Network - CNN, Particle Filter - PF, Optical Flow - OF) was assessed using Tracking Error and Center Error metrics. Results showed improved tracking accuracy with an average error of** 9.84 ± 0.08 **mm for slow movements (**2.5 **cm/s) and** 13.11±0.39 **mm for rapid movements (**4 **cm/s). Finally, a User Study was conducted to investigate whether the proposed system effectively reduced the users' workload compared to the manual repositioning of the camera.**

***Index Terms*—Robotic Surgery, Collaborative Robotics, Visual Servoing, Tool Tracking**

I. Introduction

EXOSCOPES have gained widespread acceptance in neurosurgical settings, and their ergonomic benefits have been well-established [1]. The exoscope includes an external scope that streams surgical field images onto an external 2D or 3D monitor, and allows surgeons to maintain a neutral, upright spinal position, leading to better ergonomics [2]. However, their manual repositioning disrupts surgical flow, potentially extending operation durations and increasing cognitive load for the surgeon [3]. Alternatives, including voice, joysticks, and gaze control, have been implemented [4]. In the field of surgical exoscopes, the most prevalent method, currently used in devices like AESCULAP Aeos (Braun, Melsungen, Germany) and ORBEYE (Olympus, Tokyo, Japan), involves a foot-operated joystick controller to control the movement of the exoscope mounted on a robotic arm. While this approach allows surgeons to maintain ambidexterity during procedures, its complexity has emerged as a limiting factor [5]. The inconvenience due to repositioning underscores the importance of minimizing the need for direct intervention in camera control to perform uninterrupted bimanual surgery. Synaptive Medical's 2017 Modus V introduced autonomous control using optical tracking systems with passive markers attached to the suction cannula to enable robotic camera movement [6]. However, obstructions between the instrument and the tracking system, and the need to attach passive markers can limit instrument maneuverability [7]. Instrument localization has been investigated with various techniques, including robot kinematics-based [8] [9] and image-based tracking [10]. A marker-based approach for the automation of the da Vinci endoscope was proposed by [11]. Here, the detection of the tool was achieved via ArUco which may fail in a real scenario when blood or other fluids are present. Markerless approaches provide an accurate reconstruction of the instrument's position in real-time and can be incorporated into robotic controls [12]. [13] presented an autonomous system employing the endoscope to track surgical instruments, utilizing a visual servo approach for endoscope movements. [14] introduced a markerless position-based visual-servoing (VS), utilizing a stereo microscope to track the tip of a manipulator. However, this study used colored markings for tip identification, which may not align with actual surgical scenarios. Additionally, a markerless framework for an autonomous vision-guided camera holder was developed in [15]. This system tracked and followed a selected surgical instrument using a markerless VS technique. However, a significant challenge was the low processing speed of the Convolutional Neural Network (CNN) used for tool detection in the images. This limitation resulted in decreased system responsiveness to tool movements, leading to slow and unstable tracking of the surgical instrument.

In this study, we introduce a novel hybrid tracking module to enable real-time tool tracking in an autonomous exoscope system. The key contributions of this work are:

- A novel hybrid tracking module that combines an optical flow tracking component with a particle filter predictor to forecast the future position of the tracked tool.
- Comprehensive experiments designed to compare the system's performance with traditional approaches.
- Exploration of user experiences associated with manual and autonomous control modalities through a user study.

The remainder of this paper is organized as follows. Section II describes the materials and methods of the system. Section III depicts the experimental setup for system validation and usability. Experimental results are illustrated and discussed in Section IV. Finally, conclusions are reported in Section V.

Manuscript received: December, 4, 2023; Revised March, 15, 2024; Accepted April, 25, 2024. This paper was recommended for publication by Editor Jessica Burgner-Kahrs upon evaluation of the Associate Editor and Reviewers' comments.

The authors are with the Department of Electronics, Information and Bioengineering, Politecnico di Milano, 20133 Milan, Italy. (e-mail: elisa.iovene@polimi.it; junling.fu@polimi.it; giancarlo.ferrigno@polimi.it; elena.demomi@polimi.it) (Corresponding author: Junling Fu)

Digital Object Identifier (DOI): see top of this page.

II. Materials and Methods

The proposed system is divided into three modules: a Tool Detection Module that can recognise a selected surgical instrument, a Hybrid Tracking Module that tracks and predicts the future position of the target tool, and a Visual-Servoing (VS) Control Module responsible for zeroing the error between the desired and actual pose of the robot to ensure that the instrument remains positioned near the center of the camera image. The overall system is illustrated in Fig. 1.

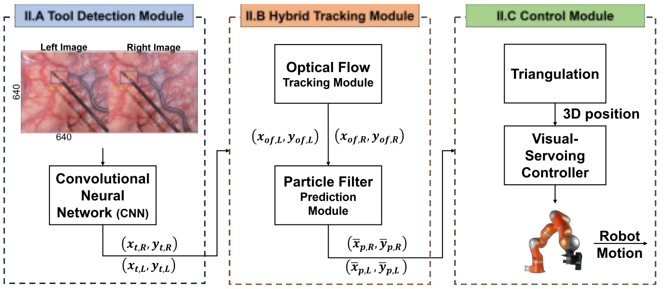
II.A Tool Detection Module
Left Image
Right Image
640
640
Convolutional Neural Network (CNN)
$(x_{t,R}, y_{t,R})$
$(x_{t,L}, y_{t,L})$
II.B Hybrid Tracking Module
Optical Flow Tracking Module
$(x_{of,L}, y_{of,L})$
$(x_{of,R}, y_{of,R})$
Particle Filter Prediction Module
$(\overline{x}_{p,R}, \overline{y}_{p,R})$
$(\overline{x}_{p,L}, \overline{y}_{p,L})$
II.C Control Module
Triangulation
3D position
Visual-Servoing Controller
Robot Motion

Fig. 1. Overall System: images acquired by the stereo camera are sent to a convolutional neural network to detect the surgical tool's position, denoted as $(x_{t,R}, y_{t,R})$ for the right image and $(x_{t,L}, y_{t,L})$ for the left image. These 2D coordinates are then passed to a hybrid tracking module which uses optical flow to track the tool's movement across successive frames, producing tracked positions $(x_{of,R}, y_{of,R})$ and $(x_{of,L}, y_{of,L})$. Subsequently, a particle filter prediction module estimates the tool's future position in image space, indicated by $(\overline{x}_{p,R}, \overline{y}_{p,R})$ and $(\overline{x}_{p,L}, \overline{y}_{p,L})$. Finally, triangulation is used to determine the 3D position of the tool from its predicted 2D positions, which is then fed into a VS controller to guide the robot's motion.

A. Tool Detection Module

A pre-trained Convolutional Neural Network (CNN), specifically YoloV5 [16] was employed to identify the tip of a grasping forceps through a bounding box. Two concatenated RGB frames from a stereo camera, with a total resolution of 1920×1080 pixels, were divided into right and left frames, each sized at 960×1080 pixels. The frames were further downsampled to 640×640 pixels and sent to the CNN. The output of the CNN included coordinates for the upper-right and lower-left corners of the bounding box, the confidence score of the prediction, and the predicted class (i.e., the instrument's tip). A model pre-trained on the object detection NN on the Common Objects in Context (COCO) [17] was fine-tuned on a custom dataset for instrument tool detection. After the identification of the instrument through the bounding box, the tooltip's actual position on the image plane was determined as the center of the bounding box, resulting in coordinates $(x_{t,R}, y_{t,R})$ for the right and $(x_{t,L}, y_{t,L})$ for the left image.

B. Hybrid Tracking Module

The hybrid tracking module consisted of two key components: an optical flow tracking module and a modified particle filter. By leveraging the optical flow, we achieved efficient tool tracking between consecutive frames. Additionally, the particle filter played a crucial role in predicting the tool's future position, effectively reducing system delays.

Optical flow refers to the observed pattern of apparent motion exhibited by objects within a series of consecutive image frames [18]. It operates under the assumption that neighboring pixels in successive video frames move with similar velocities, maintaining their spatial relationships. For this study, the optical flow tracking module offered by OpenCV, which exploits the Lukas-Kanade method with pyramids, was chosen [19]. To calculate the optical flow, the method analyses changes in pixel intensity values between adjacent frames to estimate pixel displacement:

$$\mathbf{I}(x_i, y_i, t_i) = \mathbf{I}(x_i + dx, y_i + dy, t_i + dt) \tag{1}$$

where I denotes the pixel intensity, (x_i, y_i) are the pixel coordinates of frame i at time t, and (dx, dy) is the distance traveled in the next frame after dt time. To determine the displacement of points from the initial frame to the subsequent one over time in the x-axis, $u_i = dx/dt$, and y-axis, $v_i = dy/dt$, the least square fit method is used, and it leads to:

$$\begin{bmatrix} u_i \\ v_i \end{bmatrix} = \begin{bmatrix} \sum_i f_{x_i}^2 & \sum_i f_{x_i} f_{y_i} \\ \sum_i f_{x_i} f_{y_i} & \sum_i f_{y_i}^2 \end{bmatrix}^{-1} \cdot \begin{bmatrix} -\sum_i f_{x_i} f_{t_i} \\ -\sum_i f_{y_i} f_{t_i} \end{bmatrix} \tag{2}$$

where $f_x = \dfrac{\delta I}{\delta x}$ and $f_y = \dfrac{\delta I}{\delta y}$ are the image gradients representing how the intensity of the image changes, and f_t indicates the gradient over time. The Lukas-Kanade method assumes that, given one point, the nine surrounding points have the same motion. In our scenario, we provided the optical flow algorithm with the instrument's position in camera space represented as $(x_{t,R}, y_{t,R})$ and $(x_{t,L}, y_{t,L})$, alongside eight additional neighboring reference points. These points were placed in a cross-like configuration to guarantee comprehensive coverage of the surrounding area, ensuring that the algorithm could effectively capture spatial information beyond the instrument's immediate vicinity. For each of these points, the algorithm considers a 3x3 patch resulting in a total of 81 points. As a result, the optical flow algorithm generated a vector that depicted the displacement of the nine tracked points between successive frames. By calculating the mean values of the x and y coordinates among these points, we determined the tool's coordinates in the image space, denoted as $(x_{of,R}, y_{of,R})$ and $(x_{of,L}, y_{of,L})$.

A modified particle filter was introduced to estimate the future position of the tool in the image space, based on its previous position, speed, and orientation. A particle filter is an algorithm that recursively updates an estimate of the state and finds the innovations driving a stochastic process given a sequence of observations. It uses a set of particles to represent possible states in a dynamic system. These particles evolve according to the system's dynamics and are updated with new observations. Weights are assigned to the particles based on their ability to explain observed data, and resampling is performed to generate a new set of particles that better represent the true state of the system [20]. Our particle filter acted as follows:

1) The weight of every particle is initialized (Fig. 2A):

$$w_{k|i} = \frac{1}{N} \tag{3}$$

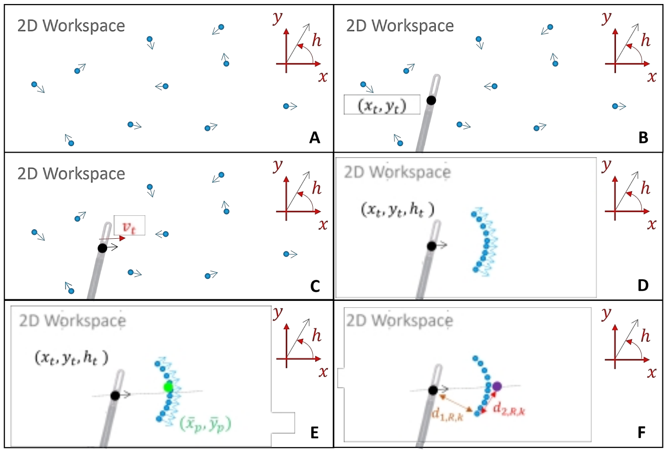

Fig. 2. Particle filter workflow. The blue dots represent the particle pose. Tool pose is identified by the black dot (x_t, y_t). The green dot is the future position $(\overline{x}_p, \overline{y}_p)$. $d_{1,R,k}$ is the distance of the k^{th} particle from actual tool position and $d_{2,R,k}$ is the distance of the k^{th} particle from probable future tool position

where $w_{k|i}$ are the weights of the k^{th} particle at instant i, and N is the total number of particles used.

2) A new estimation of the tool position, $(x_{t|i}, y_{t|i})$ at instant i is acquired (Fig. 2B).
3) The direction and the speed of the movement of the tool are computed (Fig. 2C). To compute the speed, $v_{t|i}$, the discrete derivative between two consecutive positions was used:

$$v_{t|i} = \frac{\sqrt{\Delta^2_{x|i} + \Delta^2_{y|i}}}{\Delta_{t|i}} \tag{4}$$

where $\Delta_{x|i}$, $\Delta_{y|i}$, and $\Delta_{t|i}$ are the variation of the tool's x and y coordinates, and time t, respectively, at instant i and $i-1$. Moreover, the estimated speed was filtered through a low-pass FIR filter of the first order [21].
The direction of motion, $h_{t|i}$, was determined as:

$$h_{t|i} = atan(\Delta_{y|i}, \Delta_{x|i}) \tag{5}$$

4) The future position of the particles is predicted in both images (Fig. 2D) using odometry. In particular, if the variation of the direction of the tool was under a certain threshold, the Runge–Kutta odometry was used:

$$x_{t,k|i+1} = x_{t|i} + \ v_{t|i} \ \cdot \ dt \ \cdot \cos\left(h_{t|i} + \frac{\Delta_{h,t|i} \cdot dt}{2}\right) \tag{6}$$

where $x_{t|i}$ is the x coordinate of the tool at instant i, $v_{t|i}$ is the estimated tool velocity, $h_{t|i}$ is the heading of the tool, $\Delta_{h,t|i}$ is the difference of the tool's heading between two consecutive instants, and dt is the prediction horizon.
When the variation of the direction of the tool was above a certain threshold, the exact odometry was used:

$$x_{t,k|i+1} = x_{t|i} + \frac{v_{t|i}[\sin(h_{t|i} + \Delta_{h,t|i} \cdot dt) - \sin(h_{t|i})]}{\Delta_{h,t|i}} \tag{7}$$

For simplicity, only the calculation of the x coordinate is reported, but the same holds true for the y coordinate.
5) The weighted average of the particles' predicted positions is computed (Fig. 2E):

$$\overline{x}_p|i = \frac{\sum_{k=1}^{N} w_{k|i} \cdot x_{t,k|i+1}}{\sum_{k=1}^{N} w_{k|i}} \tag{8}$$

with N number of particles.
6) The weight of the particles is updated (Fig. 2F):

$$w_{R,k|i} = \frac{\max(dist_k) - \ dist_k}{\sum_{k=1}^{N}(\max(dist_k) - \ dist_k)} \tag{9}$$

where k indicates the k^{th} particle and $dist_k$ takes into account the Euclidean distance between the predicted particle position and both the actual tool position and a potential future position of the tool.

The predicted positions in both images $(\overline{x}_{p,R}, \overline{y}_{p,R})$, $(\overline{x}_{p,L}, \overline{y}_{p,L})$ were used to extract the 3D position of the tool that was then sent to the robot controller.

C. Control Module

Once the 2D predicted coordinates were extracted in the left and right frames, the 3D position of the instrument, $\mathbf{T}^C_{tool} \in \mathbb{R}^{4\times4}$, was computed by triangulation with respect to (w.r.t.) the camera reference frame, $\{\mathbf{C}\}$. The 3D position of the tool was then transformed into the robot's reference frame, $\{\mathbf{B}\}$: $\mathbf{T}^B_{tool} = \mathbf{T}^B_C \times \mathbf{T}^C_{tool}$. Here, $\mathbf{T}^B_C = \mathbf{T}^B_{EE} \times \mathbf{T}^{EE}_C \in \mathbb{R}^{4\times4}$ represents the transformation matrix describing the camera's position in $\{\mathbf{B}\}$; $\mathbf{T}^B_{EE}$ represents the end effector's (EE) position in $\{\mathbf{B}\}$ obtained from the kinematic chain, and $\mathbf{T}^{EE}_C$ is the camera's position w.r.t the EE of the manipulator, calculated through the calibration procedure [15]. Since the goal was to keep the instrument near the center of the camera image at a fixed vertical distance, the desired position of the tool in the camera frame was expressed as $\mathbf{T}^{C_{des}}_{tool} = \left[I \mid 0 \ 0 \ z \ 1\right]$, where I is a 3×3 identity matrix, and z represents the camera's constant position along the z-axis (Fig. 3). Consequently, the desired position of the camera in the robot frame was derived as $\mathbf{T}^B_{C_{des}} = (\mathbf{T}^{C_{des}}_{tool})^{-1} \times \mathbf{T}^B_{tool}$. The transformation among reference frames is shown in Fig. 3. The 3D position of the tool, $\mathbf{P}^B_{C_{des}}$, was then fed into a VS controller that calculated the desired joint velocities to move the robot. To address

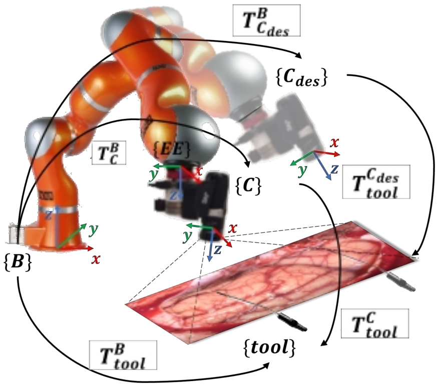

Fig. 3. Coordinate transformation to calculate the desired camera pose in the robot's reference frame

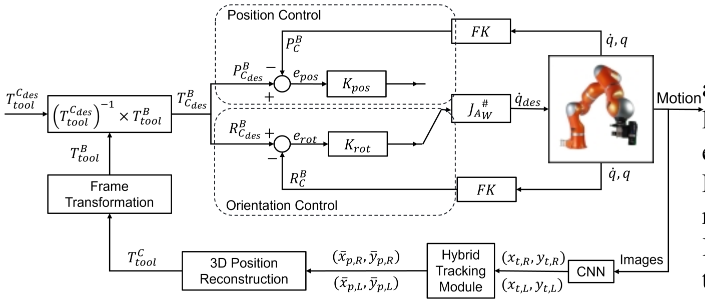

Fig. 4. Control System Overview: the tool position $(x_{t,R}, y_{t,R})$, $(x_{t,L}, y_{t,L})$, extracted from the tool detection module, is sent to the hybrid tracking module which outputs the predicted position, $(\overline{x}_{p,R}, \overline{y}_{p,R})$, $(\overline{x}_{p,L}, \overline{y}_{p,L})$. The 3D position, $\mathbf{T}_{tool}^{C}$, is estimated and transformed in $\{\mathbf{B}\}$, $\mathbf{T}_{tool}^{B}$. This position together with the desired one, $\mathbf{T}_{tool}^{C_{des}}$, gives the desired position of the camera in $\{\mathbf{B}\}$, $\mathbf{T}_{C_{des}}^{B}$, that is sent to the robot controller. The desired position, $\mathbf{P}_{C_{des}}^{B}$, and orientation, $\mathbf{R}_{C_{des}}^{B}$, are compared with the actual position, $\mathbf{P}_{C}^{B}$, and orientation, $\mathbf{R}_{C}^{B}$ extracted from the robot's forward kinematics (FK). The discrepancies are formulated as positional error, $\mathbf{e}_{pos}$, and rotational error, $\mathbf{e}_{rot}$. These errors are fed into a resolved-velocity controller to compute the joint velocities required to correct the robot's motion.

the differing needs of the surgical environment, two control strategies have been defined. The first focuses on camera translation, managing larger-scale movements such as repositioning instruments or transitioning between different areas of the surgical field. In contrast, camera orientation is specifically tailored to address micromovements within confined spaces, which are typical of brain surgery scenarios. Considering that, a position and orientation control were designed:

1) Position Control: The control strategy focused on the translation of EE, maintaining a fixed orientation. A proportional gain matrix, $\mathbf{K}_{pos} \in \mathbb{R}^{3\times3}$, was fine-tuned to optimize the system's behavior to achieve the desired position. The error was calculated as the difference between the desired and actual position of the camera in $\{\mathbf{B}\}$:

$$\mathbf{e}_{pos} = \mathbf{P}_{C_{des}}^{B} - \mathbf{P}_{C}^{B} \tag{10}$$

2) Orientation Control: The control strategy focused on the orientation of the EE, while keeping its position fixed. A proportional gain factor, $\mathbf{K}_{rot} \in \mathbb{R}^{3\times3}$, was fine-tuned to optimize the system's response to orientation errors. The position of the tool, $\mathbf{P}_{tool}^{B}$, was used to compute the desired orientation of the EE, $\mathbf{R}_{C_{des}}^{B}$, which rotated around a fixed point located halfway the length of the tool [22]. The orientation error was computed as:

$$\mathbf{e}_{rot} = \mathbf{q}_{rC} \cdot \mathbf{q}_{rC_{des}}^{-1} \tag{11}$$

where $\mathbf{q}_{rC}$ and $\mathbf{q}_{rC_{des}}$ are the current and desired quaternions.

The feedback error vector $\mathbf{e} = [\mathbf{e}_{pos}; \mathbf{e}_{rot}] \in \mathbb{R}^{6\times1}$ was then fed into a resolved-velocity controller [23] that calculated the vector of joints' velocity profiles:

$$\dot{\boldsymbol{q}}(t) = \mathbf{J}_{A_W}^{\#}(\mathbf{q})\mathbf{K}\mathbf{e} \tag{12}$$

where $\dot{\boldsymbol{q}}(t) \in \mathbb{R}^{7\times1}$ is the vector of desired joints' velocity profiles, $\mathbf{J}_{A}^{\#}(\mathbf{q}) \in \mathbb{R}^{7\times6}$ is the Jacobian matrix pseudo-inverse, and $\mathbf{K} = [\mathbf{K}_{pos}; \mathbf{K}_{or}] \in \mathbb{R}^{6\times6}$ is the positive-definite gains matrix. The overall control scheme is illustrated in Fig. 4.

III. Experimental Setup

To simulate the exoscope system and validate the proposed autonomous framework, a 7-DoFs redundant robotic manipulator (LWR 4+ lightweight robot, KUKA, Germany) with an eye-in-hand stereo camera configuration (JVC GS-TD1 Full HD 3D Camcorder) were used. Moreover, a second 7-DoFs redundant robotic manipulator (LBR IIWA lightweight robot, KUKA, Germany) was considered during the validation of the tracking and control module, to move the surgical tool on a predefined 2D trajectory, as shown in Fig. 5. The choice of a second robot was made to guarantee a high degree of repeatability during the experimental phase, by providing uniform conditions during instrument movement on a well-defined trajectory. The control frequency was set to 200 Hz.

The first step was to validate the performance of the proposed system in terms of instrument detection (Section III.A) and tracking module (Section III.B). Then, a user study was performed to evaluate whether the developed autonomous exoscope was more effective than the traditional control strategy in enhancing ergonomics and reducing workload during task execution (Section III.C).

A. Surgical Instrument Detection

The training dataset for the model comprised a total of 5900 images. Among these, 4100 images were manually recorded and annotated, while the rest were extracted from the 2017 EndoVis challenge [24]. The dataset was split into approximately 80 % for training, 10 % for validation and the remaining 10 % for testing. During training, data augmentation techniques, such as rotations, translations, brightness adjustments, and left-right flips, were applied. The network was trained on an Intel Xeon with a 12Gb Nvidia Titan X GPU for 400 epochs with a learning rate set to 0.001. A mini-batch size of 8 images was used, with a hyper-parameter Intersection over Union (IoU) set to 0.45, indicating that a predicted bounding box overlapping more than 45% with the ground truth was considered a True Positive (TP), otherwise a False Positive (FP). Finally, a confidence threshold of 0.4 was employed, signifying the minimum confidence value for a prediction to be considered valid. The accuracy was evaluated using the average precision (AP) on the testing set. The AP was defined as the area under the precision-recall curve $p(r)$: $AP = \int_0^1 p(r)dr$. Additionally, the detection time, measuring the time taken to detect the position of the target object, was computed as the mean of inference times on the testing set.

B. Surgical Instrument Tracking

The performance of the tracking and the control module was investigated in relation to the target's velocity. To validate the effectiveness of the hybrid strategy (Hybr), a comparative analysis was conducted involving three other approaches:

1) CNN: the instrument position was tracked solely using the CNN, and sent to the robot controller.
2) OF: the instrument was initially tracked by the CNN, and then Optical Flow was used to track the instrument's position between consecutive frames. The tracked position was sent to the robot controller.

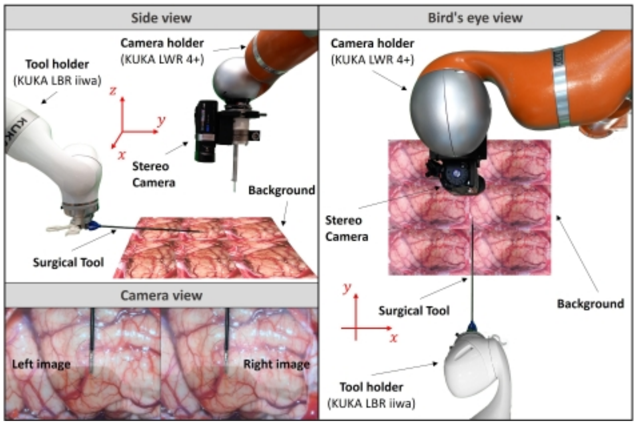

Fig. 5. Experimental setup of validation phase from different points of view. The KUKA LWR 4+ robotic arm moves the stereo camera to track the surgical tool attached to the EE of the KUKA LBR iiwa robot. The camera view shows what the camera sees during the experiments.

3) PF: the instrument was tracked by the CNN, and then its future position was predicted by the Particle Filter. The predicted position was sent to the robot controller.

The strategies were tested within two different velocities: 2.5 cm/s (low speed) and about 4 cm/s (high speed), based on typical brain surgery velocities. Moreover, two different backgrounds were considered: a green background for easy tool detection and a realistic background representing a portion of the brain during a surgical operation. During the experiments, the camera had to follow the tool that was moved in a constant trajectory consisting of the six sequential steps depicted in Fig. 6. All the tests were repeated five times for each strategy and for every scenario. The performance indexes evaluated for these tests were the Tracking Error and the Center Error. The Tracking Error, TE_{xy}, was defined as the xy distance between the position of the tool and the position of the camera:

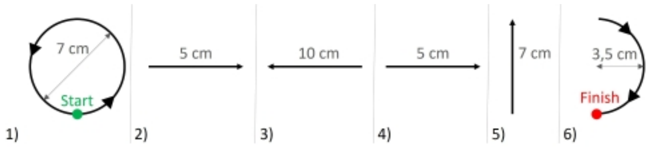

Fig. 6. Six consecutive steps of the trajectory traveled by the tool during the experiments.

$$TE_{xy} = ||\mathbf{P}_C^B - \mathbf{P}_{tool}^B||\ [mm] \tag{13}$$

where $\mathbf{P}_C^B(X_C^B, Y_C^B)$ and $\mathbf{P}_{tool}^B(X_t^B, Y_t^B)$ are the positions of the camera and of the tool respectively expressed in the base frame. The Center Error, CE_{xy}, was defined as the distance between the tool position in the camera frame, $\mathbf{P}_{tool}^C(X_t^C, Y_t^C)$, and the center of the image $\mathbf{P}_{tool}^{C_{des}}(X_t^{C_{des}}, Y_t^{C_{des}})$:

$$CE_{xy} = ||\mathbf{P}_{tool}^C - \mathbf{P}_{tool}^{C_{des}}||\ [mm] \tag{14}$$

In the Orientation mode, the tracking errors for roll, TE_r, and pitch, TE_p, were calculated to assess the accuracy in aligning the camera's angles with the desired orientation. The error in the camera's roll angle, $e_{roll} = roll_{cam} - roll_{des}$, was defined as the difference between the camera's actual roll angle, $roll_{cam}$, and the desired roll angle, $roll_{des}$. The tracking error, TE_r, was then computed as:

$$TE_r\ = \min(|e_{roll}|,\ 2\pi\ - |e_{roll}|)\ [deg]$$

TE_p was computed in the same way, while it was not calculated for the yaw angle since it was kept fixed.

Finally, the following proportional gains were selected:

$$\mathbf{K}_{pos} = 4.0 \cdot \mathbf{I}_3 \quad \mathbf{K}_{rot} = 0.7 \cdot \mathbf{I}_3$$

C. User Study

A User Study was conducted to check if the developed system was effective in reducing the users' workload during task execution when compared to the traditional exoscope control mode.

1) Experimental Protocol: Eighteen non-medical users were asked to perform a bimanual task using two surgical instruments and the camera holder in three different modes:

- **Manual**: users manually moved the camera keeping the tools at the image center. Every time they needed to move the camera, they had to switch between moving the tools and controlling the robot.
- **Auto - Translation**: the autonomous exoscope was controlled with the Position control mode (Section II.C).
- **Auto - Rotation**: the autonomous camera was controlled with the Orientation control mode (Section II.C).

For the autonomous camera modalities, users could decide when to move the camera by pressing a pedal. All users had to perform the tests using all the modalities and they had to repeat the experiments three times for each mode. Participants underwent a training session before the experiments, to familiarize themselves with the different camera control modalities and the task itself. The order in which the modalities were presented to users was randomly selected from a set of permutations. All users provided informed consent before participating. The experimental protocol was approved by the ethics committee from Politecnico di Milano, Italy (No.2023-5069). The experimental setup is shown in Fig. 7.

The workspace consisted of a 35x35 cm wooden board adorned with multiple images of a human brain. Atop these images, four distinct targets were fastened. Each target comprised two pieces of tissue partially affixed to the underlying structure, with some overlap to conceal the object which was represented by a rubber gasket. The workspace also featured designated initial and final positions for surgical instruments and a release zone for the rubber ring.

2) Task: A bimanual pick and place of a hidden object was the task considered in this study to mimic a scenario in which the surgeon employs both hands for the procedure. The tests started with the tools in the initial position (Fig. 7a). Users were tasked with locating a plastic ring hidden beneath one of the four targets (Fig. 7b). After locating the ring, they were instructed to grasp it with a surgical instrument and release it into the designated release zone (Fig. 7c). The task finished when the user returned the tools to its initial position (Fig. 7d). During the tests, users were asked to look only at the external

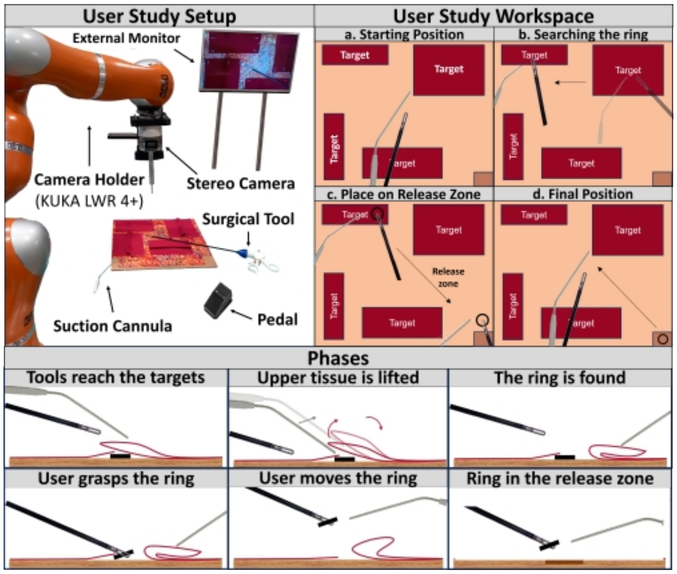

Fig. 7. User Study Setup (left-hand side) including all the elements utilized in the study: the robotic camera controller, the stereo camera, the external monitor for camera image display, the workspace for task execution, two surgical tools, and a pedal, to decide when to move the robot in the automatic control modes. The User Study workspace (right-hand side - top) was divided into 4 different targets, a release zone, and the initial/final position of the surgical instrument. On the bottom, a representation of the phases is shown.

monitor which displayed images in 2D format and ensure both surgical tools remained centered in the image. All the phases of the task are shown in Fig. 7.

3) Performance Metrics: The performance indexes evaluated during the tests were:

- A score assigned based on the distance, $d = ||\mathbf{P}^{C}_{tool} - \mathbf{P}^{C_{des}}_{tool}||$, of the tool from the image center:

$$s = \begin{cases} 0 & \text{if } d < 5\text{cm} \\ -1 & \text{if } d > 5 \text{ cm} \\ -5 & \text{if the tool is outside the FoV} \end{cases} \tag{15}$$

  The total score, S, was normalized by the number of attempts needed to complete the task: $S_r = \sum_i \frac{s_i}{N_{A,r}}$. The number of attempts, $N_{A,r}$, was defined as the number of targets the user explored before successfully locating the ring. Finally, the total score was further normalized to fall within the range [0,100].

- The duration of the movements:

$$t_r = \sum_i \frac{t_r(M^i)}{N_{A,r}} \tag{16}$$

  where t_r represents the duration corresponding to the r^{th} repetition, and $t_r(M^i)$ is the time needed to perform a specific movement, M^i. These movements included motions between the initial and target positions, motions between two different targets, and the movement from the release zone to the final position. Certain movements were excluded from the time duration computation. This included motions on the same target, as they heavily rely on the user's ability to grasp the object, as well as movements from the target to the release zone.

- The path length traveled by the tool:

$$l_r = \sum_i \frac{l_r(M^i)}{N_{A,r}} \tag{17}$$

  where $l_r(M) = \sum_k ||\mathbf{P}^{C}_{tool}(k) - \mathbf{P}^{C}_{tool}(k-1)||$ is the length of a single movement computed as the distance of the tool between consecutive k positions.

- A NASA Task Load Index survey in which the user was asked to rate on a scale from 0 to 100 six different categories (Mental Demand, Physical Demand, Temporal Demand, Performance, Effort, and Frustration) for each control strategy [25].

- A post-experiment questionnaire to investigate the users' preferences in controlling the camera.

IV. Results & Discussion

A. Surgical Instrument Detection Results

The experimental results demonstrate that the instrument detection model achieves an average precision (AP) of 99.3 % for the selected confidence threshold. Furthermore, the average detection time per frame is 0.066 ± 0.01 seconds, resulting in a processing speed of 15 Hz.

B. Surgical Instrument Tracking Results

Table I presents the mean and standard deviation of TE_{xy} and CE_{xy} for all strategies in both slow and fast scenarios with real background. Among the strategies, the Hybrid approach consistently demonstrates the lowest TE_{xy} in both slow and fast scenarios. This finding underscores the effectiveness of the Hybrid strategy in accurately tracking the surgical instrument's movement when compared to the other ones. Furthermore, the Hybrid strategy exhibits low standard deviations in both tracking and center errors, highlighting its consistency and robustness in tracking and centering the surgical instrument across various scenarios. To assess the differences among the strategies, we conducted a Wilcoxon signed-rank test with statistical significance set at $p < 0.05$. The results revealed a statistically significant difference in both TE_{xy} and CE_{xy} among all the strategies, as illustrated in Fig. 8. Furthermore, during the experimental phase, instability issues were identified in the strategies based on CNN (CNN and PF). Specifically, when the surgical tool moved at high speeds, these strategies exhibited detection failures, resulting in divergent position estimates and camera misalignments. In contrast, both the Optical Flow and Hybrid strategies remained stable and did not exhibit such instabilities.

TABLE I
MEAN AND SD OF TRACKING ERROR (TE_{xy}) AND CENTER ERROR (CE_{xy}) FOR CONVOLUTIONAL NEURAL NETWORK (CNN), PARTICLE FILTER (PF), OPTICAL FLOW (OP), AND HYBRID STRATEGY (HYBR)

	TE_{xy} [mm]		CE_{xy} [mm]	
Strat.	Low Speed	High Speed	Low Speed	High Speed
CNN	24.07 ± 0.27	33.09 ± 0.43	36.32 ± 0.18	47.80 ± 0.20
PF	24.14 ± 0.17	32.89 ± 0.65	36.46 ± 0.10	43.97 ± 0.57
OF	23.82 ± 0.35	28.49 ± 0.95	35.24 ± 0.07	41.60 ± 0.48
Hybr	**22.12 ± 0.15**	**27.06 ± 0.55**	**31.01 ± 0.11**	**35.35 ± 0.47**

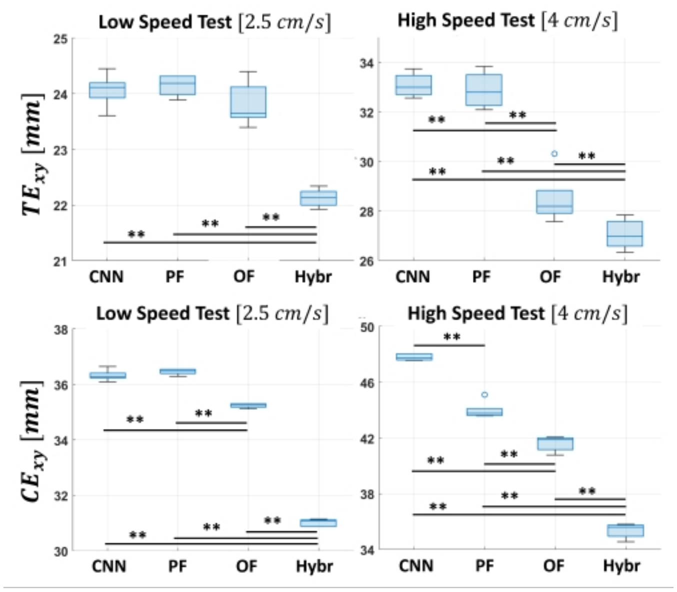

Fig. 8. Tracking Error (above) and Center Error (below). (**, p-value < 0.01)

After the tuning of the controller, both position and orientation control modes, using the Hybrid strategy and the tuned gain (Section III.B), were evaluated. The results of the tracking error and center error for the position and orientation control are reported in Table II.

TABLE II
MEAN AND SD OF THE TRACKING ERROR ($TE_{xy}, TE_{r,p}$) AND CENTER ERROR (CE_{xy}) OF THE HYBRID STRATEGY

	Position Control	
Metrics	Low Speed	High Speed
TE_{xy} [mm]	9.84 ± 0.08	13.11 ± 0.39
CE_{xy} [mm]	16.14 ± 0.14	20.80 ± 0.16
	Orientation Control	
TE_r [deg]	4.29 ± 0.06	5.68 ± 0.08
TE_p [deg]	4.63 ± 0.08	5.65 ± 0.08
CE_{xy} [mm]	22.41 ± 0.39	27.62 ± 0.43

C. User Study Results

The average of the performance metrics from the three repetitions was computed for subsequent statistical analysis since there was not learning curve among the repetitions. Data were compared using the Kruskal-Wallis test, with statistical significance set at $p\text{-}value < 0.05$, followed by a post-hoc Dunn-Sidak test. In all metrics, the automatic control modes showed a significant difference from the manual control mode while no distinction was observed between the two automatic modes. Specifically, the mean durations were 17.51 ± 2.71 s for Manual, 9.72 ± 1.58 s for Automatic - Translation, and 6.87 ± 1.62 s for Automatic - Rotation (Fig. 9 - center). The reduced execution time in autonomous mode was attributed to the user's ability to focus solely on the primary task while the camera's repositioning was handled by the robotic manipulator, consequently lowering mental workload [26]. The Scores (Fig. 9 - left) were significantly higher in the autonomous modalities, with mean values of 77.46 ± 9.69, 98.06 ± 2.61, and 96.76 ± 4.17 for Manual, Automatic - Translation, and Automatic - Rotation, respectively. The higher score indicated that users had the instrument inside the FoV for a longer time compared to the Manual mode. In a real surgical scenario, this poses a risk of complications due to instruments becoming invisible through the exoscope, increasing the potential for inadvertent contact with delicate structures [27]. Regarding the path length (Fig. 9 - right), a significant difference was noted among modalities, with mean values of 0.51 ± 0.25 m, 0.21 ± 0.10 m, and 0.20 ± 0.08 m, respectively. This difference underscores the efficiency of the autonomous modalities in reducing the overall distance traveled by the instrument during task execution. Such optimization is indicative of the effectiveness of autonomous control in reducing the movement of the instrument, potentially contributing to shorter procedural times. The qualitative analysis employed the NASA-TLX survey, where users assessed perceived workload across six subscales (Distraction, Mental Demand, Physical Demand, Situational Stress) for all modalities, as depicted in Fig. 10. The automatic modalities exhibited reduced frustration and demanded less mental and physical effort compared to manually repositioning the camera. Lower scores in the autonomous control modes suggest potential for enhanced performance compared to traditional control. High workload and stress can adversely affect decision-making and motor skills, potentially causing errors in surgery. By alleviating the surgeon's workload and stress, the proposed system may enhance surgical outcomes, enabling more accurate task execution. Moreover, the post-experiment questionnaire indicated that users found the automatic modes to be the easiest to use (42 % for Automatic - Rotation, 58 % for Automatic - Translation), offering a better field of view (8 % for Manual, 25 % for Automatic - Rotation, 67 % for Automatic - Translation), and being more suitable for accomplishing the task (8 % for Manual, 42 % for Automatic - Rotation, 50 % for Automatic - Translation).

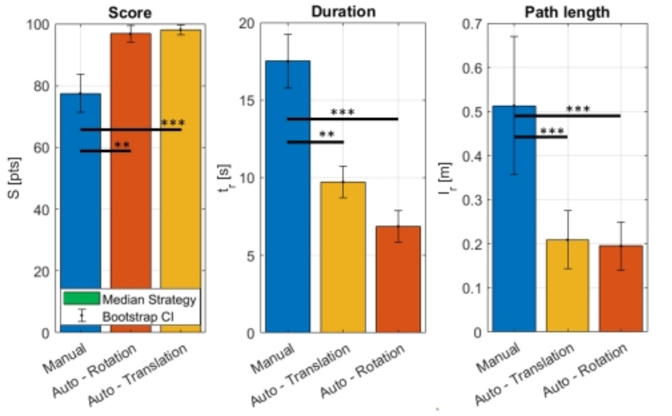

Fig. 9. Score (left), Duration (center), and Path length (right). (**, p-value < 0.01; ***, p-value < 0.001)

V. CONCLUSION

This study introduces a new position-based visual-servoing control method for a robotic camera in brain surgery, aimed

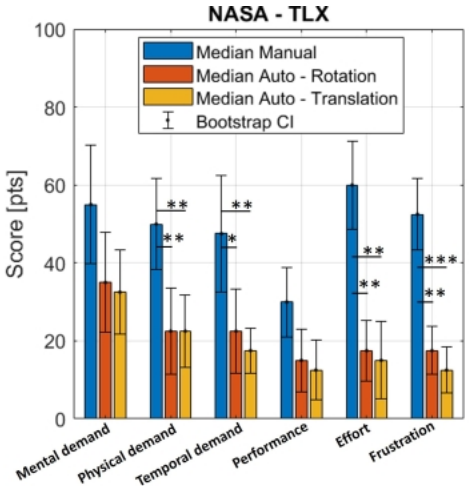

Fig. 10. Results of the NASA - TLX survey across the six subscales for all modalities. (*, p-value < 0.05; **, p-value < 0.01; ***, p-value < 0.001;)

at real-time tool tracking to enhance ergonomics and reduce mental workload. The hybrid module optimized system performance by predicting future tool positions, resulting in lower tracking errors compared to other strategies. A user study was conducted to test whether the proposed system could better support users compared to traditional manual control. From the results emerged that the automatic control modes improve overall performance. Moreover, users found automatic control less physically, temporally demanding, and stressful. Finally, a questionnaire highlighted that automatic modalities facilitated task completion and offered the best field of view during the testing. However, it's important to note that while the task prioritizes assessing performance obtained with the autonomous camera against manual control, it may not fully replicate the complexity of real clinical scenarios. Future work should incorporate tasks that better reflect neurosurgical scenarios with the involvement of medical subjects. In addition, using a dataset consisting of real clinical images of brain surgery would be essential to effectively train the neural network.The observed tracking error rates remain high. Therefore, the optimization of control algorithms and the use of high-performance hardware resources would be necessary in future development.

REFERENCES

[1] A. N. Mamelak, D. Drazin, A. Shirzadi, K. L. Black, and G. Berci, "Infratentorial supracerebellar resection of a pineal tumor using a high definition video exoscope (vitom®)," *Journal of Clinical Neuroscience*, vol. 19, no. 2, pp. 306–309, 2012. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0967586811004024

[2] N. Montemurro, A. Scerrati, L. Ricciardi, and G. Trevisi, "The exoscope in neurosurgery: an overview of the current literature of intraoperative use in brain and spine surgery," *Journal of Clinical Medicine*, vol. 11, no. 1, p. 223, 2021.

[3] R. Berguer, J. Chen, and W. D. Smith, "A Comparison of the Physical Effort Required for Laparoscopic and Open Surgical Techniques," *Archives of Surgery*, vol. 138, no. 9, pp. 967–970, 09 2003.

[4] A. Pandya, L. A. Reisner, B. King, N. Lucas, A. Composto, M. Klein, and R. D. Ellis, "A review of camera viewpoint automation in robotic and laparoscopic surgery," *Robotics*, vol. 3, no. 3, pp. 310–329, 2014.

[5] B. Fiani, R. Jarrah, F. Griepp, and J. Adukuzhiyil, "The role of 3d exoscope systems in neurosurgery: An optical innovation." *Cureus*, vol. 13, no. 6, 06 2021.

[6] D. J. Langer, T. G. White, M. Schulder, J. A. Boockvar, M. Labib, and M. T. Lawton, "Advances in intraoperative optics: a brief review of current exoscope platforms," *Operative Neurosurgery*, vol. 19, no. 1, pp. 84–93, 2020.

[7] C. Gruijthuijsen, L. C. Garcia-Peraza-Herrera, G. Borghesan, D. Reynaerts, J. Deprest, S. Ourselin, T. Vercauteren, and E. Vander Poorten, "Robotic endoscope control via autonomous instrument tracking," *Frontiers in Robotics and AI*, vol. 9, 2022.

[8] T. Da Col, G. Caccianiga, M. Catellani, A. Mariani, M. Ferro, G. Cordima, E. De Momi, G. Ferrigno, and O. De Cobelli, "Automating endoscope motion in robotic surgery: a usability study on da vinci-assisted ex vivo neobladder reconstruction," *Frontiers in Robotics and AI*, vol. 8, p. 707704, 2021.

[9] I. Avellino, G. Bailly, M. Arico, G. Morel, and G. Canlorbe, "Multimodal and mixed control of robotic endoscopes," in *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, 2020, pp. 1–14.

[10] A. Al-Shanoon and H. Lang, "Robotic manipulation based on 3-d visual servoing and deep neural networks," *Robotics and Autonomous Systems*, vol. 152, p. 104041, 2022.

[11] C. Molnár, T. D. Nagy, R. N. Elek, and T. Haidegger, "Visual servoing-based camera control for the da vinci surgical system," in *2020 IEEE 18th International Symposium on Intelligent Systems and Informatics (SISY)*. IEEE, 2020, pp. 107–112.

[12] D. Bouget, M. Allan, D. Stoyanov, and P. Jannin, "Vision-based and marker-less surgical tool detection and tracking: a review of the literature," *Medical image analysis*, vol. 35, pp. 633–654, 2017.

[13] C. Gruijthuijsen, L. C. Garcia-Peraza-Herrera, G. Borghesan, D. Reynaerts, J. Deprest, S. Ourselin, T. Vercauteren, and E. Vander Poorten, "Robotic endoscope control via autonomous instrument tracking," *Frontiers in Robotics and AI*, vol. 9, 2022.

[14] B. C. Becker, V. Sandrine, R. A. MacLachlan, G. D. Hager, and C. N. Riviere, "Active guidance of a handheld micromanipulator using visual servoing," *IEEE International Conference on Robotics and Automation*, pp. 339–344, 2009.

[15] E. Iovene, A. Casella, A. V. Iordache, J. Fu, F. Pessina, M. Riva, G. Ferrigno, and E. De Momi, "Towards exoscope automation in neurosurgery: A markerless visual-servoing approach," *IEEE Transactions on Medical Robotics and Bionics*, vol. 5, 05 2023.

[16] Ultralytics, "Yolov5 by Ultralytics," https://github.com/ultralytics/yolov5.

[17] T.-Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick, and P. Dollár, "Microsoft coco: Common objects in context," 2015.

[18] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.

[19] ——, "The opencv library," *Dr. Dobb's Journal of Software Tools*, 2000.

[20] A. Doucet, S. Godsill, and C. Andrieu, "On sequential monte carlo sampling methods for bayesian filtering," *Statistics and Computing*, vol. 10, 04 2003.

[21] M. B. Trimale and Chilveri, "A review: Fir filter implementation," in *2017 2nd IEEE International Conference on Recent Trends in Electronics, Information Communication Technology (RTEICT)*, 2017, pp. 137–141.

[22] J. Sandoval, H. Su, P. Vieyres, G. Poisson, G. Ferrigno, and E. De Momi, "Collaborative framework for robot-assisted minimally invasive surgery using a 7-dof anthropomorphic robot," *Robotics and Autonomous Systems*, vol. 106, pp. 95–106, 2018.

[23] B. S. et al., *Robotics - Modelling, Planning and Control*. Springer-Verlag London Limited, 2009, pp. 447–448.

[24] M. Allan, A. Shvets, T. Kurmann, Z. Zhang, R. Duggal, Y.-H. Su, N. Rieke, I. Laina, N. Kalavakonda, S. Bodenstedt, L. Herrera, W. Li, V. Iglovikov, H. Luo, J. Yang, D. Stoyanov, L. Maier-Hein, S. Speidel, and M. Azizian, "2017 robotic instrument segmentation challenge," 2019.

[25] S. G. Hart, "Nasa-task load index (nasa-tlx); 20 years later," *Human Factors and Ergonomics Society Annual Meeting*, vol. 50, no. 9, pp. 904–908, 2006.

[26] J. Wang, J. Cabrera, K.-L. Tsui, H. Guo, M. Bakker, and J. B. Kostis, "Clinical and nonclinical effects on operative duration: Evidence from a database on thoracic surgery," *J Healthc Eng*, 02 2020.

[27] A. Krupa, M. De Mathelin, C. Doignon, J. Gangloff, G. Morel, L. Sole, J. Leroy, and J. Marescaux, "Automatic positioning of surgical instruments during laparoscopic surgery with robots using automatic visual feedback," *ESAIM: Proceedings*, vol. 12, pp. 75–83, 01 2002.