# Improving Brain Surgery: A Visual-Servoing Robotic Camera Control System for Enhanced Ergonomics

Elisa Iovene[1], Diego Cattaneo[1,], Junling Fu[1], Federico Pessina[2], Marco Riva[2],
Giancarlo Ferrigno[1], Elena De Momi[1]

***Abstract*— This paper presents a position-based visual-servoing control approach for a robotic camera holder, aimed at enhancing ergonomics and reducing mental stress during brain surgery. The tracking system moves the robotic camera holder by following a selected surgical instrument. Once the instrument is detected, its position is sent to a position control module, which moves the robot accordingly. To enhance system performance, a hybrid module combining optical flow and a particle filter is incorporated to predict the future position of the surgical tool, effectively reducing overall system delays. The proposed system was validated using a 7 Degree-of-Freedoms (DoFs) robotic manipulator with an eye-in-hand stereo camera configuration. Each module of the system was tested, and experimental results demonstrate its capability to detect and track the surgical tool with an average tracking error of $9.84 \pm 0.08$ mm for slow movement and $13.11 \pm 0.39$ mm for fast motion.**

## I. Introduction

Work-related musculoskeletal disorders (WMSDs) pose significant challenges for neurosurgeons, impacting their quality of life and career longevity [1]. These issues are mainly caused by non-neutral positions during surgeries, especially when using microscopes and focusing on oculars. Exoscopes have been introduced as a solution, enabling surgeons to maintain a neutral, upright spinal position for better ergonomics [2]. However, manual repositioning in existing models can interrupt surgery, leading to longer operation times and reduced efficiency [3]. To further reduce the surgeon's workload, minimizing the need for direct intervention in camera control (LoA 2 [4]) is desirable.

Various techniques have been explored for automatizing the camera motion. Among these, the markerless instrument tracking approach stands out as one of the most widely used methods. It enables fast and precise reconstruction of the surgical instrument's 3D position, seamlessly integrating into robotic control frameworks [5]. This approach has been successfully implemented in multiple camera systems, ensuring smooth and controlled movements [6], [7], [8].

This paper presents a robot-assisted autonomous exoscope to reduce the workload of the surgeon during brain surgery. The paper is organized as follows. Section II describes materials and methods of the proposed system. Section III depicts the experimental setup. Experimental results are illustrated and discussed in Section IV. Finally, conclusions are reported in Section V.

## II. Materials And Methods

The proposed system is divided in three modules: a tool detection module (Section II.A.) that can recognise a selected surgical instrument, a hybrid tracking module that tracks and predicts the future position of the target tool (Section II.B.), and a visual-servoing controller responsible for zeroing the error between the desired and actual pose of the robot (Section II.C.). The overall system is illustrated in Fig. 1.

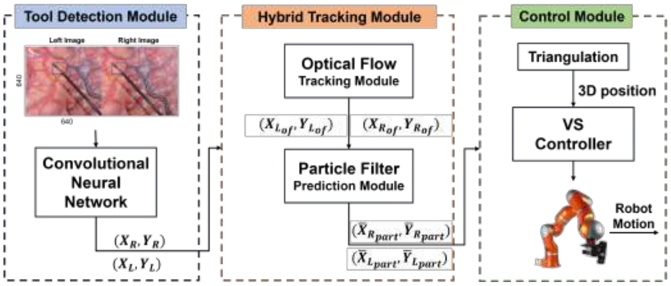

Fig. 1. Overall System: images acquired by the stereo camera are sent to a CNN which detects the surgical tool. The 2D position of the tool is sent to a tracking module, then a prediction module estimates the future position of the tool in the image space. Finally, the 3D position of the tool is extracted from the predicted position and is fed to a visual-servoing controller.

### A. *Tool Detection Module*

A pre-trained CNN YoloV5 [9] was fine-tuned for target detection. The CNN received downsampled images of size 640x640 from a stereo camera and provided the position of the instrument's tip through a bounding box. The center of the bounding box in the right and left images, $(X_R, Y_R)$, $(X_L, Y_L)$, denoted the position of the instrument in the camera space.

### B. *Hybrid Tracking Module*

To address the CNN's low-speed performance in tool detection, a hybrid tracking module was introduced. The module consisted of two key components: the optical flow tracking and a modified particle filter. By leveraging the optical flow, we achieved efficient tool tracking between consecutive frames. Additionally, the particle filter played a

[1]E. Iovene, D. Cattaneo, J. Fu*, G. Ferrigno, and E. De Momi are with the Department of Electronics, Information and Bioengineering, Politecnico di Milano, 20133 Milan, Italy. *Corresponding Author. elisa.iovene@polimi.it, diego2.cattaneo@mail.polimi.it, junling.fu@polimi.it, giancarlo.ferrigno@polimi.it, elena.demomi@polimi.it

[2]F. Pessina and M. Riva are with the Department of Biomedical Sciences, Humanitas University, Pieve Emanuele, Milan, Italy and also with the IRCCS Humanitas Research Hospital, Rozzano, Milan, Italy federico.pessina@hunimed.eu, marco.riva@hunimed.eu

crucial role in predicting the tool's future position, effectively reducing system delays.

The optical flow tracking module offered by OpenCV [10], which exploits the Lukas-Kanade method with pyramids, was chosen for this study. Optical flow refers to the pattern of apparent motion of image objects between two consecutive frames caused by the movement of object or camera. It represents a 2D vector field where each vector denotes the displacement of points from the first frame to the second. Optical flow works on two assumptions:

1) The pixel intensities of an object do not change between consecutive frames.
2) Neighbouring pixels have similar motion.

In our case, the position of the instrument in the camera space, $(X_R, Y_R)$, $(X_L, Y_L)$, was sent to the optical flow together with eight surrounding points to lower the risk of losing the tool position because of partial occlusions or small changes of the pixel intensities. Moreover, pyramids were employed to accommodate not only small movements of the tracked points but also larger ones. Pyramids allow for image scaling, enabling the system to perceive significant movements as if they were smaller, contributing to improved tracking performance. The tool position was thus detected using the CNN on the first frame and then the optical flow was used to track the tool's position, $(X_{R_{of}}, Y_{R_{of}})$, $(X_{L_{of}}, Y_{L_{of}})$, in the following, with exceptions made when the optical flow lost the points to follow or appeared to be tracking the wrong point in the image. The CNN was used also every 15 seconds to confirm precise tracking of the intended target.

A modified particle filter was introduced to get an estimation of the future tool position in the image space, on the basis of the previous position and of the speed and orientation of motion. The 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 does so by a sequential Monte Carlo method. Sequential Monte Carlo methods perform a similar role to the Kalman filter in non-linear and/or non-Gaussian environments. However, unlike the Kalman filter, the particle filter employs simulation methods to generate state and innovation estimates. This allows the particle filter to effectively handle complex and non-linear systems such as the motion of a surgical instrument. Every time the tool position, $(X_{R_{of}}, Y_{R_{of}})$, $(X_{L_{of}}, Y_{L_{of}})$, was computed, our particle filter acted as follows:

1) The direction and the speed of the movement of the tool were computed.
2) The heading of each particle was distributed normally around the direction of the movement of the tool.
3) The future position of the particles was predicted in both images. In particular, if the variation of the direction of the tool was under a certain threshold, the Runge–Kutta odometry was used:

$$X_{R,i} = x_t + \ v_t \ \cdot \ dt \ \cdot \cos\left(h_i + \frac{\Delta_h \cdot dt}{2}\right) \quad (1)$$

$$Y_{R,i} = y_t + \ v_t \ \cdot \ dt \ \cdot \sin\left(h_i + \frac{\Delta_h \cdot dt}{2}\right) \quad (2)$$

where $h_i$ is the heading of the $i-th$ particle, $\Delta_h$ is the difference of the heading of the tool at two consecutive instants, $dt$ is the prediction horizon, $x_t$ and $y_t$ are the x and y coordinate of the tool respectively, coming from the tracking module, $v_t$ is the estimated velocity of the tool, and $i$ is the number of the particle. When the variation of the direction of the tool was above a certain threshold the exact odometry was used:

$$X_{R,i} = x_t + \frac{v_t[\sin(h_i + \Delta_h \cdot dt) - \sin(h_i)]}{\Delta_h} \quad (3)$$

$$Y_{R,i} = y_t - \frac{v_t[\cos(h_i + \Delta_h \cdot dt) - \cos(h_i)]}{\Delta_h} \quad (4)$$

For simplicity, we provided the equation for the right frame; however, the same holds true for the left frame as well.

4) The weighted average of the particles' positions was computed to determine the future position of the tool:

$$\overline{X}_{R_{part}} = \frac{\sum_{i=1}^{N} weight_i \cdot X_{R,i}}{\sum_{i=1}^{N} weight_i} \quad (5)$$

$$\overline{Y}_{R_{part}} = \frac{\sum_{i=1}^{N} weight_i \cdot Y_{R,i}}{\sum_{i=1}^{N} weight_i} \quad (6)$$

with $N$ number of particles.

5) The weight of the particles was updated as follows:

$$weight_i = \frac{\max(dist_i) - \ dist_i}{\sum_{i=1}^{N} (\max(dist_i) - \ dist_i)} \quad (7)$$

where $i$ indicates the *i-th* particle and $dist_i$ 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. This additional term was intended to assign more weight to the direction of the motion being tracked.

The predicted positions in both images $(\overline{X}_{R_{part}}, \overline{Y}_{R_{part}})$, $(\overline{X}_{L_{part}}, \overline{Y}_{L_{part}})$ were used to extract the 3D position of the tool by triangulation that was then sent to the robot controller.

### C. Robot Control Module

The 3D predicted position of the tool is sent to a visual servoing controller. In this work, the motion taken into account was the translation of the robot while the orientation was kept fixed. The error was calculated from the desired position of the camera in $\{\boldsymbol{B}\}$, $\boldsymbol{P}^B_{C_{des}}$, and its actual position, $\boldsymbol{P}^B_C$, where $\boldsymbol{C}$ indicates the camera reference frame and $\boldsymbol{B}$ indicates the base reference frame as illustrated in Fig. 2. The desired position of the camera could be obtained as:

$$\boldsymbol{P}^B_{C_{des}} = \boldsymbol{P}^B_{tool} * (\boldsymbol{P}^{C_{des}}_{tool})^{-1} \quad (8)$$

where $\boldsymbol{P}^{C_{des}}_{tool} = \begin{bmatrix} 0 & 0 & d & 1 \end{bmatrix}$ as the goal was to keep the instrument near the center of the camera image with a distance $d$. This error was then fed into a resolved-velocity controller which assumes that the manipulator acts as an ideal positioning device and that calculates the desired joint velocities to move the robot.

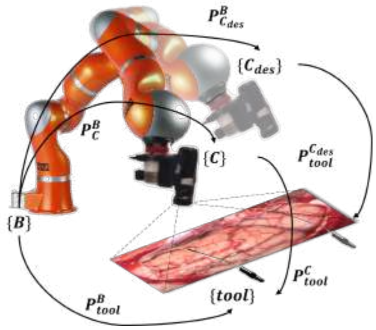

Fig. 2. Transformation. $\boldsymbol{P}^{C}_{tool}$ and $\boldsymbol{P}^{C_{des}}_{tool}$ are the actual and desired position of the tracked object in the camera space, respectively. $\boldsymbol{P}^{B}_{tool}$ is the position of the tool in the robot's base reference frame.

## 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 to move the surgical tool on a predefined 2D trajectory, as shown in Fig. 3. The choice of a second robot was made to guarantee a high degree of repeatability across the experiments.

### 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 [11]. The dataset was split into approximately 90 % for training and 10 % for validation testing. The detection accuracy was evaluated using an Intersection over Union (IoU) threshold of $\geq 45$ %, classifying predicted bounding boxes as true positives (TP) if they overlapped the ground truth by at least 45 %, otherwise as false positives (FP). The average precision (AP) was calculated 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 and estimate the position of the target object, was considered.

### B. Surgical Instrument Tracking

The performance of the tracking and the control module was investigated in relation to the target's velocity, representing the surgical instrument's movement. To validate the effectiveness of the hybrid strategy (Hybr), a comparative analysis was conducted involving three other approaches: utilizing only the CNN (CNN), applying the particle filter predictor to CNN inferences (PF), using optical flow for tool tracking between consecutive frames (OF). The performance indexes analyzed are the tracking error, defined as the distance between the camera and the tool:

$$TE = ||P^{B}_{C} - P^{B}_{tool}||\ [mm] \tag{9}$$

and the center error defined as the distance between the tool position in the image space and the center of the image $(C_x, C_y)$:

$$CE = ||X_t - C_x, Y_t - C_y||\ [mm] \tag{10}$$

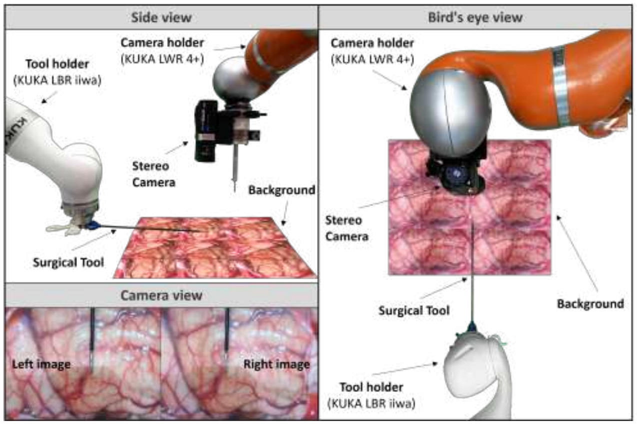

Fig. 3. Experimental setup from different points of view

The strategies were tested within two different velocity scenarios to study the robustness of the system against different conditions. The two velocities were chosen on the basis of a study carried out with neurosurgeons about the typical speeds reached during brain surgeries: the system was thus been tested with the tool moving at about 2.5 cm/s (low speed) and about 4 cm/s (high speed). The camera had to follow the tool that moved in the constant trajectory described in Fig. 4.

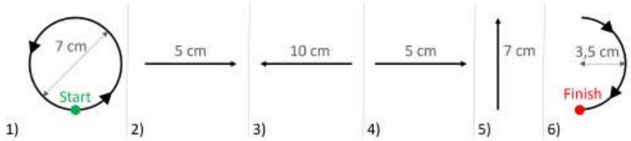

Fig. 4. Trajectory travelled by the tool during the experiments

All the tests were repeated 5 times for each strategy and for every scenario. The strategies that involved the particle filter use 300 particles for each image (left and right) to predict the future tool position.

During these tests, the robot controller was based on a proportional gain equal to 1. After the evaluation of the optimal strategy, the robot controller was fine-tuned, resulting in the selection of a proportional gain of 4. Following that, the system was further tested using the updated parameter in the same experimental setups.

## 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 $0.066 \pm 0.01$ seconds, resulting in a processing speed of 15 Hz.

### B. Surgical Instrument Tracking Results

The mean and the standard deviation of the tracking error and center error for all the strategies in the two different scenario can be appreciated in Table I. Among the strategies, the Hybrid approach consistently demonstrates the lowest tracking error in both the slow and fast scenarios. This indicates that the Hybrid strategy is effective in accurately tracking the surgical instrument's movement compared to the other strategies. Additionally, the Hybrid strategy displays low standard deviations in both tracking and center errors, showing its consistency and robustness in tracking and centering of the surgical instrument across varying scenarios.

TABLE I

MEAN AND STANDARD DEVIATION OF THE PERFORMANCE INDEXES OF THE DIFFERENT STRATEGIES

| | Tracking Error [mm] | | Center Error [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 |

The evaluation of different performance metrics was examined through the Wilcox signed-rank test, with statistical significance established at a threshold of $p < 0.05$. A statistical difference was found in both tracking error and central error among all strategies as shown in Fig 5. This reinforces the observation that the hybrid strategy consistently delivers superior performance, outperforming the alternatives.

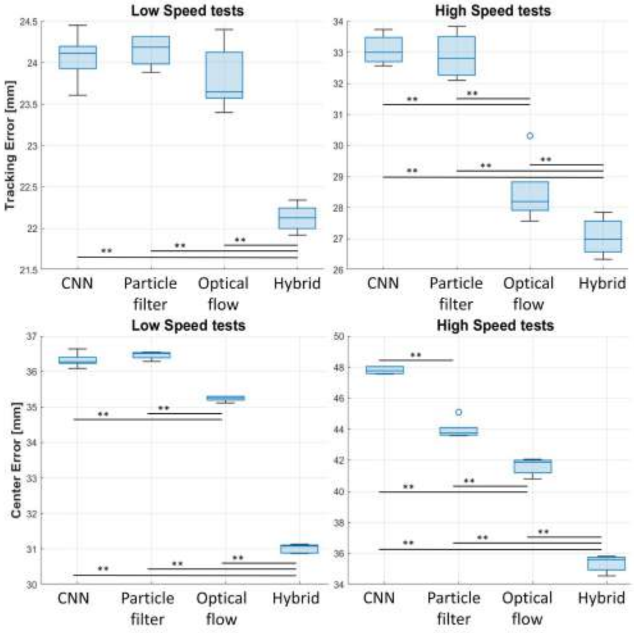

Fig. 5. Tracking Error (above) and Center Error (below). (**, $p$-value $< 0.01$)

Moreover, during the experimental phase an instability of the strategies based on CNN (CNN and PF) was discovered: when the tool moves at high speed, the detection fails, the position estimated diverges and so does the camera. While the Optical flow and the Hybrid strategies do not present instabilities. After the tuning of the controller, the Hybrid strategy presents the performance indexes reported in Table II.

TABLE II

MEAN AND STANDARD DEVIATION OF THE PERFORMANCE INDEXES OF THE HYBRID STRATEGY WITH DIFFERENT PROPORTIONAL GAINS

| | Tracking Error [mm] | | Center Error [mm] | |
|---|---|---|---|---|
| **Gain** | **Low speed** | **High speed** | **Low speed** | **High speed** |
| 1 | 22.12 ± 0.15 | 27.06 ± 0.55 | 31.01 ± 0.11 | 35.35 ± 0.47 |
| 4 | 9.84 ± 0.08 | 13.11 ± 0.39 | 16.14 ± 0.14 | 20.80 ± 0.16 |

## V. CONCLUSION

This study introduces a novel position-based visual-servoing control approach for a robotic camera holder, with the primary objective of enhancing ergonomics and alleviating mental demand during brain surgery. The integration of a hybrid module, combining optical flow and a particle filter, was introduced to further optimize system performance by predicting future tool positions. Overall the tracking system follows the motion of the surgical tool with a relatively low tracking error. In future work, the system should be evaluated in scenarios that better mimic the real world.

## REFERENCES

[1] A. Lavé, R. Gondar, A.K. Demetriades, T. R. Meling, Ergonomics and musculoskeletal disorders in neurosurgery: a systematic review, Acta Neurochir, vol. 162, pp. 2213–2220, 2020.

[2] N. Montemurro, A. Scerrati, L- Ricciardi, G.Trevisi, Gianluca, 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, pp. 223, December 2021

[3] 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, 2021.

[4] T. Haidegger, Autonomy for Surgical Robots: Concepts and Paradigms, IEEE Transactions on Medical Robotics and Bionics, vol. 1, no. 2, pp. 65-76, 2019.

[5] D. Bouget, M. Allan, D. Stoyanov, P. Jannin, Vision-based and markerless surgical tool detection and tracking: a review of the literature, Med Image Anal., vol. 35, pp.633-654, 2017

[6] 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

[7] E. Iovene et al., Towards Exoscope Automation in Neurosurgery: A Markerless Visual-Servoing Approach, IEEE Transactions on Medical Robotics and Bionics, vol. 5, no. 2, pp. 411-420, May 2023, doi: 10.1109/TMRB.2023.3258524.

[8] P.J.M. Wijsman, I.A.M.J Broeders, H.J. Brenkman, et al., First experience with THE AUTOLAP™ SYSTEM: an image-based robotic camera steering device, Surg Endosc, vol. 32, pp. 2560–2566, 2018

[9] Ultralytics-Yolov5, Available online: https://github.com/ultralytics/yolov5 (accessed on 1 Januray 2021)

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

[11] 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