Proceedings of the 2025 SICE Festival with Annual Conference
September 9-12, 2025, Chiang Mai, Thailand

Hand Gesture-Guided Robotic Manipulator for Handling Laparoscope during Minimally Invasive Surgery

Yada Suksawasdi[1], Tanyaton Oranrigsupak[1], Norapath Arjanurak[1], Ronnapee Chaichaowarat[1] , Voranaddha Vacharathit[2], and Sopark Manasnayakorn[2]

[1]International School of Engineering, Faculty of Engineering, Chulalongkorn University, Bangkok, 10330 Thailand;
Emails: ysuksawasdi@gmail.com; tanyatonoran@gmail.com; norapath.arj@gmail.com; ronnapee.c@chula.ac.th
[2] Department of Surgery, Faculty of Medicine, Chulalongkorn University, Bangkok 10330 Thailand;
Emails: voranaddha.v@chula.ac.th; sopark.m@chula.ac.th

Abstract: This paper presents a hand gesture-controlled robotic system for laparoscope manipulation during minimally invasive surgery (MIS). Using MediaPipe-based gesture recognition, neural network gesture classification model, and a UR3e robotic arm, surgeons can adjust laparoscope orientation intuitively—tilting and rotating without physical contact. The system allows users to adjust robot's speed with hand gesture inferface, and utilize Tool Center Point (TCP) function for UR3e arm to prevent collisions at the entry point. The system also leveraging YOLOv11 and Endo-Depth model for real-time depth estimation to prevent tissue collisions. Performance was evaluated through task completion accuracy, system delay, and gesture efficiency.

Keywords: Minimally Invasive Surgery, Hand Gesture Classification, Robotic Laparoscope Control, UR3e, MediaPipe, Depth Estimation, YOLO, Human-Robot Interaction

1. INTRODUCTION

Minimally invasive surgery (MIS) has revolutionized modern medicine by enabling surgical procedures through small incisions, resulting in faster recovery times, reduced postoperative pain, and minimal scarring [1] [2]. During MIS procedures, a laparoscopic cameras and laparoscopic surgical tools are inserted through small incisions where the former is used to visualize the surgical field through these small incisions [3]. The management of the laparoscope can be challenging and requires a camera assistants in addition to the operating surgeon. By enabling hands-free control via gestures, we built a system that could ultimately lead to more intuitive laparoscope manipulation by the surgeon independent of an assistant cameraman.

In laparoscopic surgery, surgeons either rely on assistants, risking communication delays and higher costs [4], or control the camera themselves, dividing focus and increasing fatigue [5]. Recent advances in surgical automation offer promising solutions such as a robotic arm controlled by the surgeon's head movements using gyroscopes [6]. Other approaches include using surface electromyography signals for Surgical Instrument Signaling (SIS), which classifies 14 hand gestures via machine learning for pick-up instruments [7], and GestoNurse, which enables robotic assistance through finger-counting gestures [8].

Hand gesture recognition [9] provides intuitive and contactless control in medical applications, reducing the need for physical interaction with instruments. For instance, the MediaPipe library has been used for hand-gesture control in ultrasound preparation, offering a markerless tracking method that minimizes interference with natural motion and tactile feedback [10].

Universal Robots' collaborative arms, such as the UR3e, are well-suited for real-time human interaction, operating safely alongside humans while maintaining precision and efficiency [11]. Prior implementations have demonstrated their capabilities in precision tasks, including a UR3 performing a ball-on-plate balancing task with real-time visual feedback [12], and a UR5 integrating hand gesture recognition (HGR) with Myo Armband sensors for robotic control [13].

This work integrates MediaPipe vision technology with a UR3e robotic arm for real-time detection and classification of surgeon hand gestures. These gestures control laparoscope positioning—tilting left-right, up-down, and rotating—through contactless interaction, enhancing human-robot collaboration.

The system aims to improve surgical workflow while maintaining precision. It addresses challenges such as reliable gesture recognition under varied lighting, real-time responsiveness, and accurate classification. Experimental validation confirms that using the hand gesture manipulating system can obtain high positional accuracy, consistent task completion time and low system delay. However, there are some variation in number of hand-gesture command used.

Beyond surgery, the system benefits other domains needing precise human-robot interaction, contributing to motion-tracking and robotic control technologies for high-stakes environments.

The paper is organized as follows: Section II describes materials and methods, Section III presents results, Section IV offers discussion, and Section V concludes with key contributions and future directions.

*The first three authors contributed equally to this work.

2. MATERIALS AND METHODS

The development of the hand-gesture-guided robot consists of three main stages: Clinical Needs Assessment, Software Implementation, and Hardware Implementation. The software component includes hand-gesture commands, a hand-tracking system, and a classification model for gesture recognition. The hardware component focuses on UR arm control, Tool Center Point (TCP) for safety factors, and speed modulation based on gesture magnitude. Finally, the system's performance is evaluated in a testing environment using four different metrics.

According to Fig. 1, the overall workflow of the project begins with the activation of the hand-gesture-camera interface. Once activated, the system detects and tracks the surgeon's hand position and gesture within the controlled environment. When a trained gesture is recognized by a classification model, the corresponding command is interpreted and transmitted to the Universal Robot (UR) arm. The UR arm then moves according to the designated hand position and executes the required action based on the identified gesture input. Additionally, the depth estimation system is integrated to ensure that the robot avoids any surface collision or damage.

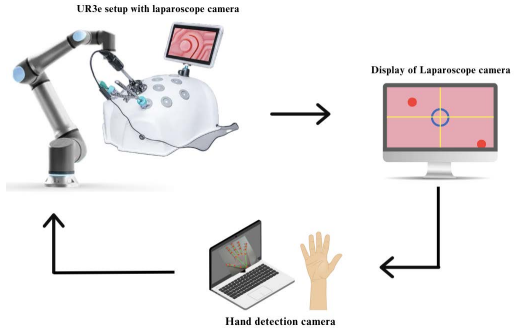

Fig. 1 The overall system workflow.

2.1. Clinical Needs Assessment

A qualitative assessment was conducted through structured interviews with surgeons from the Department of Surgery, Faculty of Medicine, Chulalongkorn University. The primary objective was to identify clinical challenges in laparoscopic procedures and evaluate opportunities for technological intervention. The interviews revealed a significant need for alternative laparoscope camera control methods, which subsequently informed the development objectives of this work. Feedback from surgeons emphasized that camera control, often delegated to an assistant, could benefit from a more responsive and intuitive interface. Therefore, this study focuses on replacing the assistant's manual camera manipulation with a robotic system guided by hand gestures.

2.2. Software: Hand Gestures recognition

This process primarily utilizes OpenCV for managing incoming video frames, Mediapipe for hand tracking, and TensorFlow for the classification model. The system is adapted from the original repository courtesy of [14].

2.2.1. Defining commands

Three specific control commands were defined for the system: 'Tilt left-right', 'Tilt up-down' and 'Rotate'. The commands were chosen base on the essential movement of the laparoscope camera during surgery. According to [15] the tool motion is restricted to four degrees-of-freedom (DOF) as observed in Fig. 2. The first three DOF includes α (Command 1: tilting left-right), β (Command 3: tilting up-down), γ (Command 2: rotating), which are rotations around the x-axis, y-axis and z-axis, respectively. z is the translation along the z-axis, since was excluded to prevent tissue damage from friction at the entry point [16], and laparoscope is already equipped with a built-in zoom function [17].

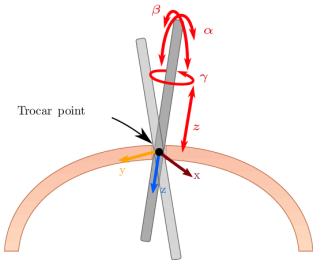

Fig. 2 Laparoscopy motion around the trocar (entry) point (Original image courtesy of Garcia et al. University of Manchester).

2.2.2. Assigning hand Gestures to commands

Three distinct hand gestures are selected to issue three chosen commands as shown in Fig. 3.:

- **Command 1: "Tilt left-right"** Hold out the thumb and index finger, moving the thumb to adjust the magnitude
- **Command 2: "Rotate"** Close palm and hold out the thumb, rotate the wrist to adjust the magnitude
- **Command 3: "Tilt up-down"** Open the palm and all fingers, moving the thumb to adjust the magnitude

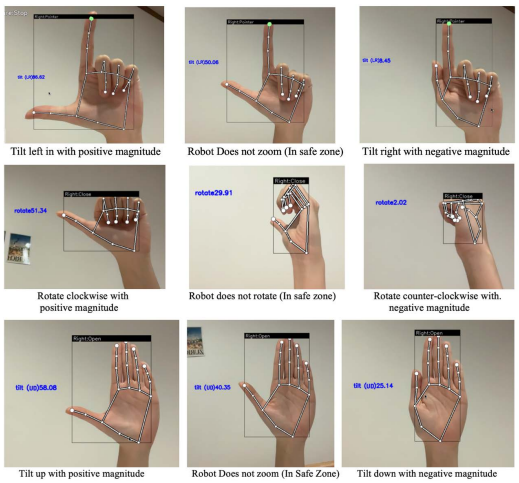

Fig. 3 Hand gesture of command "Tilt left-right", "Tilt up-down", "Rotate" with magnitude adjustment, respectively.

Each gesture is assigned three specific hand landmarks for angle calculation, with the different angle range. This angle serves as a parameter to adjust the robot's magni-

tude for the respective command, which is used to control robot's speed.

These gestures were chosen because the thumb-index provides largest angle, optimal for magnitude control in 'Tilt' commands. The remaining fingers distinguish between left-right and up-down movements as this provides a clear and comfortable gestures. For the 'Rotate' command, wrist offers smaller angel and more appropriate range for the robot's rotational movement, which does not require high speed. The close palm gesture is selected to avoid confusion with other commands

2.2.3. Receiving and preprocessing images

The system uses OpenCV to capture, mirror, and convert video frames into RGB format images. MediaPipe then detects the hand and extracts 21 landmark coordinates per frame (refer to Fig. 4) from the captured images. The image is preprocessed for scale and normalizing landmark coordinates relative to the WRIST landmark.

Fig. 4 Hand landmarks from the MediaPipe library correspond to 21 distinct locations on a hand (Original image courtesy of Google AI).

2.2.4. Training Hand-gesture classification model

The hand gesture classification model was trained using recorded data of the hand-landmark coordinates. The training data includes 42 features (x and y coordinates for each landmark), which is obtained by manually assigning correct gesture to each list of hand landmark.

The model used is a feed-forward neural network designed using TensorFlow, structured as a sequential stack of layers: an input layer, dropout layers (20% and 40%), two dense layers (ReLU activation), and an output layer (softmax activation). The model was trained over 1,000 epochs with 128 mini-batches and early stopping after 20 stagnant epochs, using validation data for evaluation. After the training is completed, the model is then converted and saved as a TensorFlow Lite model.

2.2.5. Hand-gesture Recognition Classification Performance

The model is tested with the unseen test data of different hand gestures' landmarks. The result is shown in Fig. 5. The class 0,1,2 is 'Tilt left-right', 'Rotate', and 'Tilt up-down', respectively. The accuracy of the classification is 0.97 on average.

2.2.6. Magnitude of command calculation

The process for calculating the magnitude is consistent across all gestures. This is achieved by using three specific landmarks on the hand gesture to create an angle.

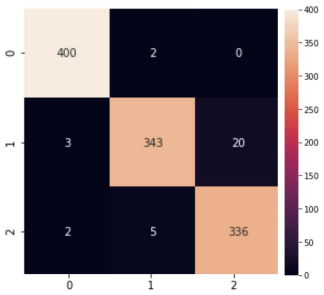

Fig. 5 Result of classifying test data of three different hand gestures, x-axis shows predicted gestures and y-axis shows real gestures.

The value of an angle influences the robot's speed, with an increase in the angle leading to a higher speed.

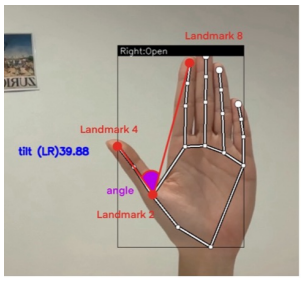

Fig. 6 Method used for calculating command magnitude from the angle between 3 landmarks.

For example, the "Tilt up-down" command the system uses the landmark [8, 2, 4] as shown in Fig. 6. The output angle decreases as the thumb nears the index finger and increases as it moves away.

The calculated angle is then normalized to a range from -30 to 30. Angles within the range of -5 to 5 are considered the "Safe Zone", where the detected gesture's magnitude is near zero. Consequently, the robot remains stationary and does not move, even if a hand command is detected.

Table 1 Hand gesture command specifications

Command	***Tilt (LR)***	***Rotate***	***Tilt (UD)***
Angle Landmark	[8, 2, 4]	[4, 0, 17]	[8, 2, 4]
Angle (min, max)	(10º, 90º)	(0º, 50º)	(20º, 60º)

2.3. Universal Robot Arm

This section details the movements for each command, the integration of hand gesture recognition, and the algorithm to control the robot. The URBasic library is implemented for UR control using URScript. Figure 7 illustrates the x, y, and z axes used as references.

2.3.1. Robot connection and receiving command

The connection between the Universal Robot Arm and the hand gesture system is established via Ethernet using Real-Time Data Exchange (RTDE). The control code acts as a client, receiving commands and magnitudes from the MediaPipe server as Python tuples.

The received hand gesture data is structured as a tuple containing a Hand-Sign ID and its magnitude (refer to

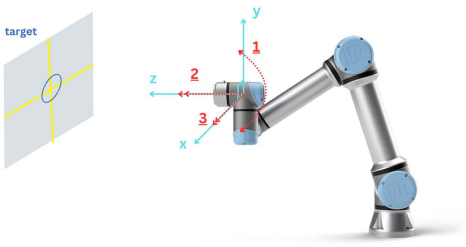

Fig. 7 Universal robot coordinate setup.

Table 1). The magnitude is normalized into real-world robot movement based on the command type (-0.52 to 0.52 radians).

2.3.2. Robot movement

To ensure the safety of the process, the laparoscope must remain stationary at the entry point, where the laparoscope passes through the incision site and into the patient's body as lateral or in-out motion may cause tissue damage. This system uses the UR arm's Tool Center Point (TCP) function, set at the laparoscope's mid-length, to allow controlled tilting and rotation while maintaining a stationary entry position. The movement of the robot is described below according to the axis from Fig. 7.

- **Command 1: "Tilt left-right"** The tip of the laparoscope rotates about y-axis, using the TCP point as a pivot.
- **Command 2: "Rotate"** The laparoscope remains at the same location, rotating cylindrically around z-axis.
- **Command 3: "Tilt up-down"** The tip of the laparoscope rotates about x-axis, using the TCP point as a pivot.

The robot's movement is mapped in 3D using the math3D library, with position (X, Y, Z) and rotation around each axis. Commands are normalized, stored as 6D vectors, and converted into Euler angles. The Euler angles describe the 3D rotation around three axes, represented as pitch, yaw, and roll relative to a fixed coordinate system. This allows the system to define the relative rotation and the robot to perform all the basic laparoscope movements shown in Fig. 2.

2.4. Emergency Stop System with Depth-Aware Safety Controls

To address concerns regarding patient safety during robot-assisted minimally invasive surgery, we designed and implemented an emergency stop mechanism that prevents the laparoscopic robot's camera from approaching critical anatomical structures too closely, enhancing surgical safety during camera navigation.

2.4.1. Custom YOLOv11 Model Training in Real Environment

A custom YOLOv11 detection model was trained on annotated laparoscopic images of the abdominal wall, gallbladder, and liver, collected from our experimental setup. The model achieved a Precision of 0.9986, Recall of 1.0000, mAP@50 of 0.995, mAP@50–95 of 0.9428, and a Fitness Score of 0.9480.

2.4.2. Depth Model Evaluation and Transition

Initially using MiDaS by Ranftl *et al.* [18] [19], we observed unstable depth predictions at short range. After evaluating MiDaS variants against the Endo-Depth model [20], we selected Endo-Depth for its superior close-range stability in laparoscopic environments, achieving significantly lower mean absolute error at 1-10 cm distances.

Table 2 Depth model performance on liver and abdominal wall in centimeters

Model	Distance	MAE	Bias	SD
Endo-Depth	1-10	1.11	0.00	1.54
MiDaS_SMALL	1-10	5.38	5.32	3.95
MiDaS_HYBRID	1-10	5.49	5.04	3.85
MiDaS_LARGE	1-10	5.64	5.33	4.09
Endo-Depth	10-35	6.32	0.00	8.22
MiDaS_LARGE	10-35	9.60	-7.54	10.68
MiDaS_HYBRID	10-35	9.85	-7.14	11.31
MiDaS_SMALL	10-35	10.27	-7.54	11.07

The system uses YOLO to detect anatomical structures and queries Endo-Depth's depth map to trigger emergency stops when targets fall within predefined safety thresholds.

2.5. Testing environment

The test environment, as shown in Fig.8, consists of the UR3e arm holding a laparoscope tube (a lightweight carbon tube with its end attached to a USB endoscope inspection camera) in a standard laparoscopic simulation box trainer. The laparoscope passes through a small incision and a display monitor from the laparoscope receives hand commands through the depth camera from the operator

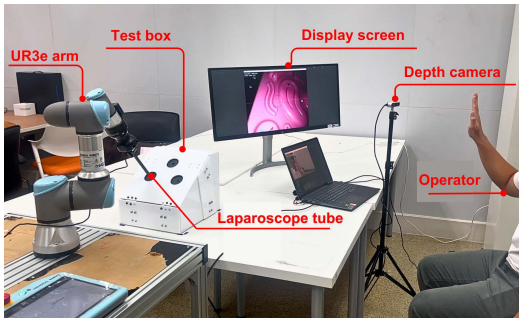

Fig. 8 System experiment setup.

In this setup, the laparoscope camera operates in a dark environment, with the only light source originating from the UR-mounted camera itself. Inside the test box, a laparoscopic suture simulation pad is positioned, where five target points are assigned (refer to Fig. 9). The UR-mounted camera captures a section of the testing tissue, aligned with the display for visualization. The five target points on the testing tissue are randomly assigned to evaluate the system's performance under realistic conditions.

The experimental protocol required participants to navigate the laparoscope camera using predefined hand gestures, moving the aim point from an origin position to five designated target locations. Each target point was tested three times.

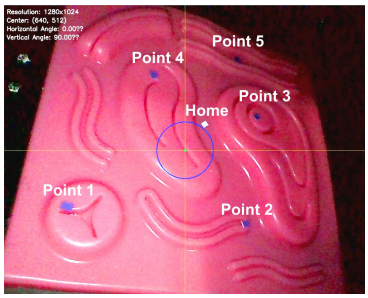

Fig. 9 Visualization of five target points (purple) and a home point (white).

3. PERFORMANCE EVALUATION

This section presents four metrics used to evaluate the system's accuracy, efficiency, and responsiveness. These evaluations were conducted with five engineering students, to establish a baseline technical performance.

3.1. Positional Accuracy

Objective: To evaluate the system's spatial precision in positioning the camera to the target point.

Method: Participants used hand gestures to move the camera to predefined target points inside a laparoscopic box. The Euclidean distance (in mm) between the camera's endpoint and the center of each target was measured. A trial was considered successful if the endpoint fell within the predefined target zone (Fig. 10).

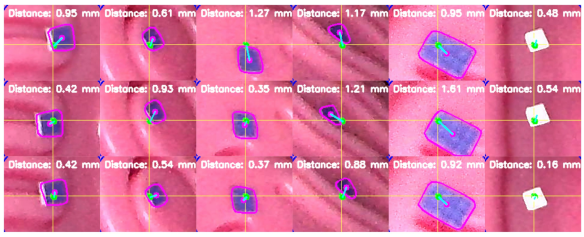

Fig. 10 Visualization of the camera's aim (green) and target points (purple) during trials. A trial is successful when the camera's center overlaps the target zone.

Results: Out of 89 trials, 84 were successful (94.38%). The mean, median, and standard deviation of the distance error were 0.65 mm, 0.56 mm, and 0.40 mm, respectively.

Discussion: The system demonstrated high spatial accuracy and consistency in reaching the target zones, validating its suitability for precise camera positioning in constrained environments.

3.2. Task Completion Time

Objective: To assess how efficiently users could navigate the camera to targets using hand gestures.

Method: The duration from the starting point to the target point was measured in seconds for each participant.

Results: The average completion time across five participants was 24.12 seconds. A two-way ANOVA showed no statistically significant difference ($F = 1.21$, $p = 0.31$).

Discussion: Although not significantly different among participants, the average time confirms that gesture-based control achieves comparable efficiency across users with minimal learning.

Table 3 Two-way ANOVA results of each metric tested

Performance Metrics	Average	F-value	p-value
Completion Time (s)	24.12	1.21	0.31
Number of Command	3.80	2.98	0.02
System delay (ms)	62.02	2.88	0.03

3.3. Number of Gesture Commands Used

Objective: To quantify gesture efficiency based on the number of commands issued per task.

Method: The number of gesture commands needed to move the camera from one point to another was recorded.

Results: The average number of commands used per task was 3.80. Statistically significant differences were observed among participants ($F = 2.98$, $p = 0.02$).

Discussion: Variability in command count reflects individual control strategies. In this study, "intuitiveness" refers to how easily participants could perform camera movements using hand gestures without explicit instruction or repeated trial and error. Some participants favored large movements, requiring fewer commands, while others preferred small, precise movements, increasing the command count. These behavioral differences underline the need for adaptive control tuning to accommodate diverse user preferences.

3.4. System Delay

Objective: To measure the time between gesture initiation and robot response.

Method: System delay was calculated from 60 fps video recordings by counting the number of frames between a gesture and the corresponding visual system response.

Results: The average delay was 62.02 ms. Gestures such as 'Tilt Right' and 'Tilt Left' showed the highest delays (73.81 ms), while 'Tilt Up' and 'None' had lower delays (44.45 ms). See Table 4.

Discussion: While the system performs well in real time, reducing latency for complex gestures could enhance responsiveness during dynamic tasks.

Table 4 Average Delay Time for each Gesture

Command Gesture	Average Delay Time (ms)
None	44.45
Tilt Down	55.00
Tile Up	45.24
Tilt Right	73.81
Tilt Left	72.92

4. CONCLUSION AND FUTURE WORK

This study developed and evaluated a hand gesture-controlled robotic system for laparoscope manipulation, aimed at improving precision, usability, and efficiency in minimally invasive surgeries while eliminating the need

for a separate camera operator. Integrating real-time gesture recognition with a UR3e arm, the system enables intuitive, contactless control over laparoscope movements such as tilting and rotating. The system achieved a 94.38% camera positioning success rate, with a mean positional error of 0.65 mm and consistent completion times.

Nevertheless, variability in command frequency and gesture delay points to areas for refinement. Enhancing classification accuracy and reducing latency for high-delay gestures will improve usability and responsiveness.

One limitation of this work is the use of large hand gesture commands to direct laparoscope manipulation, which could be restricted in a real-world setting in a surgeon already using both hands to control laparoscopic instruments and in whom instrument release to control camera movement with a hand gesture may not be practical. Future work includes creating a library of more subtle and precise hand gestures that could be carried out even while the surgeon's hands are engaged with instruments so as not to interrupt the flow of surgery. Further evaluation of the emergency stop system could include tracking activation frequencies and examining specific scenarios to further validate potential collision. Additionally, a direct comparison with alternative control methods, such as joysticks, voice interfaces, or traditional approaches could evaluate differences in user intuitiveness, system efficiency, and physical demands.

REFERENCES

[1] U. o. N. C. The Department of Urology, "Robotic and minimally invasive surgery," n.d. Accessed: January 16, 2025.

[2] A. Pakula and A. Young, "East monthly literature review," September 2020. Accessed: October 25, 2024.

[3] S. Belsley, "What tools are used in laparoscopic surgery?," n.d. Accessed: October 25, 2024.

[4] A. Okamura, "Haptic feedback in robot-assisted minimally invasive surgery," *Curr. opin. urol.*, vol. 19, pp. 102–7, 02 2009.

[5] S. Lünse, E. Wisotzky, S. Beckmann, C. Paasch, R. Hunger, and R. Mantke, "Technological advancements in surgical laparoscopy considering artificial intelligence: a survey among surgeons in germany," *Langenbecks Arch. Surg.*, vol. 408, p. 405, 10 2023.

[6] K. Tadano and K. Kawashima, "A pneumatic laparoscope holder controlled by head movement," *Int. J. Med. Robot. Comput. Assist. Surg.*, vol. 11, no. 3, pp. 331–340, 2015.

[7] M. L. B. Freitas, J. J. A. Mendes, T. S. Dias, H. V. Siqueira, and S. L. Stevan, "Surgical instrument signaling gesture recognition using surface electromyography signals," *Sensors*, vol. 23, no. 13, 2023.

[8] M. Jacob, Y.-T. Li, and A. Akingba, "Gestonurse: A robotic surgical nurse for handling surgical instruments in the operating room," *J. Robot. Surg.*, vol. 6, 03 2011.

[9] N. Arjanurak, T. Oranrigsupak, Y. Suksawasdi, and R. Chaichaowarat, "Hand gesture-guided manipulator for enhancing human-robot collaboration," in *Proc. Int. Conf. Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology*, pp. 1–6, 2025.

[10] K. Pornpipatsakul, A. Chenviteesook, and R. Chaichaowarat, "Ultrasound probe movement analysis using depth camera with compact handle design for probe contact force measurement," in *Proc. IEEE EMBS Annu. Int. Conf.*, pp. 1–4, 2023.

[11] Universal Robots, "Ur3e collaborative robot," 2025. Accessed: Feb. 3, 2025.

[12] S. Kitchatr, A. Sirimangkalalo, and R. Chaichaowarat, "Visual servo control for ball-on-plate balancing: Effect of pid controller gain on tracking performance," in *Proc. IEEE Int. Conf. Robotics and Biomimetics*, pp. 1–6, 2023.

[13] A. Chico, P. J. Cruz, J. P. Vásconez, M. E. Benalcázar, R. Álvarez, L. Barona, and L. Valdivieso, "Hand gesture recognition and tracking control for a virtual ur5 robot manipulator," in *Proc. 2021 IEEE Fifth Ecuador Technical Chapters Meeting (ETCM)*, pp. 1–6, 2021.

[14] K. Takahashi, "Hand gesture recognition using mediapipe." GitHub repository, 2024. Available at: https://github.com/Kazuhito00/hand-gesture-recognition-using-mediapipe, Accessed: Feb. 2, 2025.

[15] M. M. Marinho, M. C. Bernardes, and A. P. L. Bo, "Using general-purpose serial-link manipulators for laparoscopic surgery with moving remote center of motion," *J. Med. Robot. Res.*, vol. 1, Aug. 2016.

[16] A. Majewicz, S. Payandeh, and A. M. Okamura, "Friction dynamics of trocars: Effects on tool motion and surgeon perception," in *Proc. IEEE Int. Conf. Robotics and Automation*, pp. 2342–2347, 2014.

[17] Y. Yamazaki, T. Okamoto, K. Masamune, I. Sakuma, and T. Dohi, "A new laparoscope manipulator with an optical zoom," in *Proc. Int. Conf. Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pp. 121–128, Springer, 2004.

[18] R. Ranftl, K. Lasinger, D. Hafner, K. Schindler, and V. Koltun, "Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 3, pp. 1623–1637, 2022.

[19] R. Ranftl, A. Bochkovskiy, and V. Koltun, "Vision transformers for dense prediction," in *Proc. IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 12159–12168, 2021.