A Touchless Control Interface for Low-Cost ROVs

KAGAN KAPICIOGLU
Mechatronics Engineering
Bahcesehir University
Istanbul, Turkey
0000-0002-9220-307X

ENIS GETMEZ
Software Development
Turkol Yazılım Ltd. Sti.
Istanbul, Turkey
0000-0002-8140-1927

BATUHAN EKIN AKBULUT
Mechatronics Engineering
Bahcesehir University
Istanbul, Turkey
0000-0001-6445-8801

ARDA AKGUL
Mechatronics Engineering
Bahcesehir University
Istanbul, Turkey
0000-0001-7070-1708

BURAK UCAR
Mechatronics Engineering
Bahcesehir University
Istanbul, Turkey
0000-0002-4449-7827

BERKE KANLIKILIC
Mechatronics Engineering
Bahcesehir University
Istanbul, Turkey
berke.kanlikilic@gmail.com

MEHMET KOC
Electrical and Electronic Engineering
Bahcesehir University
Istanbul, Turkey:
0000-0003-1469-8852

BERKE GUR
Mechatronics Engineering
Bahcesehir University
Istanbul, Turkey
0000-0002-9391-0905

***Abstract*—In this paper, a fully touchless control interface for low cost remotely operated vehicles (ROVs) is presented. This interface aims to decrease training time, reduce workload, and ensure the operation ergonomics of ROV operators. Fully touchless control interface is achieved by a machine learning (ML) algorithm for ROV operator's face and orientation recognition, and controling the angle of an ROV-based camera. Furthermore, a Leap Motion sensor captures hand gestures and movements, thereby allowing the ROV operator to execute maneuvers and perform other functions (e.g., gripper or lighting control) based on pre-determined hand gestures. Fusion of face and hand gestures allows the operator to control ROV in a fully touchless way. The proposed system is tested in a realistic underwater simulation environment designed specifically for typical tasks that are present in student competitions. Trials with inexperienced operators show that the touchless interface can cut training times, speed up operations, reduce workload, and can provide the operator with a more natural feeling of command and control as well as better ergonomy.**

Keywords—Touchless control interface, human machine interfaces, Leap Motion, machine learning, ROV.

I. INTRODUCTION

Due to the discomfort associated with wearable devices, human machine interfaces (HMI) with touchless control is gaining significant importance in teleoperation applications [1]. Operator training for teleoperated robotic systems is a cumbersome task mainly due to the time it takes for the familiarization of the operator to specialized command and control hardware such joysticks and virtual reality (VR) headsets. Remotely operated vehicles (ROV) are a type of teleoperated robotic systems that are used in underwater observation, monitoring, search and rescue, retrieval, and other key missions. Currently, numerous low-cost ROV systems are being developed and deployed by non-specialists with limited resources and training in ROV operations. These low cost ROVs are more frequently exploited in missions and tasks that are normally ascribed to expensive, commercial ROV systems creating great opportunities to further explore the oceans and sustainable exploitation of sub-sea resources [2]. Contemporary low-cost mini-ROV's are highly agile and maneuverable, making them difficult to pilot and control. Most ROV's are equipped with robotic manipulators and other devices for underwater intervention. Furthermore, even the modern low-cost ROV's are equipped with a wealth of sensors, streaming data at a rate that generally exceeds human perception capabilities.

In this work, a touchless control interface for low-cost ROV is presented. The proposed system is designed to meet several objectives including improving ROV operation ergonomics and reducing operator workload, decreasing ROV operator training times, and providing ROV operators with a customizable user interface that can be programmed to recognize desired personal gestures. The proposed touchless control interface is integrated, tested, and validated on a low-cost, highly maneuverable, compact and portable mini-ROV (named 'TurtleBot') developed by Bahcesehir University Autonomous Underwater Systems group (BAUROV), in a simulation environment. TurtleBot is equipped with eight thrusters providing omni-directional motion capability in six degrees-of-freedom (DoF) (see Fig. 1).

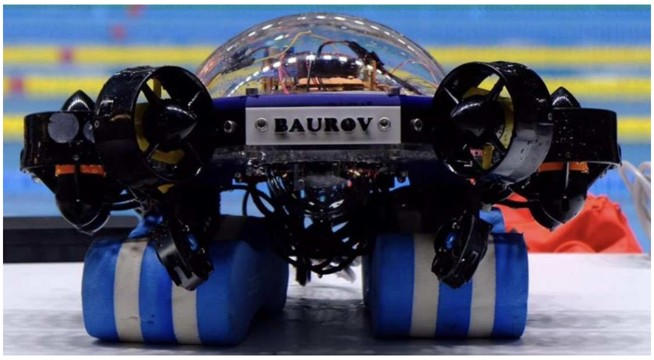

Fig. 1. TurtleBot

From the human-machine interaction perspective, the ROV and the operator need to act like one uniform body to provide a better sense of control [3]. In this scope, the proposed touchless control interface includes a machine learning (ML) algorithm for controlling the camera look angle by detecting the operator's face orientation and recognizing hand gestures for maneuvering the ROV in the desired direction. This automation provides a more natural sense of control without any wearable headsets or other devices which may cause discomfort to the operator. The Leap Motion sensor was used for commanding ROVs in the UWSim underwater simulator environment and operator satisfaction was measured [4]. This sensor was also used to control the Girona500 AUV for very simple motion primitives [5].

In this work, a customizable user interface is developed and integrated to the TurtleBot controller. The operator can program the control inputs according to desired gesture not only for the movement of the ROV, but also gripper (or any other tool) control. It is also possible for two operators to simultaneously interact with the ROV, reducing operator workload. Preliminary tests of the interface were performed in a simulation environment which hosts simple obstacles such as hoops, squares and triangular frames for passing through. Customization of the simulation environment also enables the diversification of tests as well as expansion to underwater manipulation tasks.

II. ROV System

In this section, the mechanical, electronic, communication and control systems of the ROV used together with the touchless interface system is explained. The TurtleBot V2.0 (see Fig. 2) is compact and lightweight, making its deployment relatively easy. The electronics (see Fig. 3) are placed inside a water-tight housing and can provide omni-directional motion control. For this purpose, the Pixhawk autopilot card, which controls the thrusters in TurtleBot and provides a stable operation using onboard sensors (IMU, Compass, Gyro, Depth, etc.) is used. A RaspberryPi Model 3B acts as a communication interface between the wet and dry ends through an ethernet connection. Communication and control is achieved via the ArduSub [6] open-source control architecture.

Fig. 2. Physical dimensions of the TurtleBot

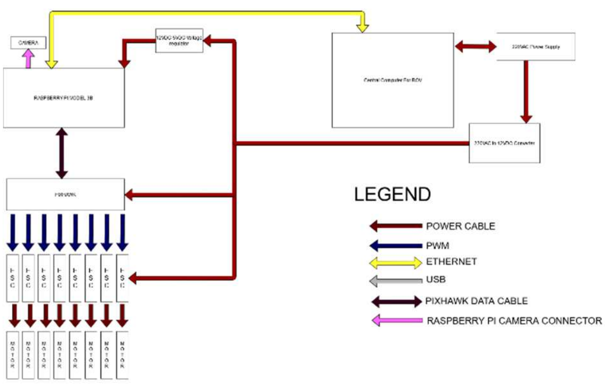

Fig. 3. Electronic Hardware

In addition, two Newton Subsea Grippers placed in the front of the vehicle allows underwater object manipulation. A reconfigurable chassis provides the ability to install various sensors, grippers or other tools, allowing TurtleBot to be reconfigured and customized for the mission. TurtleBot has logged nearly 500 hours of dive time in pools and open-water environments. An onboard auto-pilot with hover and stable motion capabilities is employed since the proposed touchless interface performs better with a stable ROV.

III. Touchless Interface

At-the wet-end, the touchless control interface includes an onboard 2-axis servo-controlled high-definition camera (see Fig. 4) capable of performing automatic pan and tilt motion as well as the onboard control interfaces. On the dry-end, the system consists of a camera for face recognition and tracking, and a Leap Motion sensor for hand gesture recognition.

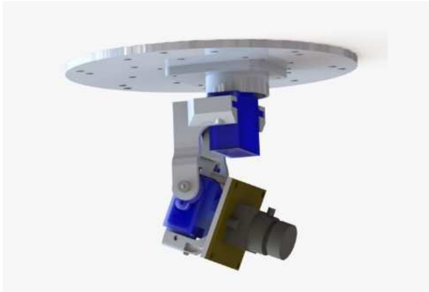

Fig. 4. The 2-Axis Servo Controlled Camera System

The schematic of the touchless control interface is shown in Fig. 5. Two different control interfaces (namely, the hand-gesture and face orientation recognition systems) are combined in the central computer which is responsible for controlling the ROV. The face orientation recognition system is responsible for arranging the camera angle according to the ROV operator's look direction by controlling the pan-tilt gimbal. The Leap Motion sensor is responsible for capturing the hand gestures.

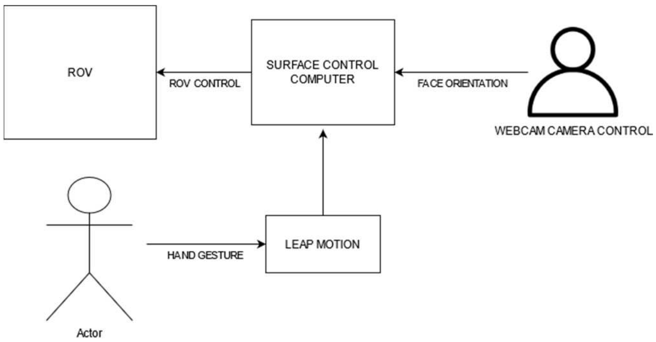

Fig. 5. Schematic of the Touchless Control Interface

A. Machine Learning Algorithms for Gimbal Control

The ROV pilot has to rely only on the camera inside the ROV's dome to navigate the vehicle, and to see or recognize objects in an open-water environment. Due to the limited field of view provided by the dome, this onboard camera needs to be mobile to allow the pilot a better view. The orientation of this onboard camera is controlled by an machine learning (ML) algorithm that recognizes the operator's face and estimates its orientation. A pre-existing dataset is used for the training of the ML algorithms [7][8][9]. The recognition and tracking of the operator's face is performed by a laptop webcam (see Fig. 6). The ROV-based onboard camera's orientation is aligned with the estimated orientation of the operator's face. Hence, the ROV operator can change the camera angle and view the surrounding environment simply by turning his/her face to the desired look direction.

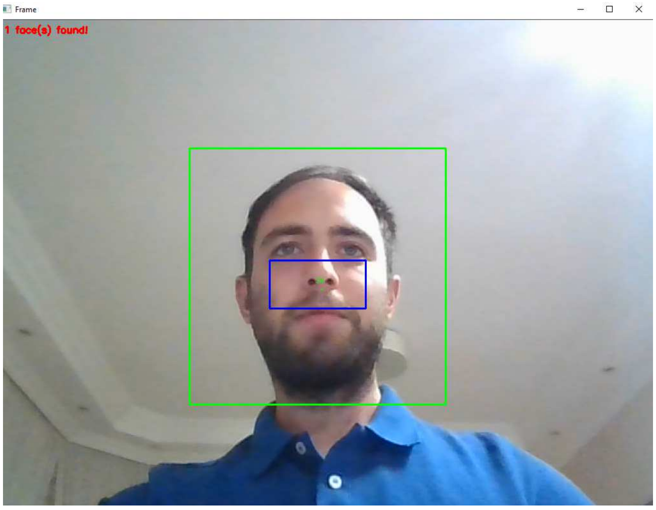

Fig. 6. Face Tracking

The ML algorithm tracks a total of 61 points on the operator's face (as seen in the Fig. 7) to recognize and localize facial features (e.g., nose, eyes, jaw) [9]. Point 34 is chosen as a reference point because it is is located in the middle of the face and can be recognized robustly for different faces (see Fig. 7). This point is used for determining the orientation of the ROV operator's face.

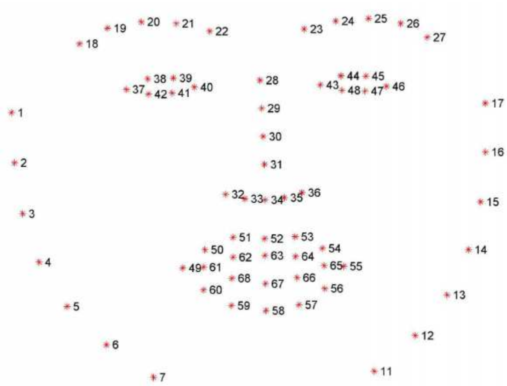

Fig. 7. Point Cloud for Face [9]

The camera images used for facial recognition (shown in Fig.6) has a resolution of 1024 x 768 pixels (w x h). Since every person has a different head size, 12 volunteers (8 males and 4 females) with an age of 23 ± 3 years, height of 173 ± 13 cm, and mass of 69.5 ± 17.5 kg) were selected for participating in experiments in defining the rectangle that covers the faces in the middle of the above mentioned pixel. Analyzing the collected data revealed that the rectangle (green rectangle in Fig. 8) which has the points at (300, 200) pixels (lower left point) and (700, 600) pixels (upper right point) supply suitable boundaries for face identification. It is essential to detect the orientation of identified face according to point 34, as shown in the Fig. 7. The same subjects again were used for identifying another rectangle which can cover the mentioned point even if the operator changed. Furthermore, this rectangle is used to determine the orientation of the face according to position of point 34 relative to the rectangle. The same procedure is utilized to obtain the blue rectangle with corner points at (425, 375) pixels (lower left point) and (575, 450) pixels (upper right point). The entire system is shown in the Fig. 8.

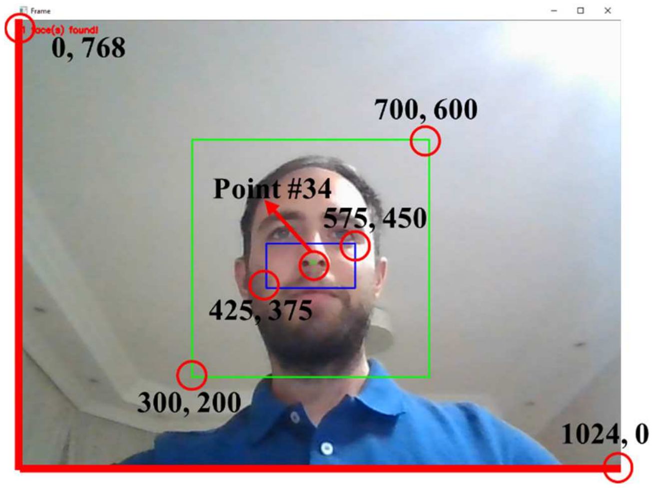

Fig. 8. System Structure of Face Tracking

If point 34 is inside the blue rectangle, then the operator is looking straight ahead and there is no need to change the orientation of the camera. Using a simple mirroring method (because the integrated camera looks towards the operator's face), if point 34 goes out of the blue frame in the left direction (see Fig. 9), this means that the operator is looking to the right of the ROV. In this case the gimbal turns the camera to the right, allowing the operator to see whats right of the ROV. The same procedure is applied for movements for looking in left, up, down, right-up, right-down, left-up, and left-down directions. The proposed gimbal control system is compsed in Python. The face orientation is correlated with simple pulse-width modulation (PWM) signals and sent through the corresponding pan and/or tilt servo motors. Thereby, the look direction of the camera is controlled by this proposed system in a fully touchless way.

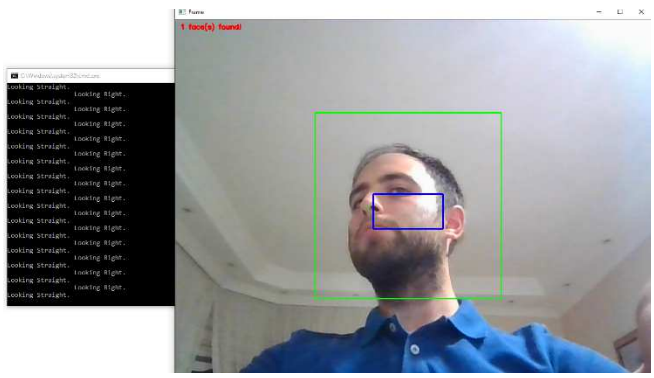

Fig. 9. Directing Gimbal to Right Direction

B. Measuring Hand Gestures

Leap Motion consists of two LEDs (infra-red and visible light) and two corresponding cameras to measure distance or depth. The sensor can track a single or two hands (see Fig. 10) [10]. The hardware comes with the plug and play software which allows users to capture pre-determined hand gestures. Eight of these pre-determined hand gestures are paired with with ROV maneuvers and payload functionality which are forward-backward, lateral, left-right, throttle up-down, and gripper clutch open-close.

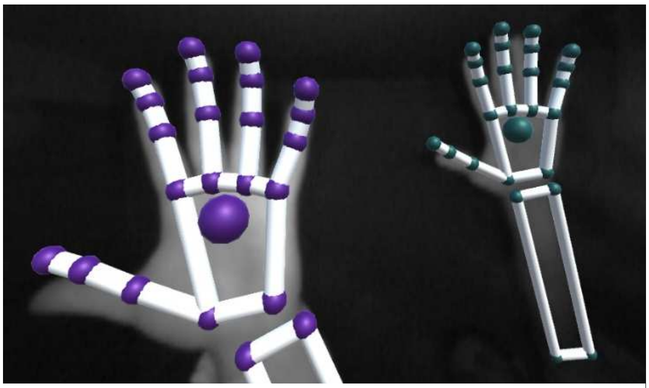

Fig. 10. Hand Skeleton Output from Leap Motion

Motion primitives are related to certain essential ROV maneuvers. Hand gestures and maneuvers are matched based on the suggestions of the previously mentioned 12 volunteers. This capability of customizing hand gesture-motion command pairs (such as forward-backward, lateral, left-right, throttle up-down) provides a great flexibility to ROV operators. To provide a more natural sense and control, a closed fist is correlated with gripper close and vice versa (as seen in Fig. 11 and Fig. 12).

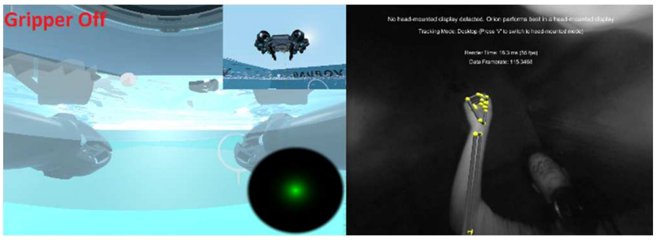

Fig. 11.. Close Fist Gesture for Closing the Gripper

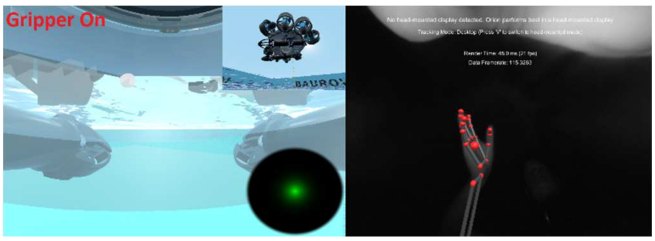

Fig. 12.. Open Fist Gesture for Opening the Gripper

Other maneuvers are correlated with hand gestures in a meaningful way (see Fig. 13-17) to achieve direct gesture coordination even for inexperienced operators. In addition, the ability of the Leap Motion sensor to track two hands allows one to combine several gestures at the same time. This in turn, allows the ROV operator to perform more complex tasks. Capturing two different hands from two different operators is also possible, allowing multiple operators to handle cooperative tasks effectively.

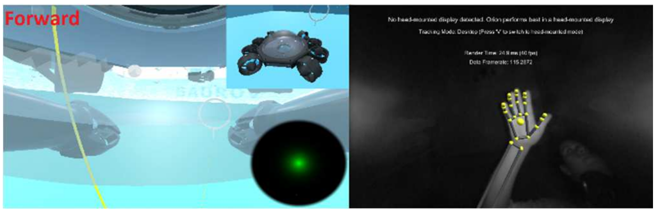

Fig. 13.. Keeping the Hand Straight Stands for Forward Motion

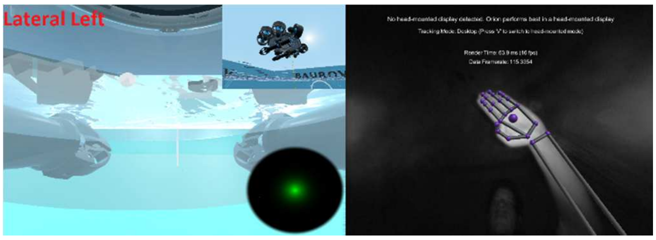

Fig. 14.. Turning the Hand Left Stands for Lateral Left Motion

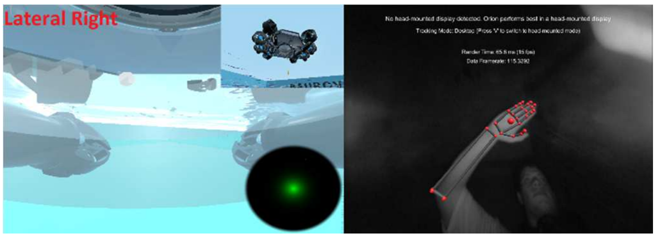

Fig. 15.. Turning the Hand Right Stands for Lateral Right Motion

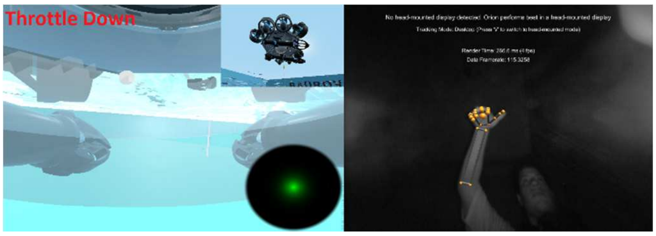

Fig. 16. Lowering the Hand Stands for Throttle Down Motion

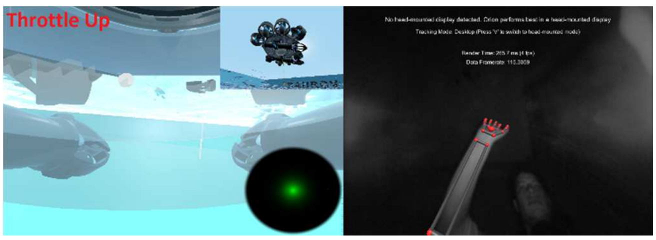

Fig. 17. Raising the Hand Stands for Throttle Up Motion

IV. Test Setup

BlueSim open-source simulation environment [11] is used for validating of the proposed system. The simulation environment can be seen in Fig. 18.

Fig. 18. Simulation Environment

BlueSim can be modified and customized using different extended reality engines. In this case, the TurtleBot and pool environment which hosts circular, rectangular, and triangular frames or obstacles (see Fig. 19 and Fig. 20) are implemented in .gltf format. This Simulation environment allows to modify the size of the pool and depth according to needs. The refresh rate can be set between 60-500 Hz to realize the robot motion in a more realistic way.

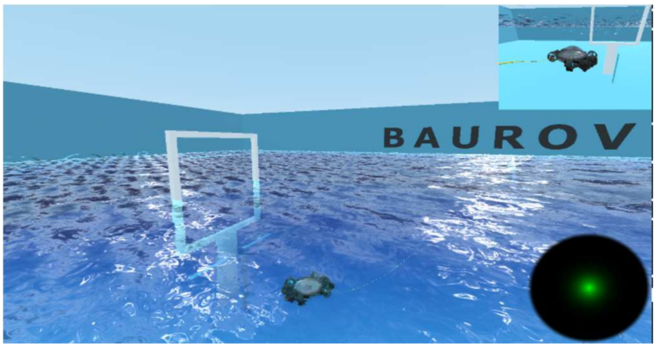

Fig. 19. Square Shaped Obstacle

Fig. 20. Circle Shaped Obstacle

The simulation environment allows users to operate the vehicle through specific keys through keyboard and with joystick. It also allows the operator to choose the driving viewpoint. To catch a real driving sense in the simulation, the driving viewpoint is set where the camera is located at TurtleBot (see Fig. 21). Also, the pan-tilt motion of the driving camera allows the user to observe the environment. The test setup is kept simple because the volunteer operators had no prior experience in piloting an underwater robot. The path that the robot is expected to navigate is designed with the starting point being a square shaped obstacle and the ending point a circular shaped point. Every subject is asked to navigate the ROV and pass through these frames during trials . This path is traversed by the volunteer operators twice, once using a joystick and next using the proposed touchless system. Every subject is given 1 minute of training time.

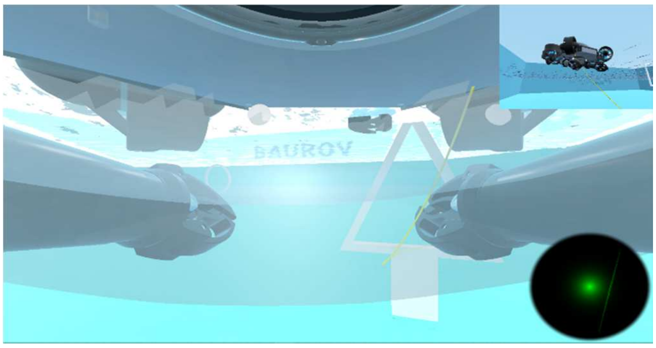

Fig. 21. Driving View

A. Observations and Result

Compared to the conventional joystick based controls, it was observed that subjects spent an average of 20 seconds less to get used to controlling the robot with the proposed touchless interface system. Around 58% had difficulty while using the joystick buttons which include maneuver sticks, gripper open-close buttons, and buttons for arranging camera view angle. The average time for completing the entire path is 71.5 seconds with joystick while an average of 63.7 seconds was sufficient with the proposed touchless system. This basic test setup showed that subjects spent nearly 13% more time to complete the path using joysticks. It is believed that this time difference will increase when the navigation paths become more complex, and the use of the gripper is required with manipulation tasks.

V. Conclusion

In this work, a fully touchless control interface for ROVs is presented. The proposed touchless system includes an integrated laptop camera for face gestures and a Leap Motion sensor for hand gesture recognition. The proposed system is tested in a realistic simulation environment. This simulation environment hosted obstacles (circular, square-shaped and triangular frames for passing through) for training the ROV operator in some basic maneuvers (forward-backward, lateral, left-right, throttle up-down maneuvers). The main advantages of touchless interface system are reduced training time and costs, and improved operator ergonomics. Customization of hand gestures and being able to assign its corresponding reactions on the vehicle provides flexibility to the operator and a more natural feeling. Also, the proposed simulation environment can be modified by changing the obstacles, adding more complex missions such as manipulation tasks using a gripper, changing the ROV or the pool, arranging the camera view angle etc. This allows ROV operator to train with the proposed system according to the needs. Test results show that the proposed system decreased the training time of the inexperienced subjects as well as speeding up ROV operations by 13%.

VI. Future Work

Future work is expected to mainly focus on following ideas.

- Building a More Complex and Realistic Simulation Environment: Tasks that require gripper or complex movements like pitch, roll or yaw maneuvers needs hand – eye coordination. Complex tasks give better results while comparing proposed system and classical systems.
- User Friendly Graphical User Interface (GUI): By allowing the proposed system to customize by simple drag and drop functionality for correlating hand gestures and correspondence on ROV supply more natural feelings and flexibility to ROV operator just before start driving.
- Pool and Open-Water Tests: Real life tests give better results in experiments. There are more physics and unpredictable factors at open-water environment which can directly affect the ROV operator while driving.

Acknowledgment

This project is fully funded by Bahcesehir University Technology Transfer Office (TTO) under the scientific research projects, project number BAUBAP 2020-II.28. We would like to thank the BAUROV Team for allowing us to use their mini-ROV TurtleBot for this project.

References

[1] S. Prakasam, M. Venkatachalam, S. Manickam, N. Pradheep, “Gesture recognition using a touchless sensor to reduce driver distraction," Int Res J Eng Tech, vol. 3, no. 9, pp. 221-224, 2016.

[2] R. Miao and S. Pang, “Development of a low-cost remotely operated vehicle for ocean exploration,” MTS/IEEE OCEANS 2015, Washington, DC, 2015, pp. 1-7.

[3] J. C. García et al., “Towards an immersive and natural gesture controlled interface for intervention underwater robots," OCEANS 2015 - Genova, Genoa, 2015, pp. 1-5.

[4] J. C. García et al., “A natural interface for remote operation of underwater robots,” IEEE Computer Graphics and Applications, vol. 37, no. 1, pp. 34-43, 2017.R. Nicole, “Title of paper with only first word capitalized,” J. Name Stand. Abbrev., in press.

[5] J. Bosch, P. Ridao, R. Garcia and N. Gracias, “Towards omnidirectional immersion for ROV teleoperation,” Proc XXXVII Jornadas de Automatica Conference, Madrid, Spain, 2016.

[6] Overview · GitBook", *Ardusub.com*, 2021. [Online]. Available: https://www.ardusub.com.

[7] C. Sagonas, E. Antonakos, G. Tzimiropoulos, S. Zafeiriou and M. Pantic, "300 faces in-thewild challenge: database and results,” Image and Vision Computing, vol. 47, pp. 3-18, 2016.

[8] C. Sagonas, G. Tzimiropoulos, S. Zafeiriou and M. Pantic, "A Semi-automatic Methodology for Facial Landmark Annotation," 2013 IEEE Conference on Computer Vision and Pattern Recognition Workshops, Portland, OR, 2013, pp. 896-903.

[9] C. Sagonas, G. Tzimiropoulos, S. Zafeiriou and M. Pantic, “300 faces in-the-wild challenge: The first facial landmark localization challenge,” 2013 IEEE International Conference on Computer Vision Workshops, Sydney, NSW, 2013, pp. 397-403.

[10] "Digital worlds that feel human | Ultraleap", *Ultraleap.com*, 2021. [Online]. Available: https://www.ultraleap.com

[11] "GitHub - bluerobotics/bluesim: BlueROV2 Simulator", GitHub, 2021. [Online]. Available: https://github.com/bluerobotics/bluesim.