

USE OF EYE-TRACKING TO CONTROL A COBOT MOVEMENT: RESULTS COMING FROM THE APPLICATION OF TOBII PRO X2-30 TO AN OMRON TM5-700 ROBOT

MARCOS FAUNDEZ-ZANUY, ALEIX MORAL-VIÑALS & JOSEP LOPEZ-XARBAU
Tecnocampus, Universitat Pompeu Fabra, Spain

ABSTRACT

In this paper we present a connection setup to operate a collaborative robot (cobot) Omron TM5-700 by means of a Tobii Pro X2-30 eye-tracker. The application has been designed with the software OMRON TMFlow as well as the Tobii SDK (software development kit). The possibility to operate a collaborative robot by gaze can be used as a third arm, which permits to do more sophisticated activities by human beings but it can also facilitate the manipulation of dangerous or perilous substances, or in unsafe environments where the human being can stay far away of the dangerous area. It can be very useful for a plenty of safety and security applications. To demonstrate the correct operation of the cobot control application, we have implemented a practical demo and experimental setup. It consists of accuracy measurements, where the control of the position of the cobot is defined by means of gaze, which defines a set of points in (X, Y) plane. Later on, the robot picks up an ink-pen and draw a graph in a piece of paper. This drawing consists of connecting these pre-defined dots by straight lines. To this aim, we have acquired a set of figures (parallelogram, pentagon, etc.), and compared the desired printed image on PC screen with the result produced by the cobot.

Keywords: collaborative robot, eye-tracker, gaze.

1 INTRODUCTION

Eye-tracking is a technology with applications in a large set of life areas. The eye tracking market is expected to grow from USD 368 million in 2020 to USD 1,098 million by 2025; this implies grows at a compound annual growth rate (CAGR) of 24.5% during the forecast period [1]. Scientific literature has also experimented an important grow in the last years. A Google scholar search of the terms “eye-tracking” and “safety” reveals 15,400 papers, with half of them produced in the last recent five years. Currently, those areas with the highest penetration of the use of this technology are health and retail, although the trend is that, in the coming years, automotive, neuromarketing and industry 4.0 will stand out. Eye-tracking technology can be very useful for a plenty of safety and security applications. Forecasts suggest that the most frequent applications of this technology in the coming years could be augmented reality and virtual reality, its incorporation into mobile devices, entertainment and gaming, or contactless biometric solutions. Especially after COVID-19, machines operated without touching any surface are more desirable in order to minimize the risk of virus contagion.

- Safety for construction workers. Some examples of research on safety improvement using eye-tracking on construction workers can be found in Jeelani, et al. [2], Hasanzadeh et al. [3] and Martinez-Marquez et al. [4]. Recent studies have also found that a strong correlation exists between viewing patterns of workers, captured using eye-tracking devices, and their hazard recognition performance. The analysis of the viewing patterns of workers can provide a better understanding of their hazard recognition performance.



- Automotive is one of the most important application areas of eye-tracking. The most direct application in this sector is in driver monitoring systems (DMS), where eye-tracking technology is being of vital importance. With the aim of creating safer and more advanced cars, the combination of facial recognition and eye-tracking enables information on the driver's attention, alertness or concentration at the driving task. So we can create warnings and notifications in driving. Some examples can be found in Carr and Grover [5], Vetturi et al. [6], Rjabovs and Palacin [7], Khan and Lee [8], and Le et al. [9]. Also the safe mobility of pedestrians has been analysed [10], [11].
- Neuromarketing is probably the first commercial field where eye-tracking technology entered. For advertising and marketing it is crucial to know the behaviour of customers, their preferences, their decisions and to predict their purchasing behaviour. For this they use neuromarketing techniques such as EEG, eye-tracking or implicit response tests. Regarding eye-tracking technology, it allows to identify, through an exhaustive eye-tracking, behaviour patterns during the visit to the point of sale. Therefore, the applications are endless, from software/hardware usability, advertising testing and product in stores, to analysis of the conception of the corporate image. See for instance the study about neuroergonomics [12].
- Robotics and industry in safe environments is a high potential area too. Eye-tracker widens the possibilities in collaborative robotics (human-robot). For example, with applications in camera inspection tasks, directing the camera with the eyes to the point where the inspection photograph is to be taken, and giving the order to shoot the photo by winking of one eye. Or the concept of the "third arm", which would allow people who in the production plant have to operate certain machinery or tools for which they need both hands, to instruct a robot with their eyes to carry out additional operations on the machinery. When manipulating perilous substances by means of a robot, eye-tracking can be a good solution to minimize the risk for human beings. See Chadalavada et al. [13] for instance. Another important application is the robot control by handicapped people in order to overcome their limitations and to enable an autonomous life [14].

In this paper, we setup the communication between Eye-tracker and cobot and we perform a set of experiments based on drawing performance based on gaze. This is a simple environment test that has already been explored by different authors using different setup [15]. Handwriting analysis has a wide range of applications in e-security and e-health [16], and a large set of tasks can be performed [17]. The possibilities of combining handwriting analysis and eye-tracking is large in e-health and e-security and will be probably developed in near future years.

We consider interesting to describe the setup of the system as a large amount of robots and eye-trackers exists and the interconnection is not trivial and must be ad hoc designed.

2 EXPERIMENTAL SETUP DESCRIPTION

In this paper, we have configured a setup based on the devices available in our lab: a cobot Omron TM5-700 and an eye-tracker Tobii Pro X2-30 attached to a laptop.

2.1 Elements description

Our system uses two different computers which communicate to each other by means of an Ethernet connection. Fig. 1 shows the experimental setup, which consists of the following eight parts:



1. OMRON Cobot-arm model TM5-700. Collaborative robots are designed to work safely with human operators thanks to technologies like force feedback, low-inertia servo motors, elastic actuators and collision detection technology that limit their power and force capabilities to levels suitable for contact. The safety standard ISO 10218-1, ISO 10218-2 and technical specification ISO TS-15066 define the safety functions and performance of the collaborative robot. We have used this robot because it was the unique collaborative robot available in our laboratory.
2. 3D grip attached to the cobot in order to hold/left the marker pen that performs the drawing on the surface area. The marker pen is a standard one, as the robot can use the grip to hold pens of different thick.
3. Drawing surface of standard DIN A3 size, where the cobot performs the drawings by moving its arm.
4. Cobot controller screen, where the cobot is operated by a dedicated computer hidden in a box below the cobot. The computer executes a software programmed with the software OMRON TMFlow, which is a graphical human-machine interface (HMI). This facilitates the programming of a specific algorithm.
5. Cobot Remote control (robot stick) to start/stop the possibility to receive instructions by the robot.
6. Tobii Pro X2-30 Eye-tracker attached to the bottom part of the laptop screen.
7. Eye-tracker laptop, which operates the eye-tracker software programmed in python.
8. Ethernet connection in order to send instructions (communication) from eye-tracker computer to cobot computer.

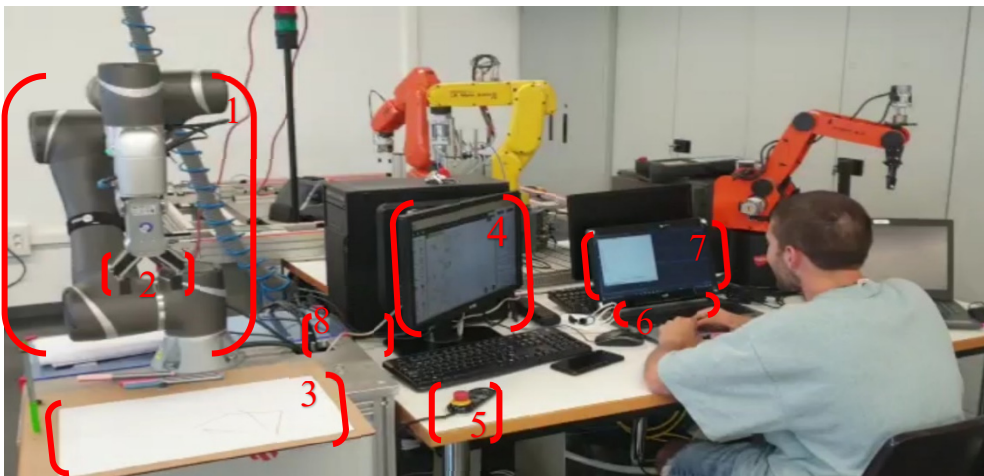


Figure 1: Experimental setup including: (1) Cobot-arm TM5-700; (2) 3D grip; (3) DIN A3 drawing surface; (4) Cobot controller screen; (5) Cobot remote control (3); (6) Tobii Pro X2-30; (7) Eye-tracker laptop; (8) Ethernet connection.

2.2 Technical block description

Fig. 2 shows the technical scheme of communication data flow between eye-tracker and cobot. This scheme evidences the server-client architecture. The server provides the data acquired by the eye-tracker while the client is the cobot.

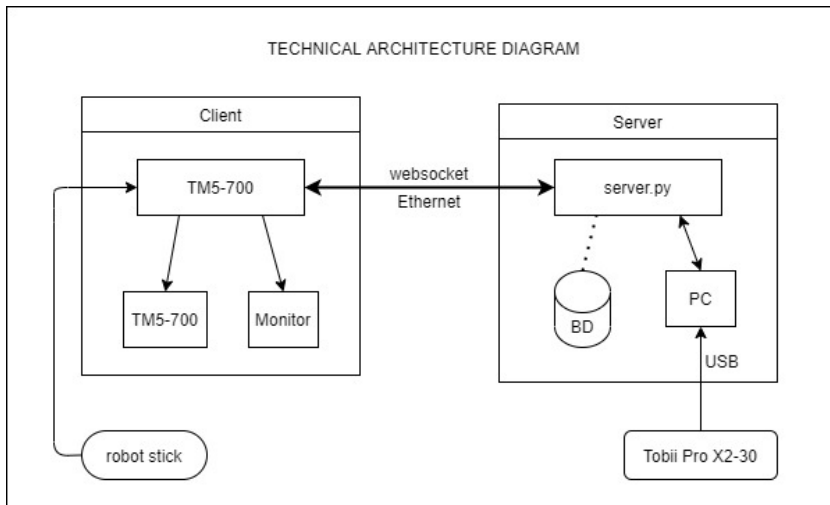


Figure 2: Technical scheme of communication data flow between eye-tracker and cobot.

Despite the fact the eye-tracker includes an automatic calibration tool, we decided to trim it by means of a manual calibration, as depicted in Fig. 3. This calibration has been done once the working area depicted in Fig. 1 was established. This calibration process permits an improved accuracy for the working area. It consists of looking at a predefined set of points (see Fig. 3). This process requires less than three minutes.



Figure 3: Eye-tracker manual calibration process.

2.3 Socket-host Python

Fig. 4 shows the flow chart of the algorithm programmed in python and executed on the eye-tracker computer. Basically, the application can be divided into two parts, the first part, the

left branch, would correspond to the initialization, server establishment, connection between the client and the Tobii Pro X2-30 and finally the capture of the first point, called the reference point from now on.

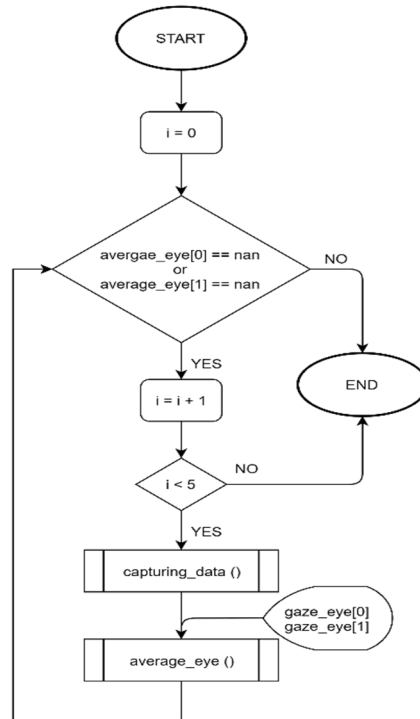


Figure 4: Flow chart of ant_nan () function.

Once the first part is finished, the second one begins, which is the branch on the right part in Fig. 4. This refers to the capture loop. This loop is performed indefinitely until the robot's message does not match what is expected, in which case the resulting dot graph is extracted and ends by disconnecting the server. We have used eight python libraries: Tobii_research, time, socket, pylab, numpy, PIL, winsound and func_application_CRAWLE.

The most important python functions are described below:

- `socket.socket ()`: In this function the server is initialized and the connection with the client is established (cobot). In case there is no established server the server remains opened waiting. The connection is established via TCP/IP by assigning the relevant IP and connection port.
- `find_eyetracker ()`: This function is available in the Tobii_research library application. It is used to search for a Tobii device connected to the machine and establish a connection with it once located. This has been slightly modified to accommodate programming needs and to discard nonrelevant data from the device.
- `capturing_data ()`: This is the Tobii capture function, available at the Tobii_research library application. Basically, this function performs the subscription to the Tobii Pro X2-30 to start data capture (left eye location and right eye location), giving a certain

amount of time for the user to have time to focus and locate the desired point to capture. At the end of the capture time, the last captures of each eye are stored, so that they can then be processed and sent to the cobot. In order to warn the user of the start of the data capture state, a warning noise is emitted thanks to the winsound library application. It should be noted that the data obtained with this function (left eye location and right eye location) have two components (x, y). These are called *gaze_left_eye* and *gaze_right_eye*.

- *average_eye ()*: This function obtains the final lookout point captured by Tobii from the components (x, y) of the captured points (*gaze_left_eye* and *gaze_right_eye*).
- *ant_nan ()*: When the eye-tracker does not directly record an acceptable value or basically does not detect the user's eyes, in the time set in the *capturing_data()* function, the extracted data is *nan* (not a number). This function is used to detect these situations and repeat the capture, that is, to run the *capturing_data()* function again, until either an acceptable capture is obtained or five capture attempts have been made and no one is acceptable. To do this, the loop of the Fig. 4 has been established. At the end of the software it continues as established in the flow chart of the application (Fig. 4).
- *eyetracking_data_array ()*: This function is responsible for storing each capture made during the session in a two-dimensional array. The rows of the matrix correspond to the coordinates (x, y) of the captures, and the columns to the number of captures made, resulting in a $[2 \times n]$ matrix, where n is the number of captures taken.
- *conv_to_robot ()*: This function is used to perform two actions: The first one is to convert the captured points to the work area established by the robot. In our case, this working area has been setup to DIN A3. The second one is to adapt the format of the data obtained to send them through the websocket protocol. To do this, this data must be converted to byte format, as required by the protocol.
- *conv_to_host ()*: This function, like the previous one, serves to modify the format of the data. In this case the data sent by the robot to the server.
- *plot_array ()*: Finally, the last executed function of the software, serves to depict in graph form the data matrix acquired previously. To do this, we set the x-axis of the graph to the x-components of the matrix and the same to the y-axis and the y-components of the matrix.

2.4 TMFlow

The software related to the operation of the cobot, programmed in TMFlow, can be summarized by the flow chart of Fig. 6.

The robot component of the designed control application starts from the resting point [PO], waits for the entry of the variable sent by the application, once obtained converts it into an array or array of data in string format.

Second, it checks if these variables are coordinates or [nan]. In the first case it continue with the software execution. In the second case, end the software returning to the resting point [PO].

Third, it extracts the component (x, y) from the previous variable, obtaining two independent variables, and then assigns these variables to the component (x, y) of the target point [PE], equivalent to the location captured by the eye-tracker but in the robot's work area.

The configuration of the points depends on the needs or desires of the application and the programmer. The components are, firstly, the coordinates (x, y, z) with respect to the base of the robot, and secondly, the rotation in (x, y, z) also with respect to the base. For example, for the developed application the point [PE] has been defined as follows:

$$[PE] = [var_Eye0, va_Eye1, z, Rx, Ry, Rz], \quad (1)$$

where var_Eye0 and var_Eye1 , are the variables which contain the (x, y) values acquired by the eye-tracker. Z is a fixed value, as the robot always draws on a 2D surface and Rx , Ry and Rz correspond to the robot axis rotation.

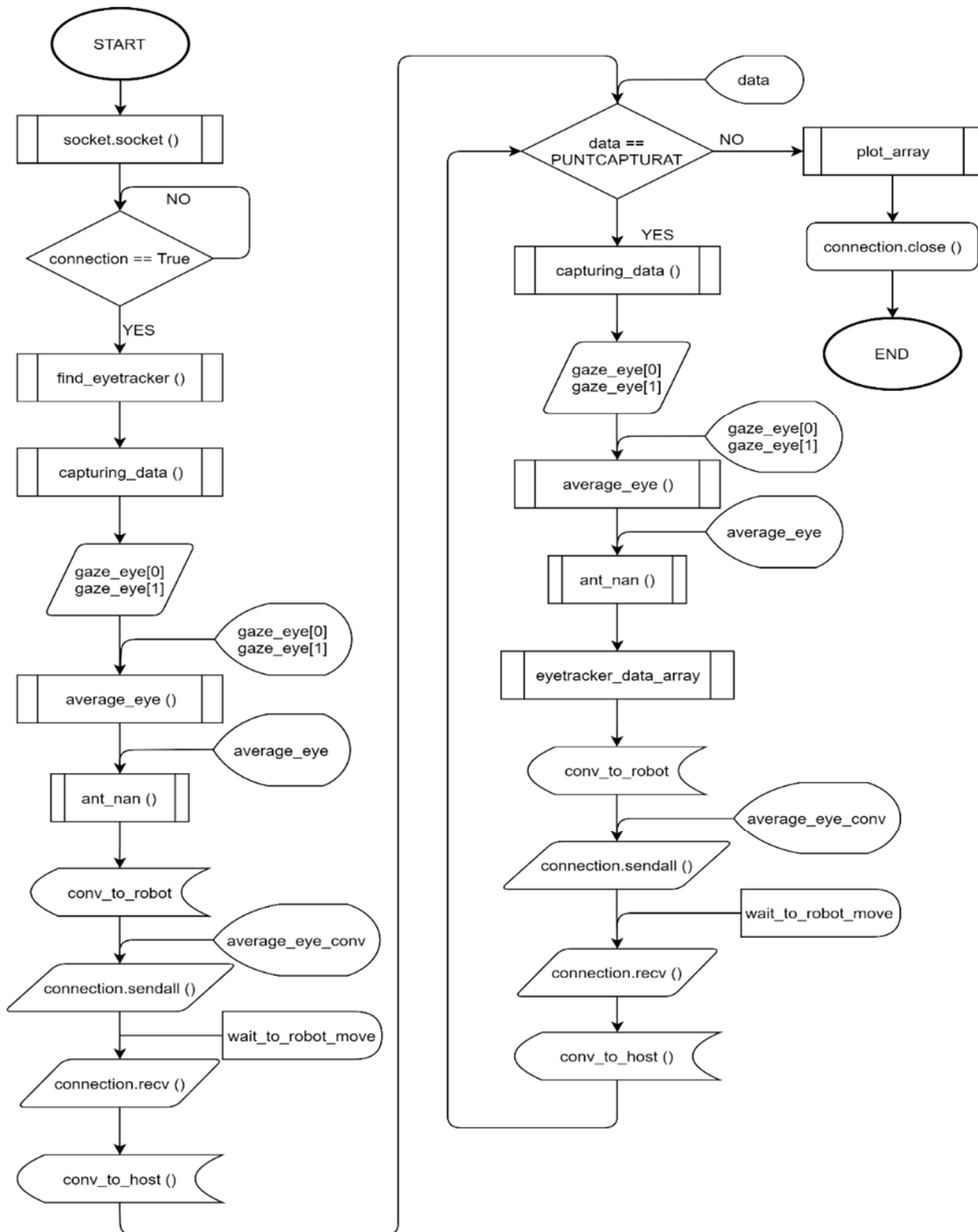


Figure 5: Eye-tracker flow chart.



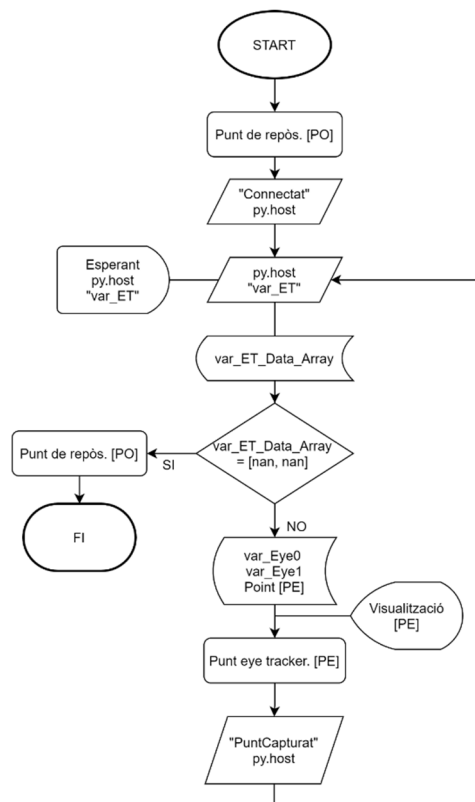


Figure 6: Cobot flow chart.

Finally, once it has reached the position determined by the [PE] point, a “CAPTURED POINT” message is sent to the server indicating that the robot has reached its goal and is waiting for another input. At this point, the data capture software comes back into action, and the robot waits.

3 EXPERIMENTS AND RESULTS

In order to check the system, we have established a set of simple exercises, consisting of drawing a simple set of figures (triangle, rectangle, pentagon and star. Fig. 7 represents these figures). The user must select first the desired image he wants to draw. Then the system shows the figure, the user looks at the corners of the figure (marked in red in Fig. 7), and the eye-tracker acquires the set of points. In this YouTube link we have included a demonstration of the system operation: <https://youtu.be/U6KYuhel9TK>.

Once the acquisition/performing system has been setup, it is quite straight forward to adapt the system to a more realistic application such as robot control by handicapped people, in dangerous environments (toxic, hazardous, explosive substances, etc.) This is beyond the goal of this paper, as we do not have access to these scenarios.

Fig. 8 shows the experimental results performed by the cobot on a DINA3 paper. On the left column we can observe the result of the drawing performed by the cobot. On the right it

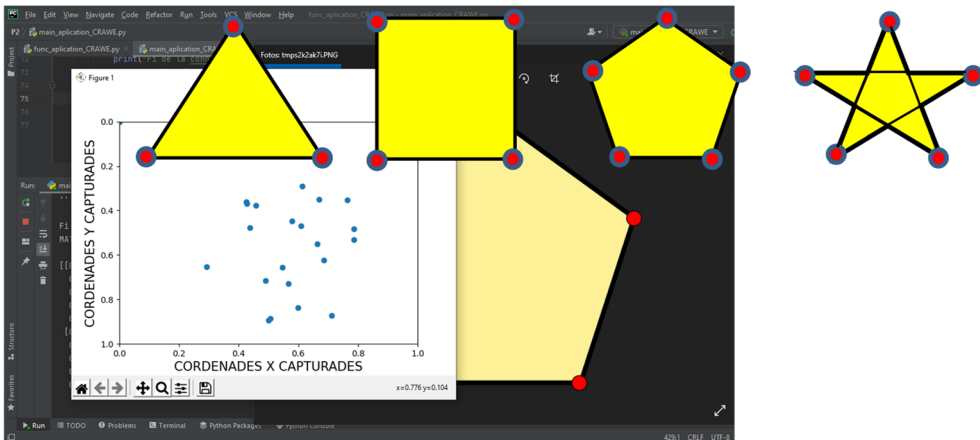


Figure 7: Top, from left to right: triangle, rectangle, pentagon and star figures presented on the screen of the eye-tracker. Bottom: user interface with several acquired dots.

is depicted the acquired points by the eye-tracker when the user looks to the corners of the desired image (from top to bottom: triangle, rectangle, pentagon and star inside a pentagon). In our software design the starting and ending point are marked separately by the user. For this reason, the figures look opened and not closed. In case of programming the application for performing a line connecting the last acquired point to the first one, the figure would be closed. Table 1 highlights the accuracy and precision of the system (accuracy refers to how close a measurement is to the true or accepted value. Precision refers to how close measurements of the same item are to each other).

Worth mentioning is that the user has performed the tasks without a lot of intensive training and it has been tested with several users. All of them agree that the system is easy-to-use.

4 CONCLUSIONS

In this paper we have presented an experimental setup to connect an eye-tracker Tobii PRO X2-30 with a collaborative robot (cobot) OMRON TM5-700. The most challenging part has been the communication between both elements. Once this has been solved, this opens the possibility to large amount of applications for robotic arm control by means of gaze.

The application has been designed with the software OMRON TMFlow for the robot side as well as the Tobii SDK for the eye-tracker. Although different robot and eye-tracker would require a new setup we think that this paper can be useful for the new system design involving different equipment. Eye-tracking technology can be very useful for a plenty of safety and security applications, including safety for construction workers, automotive and industry safe environments.



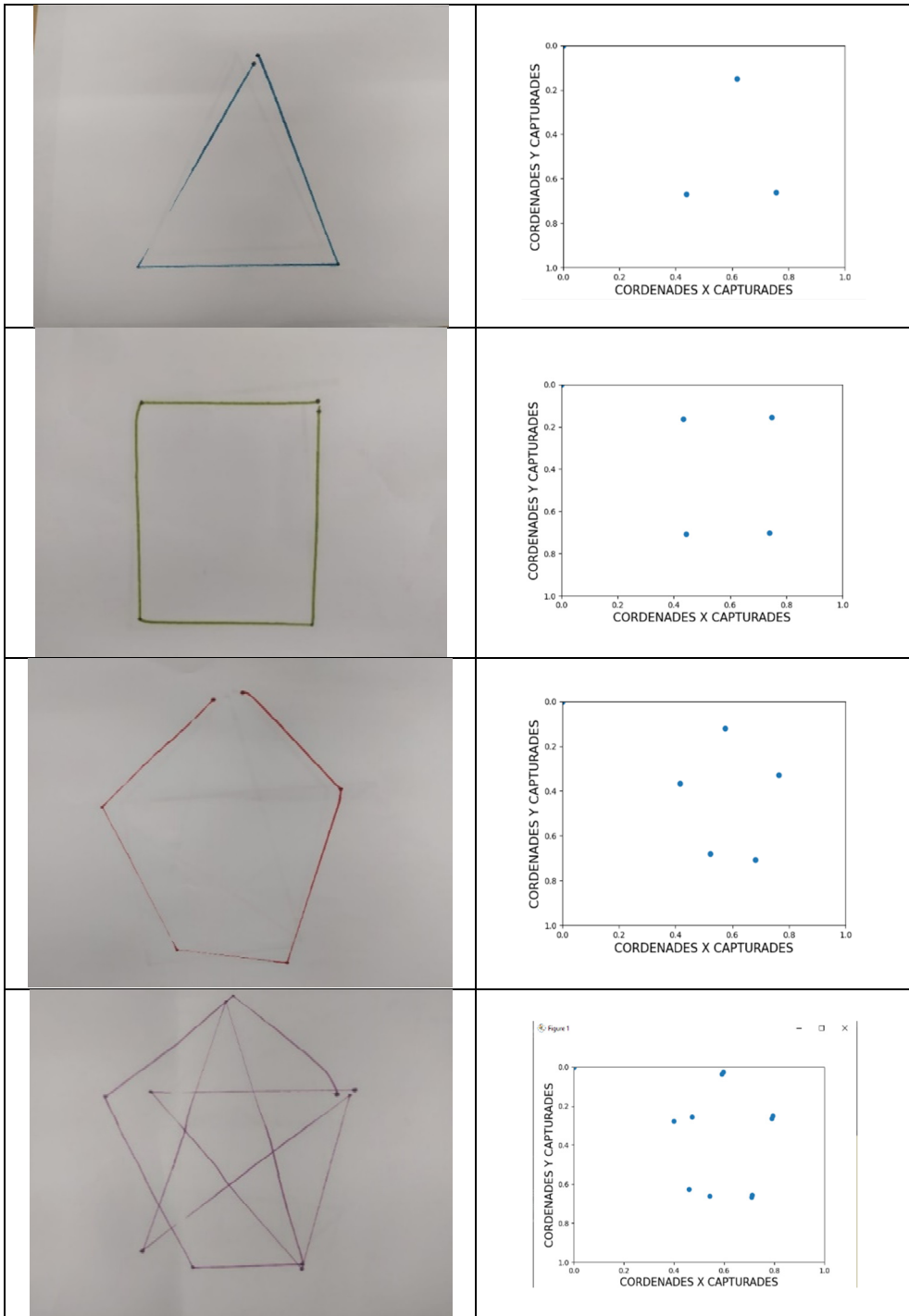


Figure 8: Triangle, rectangle, pentagon and star inside a pentagon. Drawing performed by the robot (left column) and points acquired by the eye-tracker (right column).

Table 1: Accuracy and precision of measurements. (SD = standard deviation; RMS = root mean square.)

Validation accuracy			Validation precision (SD)			Validation precision (RMS)		
degrees	pixels	mm	degrees	pixels	mm	degrees	pixels	mm
0.70°	22	5.5	0.22°	7	1.7mm	0.33°	10	2.6

ACKNOWLEDGEMENT

This work has been funded by Spanish grant Ministerio de ciencia e innovación PID2020-113242RB-I00.

REFERENCES

- [1] Markets and Markets, Eye tracking market with COVID-19 impact analysis by offering (hardware, software, services), tracking type (remote and mobile), application (assistive communication, and human behavior and market research), vertical, and geography: Global Forecast to 2025. <https://www.marketsandmarkets.com/Market-Reports/eye-tracking-market-144268378.html>. Accessed on: 21 Sep. 2021.
- [2] Jeelani, I., Han, K. & Albert, A., Automating and scaling personalized safety training using eye-tracking data. *Automation in Construction*, **93**, pp. 63–77, 2018. <https://doi.org/10.1016/j.autcon.2018.05.006>.
- [3] Hasanzadeh, S., Esmacili, B. & Dodd, M.D., Measuring the impacts of safety knowledge on construction workers' attentional allocation and hazard detection using remote eye-tracking technology. *Journal of Management in Engineering*, **33**(5), 2017.
- [4] Martinez-Marquez, D., Pingali, S., Panuwatwanich, K., Stewart, R.A. & Mohamed, S., Application of eye tracking technology in aviation, maritime, and construction industries: A systematic review. *Sensors*, **21**(13), p. 4289, 2021. <https://doi.org/10.3390/s21134289>.
- [5] Carr, D.B. & Grover, P., The role of eye tracking technology in assessing older driver safety. *Geriatrics*, **5**(2), pp.1–14, 2020. <https://doi.org/10.3390/geriatrics5020036>.
- [6] Vetturi, D., Tiboni, M., Maternini, G. & Bonera, M., Use of eye tracking device to evaluate the driver's behaviour and the infrastructures quality in relation to road safety. *Transportation Research Procedia*, **45**, pp. 587–595, 2020. <https://doi.org/10.1016/j.trpro.2020.03.053>.
- [7] Rjabovs, A. & Palacin, R., Investigation into effects of system design on metro drivers' safety-related performance: An eye-tracking study. *Urban Rail Transit*, **5**, pp. 267–277, 2019. <https://doi.org/10.1007/s40864-019-00115-1>.
- [8] Khan, M.Q. & Lee, S., Gaze and eye tracking: Techniques and applications in ADAS. *Sensors*, **19**(24), p. 5540, 2019. <https://doi.org/10.3390/s19245540>.
- [9] Le, A.S., Suzuki, T. & Aoki, H., Evaluating driver cognitive distraction by eye tracking: From simulator to driving. *Transportation Research Interdisciplinary Perspectives*, **4**, 100087, 2020. <https://doi.org/10.1016/j.trip.2019.100087>.
- [10] Gruden, C., Otković, I.I. & Šraml, M., Safety analysis of young pedestrian behavior at signalized intersections: An eye-tracking study. *Sustainability*, **13**(8), p. 4419, 2021. <https://doi.org/10.3390/su13084419>.
- [11] Babić, D., Babić, D., Fiolić, M. & Ferko, M., Factors affecting pedestrian conspicuity at night: Analysis based on driver eye tracking. *Safety Science*, **139**, 105257, 2021. <https://doi.org/10.1016/j.ssci.2021.105257>.



- [12] Peysakhovich, V., Lefrançois, O., Dehais, F. & Causse, M., The neuroergonomics of aircraft cockpits: The four stages of eye-tracking integration to enhance flight safety. *Safety*, **4**(1), p. 8, 2018. <https://doi.org/10.3390/safety4010008>.
- [13] Chadalavada, R.T., Andreasson, H., Schindler, M., Palm, R. & Lilienthal, A.J., Bi-directional navigation intent communication using spatial augmented reality and eye-tracking glasses for improved safety in human–robot interaction. *Robotics and Computer-Integrated Manufacturing*, **61**, 101830, 2020. <https://doi.org/10.1016/j.rcim.2019.101830>.
- [14] Dahmani, M., Chowdhury, M.E.H., Khandakar, A., Rahman, T., Al-Jayyousi, K., Hefny, A. & Kiranyaz, S., An intelligent and low-cost eye-tracking system for motorized wheelchair control. *Sensors*, **20**(14), p. 3936, 2020. <https://doi.org/10.3390/s20143936>.
- [15] Scalera, L., Seriani, S., Gallina, P., Lentini, M. & Gasparetto, A., Human–robot interaction through eye tracking for artistic drawing. *Robotics*, **10**, p. 54, 2021. <https://doi.org/10.3390/robotics10020054>.
- [16] Faundez-Zanuy, M. et al., Handwriting biometrics: Applications and future trends in e-security and e-health. *Cogn. Comput.*, **12**, pp. 940–953, 2020. <https://doi.org/10.1007/s12559-020-09755-z>.
- [17] Faundez-Zanuy, M., Mekyska J. & Impedovo, D., Online handwriting, signature and touch dynamics: Tasks and potential applications in the field of security and health. *Cogn. Comput.*, 2021. <https://doi.org/10.1007/s12559-021-09938-2>.

