Performance evaluation of a robotic architecture for drawing with eyes

ISSN 1590-8844
International Journal of Mechanics and Control, Vol. XX, No. XX, 20XX

PERFORMANCE EVALUATION OF A ROBOTIC ARCHITECTURE FOR DRAWING WITH EYES

Lorenzo Scalera[*] Eleonora Maset[**] Stefano Seriani[***]
Alessandro Gasparetto[*] Paolo Gallina[***]

[*] Polytechnic Department of Engineering and Architecture (DPIA), University of Udine, Udine, Italy

[**] Department of Agricultural, Food, Environmental and Animal Sciences (DI4A), University of Udine, Udine, Italy

[***] Department of Engineering and Architecture (DIA), University of Trieste, Trieste, Italy

ABSTRACT

Eye tracking is a sensing technology that allows a computer to monitor eye movements and determine where a subject is looking. In this paper, we evaluate the performance of a robotic architecture that enables to control a robot arm through eye tracking and to draw using the motion of the eyes only. The usability of the system is assessed by a drawing experiment where 10 naïve subjects learned to operate the robot manipulator with eyes. Results suggest that the gaze-based human-robot interface may be considered an intuitive and efficient technology to perform a drawing task, and could be beneficial beyond amputees and patients with various forms of movement impairments.

Keywords: robotics; eye tracking; human-robot interaction; collaborative robotics.

1 INTRODUCTION

Nowadays, robotic systems are often adopted by artists as a technology to imitate human painting and to develop novel forms of expression. In recent years, several examples of machines and robots equipped for painting purposes have been developed and described in the literature. In most cases, the motion of the robotic system is planned on the basis of image processing algorithms that aim at reproducing a non-photorealistic rendering of an input image [1–3].

Notable examples can be found in the works of Tresset and Leymarie [4], where a robotic installation developed to draw sketches of people by extracting salience lines from images is presented. Moreover, the painting robot developed by Lindermeier et al. [5] is capable of creating brush strokes based on non-photorealistic techniques and visual feedback. Further examples are given by Song et al. [6], who devised an artistic pen drawing system for arbitrary surfaces, and by Karimov et al. [7], who implemented a Cartesian robot capable of creating full-color images with a human-like kinematics. Other recent examples of artistic robots include the interactive painting system shown in [8], the airbrush robotic architecture presented in [9], the adoption of mobile robots for artistic painting as in [10, 11], and the automation of the palette knife painting technique described in [12].

In the majority of these cases, the interaction between the robotic system and the human artist is limited to the choice of software and hardware parameters, and the painting process is mainly handled by the algorithm and based on a starting input image. Only few examples of robotic painting systems remotely controlled by humans can be found in the literature. These are mainly related to robotic telemanipulation, such as in [13], where a flexible force-vision-based interface allows operators to make a remote robot draw. Furthermore, in [14], a human-machine interface is developed based on a brain-computer interface and a robotic architecture for neurorobotics painting. The system measures the brain activity of the user and associates the recorded cerebral signals into simplified movements of the manipulator. More

---

Contact author: Lorenzo Scalera.
[*] University of Udine, via delle Scienze 206, 33100 Udine, Italy, {lorenzo.scalera, alessandro.gasparetto}@uniud.it
[**] University of Udine, via delle Scienze 206, 33100 Udine, Italy, eleonora.maset@uniud.it
[***] University of Trieste, via A. Valerio 6/1, 34127 Trieste, Italy, {sseriani,pgallina}@units.it

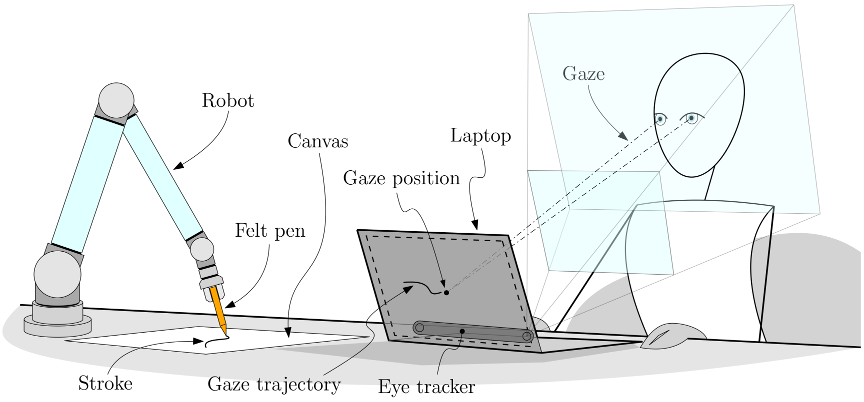


Figure 1 Scheme of the robotic architecture for drawing with eyes.

recently, in [15] the authors proposed a telerobotic system that uses a digital tablet and a motion capture suit as input devices to perform accurate drawings via a high bandwidth wireless internet access.

Within this framework, eye tracking is a promising candidate technology for robotic teleoperation. Eye tracking allows a computer to know where a person is looking and represents a simple and effective interaction method for people with minimal motor abilities [16]. In [17], an intuitive eye tracking-controlled robot operates in 3D space and enables primitive tele-writing and drawing tasks. In [18], eye tracking shows better performance with respect than head tracker in a human-robot collaboration scenario, whereas in [19] an eye tracker device is adopted for the teleoperation of a mobile robot. The authors in [20] adopted a binocular eye tracker for the 3D continuous control of a robotic support arm system. Furthermore, in [21], an eye-tracking device is used for the reliable control of robot trajectories, together with a brain-computer interface used to change robot Cartesian stiffness. Recently, in [22], the authors proposed a lightweight, head-worn interface for robust and real-time robot control using head and eye gaze information.

In the field of art, some attempts to draw using eye trackers have been performed, starting the so called *eye gaze art*, as described in [23]. In this context, a pioneeristic work is the Eagle Eyes, an eye control system that allows persons with disabilities to draw on a screen [24]. Moreover, in [25], a gaze and voice controlled control system is described, whereas in [26], a drawing tool controlled by the gaze and based on movable shapes is shown. In contemporary art, the eye-tracking technology is used by the artist Graham Fink to create figurative portraits on a screen [27]. In the same context, a preliminary robotic system capable of drawing with eyes is described in [28, 29]. The robotic system was first presented at Trieste Next (Italy), Festival of Scientific Research, in 2019. During the festival, the system was tested by the public with great interest. The proposed system could be exploited as a tool for drawing and painting by amputees or people with movement impairments. However, a quantitative performance evaluation of that robotic architecture have not been carried out yet.

In this paper, we present the analysis and performance evaluation of an eye-tracker based architecture that allows a user to paint with eyes. The setup exploits an eye-tracker device as a human-robot interface to steer a remote industrial manipulator during a drawing task (Figure 1). The paper is an extended version of the preliminary conference work published in the 3rd International Conference of IFToMM Italy (IFIT 2020) [28]. In particular, with respect to [28], (a) we further analyse the eye tracking system and the software architecture; (b) we investigate the performance of the system with a drawing experiment on 10 naïve subjects, who learned to operate the robot manipulator with eyes; and (c) we quantitatively evaluate the results using different performance metrics. The evaluation of the eye tracking system for drawing with eyes is preparatory for a deeper performance assessment of the whole robotic architecture.

The remaining of the paper is organized as follows: Section 2 briefly recalls the eye-tracking technology, Section 3 describes the robotic system, whereas in Section 4 the process of drawing with eyes is presented. The experimental tests and the results are illustrated in Section 5, and the conclusions of the paper are given in Section 6.

2 EYE TRACKING

Eye tracking is a sensing technology that allows a computer or another device to record the movements of the eyes and to know where a person is looking (point of gaze). This technology permits an easy and natural interface between a human and an external device, and finds application in several fields. In psychology, for example, the study of eye movements provides insights in cognitive processes, attention, behavior and decision-making [30]. In the field of computer science, eye tracking can facilitate the learning of software, as well as enhance the visual experience in video-games and augmented graphical displays [31]. Furthermore,

Figure 2 Overview of the experimental setup.

eye tracking is applied in marketing to detect where the consumer's gaze points are focused on products of interest [32]. However, the most important application of eye tracking is to provide communication and interaction for patients with various forms of degenerative neuromuscolar or neurological diseases [33]. Indeed, eye movements are preserved in many movement disorders leading to paralysis from stroke, spinal cord injury, Parkinson's disease, multiple sclerosis, and muscular dystrophy among others [20].

The physiology of eye movements has been the subject of numerous studies, as testified by a flourishing literature on the topic, e.g., in [34]. From the literature, we learn that, when we look at a static image with stationary head, our visual perception is guided by alternating two main types of eye movements: *saccades* and *fixations* [35]. Saccades are discrete and rapid eye movements that are directed from one point of interest of an image to another in a goal-oriented fashion. Saccadic eye movements have an average duration of 20-30 ms and can be triggered voluntarily or involuntarily. Conversely, fixations eye movements allows our eyes to pause over informative regions of interest. During this time interval, which lasts 50-600 ms, novel information are acquired by the visual cortex. Indeed, the majority of visual processing on static images takes place during fixations. When we are moving or our eyes focus on dynamic targets, other types of eye movements arise to align the fovea with the visual point of interest. These are vergence eye movements, smooth pursuit, and the vestibular ocular reflex. However, saccadic eye movements and fixations play the largest role in gathering information from static images. These are the eye movements considered in this work, since in our experimental tests participants are looking at static images on flat monitors with stationary head.

3 SYSTEM OVERVIEW

The eye tracker adopted in this work is a low-cost Tobii Eye Tracker 4C, which can be easily mounted on a laptop or a monitor. The eye tracker works with near infrared (NIR 850 nm) illuminators that create a pattern of infrared light on the eyes, as shown in Figure 1. The cameras of the eye tracker take high resolution images of the user's eyes and of the projected pattern, whereas machine learning and image processing algorithms estimate the eyes' positions in space and the point of gaze on the screen. For the tests we adopted a monitor with a resolution of 1920×1080 pixel, and a pixel size of $0.3\ mm$. Prior to starting the eyes-drawing process, the eye tracker is calibrated by looking at a sequence of points displayed on the screen one after the other, using the Tobii Pro Eye Tracker Manager software. The calibration process measures the characteristics of the subject's eyes and calculates personal differences such as where the pupils are located in relation to the cornea.

As far as the authors' knowledge, the performance of the Tobii 4C eye tracker has not been analyzed and published in the literature yet. However, the performance of the Tobii EyeX, which is similar to the 4C, has been evaluated in [36]. In that paper, the sampling rate of the device is found to be slightly lower than the nominal frequency reported in the technical sheet (55 Hz vs. 60 Hz). Furthermore, the performances of the device are function of the gaze angle away from straight ahead, central fixation. For the model Tobii EyeX, accuracy and precision assume values of $< 0.4°$ and $< 0.2°$, near the center of the monitor. These values decrease to $< 0.6°$ and $< 0.25°$, respectively, at more than 5 degrees away from the center of the monitor [36]. However, in this work we do not only investigate the performance of the eye-tracker device itself, but also the usability and intuitiveness of the whole system for drawing with eyes, as well as the capabilities of inexpert subjects.

The robotic manipulator chosen in this work is a UR5 industrial robot by Universal Robots with 6 degrees of freedom. The robot is available at the Mechatronics and Robotics Lab at University of Udine, in Italy. The manipulator can hold a maximum payload of 5 Kg and has a working radius of 850 mm. The safety features of the robot allows us to use it alongside an operator without the need of protective barriers as outlined by collaborative robotics [37, 38]. The robot is equipped for painting with an aluminum support for pens or brushes, as illustrated in Fig. 2. The robotic system is controlled in Matlab® environment. The commands for the robot are written in the UR Script Programming Language and sent to the robot controller using the TCP/IP protocol. The trajectories for the manipulator are planned in the Cartesian space using standard trapezoidal motion laws and imposing the values of acceleration $a = 0.3\ m/s^2$, cruise speed $v = 0.3\ m/s$, and blending radius $r = 0.001\ m$.

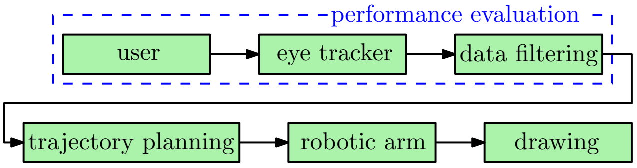


Figure 3 Flowchart of the process for drawing with eyes.

The robot performs the painting on a planar surface, which position and orientation are calibrated with respect to the robot base prior to starting the drawing task. In particular, by acquiring at least three positions of the end-effector with a pen touching the painting surface, the minimum-square error approximating plane can be computed. Robot state data, such as joint and end-effector kinematic variables, are indeed continuously sent from the robot controller to the computer via Ethernet at a rate of 125 Hz.

4 DRAWING WITH EYES

The process of painting with eyes is based on the observation of a test image shown on a screen, whose main features have to be followed with the eyes' motion. In this way, unskilled subjects are facilitated in the drawing task. However, the system can be used without a test image as well, by drawing with eyes directly on a white screen.

After the calibration of the eye tracker, the drawing operation is started by running a custom Matlab® software that displays an image on the screen and allows, at the same time, to log the gaze positions registered by the eye tracker. The gaze data is not acquired continuously, but only when the user keeps a predefined key on the keyboard pressed. In this manner, we avoid recording unwanted saccadic eye movements. In future developments of the work, which could adapt the task to tetraplegic users as well, we plan to introduce additional technologies for this purpose, as for instance a head-tracker device, or a voice recognition system.

Before sending the data to the robot, each set of raw data points needs to be filtered, in order to remove the contributions given by the fixations. Indeed, fixations do not add any artistic contribution to the final artwork and could create problems in the robot motion due to the multitude of points very close to each other. To remove fixations, data are filtered with a moving average filter (with a windows size set to $\omega = 5$ elements), followed by a filter based on the distance between subsequent points. In particular, the algorithm computes the distance between each couple of consequent points of the acquired strokes. If this distance is less than a threshold value, the second point of the couple is removed. The threshold distance was set equal to $d = 0.016$ by analysing the results of pilot tests, where the vertical coordinate of raw data was scaled between 0 and 1. In this manner, the total number of points belonging to each fixation is considerably reduced, and the whole stroke is smoothed.

Finally, the gaze coordinates are scaled within the painting surface limits and sent to the robot controller as command strings. The acquisition of gaze positions can continue in parallel with the painting process, and new strokes are painted as soon as the robot completes the previous ones. The time needed for painting depends on the number of strokes and their length, as well as by the robot velocity and acceleration. The whole process for drawing with eyes is summarized in Figure 3.

5 EXPERIMENTAL RESULTS

The experimental tests presented in this paper were performed with multiple users, and we provided objective quality metrics to investigate how the system works and how easily it can be learned. In particular, participants were shown a set of geometrical shapes and silhouettes, and were asked to recreate these using the eye tracker system. Then, the generated drawings have been quantitative compared to the original silhouettes both before and after applying the fixations filter.

In this study, the participants were recruited among the population of students and professors of University of Udine and University of Trieste. A total of 10 people (4 females) participated, ranging in age from 24 to 62 years with mean of 36.1 and standard deviation of 12.2 years. Only 3 participants had previous experience with eye tracker devices. None of the subjects involved in the experiment had a mobility disorder or a motion impairment, one of them was wearing glasses.

After the eye tracker calibration, all participants were shown four geometrical shapes in sequence (a line, a triangle, a square, and a circle), and they were instructed to follow the contours of the shapes with their gaze. The size of the input shapes was chosen to be half of the screen height, and they were displayed at the center of the monitor, where the accuracy of the eye-tracker device is the highest [36]. Each test with a sequence of four shapes was repeated for three times, and the calibration of the eye tracker was performed at the beginning of each of the three trials. Figure 4 reports the reference test shapes and few samples of raw and filtered images generated by the users. After the tests with the geometrical shapes, participants were asked to reproduce a more complex silhouette, i.e., the contour of a flower (Figure 7).

Before the start of the experiment, subjects were informed about the tests and they were asked for some personal information for statistical purposes (age, gender, whether mobility disorders were present and glasses worn). Then, they were instructed to sit in front of the experimental set-up with the head still, and to press the proper key to draw with eyes, followed by another one to advance to the next shape. No time limit was imposed on the participants and they were not required to perform the task as quickly as possible.

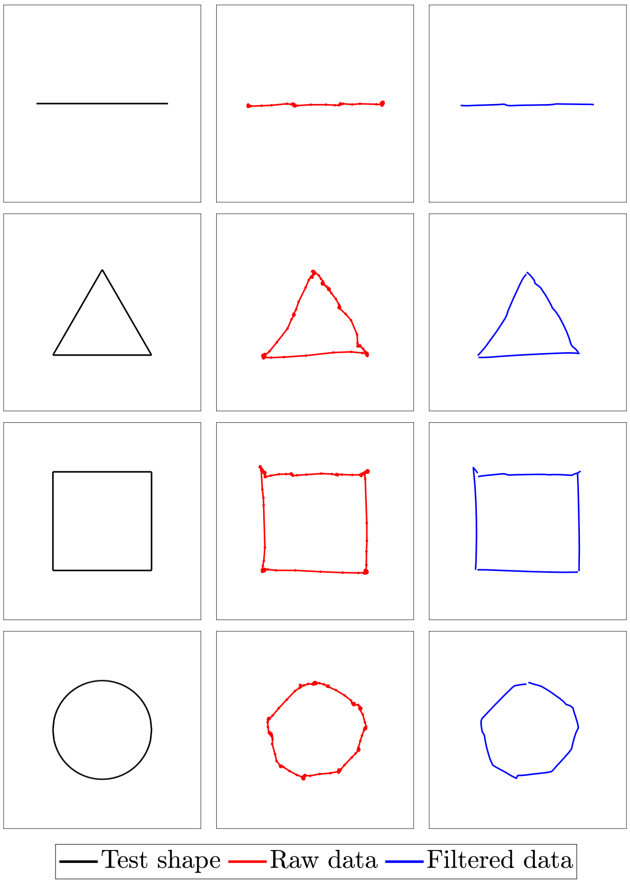


Figure 4 Reference test shapes and samples of raw and filtered user generated images.

Two quantitative metrics have been applied for the data analysis of the geometrical shapes: the distance error between each gaze point and the closest reference segment, and the translation error between each generated image and the ground-truth shape. The distance error index was implemented by following the efficient algorithm illustrated in [39]. This method was applied to compare the eye tracker data with the original silhouettes by computing the distance between each gaze point and the closest line segment of the geometrical shape, then averaging all the single values on each shape. The circle was approximated to a regular convex polygon with 60 sides.

To compute the translation error, we applied the Iterative Closest Point algorithm (ICP) [40] to estimate the rigid transformation needed to reach the best alignment between the reference image and the corresponding user generated eye tracker data. The retrieved translation values were assumed as horizontal and vertical errors. Since the estimated rotation was negligible in most cases, it was not considered for the performance evaluation.

Figures 5 and 6 report the box plot representation of the experimental results with the geometrical shapes. In the box plots, the central mark indicates the median, the bottom and top of each box represent the first and third quartiles, the whiskers extend to the most extreme data points not considered outliers. Figure 5 shows the distance error between reference and user generated data for the three trials of the test. It can be seen that from the first trial, users are able to reproduce the proposed images with low values of error (on average $< 2\%$ with respect to the monitor height), but no significant performance improvement can be appreciated along the three trials. Figure 6 reports the box plots of the total drawing time needed to reproduce the geometrical shapes in the trials. Also in this case, the total drawing time does not change considerably over the tests.

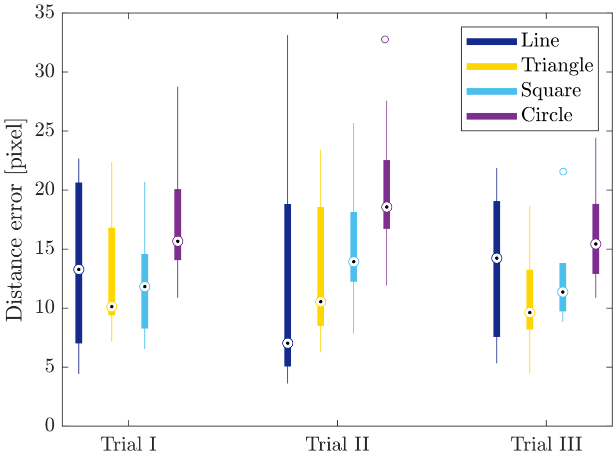


Figure 5 Box plot representation of the experimental results showing the distance error between reference and user generated images in the three trials of the tests.

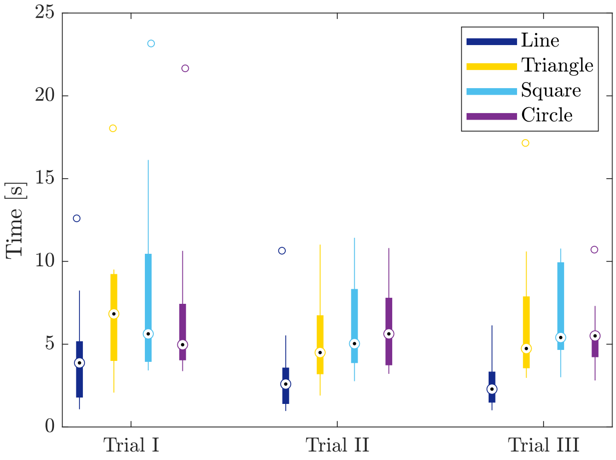


Figure 6 Box plot representation of the experimental results showing the total drawing time in the three trials of the tests.

Furthermore, in Table I the values of distance error, and of the translation error along the horizontal and vertical axis are described. These values have been computed using filtered data and averaged across participants and trials. From the

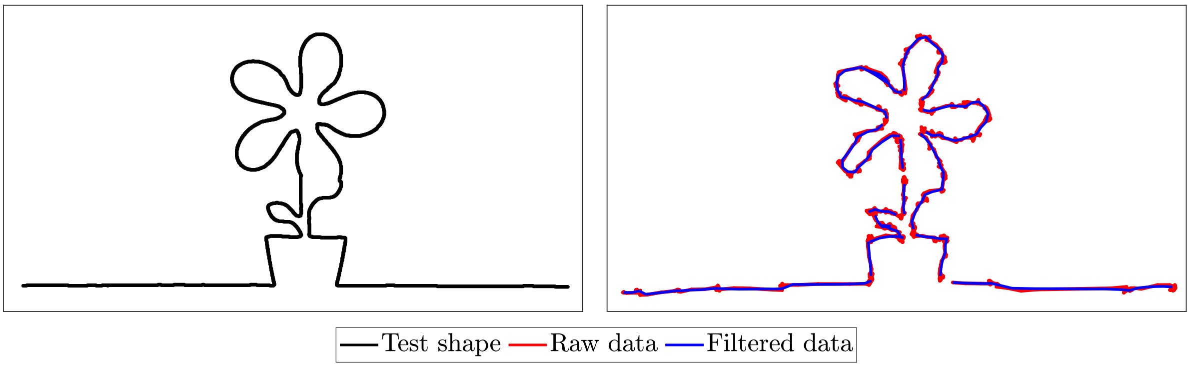
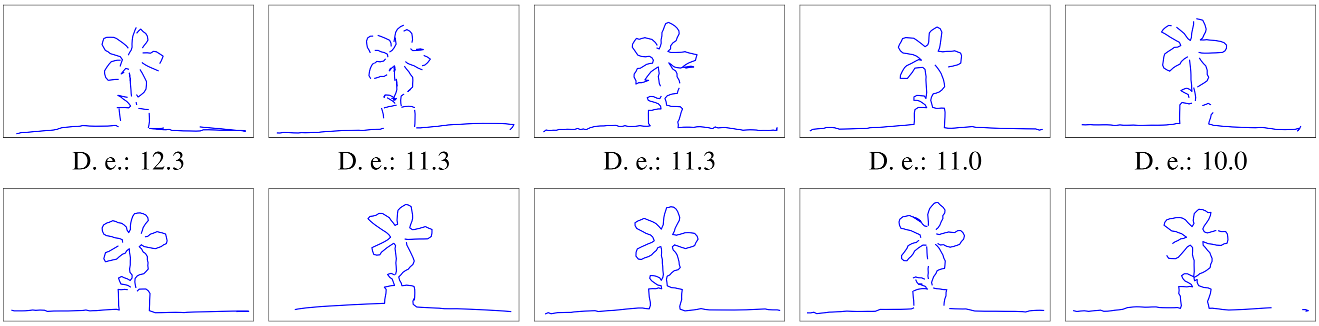


Figure 7 Reference test shape and a sample of raw and filtered user generated image.


D. e.: 12.3 D. e.: 11.3 D. e.: 11.3 D. e.: 11.0 D. e.: 10.0

D. e.: 8.4 D. e.: 7.6 D. e.: 6.6 D. e.: 5.9 D. e.: 5.7

Figure 8 Flower images generated by the 10 subjects that participated to the experimental tests, together with the values of distance error (D. e., in pixel) with respect to the nominal image.

Table I - Mean and standard deviation of the distance error and of the translation error along the horizontal and vertical axis (trasl. x and y). Values are reported in pixel.

	Dist. error	**Trasl. x**	**Trasl. y**
Line	13.2 ± 7.8	2.1 ± 12.3	-14.3 ± 21.7
Triangle	11.9 ± 5.2	2.2 ± 16.9	-4.4 ± 23.4
Square	14.0 ± 6.3	5.6 ± 17.7	-15.9 ± 34.8
Circle	18.0 ± 5.8	2.3 ± 8.1	-9.9 ± 20.4

table it can be noticed that a shift along the vertical axis is often needed to match the eye tracker data with the reference images. This can be due to a non-optimal calibration of the eye tracker at the beginning of the trials or due to involuntary motion of the head during the tests.

Table II shows the values of the nominal path, raw and filtered gaze path, drawing time and average gaze velocity across participants and trials for the four test images. The raw and filtered gaze path values are computed by adding the distance between each pair of consecutive points in a drawn stroke, whereas the drawing time is acquired as the time during which the drawing key is pressed by the subject performing the test. Finally, the average gaze velocity is computed for each of the four test images as the ratio between filtered gaze path and drawing time. From the values in the table it can be noticed that after the filtering the length of the path decreases due to the removal of fixations, approaching the nominal value.

Figure 7 reports the reference image of the flower along with an example of raw and filtered user generated image. Finally, in Figure 8 the flower images generated by the 10 subjects involved in the experimental tests are reported together with the mean values of distance error between nominal and generated image. In this case, the error was computed as the distance between each gaze point and the closest pixel of the reference image, averaged on all the filtered data. The images in Figure 8 show that all the 10 naïve participants were able to draw a flower with eyes after the proposed experience with eye tracking. Furthermore, the reference images are drawn on paper by the robot, showing that the employed gaze-based human-robot interface can be considered an intuitive and efficient tool to perform a drawing task using the eyes.

6 CONCLUSIONS

In this paper, the performance of an eye-tracker based architecture that enables to control a robot arm and to draw using the motion of the eyes only have been evaluated. The usability of the system has been assessed by a drawing experiment where 10 naïve participants learned to operate

Table II - Values of nominal path, raw and filtered gaze path, drawing time and average gaze velocity (mean ± st. dev.) across participants and trials for the four test images.

	Nominal path [pixel]	**Raw gaze path** [pixel]	**Filtered gaze path** [pixel]	**Drawing time** [s]	**Average gaze velocity** [pixel/s]
Line	720	1463 ± 818	851 ± 425	3.5 ± 2.8	524 ± 201
Triangle	1620	2844 ± 902	1655 ± 204	6.4 ± 4.1	546 ± 205
Square	2160	3643 ± 1502	2265 ± 605	7.9 ± 6.2	568 ± 181
Circle	1697	2894 ± 885	1791 ± 427	6.3 ± 3.7	517 ± 132

the robot manipulator with eyes. The subjects involved in the tests have been asked to reproduce with the eyes a series of reference images and silhouettes shown on a screen. Results have been evaluated using different performance metrics, such as error between nominal and generated images, translation error, total gaze path and drawing time. Results suggest that the gaze-based human-robot interface can be learned quickly and intuitively. Therefore, it could represent an efficient tool to perform a drawing task, and can be considered beneficial beyond amputees and patients with various forms of movement impairments.

We plan to further investigate the application of eye tracking technology to robotic painting and drawing. In particular, future developments of this work will include novel strategies for the removal of the fixations using clustering techniques, as well as the comparison of different filtering methods. Finally, we plan to apply artificial intelligence techniques for the elaboration of gaze paths to be executed by the robotic manipulator.

ACKNOWLEDGEMENTS

The authors would like to thank the volunteer students and professors that participated to the experimental tests.

REFERENCES

[1] Jörg Marvin Gülzow, Patrick Paetzold, and Oliver Deussen. Recent developments regarding painting robots for research in automatic painting, artificial creativity, and machine learning. *Applied Sciences*, 10(10):3396, 2020.

[2] Lorenzo Scalera, Stefano Seriani, Alessandro Gasparetto, and Paolo Gallina. Non-photorealistic rendering techniques for artistic robotic painting. *Robotics*, 8(1):10, 2019.

[3] Artur Karimov, Ekaterina Kopets, Georgii Kolev, Sergey Leonov, Lorenzo Scalera, and Denis Butusov. Image preprocessing for artistic robotic painting. *Inventions*, 6(1):19, 2021.

[4] Patrick Tresset and Frederic Fol Leymarie. Portrait drawing by Paul the robot. *Computers & Graphics*, 37(5):348–363, 2013.

[5] Thomas Lindemeier, Jens Metzner, Lena Pollak, and Oliver Deussen. Hardware-based non-photorealistic rendering using a painting robot. In *Computer Graphics Forum*, volume 34, pages 311–323. Wiley Online Library, 2015.

[6] Daeun Song, Taekhee Lee, Young J Kim, Sungmin Sohn, and Young J Kim. Artistic pen drawing on an arbitrary surface using an impedance-controlled robot. In *IEEE Int. Conf. on Rob. and Autom. (ICRA)*, 2018.

[7] Artur I Karimov, Ekaterina E Kopets, Vyacheslav G Rybin, Sergey V Leonov, Anzhelika I Voroshilova, and Denis N Butusov. Advanced tone rendition technique for a painting robot. *Robotics and Autonomous Systems*, 115:17–27, 2019.

[8] Otoniel Igno-Rosario, Claudia Hernandez-Aguilar, Alfredo Cruz-Orea, and Arturo Dominguez-Pacheco. Interactive system for painting artworks by regions using a robot. *Robotics and Autonomous Systems*, 121:103263, 2019.

[9] Lorenzo Scalera, Enrico Mazzon, Paolo Gallina, and Alessandro Gasparetto. Airbrush robotic painting system: Experimental validation of a colour spray model. In *Int. Conf. on Robotics in Alpe-Adria Danube Region*, pages 549–556. Springer, 2017.

[10] Anurag Sai Vempati, Harshit Khurana, Vojtech Kabelka, Simon Flueckiger, Roland Siegwart, and Paul Beardsley. A Virtual Reality Interface for an Autonomous Spray Painting UAV. *IEEE Robotics and Automation Letters*, 4(3):2870–2877, 2019.

[11] María Santos, Gennaro Notomista, Siddharth Mayya, and Magnus Egerstedt. Interactive multi-robot painting through colored motion trails. *Frontiers in Robotics and AI*, 7:143, 2020.

[12] Andrea Beltramello, Lorenzo Scalera, Stefano Seriani, and Paolo Gallina. Artistic robotic painting using the palette knife technique. *Robotics*, 9(1):15, 2020.

[13] Camilo Perez Quintero, Masood Dehghan, Oscar Ramirez, Marcelo H Ang, and Martin Jagersand. Flexible virtual fixture interface for path specification in tele-manipulation. In *2017 IEEE Int. Conf. on Rob. and Autom. (ICRA)*, pages 5363–5368. IEEE, 2017.

[14] Salvatore Tramonte, Rosario Sorbello, Christopher Guger, and Antonio Chella. Acceptability Study of A3-K3 Robotic Architecture for a Neurorobotics Painting. *Frontiers in Neurorobotics*, 12:81, 2019.

[15] Lingyun Chen, Abdalla Swikir, and Sami Haddadin. Drawing Elon Musk: A Robot Avatar for Remote Manipulation. In *2021 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*. IEEE, 2021.

[16] Päivi Majaranta and Andreas Bulling. Eye tracking and eye-based human–computer interaction. In *Advances in physiological computing*, pages 39–65. Springer, 2014.

[17] Sabine Dziemian, William W Abbott, and A Aldo Faisal. Gaze-based teleprosthetic enables intuitive continuous control of complex robot arm use: Writing & drawing. In *6th Int. Conf. on Biomed. Rob. and Biomech.*, pages 1277–1282. IEEE, 2016.

[18] Oskar Palinko, Francesco Rea, Giulio Sandini, and Alessandra Sciutti. Robot reading human gaze: Why eye tracking is better than head tracking for human-robot collaboration. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5048–5054. IEEE, 2016.

[19] Daniel Gêgo, Carlos Carreto, and Luis Figueiredo. Teleoperation of a mobile robot based on eye-gaze tracking. In *12th Ib. Conf. on Inform. Syst. and Tech. (CISTI)*, pages 1–6. IEEE, 2017.

[20] Roni O Maimon-Mor, Jorge Fernandez-Quesada, Giuseppe A Zito, Charalambos Konnaris, Sabine Dziemian, and A Aldo Faisal. Towards free 3d end-point control for robotic-assisted human reaching using binocular eye tracking. In *2017 Int. Conf. on Rehab. Robotics (ICORR)*, pages 1049–1054. IEEE, 2017.

[21] Lucia Schiatti, Jacopo Tessadori, Giacinto Barresi, Leonardo S Mattos, and Arash Ajoudani. Soft brain-machine interfaces for assistive robotics: A novel control approach. In *2017 Int. Conf. on Rehab. Robotics*, pages 863–869. IEEE, 2017.

[22] Lukas Wöhle and Marion Gebhard. Towards robust robot control in cartesian space using an infrastructureless head-and eye-gaze interface. *Sensors*, 21(5):1798, 2021.

[23] Chris Creed. Assistive technology for disabled visual artists: exploring the impact of digital technologies on artistic practice. *Disability & Society*, 33(7):1103–1119, 2018.

[24] James Gips and Peter Olivieri. Eagleeyes: An eye control system for persons with disabilities. In *The eleventh international conference on technology and persons with disabilities*, pages 1–15, 1996.

[25] Jan Van der Kamp and Veronica Sundstedt. Gaze and voice controlled drawing. In *Proc. of the 1st Conf. on novel gaze-controlled applications*, pages 1–8, 2011.

[26] Henna Heikkilä. Tools for a gaze-controlled drawing application–comparing gaze gestures against dwell buttons. In *IFIP Conference on Human-Computer Interaction*, pages 187–201. Springer, 2013.

[27] Graham Fink. Eye drawings. `https://grahamfink.com/eye-drawings`, 2020.

[28] Lorenzo Scalera, Stefano Seriani, Alessandro Gasparetto, and Paolo Gallina. A novel robotic system for painting with eyes. In *The Third Int. Conf. of IFToMM ITALY*, pages 191–199. Springer, 2020.

[29] Lorenzo Scalera, Stefano Seriani, Paolo Gallina, Mattia Lentini, and Alessandro Gasparetto. Human–robot interaction through eye tracking for artistic drawing. *Robotics*, 10(2):54, 2021.

[30] Jacob L Orquin and Kenneth Holmqvist. Threats to the validity of eye-movement research in psychology. *Behavior research methods*, 50(4):1645–1656, 2018.

[31] Peter M Corcoran, Florin Nanu, Stefan Petrescu, and Petronel Bigioi. Real-time eye gaze tracking for gaming design and consumer electronics systems. *IEEE Trans. on Cons. Electr.*, 58(2):347–355, 2012.

[32] Florina Ungureanu, Robert Gabriel Lupu, Adrian Cadar, and Adrian Prodan. Neuromarketing and visual attention study using eye tracking techniques. In *2017 21st international conference on system theory, control and computing (ICSTCC)*, pages 553–557. IEEE, 2017.

[33] Emanuele Pasqualotto, Tamara Matuz, Stefano Federici, Carolin A Ruf, Mathias Bartl, Marta Olivetti Belardinelli, Niels Birbaumer, and Sebastian Halder. Usability and workload of access technology for people with severe motor impairment: a comparison of brain-computer interfacing and eye tracking. *Neurorehab. and neural repair*, 29(10):950–957, 2015.

[34] Keith Rayner. The 35th sir frederick bartlett lecture: Eye movements and attention in reading, scene perception, and visual search. *Quarterly journal of experimental psychology*, 62(8):1457–1506, 2009.

[35] Dario D Salvucci and Joseph H Goldberg. Identifying fixations and saccades in eye-tracking protocols. In *Proc. of the 2000 Symp. on Eye tracking research & appl.*, pages 71–78, 2000.

[36] Agostino Gibaldi, Mauricio Vanegas, Peter J Bex, and Guido Maiello. Evaluation of the Tobii EyeX Eye tracking controller and Matlab toolkit for research. *Behavior research methods*, 49(3):923–946, 2017.

[37] M Wu, Y He, and S Liu. Adaptive impedance control based on reinforcement learning in a human-robot collaboration task with human reference estimation. *Int. Journal of Mechanics and Control*, 21:21–32, 2020.

[38] L Scalera, A Giusti, R Vidoni, V Di Cosmo, DT Matt, and M Riedl. Application of dynamically scaled safety zones based on the iso/ts 15066: 2016 for collaborative robotics. *Int. Journal of Mechanics and Control*, 21:41–49, 2020.

[39] Christer Ericson. *Real-time collision detection*. Crc Press, 2004.

[40] Paul J Besl and Neil D McKay. Method for registration of 3-d shapes. In *Sensor fusion IV: control paradigms and data structures*, volume 1611, pages 586–606. International Society for Optics and Photonics, 1992.