IncludeVote: Development of an Assistive Technology Based on Computer Vision and Robotics for Application in the Brazilian Electoral Context

Felipe Augusto da Silva Mendonça[1][a], João Marcelo Xavier Natário Teixeira[2][b] and Marcondes Ricarte da Silva Júnior[1][c]

[1]*Informatics Center, Universidade Federal de Pernambuco, Recife, Brazil*

[2]*Electronics and Systems Department, Universidade Federal de Pernambuco, Recife, Brazil*

Keywords: Assistive Technology, Computer Vision, Robotics.

Abstract: This work presents the development of an assistive technology based on computer vision and robotics, which allows users with disabilities to carry out the complete voting process without the need for assistance. The developed system consists of a HeadMouse associated with an auxiliary robotic arm tool that contains an adapted interactive interface equivalent to the interface of the electronic voting machine. For the development of the HeadMouse, techniques based on computer vision, face detection and recognition of face points were used. It is a tool that uses the movements of the face and eyes to perform the function of typing votes through the adapted interface for the robotic arm to carry out the entire voting process. Tests carried out showed that the developed system presented satisfactory performance, allowing a user to carry out the entire voting process in a time of 2 minutes and 28 seconds. It was also possible to conclude that the system has an average throughput of 1.16 bits/s for movements with the mouse cursor. The developed system should be used by people with motor disabilities as an assistive technology, to aid in the voting process, promoting social inclusion.

1 INTRODUCTION

Currently, robotics presents a great growth and the study regarding the ways in which humans interact with robots has become a multidisciplinary field in the face of the various possible contributions to society. In industries, for example, the presence of robots has a wide application in repetitive and precision activities (Fiorio et al., 2014). One can find initiatives applied to electronics, medicine, ergonomics, domestic use, among others. It can be seen, therefore, that robots have been designed to be anywhere and perform the most varied tasks.

With this, it is possible to affirm that the world we live in has become highly dependent on computerized technologies, with the use of robotics having gained more and more space. The application of these computerized technologies is even in the Brazilian electoral process. Bearing in mind that in the past the voting process was carried out using paper ballots, in which the voter would have to fill in the ballot with the votes and deposit them in a canvas urn, responsible for storing the votes of voters in each electoral section. The use of paper ballots in the Brazilian electoral process began to be replaced by the use of electronic voting machines, which are responsible for computing voters votes quickly and safely, from 1996 onwards.

Another growing field of robotics is its use in helping people with disabilities. For example, robots programmed to detect obstacles can be found, capable of calculating the distance between the obstacle and a visually impaired person, alerting them to their proximity. Positive effects are also already found in the use of robots in the treatment of autistic children. It can be noted that robotics allied to the development of assistive technologies, contributes to people with disabilities being able to live, learn and work autonomously, through technologies that reduce, eliminate or minimize the impact of existing difficulties (Edyburn, 2015).

The Electoral Justice Accessibility Program has collaborated in equalizing opportunities for the exercise of citizenship for voters with disabilities or re-

[a] https://orcid.org/0000-0002-4904-9986
[b] https://orcid.org/0000-0001-7180-512X
[c] https://orcid.org/0000-0003-0359-6113

Mendonça, F., Teixeira, J. and Silva Júnior, M.
IncludeVote: Development of an Assistive Technology Based on Computer Vision and Robotics for Application in the Brazilian Electoral Context.
DOI: 10.5220/0011787000003417
In *Proceedings of the 18th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2023) - Volume 4: VISAPP*, pages 755-765
ISBN: 978-989-758-634-7; ISSN: 2184-4321

duced mobility. As an example, the program establishes that electronic voting machines, in addition to Braille keys, must also be enabled with an audio system, providing the Regional Electoral Courts (TRE) with headphones in special polling stations. The poll workers are also guided by the Electoral Courts to facilitate the entire adaptation process, including partnerships to encourage the registration of employees with knowledge of Língua Brasileira de Sinais (LIBRAS) (TSE, 2020).

Although the adoption of these initiatives are relevant, it is believed that broad and unrestricted access, with security and autonomy to people with disabilities or reduced mobility in the electoral process, can still advance further. In Brazil, according to data from the IBGE (Brazilian Institute of Geography and Statistics) through the National Health Survey (PNS), in 2019 the percentage of the population aged two years or more with some type of disability was 8.4%, with 6.5% with motor disabilities, representing about 13.3 million Brazilians who have difficulties not only in locomotion, but also in accessing entertainment and communication resources (IBGE, 2019).

According to data from the Superior Electoral Court (TSE), the number of voters with disabilities jumped from 1,059,077 in 2018 to 1,281,611 in 2020, which corresponds to an increase of 21% in the previous electorate. The TSE highlights that 32.56% of voters with disabilities have some mobility impairment and another 5.57% have difficulty voting. The current voting process for people who do not have arm/hand movement involves allowing a chaperone who can enter the voting booth and enter the numbers into the ballot box. Such an initiative, therefore, does not guarantee the secrecy of the vote. Another relevant point to note is that the voter turnout and abstention rate shows that voters with mobility impairments or with some difficulty in voting had 41.74% and 90.37% abstention respectively. These numbers show that this portion of the population is significant and is growing, requiring improvements and facilitators to be created, increasingly contributing to inclusion (TSE, 2022).

Thus, the development and application of an efficient human-computer interface, for people with quadriplegia, combined with a system that helps voters with disabilities to vote without a person helping them, respecting the secrecy of the vote, becomes increasingly necessary. The present work proposes a HeadMouse and Auxiliary Arm system, based on computer vision techniques, such as detection and facial recognition, in addition to robotics, to provide a voter with severe motor disability to carry out the entire voting process.

The main contributions of this work are:

- Development of a support system for people with motor disabilities to vote;
- Validation of the system through different tests.

The tests were carried out with individuals without motor disabilities and focused on validating the functioning of the tool. Assuming that the tool was designed for users who only have head movements, the test subjects were informed that in order to perform the tests they had to remain immobile and could only use their head movements, in order to simulate the use by a motor impaired person. Test results were compared to see if the user would be able to vote using the inclusive tool and without the tool, comparing both voting times. By using the tool, an individual without disability, but with the movement restrictions of a person with motor disability, only with the movement of the head could carry out the voting process and that during the day of the election, it would be possible a total of 120 disabled people carry out the voting process using the tool, considering the time period of 8AM to 5PM.

The remainder of this paper is structured as follows. Section 2 will deal with related work, both on tools based on computer vision, and the joining of tools with auxiliary technologies using robotics. In section 3, the methods used for the development of the system will be presented, as well as the tests performed with users. In section 4, the results obtained in the development and in the tests carried out will be presented. Finally, section 5 will present the final considerations on the research carried out.

2 RELATED WORK

Robotics is a relatively young field of study compared to other areas of study, however, it has highly ambitious goals, such as making it possible to perform detailed tasks, such as surgeries, by robots. One of the main focuses of study in robotics lately revolves around the creation of machines that can behave and perform activities like humans. This attempt to create intelligent machines leads us to question why our bodies are designed the way they are. A robotic arm is a mechanism built by connecting rigid bodies, called links, to each other through joints, so that relative motion between adjacent links becomes possible. The action of the joints, normally by electric motors, makes the robot move and exert forces in desired ways (Lynch and Park, 2017).

A relevant area of study in robotics involves the application of robots in an inclusive context, as an auxiliary tool in assistive technologies, enabling a person with a disability to overcome some physical limitations. Studies have shown that people with upper limb disabilities experience major limitations in their activities in terms of participation and performance (Frullo et al., 2017).

Computer technologies play an important role in human life. The effective use of these technologies by people with disabilities is difficult or unfeasible. A human-computer interface with an efficient design for people with disabilities is of great help, and this can open up, for example, employment opportunities for these people and a better quality of life. Recently, different assistive computing technologies have been proposed for different groups of people with disabilities. One of the techniques that can be used is computer vision-based tracking that uses image processing or sensors that use physiological signals to detect head movements for mouse control. Computer vision-based solutions generally rely only on a camera, tend to be cheaper than sensor-based techniques and can obtain all visual information for future applications.

(Su et al., 2005) in their research present a computational interface based on low-cost computer vision that allows people with disabilities to use head movements and facial gestures to manipulate computers. The proposed tool uses face detection, eye detection and template matching to control the mouse using the head and eyes. Using an equation of motion, the authors transform the user's current position on the screen into mouse movements. In one of the experiments performed, ten users were asked to use the HeadMouse to move the cursor to the positions of five blocks of 40x40 pixels that were randomly generated within a window of 1024x768 pixels. Each user performed the experiment twice. The first using the headmouse and the second using a standard mouse. On average, each user took 3.279 and 0.683 seconds to locate a tile using the standard headmouse and mouse respectively.

(Palleja et al., 2013) propose the implementation of a headmouse based on the interpretation of head movements and facial gestures captured with a camera. The authors propose the combination of face detection, matching algorithm models and eye movement to emulate all mouse events. The headmouse implementation was compared with a standard mouse, a touchpad and a joystick. The validation results show movement performances comparable to a standard mouse and better than a joystick, as well as good performances when detecting face gestures to generate click events: 96% success in the case of mouth opening and 68% in the voluntary blinking.

Using a deep learning-based approach, (Abiyev and Arslan, 2020) proposes a human-machine interface for people with disabilities with spinal cord injuries. In which, the proposed human-machine interface is an assistance system that uses head movements to move the mouse and blinks to trigger click events. The system is composed of two convolutional neural networks (CNN). The head mouse control is done by the first CNN and is used for the recognition of the head profile directions, which can be left, right, up, down and no action, if the head is stationary. The second CNN is used for the recognition of eye states, which can be closed (click) or open (no action). Finally, the authors performed tests with the tool and reached a value of 99.76% accuracy for head movements and 97.42% for eye movements.

The authors (Sampaio et al., 2018) and (Kamakshaiah et al., 2022) work with approaches that use a cascade of regressors to detect face points, such as the Dlib library (King, 2009). (Sampaio et al., 2018) in his work develops a tool that combines face detection and detection of face points for mouse movement and keyboard use. The author uses a training base provided by the OpenCV library (Bradski, 2000) for face detection. The face detection algorithm results in the rectangular regions of the faces found. For the detection of points of interest on the face, the author uses the Dlib library. From the points of interest of the face, it is possible to move the mouse to where the user wants in addition to clicking and activating keys. The author carried out tests of efficiency of movement and of writing and typing to validate the developed system. The mouse cursor movement efficiency test showed that the system allows users to move the cursor and perform mouse clicks smoothly and accurately with throughput of 1.20 bits/sec. The writing and typing tests with face, eye and mouth movements proved to be possible to perform 1.15 movements with the face per second.

Similarly, (Kamakshaiah et al., 2022) features a headmouse based on mouth tracking. Mouth position gap information between the front frame and the next frame in head motion video frames is used to confirm whether a mouth tracking is motion information or command information. The system consists of facial detection using the Dlib library combined with the calculation of the Mouth-Aspect-Ratio (MAR), in which it is verified whether the mouth is open or closed. MAR is used to trigger mouse movement or mouse click events.

Assistive tools can be used to compensate for these physical limitations. Examples of these structures are robotic arms, which are devices that can be mounted on a user's wheelchair or workstation, which are controlled through a joystick or some adaptable interface that can grab and carry objects. The purpose of these devices is to increase autonomy using robotics (Lebrasseur et al., 2019) propose the use of a robotic arm as an assistive tool to perform some basic actions such as drinking a drink and came to the conclusion that the use of robotic arms as an assistive technology led to a significant decrease of up to 72% task completion time.

In the same vein, (Higa et al., 2014) presents a computer vision-based assisted robotic arm for people with disabilities. The experiment carried out by the authors consists of taking a bottle of water in a pre-defined space through the robotic arm. Regarding the results found, position errors of the order of a few millimeters were observed in the experiment. The experimental results of the drinking water task with physically fit individuals showed that they could perform the tasks without any problems. (Gao et al., 2015) propose a computer vision-based user interface for a mobile robotic arm for people with severe disabilities. The system allows a robotic arm to feed a severely disabled person according to the user's plate preference. When selecting the dish using eye movements, the position coordinates are transmitted to the robotic arm, and this starts a feeding program which the robotic arm finisher extends the food at the selected dish location, picks up some with the utensil and takes it to the user's home mouth. The results found demonstrate that the algorithms can produce significant improvements in the performance of activities of daily living.

For the development of the headMouse, an approach similar to that of (Sampaio et al., 2018) was used, in which the tool is based on the detection of facial points, aiming to transform head movements into mouse movements through an interpolation and an equation of motion smoothing that will be described in the methodology section.

3 METHODOLOGY

The first stage of the research, called HeadMouse Architecture and Development, was dedicated to the development of the natural computational interface tool that will allow voters to vote with little assistance, in addition to providing total vote secrecy. The second stage includes the architecture of the auxiliary system, which was dedicated to the development of the system that will receive user input, through an adapted interactive interface, as well as the development of the entire backend so that the robotic arm can carry out the voting process. The third stage of the methodology will be dedicated to carrying out the tests and analyzing the results obtained.

3.1 Headmouse Architecture and Development

The HeadMouse control system is designed for people with physical disabilities who suffer from motor neuron disease or severe cerebral palsy. The system calculates the head position and converts it to actual mouse positions. This system also detects when the user blinks, and uses this information to control the main mouse button. The headmouse system includes the following parts: image acquisition and facial recognition, blink analysis and conversion to mouse control signal. The architecture of the proposed system is presented in Figure 1. The entire HeadMouse system was developed in python language due to the easy integration of the libraries used.

To obtain the input image, the computer's webcam will be used with the help of the OpenCV (Bradski, 2000) library. Then, the face recognition step begins. The developed system uses the Face Mesh module of the MediaPipe (Lugaresi et al., 2019) library to implement the detection of facial points. From the detected facial points, it is possible to obtain the exact position of the user's face in the input image. It is also possible to identify various elements of the user's face, such as mouth, nose and eyes, where calculations will be performed to detect when the user blinks.

The analysis of ocular states is performed by calculating the EAR (Eye Aspect Ratio). (Soukupova and Cech, 2016) proposed a real-time algorithm to detect blinks in a video sequence from a standard camera. The algorithm estimates the position frame. Equation 1 shows the formula used to obtain the EAR. The output value is a scalar quantity met by detecting a face from an image, finding the Euclidean distance from the corresponding eye coordinates, and plugging it into Equation 1.

$$EAR = \frac{\|p_2 - p_6\| + \|p_3 - p_5\|}{2 * \|p_1 - p_4\|} \tag{1}$$

To perform the mouse cursor movement functions and to trigger the mouse click event, the system needs to interpret the position of the face and its elements in isolation, in addition to transforming these positions into functionalities. As already mentioned, the detection of points on the face allows, from the distance between these points, to determine the position of the face and the ocular state.

Movement:

To perform the mouse movement action, we took advantage of the detection of face points to obtain the

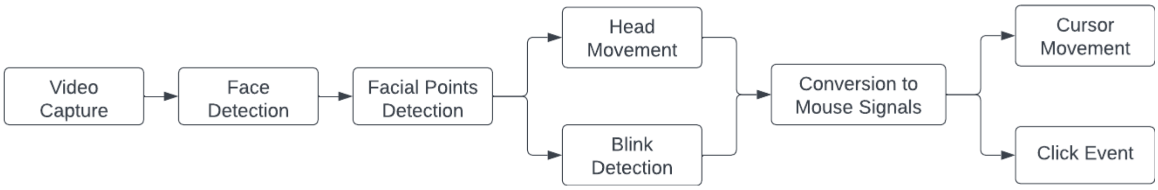

Figure 1: Headmouse System Architecture.

central point of the eyes. Initially, the landmarks corresponding to the right and left iris were highlighted, creating a circle around the user's eyes with the help of OpenCV, making it possible to calculate the central point of the eyes from this created circle. After obtaining the central point of each eye, it was possible to calculate the midpoint between the two eyes.

To transform the movement of the head into movement of the mouse, a rectangle of predefined size was placed in the center of the image obtained, which is 640x480 pixels. Then, an interpolation is performed between the calculated midpoint in relation to the rectangle placed in the center of the image and the mouse pointer in relation to the entire device screen. To move the mouse, the Autopy library was used, which allows the user to move the mouse to a desired position just passing the points as a parameter. Thus, the points obtained with the interpolation were passed to Autopy's movement function.

Directly converting the position of the midpoint between the eyes to the position of the mouse cursor presented two problems in its implementation. The first is related to sharp sensitivity, small head movements generate large cursor movements, thus making it impossible for the user to position the cursor accurately. The second problem is related to the instability of the cursor positioning, caused by problems in the variation of the detection of facial points. This problem exists because even if the user does not perform any movement, the detection of the face does not always happen in the same place, since it is an estimate. To solve the aforementioned problems, it was necessary to implement a smoothing function.

There are several transfer functions used to smooth out motion and mitigate the effects caused by detection problems. (Palleja et al., 2013) proposed the application of a quadratic relationship with the mouse pointer displacement to provide different sensitivity when performing large and small pointer displacements. This enhanced control procedure is defined by computing two intermediate position increments (xinc, yinc), as seen in Equation 2.

$$xinc = A * (xm - xpm)^2 * sign(xm - xpm)^2 \qquad (2)$$

$$yinc = A * (ym - ypm)^2 * sign(ym - ypm)^2 \qquad (3)$$

In Equation 2 and 3, A is a configuration parameter that defines the speed at which the pointer moves, which was set to 0.001; xm, ym is the absolute position of the pointer obtained through the interpolation performed; Xpm and Ypm is the previous position of the pointer on the screen; the final position of the screen pointer Xf and Yf is calculated using Equation 4 and 5. Then, the mouse pointer is finally placed at the screen coordinates Xf and Yf.

$$xf = xinc + xpm \qquad (4)$$

$$yf = yinc + ypm \qquad (5)$$

Click:

To trigger the click event, the EAR of each eye was calculated to obtain the average between the results. If the average between the results is below 0.35, the click event is triggered. This implementation is also affected by face detection issues, causing unwanted clicks at various times. To try to mitigate this problem, the condition was set that the event would only be triggered if the EAR average was below 0.35 in three consecutive frames. To trigger the click event, the Autopy click function was used.

3.2 Auxiliary System Architecture

The system is composed of an interactive interface, responsible for inputting data from the HeadMouse and a back-end responsible for moving the robotic arm. This back-end is composed of a server that receives and processes the information sent through the interface and forwards it to the robotic arm.

3.2.1 Interactive Interface

The interface was developed in flutter and made entirely for desktop. Figure 2 shows the tool's initial screen, where some instructions were given to the user, aiming at an initial introduction to the tool. The blink button on the interface triggers an alert congratulating the user for learning and instructing him to continue with the next steps. Finally, the user should wait for the polling station's signal to start voting. After voting began, the user had the ballot box keyboard displayed on the screen, being able to type in the numbers chosen for each candidate, as shown in Figure 2 and Figure 3.

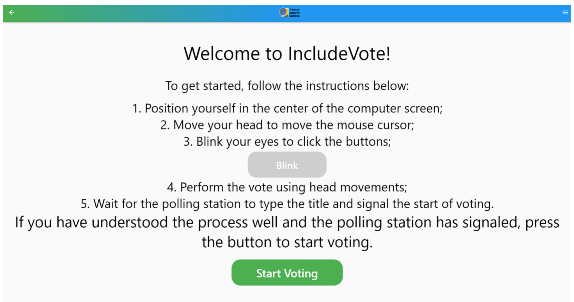

Figure 2: Initial interactive interface.

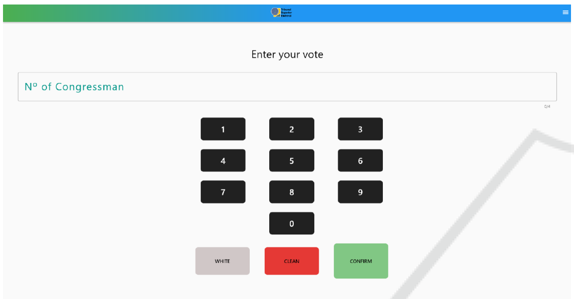

Figure 3: Adapted Interactive Interface.

3.2.2 Robot Movement

The physical arrangement of the system is pre-defined and static, all available movement positions are pre-defined. The robotic arm is positioned on a table next to the electronic voting machine, as shown in Figure 4. The arm model used to perform the necessary movements is the Gen3 Lite, developed by Kinova®. After voters type in the vote of their choice and press the confirm button, the data is sent to a backend, processed, and sent to the robot so it can carry out the voting process.

A manual calibration of the position of each of the keys was performed, using the robotic arm's python API, Calibration is performed only once at the beginning of the use of IncludeVote, in order to ensure that the robot types the vote chosen by the user correctly. Upon receiving the candidate's number, the robotic arm accesses the key positions corresponding to the requested keys and performs typing. At the end of the voting process, the robot sends feedback to the interface that frees the screen to type the next vote until the position of president, after which the voter ends the voting and is released.

Figure 4: Physical Arrangement.

3.3 User Tests

The test methodology adopted consists of two batteries of tests and a satisfaction survey, the first battery of test focused on tests carried out with the electronic voting machine, and the second battery focused on tests using only the developed HeadMouse, the second battery of tests is performed with the aid of an auxiliary tool called FittsStudy. The tests cover user performance and usability qualitative aspects. The two batteries of tests were carried out with 15 people, 11 male and 4 female, women between 18 and 39 years old and men between 18 and 59 years old. The tests were submitted to people without motor disabilities and with daily contact with computers and smartphones.

The test protocol defined for tests using IncludeVote initially consisted of positioning the user in the center of the computer's camera, at a distance of 80cm between the user's head and the computer screen. It is worth noting that all 15 subjects underwent all tests. The test application order was defined this way so that the user had the least possible interaction with the HeadMouse and robotic arm set in order to simulate a real application, where a person with severe physical disabilities arrives at the polling station and uses IncludeVote to carry out the voting process.

The system was used equally for all users in all tests, and the hardware resources made available were the same. A notebook with a 2.80Ghz Intel Core i7-1165G7 processor, 16GB RAM memory, 64-bit Windows 11 operating system, with the webcam having a resolution of 640x480 pixels was used.

3.3.1 Application Tests

The application tests of the IncludeVote tool were carried out using an electronic voting machine. The first test consisted of the user performing the entire voting process using the HeadMouse and the auxiliary system, the second test was performed with the user performing the entire standard voting process, without the aid of the robotic arm. The choice of the test with IncludeVote as the first test was made so that the user had as little contact as possible with the tool, with the individual having received only a few instructions for using the tool, without receiving any training.

Initially, the voter was positioned in the center of the notebook and exposed to the voting start screen, shown in section 3.2.1. After that, the user was released to start the voting process, which would follow the voting flow of the general elections: federal deputy (Deputado Federal - DF), state deputy (Deputado Estadual - DE), senator (Senador - SE), governor (Governador - GV) and president (Presidente - PR). When voting started, a timer was started to count the time that the voter took to complete the voting process. In addition to counting the total time, the time it took the user to type the votes for each position was also taken into consideration. Thus, it was possible to count the total voting time without robot interactions with the electronic voting machine in order to make comparisons with the following tests. The number of times the user had to type confirm was also counted to get an idea of how many times the user voted wrongly or had an involuntary click.

The second test was performed as a way of validating the first, where the voter was exposed to the standard voting process, having to perform it normally, typing the keys without the help of IncludeVote and the robotic arm. This approach becomes interesting for drawing a comparison between voting using the tool and conventional voting. In this test, the entire process time was accounted for, as well as the typing times between each position. The number of times the user had to correct the vote was also computed.

3.3.2 Usability Tests

The mouse cursor movement and main mouse button activation tests were based on the ISO/TS 9241-411 standard, which deals with evaluation methods for the development of physical input devices on human-computer interaction ergonomics. It aimed to evaluate whether the system allows the user to control the cursor and activate the main mouse click function efficiently.

The test environment chosen to carry out this stage of the research was FittsStudy. FittsStudy follows the protocol defined by the ISO/TS 9241-411 standard that defines how the evaluation of pointing devices must be performed, determining how the test environment must be structured and the calculations that must be performed.

The test environment is composed of circular regions, over which the user must position the mouse cursor, organized as 11 regions in a circular format. The evaluation parameters are calculated as a function of the distance D between the regions and their size W, as described in Figure 5.

The environment was configured to perform three sequences of clicks in the circular regions. Thus, two tests were performed for each individual. In the first test applied, it was defined that the user would have to use the HeadMouse tool developed, while the second test was performed with a standard mouse connected to the notebook.

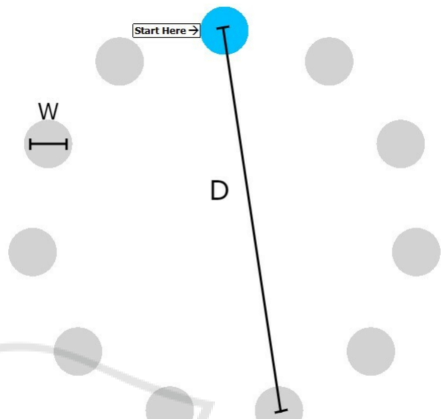

Figure 5: Test Environment.

The sequence of circle sizes varied from the largest to the smallest, varying the level of difficulty, on purpose, so that when users reached the last sequence where the circles were smaller, they had already overcome the learning barrier.

During the test, the tool collected the time to perform the tasks. Through time, the average click time of the targets (MT) was calculated. By dividing the calculated average time and the environment difficulty index (ID), it is possible to calculate the performance indicator Throughput (TP), given in bits/s. This indicator determines how much information the user was able to enter into the operating system. The higher the value, the greater the volume of information that the user can enter into the operating system. Through the tool, it was also possible to evaluate the profile of the click between the click trajectory of one circle and another, defining it as a hit, an error or an outlier, which was when the user performed a click very far from the correct point, many times due to failure to detect blinks.

3.3.3 Qualitative Assessment

A satisfaction survey was carried out to complement the other tests, analyzing some qualitative characteristics such as ease of use and comfort. The satisfaction survey was conducted with 15 users. The survey had questions that involved 3 topics of interest, first questions were asked in order to outline the user's profile, then the questions were related to the ease of use of the tool. Finally, questions related to comfort when using the tool were asked in order to make it possible to map the user's perception of the system.

With the exception of the themes of the questions related to the user's profile, which contained more direct questions, the answers should be classified into 5 levels, demonstrating their agreement or disagreement with a given statement. Options ranged from: 1) Strongly Disagree; 2) Disagree; 3) Neither Agree nor Disagree; 4) Agree; 5) Totally Agree.

4 RESULTS AND DISCUSSION

After applying the two batteries of tests and the satisfaction survey, it was possible to trace the profile of the user who participated in the survey. The age range of individuals who participated in the test ranged from 18 to 59 years. A question was asked about the ease of use of the tool, with 86.7% of the individuals having fully agreed and 13.3% having partially agreed with the statement. Individuals were also asked whether the tool allowed easy selection of buttons and options on the screen, with 66.7% of individuals fully agreeing and 33.3% partially agreeing with the statement. Finally, individuals were asked about their level of comfort in using the tool, with 60% of individuals responding that the tool is very comfortable and the other 40% responding that the tool was just comfortable.

Using the measured time data that is presented in Figure 6, it presents the average of the typing times for each position, both for typing using the HeadMouse and for typing being done manually by the user. Thus, it is possible to verify that the proportion between the typing time using the tool and the typing times done manually vary between 1.66 and 3.09, with the Federal Deputy field (4 digits) as the user takes more time to type being even longer than the typing time for the position of State Deputy, which has a greater amount of numbers to be typed (5 digits). This is because the user's first contact with the tool occurs here.

Tab 1 presents the general averages of voting times for both the number typing process and the complete process. It is possible to verify through Tab 1. that the average of the total time spent in the complete process was approximately 2 minutes and 28 seconds and it is worth noting that half of this time is allocated to the robot to perform the typing of the

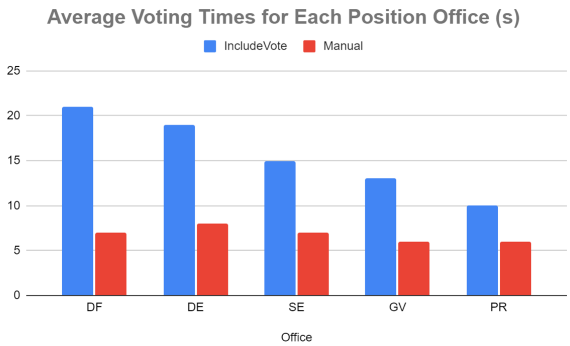

Figure 6: Average Voting Times in Seconds (DF: Federal Deputy; DE: State Deputy; SE: Senator; GV: Governor; PR: President).

Table 1: Average Voting Times.

Average Voting Times			
Complete Process		**Manual Typing**	
IncludeVote	**Manual**	**IncludeVote**	**Manual**
0:02:28	0:00:35	0:01:18	00:00:35

vote. If we take into account that the voter identification step takes 1 minute and 30 seconds, it would be possible for a person with a disability to complete the entire voting process, including identification, in a time of 4 minutes. Taking into account that there are 9 hours of voting, and considering another 30 seconds in the total time, which would be the time for the voter to enter and leave the session, it would be possible to provide that approximately 120 voters with disabilities, in each session, carry out the voting process.

In this test, the number of times the user entered the wrong number and had to press the correct key was also counted. In all tests, only one subject had to correct more than once. This shows that the interface developed has a high degree of interactivity and is well integrated with the HeadMouse tool developed.

Regarding the tests performed with the FittsStudy tool, Figure 7 presents a graph of the average throughputs in each of the test rounds performed by the users. Figure 8 shows a graph of the average time taken to perform each circle-to-click task. Finally, Figure 9 presents a comparison between the hit rates of the tasks for each of the tools used.

Analyzing the graphs in Figure 7 and 8, it is possible to determine some characteristics of the developed system. It is possible to verify that the biggest decrease in task completion time and increase in throughput occurs during the first round, approximately 20%, on average. This behavior indicates the learning speed of the developed system, since the user needs little time to familiarize herself/himself and use it effectively.

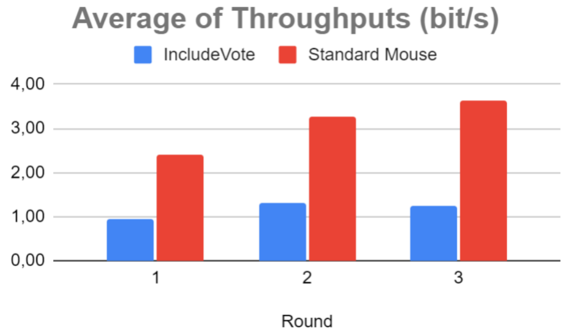

Figure 7: Average of Throughputs for each Task.

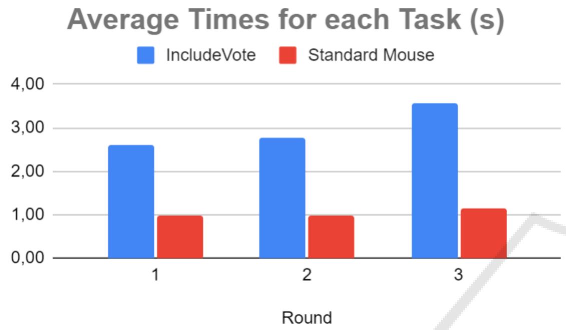

Figure 8: Average of Times for each Task.

Regardless of the difficulty of the proposed test environments, users on average performed the test with close throughput values ranging between 0.95 and 1.23 bits/s, having a global average of 1.16 bits/s. In his experiment (Sampaio et al., 2018) performs movement efficiency tests and writing and typing to validate the headmouse he developed. The mouse cursor movement efficiency test showed that the system has a throughput of 1.20 bits/s, a value close to the one found in this work. This is an indication that users were able to complete the task with the same efficiency, regardless of difficulty. Analyzing the individual throughput values of each participant, it was possible to verify that there is not such a large variation in the values between the first and second rounds, with the values varying, that is, the lowest value was not always the first, not following a fixed pattern.

The throughput of the third round also had great variation, but because it was a test where the circles were much smaller and consequently much further away from each other, and due to the large loss of clicks by the user, it is not interesting to make a comparison between the three tests. It was identified that this problem of lost clicks is caused by the fixed value of the EAR defined above. For people with very small eyes, the system presented false blink detection and this also dropped the hit rate of some individuals as a blink detection error represented an involuntary click that the tool identified as an error. As can be seen in the graph presented in Figure 9, the hit rate varied greatly between the two tools.

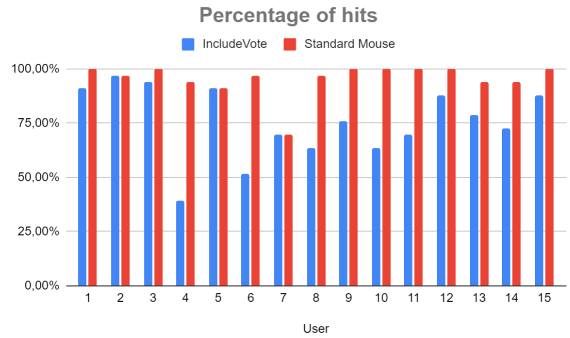

Figure 9: Percentage of hits.

Tab 2 presents a compilation of the global averages obtained in the tests with each tool. As it can be seen, the global average of the time required to perform each click task was approximately 1.04 seconds for the standard mouse and 2.99 seconds for IncludeVote. (Su et al., 2005) obtained similar results in his experiment, where each user should move the cursor to the positions of five randomly generated blocks within the screen, performing the experiment twice with each user, the first using the headmouse and the second with the standard mouse, where each user took 3.279 and 0.683 seconds to perform the task using the Headmouse and the standard mouse respectively. In Tab 2, it is also possible to verify that the global average of the IncludeVote hit rate was approximately 75.56%, whereas the test with the standard mouse showed an overall hit rate of 95.56%.

This click loss problem was not observed in the tests with the auxiliary system, as the test was not focused on the performance of the HeadMouse tool, but on the performance of the auxiliary system as a whole, having obtained significant values regarding the voting time and the number of times the user had to correct the vote. Another interesting parallel between the two tools can be drawn by analyzing the time each user takes to perform the tasks. Comparing the average time to perform the tasks, it was possible to conclude that when the user is below average in tests involving the auxiliary system, he is also below average in usability tests performed with FittsStudy.

5 CONCLUSION

The present work aimed to develop and evaluate the application of a system that allows a voter with motor disabilities to carry out the voting process. The interaction with the system occurs through a HeadMouse

Table 2: Global Averages.

Global Averages					
Throughputs (bits/s)		**Times (s/task)**		**Hits (%)**	
IncludeVote	**Standard Mouse**	**IncludeVote**	**Standard Mouse**	**IncludeVote**	**Standard Mouse**
1.16	3.10	2.99	1.04	75.56%	95.56%

based on computer vision that works by moving the head and blinking the eyes. Thus, the user, through an adapted interface, can select the voting preference. The process of typing in the electronic ballot box the vote chosen by the user is carried out through a robotic arm, the robot receives the vote through the interface and types it in the electronic ballot box. The tests made it possible to visualize whether the objectives of the developed system were achieved and the features presented achieved the expected performance.

The validation results obtained with a group of untrained individuals and without mobility impairments showed good performance in the tests performed with the voting system and in the movement and actuation tests, being comparable to those of a standard mouse. The test carried out with the electronic voting machine showed that the system allows the voting process to be carried out in an average time of 2 minutes and 28 seconds, allowing up to 120 voters with physical disabilities to carry out the voting process. As for the proportion of time the user takes to type the numbers using the HeadMouse and a standard mouse, it was 1.66 and 3.09 for the best and worst times, respectively. This ratio represents how many times the HeadMouse's typing time is longer than that of the standard mouse. The cursor movement and mouse activation tests showed that the system allows users to move the cursor and perform mouse clicks with a throughput of 1.16 bits/s with an average time to perform tasks being approximately 2.99 with a hit rate of 75.56%. Thus, it can be considered that it is a fast learning system. The satisfaction survey showed that the developed system is easy to use, not being an uncomfortable experience.

Future works could be developed aiming at improvements in the system, considering that the system, especially the HeadMouse, also due to the low resolution used, has some limitations, such as the detection of blinks of some users. Another point of attention is the rectangle used as reference for moving the head. Currently, the rectangle is fixed in the center of the screen, but in the future it would be interesting to generate the rectangle having the user's head as reference. Another point of improvement in the proposed system would be the decrease in the robustness of the robotic arm used, opting for a robotic arm with a lower price, thus providing the application of the system in several electoral sessions throughout Brazil. In addition to improvements in the applied techniques, tests with users of reduced mobility could be carried out, thus making adjustments to the system based on the needs that would arise and identifying other open points.

A video demonstration of IncludeVote is available at link.

ACKNOWLEDGEMENTS

This work has been supported by the research cooperation project between Softex (with funding from the Ministry of Science, Technology and Innovation—Law 8.248) and CIn-UFPE.

REFERENCES

Abiyev, R. H. and Arslan, M. (2020). Head mouse control system for people with disabilities. *Expert Systems*, 37(1):e12398.

Bradski, G. (2000). The opencv library. *Dr. Dobb's Journal: Software Tools for the Professional Programmer*, 25(11):120–123.

Edyburn, D. L. (2015). Expanding the use of assistive technology while mindful of the need to understand efficacy. In *Efficacy of assistive technology interventions*. Emerald Group Publishing Limited.

Fiorio, R., Esperandim, R. J., Silva, F. A., Varela, P. J., Leite, M. D., and Reinaldo, F. A. F. (2014). Uma experiência prática da inserção da robótica e seus benefícios como ferramenta educativa em escolas públicas. 25(1):1223.

Frullo, J. M., Elinger, J., Pehlivan, A. U., Fitle, K., Nedley, K., Francisco, G. E., Sergi, F., and O'Malley, M. K. (2017). Effects of assist-as-needed upper extremity robotic therapy after incomplete spinal cord injury: a parallel-group controlled trial. *Frontiers in neurorobotics*, 11:26.

Gao, F., Higa, H., Uehara, H., and Soken, T. (2015). A vison-based user interface of a mobile robotic arm for people with severe disabilities. In *2015 International Conference on Intelligent Informatics and Biomedical Sciences (ICIIBMS)*, pages 172–175. IEEE.

Higa, H., Kurisu, K., and Uehara, H. (2014). A vision-based assistive robotic arm for people with severe disabilities. *Transactions on Machine Learning and Artificial Intelligence*, 2(4):12–23.

IBGE (2019). Pesquisa nacional de saúde, 2019 - informações sobre as condições de saúde da população, a vigilância de doenças crônicas não transmissíveis e os fatores de risco a elas associados. https://www.ibge.gov.br/estatisticas/sociais/saude/9160-pesquisa-nacional-de-saude.html?=&t=resultados.

Kamakshaiah, K., Sony, P., and Neeraja, B. (2022). Human computer interaction based head controlled mouse. *Journal of Positive School Psychology*, 6(8):145–152.

King, D. E. (2009). Dlib-ml: A machine learning toolkit. *The Journal of Machine Learning Research*, 10:1755–1758.

Lebrasseur, A., Josiane Lettre, O., OT, P. S. A., et al. (2019). Assistive robotic arm: Evaluation of the performance of intelligent algorithms. *Assistive Technology*.

Lugaresi, C., Tang, J., Nash, H., McClanahan, C., Uboweja, E., Hays, M., Zhang, F., Chang, C.-L., Yong, M. G., Lee, J., et al. (2019). Mediapipe: A framework for building perception pipelines. *arXiv preprint arXiv:1906.08172*.

Lynch, K. M. and Park, F. C. (2017). *Modern robotics*. Cambridge University Press.

Palleja, T., Guillamet, A., Tresanchez, M., Teixidó, M., del Viso, A. F., Rebate, C., and Palacín, J. (2013). Implementation of a robust absolute virtual head mouse combining face detection, template matching and optical flow algorithms. *Telecommunication Systems*, 52(3):1479–1489.

Sampaio, G. S. et al. (2018). Desenvolvimento de uma interface computacional natural para pessoas com deficiência motora baseada em visão computacional. *Mackenzie*.

Soukupova, T. and Cech, J. (2016). Eye blink detection using facial landmarks. In *21st computer vision winter workshop, Rimske Toplice, Slovenia*.

Su, M.-C., Su, S.-Y., and Chen, G.-D. (2005). A low-cost vision-based human-computer interface for people with severe disabilities. *Biomedical Engineering: Applications, Basis and Communications*, 17(06):284–292.

TSE (2020). Programa de acessibilidade. https://www.tse.jus.br/eleicoes/processo-eleitoral-brasileiro/votacao/acessibilidade-nas-eleicoes.

TSE (2022). Estatísticas eleitorais, pessoas com deficiência, comparecimento e abstenção - 2020. https://dadosabertos.tse.jus.br/dataset/comparecimento-e-abstencao-2020.