Development of a Computer Interface for People with Disabilities based on Computer Vision

Gustavo Scalabrini Sampaio and Maurício Marengoni

Postgraduate Program in Electrical Engineering and Computing, Universidade Presbiteriana Mackenzie, São Paulo, Brazil

Keywords: Human-computer Interfaces, Computer Vision, Face Tracking, Face Landmarks, People with Disabilities.

Abstract: The growing of the population with disabilities in the world must be accompanied by the growing of research and development of tools that help these users on basic computer activities. This paper presents the development of a system that allows the use of personal computers using only face movements. The system can be used by people with motor disabilities who still have head movements, such as superior members amputees and tetraplegic. For the development of the proposed system, the most efficient techniques in previous works were collected and analyzed, and new ones were developed in order to build a system with high performance and precision, ensuring the digital and social inclusion of the target public. Tests have shown that the tool is easy to learn, has a good performance and can be used in everyday computer applications.

1 INTRODUCTION

Computing and communication, from the point of view of resources and usage, are constantly growing, offering to the users tools for learning, working, entertaining, getting information and socializing. This development have been accompanied by new ways for user interaction, such as touchscreens, voice recognition, virtual reality glasses, among others. Even with these technological advances, which have given, for the majority of users, more convenience and easier interfaces, people with disabilities still have difficulty to interact with computers.

The World Health Organization (WHO, 2011) points out that about 1 billion people in the world have some kind of disability and describes the barriers these people face, one of which is the access to information and communication. Devices like smartphones and computers, despite having some accessibility resources, are not developed taking into account the needs of people with disabilities, often making it impossible for these people to use these devices. For the WHO, the computer technologies developers and researchers should strive to provide for users with disabilities the same experience that other users have, thus ensuring their social inclusion and academic and professional development.

This work was developed taking into account the needs of users with disabilities, the WHO suggestion for development computer interfaces based on face movement and to ensure, through the access of the media, the social and digital inclusion of this public. Will be presented the development of a natural computational interface based on computer vision, that allows access to the resources of the Windows operating system by the user through face movements, eyes blinking and mouth opening and closing. No movement of the lower parts are required, so the tool can be used for people with numerous degrees of motor disabilities, especially users who have low precision or amputation of the upper limbs up to tetraplegics. The proposed system was developed using concepts and techniques that proved to be efficient in similar systems, and presents high processing performance, contributing to a better user experience. In addition to performance, this paper presents other contributions:

- The mouse cursor movement occurs smoothly and accurately.
- Unlike other systems, allows a simulation of typing by face movements.
- Allows the user to configure resources, such as the system's operating mode, the keystroke or the click that will be activated, enable or not the functions related to the eyes and mouth and to adjust the sensitivity of the cursor movement.
- Has a control mechanism, where the user can start, stop or pause its operation.
- Implements auxiliary algorithms to improve the efficiency of facial detection and perform a system user search automatically.

Sampaio, G. and Marengoni, M.
Development of a Computer Interface for People with Disabilities based on Computer Vision.
DOI: 10.5220/0006616205660573
In *Proceedings of the 13th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2018) - Volume 4: VISAPP*, pages 566-573
ISBN: 978-989-758-290-5

2 RELATED WORKS

The analysis of similar works to the proposed is important to determine the state of the art and to find the main characteristics of the system in use today. In a constructive way, the positives and negatives aspects of each work were identified. In general, systems with interfaces using face movements are basically composed of a face detection module, a module for converting face movement to mouse cursor positioning, and a module for click simulation or other functions for computer interface. Among the works analyzed, the works presented by (Betke et al., 2002) and (Pallejà et al., 2011) stand out, for making the tool available for download and using techniques that make it possible to use in computers with a webcam.

The analysis of other works was help to define the functionalities of the proposed system and the most efficient techniques to execute some function. The works of (Tu et al., 2005), (Fu and Huang, 2007), (Xu et al., 2009), (Pallejà et al., 2011) and (Kraichan and Pumrin, 2014) use for face detection the technique proposed by (Viola and Jones, 2001) that has as training method the Adaboost and Harr Cascade as detection method. (Ji et al., 2014) and (Gor and Bhatt, 2015) indicate that this technique, coupled with the search area reduction through the skin color segmentation, improves detection performance and accuracy. The technique presented by (Xu et al., 2009) that uses Active Appearance Models (AAM) to map the points on the face and use them to move the mouse cursor is a simple way to accomplish this task. Apply a transfer function to perform the mapping between face position and cursor position as shown by (Kjeldsen, 2006) and (Pallejà et al., 2011) is important to ensure accuracy and smoothness of cursor movement.

The performance in this type of system is essential. Delays in processing reduce the mouse cursor positioning precision and the efficiency of state definition of eyes and mouth. (Kraichan and Pumrin, 2014) shows that the face detection module can have its performance increased representatively by not performing face detection in a range of frames, where the system uses the face position found in the last detection to perform its analysis. This technique works because the user, in a range of 3 to 5 frames, considering a system with performance of at least 30 frames per second (fps), can not perform movements with great displacement. These techniques were used for the development of the proposed system and new ones were added, in order to produce a tool that uses only the computer webcam and has an acceptable performance.

3 SYSTEM DEVELOPMENT

The proposed system was designed to perform its functions using a webcam and uses, in addition to face movement, the opening and closing of eyes and mouth to control the resources of the Windows operating system. The system does not require calibration and it has 2 operation modes:

- The mouse mode allows the user to perform the mouse functions with the face movement. The mouse cursor is controlled by the direction of the user's nose, this reference point gives the user precision to select icons and buttons of at least 30x30 px. The mouth opening and closing simulates the clicks with the mouse's left button, and the blink of the right and left eyes allow the simulation of functions chosen by the user. This mode has a drag function, when the mouth is opened for about 1s an object can be dragged, when the user close his mouth the object is released. This mode allow the user, together with a virtual keyboard, access the internet, social networking and learning tools.
- The keyboard mode, not present in other works, allows the user to simulate keyboard typing using face movements and the eyes and mouth opening and closing, face movement can simulate up to 4 keystrokes, activated by moving the face up, down, left and right, the user can configure which keyboard key will be simulated on each move. This mode mode allows the user to play simple games and use tools developed to people with disabilities based on selectable symbols.

The programming language used for the development of the proposed system was C++, with OpenCV and Dlib libraries. The techniques used to implement the system's functionalities will be detailed below.

3.1 Pre-processing

In order to reduce the search area and increase accuracy in face detection, some pre-processing has been implemented. These pre-processing aims to find the image region that has skin color, that region is extracted and passed to the face detector. The figure 1 shows the preprocesses executed step by step.

The first procedure performed is the image brightness adjustment, necessary to improve the efficiency of the skin color segmentation. The color domain is converted from RGB **(a)** to YCrCb **(b)**. In the YCrCb domain the Y channel represents the brightness and the Cr and Cb channels together represent the colors. The brightness adjustment is performed on the Y channel using histogram equalization technique **(c)**. After brightness adjustment, the skin color

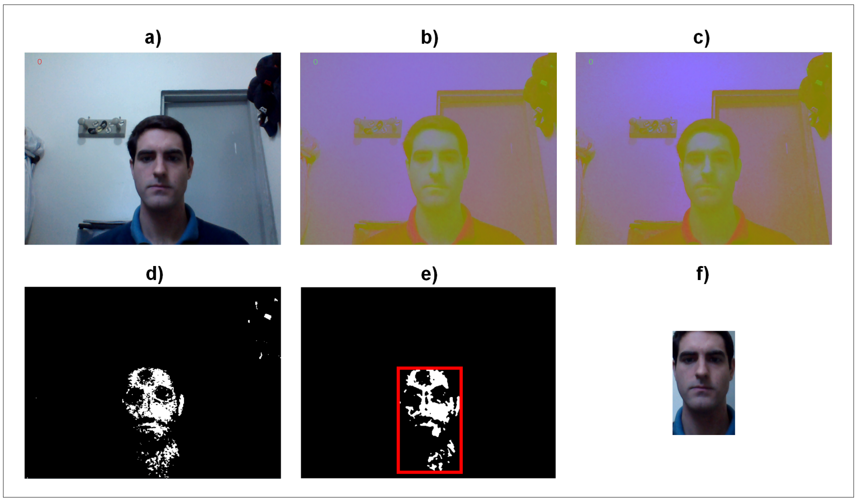

Figure 1: Pre-processing for the face detection function of the proposed system. a) Input frame; b) RGB color domain conversion to YCrCb; c) Brightness adjustment by histogram equalization; d) Skin color segmentation; e) Noise filter; f) Final image where face detection is performed.

is segmented by filtering the pixels based on intervals of YCrCb channels values (**d**), the range of values of the Cr and Cb channels is small, varying further in the Y channel. After skin color segmentation, the resulting mask presents noise that, in practice, are sparse pixels having color in the interval defined by the segmentation filter, this noise is removed applying a median blur filter in the image (**e**). Finally the initial search region for face detection is defined by a rectangle of dimensions delimited by the pixels filtered of the mask (**f**). It is observed the significant reduction of the search area and the consequent improvement in face detection performance.

3.2 Face Detection

The face detection is the main element of the proposed system, and it is strongly related to system performance. The search for the user's face is performed with the help of the OpenCV library and it happens in 2 different search modes.

The first search mode is performed when no face has been found yet, the system then performs all the pre-processing in each frame defining the region that should be used by the face detector, the process is repeated until a face is found. Once one or more faces are detected its positions represented by rectangles are stored in a vector, then the system filters the face closest to the center of the image and assumes this face as the user of the system. The detected face position receives a 13% additional dimension offset, this value was determined empirically considering processing speed and detection accuracy. Figure 2 presents the face detection in the region defined by skin color segmentation. The second face search mode is performed every 3 frames, in the area defined by the rectangle show in figure 2, which allows small user movements among frames. If a face is found the system continue, otherwise it returns to the first face search mode. The second face search mode significantly increases the system's performance. It was verified that this technique practically double the system performance.

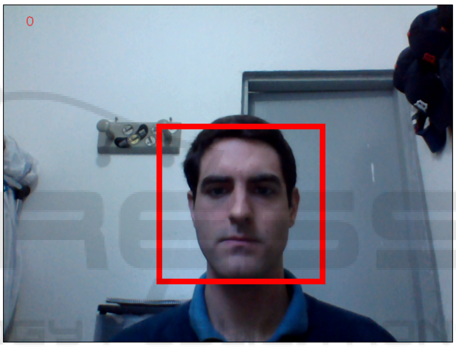

Figure 2: Face detection performed in the region defined by skin color segmentation, the region detected is augmented by an offset of 13%.

3.3 Face Landmark Detection

Having the face location is not enough for the system to execute the control of the operating system, so it is necessary to look for more elements that allow the system to identify the face's position and other face elements such as eyes, mouth and nose.

The proposed system use face landmarks to estimate the face position. These points provides also information about the position of the eyes and mouth. Unlike the optical flow technique, which needs to

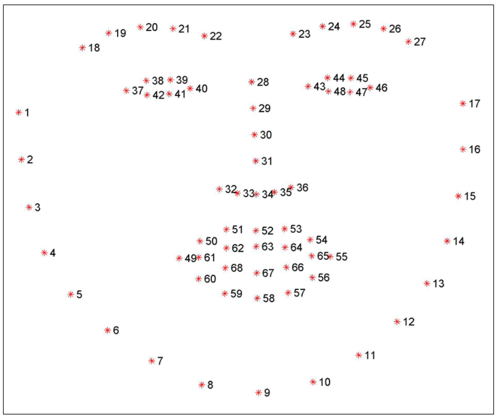

Figure 3: Position of the 68 face landmarks detected. Source: (Sagonas et al., 2013).

perform the analysis using information of 2 or more frames, the use of the face landmarks allows estimating the face position only with the information of the last frame. The face position is used to control the mouse cursor and simulate typing.

The implementation of this feature was performed with help of Dlib library, that provides an algorithm for face landmark detection based on the algorithm described by (Kazemi and Sullivan, 2014), that uses regression trees to estimate the position of face landmarks. This algorithm runs in the area of face detected and finds 68 face landmarks, proposed by (Sagonas et al., 2013), as shown in figure 3.

3.4 Mouse Cursor Positioning

After all detection operations, the system has the necessary data to transform the face position into the mouse cursor position. This task is performed on all frames and use the face landmarks, the points 40 and 43 in figure 3 are used to define the x axis. Points 1 and 4 are used as limits for the y axis. Point 31, located at the tip of the user's nose, was chosen as reference point to control the mouse cursor because its movement is natural to the user.

The operation of converting the face position to mouse cursor position has 4 steps. The first is to find the percentage value of the reference point position in relation to the limits of the axis, this percentage is calculated considering a dead zone of 20% at each end of the limit, necessary to avoid values higher or lower than the established limits, a situation that can happen due to the constant detection of the face landmarks. The second is to convert this percentage to the corresponding pixel value of the captured frame, by defining the reference cursor position. This direct conversion generates an unstable positioning, since small displacements on the reference point cause large reference cursor displacements in the frame. The implementation of the third step was necessary to eliminate the instability of positioning and to ensure precision and smoothness in the cursor movement. In this step the system analyzes the previous and current reference cursor position, this last is the position desired by the user, and it calculates its new position through of a transfer function. The transfer function was developed based on the behavior of the natural logarithmic curve, where the values resulting from the function *ln(x)* are used as percentage of movement in relation to the reference cursor position and the position desired by the user. The last step is to perform the position conversion of reference cursor, which has a scale equal to the frame resolution, to the position in the scale of the user's monitor. With this last calculated position, a command is issued to the operating system to position the mouse cursor.

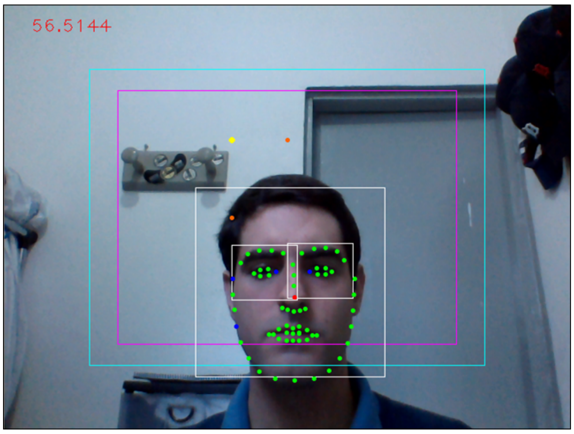

Figure 4: Reference points and regions used in the proposed system.

The steps shown are used in mouse mode, for keyboard mode operations the execution occurs until step 2, that position is then interpreted as the user face direction. For each face direction a keyboard key typing is simulated by the operating system. The figure 4 illustrates the points and regions used to interpret face movement. The green dots indicate the face landmarks detected. The blue dots represent the reference limits of the axes. The red dot represents the reference point for moving the mouse cursor. The orange dots represent the axis reference position converted to the position in the captured frame. The yellow dot shows the reference mouse cursor. The white rectangles indicate the face and eyes detected. The cyan rectangle represents the reference region of the user screen. The magenta rectangle represents the limits for setting the face position in keyboard mode.

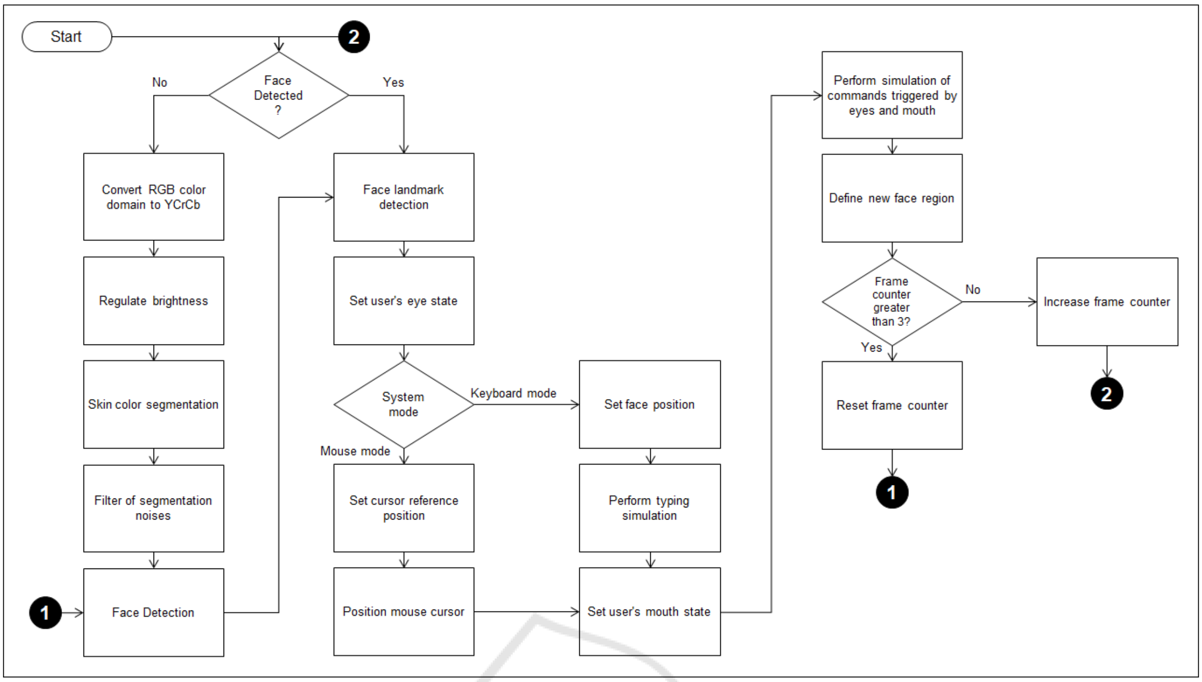

Figure 5: Proposed system flowchart.

3.5 Clicks and Typing

The clicks and typing simulation are performed through the face movements, in keyboard mode, and by the interpretation of the eye and mouth movements. The user can choose the function that each movement perform, these function may be a click or a typing simulation. In mouse mode the mouth is only used to perform the mouse left button click. For the system to activate a certain function, it is first necessary to define the state of the mouth and eyes.

The mouth state definition is performed using points 22 and 9, in figure 3, as y axis reference and points 63 and 67, the latter being located at the top and bottom of the lips and varying in distance during opening and closing of the mouth. The ”open” state is detected if the distance between points 63 and 67 is greater than 3% of the distance between points 22 and 9. This value allows the user to remain with the mouth slightly open to breathe without activating the click, a small opening movement perform the action. A drag function was implemented in the system, to do this the user must remains with the mouth open for 50 frames, which corresponds to 0.5s to 1s, when closing the mouth the object is released.

The definition of eye states could not be performed using the face landmarks directly, since the camera's low resolution prevents the points detected in the eye region to vary the distance between them satisfactorily. These points were only used to determine the eyes region and extract that region for analysis. The eye state definition is simply the detection of the eye, using the same technique used for the face detection proposed by (Viola and Jones, 2001), but with a trained base with positive images of open eyes. If the eye is detected its state is ”open”. To increase the system performance, this operation is performed only every 5 frames. The flowchart in figure 5 illustrates the operations performed by the system.

4 TESTS AND RESULTS

To demonstrate that the tool is easy to use, learn and has a good performance, 4 types of tests were performed with 10 users without any type of disability. Each test encourages the use of certain functionality or simulates an environment of use, allowing to evaluate if the tool meets the need of the user to perform computer tasks. In terms of performance, the proposed tool performs its operations by processing 100 fps. Value 3 times greater than the Camera Mouse proposed by (Betke et al., 2002) and Head Mouse proposed by (Pallejà et al., 2011).

The system was configured equally to all users, and the testing room have no special illumination. The lights were artificial and placed at the ceiling of the room. The machine has a Intel Core processor i3-3110M dualcore of 2.40Ghz, 8GB memory, Windows 10 of 64 bits and a 640x480 pixels webcam.

4.1 Test 1 - Mouse Functions

(Bian et al., 2016) presents a standardized form, defined by ISO/TS 9241-411 standard, to evaluate the performance of the movement and clicks of a pointing interface. (Soukoreff and MacKenzie, 2004) presents more details about this kind of test and points out several recommendations. This standard defines the test environment and the calculations of values of the measured parameters. The test environment is composed of circular regions organized in a circle shape. The parameters are measured according to the distance values between the targets (**D**) and the size of the target (**W**), the figure 6 shows an example of a test environment, the target is presented to the user in green color. The tool performance is measured according to its throughput (**TP**) defined by

$$TP = \frac{ID}{MT} \tag{1}$$

representing the amount of information that the user can transmit to the operating system using the tool in bits/s. **ID** represents the difficulty index of the environment and is defined by

$$ID = log_2(\frac{D}{W} + 1), \tag{2}$$

MT is the average time that the user moves the mouse cursor from one target to another. (Soukoreff and MacKenzie, 2004) draws attention to this measure of time, since the standard disregards the target selection time, which in practice could generate measurements based only on velocity, eliminating the precision factor. According to (Soukoreff and MacKenzie, 2004) measuring only velocity with absence of precision becomes a non-informative measure and recommends that collected values that do not consider precision should be taken from the analysis. To ensure the credibility of the tests performed, the test environments used to evaluate the proposed tool consider the time of movement and selection.

Table 1: Environments features of movement and click tests.

Environment	D	W	ID
1	534	76	3.00
2	534	57	3.37
3	305	57	2.67

The users were submitted to 3 different environments, for each environment 3 test rounds were executed. Users performed the tests without ever having used the tool, which made it possible to analyze the usage learning curve and to verify if the tool can be used intuitively, only by explaining to users that the tasks should be performed with face movements. Table 1 presents the characteristics of the 3 test environments. The table 2 was generated with the mean of the measured values for each environment, where **TCT** represents the task completion time in seconds, **MT** represents the mean time of movement and selection from one target to another in seconds and **TP** represents the throughput in bits/s (the larger the values the better the throughput). The Final Average showing the final result of the test. In this table is possible to verify that once the user learns how to use the system the time for task completion reduces, even with the increase of the difficulty index among the environments. For comparison the mouse throughput is approximately 4.50 bits/s. The figure 7 shows the path traveled by the mouse cursor and the click points of one of the tests. It can be verified that there are no noisy paths and few adjustments in the region of the targets. After some interactions with the tool the user demonstrates ability using the tool, facilitating the accomplishment of the tasks after each repetition. The table 3, together with the figure 8, demonstrate the task completion time reduction and increased throughput of the test, ”1-2” and ”2-3” columns show the decrease and growth between rounds 1 and 2, and 2 and 3, respectively.

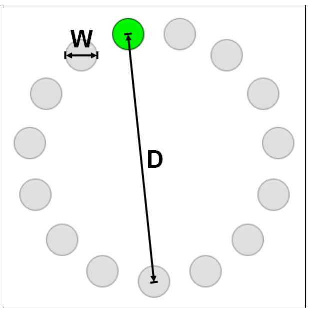

Figure 6: Test environment of movements and clicks with indication of measures.

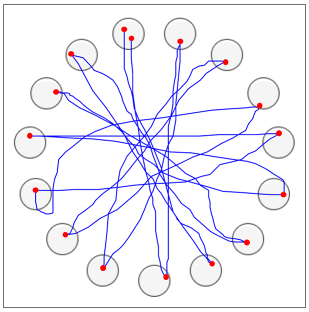

Figure 7: Path traveled by mouse cursor and click points, test performed in test environment 1.

This test shows that the proposed tool is easy to use, has a high precision and it allows the user to move the mouse cursor smoothly.

Table 2: Average results of the rounds of each test environment.

	Rounds											
	1			2			3					
Environment	TCT	MT	TP	TCT	MT	TP	TCT	MT	TP	MTCT	MMT	MTP
1	67.48	4.50	0.82	43.68	2.91	1.07	40.82	2.72	1.16	50.66	3.38	1.02
2	47.00	3.13	1.14	46.24	3.08	1.16	43.25	2.88	1.25	45.50	3.03	1.18
3	40.10	2.68	1.06	40.34	2.69	1.06	36.34	2.42	1.20	39.11	2.61	1.10
							Final Average			45.09	3.01	1.10

Table 3: TCT Reduction and TP growth between the test rounds.

TCT Reduction		TP growth	
1-2	2-3	1-2	2-3
35 %	7 %	31 %	8 %
2 %	6 %	1 %	8 %
-1 %	10 %	-1 %	14 %

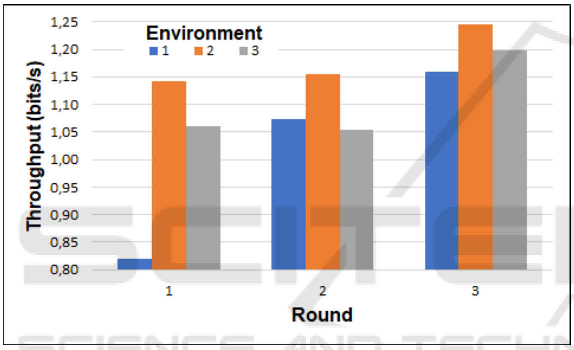

Figure 8: Increased throughput between test rounds.

4.2 Test 2 - Virtual Keyboard Writing

The second test was a simulation of a common task in personal computers: typing a text. With this feature it is possible to conduct searches on the internet and use social sites, it is also possible to access training and distance learning sites as well as professional tools.

Three sentences were written using a virtual keyboard, the keys on the keyboard measure 30x30 px. The sentences have different sizes to allow the analysis of learning the task and the use of the tool in a small and restricted area. The phrases elaborated were ”Bom dia” (Good morning), ”Visão Computacional” (Computer vision) and ”Vamos assistir ao jogo” (Let’s watch the game). The user was instructed not to write special characters and space, if an error occurred it was not necessary to delete the wrong letter. The writing time averages were 30s for sentence 1, 83s for sentence 2 and 81s for sentence 3, so writing sentence 3, the longest, took a time close to sentence 2, this means that the user has learned the task and can execute it faster every attempt.

4.3 Test 3 - Click Evaluation

The third test had more technical goals, for evaluating the effectiveness of clicks with the eyes and mouth. A test environment was built with a large 200x200 px button positioned in the center of the screen, the test system itself instructed the user to click that button using the opening and closing of the eyes and mouth. For each eye and for the mouth it was counted how many clicks were possible in 20s. On average, users were able to click 48 times with their mouth, 13 times with their right eye and 11 times with their left eye. The efficiency of the clicks with the mouth is much higher then clicks performed with the eyes, this difference comes from the technique used to determine the state of the mouth and eyes.

4.4 Test 4 - Face Typing Evaluation

The fourth test was carried out to evaluate the efficiency of keyboard typing. The test environment presenting 4 squares with size 200x200 px in the center of the screen forming a cross. These squares alternated between the green and gray colors, when a square is green the user had to move the face in the direction of the square in order to ”type it”, systemically the user activated the keys A, S, D and W. A sequence of 20 movements was elaborated, each direction was requested 5 times. The task of typing can be used in simple games and in request tables, where an icon in a position describes an item (such as drink, fruits, etc) desired by the user. On average, users were able to perform the task in 23s. Most users did the test at a frequency below 1 typing per second, enabling the use of this functionality in real applications.

5 CONCLUSION

The system proposed used efficient techniques for face detection, interpretation of face movements, conversion of face position to mouse cursor position and other techniques to allow users to interact with personal computers in a simple and intuitive way. The system using only a webcam and delivering a performance of 100 fps. Face detection has high performance, from the implementation of auxiliary techniques, such as skin color segmentation and detection cut between frames. Face landmarks allowed to identify the face movement, this signal is converted by a transfer function to a accurate and smooth mouse cursor movement. The system also enables the face movement to perform the typing simulation. The opening and closing of eyes and mouths have been translated for simulation of clicks and typing.

From the tests performed, it was demonstrated the efficiency of the tool and the ease of users to learn how to interact with the system and perform simple computer tasks. People with disabilities of the upper limbs and spinal cord injury, as long as they have the head movement, can use this tool and enjoy the resources available in the computer and the internet. The digital inclusion of these people can stimulate the increase of their self-esteem and provide opportunities for academic and professional development.

As future works, the proposed system must undergo more tests of comparison with other existing similar tools. Tests should also be performed with users with disabilities in order to confirm the usability of the system. After these tests and possible adjustments in its functionalities, the tool must be made available to the public.

Human-computer interfaces aimed to people with disabilities should always be research and development topic, as society must ensure that these users are included in all activities. Future systems may determine more efficient techniques of using the opening and closing of the eyes as an interface to the system, considering a low-resolution and reduced-size input image without performance decrease. These systems should always provide the best performance possible, as low performance makes the system unusable in real situations or affect their usability.

REFERENCES

Betke, M., Gips, J., and Fleming, P. (2002). The camera mouse: Visual tracking of body features to provide computer access for people with severe disabilities. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 10(1):1–10.

Bian, Z.-P., Hou, J., Chau, L.-P., and Magnenat-Thalmann, N. (2016). Facial position and expression-based human–computer interface for persons with tetraplegia. *IEEE Journal of Biomedical and Health Informatics*, 20(3):915–924.

Dlib (2017). Home page. Available in: http://dlib.net/. Access on May 31, 2017.

Fu, Y. and Huang, T. (2007). hMouse: Head tracking driven virtual computer mouse. In *2007 IEEE Workshop on Applications of Computer Vision*. IEEE.

Gor, A. K. and Bhatt, M. S. (2015). Fast scale invariant multi-view face detection from color images using skin color segmentation & trained cascaded face detectors. In *2015 International Conference on Advances in Computer Engineering and Applications*. IEEE.

Ji, S., Lu, X., and Xu, Q. (2014). A fast face detection method combining skin color feature and AdaBoost. In *2014 International Conference on Multisensor Fusion and Information Integration for Intelligent Systems (MFI)*. IEEE.

Kazemi, V. and Sullivan, J. (2014). One millisecond face alignment with an ensemble of regression trees. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1867–1874.

Kjeldsen, R. (2006). Improvements in vision-based pointer control. In *Proceedings of the 8th international ACM SIGACCESS conference on Computers and accessibility - Assets 06*. ACM Press.

Kraichan, C. and Pumrin, S. (2014). Face and eye tracking for controlling computer functions. In *2014 11th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*. IEEE.

OpenCV (2017). Home page. Available in: http://opencv.org/. Access on May 31, 2017.

Pallejà, T., Guillamet, A., Tresanchez, M., Teixidó, M., del Viso, A. F., Rebate, C., and Palacín, J. (2011). Implementation of a robust absolute virtual head mouse combining face detection, template matching and optical flow algorithms. *Telecommunication Systems*, 52(3):1479–1489.

Sagonas, C., Tzimiropoulos, G., Zafeiriou, S., and Pantic, M. (2013). 300 faces in-the-wild challenge: The first facial landmark localization challenge. In *2013 IEEE International Conference on Computer Vision Workshops*. IEEE.

Soukoreff, R. W. and MacKenzie, I. S. (2004). Towards a standard for pointing device evaluation, perspectives on 27 years of fitts' law research in hci. *International Journal of Human-Computer Studies*, 61(6):751–789.

Tu, J., Huang, T., and Tao, H. (2005). Face as mouse through visual face tracking. In *The 2nd Canadian Conference on Computer and Robot Vision (CRV 05)*. IEEE.

Viola, P. and Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*. IEEE Comput. Soc.

WHO (2011). *World Report on Disability*. World Health Organization.

Xu, G., Wang, Y., and Feng, X. (2009). A robust low cost virtual mouse based on face tracking. In *2009 Chinese Conference on Pattern Recognition*. IEEE.