ROCC – A Webcam-Mouse for People with Disabilities

Paulo A. A. Esquef[1,2] and Rogério Caetano[1]

***Resumo*— Este artigo apresenta o sistema ROCC, que emula as funcionalidades básicas de um *mouse* de computador pessoal. ROCC é projetado para oferecer uma forma alternativa e simples para usuários com necessidades especiais interagirem o computador. Uma nova técnica para detectar a base do nariz é proposta como recurso que permite controlar o posicionamento do cursor de tela. Ademais, é apresentado um esquema capaz de diferenciar se o usuário está com os lábios cerrados ou abertos, permitindo implementar a função de *click*. As técnicas de processamento de imagem usadas no sistema são descritas e exemplificadas ilustrativamente. No mais, considerações sobre o desempenho do sistema são discutidas.**

***Palavras-Chave*— Processamento de imagem, interação homem-computador**

***Abstract*— This paper introduces ROCC, a webcam-based system that emulates common functionalities of a computer mouse device. ROCC is designed to provide physically disabled people with an alternative solution to interact with personal computers. A novel technique for detecting the nose region between the eyes is proposed as a way to control the cursor's movements. Moreover, a scheme that distinguishes between two mouth status: with closed or open lips, is introduced as a means to trigger the click function. The image signal processing techniques used in the system are described, illustrative examples are offered, and considerations on the system performance are discussed.**

***Keywords*— Image processing, human-computer interaction**

I. Introduction

In the last two decades, the area of computer vision has been making tremendous progresses. Part of those advances was moved by the progressive increases in the processing power of personal computers (PCs) and its popularization as a common household appliance. Moreover, optical devices, such as color webcams and miniature infrared cameras, are more affordable to the general public. Nowadays, applications of computer vision range from industrial processes to entertainment business and security systems [1], [2].

One important branch of computer vision is human-computer interaction and its role in assisting physically disabled people. Early means of providing control over the screen cursor, other than using a mouse device, included systems based on electro-oculogram (EOG) measurements [3]. Besides requiring a custom hardware to perform data acquisition and pattern recognition, EOG-based systems have the inconvenience of being intrusive.

An alternative to EOG-based mouse emulators are video-based systems that aim at performing eye tracking or gaze estimation. Some of those systems require a head-mounted apparatus, while others ask for infrared cameras or even two different cameras for exploring stereo vision [4]. In some cases, only control over the cursor positioning is provided. The click function is either realized by other devices, such as a foot switch, or implemented by leaving the cursor rested over a given object for a pre-defined amount of time.

The remaining of this paper presents the proposed webcam-mouse system, which is called ROCC, and describes its processing stages and graphical user interface.

[1]Laboratory of Electronics, Fundação Des. Paulo Feitoza, Manaus-AM, Brazil, Email: rcaetano@fpf.br. [2] Instituto Nokia de Tecnologia, Manaus-AM, Brazil, Email: pesquef@yahoo.com

II. Overall System Description

ROCC integrates hardware and software resources. The primary hardware devices include a PC with support to USB connection, running on Windows XP operating system, and a Windows-compatible USB color webcam. The primary software is a graphical user interface (GUI), through which all control tasks, functionalities, and image processing needed for cursor control are realized. Simulation software Matlab [5] was employed to design the GUI.

Using a webcam as a measuring device offers several advantages. First of all, it is non-intrusive, leaving the user free of wires and attached apparatus. Second of all, webcams are nowadays popular and cheap computer peripherals, making feasible devising low-cost systems for the end-user. On the other hand, the proper operation of the system depends on satisfying a set of usage conditions that are listed below.

1) The level of light exposure must be properly adjusted on the webcam to avoid too dark or too bright images;
2) The user must be placed in front of the webcam and the monitor that shows the GUI. Deviations of about 10 degrees are allowed, though;
3) The distance between the user and the webcam can vary from 30 to 80 cm. In any case, it must be guaranteed that the whole head of the user is seen within the video frame;
4) Only one face, the user's, must be present in first plane in the video frame;
5) The user's face must be free of hindering elements or objects. Thus, users must not have a beard or a mustache. Similarly, using large sun-glasses or wearing caps that cover the forehead is forbidden;
6) Apart from the face and neck, other skin regions that surround the face must be covered by cloths;
7) The background scene as well as the user's cloths can be colored, except for reddish and yellowish tonalities;

ROCC's main objective is to emulate two common functionalities of a mouse device: control over the position of the cursor on the screen and trigger the selection of objects over which the cursor is placed.

As regards the first objective, the adopted strategy was to detect and track a specific point on the user's face. For that purpose, a technique that allows detecting the nose region

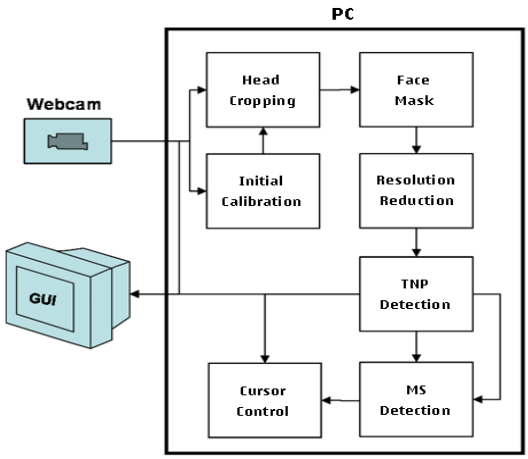

Fig. 1. Stages of the processing chain.

between eyes is proposed. It is based on an early method introduced by Kawato [6].

As for the second aim, it was decided to detect a change of status of a given face element in order to trigger the object selection. The most obvious choice would be checking for intentional eye blinking. This solution was discarded because robust discernment between involuntary and intentional eye blinks were found difficult to accomplish when the user is far from the webcam or wearing eye-glasses.

Instead of the eyes, it was decided to check the status mouth, i.e., with closed or open lips, as the trigger for the click emulation. The immediate advantage is that the mouth is a big facial element on the face. Moreover, its movements, besides being slower than those of the eyes, do not interfere with the visual process. One drawback of this solution is the possible occurrence of false alarms when the user is engaged in conversation. Another disadvantage is that the skin of regions that surrounds the mouth and lips must be visible. This restriction prevents users that have a dense beard or mustache from using the ROCC program.

III. Video Frame Capture

The first step of the system is to setup and initiate the video capture from the webcam and transfer the obtained frames to Matlab workspace. Video frame capture in Matlab is carried out on driver's level. Once started, video capture is carried out continuously, with frames being stored in a circular buffer. The frame capture ratio is defined by the webcam configuration.

The upload of captured frames to Matlab workspace is performed asynchronously, following an on-demand scheme, regardless of the actual frame capture ratio. Thus, the most recent frame stored on the capture buffer is uploaded to the workspace. This frame is then used as the input of a processing chain that is in charge of locating the top of the nose point (TNP) as well as detecting the mouth status (MS). A copy of the original frame is saved for visual feedback through the GUI. A block diagram depicting the functional stages of the processing chain is shown in Figure 1.

IV. Image Segmentation

The objective behind image segmentation is to crop the image in order to isolate the user's head from the observed

Fig. 2. Initial calibration with the oval contour seen in white.

scene. For that, the system uses a skin color segmentation method [7]. Since ROCC is meant to serve one individual at a time, it was decided to employ an initial calibration step to tune the segmentation task according to the color of the user's face and illuminations conditions.

A. Initial Calibration

The initial calibration is done through the GUI, which shows in real-time the video of the observed scene superposed with a reference oval contour. In the sequel, the user is required to position his/her head in order to match, as close as possible, the contour of his/her face with the provided reference, as demonstrated in Figure 2.

The user or his/her assistant completes the task by pushing a certain button on the GUI. The captured image at that moment is converted from RGB to YCbCr color space [1]. Then, the histogram of pixel values associated with the Cr component is computed for the set of pixels that are inside the reference contour. To cover the ranges with null count, a smooth envelope of the histogram curve is computed. Finally, the envelope is compared with a threshold to select a range of pixel values associated with skin color in the Cr component.

The threshold is computed as a fraction of the average value of the histogram envelope. For instance, choosing the fraction between 1/5 and 1/2 yields satisfactory results. Finally, the minimum and maximum Cr values whose count is above the threshold are taken as the range limits for skin color pixels. Figure 3 illustrates the intermediate results of the initial calibration procedure.

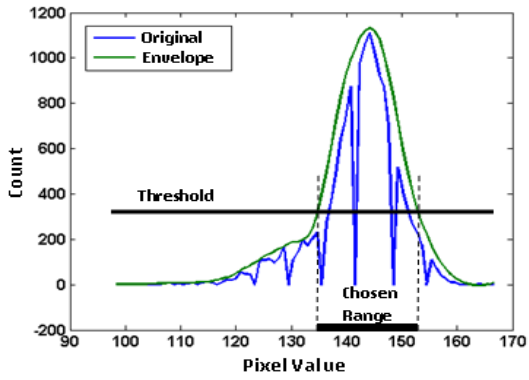

Fig. 3. Histogram of Cr values within the face region.

B. Head Cropping

For head cropping, the Cr component of each frame is obtained and a binary image is computed: pixels whose Cr values

are inside the previously computed range (see section IV-A) are set to 1, otherwise, they receive null values.

The attained binary image should reveal the user's face as the largest object, although other smaller clusters also occur. These regions are usually due to objects in the background scene whose color is close to skin color. Thus, the most obvious way to isolate the face region consists of taking only the region with the largest area.

Finally, the cropping limits are set as those of the left, right, top, and bottom boundaries of the binary image associated with the face region. Figure 4 shows the intermediate steps of the cropping procedure.

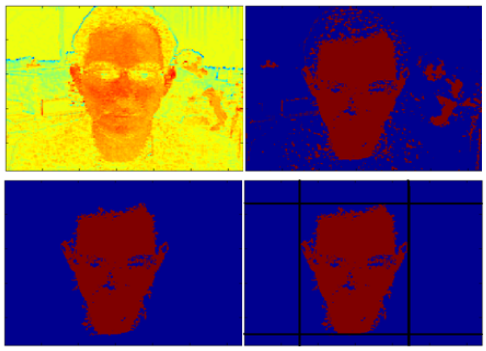

Fig. 4. Intermediate images used for head cropping. From the top-left image proceeding clock-wisely: Cr component; thresholded binary image; region of largest area; and idem with cropping limits shown as black lines.

V. Top of the Nose Point Detection

A. Detection Image

In [6] Kawato and Tetsutani describe a method for detecting and tracking the nose region in between the eyes, the TNP, from grayscale images. The underlying idea explores the contrast between the dark regions associated with the eyes and the bright regions associated with the nose and forehead. For that, Kawato proposes a circle-frequency filter that works as a template matching tool. The novelty introduced in ROCC is the replacement of the circle-frequency filter with a cross-like filter pattern, as shown in Figure 5. This filter will be referred to as cross pattern (CP).

With reference to Fig. 5, when the CP is centered at the TNP, the path to be followed is AoBCoD, corresponding to forehead, TNP, left eye, right eye, TNP, and nose, respectively. The resulting curve of pixel values will have a bell-like shape turn upside-down, which can be matched to a Gauss window. One advantage of such filter element is that the width of the Gauss window can be easily adjusted to fine tune the matching procedure to a given user.

As with Kawato's technique, the matching procedure can be realized through 2D image filtering. The filtering is performed upon a decimated low-resolution version of the original grayscale image. For instance, input images of size 240×320 can be decimated by a factor of 2 to 4. The decimated images help to produce smoother curves of pixel values.

In the CP case, the 2D filter element is built starting from a squared null element with same size as the height of the cross. Typically, the height of the cross is chosen between 12

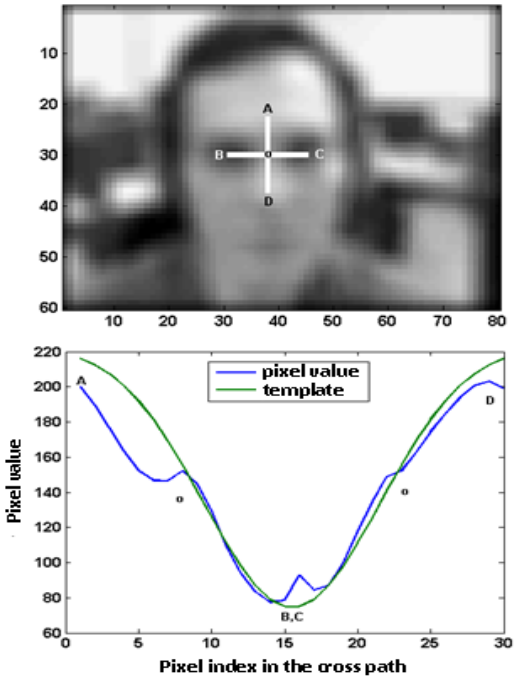

Fig. 5. Cross-like filter pattern: CP filter centered at the TNP (top) and curve of pixel values in the route AoBCoD vs. a Gauss window template (down).

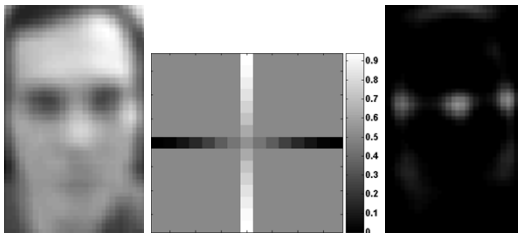

Fig. 6. From left to right: low-resolution grayscale image; cross filter element; and detection image.

and 18 pixels. Then one fills in the path AoBCoD with values taken from a zero-mean Gauss window.

In Kawato's method each pixel of the filtered image is squared to form the detection image [6]. This resource makes local minima appear as local maxima, rendering the TNP detection more difficult. Therefore, in ROCC, another modification is introduced, in that a half-wave rectification is applied to the pixels of the filtered image, before taking their squares. The outcome is a cleaner detection image. Figure 6 shows from left to right: a cropped and decimated grayscale image; the filter element (offset by 0.5 for better visualization); and the detection image.

Finally, the TNP is selected as the global maximum of the detection image that lies inside the face region. For that, a mask to isolate the face region is needed.

B. Face Mask

As mentioned in section V-A, the cropping of the original image to isolate the head yields, as a by product, a crude mask of the face region. This binary image, denoted here by M_0, may contain holes, cavities, or other undesirable artifacts.

First of all, M_0 should be cropped and decimated, as done with grayscale image used to obtain the detection image (see previous section). The resulting image, M_1 may contain holes

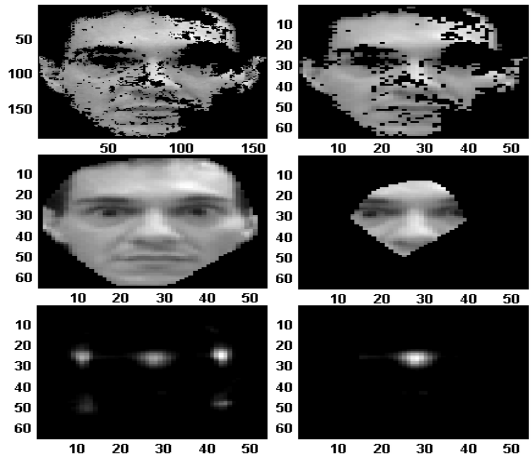

Fig. 7. Face masks: M_0 (top-left); M_1 (top-right); M_2 (middle-left); M_3 (middle-right); detection image (bottom-left); detection image masked by M_3 (bottom-right).

and cavities. Hence, a solution to close then is to compute the filled convex hull of M_1 to generate the mask M_2.

The goal here is not to build an exact face mask, but to obtain a mask that is capable of isolating the TNP from other local maxima in the detection image. Some of these maxima tend to occur on the boundaries of the face. In contrast, the TNP is located well inside M_2. Hence, an easy way to get rid of these maxima M_2 is eroded by a square structuring element of side 18, yielding a reduced face mask M_3.

Figure 7 shows the steps taken for computing M_3 from M_0 in an example where M_0 contains several holes and cavities. In order to aid visualization, the masks are applied to the corresponding grayscale images.

Note that the bottom images in Figure 7 represent the real usage of mask M_3. Moreover, it can be observed that masking the detection image is effective to isolate the TNP from other local maxima. Thus, the TNP can be taken simply as the global maximum inside the masked region of the detection image.

C. Performance

The performance of CP filter was evaluated against Kawato's proposition. Kawato applied his technique to The Database of Faces [8], which contains 400 faces. He reports that in 394 of the cases the TNP is successfully detected [6]. However, it should be noted that, in Kawato's method, besides the circle-frequency filtering, geometrical criteria are adopted to pre-select some local maxima as TNP candidates. Moreover, these candidates are verified by means of a more sophisticated eye template matching, before deciding for the winner.

The proposed CP filtering algorithm, even without the use of a reduced mask of the face, attains similar levels of successful TNP detection in The Database of Faces [8]. More specifically, 392 correct TNP detections out of 400 grayscale faces were obtained. However, the selection criterion is much simpler, with no need of pre-selection geometrical criteria or eye template matching verification. Naturally, this counts favorably to the CP filtering scheme.

Unfortunately, performance evaluation using the reduced face mask M_3 could not be verified because The Database of Faces contains only grayscale images. Nevertheless, it can be speculated that a better performance for the CP filtering could be achieved would M_3 be applied to the detection image.

VI. Mouth Status Detection

In ROCC mouth status (MS) detection is carried out upon the cropped and decimated grayscale frames that were used in TNP detection. The underlying idea is to explore the pixel value contrast between the brightness of the face skin area and the darkness of the interior of an open mouth.

MS detection assumes the TNP location as known. This information serves a reference for two tasks: 1) defining a region of interest (ROI) that contains the mouth; 2) obtaining the average pixel level associated with the face region.

A. Mouth Region Selection

It is known for a long time that the height of the forehead is approximately the same as the vertical distance between the mouth and eye levels. As seen in section IV-B, the upper limit for image cropping for head selection is the top of the forehead. Hence, one can use the distance, d, in number of pixels, between the TNP and the top of the image, to locate the mouth level on the face. Of course, in order for this strategy to work, it is necessary that the forehead be visible.

The ROI for MS detection is defined as the intersection of two binary images: the face mask M_2 and a horizontal stripe ranging from $1.6d$ to $2.5d$ with respect to the top of the grayscale image. An illustration of the ROI selection criterion is displayed in Figure 8. An example of ROI selection for MS detection in a real case is seen in Figure 9.

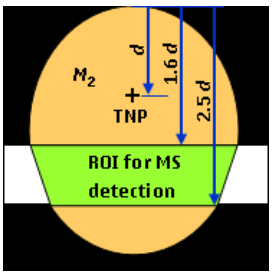

Fig. 8. Criterion to obtain the ROI (green region) for MS detection.

B. Image Thresholding

Concerning the MS detection, the expectation is that, when the mouth is open, the pixels belonging to the interior of the mouth be much darker than the average pixel value of the face. Therefore, pixel level thresholding can be applied as a selection criterion.

An estimate of the average grayscale level of the face, hereafter denoted AGL, is computed as the mean value of a vertical stripe of pixels centered at the TNP. For instance, a column of 11 pixels suffices for the aimed purpose, since it spans part of the forehead and nose, which are representative of face skin, as seen in the left panel of Figure 9.

The AGL is used as an adaptive reference for thresholding the grayscale image. In other words, a threshold proportional to AGL will be experimentally chosen so that the pixel values associated with the mouth region lie below the threshold when the mouth is open.

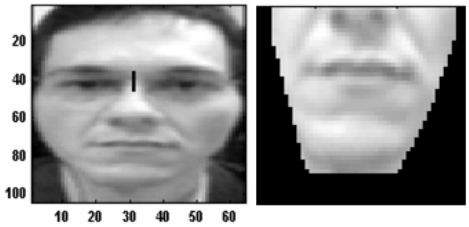

Fig. 9. ROI selection for MS detection: input image (left) and ROI (right).

However, before performing image thresholding, it is convenient to emphasize the dark regions of the face region. Since the mouth represents a dark horizontal area on the face, a suitable resource consists of filtering the grayscale image with an infimum operator, computed within a horizontal rectangular filter element. As a consequence, dark regions tend to be expanded horizontally. The rectangular filter has been chosen to have dimensions 2 rows per 10 columns. The effect of such filtering to the grayscale image is depicted in Figure 10.

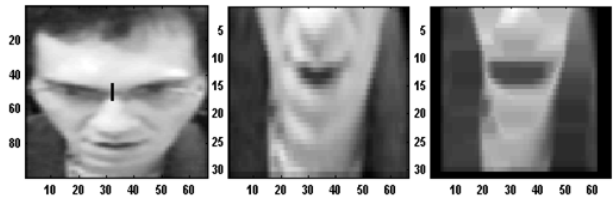

Fig. 10. Pre-processing before image thresholding: input image (left); corresponding ROI for MS detection without using M_2 (middle); and idem, but filtered with the aforementioned infimum operator (right).

When it comes to the value of the threshold, it has been found by trial and error that applying a threshold between 1/10 and 1/2 times the value of AGL leads to satisfactory results. The attenuation factor can be used as a parameter that controls the balance between miss and false MS detection. In this case, the closer to 1/2, the higher the probability of false detection.

Figure 11 illustrates the effect of the threshold choice on the MS detection. The panels show the unmasked filtered ROI images on the left column. The corresponding masked detection images are shown on the right column. For the detection images of the first two rows, the threshold was chosen as 0.4AGL, whereas for that of the last row 0.65AGL was adopted. It can be observed that while choosing 0.4AGL yields satisfactory MS detection results, setting the threshold as 0.65AGL leads to false detection.

C. Feature Detection

Ideally, after applying the threshold and the face mask M_2, the resulting binary image should be a black rectangle if the mouth is shut, as opposed to a white stripe within a rectangular black background, if the mouth is open.

However, depending on the proportions of the user's face, the nostrils may be inside the selected region. If so, the nostrils pixels that lie below the threshold can trigger the MS as open. To avoid that the following criterion was adopted: **consider MS as open only if the count of white pixels in any row within the ROI exceeds 12 pixels**. This way, spurious white spots in the binary image do not trigger an MS change.

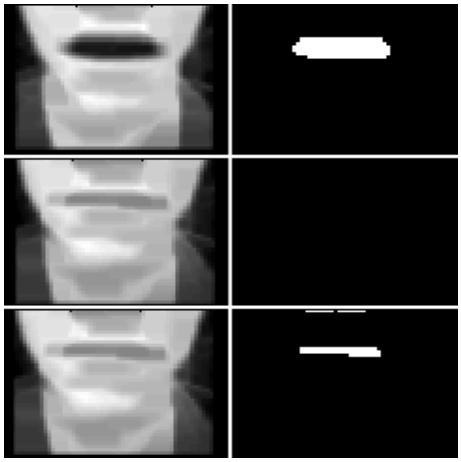

Fig. 11. MS detection: filtered ROI images for MS detection (first column) and respective detection images (second column).

VII. Graphical User Interface

ROCC has a GUI that was designed via Matlab's GUIDE tool. The GUI has three different screens: the opening screen; the initial calibration screen; and the regular operation screen. Those screens are shown in Figure 12.

A. Opening Screen

The opening screen pops up when the GUI is called from Matlab's command window. By pushing the main control button "Begin" the initial calibration is started.

The three vertical slides placed below the main control button are in charge of controlling specific processing parameters. Parameter p_1 affects the size of the cross filter element. Parameter p_2 controls the width of the Gaussian window that is used to match the curve of pixel values (see section V-A). Finally, parameter p_3 affects the sensitivity of MS detection.

The popup menu placed on the bottom-left corner is used to load pre-defined values of parameters p_1, p_2, and p_3 that have been tuned to particular users. The bottom-right button labeled "Change center" is related to the mechanism employed to control the cursor position on the screen.

B. Calibration Screen

Video acquisition initiates soon after the initial calibration is launched. Thus, the user gets a feedback of his/her image taken from the webcam. Moreover, superposed to that image there appears an oval contour. The user is then requested to place his/her face inside the contour, as shown in the top-right image of Figure 12, and then press the main control button, whose label now reads "Matched", to complete the initial calibration.

C. Regular Operation

As seen in the bottom images of Figure 12, the slides for parameter control are enabled as well as the other GUI controls, except from the "Change center" button. Moreover, the detected TNP (red dot) and a circular reference contour (drew in white) appear superposed to the user's image. In the beginning, the center of the contour moves along with the TNP, as the user moves his/her head.

In the sequel, the user is required to settle down in a comfortable position and then open his/her mouth. When the

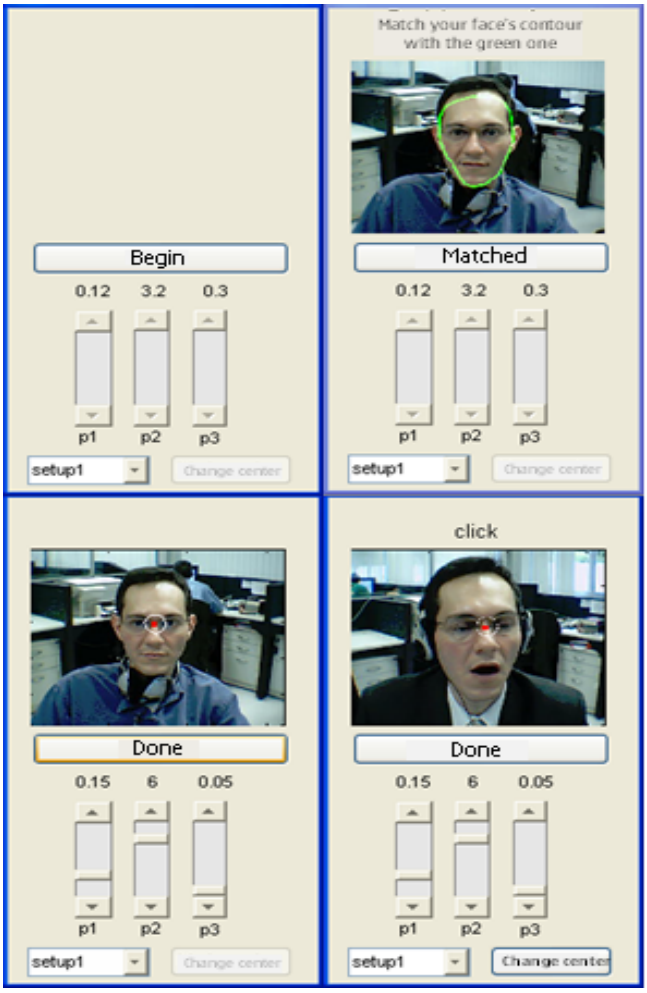

Fig. 12. GUI screens: opening (top-left); initial calibration (top-right); and regular operation (bottom).

MS is detected as open, as seen in the bottom-right image of Figure 12, the position of the circular contour is fixed at the current location. Moreover, the button "Change center" is enabled. From that moment on, the user can move the TNP outside the reference circular contour. The direction and orientation of the TNP when it leaves the contour, respective to the center of the contour, define the direction and orientation of the cursor's movement.

Once the cursor is put in movement, it keeps moving toward the chosen direction until a command for cursor stop or change of direction is launched. The cursor's movement ceases when MS is detected as open. As for changing the direction, the user should bring the TNP back to the resting position, i.e., inside the contour, and then, drag the TNP out of the contour toward the desired direction and orientation. Ideally, the user should keep the TNP always in the resting position, except when sending a command for change of direction.

The user can change the resting position by pressing the button labeled "Change center". Then, the center of the circular contour is dislodged to coincide with the TNP and starts moving along with it again. The new resting position is fixed when the user opens his/her mouth.

If the cursor is in repose, by opening his/her mouth the user selects the object over which the cursor is placed. The selection capability applies to GUI objects and also to other clickable objects in active desktop windows.

Finally, to close the ROCC program, the user can either click on the main control button "Done" or use the default close button "X" on the top-right corner of the GUI.

VIII. Conclusions

This paper presented ROCC, a webcam-mouse system devised for people with physical disabilities. First, an overall system description was provided, covering aspects such as needed hardware and software components, usage conditions to be met, and main functionalities.

In the sequel, the image processing stages employed within the system were described. In particular, a strategy based on skin color selection for segmenting the user's face from the observed video frames was presented. Moreover, a novel proposition for detecting the top of the nose region was introduced. The method was demonstrated to perform as effectively as a competitor procedure on detecting the desired feature, while being simpler computationally. Tracking the selected nose point was employed as a means to control the movement of the cursor on the monitor screen.

As for the click function, it was introduced a method that monitors the mouth region and discerns between two mouth status: with closed or open lips. A change in the mouth status (from closed to open) is used to interrupt the movement of the cursor and select objects on screen.

Finally, the GUI that encapsulates all the processing chain and control handles of the system was detailed. ROCC has been tested by 5 people, two with physical disabilities (paraplegia) and one afro-descendent. All subjects were able to operate the system without problems.

Acknowledgment

The authors would like to thank Ms. Socorro Ferreira and Mrs. Raimunda Santos for the invaluable feedback on the system's usability.

Referências

[1] J. R. Parker, *Algorithms for Image Processing and Computer Vision*, Wiley, 1996.

[2] D. Huber, "Computer vision home page," Webpage Compilation, 2006, http://www.cs.cmu.edu/∼cil/vision.html.

[3] J. Gips, P. Olivieri, and J. J. Tecce, "Direct control of the computer through electrodes placed around the eyes," in *Human-Computer Interaction: Applications and Case Studies*, M. J. Smith and G. Salvendy, Eds., pp. 630–635. Elsevier, 1993.

[4] D. O. Gorodnichy, S. Malik, and G. Roth, "Affordable 3d face tracking using projective vision," in *Proc. Int. Conf. Vision Interface*, May 2002, pp. 383–390.

[5] The MathWorks, "Documentation for mathworks products," Online Publication, 2006, http://www.mathworks.com/access/helpdesk/.

[6] S. Kawato and N. Tetsutani, "Real-time detection of between-the-eyes with a circle frequency filter," in *Proc. 5th Asian Conf. on Computer Vision*, Melbourne, Australia, 2002, pp. 442–447.

[7] J. Terrillon and S. Akamatsu, "Comparative performance of different chrominance spaces for color segmentation and detection of human faces in complex scene images," in *Int. Conf. on Face and Gesture Recognition*, 2000, pp. 54–61.

[8] AT&T Laboratories Cambridge, "The database of faces," website, Retrieved Mar. 2006, http://www.cl.cam.ac.uk/Research/DTG/attarchive/facedatabase.html.