Towards automatic retrieval of blink-based lexicon for persons suffered from brain-stem injury using video cameras

Dmitry O. Gorodnichy
Institute for Information Technology (IIT-ITI)
National Research Council of Canada (NRC-CNRC)
Montreal Rd, M-50, Ottawa, Canada K1A 0R6
dmitry.gorodnichy@nrc-cnrc.gc.ca
www.perceptual-vision.com/blinks

Abstract

Directly connected to the brain, the eyes are the last part of our body we lose control of. For some persons, such as those suffered from a brain-stem stroke, the eyes provide the only means of communication with the world. The eye blinks for such persons are used to make their lexicon and the goal of many rehabilitation centers worldwide is to build tools that would allow automatic detection of the eye blink based lexicon. The tools designed so far are very cumbersome and still do not show the desired performance. At the same time, recent advances in computer hardware and computer vision, in particular, in motion and change detection, offered practitioners a new way for detecting blinks based on video observations of the person's face. This paper overviews different techniques to the problem and describes a vision-based system which is presently being tested in one of the rehabilitation centres. We show how to reliably detect a two-eye blink with a help of an off-the-shelf web-camera and present an approach to the detection a single-eye blink (wink) — this type of blinks is much harder to detect due the lack of spacial constrains, it is however the only type of movement some patients can exhibit.

1 Introduction

1.1 Evolution of perceptual vision systems

As soon asc video-cameras became affordable and computers became powerful enough to process video in real-time, a new application for computer vision research arrived, that of designing *Perceptual Vision Systems* [1]. These are the systems that can see and understand the visual cues of computer users. The ultimate goal in designing these systems is to narrow the gap between how people perceive the world and how computers do. Tracing back the past years, it is fascinating to see how perceptual vision systems have evolved, getting us closer to this goal: from crude face skin tracking based perceptual user interfaces [2], to pixel-precise *Nouse (Use your nose as a mouse)* [3, 4] with *Double-blink* "clicking" [5, 4], to stereo-tracking[6, 4], to perceptual vision systems able to automatically detect and recognize faces [7, 8] and memorize certain facial orientations and expressions [9].

There is considerable interest in this technology. It is worth recalling, for example, following the last year's *CNN Money Magazine* report on computer games [10], that the Sony-made webcam-based *EyeToy*, which uses simple motion detection, was the top selling game in UK for four consecutive weeks! Another very important application for these systems is coming from the security industry. – Video provides one of the most useful types of information, which can be analyzed for the presence on illegal activities, wanted persons etc, and perceptual vision systems can be used as a tool in performing such analysis.

This paper focuses on yet another critical application for this technology, which comes from the industry for disabled. We have realized how critical it is when, after our work on *Nouse* face-operated interfaces, which, at the time, were designed mainly for entertainment purposes and tested on interactive video games, we were approached by several rehabilitation institutions who searched for vision-based technology to enable hands-free computer-human communication for users with special needs [11, 12].

There are many computer users world-wide who, because of their physical condition, cannot use their hands. Some of them are also not able to move their heads. Most of them however have good control of their eyes and therefore communicate with the world solely using the movement of their eyes and eye lids. This is how from using blink detection to simply replace a mouse click we have started thinking about a wider range of using blink detection mechanisms to enable hands-free communication with computers for handicapped users.

1.2 Communication by blinking

Many people who suffered a brain stem stroke are limited to only being able to eye blink. In order to access the equipment in their environments and perform face-to-face communication, these people use eye blinks. For example, a person performs one blink for 'yes' and two for 'no'. There are two types of communication systems: a low-tech system, which uses a grid of letters spoken out by a communication partner with the patient indicating the letter desired to make up words and sentences, and a high-tech system, which directly understands a user's command based on the way s/he blinks, instead of making him/her spell the command a letter by letter.

Because of different ways of blinking, designing a generic blink-detection technique is very difficult. Therefore a communication system is usually designed so as to tailor a specific individual. The following are types of blinks we have to deal with: Type 1 – a patient has good control over both eyelids and can perform consistent two-eye blink, sometimes accompanied by a small observable eyebrow movement and no other observable physical movement; Type 2 – a patient has good control over one eyelid only, closing of which is usually accompanied by a movement of the cheek and/or the brow.

1.3 Solutions to blink detection

Detection of eye blinks for use as control in somebody's environment is a complex task and various technological solutions have been tried to achieve this, but with limited success. One of the most commonly presently used techniques for people with now brain stem injury is the one based on the Electromyograph (EMG) readings, which are obtained by using three small electrodes are attached to the skin with micro-pore tape around the oblicularis oculi muscle [13]. This muscle wraps around the eye and is used in eye closure. Although in principle the EMG-based system should be effective in detecting the muscle signals from patients' eye blinks, it reality it is not. The reasons for this are related to electrical noise in patients environment and various factors concerned with electrode and EMG signal quality. EMG is also liable to interference and increase in the signal to noise ratio, due to skin conductance changes and slight positioning changes of the electrodes. In a patient with brain stem stroke the skin has a tendency to be greasy, which causes variability in the skin conductance. This causes problems with the EMG data, such as base-line drift. The likelihood of the EMG unit working after a shift in position and the electrodes being displaced is very small.

Recently, the new concept of using a webcam for the problem has been applied and the first tests showed promise for some people who have not had success with control and communication through other high-tech solutions [13]. There are several advantages of the webcam-based system compared to the EMG. Firstly there is limited set-up except the placement of the webcam. Also, the webcam would sit at a distance from any patient with minimum intrusion so that nothing needs to be attached to the patient. The webcam is also a cheap, commercially available piece of hardware. One of the foreseen advantages is that if the patient moves position, for example after a spasm, then the webcam simply compensates, by looking at a different part of the picture.

1.4 Vision-based detection of blinks

The approaches to vision-based detection of eye blinks can be divided into three categories. The first category uses dedicated video equipment capable of capturing high-resolution still images [14, 15]. Capturing eyes in high resolution, such that eye pupils diameter is higher than 10 pixels, allows one to extract pupils using Hough-like transforms and template matchingvto decide on whether the eye is open or closed. The disadvantage of this approach is that it requires expensive camera and a lot of processing power.

The second category uses structured infrared light in addition to the regular video camera (e.g. see [16, 17]). These systems register the infrared light reflection in the users' eyes in order to locate the eye pupils and based on whether the pupils are detected or not make the decision on eye lid motion. In certain cases these systems are reported to perform very well. However they have problems detecting eye pupils in the presence of eye glasses and in day light. They are also partially intrusive.

The first and the second categories of approaches do not use the dynamic component of video, which, as discussed in [8, 18] is intensively used in biological vision systems in a similar scenarios. Consequently, there exists a third type of approaches to the eye blink detection, based on detecting the eye lid motion. It is this type we dedicate our effort to.

2 Detecting blinks from motion

In order to recognize blink-based commands from the user using the motion component of video, we developed the following hierarchy of tasks to be executed (see Figure 1). The first task after a video frame has been captured, consists in detecting the visual change caused by facial motion. In order to localize an area in the image where the processing should be done, this task can be preceded by the face detection task. The second task is to analyze the detected change in order to see whether it contains the information about the eye lid motion. If it does, then, based on the eye-lid motion history, determine whether the eye closing and opening was intentional or not, and if it was intentional, then recognize a blink-spelled word.

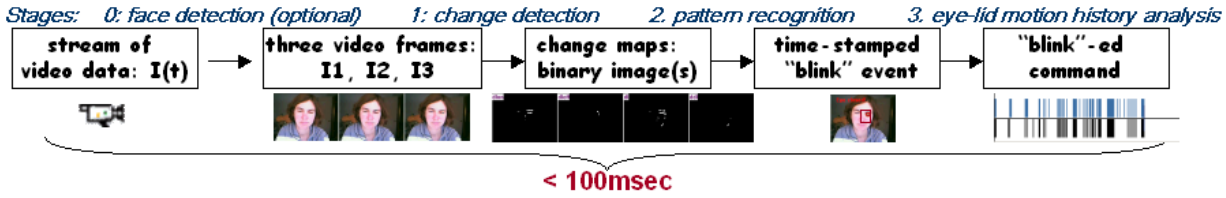

Figure 1: Detecting eye blink-based commands from a user using the motion component of video.

What is important is that all of these tasks have to be performed fast enough so that eye blinks can be captured in real time. In particular, all video processing should be done within 0.1 second, at maximum. Otherwise, unless a user deliberately slows down the blinking motion, blinks may not be captured by a video. With this fact in mind, let us describe now how we approach each of the described tasks.

2.1 Detecting change

A common approach to detecting moving objects in video is based on detecting the intensity change caused by the object motion. The most straightforward way of detecting such a change is to use binarized thresholded difference between two consecutive video frames [19, 20, 21]. This technique however fails to detect eyes when a head moves, because many candidates also appear around the face boundary as well as around nose, mouth and other parts of the face.

Therefore, taking into account that a human head is rarely static, in [5, 4] we introduced the technique for detection of, what we call, the second order change, which is the change in the change and which allows one to distinguish the pixels changed because of the head motion from those changed due to the eye lid motion. It uses three consecutive video frames instead of three and is based on the following idea.

Provided that the frame capture and processing rate is high (around 50 frames per second in our case), the head motion can be considered almost planar, with the velocity $\vec{V}$ of the head in the image plane being practically the same during the entire three-frame sequence. This makes it possible to estimate $\vec{V}$ and to cancel all candidates which have moved by $\vec{V}$ pixels. The result of applying this technique for the detection of a two-eye blink is shown in Figure 2.

Soon after the work [5, 4], it has been realized that the original algorithm does not discriminate positive change (due to eye closing) from negative change (due to eye opening). This extra piece of information, as will be shown later, can robustify the detection of eye blinks. Hence below we present the modification of the algorithm which allows one to obtain this information.

First (see Figure 3), the first order change image dI is computed as a sum of two binarized differences dI^+ and dI^- between the current I^t and the last I^{t-1}frames as

$$\begin{aligned} dI^+_{ij} &= \delta(I^t_{ij} - I^{t-1}_{ij} > T) \\ dI^-_{ij} &= \delta(I^{t-1}_{ij} - I^t_{ij} > T) \\ dI_{ij} &= dI^+_{ij} + dI^-_{ij} \end{aligned} \tag{1}$$

where J_{ij} designates the value of pixel (i, j) in image J, T is a threshold value which takes into account the level of noise in the image and which is taken equal to 20 in our experiments, and δ is Heavyside (step) function.

Then, using the first order change image $dI1$ computed between the last I^{t-1} and the second last I^{t-2} images, a vector $\vec{V} = (v_i, v_j)$ is found such that, by shifting image $dI1$ by (v_i, v_j) number of pixels up/down left/right, it coincides the most with image dI. Then, by shifting image $dI1$ by the thus obtained vector $\vec{V}$ and comparing it with image dI, the first order change image dI is decomposed into two binary images: dI_{long}, which shows long-lasting (global) change such as the one caused by head motion, and ddI, which shows the most recent (local) change such as the one caused by the motion of facial features.

Finally, by subtracting dI_{long} image from images dI^+ and dI^- one can obtain the images which show the most recent second-order changes, which in the case of eye blinking correspond to opening and closing of the eyes. To compensate for the assumption that head motion is planar, we, before performing the last step, dilute dI_{long} a few times using morphological dilation operation so that it covers a larger area around the boundary of the head and other long-lasting motion areas.

The images of facial changes due to eye closing and eye opening computed by the above algorithm are shown in Figures 4, 6 and 7.

2.1.1 On robustness of linear change detection

Computing change images in the above procedure by subtracting one frame from the other is fast, however it may not be very reliable, because of the noise present in image. It also fails when the entire image becomes brighter or darker as when switching the lights on and off.

In order to make change detection more robust, one can use non-linear change detection approaches, such as those

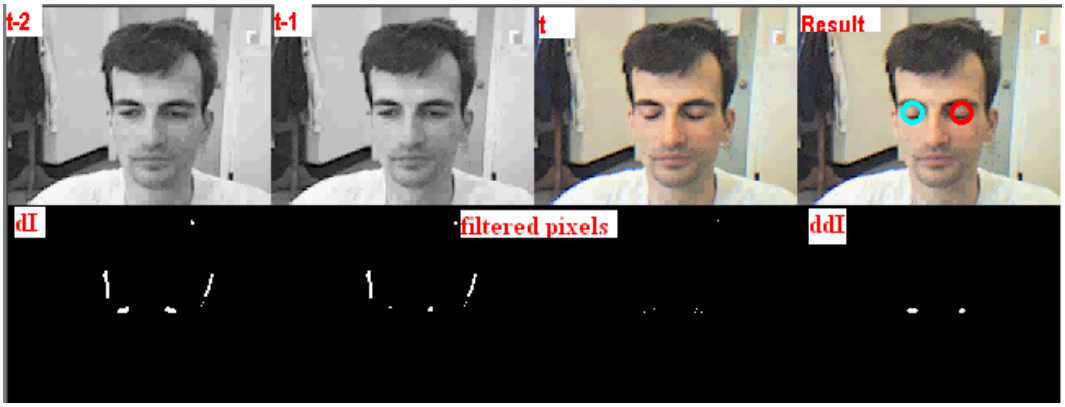

Figure 2: The first order (bottom left image)and the second order change (bottom right image) in the video sequence (three last frames of which are shown in the upper row) which captured an eye blink (from [5, 4]). The pixels which are changed due to the head motion, removed by the second order change detection procedure, are shown at the two bottom center images.

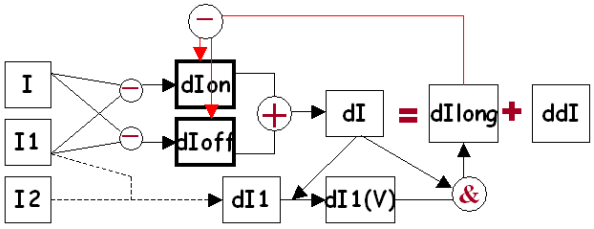

Figure 3: A diagram of the second order change detection: after the global motion vector $\vec{V}$ is estimated, the first order change dI is decomposed into the long-term change dI_{long} and the most recent change ddI. The long-term change is then used to refine the positive dI_{on} and negative dI_{off} changes.

which compute change based on the intensity in the neighbourhood of a pixel rather than on the intensity of the pixel itself [22]; only when the intensity in the neighbourhood changes non-linearly, the pixel is considered changed. This however may significantly slow down the process and therefore should be used with caution. For the present moment, because of this reason, we do not use the non-linear change detection.

2.2 Time consideration

Computing the second order change takes extra processing time of the perceptual vision system, which, as mentioned, risks making the system unsuitable for the application. The longest step in the procedure is finding the shift vector $\vec{V} = (v_i, v_j)$, which is performed by looping through all possible combinations of $v_i = -v_{max}, ..., v_{max}$ and $v_j = -v_{max}, ..., v_{max}$ and comparing an image region in one frame to that in the other for each loop. The larger v_{max}, the larger the amount of long-lasting change can be detected, but also the longer the processing time. Table 1 shows the processing time of the system, running on a laptop which has 1GHz Celeron processor and 256Mb of RAM, as a function of the maximal expected head shift in the image. As can be seen, for this computer only $N = 1$ and $N = 0$ (which implies that no second-order change detection is performed) keep the system running in real-time. For the video camera we use an off-the-shelf web-cam, which is adjusted to capture images at the resolution of 160 x 120. This resolution, as discussed in [9], while being small to allow fast processing, is sufficient enough for many perceptual vision scenarios.

Table 1: Processing time needed for second-order change detection (in miliseconds).

v_{max}	0	1	2	3	4	5
t (msec)	40	80	100	140	180	230

v_{max} is the maximal expected motion of the object in the image (in pixels). The processing is performed on the entire 160 x 120 image.

2.3 Analyzing change images

Recognizing a two-eye blink in the change images is not difficult, because two simultaneously closing and opening eyes provide an excellent spatio-temporal constraint which is observable in the change images as two white blobs separated by a predefined distance. To detect these blobs in the change images, several techniques can be used. For example, in [5] we use two-class vector quantization technique. Another technique which we found very useful for the purpose and which also works for detecting single-eye winks is based on evaluating the moments M_{nm} of the change images, which are defined by Eq. 2 below. The zero-th moment M_{00} shows the number of pixels N changed in the image, M_{01} and M_{10} define the location of mean center

Figure 4: Detecting changes in the face of the user, while she is winking with her left (right in the figure) eye: the first order change image dI, which is made by combining positive dI_{on} and negative dI_{off} change images, is decomposed into a long lasting change (not shown) dI_{long} and the most recent change ddI (shown at right). dI_{long} image can then be used to remove from dI_{on} and dI_{off} images those pixels which have changed due to the head motion.

(X, Y) of the changed pixels, while M_{02}, M_{20} and M_{11} can be used to estimate the size (dX, dY) and the $tilt$ (defined the principle orientation α) of the changed area which can be computed using the following equations:

$$N = M_{00}, \;\; X = \frac{M_{10}}{M_{00}}, \;\; Y = \frac{M_{01}}{M_{00}}$$
$$dX = \sqrt{\frac{M_{20}}{M_{00}}}, \;\; dY = \sqrt{\frac{M_{02}}{M_{00}}}$$
$$tilt = \tan\alpha = \frac{M_{11}}{M_{20} - M_{02}} \quad \text{where}$$
$$M_{mn} = \sum_{i,j} i^m j^n J_{ij} \;\; (m, n = 0, 0; 1, 0; 0, 1)$$

$$M_{mn} = \sum_{i,j} (i - X)^m (j - Y)^n J_{ij}, \;\; (m, n = 0, 2; 2, 0; 1, 1) \tag{2}$$

where J stands for one of the change images: dI, dI^+, dI^- or ddI.

Using these properties of moments, we can detect a two-eye closing and opening by the following condition, applied to negative (for closing) and positive (for opening) change images (see Figure 5):

$$(r < N < r \cdot iod) \textbf{ and } (r < dX < 2iod) \textbf{ and } (dY < r) \textbf{and} (tilt < 0.25), \quad r = iod/4. \tag{3}$$

r in this equation indicates the width of the eye.

As can be seen, this condition does not depend on the location of the face in the image plane, depending only on the intra-ocular distance (iod), which is the function of the distance between the camera and the user and can therefore be set manually in advance.

For the detection of a single-eye wink, we can rewrite the above constraint (Eq.3) as follows

$$(r/2 < N < r \cdot iod) \textbf{ and } (r/2 < dX < iod) \textbf{ and } (dY < iod/2), \quad r = iod/4. \tag{4}$$

Using this condition alone however will almost surely fail to detect an eye wink, because single-blob changes of this size occur very often in video. Therefore an extra constraint on the location of the blinking eye in the image (X_e, Y_e) should be added. This position can be set either manually or by using the automated face detection technique from [7] which is now comes with the OpenCV library. In the latter case, the following formula, obtained empirically, can be used define the position of the eyes:

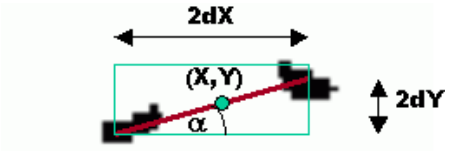

Figure 5: The moments of the change image can be used to detect an eye blink. The red line connects point $(X - dX, Y - dX * tilt)$ and point $(X + dX, Y + dX * tilt)$, where $X, dX, dY, tilt$ are defined by Eq. 2. When a change is caused by the motion of two eye lids, this line connects exactly the locations of the eyes.

$$\begin{aligned} Y_e &= Y_{fd} + W/4 \\ X_e &= X_{fd} + W/4, \quad \text{for one eye, and} \\ X_e &= X_{fd} + 3W/4, \quad \text{for the other,} \end{aligned} \tag{5}$$

where (X_{fd}, Y_{fd}) are the coordinates of the top left corner and W is the width of the square returned by the face detection procedure.

Using the contraint on the eye location is still not sufficient for the detection of a single-eye wink, because there are still some head motions such as, for example, slight head rotation, which can produce the change satisfying the above constraints (see[1] Figure 7). Therefore, in order to circumvent this problem, we propose to compute moments in different overlapping regions of the video frame. In particular, as illustrated in Figure 7, the moments computed in the area covering the entire face can be used to detect the head motion (or any other facial motion), while the moments computed in the vicinity of the eye can indicate the eye lid motion. Other regions such as around the nose, mouth and the

[1]Because of the Privacy Act, the actual images of the people for whom the technology is developed are replaced by the similar images of other people.

other eye, analyzing of which can contribute to the description of the wink, can also be considered.

2.3.1 On robustness of moments

Generally speaking, using moments for the description and recognition of objects in the image is not robust, because the presence of outliers may cause drastic changes to the result. In our case however the outliers, which are the pixels that have changed because of the head motion, are partially removed by the second order change detection. Then by computing the moments in different regions of the video frame one can detect the remaining outliers and thus resolve the problem.

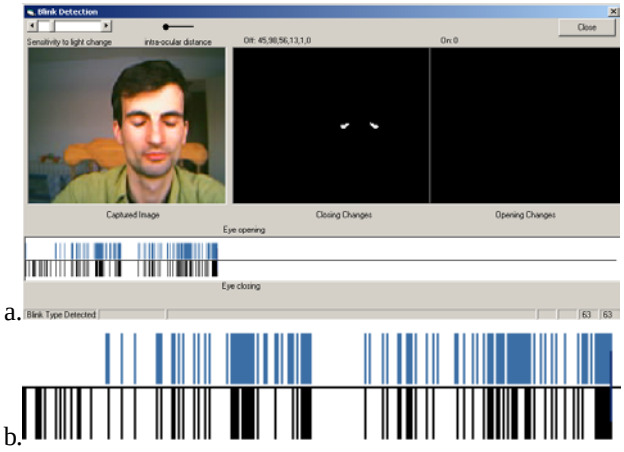

Figure 6: The program written to detect two-eye blinks. The black-and-white images show the positive and negative changes detectable from the video. Every time an eye closing or opening is detected, a vertical bar is drawn on the time line shown at the bottom of the image. The only parameter needed to be set is the intra-ocular distance.

2.4 Analyzing blink-looking events

Despite the techniques described above, the experiments show that there are still some facial motions which the vision system can misclassify as eye closing or opening. Therefore, in order to insure that the detected event is indeed caused by an eye lid motion, we keep track of all eye closings and openings detected by the system, by adding them time-stamped on the time-line. This creates a time signature pattern, one of which is shown in Figure 6; every time an eye closing is detected, a vertical bar is drawn above the time line and every time an eye opening is detected a vertical bar is drawn below the time line, starting from the left (zero time) and moving to the right.

As can be seen from these signatures (Figure 6.b), the detected eye openings and closings can be categorized as being one of the following three types: isolated openings (or closings), many openings and closings happening at the same time, and eye closings immediately followed by eye openings. Naturally, while the first two types of detected changes are most likely contributed to the motion of the head or other facial parts, the third type is a clear indication that there indeed was an eye blink.

Keeping track of eye openings and closings can also be used to evaluate the emotional and/or physiological state of the user, such as the level of fatigue (e.g. see [23]). It also makes it possible to extend the binary (or boolean) way of hands-free communication, containing only ‘yes” and ‘no” commands, to a richer, vocabulary-based communication, in which different commands are decoded using time-stamped blink events.

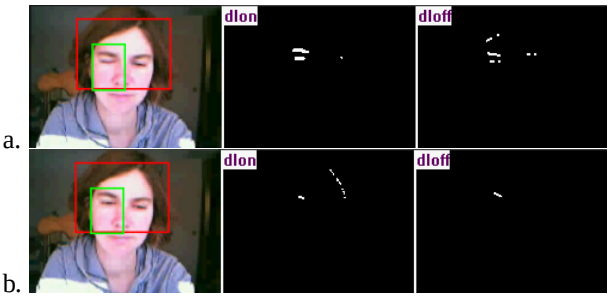

Figure 7: Analyzing different regions of the change images can be used to discriminate a wink from head motion: in (a) a person winks with her left (right in the image) eye, in (b) she slightly moves her head..

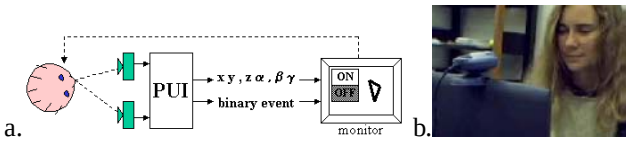

Figure 8: The setup for perceptual vision system most suitable for eye blink detection.

3 Implementation

The described techniques have been implemented as part of the *Perceptual Vision Library* we are developing. The library provides a set of functions which can be used to design a custom-made perceptual vision system. It uses *Microsoft DirectX* [24], *Intel Open CV* [25] and *Intel Image Processing (IPL)* [26] libraries and runs on any computer equipped with a USB or firewire camera.

The library is most suitable for the setup shown in Figure 8: a user sits in front of a video camera mounted on top a computer screen at a distance of not more than half a meter.

A visual feedback, such as a picture of the captured video or a menu which opens up according to the users' facial movements, needs to be provided to the user so that s/he knows whether a hands-free command has been received successfully.

The entire procedure for reading blink-based commands described in the paper, using the library functions calls, can be summarized as follows (in Visual Basic like code):

```
'''initialize camera and set parameters
Sub FormLoad()
    pvsInitializeCamera(160, 120, 1)
    nIOD = 20
End Sub

''' define change caused by eye lid motion
Function EyeLidMoved() As Boolean
    ...
End Function

''' for each captured frame
Sub Timer_Timer()
    pvsInitializeNextFrame()
    nTimestamp = pvsGetFrameNumber()
    pvsComputeChanges()

' detect eye closings by computing N,X,Y,dX,dY,tilt
' in negative change image
    If EyeLidMoved( pvsAnanyzeChange(roi1,dIoff),
                    pvsAnanyzeChange(roi2,dIoff) )
    Then addToList(timeline,bClosing,nTimestamp)

' detect eye openings by computing N,X,Y,dX,dY,tilt
' in positive change image
    If EyeLidMoved( pvsAnanyzeChange(roi1,dIon),
                    pvsAnanyzeChange(roi2,dIon) )
    Then addToList(timeline,bOpening,nTimestamp)

' detect blink by analyzing time-line signatures
' in the last K time-stamped eye lid motions
    If isThereBlink(timeline, K)
    Then ... 'Blink detected!
End Sub
```

The first procedure initializes the camera to capture grey-scale 160x120 images. The second procedure defines a function which uses a set of conditions similar to those in Eqs. 3, 4 to return *true* if the appearance of the changes described by numbers $N, X, Y, dX, dY, tilt$ satisfies the appearance of the changes caused by eye lid motion. In the main procedure, this function is called twice: the first time – to detect eye closings, the second time – to detect eye openings. The final decision on whether a blink occurred or not is made based on the analysis of last K time-stamped detected eye closings and openings. The interface of the program based on this code, which is designed to detect two-eye blinks, is shown in Figure 6. The time line is shown at the bottom.

4 Conclusion

In this paper we studied the problem of detecting computer user's eye blinks using a video camera, with a specific application in mind – that of helping people suffered from brain-stem injury to communicate with a computer by blinking. Although the work on the project is still under development, we believe that the paper have presented the main issues related to the problem as well as the main ways to resolve these issues.

The experiments show that eye blinks become well detectable when the intra-ocular distance is about 20 pixels in the image. Therefore when observing a person's face from a hand-length distance, which is the most frequent case, the resolution of 160x120 suffices for the application. Hence there does not seem to be of any advantage to use higher resolution, thinking that it could improve blink detection, as it very unlikely will. Instead the effort should be put on image processing and pattern recognition parts of problem, such as those defined in the paper, namely: 1) detecting the changes caused by motion, 2) analyzing the detected changes, and 3) combining the obtained results over time. This paper provided an overview of the techniques which can be used for each of these tasks. However, the decision on which particular technique to use in each particular case should be decided based on properties of the individual's eye blink motion and computer power available.

We hope that this work will stimulate more research in this area and will bring the vision-based access to communication technologies, so much needed by many people with special needs, one day closer.

Acknowledgement

This work has been inspired by the last year's visit of the author to the Access to Communication and Technology department of the West Midlands Rehabilitation Centre, Birmingham, UK, which serves the patients, some of which are people who had a brain stem stroke, and communication with Dr. Clive Thursfield.

References

[1] M. Picardi and T. Jan, "Recent advances in computer vision," in *The Industrial Physicist, Vol 9, No 1. Online at http://www.aip.org/tip/INPHFA/vol-9/iss-1/p18.html*, 2003.

[2] G. Bradski, "Computer vision face tracking for use in a perceptual user interface," *Intel Technology Journal*, , no. 2, 1998.

[3] D.O. Gorodnichy, "On importance of nose for face tracking," in *Proc. IEEE Int. Conf. on Automatic Face and Gesture Recognition (FG 2002)*, Washington DC, May 20-21 2002, pp. 188–196.

[4] D.O. Gorodnichy and G. Roth, "Nouse 'Use your nose as a mouse' perceptual vision technology for hands-free games and interfaces," *Image and Video Computing*, vol. In press, 2004.

[5] D.O. Gorodnichy, "Second order change detection, and its application to blink-controlled perceptual interfaces," in *Proc. IASTED Conf. on Visualization, Imaging and Image Processing (VIIP 2003), pp. 140-145, Benalmadena, Spain, Sept.8-10*, 2003.

[6] D.O. Gorodnichy, S. Malik, and G. Roth, "Affordable 3D face tracking using projective vision," in *Proc. Int. Conf. on Vision Interface (VI 2002)*, Online at www.visioninterface.net/vi2002, Calgary, May 2002, pp. 383–390.

[7] G. Shakhnarovich, P. A. Viola, and B. Moghaddam, "A unified learning framework for realtime face detection and classification," in *Int. Conf. on Automatic Face and Gesture Recognition (FG 2002), USA, pp. 10-15*, 2002.

[8] Dmitry O. Gorodnichy, "Facial recognition in video," in *Proc. Int. Conf. on Audio- and Video-Based Biometric Person Authentication (AVBPA'03), LNCS 2688, pp. 505-514, Guildford, UK*, 2003.

[9] Dmitry O. Gorodnichy and Oleg P. Gorodnichy, "Using associative memory principles to enhance perceptual ability of vision systems," in *Proc. of CVPR Workshop on Face Processing in Video (FPIV'04), Washington DC, 2004*.

[10] "Holiday's hottest toy?," in *CNN Money magazine, September 12*, 2003.

[11] "La nariz utilizada como mouse," in *La Nacion Line, Argentina, June 3*, 2003.

[12] "Esperanca contra a excusao," in *Planeta Digital, pp. 4-5, Brasil, August 6*, 2003.

[13] N. Gregory, C. Thursfield, and D.O. Gorodnichy, "Eye blinks for control. Case study," in *Proc. of 7th European Conference for the Advancement of Assistive Technology, Dublin, Ireland, September (ISBN 1 58603 373 5)*, 2003.

[14] Moriyama T., Kanade T., Cohn J.F., Xiao J., Ambadar Z., J. Gao, and H. Imamura, "Automatic recognition of eye blinking in spontaneously occurring behavior," in *Proc. of Intern. Conf. on Pattern Recognition (ICPR 2002), Vol.IV, pp. 78-81, Quebec*, 2001.

[15] Jurgen Rurainsky and Peter Eisert, "Eye center localization using adaptive templates," in *Proc. of CVPR Workshop on Face Processing in Video (FPIV'04), Washington DC, 2004*.

[16] Z. Zhu, Q. Ji, K. Fujimura, and K. Lee, "Combining kalman filtering and mean shift for real time eye tracking under active ir illumination," in *Proc. of Int. Conf. on Pattern Recognition (ICPR 2002), Vol.IV, pp. 318-321, Quebec*, 2001.

[17] Xiaozhou Wei, Zhiwei Zhu, Lijun Yin, and Qiang Ji, "A real-time face tracking and animation system," in *Proc. of CVPR Workshop on Face Processing in Video (FPIV'04), Washington DC, 2004*.

[18] Laurent Itti, "Biological models of vision and attention for face detection in natural scenes," in *Proc. of CVPR Workshop on Face Processing in Video (FPIV'04), Washington DC, 2004*.

[19] J. L. Crowley and F. Berard, "Multi-modal tracking of faces for video communications," in *Proc. CVPR 97*, 1997, pp. 640–645.

[20] L.-P. Bala, K. Talmi, and J. Liu, "Automatic detection and tracking of faces and facial features in video sequences," in *Picture Coding Symposium, Sept. 1997, Berlin, Germany*, 1997.

[21] K. Grauman, M. Betke, J. Gips, and G. Bradski, "Communication via eye blinks detection and duration analysis in real time," in *Proc. CVPR 01*, 2001.

[22] E. Durucan and T. Ebrahimi, "Change detection and background extraction by linear algebra," in *IEEE Proc. on Video Communications and Processing for Third Generation Surveillance Systems, 89(10)*, 2001, pp. 1368–1381.

[23] Ric Heishman, Zoran Duric, and Harry Wechsler, "Using eye region biometrics to reveal affective and cognitive states," in *Proc. of CVPR Workshop on Face Processing in Video (FPIV'04), Washington DC, 2004*.

[24] "Microsoft DirectX 8 SDK," in *http://www.microsoft.com/downloads*.

[25] "Opencv library," in *http://sourceforge.net/projects/opencvlibrary*.

[26] "Intel image processing library (ipl)," in *http://developer.intel.com/software/products/perflib/ijl*.