Perceptual Cursor - A Solution to the Broken Loop Problem in Vision-Based Hands-Free Computer Control Devices

Gorodnichy, Dimitry

For the publisher's version, please access the DOI link below./ Pour consulter la version de l'éditeur, utilisez le lien DOI ci-dessous.

https://doi.org/10.4224/8913964

National Research Council Canada
Conseil national de recherches Canada

Institute for Information Technology
Institut de technologie de l'information

NRC · CNRC

Perceptual Cursor – A Solution to the Broken Loop Problem in Vision-Based Hands-Free Computer Control Devices *

Gorodnichy, D.
February 2006

* published as NRC/ERB-1133. 23 pages. February 2006. NRC 48472.

Canada

National Research Council Canada
Institute for Information Technology

Conseil national de recherches Canada
Institut de technologie de l'information

ERB-1133

NRC·CNRC

Perceptual Cursor – A Solution to the Broken Loop Problem in Vision-Based Hands-Free Computer Control Device

Gorodnichy, D.
February 2006

Canada

NRC 48472

NRC 48472 / ERB-1133
Printed February 27, 2006

Perceptual Cursor – a solution to the broken loop problem in vision-based hands-free computer control devices

Dmitry O. Gorodnichy
Computational Video Group
Institute for Information Technology
National Research Council Canada
Montreal Rd, M-50, Ottawa, Canada K1A 0R6

Abstract

This paper introduces a new concept for the area of user interfaces called Perceptual Cursor. As opposed to the regular cursor, which is by definition an item to mark a position of applied control, Perceptual Cursor also serves a purpose of providing the visual information about the perceived sensory data governing the cursor control. By fulfilling two purposes in one object, Perceptual Cursor eliminates one of the major problems in designing hands-free user interfaces — that of the absence of "touch" or tangible feedback in these interfaces. The paper also introduces a number of propositions related to cursor design that need to be followed when designing a hands-free cursor-controlled system and describes two specific designs for the Perceptual Cursor that are currently being tested with the hands-free addition of the Perceptual Vision Interface *N*ouse.

Contents

1 Presentation of the problem **4**

1.1 On importance of cursor . . . 4

1.2 On importance of feedback in cursor control . . . 4

1.3 Status quo: broken feedback loop . . . 8

2 Mouse-controlled cursor **9**

3 Nouse-controlled cursor. Defining Perceptual Cursor **11**

3.1 Implementation Procedure: . . . 11

3.2 **Control Procedure:** . . . 12

3.3 Example 1. Nouse cursor as a "flying" mini-videoimage . . . 13

3.4 Example 2. Other designs . . . 15

4 Concluding remarks **15**

4.1 On the history of the new concept development . . . 15

4.2 On user study and design issues . . . 17

4.3 On motivation . . . 18

4.4 On future work and prospectives . . . 19

4.5 Addendum: New term . . . 20

5 Acknowledgements **20**

1 Presentation of the problem

1.1 On importance of cursor

With advances made in computer vision, a lot of research effort has been focused recently on designing vision-based Perceptual User Interfaces (PUI), which are the systems that use a videocamera to detect the visual cues of the user, such as the motion of the face, to control a program [25, 9, 22, 19]. Figure 1 illustrates a typical setup for these systems: a USB camera is mounted on top of a computer monitor facing the user who sits in front of the monitor working with the computer.

One of the most important tasks for PUIs is seen in enabling a hands-free control of the cursor. The reason for this is that the cursor, which is normally controlled by hand-operated input devices such as mouse, joystick, track pad and track ball, serves as a prime link between the user and the computer commands the user wants to execute. In particular, the cursor is used to choose the items in Windows menu, position oneself in the desired location in an editor or other window program, highlight a text and objects on a screen, switch focus of attention between the windows, activate and deactivate programs, etc.

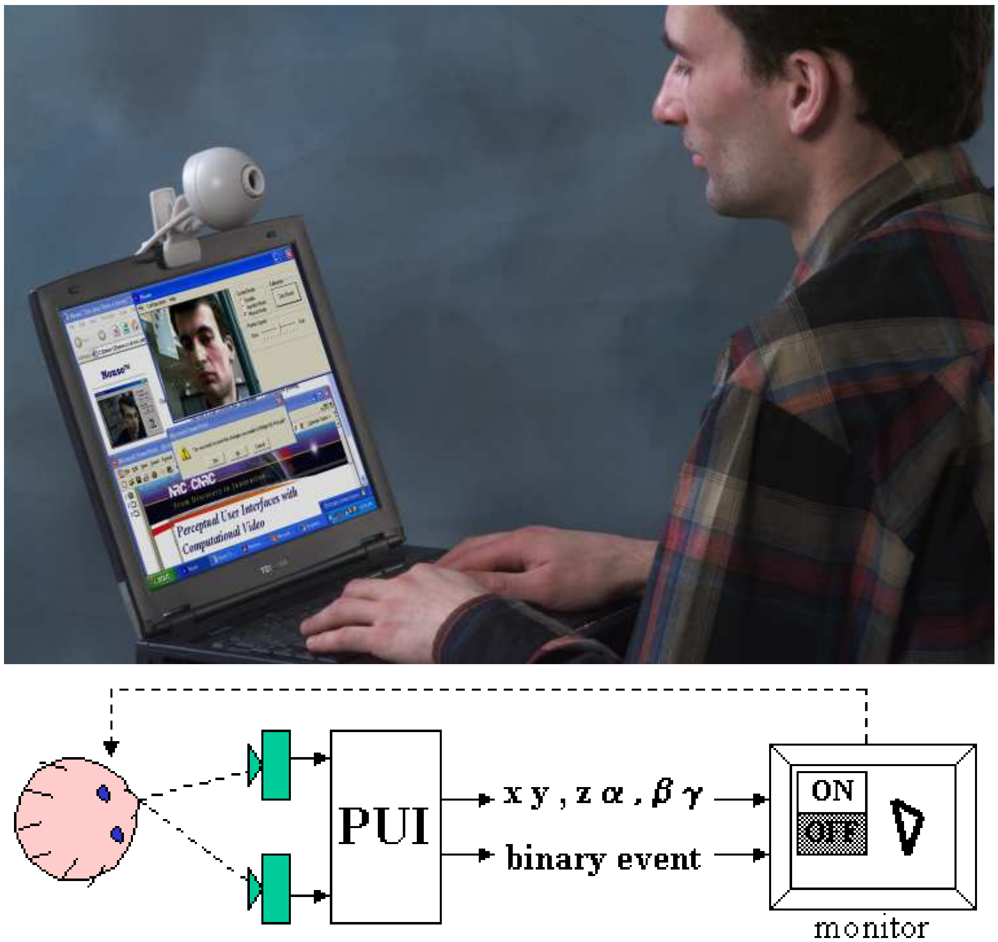
Moving the cursor is also the least constrained and required the least skill technique the user can perform when interacting with a computer. Other, more sophisticated ways of interacting with a computer, such as typing with a keyboard, can be simulated using the cursor motion and the cursor-operated on-screen tools, such as on-screen keyboard.

This explains why for users with hand motion deficiency, the ability to move the cursor with the motion of the face appears to be a long-waited solution for their computer needs, while for all other users hands-free face-motion-controlled cursor is welcomed as an additional degree of computer control much needed when hands are busy with other tasks [1].

1.2 On importance of feedback in cursor control

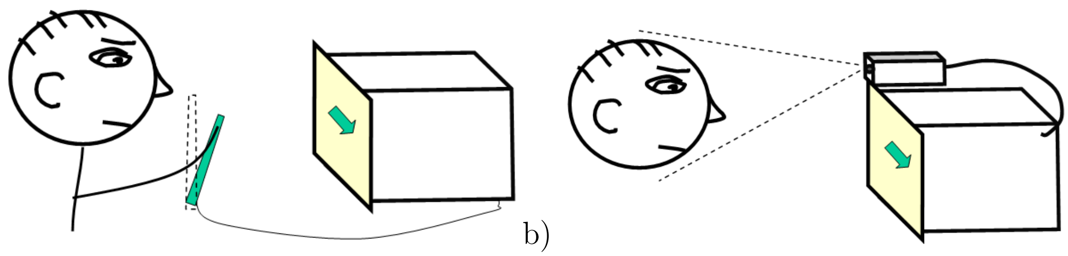
Despite the significant advances recently made in vision-based face tracking, of which we acknowledge the following techniques as the ones contributing the most to the field: developed for automatic detection of faces [26] and eyes[17], 3D-face-model-based tracking [20] and robust sub-pixel precision nose tracking [19] should be specially acknowledged, the desired hands-free face-tracking-based cursor control has not been achieved yet.

The problem is that, while making it possible to receive user commands remotely, PUIs introduce one major problem — the absence of "touch" (also referred to as feedback connection) with the cursor. This problem, illustrated in Figure 2, has a)



b)

Figure 1: Vision-based perceptual user interface: the system at work (a) and the system diagram (b). Photo-picture shows a user at the moment of selecting a "Yes" button in a pop-up "Continue? Yes || No || Cancel" window, which is performed hands-free by moving a cursor with the nose as with mouse and performing a double blink to simulate the mouse button press event.



a) b)

Figure 2: When we hold a mouse/joystick to move the cursor (a), we are constantly aware of where the mouse/joystick is, which makes moving a cursor with a mouse/joystick natural, whereas when we move a cursor remotely, as with a video camera that tracks the person's face position (b), the absence of the constant awareness of how our motion is perceived by the camera makes such a remote cursor control difficult, since the camera may not and often does not perceive our motion the way we think it does, due to light changes, changes to the relative to camera user position, or the deficiency of a vision detection algorithm.

been mentioned back in our first paper on robust vision-based interfaces published in 2002 [19], where we find the following quote: *"One has to realize that, unlike hands-operated interfaces, hands-free interfaces do not have a feedback connection. By holding a mouse, a user not only controls the program, but s/he also keeps the knowledge of where the mouse is. No matter how robust the perceptual user interface is, it can lose the user; it might be even more appropriate to say that a user loses the interface".*

The need for the visual feedback on the performed perceptual commands has been further emphasized in our subsequent work [17], where we find the first drawing of the PUI (reproduced in Figure 1b) that explicitly shows the visual feedback connection and asserts that *"a visual feedback is provided to the user as means of verifying the successfulness of the received face-operated command".*

Indeed, when observed by a camera, a user may only guess (and hope) that the camera and computer perceive his or her commands as he or she intents. But, in fact, due to either light changes, changes in relative camera-user position, or simply because of the deficiency of a vision detection algorithm, the user's face motion may and often does get recognized incorrectly. Such a lack of constant awareness of how our motion is perceived by a computer makes remote cursor control difficult. It can be even further shown the following.

Proposition 1: *What a user needs in order to control a cursor is not a more precise or robust motion detection technique, but a clear real-time closed-loop feedback connection that would allow a user to learn the mapping between the coordinates of the body part controlling the cursor as perceived by the computer and the coordinates of* *the cursor on a screen.*

To prove this a reader is invited to perform a simple experiment, in which, after rotating a mouse 90-120 degrees, the reader is asked to move the cursor to a desired location on a screen. To create a feeling of non-linear mapping between the hand motion and the cursor motion, which inherent to vision-based cursor control, the reader can repeat the experiment using an optical mouse on a surface partially covered with glass. In can then be observed that, while in the beginning of the experiment a user may get confused not knowing how the motion of the hand is converted to the motion of the cursor, after a few runs of the experiment, by simultaneously coordinating the hand motion with the observed cursor motion, the user will likely be able to learn the boundaries of the nonlinear regions of the terrain traversed by the hand as well as trajectories of the hand leading to a desired cursor location.

By rewriting Proposition 1 in a formal manner (see Figure 3), the following corollary can be proved based on the observations obtained in the above described experiment.

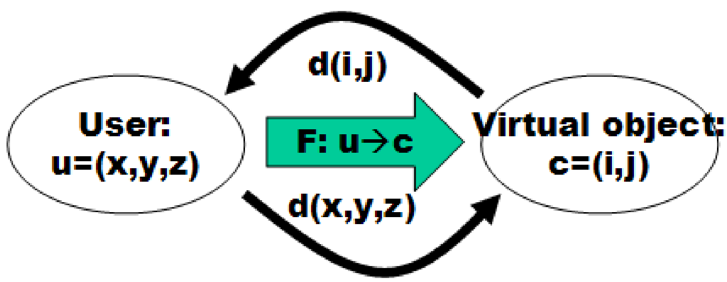


Figure 3: It is the feedback loop that allows a user to learn the motion needed to control a cursor, as the mapping from one coordinate space to another.

Proposition 2: *Provided that the mapping function* Φ*, which maps the user parameters* $\vec{u}$ *to the cursor parameters* $\vec{c}$

$$\Phi : \vec{u}(t) \rightarrow \vec{c}(t) \tag{1}$$

exist and is deterministic, the conditions should be created in PUI for a user to learn it.

The user parameters would normally be made of the Euclidian coordinates of the user's body part used to control a cursor:

$$\vec{u} = (x, y, z),$$

and the cursor parameters are the cursor coordinates on the screen

$$\vec{c} = (i, j).$$

Corollary 1: *The conditions for learning a user-cursor mapping function are met if, at any time t, a user is able to see both the receptor data, i.e. the sensor perceived data $u(t)$, and the affecter data, i.e. the cursor motion data $c(t)$, and, in particular, how the changes in the former affect changes in latter.*

This can be written formally as follows.

Corollary 2: *The mapping function*

$$\vec{c}(t) = \Phi(\vec{u}(t)), \tag{2}$$

is learnt, if for every time instance t and user location $\vec{u}$, derivatives

$$\frac{\delta c_k(t)}{\delta u_l(t)}, \; k = \{i, j\}, \quad l = \{x, y, z\} \tag{3}$$

are known.

As the above experiment shows and as also supported by the theory of neural networks, learning of a mapping function by its Jacobian (i.e. the matrix made of all partial derivatives of Eq.3) is very well performed by humans, provided that the Jacobian is correct at every location at every time instance. However, if this Jacobian is not available or correct, then learning this mapping function becomes very difficult or even impossible. As illustrated in Figure 4 and explained below, this is the case with current PUIs.

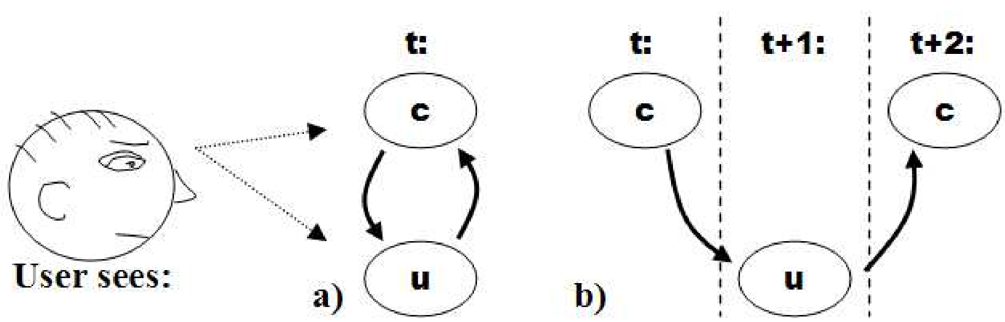


Figure 4: Viewing the affector (cursor) data and receptor (sensor) data alternatively breaks the feedback loop, needed to learn the mapping between the two.

1.3 Status quo: broken feedback loop

As we examine vision-based cursor control systems available on the market [2, 3, 4, 5, 6] or research literature [24, 10, 11, 12, 8, 23, 13, 19, 20] dedicated to designing vision-based perceptual user interfaces (PUIs), we note that in order to provide a user with the knowledge on how the face is detected by the camera, conventional PUIs use a separate window somewhere on a screen in addition to normal cursor, which shows the capture video image of face with the results on vision detection overlaid on top of it[1].

This is the case for the Nouse (the additional window showing the result of nose tracking can be seen in Figure 1a). This is also how the visual feedback is provided for CameraMouse [2] and Quilieye [5], which are two examples of the commercially sold vision-based computer control programs (The documentation for these programs clearly indicates the requirement of the second window which is used to either calibrate the system, verify the results of tracking or both).

The drawback of this visual feedback is that the user has to look both at the cursor (to know where/how to move it, e.g. to open a Windows menu) and at the image showing the results captured by the videocamera (to know how to move his head in order to achieve the desired cursor motion). Since a user cannot view two different locations at the same time, this violates the condition of Corollary 1. That is, instead of observing receptor and affecter data simulteneously as in Figure 4a, a user observes them alternatively as in Figure 4b.

Furthermore, the additional window such as the one shown in Figures 1b produces another problem. It occludes other window applications, making the windows desktop more cluttered and less organized.

In this paper we show that this problem can resolved by introducing a new concept, called Perceptual Cursor, which serves both the purpose of marking a position (as normal cursor) and the purpose of providing a user with the feedback on how remote user motions are perceived by a sensor. As such, Perceptual Cursor does not replace the regular cursor, but rather is used in the interface in addition to it, taking its functionality only when requested by the user. In the following, we present a generic framework for designing the Perceptual Cursor and a particular implementation of it for the vision-based Nouse PUI.

2 Mouse-controlled cursor

For an easier description of the idea, we use the word "Mouse" to refer to any tangible input device, which is an input device that a user have to touch in order to control a computer.

Definition 1: *Cursor is a movable item used to mark a position.*

Controlled by such input devices as mouse, joystick and keyboard, the cursor has

[1] While some references may not mention explicitely the provision of the visual feedback window, the possibity of using it for better performance of the system is always assumed.

become a part of any Windows environment, in which it is defined by its location and the state. Depending on its state and location, the cursor may change its appearance (e.g. shown as a flashing rectangle in an Editor, or shown as an arrow on Windows Desktop). Its purpose however is always the same — *to mark a position.*

The location of the cursor is characterized by two parameters (i, j). Its state is characterized by the mouse events such as ButtonUp and ButtonDown[2]. They can be changed either externally – by a PS2/USB input device such as mouse, or internally – by a Windows application calling such functions as (in pseudo-code notation):

```
mouseEvent(MOVE, X,Y);
mouseEvent(BUTTONDOWN);
mouseEvent(BUTTONUP);
```

(4)

The diagram of the mouse-based cursor control is shown in Figure 5.

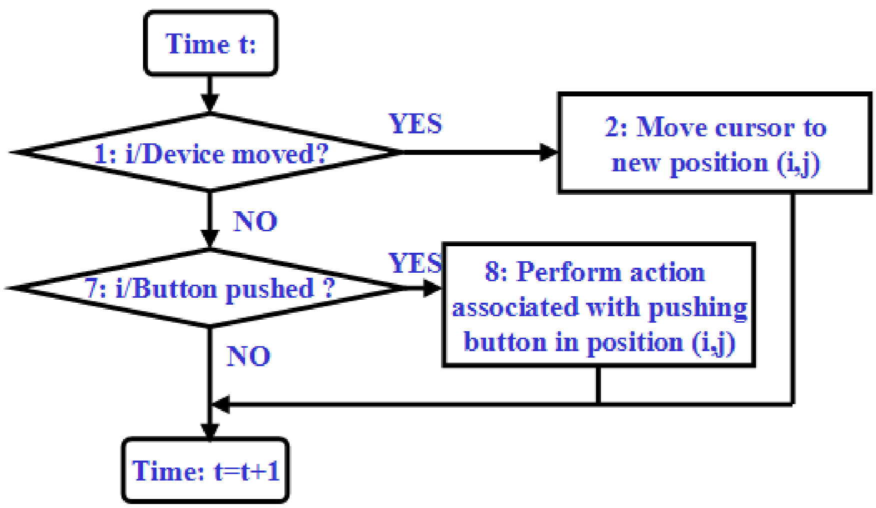


Figure 5: Flowchart of a mouse/joystick-based cursor control.

[2]For the simplicity, we consider two mouse events only, though other events, such as scroll wheel events etc. are also possible.

3 Nouse-controlled cursor. Defining Perceptual Cursor

For an easier description of the idea, we use the term "Nouse" to refer to any hands-free input device, which is an input device that allows a user to control a computer by remote detection of user's motion such as the motion of his/her nose or other part of the body.

Definition 2: *Perceptual Cursor is a cursor that comprises a sensing feedback information.*

In other words, Perceptual Cursor consists in both a visual representation of the sensing video signal and a pointing object, and is used both for pointing and feedback.

The Perceptual Cursor could take on many looks, it could vary in colors, e.g. green when it is tracking; red when it is no longer tracking (as shown in Figure 7; or different size to indicate when the user is moving towards or away from the camera; or different shapes to show other desired feedback features to the user.

When implementing Perceptual Cursor, the following conditions have to be observed.

Condition 1: Perceptual Cursor should be visible at all times[3] and its appearance should be updated in real-time so that the information obtained from the video-processing program can be shown to a user with no delay.

Condition 2: The operation of Perceptual Cursor should not affect or intervene the operation of other window/user interface components, including the normal cursor.

Condition 3: It should be not conspicuous (i.e. not taking undesired attention of the user), yet well visible when desired (i.e. when activated by user).

To meet these conditions, the following procedure can be used.

3.1 Implementation Procedure:

Perceptual Cursor can be implemented as a Windows window

- that is of a small size (not larger than a Windows icon size, which is 32x32 pixels),
- that is made visible at all times, constantly redrawn in Windows on top of all other windows (as the highest priority window),
- the a corner (or middle, or any other highlighted part) of which is used for pointing, and the rest of which shows the desired feedback information.

[3] More exactly, at all times when desired by user, as user may wish to switch it on and off.

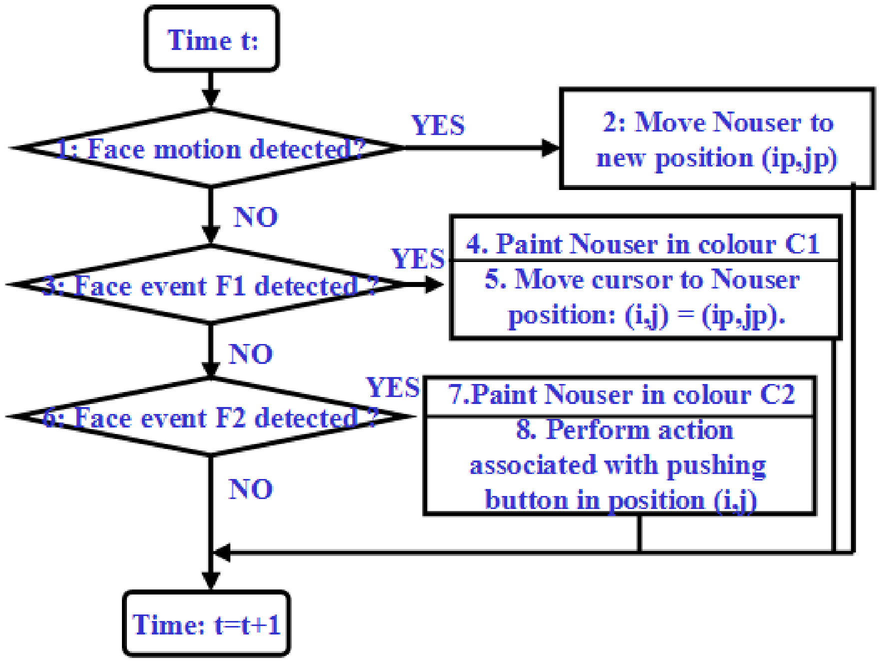


Figure 6: Flowchart of a hands-free Nouse-based cursor control using Perceptual Cursor.

Operating the normal cursor hands-free using the Perceptual Cursor becomes natural with the following control procedure.

3.2 Control Procedure:

- At every time instance, the location and the appearance of the Perceptual Cursor corresponds to the remote input data registered by a sensor.
- In order to transfer control from the Perceptual Cursor to the normal cursor, a set of binary remote events is defined (e.g. facial events "Double-Blink", mouth opening etc. – for a completely remote system, or external switch/keyboard events – for other systems).
- Of these binary remote events, there are at least two that serve the purpose of a) moving the actual cursor to the current location of the Perceptual Cursor, and b) clicking the actual cursor in its current location.
- Additional remote events and specially designed Perceptual Cursor appearance

schemes can be used to facilitate and robustify the transfer of control from Perceptual Cursor to the normal cursor.

The diagram of thus constructed vision-based cursor control using Peceptual Cursor is shown in Figure 6. In addition to steps that are the same for regular cursor operated by mouse/joystick (compare to Figure 5), a few extra steps are added in execution of the program which control the location and appearance of the Perceptual Cursor in such a way as to provide, in a very intuitive way, the pieces of information related to the current tracking status. When the location of the Perceptual Cursor is in the desired position and its state (coded by its appearance) is the state that allows transfer of the control from Perceptual Cursor to cursor, only then the appropriate cursor events, such as (4), are executed.

The choice of colours and shapes for the Perceptual Cursor appearance as a function of the sensed data is a matter of a design, which ideally should be ergonomic and easy to implement. Below we describe two designs developed for the Hands-free edition of the Perceptual Vision Interface Nouse (*Nouse PUI*).

3.3 Example 1. Nouse cursor as a "flying" mini-videoimage

In the Nouse PUI a part of the user face, such as a nose, is used to point to and/or to select items (such as menus) in the Windows OS environment. Such pointing and selection of items is achieved by a two-step process:
1) moving Perceptual Cursor towards a desired menu item, and
2) when Perceptual Cursor is on the desired item, it is activated by the Double-blink or other facial gesture detection mechanism in order to select the desired item.

Due to the fact that facial displacement sensed by a camera is much less than the range of cursor motion, the vision-based cursor control operates the best in a joystick[4] mode [19, 20]. In the joystick mode, the motion and activation of the cursor is achieved by sending the commands from a user's hand via a joystick handle though the wire to the computer. The displacement of the hand x from rest ("zero") position x_0 of the joystick is transferred to the displacement of the cursor on a screen, using the following transfer function:

$$i_{new} = i_{old} + \alpha(x - x_0), \qquad (5)$$

where α is some constant computed based on the knowledge of allowed range of hand motion.

In the vision-based Nouse PUI control, the procedure for the motion and activation of the cursor in the joystick mode is similar to the one with the real joystick, except for the following two very important differences:

[4] Recently a combined multi-modal mode for vision-based cursor control has been proposed, which for simplicity is not considered here.

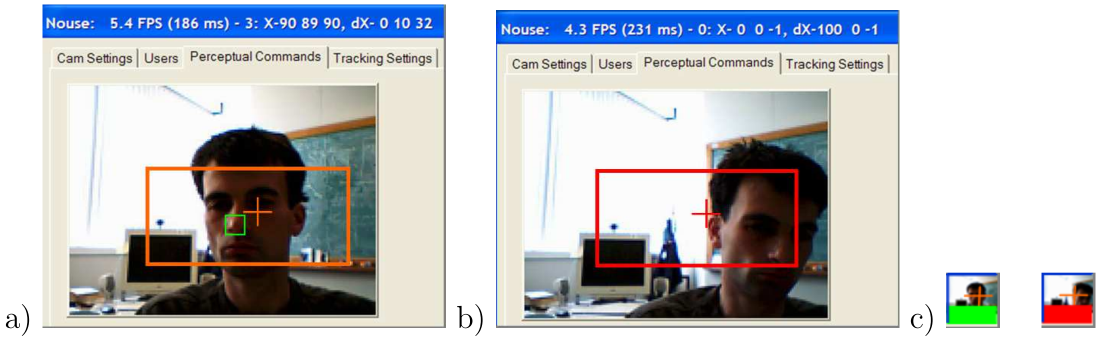


Figure 7: Visual feedback images of two typical results observed in visual tracking of a user's face: face is tracked (a), face is lost (b). This visual feedback Perceptual Cursor allows one to incorporate this visual feedback inside the moving cursor (c).

1. The user does not know (or feel) where the rest (”zero”) position is.

2. The user doesn't know when he is in hold of ”virtual joystick” and when not, i.e. whether he is currently tracked or not, and whether his facial motion (either for moving a cursor or for activating it, as with opening a mouth or double-blinking) is recognized or not.

Figures 7a,b show two common instances happening during the vision-based face tracking. The first figure shows the situation when the nose (or face) is successfully detected and therefore can be used to move the cursor. The second one shows the situation when nose (face) is not detected by the vision system. These figures also show visual information that is important for the user to see when operating with the nose, such as
1) the range of the reliably perceived face motion, which is usually obtained in the calibration procedure and which is used the user-cursor transfer function (Eq.5);
2) the Rest ”zero” position (marked as cross), which is similar to rest position of a joystick. The face displacement left of the ”zero” position results in cursor moving to left, the face displacement right of the ”zero” point results in cursor moving to write, while approaching the boundary of the range motion speeds up the cursor motion, yet may result in losing the facial feature;
3) the current position of the reference point (nose), the relative position of which is used to move the cursor (marked as box when detected); and
4) the colour of the augmentation: red - when not tracking, green - when tracking the face.

All this information has to be incorporated in the Perceptual Cursor appearance. In the design we propose, the Perceptual Cursor has an appearance of a 32x32 pixel box (see Figure 7.c), the left top corner of which is highlighted and used to point, and the inside part of which consists of two parts which show a visual feedback about the current tracking status. The upper part is a 32x24 rectangle used to show a current video image with overlaid cross to indicate the "zero" position. - By looking at this rectangle a user always knows the position of his face with respect to the zero position, which makes it easier for him to control using the face motion.

The lower part of the Perceptual Cursor is a 32 x 8 rectangle (optionally divided into four parts, each of 8x8 pixels size), the colour which (or the colour of its four parts) is used to indicate the current state of tracking: all red - face is not seen(or detected for whatever reason), all green - the program is in tracking mode and runs with no problem. The 8x8 pixel parts can also be used to show any detected facial events: e.g. double blink - highlights two of them in yellow, while four blinks - highlights them in blue and so on.

The example of performing a Windows item selection task hands-free (similar to the task shown in Figure 1a) using Perceptual Cursor is shown in Figure 8: the inner part of it shows the current status of tracking, while the top right corner of it is used to point. After positioning the Perceptual Cursor in the desired location ("Cancel", user Double-blinks once to move the cursor to it, and the second time to click on it.

3.4 Example 2. Other designs

In the second design, the cursor looks like a "live" icon of a joystick with a bar on top of a circle ("Live" meaning that it is alive, changing as you move your face). While the circle serves as a regular cursor, the bar on top of it changes its size/colour/orientation to show the sensing information.

Another design of the Perceptual Cursor, which is not shown in this paper, includes an augmentation of a joystick-looking icon on top of the decimated 32x24 video image. Originated from the "zero" (rest) face position, the live virtual "joystick" icon changes its orientation and size to reflect in real-time motion of the user's face as perceived by the vision system.

Some operating system also allow it to make the Perceptual Cursor transparent, which is an additional convenience for the cases when Perceptual Cursor is used in small-font windows environment. These however, as emphasized in the concluding remarks, do not appear to be critical issues for the concept to work.

4 Concluding remarks

4.1 On the history of the new concept development

It has been envisaged back in 2001 that "*soon most laptops will be equipped with build-in 'eye' (camera) above the screen*" [19]. Indeed, video-cameras (in particular,

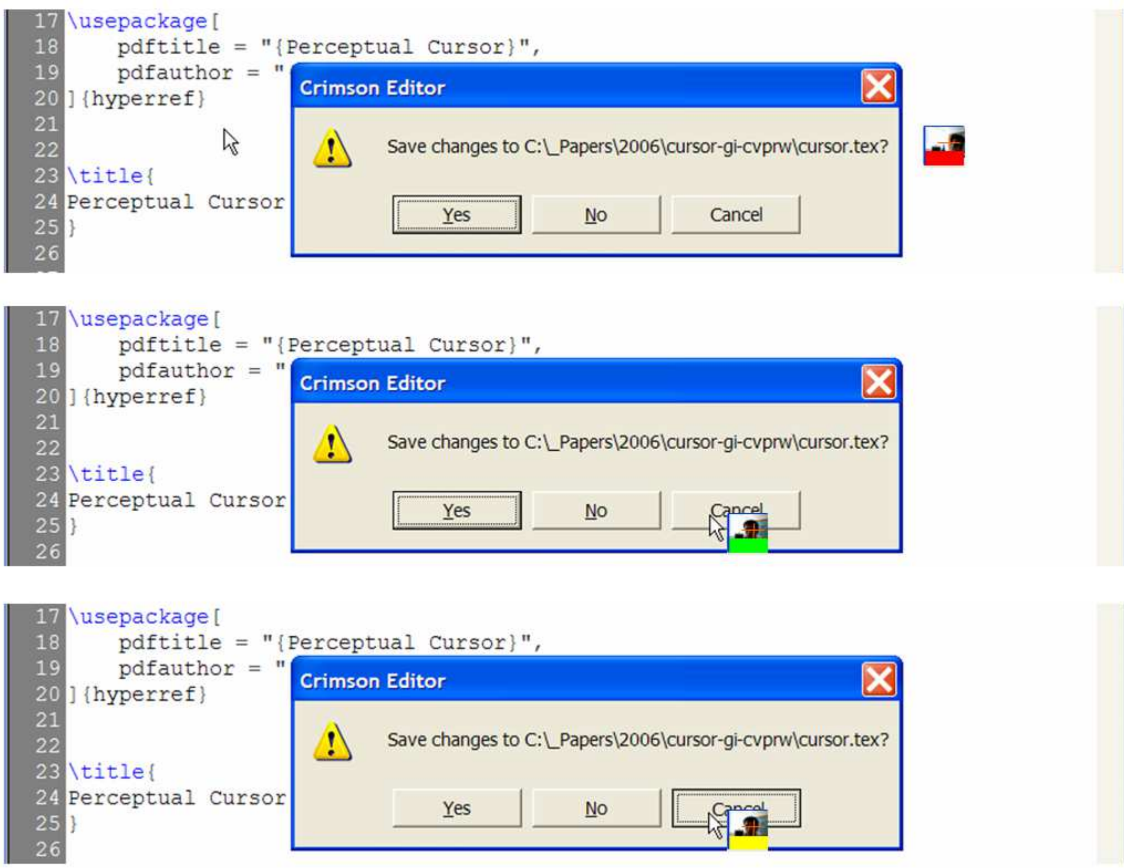


Figure 8: Selecting a Windows item hands-free in vision-based PUI using Perceptual Cursor.

webcams) have become almost as common at computer workspace as a mouse.

There has been also a lot of hope that these cameras would make computer more easier to interact with and would provide users with convenient vision-based hands-free interfaces.

Despite advances in computer vision, such hands-free interfaces however have not yet become as convenient as we wish them to be.

Ironically, what makes current hands-free interfaces (such as based on vision) inconvenient is that they are *hands-free*. That is, a user does not have a feeling or knowledge on where his "grasp" is. This becomes evident to anybody trying to operate something hands-free without a feedback.

While this problem has been identified a while ago (e.g. it is clearly mentioned in cited above paper [19]), the solution to it does not appear to be used or published anywhere until the current presentation.

The idea of our solution to the problem, presented in this paper, has originated soon after the problem has been identified in 2002. Several attempts to implement it have been done in 2002 and 2003, when the first implementations of an always visible flying 32x32-pixel window, which changed the colour depending on the success status of tracking, have been done. Due to computer power limitations and possibly lack of the required programming skills these implementations were not completely real-time though. The first real-time implementation of what is defined in the papers as the Perceptual Cursor has been done in December 2004.

4.2 On user study and design issues

This paper described two particular designs of the Perceptual Cursor that were found convenient for our vision-based hands-free cursor control systems. One is made of a two-part 32x32 window, the top part of which is used to show the visual image and the bottom part of which imitates the multi-coloured buttons which can be activated by face motion. The other one uses a simple stick-on-bun icon which changes its shape and colour according the status of tracking. While being simple to implement, these designs appear to provide enough flexibility to be tailored to an arbitrary vision-based system.

Other designs of Perceptual Cursors may also be suggested – and we leave it to the researchers from HCI (Human-Computer Interaction) community to do investigate the Perceptual Cursors designs that yield highest efficiency and convenience for users. In particular, should there be such an interest, a proper user study on the factors of the Perceptual Cursor design should be conducted, including a Fitt's law study that measures the speed and accuracy of cursor control as a function of Perceptual Cursor attributes.

It is understood however that these studies, while important, are not as critical as the proposed invention itself. Our invention, in essence, is similar to another invention which has become very well-known worldwide, being used in many vendor stores worldwide — the invention that combines a keyboard with a screen, resulting in a touch-sensitive screen. Various designs of such touch-sensitive screen-keyboards have been developed since the new concept has been invented and a suitable technological solution has been described, but these designs are only of minor significance compared to the invention itself

We feel that the situation with the Perceptual Cursor is very similar. The visual appearance of the Perceptual Cursor does not appear to be very critical for the Perceptual Cursor to work as long as it adheres to main principle: **showing the receptor and effecter data in one object**. How to implement the new concept using Visual C++ in Windows OS has been described in this paper too.

4.3 On motivation

To a person normally operating with his or her hands, like most of us are, the work on designing vision-based face-operated interfaces may appear unmotivated and the dedication of our efforts to solve the problem even strange. Why would indeed someone propose replacing hands-operated input devices with hands-free ones and then struggle with the problems "hand-free-iness" brings upon us? Beyond a mere scientific curiosity, of course.

The answer to this question is in those hundreds of emails I have received since our first demonstration of the nose tracking technique (named the Nouse) which allowed a user for the first time ever to write a word with a nose as with pen and a chalk, and the unprecedented media coverage of our face tracking success story around the world. While some of the correspondents might have overestimated the power of the new technology, thinking of it as a panacea of many problems, there has been a very clear indication of what makes this technology so attractive. This is summarized below:

- Its affordability – As in many cases, a person has to spend a lot of money and effort to buy a specially designed gear (such as laser-based or IR equipped trackers) to find out that it does not meet the person's criteria. Besides, each person has to buy his/her own device like this and have it with him/her all the time.

- Its convenience – It does not require a person to wear anything on his/her head, neither it requires any special tuning for each person. This makes it possible to use the same camera-computer setup and room for many people. Places such as Internet cafes for people with special needs can be thus created.

- Its performance – Many people, who have constrained mobility of their hands, do not require the full range of computer control commands and find the performance exhibited by Nouse sufficient for their needs, provided that it can be incorporated into their programs, the most important of which are controlled by a cursor movement.

As a result, several municipal and provincial Canadian health organizations have contacted IIT-NRC over the last three years expressing their interest in hands-free vision-based computer control that the Nouse technology makes possible, and the working relationship has been established with the Elisabeth Bruyere Research Institute and SCO Health Service in Ottawa.

It is having in mind these concrete people – the residents of these health institutions, for whom our technology would cater the most – how the technological problems have been discovered and the solutions to these problems have been researched [21]. Despite a long history of building hands-free technology for disabled, some of the technological problems and solutions that we have discovered, such as the one presented in this paper, still have not been reported in literature.

4.4 On future work and prospectives

It is clear that Perceptual Cursor may also be used to facilitate hands-free control of any, not necessarily vision-based, perceptual user interfaces.

At the same time, it is also understood that providing the missing feedback is not the only problem that needs to be resolved for vision-based cursor systems to work.

For example, there is another problem that seems not to be tackled by Computer Vision researchers. This problem is related to the head-tracking-based cursor control and deals with the fact that the range of head displacement a user can exhibit is much less than the range of hand-operated pointing devices such as mouse or a joystick. This problem is even serious when considering it for people with disabilities who are the prime consumers-to-be of the technology in question.

In our research we have addressed this problem too and a solution to it, based on a new multi-modal space transformation technique that converts the constrained facial position to that of the cursor, will be the focus of our next publication.

Finally, as described in our earlier publication on building user-seeing computers [7], besides cursor control, there are several other important computer vision tasks that need to be resolved to make hands-free computer control viable and efficient, such as automatic identification of users (so that proper user setting can be selected hands-free), facial event recognition such as Blink detection (for mouse-click replication), hands-free typing using an on-screen keyboard suitable for low a range motion, etc.

While many of these tasks have been already successfully resolved by our efforts [17, 14, 16, 19, 7, 15, 18], combining all of them into a single runnable end-user software which would seamlessly integrate into an Operating System requires a different from the computer-vision/pattern recognition set of skills. This means that in order to build such a software a team with all the diverse skills may be required.

4.5 Addendum: New term

To conclude the paper, we also propose another term for the new user interface concept. The term Perceptual Cursor, used throughout the paper, captures well the gist of the concept as applied to the perceptual user interfaces. When applied to vision-based perceptual user interfaces such as those tracking the user's nose (or a face), another term for it is proposed – *Nousor*, which obtained by blending two other words: *Nouse* + *Cursor*, both of which can now be found in English dictionary:

Cursor – noun [C] /kur-ser, -sor/
a movable item used to mark a position: as a visual cue (as a flashing rectangle) on a video display that indicates position (as for data entry) (From Merriam-Webster Online Dictionary: www.m-w.com/dictionary/cursor).

Nouse – noun [C] /naus/
a pointing mechanism for a personal computer which is activated by movements of the nose. (From Macmillan English Dictionary: www.macmillandictionary.com/New-Words/041004-nouse.htm).

5 Acknowledgements

The results presented in this paper have been partially inspired and refined by the author as a result of his communication with the Elisabeth Bruyere Research Institute and SCO Health Service of Ottawa, with the particular contacts with Bocar N'Diaye very much appreciated. The first implementation of the Perceptual Cursor as a 32x32 pixel window which is always visible on top of all other Windows applications (done in the fall of 2002) was made with a help of Steve McDonald. The real-time version of Perceptual Cursor (done in December of 2004) was done with help of Gilles Bessens, who has also proposed and demonstrated the transparent implementation of it. The partial funding for this project from the IIT-NRC DG reserve fund of 2004 is also acknowledged. Finally, the author acknowledges the fact that the material presented in this report has been originally submitted on December 19, 2005 for publication at the HCI track of the Graphics Interface conference, where it has been rejected for the reason of the lack of clear motivation and the lack of user study; the received HCI reviewers' comments however were much appreciated and the answers to them are partially incorporated in the current publication.

References

[1] In *New Scientist, Breaking News Report. Nose-steered mouse could save aching arms, September 16, 2004 (online link). Also CNN, Special Report. Follow your nose, September 30, 2004 (online link), and Discovery Channel, The Nose Knows , December 7, 2004 (online link)*. 4

[2] In *CameraMouse website: http://www.cameramouse.com/product.htm*. 8, 9

[3] In *SmartEye website: http://www.smarteye.se/demo.html*. 8

[4] In *MouseVision website: http://www.mousevision.com/assistech/html/products/handieye.htm*. 8

[5] In *QualiEye website: http://www.qualisoftware.com/moreinfo.php?id=4*. 8, 9

[6] In *NaturalPoint website: http://www.naturalpoint.com/trackir/products/howitworks*. 8

[7] Gilles Bessens and Dmitry O. Gorodnichy. Towards building user seeing computers. In *Second Workshop on Face Processing in Video (FPiV-05) jointly with AI/GI/CRV'05, 9-11 May 2005, Victoria, BC, Canada, 2005. NRC 48210*, 2005. 19

[8] M. Betke, J. Gips, and P. Fleming. The camera mouse: Visual tracking of body features to provide computer access for people with severe disabilities. In *IEEE Trans Neural Syst Rehabil Eng., 10(1):110*, 2002. 8

[9] G. Bradski. OpenCV: Examples of use and new applications in stereo, recognition and tracking. In *Proc. Intern. Conf. on Vision Interface (VI 2002) (online at http://www.cipprs.org/vi2002)*, page 347, Calgary, May 2002. 4

[10] V. Chauhan and T. Morris. Face and feature tracking for cursor control. In *12th Scandinavian Conference on Image Analysis*, 2001. 8

[11] T. Darrel, N. Checka, A. Oh, and L.-P. Morency. Exploring vision-based interfaces: How to use your head in dual pointing tasks. In *Technical Report AIM-2002-001, Artificial Intelligence Laboratorty@MIT*, 2002. 8

[12] J. W. Davis and S. Vaks. A perceptual user interface for recognizing head gesture acknowledgements. In *Workshop for Perceptive User Interface, pages 17, Orlando, FL*, 2001. 8

[13] L. El-Afifi, M. Karaki, and J. Korban. Hand-free interface- a fast and accurate tracking procedure for real time human computer interaction. In *FEA Student Conference, American University of Beirut*, 2004. 8

[14] Dmitry O. Gorodnichy. Towards automatic retrieval of blink-based lexicon for persons suffered from brain-stem injury using video cameras. In *In CD-ROM Proc. of First IEEE CVPR Workshop on Face Processing in Video (FPIV'04), Washington DC, USA, NRC 47139*, 2004. 19

[15] Dmitry O. Gorodnichy and Gilles Bessens. From recognition in brain to recognition in perceptual vision systems. case study: identifying computer users with low-resolution webcams. In *Poster at CVR'05 Conference on Computational Vision in Neural and Machine Systems. York University, Toronto, June 15-18, 2005. NRC 48208*, 2005. 19

[16] Dmitry O. Gorodnichy and Oleg P. Gorodnichy. Using associative memory principles to enhance perceptual ability of vision systems. In *In CD-ROM Proc. of First IEEE CVPR Workshop on Face Processing in Video (FPIV'04), Washington DC, USA, NRC 47139*, 2004. 19

[17] 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. 4, 6, 19

[18] D.O. Gorodnichy. Seeing faces in video by computers. Editorial for special issue on face processing in video sequences. *Image and Video Computing*, 24(5):1–6, 2006. 19

[19] D.O. Gorodnichy and G. Roth. Nouse 'Use your nose as a mouse' perceptual vision technology for hands-free games and interfaces. In *Image and Video Computing, Volume 22, Issue 12 , Pages 931-942, NRC 47140*, 2004. 4, 6, 8, 13, 15, 17, 19

[20] J.Tu, T.Huang, and H.Tao. Face as mouse through visual face tracking. In *Second Workshop on Face Processing in Video (FPiV'05). In Proc. of CVR'05, ISBN 0-7695-2319-6*, 2005. 4, 8, 13

[21] NRC-CNRC. *Nouse* (Nose as Mouse) perceptual interfaces technology. http://synapse.vit.iit.nrc.ca/Nouse, 2001. 19

[22] M. Picardi and T. Jan. Recent advances in computer vision. In *The Industrial Physicist, Vol 9, No 1. Online at (online)*, 2003. 4

[23] R. Ruddarraju and et al. Perceptual user interfaces using vision-based eye tracking. In *5th international conference on Multimodal interfaces, pages 227 233, Vancouver, British Columbia, Canada*, 2003. 8

[24] K. Toyama. Look, ma-no hands! hands-free cursor control with real-time 3d face tracking. In *Workshop on Perceptual User Interfaces, San Fransisco*, 1998. 8

[25] M. Turk, C. Hu, R. Feris, F. Lashkari, and A. Beall. TLA based face tracking. In *Proc. of Intern. Conf. on Vision Interface (VI 2002), (online at http://www.cipprs.org/vi2002)*, pages 229–235, 2002. 4

[26] Paul Viola and Michael Jones. Rapid object detection using a boosted cascade of simple features. In *CVPR 2001. Also "Robust real-time face detection", In International Journal of Computer Vision, 57 (2), 137-154, 2004.* 4