Nouse 'Use Your *N*ose as a M*ouse*' – a New Technology for Hands-free Games and Interfaces

D.O. Gorodnichy [1]*, S. Malik [2], G. Roth [1]
[1] Computational Video Group, IIT, National Research Council, Ottawa, Canada K1A 0R6
[2] School of Computer Science, Carleton University, Ottawa, Canada, K1S 5B6
http://www.cv.iit.nrc.ca/research/Nouse

Abstract

With the invention of fast USB interfaces and recent increase of computer power and decrease of camera cost, it has become very common to see a camera on top of a computer monitor. Vision-based games and interfaces however are still not common, even despite the realization of the benefits vision could bring: hand-free control, multiple-user interaction etc. The reason for this lies in the inability to track human faces in video both precisely and robustly. This paper describes a face tracking technique based on tracking a convex-shape nose feature which resolves this problem. The technique has been successfully applied to interactive computer games and perceptual user interfaces. These results are presented.

Keywords: HCI, perceptual user interfaces, face tracking, feature detection, shape from shading, evidence theory.

1 Introduction

1.1 Perceptual user interfaces

We consider the problem of tracking faces using a video camera and focus our attention on the design of vision-based perceptual user interfaces. These are the systems which use a videocamera to track user's face position in 3D in order to convert it to a position of a cursor or another virtual object in 2D screen. They are aimed at providing a hands-free alternative to mouse, joystick, track pad or track ball. Let us list a few applications of such systems.

First, face-tracking-based program control can be seen as a replacement for an inconvenient track stick or track ball currently used in many laptops. It can also be seen as an additional way of interfacing with a computer, which can be used, for example, to switch a focus of attention in windows environment. Our presentiment is that soon most laptops will be equipped with build-in 'eye' (camera) above the screen. This application therefore will be very useful.

Second, as mentioned in [3], vision-based perceptual user interfaces can be used to control commercial computer games and immersive 3D worlds. In addition to being hands-free, these interfaces provide a way for multiple-user interaction – several users can be tracked at the same time with several cameras.

Third, as described by [19], this technology has applications in the industry for disabled. Users who have difficulty using a standard mouse could manipulate an on-screen cursor by moving their head.

Finally, face tracking has applications in video-coding, video-conferencing, content-based image retrieval and security industry.

1.2 Previous work

The mentioned applications require face tracking to be fast, affordable and, most importantly, precise and robust. In particular, the precision should be sufficient to control a cursor, while the robustness should be high enough to allow a user the convenience and the flexibility of head motion.

A few hardware companies have developed hands-free mouse replacements. In particular, in accessibility community, several companies developed products which can track a head both accurately and reliably. These products however either use dedicated software or use structured environment (e.g. markings on the user's face) to simplify the tracking process.

At the same time, recent advances in hardware, invention of fast USB and USB2 interfaces, falling camera prices, and increase of computer power brought a lot of attention to the real-time face tracking problem from the computer vision community. The obtained vision-based solutions though still do not exhibit required precision and robustness. Let us review these solutions.

The approaches to vision-based face tracking can be divided into two classes: image-based and feature-based approaches [4, 13, 11, 23]. Image-based approaches use global facial cues such as skin colour, head geometry and motion. They are robust to head rotation and scale and do not re-

*The author the correspondence should be sent to.

quire high quality images. These approaches however lack precision and therefore can not be used to control the cursor precisely.

In order to achieve precise and smooth face tracking, feature-based approaches are used [2, 18, 6, 15, 17, 21, 19, 5, 22, 24]. These approaches are based on tracking individual facial features. These features can be tracked with pixel accuracy, which allows one to convert their positions to the cursor position. The disadvantage of these approaches however is that they usually require expensive high-resolution cameras. They are also not robust to the head motion, especially to head rotation and scale. This is the reason why vision-based games and interfaces are still not common.

Recently it has been shown that the robustness of local feature tracking can be significantly improved if instead of commonly used edge-based features such as corners and edges of brows, mouth and nostrils, curvature-based features such as the tip of the nose are used [7]. This finding has set the basis for the *Nouse 'Use your Nose as a Mouse'* technology developed at the National Research Council of Canada, which allows one to track faces both robustly and precisely with affordable web-cameras. This paper describes the technology and shows how to use it for designing hands-free nose-operated games and interfaces.

The paper is organized as follows. Section 2 emphasizes the importance of appropriate feature selection for successful tracking, introduces the concept of a convex-shape feature and defines the nose feature. Section 3 describes the *Nouse* face tracking technique. The performance evaluation of this technique and its application to interactive games and interfaces are shown in Section 4. The last section presents the conclusion.

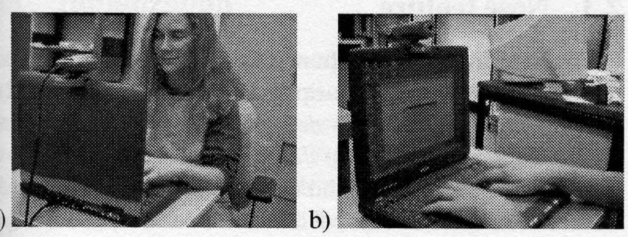

Figure 1: A user mounts a camera to control computer applications with her head.

2 Tracking and features

In feature-based approaches for face tracking, tracking of a feature is commonly accomplished by means of template matching technique, which consists of scanning a window of interest with the peephole mask and comparing each thus obtained feature vector with the template vector [8]. The peephole mask and the values of the template vector are learnt in advance and do not change during the tracking procedure.

In order to achieve robust and precise tracking, we introduce the following propositions.

Proposition 1: *One feature only should be used for the final decision on the head position in video.*

This eliminates the jitter problem which arises when tracking several features due to the fact that facial features move independently from each other. This also provides a user with an intuitive way of controlling with the head motion by simply visualizing the feature as a tip of a joystick or mouse ball.

Proposition 2: *The tracked feature should always be clearly visible for all face positions and expressions, including the cases with the users who wear eyeglasses, mustaches or beards.*

It follows from these propositions that the problem of robust tracking is the problem of constructing such a feature template vector which stays invariant during the motion a user may exhibit during the tracking.

2.1 Desired feature properties

In image processing, a feature is often thought of as a point on the object surface which has large change of intensity gradient in the image. This explains why most commonly used in face tracking facial features are corners and edges of mouth, brows, nostrils and eye pupils. These edge-based features however create a problem in tracking the objects which may rotate, since these features are not invariant to the rotation and the change of scale of the object.

In order to select a robust facial feature, we use the pattern recognition paradigm of treating features. According to this paradigm, a feature is associated with a vector made of feature attributes. Feature attributes can be pixel intensities or they can be the parameters of geometric primitives. In the case of template-based feature tracking, feature attributes are the intensity values obtained by centering a peephole mask on the position of the feature.

For a facial feature to be easily recognized and robustly tracked it should possess the uniqueness and robustness properties, defined as follows.

The *uniqueness property* states that a feature vector should lie as far as possible from other vectors in multi-dimensional space of feature attributes. Designating a vector obtained by centering a peephole mask on pixel $u(i,j)$ as $\vec{V}_u$, this can be written as

$$\rho(V_f, V_u) \to max. \quad (1)$$

The *robustness property*, on the other hand, states that a feature vector should not change much during the tracking process. That is

$$\rho(V_f^{t=0}, V_f^t) \to min. \quad (2)$$

These two properties define what is called the *attraction radius* of the feature, which is the largest distance from within which a feature is guaranteed to be correctly recognized [10]. The larger the attraction radius of a feature, the more robust the feature.

In addition, we require a feature to have the *continuity property* which is the following. The closer pixel u in an image is to the pixel corresponding to feature f, the smaller the distance between vector $\vec{V}_u$ and feature vector $\vec{V}_f$ should be.

The continuity property is very important for precise and smooth tracking, as it allows one to robustly obtain the position of the feature in an image with the sub-pixel accuracy. In order to do this, the evidence theory is employed [20], and the refined position of the feature $\hat{u}$ is calculated using the evidence-based convolution filter applied in the vicinity Ω of the best match pixel u, as

$$\hat{u} = \sum_{k \in \Omega} \omega_f^k u^k, \tag{3}$$

where weights ω_f^k are set proportional to the correlation between vector $\vec{V}_u^k$ and the template vector $\vec{V}_f$. It can be seen that, due to the continuity property, in the vicinity of the feature these weights are monotonically decreasing with the distance to the feature, and thus applied convolution filter refines the position of the feature.

Imposing the continuity property and using the evidence-based filter is similar to the approach taken in [8] where the adaptive logic network (ALN) [1] is used to detect eye pupils. In that work the continuity is achieved by selecting the output scheme for learning the eye features and it is the ALN that does the filtering.

An example of the feature which possesses the continuity property is shown in the sections.

2.2 Convex-shape features

Definition: *Convex-shape features are those points on the surface of the object which are seen by the camera as the points of same curvature during the motion of the object.*

It is known from the shape-from-shading theory [12] that for a surface of constant albedo the intensity of pixel $u(i,j)$, where a surface point, is projected is related to the gradients (p,q) of the surface as

$$I(i,j) = aR(p,q) \tag{4}$$

where R is the reflectance map, which depends on the position of the light source, the observer, and the type of surface material, and a is a constant that depends on the surface albedo and the gain of the imaging system. For Lambertian surfaces, reflectance R is proportional to the cosine of the angle ϕ between the surface normal and the incident ray

$$I(i,j) \sim a cos(\phi). \tag{5}$$

Since any convex surface can be approximated using the Taylor series as a sphere around the surface extremum, we have the following properties for convex objects of homogeneous colour with curvature radii much smaller than the distance to the camera and to the light source.

Corollary 1: *The intensity values around the extremum of a convex curvature stay invariant during rotations of the surface.*

Corollary 2: *In the image of a convex surface the intensity value of a pixel $I(i,j)$ is a continuous function of pixel coordinates (i,j).*

First corollary asserts that convex-shape features possess the robustness property, while from the second corollary it follows that convex-shape features also possess the continuity property, provided that feature attributes are made of pixel intensities around the feature.

As for the uniqueness property, convex-shape features cannot be assumed to have it. Therefore extra measures have to be undertaken to facilitate their recognition, such as using global image-based cues (see e.g. [16]).

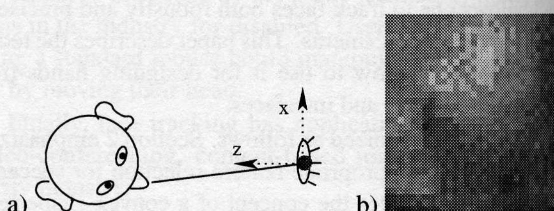

Figure 2: The nose seen by a camera: system of coordinates (a), the intensity values around the nose tip (b).

2.3 Nose feature

Fortunately for humans, a human face has a salient convex-shape feature, which is the tip of the nose (see Figure 2.2).

Definition: *Nose feature is defined as the point on the nose surface that is the closest to the camera.*

Due to the symmetry and the convex shape of the nose, the nose feature is always visible in the camera, and it stays almost the same during the rotations of the head. It also does not change much with head moving towards and from the camera. Thus, the nose tip defined above can always be located, which is a very important property of the nose which does not hold for any other facial feature. It gives a user the flexibility and convenience of head motions.

It should be noted that the nose feature, defined as above, is not associated with a particular point on a nose. Instead, it is associated with the extremum of the nose curvature, which *moves* on the nose surface. The smoothness of the nose feature motion on the image plane is guaranteed by the continuity property of convex-shape features.

As follows from the previous section, the intensity pattern around the tip of the nose is not affected much by the orientation of the nose and the distance from the nose to the camera. This helps in selecting the feature attributes to be used in designing the nose feature template. In particular, it follows from the above that the nose template vector can be made by using the gray intensities around the extremum of the nose surface. The size of the template vector is chosen so that it preserves the information about the spherical surface around the tip of the nose.

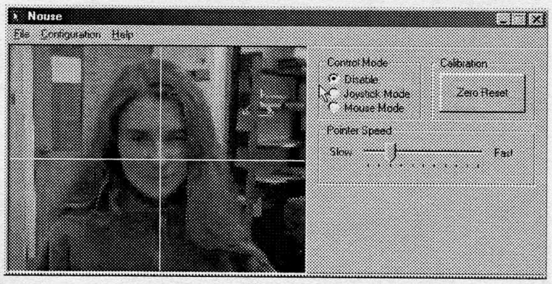

Figure 3: *Nouse* user interface at the initialization stage.

3 The *Nouse* technology

Nouse stands for *"Use your nose as a mouse"* and is the name of the technology which allows one to operate the computer with the motion of his/her nose. The *Nouse* technology is based on the principles and definitions made in the previous section. It consists of two stages: initialization and tracking.

3.1 Initialization

The following setup is necessary for operating a computer with the *Nouse* face tracking technology. A user sits comfortably in front of the computer monitor, on top of which is mounted a video camera (see Figure 1). The distance from the camera to the user is such that user's hands are on the keyboard at all times, with the keyboard being located close to the monitor. While operating with the nose, the user is expected to look at the computer screen; however, there are no constraints for the head motion. The camera is not assumed to be calibrated, high-resolution or aligned with the computer. Further more, we will assume that the camera is a generic USB camera, one of those which are widely available on the market.

Nouse requires an initialization procedure which, first, ensures that user's face is seen by the camera, and second, sets the zero position of *Nouse*. Zero position is used in tracking to calculate the offsets needed for operating *Nouse* in the mouse and joystick modes.

Figure 3 shows the *Nouse* user interface at the initialization stage. Initialization is done by adjusting the camera so that the tip of the nose is seen in the middle of the image. When this adjustment is achieved, the template of nose, which shows the position of the nose surface with respect to the light source, is acquired. Under the same lighting condition, the same nose template can be used for different users. However initializing *Nouse* for each user makes it more robust.

3.2 Tracking

Hands-free control is achieved by tracking the convex-shape nose feature in video stream captured by the camera. The entire procedure is outlined below.

Since the convex-shape features may not possess the uniqueness property, the first step is to calculate the area for local search of the nose feature using skin colour, frame subtraction and the knowledge of the feature position in the previous frame, if available.

The second step consists in scanning the local search area and finding pixel u which has the shape the closest to that of the selected feature in terms of correlation with the template feature. Before proceeding to this step, the image is preprocessed with the Gaussian filter to smooth the defects of images caused by low quality of the cameras.

The final step is to refine the position of the best match, using the evidence-based convolution filter described in Section 2. The obtained match of the convex-shape feature is both robust (to rotation and scale) and precise (computed with sub-pixel accuracy).

3.3 Mouse and joystick modes

The *Nouse* technology allows a user to use their nose position to control the Windows cursor in one of two control modes. Joystick mode operates in a fashion similar to an analog joystick: offsetting the nose from the center of the screen causes the mouse cursor to move in a similar direction. The speed of the cursor is determined based on the offset amount.

Mouse mode attempts to mimic an actual mouse more closely: offsetting the nose from the center position causes the cursor to move in a similar direction, but the movement of the nose back to the center position has no effect. This allows the user to simulate the common continuous dragging process that computer users frequently perform. For example, when users runs out of drag space on their mouse pad, they lift the mouse back to the center of the pad and continue dragging. This allows precise positioning of the cursor and thus *Nouse* attempts to mimic this process.

4 Performance evaluation

In computer vision and feature tracking, in particular, it is very difficult sometimes to evaluate the performance of a new technique. One reason for this is the difficulty in reproducing the exactly the same video setup. The other reason is that it is often impossible to recreate all possible environments and conditions, where the technique can be applied.

In our work, we strive to provide such an evaluation. This is done in three ways. First, we apply the proposed technique to specific applications where its advantages can be clearly seen. The applications of using nose instead of a joystick and a mouse in computer games and hand-free user interfaces are presented in the next subsection. Second, we promote public evaluation of the technique by making the binary code of the used programs available on our website. Thanks to the affordability of USB webcams, anybody can try it. Finally, we evaluate the performance by conducting an extensive set of experiments, and, while the paper shows only a few snapshops of these experiments, full videos of real-time performance are also made available on our website.

The tests are aimed at evaluating the robustness, precision and also the convenience of the nose-based tracking. Due to the minimalistic nature of tracking only one feature and because fast USB cameras are used, the speed of tracking, which is problem of many tracking approaches, is not a problem in our experiments. The tests have been conducted with different USB cameras, different people and different lighting conditions.

4.1 Robustness test

For the robustness test, a user is asked to move the head in all possible rotations while still looking at the computer screen: "yes"(up-down), "no"(left-right), and "don't know" (clockwise) motions and also "scale" (body) motion, and the range of head motion within which the convex-shape nose tip feature is successfully tracked is determined. Figure 4 shows the allowable range of head motion as obtained by experiments. The nose tip is tracked for up to about 35-40 degrees of rotation of head in all three direction. This range of allowable motion practically covers the range a user may exhibit while looking at the screen.

4.2 Precision test

For the precision and convenience tests, a paint program *NousePaint* has been designed which draws the detected nose tip feature on the top of the image (Figure 5). A user is asked to write with the nose, thinking of the nose as of a pencil, while another user is asked to guess what is being written. This motion is equivalent to the motion of the mouse ball when drawing with the mouse.

a)
b)
c)
d)

Figure 4: The figure shows the range of motion within which the nose feature is tracked.

The speed of writing can be as fast as user is able to handle. For example (see Figure 5.a and our website), it took 25 seconds to write with the nose the logo for the *Nouse* home webpage, which included time needed for clicking "Enable/Disable Drawing" buttons (seen on the right of the window) with a mouse.

For another set of experiments, starting from the normal head position, a user is asked to draw horizontal and vertical lines by rotating the head only ("Yes" and "No" motions) and then to make a circular motion with the head. The motion of the nose tip in this experiment can be considered as an equivalent to a joystick handle motion. The result is shown in Figure 5.b. The subpixel accuracy and the continuity of tracking can be seen.

The experiments showed that, due to the robustness and precision of the convex-shape feature tracking, a user can move the head the way it makes him comfortable. This makes it possible for a user to write quite complex words and patterns hands-free without having a neck fatigue.

a) b)

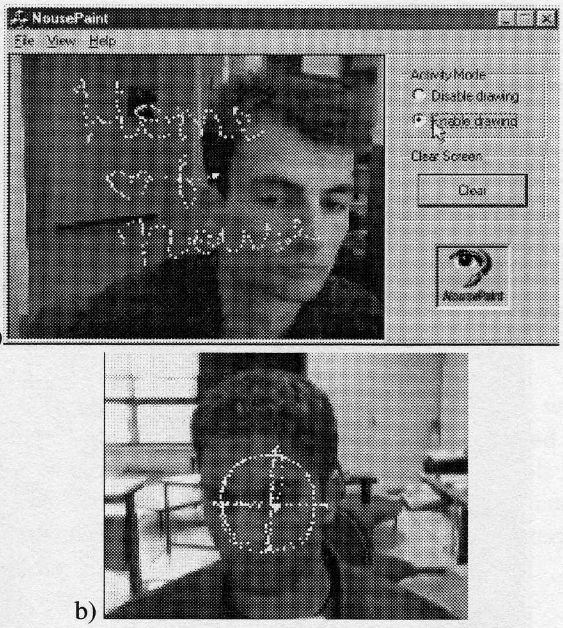

Figure 5: Tests for nose-operated interfaces: a) mouse-type motion, b) joystick-type motion.

4.3 Applications

The *Nouse* technology allows one to track faces both precisely and robustly even in low-resolution video streams such as those coming from low-quality USB web-cameras. This makes it possible to build affordable vision-based user interfaces, which can be used for hand-free vision-based control of computer programs.

Nouse has been applied for navigating in Windows environment (see Figure 1.b). It has been found that the robustness and precision of *Nouse* allowed a user to select an item in Windows menu using the motion of the nose. However, because the size of the computer screen (in pixels) is much larger than the size of the image in which the nose moves, it takes considerable amount of time and energy for the user to get to the desired item. Thus, another application was considered to emphasize the advantage of hand-free program control offered by the *Nouse* technology.

Since video games are an extremely popular application on personal computers, we tested the applicability of *Nouse* technology to gaming by incorporating the technology into a few popular games.

4.3.1 Aim-n-shoot *Bubble Frenzy* game

This game traditionally involves simple left/right mouse movements or key presses in order to aim a bubble turret in the desired direction (see Figure 6). Once properly targeted, the user then presses the space bar to launch a coloured bubble in the turret direction. By matching three or more bubbles of the same colour together, they fall out of the play area. The goal is to clear the screen of the existing bubbles without filling the play area completely.

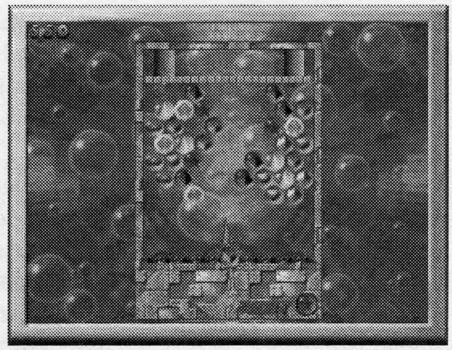

Figure 6: Traditionally aimed with mouse or keyboard, the turret in *Bubble Frenzy* game can now be aimed by pointing with the nose.

We used *Nouse* to replace mouse, allowing a player to point the direction of shooting with the nose. It should be emphasized that the precision of *Nouse* is such that very slight rotations of head left and right are sufficient to cover the entire 180^o range of the turret aim.

The users of the game have a choice of switching back and forth from *Nouse* operation to mouse operation, which helped them to evaluate the new hand-free interface technology. There were more than fifty people who tried the game. It has been agreed that playing the game hands-free with *Nouse* is not only more fun, but is also less tiring than playing the game with mouse. Some users experienced severe wrist fatigue when they played the game with mouse for longer than 15 minutes. This does not happen with *Nouse*. Using the nose to aim the turret was found very natural, while the precision of aiming with the nose was as good as with mouse.

If a camera loses the nose, which happens, for instance, when a user leaves the desk and then comes back, a user notices it right away – by seeing that the bubble turret no longer follows the direction of the nose. S/he then presses a keyboard button which enforces *Nouse* to search for the nose using the global cues of the entire image, after which *Nouse* redetects the nose and the user can continue playing the game with the nose.

4.3.2 Navigating in Virtual 3D Worlds

The *Bubble Frenzy* game uses *Nouse* in a joystick mode. The applicability of *Nouse* in mouse mode is shown using a navigation game. For this case, a user has to navigate in a virtual 3D terrain environment. The usual way of navigating in such games is using different key buttons to look right/left, up/down and go forward/backward. The *Nouse* technology provides a more natural way of navigation. In particular, rotating the head left or right causes the first-person 3D view to also rotate left or right. By using *Nouse*'s mouse-mode, a user is able to rotate a full 360 degrees without physically rotating the same amount in his or her chair. Similarly, rotating the head up or down causes the 3D view to look up or down.

Translation is still accomplished by other means, but the applicability of the technology is clearly evident. It has been observed that many game players commonly exhibit slight head movements when playing first-person 3D shooters such as *Quake* or *Unreal*. Further, these natural head movements commonly occur when users are sneaking around dark corners, peering over ledges, or dodging bullets. This shows the potential of applying the *Nouse* technology to 3D games if applied in the proper context.

4.3.3 *NousePong*

The *NousePong* game has been written to demonstrate another advantage of vision-based interfaces, which is multiple user interaction. Two web-cameras are mounted to face opponents heads as shown in Figure 7. Because of the robustness of the convex-shape nose feature to rotation, the cameras do not have to be aligned with each other. The game consists in bouncing the ball back and forth over a virtual table using the head. Each camera tracks the motion of a player's head in order to convert it to the motion of the paddle.

The *NousePong* game showed that with the aid of the *Nouse* technology computer games may become not only the games for brains and eyes but also the games which actually involve the motion for the body. For example, a feature which can be implemented to add more physicality to the *NousePong* game would be to spin the ball with the motion of head.

4.4 Mistracking problem

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. This does happen with *Nouse* too.

The only way to resolve this problem is to provide a user with a missing feedback. In the case of *Nouse*, the best feedback a user can get is the image captured by the video camera. By visually verifying that s/he is in the middle of the image and that the nose is tracked correctly, a user can ensure that s/he has a good 'grasp' of *Nouse*. This is done in *NousePaint* nose-drawing program and also in *NousePong* game, where the images captured with both cameras are shown on the screen at all times (Figure 7.b).

a)

b)

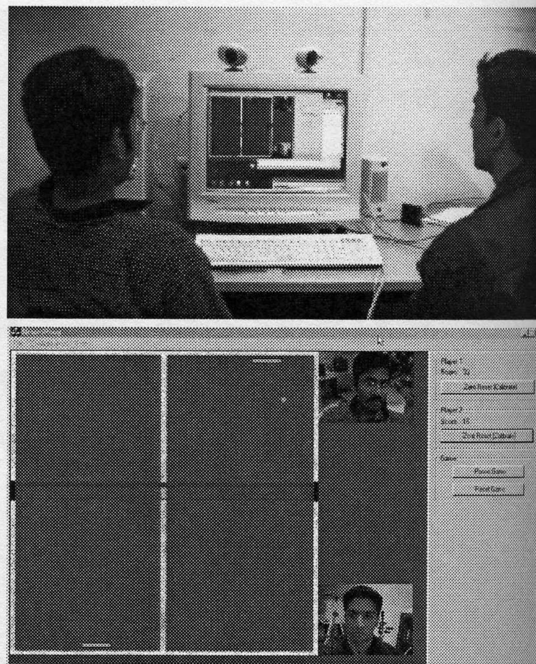

Figure 7: Two users are playing a pong game using their heads to bounce a ball: a) setup and b) GUI.

Another way of having the desired feedback is to know where the camera is, and as soon as a user feels that s/he lost control with *Nouse*, s/he puts his face in front of the camera and sends a signal to *Nouse* (e.g. by using a keyboard) to reset the search of the nose. This method is used in *Bubble Frenzy* and navigation games.

5 Conclusion

Because of the low prices and the ease of installation, USB cameras have become very popular nowadays. However, up till now these cameras are barely used for anything more intelligent than web-casting and video-surveillance. Automatic motion detection is considered almost the top high-level vision task which can be performed using these cameras. This paper demonstrates that USB cameras can also be used for designing vision-based perceptual user interfaces.

We described *Nouse* – a new technology for face tracking based on tracking the convex-shape nose feature. We showed theoretically and by extensive experiments that this technology can be used to operate computer hands-free. It exhibits robustness and precision which are sufficient for many applications and which allow a user the flexibility and convenience of motions.

The *Nouse* technology can also be used for tracking several faces at the same time using several cameras. All this makes this technology a promising hands-free alternative and/or extension to conventional pointing devices such as mouse, joystick and track ball. All programs described in the paper are available at our website and the best way to appreciate the new technology is to download and to try it yourself.

The next step in applying the *Nouse* technology is extending it to 3D face tracking. Most computers have two USB slots. So, two USB cameras can be easily used to track a face. As we show in our next work [9], with the recent advances in the projective vision theory, it is possible to do robust and precise stereo tracking with such uncalibrated stereo setups as those made by two arbitrarily positioned USB cameras.

Finally, being a new technology for communicating with the computer, the *Nouse* technology needs to be studied by Human Computer Interaction (HCI) methods [14]. This includes collection of extensive experimental data and comparing the technology with other human-computer interaction devices.

Acknowledgements

Bubble Frenzy game, which was used to test the *Nouse* technology, has been supplied by *Extended Reality Ltd.* and can be downloaded from *http://www.extendedreality.com.*

References

[1] W. Armstrong and M. Thomas. Adaptive logic networks. *Handbook of Neural Computation, Section C1.8, IOP Publishing and Oxford U. Press, ISBN 0 7503 0312 3*, 1996.

[2] S. Baluja and D. Pomerleau. Non-intrusive gaze tracking using artificial neural networks. Technical Report CMU//CS-94-102, CMU, January 1994.

[3] G. Bradski. Computer vision face tracking for use in a perceptual user interface. *Intel Technology Journal*, (2), 1998.

[4] R. Brunelli and T.Poggio. Face recognition: Features versus templates. *IEEE Trans. on Pattern Analysis and Machine Intelligence, 15(10), pp. 1042-1052*, 1993.

[5] A. Colmenarez, B. Frey, and T. Huang. Detection and tracking of faces and facial features. In *ICIP proceedings*, 1999.

[6] A. Gee and R. Cipolla. Fast visual tracking by temporal consensus. *Image and Vision Computing*, 14(2):105–114, 1996.

[7] D. Gorodnichy. On importance of nose for face tracking. In *Proc. Intern. Conf. on Automatic Face and Gesture Recognition (FG'2002)*, Washington, D.C., May 2002.

[8] D. Gorodnichy, W. Armstrong, and X. Li. Adaptive logic networks for facial feature detection. In *Proc. Intern. Conf. on Image Analysis and Processing (ICIAP'97), Vol. II (LNCS, Vol. 1311), pp. 332-339, Springer*, 1997.

[9] D. Gorodnichy, S. Malik, and G. Roth. Affordable 3D face tracking using projective vision. In *Proc. Intern. Conf. on Vision Interface (VI'2002)*, Calgary, May 2002.

[10] D. Gorodnichy and A. Reznik. Increasing attraction of pseudo-inverse autoassociative networks. *Neural Processing Letters*, 5(2):123–127, 1997.

[11] E. Hjelmas and B. K. Low. Large receptive fields for optic flow detectors in humans. *Computer Vision and Image Understanding*, 83:236 – 274, 2001.

[12] B. K. P. Horn. Understanding image intensities. *Artificial Intelligence, Vol. 8, pp 201-231*, 1977.

[13] R. Hutchinson and W. Welsh. Comparison of neural networks and conventional techniques for feature location in facial images. In *First IEE International Conference on ANN*, pages 201–205, October 1989.

[14] J.-F. Lapointe and P. Boulanger. Live virtual reality system for the control and monitoring of space operations. In *I-SAIRAS 2001, St-Hubert, 7 pages. NRC 44885*, 2002.

[15] G. Loy, E. Holden, and R. Owens. 3D head tracker for an automatic lipreading system. In *Proc. Australian Conf. on Robotics and Automation (ACRA2000)*, Melbourne, Australia, August 2000.

[16] R.Feraud, O. andJ. E. Viallet, and M. Collobert. A fast and accurate face detector based on neural network. *IEEE Transactions on Pattern Anaylis and Machine Intelligence*, 23(1), 2001.

[17] R.Newman, Y. Matsumoto, S. Rougeaux, and A. Zelinsky. Real-time stereo tracking for head pose and gaze estimation. In *Proc. IEEE Intern. Conf. on Automatic Face and Gesture Recognition Gesture Recognition (FG2000)*, 2000.

[18] L. D. Silva, K. Aizawa, and M. Hatori. Detection and tracking of facial features. In *SPIE Visual Communications and Image Processing'95 (VCIP'95), Vol. 2501, pp. 2501/1161–2501/1172*, 1995.

[19] K. Toyama. Look, Ma – no hands! Hands-free cursor control with real-time 3D face tracking. In *Proc. Workshop on Perceptual User Interfaces (PUI'98)*, San Fransisco, November 1998.

[20] F. Voorbraak. Reasoning with uncertainty in AI. In *Reasoning with Uncertainty in Robotics (RUR'95) Intern. Workshop proceedings*, pages 52–90, 1995.

[21] M. Xu and T. Akatsuka. Detecting head pose from stereo image sequence for active face recognition. In *Proc. IEEE Intern. Conf. on Automatic Face and Gesture Recognition (FG'98)*, 1998.

[22] J. Yang, R. Stiefelhagen, U. Meier, and A. Waibel. Real-time face and facial feature tracking and applications. In *Proc. AVSP'98, pages 79–84, Terrigal, Australia*, 1998.

[23] M. Yang, N. Ahuja, and D. Kriegman. Detecting faces in images: A survey. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 24(1):34–58, 2002.

[24] K. Yow and R. Cipolla. Finding initial estimates of human face location. In *In Proc. 2nd Asian Conf. on Comp. Vision, volume 3, pages 514–518, Singapore*, 1995.