Available online at www.sciencedirect.com

ScienceDirect

Computer Vision and Image Understanding 108 (2007) 35–40

Computer Vision and Image Understanding

www.elsevier.com/locate/cviu

Face as mouse through visual face tracking

Jilin Tu [a,*], Hai Tao [b], Thomas Huang [a]

[a] *Electrical and Computer Engineering Department, University of Illinois at Urbana and Champaign Urbana, IL 61801, USA*
[b] *Department of Computer Engineering, University of California at Santa Cruz, Santa Cruz, CA 95064, USA*

Received 7 September 2005; accepted 21 November 2006
Available online 12 January 2007
Communicated by Mathias Kolsch

Abstract

This paper introduces a novel camera mouse driven by visual face tracking based on a 3D model. As the camera becomes standard configuration for personal computers (PCs) and computation speed increases, achieving human–machine interaction through visual face tracking becomes a feasible solution to hands-free control. Human facial movements can be broken down into rigid motions, such as rotation and translation, and non-rigid motions such as opening, closing, and stretching of the mouth. First, we describe our face tracking system which can robustly and accurately retrieve these motion parameters from videos in real time [H. Tao, T. Huang, Explanation-based facial motion tracking using a piecewise Bezier volume deformation model, in: Proceedings of IEEE Computer Vision and Pattern Recognition, vol. 1, 1999, pp. 611–617]. The retrieved (rigid) motion parameters can be employed to navigate the mouse cursor; the detection of mouth (non-rigid) motions triggers mouse events in the operating system. Three mouse control modes are investigated and their usability is compared. Experiments in the Windows XP environment verify the convenience of our camera mouse in hands-free control. This technology can be an alternative input option for people with hand and speech disability, as well as for futuristic vision-based games and interfaces.

Keywords: Camera mouse; Visual face tracking; Hand-free control; Perceptual user interface

1. Introduction

Researchers have long been speculating the possibility of using computers to track human body movement in video so that computers can automatically respond to the inferred human intentions [2]. A perceptual user interface that can manipulate mouse operations is called a camera mouse [3]. A camera mouse system is usually composed of a visual tracking module and a mouse control module. The visual tracking module retrieves motion parameters from the video, and the mouse control module specifies the rules of control. The framework is illustrated in Fig. 1.

Facial features have been the most convenient body part for visual tracking and perceptual user interface. Researchers have proposed to navigate the mouse using the movement of eyes [4,5], nose [6]. However, we notice that the movement and 2D location of facial features in video usually does not coincide with the subject's focus of attention on the screen. This indeed makes the navigation operations un-intuitive and inconvenient. In order to avoid that problem, it has been proposed to navigate the mouse cursor by 3D head pose. The estimation of 3D head pose usually requires tracking of more than one feature. Head pose can be inferred by stereo triangulation if more than one camera are employed [5] or by inference from anthropological characteristics of face geometry [7]. Based on the technical developments in this area, some commercial products have been developed in recent years [8,9].

For the mouse control module, the conversion from human motion parameters to mouse cursor navigation can be categorized into direct mode, joystick mode, and differential mode. In the direct mode, a one-to-one

[*] Corresponding author.
E-mail address: jilintu@uiuc.edu (J. Tu).

1077-3142/$ - see front matter © 2007 Elsevier Inc. All rights reserved.
doi:10.1016/j.cviu.2006.11.007

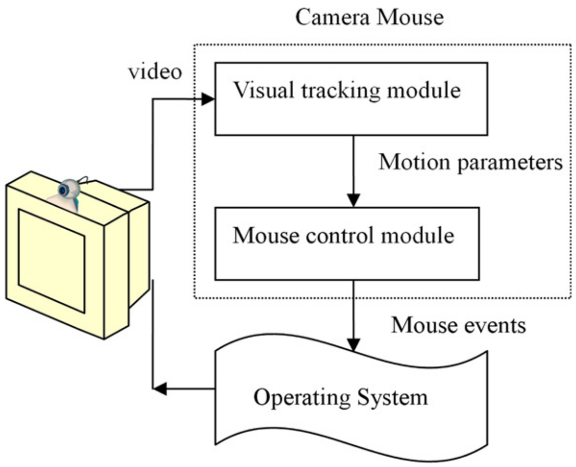

Fig. 1. The framework of a camera mouse.

mapping from the motion parameter domain to screen coordinates is established by off-line calibration or by design based on a prior knowledge about the human-monitor setting [7]. In the joystick mode, mouse cursor is navigated using the direction (or sign) of the motion parameters, and the speed of the cursor motion is determined by the magnitude of the motion parameters [10]. In the differential mode, the cumulation of displacement of the motion parameters drives the navigation of the mouse cursor, and some extra motion parameter switches on/off the cumulation mechanism so that the motion parameters can be reset without influencing the mouse cursor. Therefore, this mode is very similar to a hand-held mouse mode: user can lift the mouse and move it back to the origin on the mouse pad after performing a mouse dragging operation [10]. After the mouse cursor is navigated to the desired location, the execution of mouse operations (mouse button clicking) is triggered by detection of specific motion patterns. The most straightforward trigger is one that detects whether the motion parameters exceed specified thresholds. In [3,8], a mouse-click event is triggered by "dwell time", e.g. a mouse click is triggered if the user keeps the mouse cursor still for 0.5 s. In [11], the confirmation and cancellation of mouse operations is conveyed by head nodding and head shaking. A timed finite state machine is designed to detect the nodding and shaking from the raw motion parameters.

In this paper, we introduce a camera mouse system based on a 3D model-based visual face tracking algorithm [1]. This approach utilizes only one camera as video input, but is able to retrieve 3D head motion parameters and non-rigid facial deformation parameters. Based on the motion parameters retrieved, we design three mouse control modes. In the experiments, the controllability of the three mouse control modes is compared. Finally, we demonstrate how an user can navigate in the Windows XP environment and play games with our camera mouse.

2. 3D face modeling

The 3D geometry of human facial surface can be represented by a set of N vertices $\{(x_i, y_i, z_i)|i = 1, \ldots, N\}$ in space. In order to model facial articulations, some key facial deformations need to be defined. These key facial deformations are called Action Units (AUs). For our tracking system, six action units are defined as shown in Table 1. If the human face surface is represented by a long vector given by a concatenation of the 3D vertices $V = (x_1, y_1, z_1, x_2, y_2, z_2, \ldots, x_N, y_N, z_N)^{\mathrm{T}}$, the Action Units can be modeled as the displacement of the vertices of the deformed facial surface from a neutral facial surface, i.e., $\Delta V^{(k)} = (\Delta x_1^{(k)}, \Delta y_1^{(k)}, \Delta z_1^{(k)}, \Delta x_2^{(k)}, \Delta y_2^{(k)}, \Delta z_2^{(k)}, \ldots, \Delta x_N^{(k)}, \Delta y_N^{(k)}, \Delta z_N^{(k)})^{\mathrm{T}}$, where $k = 1, \ldots, K$ with K being the total number of key facial deformations. Therefore, an arbitrary face articulation can be formulated as $V = \bar{V} + Lp$ where $\bar{V}$ is the neutral facial surface, and $L_{3N\times K} = \{\Delta V^{(1)}, \Delta V^{(2)}, \ldots, \Delta V^{(K)}\}$ is the Action Unit matrix, and $p_{K\times 1}$ is the AU coefficient that defines the articulation. A piecewise Bezier volume deformation (PBVD) model is developed [1], and the six AUs can be manually crafted. An illustration of the PBVD face model with specified AU weights is shown in Fig. 2.

Taking head rotation and translation into account, the motion at each vertex V_i, $i = 1, \ldots, N$ on face surface in video can be formulated as $V_i' = R(\theta, \phi, \psi)V_i + T$ where $R(\theta, \phi, \psi)$ is rotation matrix, and $T = [T_x T_y T_z]'$ defines head translation.

Table 1
The Action Units

AU	Description
1	Vertical movement of the center of upper lip
2	Vertical movement of the center of lower lip
3	Horizontal movement of left mouth corner
4	Vertical movement of left mouth corner
5	Horizontal movement of right mouth corner
6	Vertical movement of right mouth corner

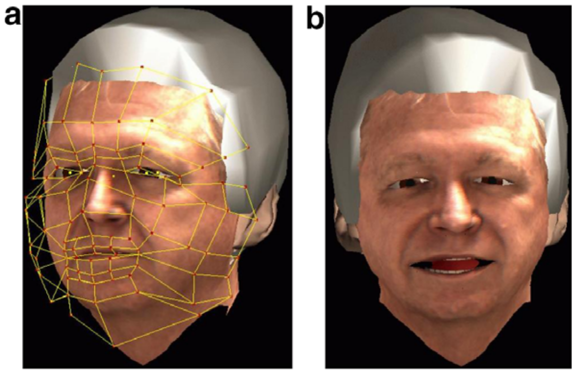

Fig. 2. The modeling of facial articulations. (a) Neutral face with control points for the piecewise Bezier volume deformation model. (b) Synthesized smiling face.

Assuming pseudo-perspective camera model $M = \begin{bmatrix} fs/z & 0 & 0 \\ 0 & fs/z & 0 \end{bmatrix}$ where f denotes the focal length, and s denotes scaling factor, the projection of the face surface to image plane can be described as

$$V_i^{\text{Image}} = M[R(\theta, \phi, \psi)(\bar{V}_i + L_i p) + T] \tag{1}$$

where V_i^{Image} is the projection of the *i-th* vertex node to image plane. Therefore, a face motion model is characterized by rigid motion parameters including rotation θ, ϕ, ψ, translation $T = [T_x T_y T_z]$, and non-rigid facial motion parameter, the AU coefficient vector p.

3. Visual face tracking

3.1. Initialization of face tracking

We note that Eq. (1) defines a highly nonlinear system, and hence the tracking of the model parameters is carried out in an extended Kalman filtering framework in which the non-linear system is approximated by a local linear model. Initialization of the tracking is done by manual labeling or by automatic detection of the face [12] and facial features [13,14] in the first frame of the video. The generic 3D face model is then adapted and deformed to fit the localized 2D facial features. An initialization result is shown in Fig. 4. The facial feature localization result is illustrated by the red face mask outline, and the yellow mesh indicates the 3D face model fitted to the face.

3.2. Tracking by solving differential equations [1]

As the tracking is carried out in an extended Kalman filtering framework, the displacement of V_i^{Image}, $i = 1, \ldots, N$ can be estimated as optical flow $\Delta V_i^{\text{Image}} = [\Delta X_i, \Delta Y_i]^{\mathrm{T}}$, $i = 1, \ldots, N$ per frame. The model parameter displacements, $\mathrm{d}W$, $\mathrm{d}p$, $\mathrm{d}T$ can be computed by LMSE fitting with Jacobian $\frac{\Delta V_i^{\text{Image}}}{\mathrm{d}W, \mathrm{d}p, \mathrm{d}T}$ estimated at each vertex node V_i, $i = 1, \ldots N$ formulated by Eq. (2).

$$\frac{\Delta V_i^{\text{Image}}}{\mathrm{d}W, \mathrm{d}p, \mathrm{d}T} = M\left[\begin{bmatrix} 1 & 0 & -x/z \\ 0 & 1 & -y/z \end{bmatrix} \begin{bmatrix} G_0 - \frac{x}{z} G_2 \\ G_1 - \frac{y}{z} G_2 \end{bmatrix} \begin{bmatrix} [\mathrm{RL}]_0 - \frac{x}{z}[\mathrm{RL}]_2 \\ [\mathrm{RL}]_1 - \frac{y}{z}[\mathrm{RL}]_2 \end{bmatrix}\right] \tag{2}$$

where

$$G = \begin{bmatrix} 0 & z_1 & -y_1 \\ -z_1 & 0 & x_1 \\ y_1 & -x_1 & 0 \end{bmatrix}, \quad \begin{pmatrix} x_1 \\ y_1 \\ z_1 \end{pmatrix} = \bar{V} + Lp, \begin{pmatrix} x \\ y \\ z \end{pmatrix} = R \begin{pmatrix} x_1 \\ y_1 \\ z_1 \end{pmatrix} + T$$

and G_i and $[\mathrm{RL}]_i$ denote the *i*-th row of the matrix G and RL, respectively.

3.3. Implementation of the tracking system

In our tracking system, the optical flow is computed by template matching with normalized correlation. As normalized correlation is known for tolerating changes in ambient light, the tracker can sustain illumination variations. The searching region for optical flow computation is of size 5 by 5, therefore optical flow estimation error caused by occlusion or outlier is upper-bounded, and the tracker can therefore tolerate minor occlusions/outliers through global LSME fitting Eq. (2). Automatic re-initialization of the tracker is triggered by an increase of the LMSE model fitting error and the confidence measure drop of the optical flow(derived from the sum of the normalized correlation at all vertices). All these advantages make our tracking system a good candidate for the visual tracking module in the camera mouse framework.

4. Mouse cursor control

Three mouse cursor control modes, direct mode, joystick mode and differential mode are implemented for the mouse control module. For the direct mode, the face orientation angle R_x, R_y (the rotation angle with respect to x and y coordinate) are mapped to the mouse cursor coordinates (X, Y) in the screen. As the reliable tracking range of R_x and R_y is about 40°, and the resolution of the screen is 1600×1200, we therefore empirically specify the mapping function as

$$X = \frac{1600}{40^\circ}(R_y - R_y^0)$$
$$Y = \frac{1200}{40^\circ}(R_x - R_x^0)$$

where R_x^0 and R_y^0 are the initial face orientation angles.

For the joystick mouse control mode, the following control rule is employed.

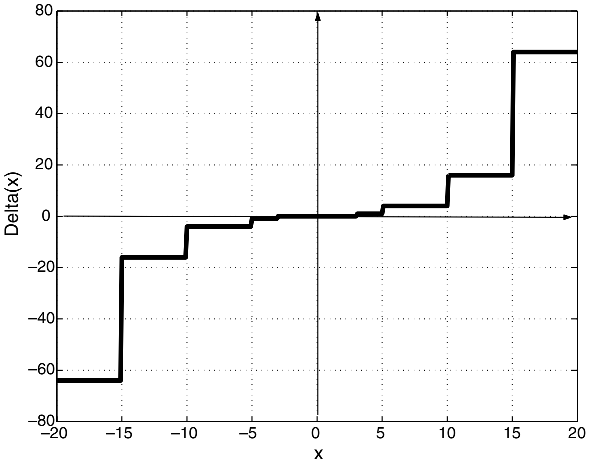

Fig. 3. The rule function $C(x)$.

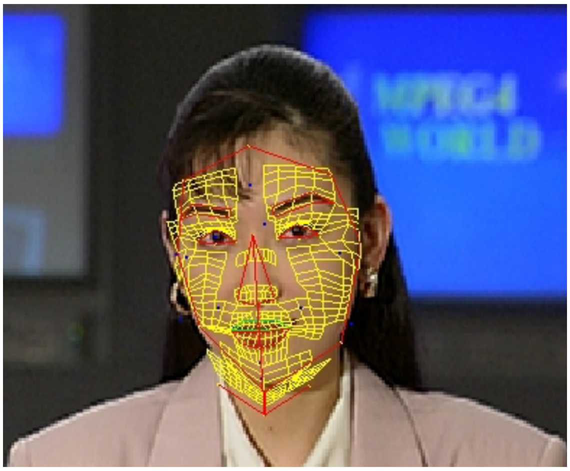

Fig. 4. Initializing the tracker.

$$X^{t+1} = X^t + C(R_y - R_y^0 + \Delta R_y)$$
$$Y^{t+1} = Y^t + C(R_x - R_x^0 + \Delta R_x)$$

with the rule function $C(x)$ defined by Fig. 3. The constants and thresholds in $C(x)$ function are chosen empirically. The term ΔR_x and ΔR_y defines *negative inertia* [15] that allows the cursor to respond more promptly to user movements.

For the differential mouse control mode, we have the following control rule

$$X^{t+1} = X^t + \alpha \Delta T_x^t b^t$$
$$Y^{t+1} = Y^t + \beta \Delta T_y^t b^t$$

where boolean variable b^t is true when $T_z^t >= T_z^0$.

The mouse cursor is navigated by the cumulation of head translation displacements ΔT_x^t and ΔT_y^t. Clutching of the cumulation mechanism is determined by distance from face to the camera, i.e., the mouse cursor navigation is turned on when the face moves closer to the camera than where it was at the beginning.

The mouse button click events are triggered by specified mouth motions detected in the non-rigid facial motion parameters, i.e., the detection of mouth opening triggers left-button-click event, and the detection of mouth corner stretching triggers right-button-click event.

Our camera mouse is seamlessly integrated with the Windows operating system, and the computer user can navigate in Windows and operate softwares by moving his face.

5. Experiments

The setup of our camera system is shown in Fig. 5. A Logitech Pro 5000 webcam is mounted beneath the screen, and captures the user's face. The size of the screen is about 16 by 12 inches, and the subject is sitting about 20 inches away from the screen. We first test the controllability by clicking on a target picture 10 times in a painting software. As shown in Fig. 6, the average localization error for the direct mode is about 10 pixels, that for the joystick mode is about 3 pixels, and that for the differential mode is about 5 pixels. The localization error is mostly caused by the measurement error introduced by tracking, and partially because the subject can not really hold his head still.

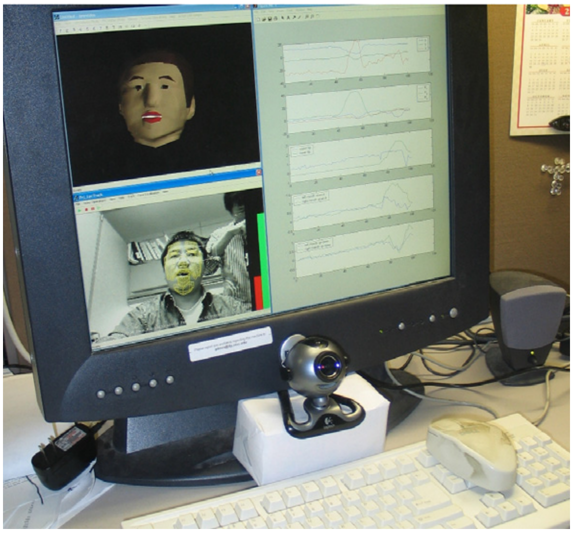

Fig. 5. The setup of the camera mouse.

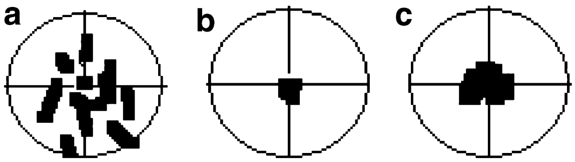

Fig. 6. The localization error of the mouse modes. (a) The direct mode. (b) The joystick mode. (c) The differential mode.

We then generate 4 by 4 arrays of square buttons in the screen with the button width specified among 0.1, 0.3, 0.65, 1, 1.5, and 3 inches. The buttons will be highlighted one by one in random order and stay on screen till the user clicks on it. And the goal of the user is to click all the buttons in

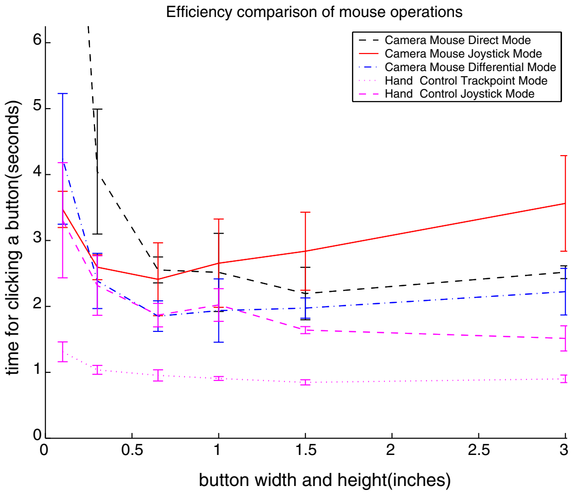

Fig. 7. The efficiency of the mouse modes.

minimum time. For each array of different button size, the subject carries out the experiment 5 times. The average time required to carry out button clicking is recorded as an indication of convenience for the subject to carry out mouse cursor navigation and button clicking. Fig. 7 shows the comparison of the average button clicking time of the three camera mouse modes with the standard deviation. The performance of a hand controlled trackpoint on IBM laptop keyboard and that of a Logitech WingMan joystick mouse are also provided to serve as baseline comparisons. From the experiment, we make the following observations:

- The cursor in direct mode is correlated with focus of attention. So it is relatively convenient to navigate across the screen and to click larger buttons, but its accuracy is awkward as it takes forever to click a button of size 0.1 in. due to noise introduced in tracking.
- It is more convenient to navigate the cursor with accuracy in joystick mode for camera mouse while it take longer time to navigate across the screen and click larger buttons. The mouse cursor is correlated with the direction of focus of attention, but not as strongly as the direct mode. However, the user can gaze at the focus of attention if it is not consistent with the head pose.
- The differential mode is supposed to match the most with human's mouse-navigation habit. And it outperforms the joystick mode for clicking larger buttons. However, this mode requires the user to frequently move head in large motion range, and the mouse cursor is not correlated with human's focus of attention. This could be disturbing if the user wishes to read the screen at the same time.
- The average button clicking time is about 1 s by hand controlled trackpoint mouse. And the camera mouse modes are at least twice slower than the trackpoint mode. However, their performances are comparable to that of the Logitech WingMan joystick mouse.

The camera mouse runs on a PC with 2 GHz XEON CPU at frame rate 22 fps when tracking rigid motions and 17 fps when tracking both rigid and nonrigid motions. The experiment lasted about 1.5 h. There was four times the tracker lost the target mostly when the subject is clicking the 3-in. button array as the subject has to turn his head to near profile view for navigating mouse cursor to the button at one corner of the screen. When the user is assuming near-frontal view (within ±40°) and does not move too fast, the tracker achieves reasonably stable tracking performance.

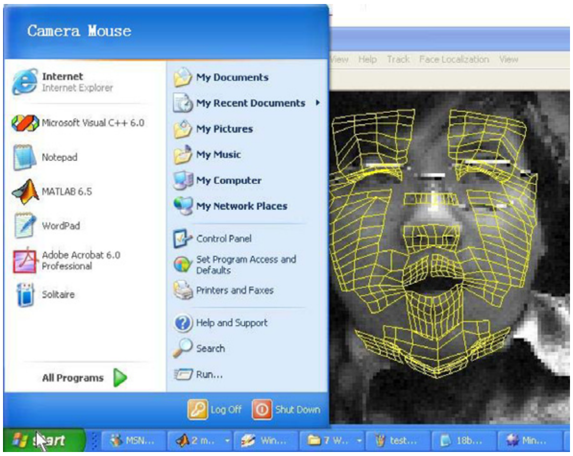

Fig. 8. Using camera mouse to navigate in Windows XP.

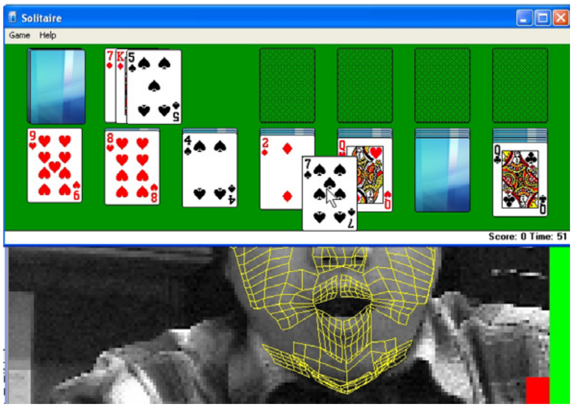

Fig. 9. Using the camera mouse to play Solitaire. The user is dragging the Spade-7 from right toward the Heart-8 in the left by turning his head with the mouth opened.

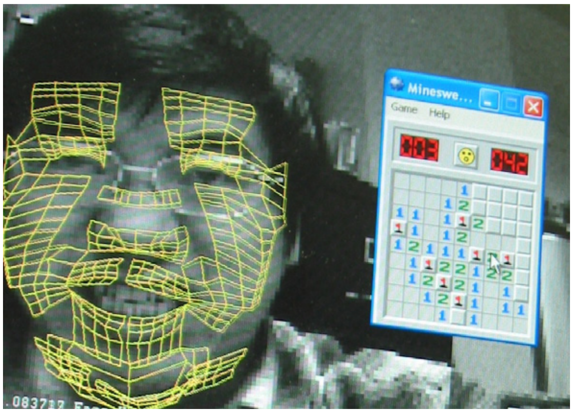

Fig. 10. Using the camera mouse to play Minesweeper. The user is sweeping the remaining squares at the cursor location(where 2 mines have been marked) by simultaneous mouth opening and stretching which corresponds to clicking the left and right mouse buttons at the same time.

Since our system is seamlessly integrated to Windows XP environment, we can use our camera mouse to navigate in the Windows environment. Fig. 8 shows the user is activating the Windows Start menu by opening his mouth. Figs. 9 and 10 show how the user plays Solitaire and Minesweeper using the camera mouse.

6. Conclusion

In this paper, we proposed a camera mouse based on a 3D model based visual face tracking technique. Three mouse control modes, direct mode, joystick mode, and differential mode, are implemented. The experiments compared the controllability of the three mouse modes, and the pros and cons of the control modes are analyzed. By comparison, we conclude that the joystick mode is the best choice among the three camera mouse modes while taking into consideration both the navigation convenience and the user screen reading habits. Comparing to the reported camera mouses [7,8,10], our camera mouse is navigated by head pose and the mouse events are triggered by detection of mouth motions, we believe the mouse operations with our system are more intuitive and convenient for users. While our camera mouse is still not as convenient as hand controlled mouses, it provides an alternative HCI solution for people with handicap and for HCI scenarios in which the subject's hands are occupied with other tasks.

Appendix A. Supplementary data

Supplementary data associated with this article can be found, in the online version, at doi:10.1016/j.cviu.2006.11.007.

References

[1] H. Tao, T. Huang, Explanation-based facial motion tracking using a piecewise bezier volume deformation model, in: Proceedings of the IEEE Computer Vision and Pattern Recognition, vol. 1, 1999, pp. 611–617.

[2] M. Turk, G. Robertson, Perceptual user interfaces, in: Communications of the ACM, vol. 43, 2000, pp. 32–34.

[3] M. Betke, J. Gips, P. Flemimg, The camera mouse: visual tracking of body features to provide computer access for people with severe disabilities, IEEE Trans Neural Syst Rehabil Eng. 10 (1) (2002) 1–10.

[4] R.J. Jacob, What you look at is what you get: eye movement-based interaction techniques, in: Human Factors in Computing Systems, 1990, pp. 11–18.

[5] R. Ruddarraju, Perceptual user interfaces using vision-based eye tracking, in: The 5th International Conference on Multimodal Interfaces, Vancouver British Columbia, Canada, 2003, pp. 227–233.

[6] D. Gorodnichy, On importance of nose for face tracking, in: 5th International Conference on Automatic Face and Gesture Recognition, Washington, DC, USA., 2002.

[7] 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] <http://www.cameramouse.com/>.

[9] <http://www.mousevision.com/>.

[10] D. Gorodnichy, S. Malik, G. Roth, Nouse 'use your nose as a mouse'—a new technology for hands-free games and interfaces, in: International Conference on Vision Interface, Calgary, 2002, pp. 354–361.

[11] J.W. Davis, S. Vaks, A perceptual user interface for recognizing head gesture acknowledgements, in: Workshop for Perceptive User Interface, Orlando, FL, 2001, pp. 1–7.

[12] P. Viola, M. Jones, Fast and robust classification using asymmetric adaboost and a detector cascade, in: Advances in Neural Information Processing System, vol. 14, MIT Press, Cambridge, MA, 2002.

[13] T.F. Cootes, C.J. Taylor, Active shape model search using local grey-level models: a quantitative evaluation, in: 4th British Machine Vision Conference, BMVA Press, 1993, p. 639648.

[14] J. Tu, Z. Zhang, Z. Zeng, T. Huang, Face localization via hierarchical condensation with fisher boosting feature selection, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'04), vol. 2, 2004, pp. 719–724.

[15] R.C. Barrett, E.J. Selker, J.D. Rutledge, R.S. Olyha, Negative inertia: a dynamic pointing function, in: CHI '95 Proceedings, Bergen, 1995.