Camera as Mouse and Keyboard for Handicap Person with Troubleshooting Ability, Recovery, and Complete Mouse Events

Kohei Arai arai@is.saga-u.ac.jp
Information Science
Saga University
Saga, Japan

Ronny Mardiyanto ronny_mardiyanto@yahoo.com
Electrical Engineering
Institut Teknologi Sepuluh Nopember
Surabaya, Indonesia

Abstract

Camera mouse has been widely used for handicap person to interact with computer. The utmost important of the use of camera mouse is must be able to replace all roles of typical mouse and keyboard. It must be able to provide all mouse click events and keyboard functions (include all shortcut keys) when used by handicap person. Also, the use of camera mouse must allow users do troubleshooting by themselves. Moreover, it must be able to eliminate neck fatigue effect when it is used during long period. In this paper, we propose camera mouse system with timer as left click event and blinking as right click event. Also, we modify original screen keyboard layout by add two additional buttons (button “drag/ drop” is used to do drag and drop of mouse events and another button is used to call task manager (for troubleshooting)) and change behavior of CTRL, ALT, SHIFT, and CAPS LOCK keys in order to provide shortcut keys of keyboard. Also, we develop recovery method which allows users break for a while and then come back again to eliminate neck fatigue effect. The experiments involving several users have been done in our laboratory. The results show that the use of our camera mouse able to allow users do typing, left and right click events, drag and drop events, and troubleshooting without hand. By using this system, handicap person can use computer more comfortable and reduce the eyes fatigue.

Keywords: Camera mouse, timer, blinking, mouse events, keyboard shortcuts, troubleshooting, fatigue.

1. INTRODUCTION

Recently, the numbers of handicapped persons who have limited in mobility have been rising. These may be caused by accident, congenital, diseases, and etc. These make person become cannot move their foot, hand, or other mobility organs. These limitations make them feel loose their life as they become dependent with others [9][10]. The rising numbers of handicapped person encourage the growing of computer input development to allow them interact with computer with their limitation to assist their daily life.

Computer input special design for handicapped person have been developed by many researchers. Computer input device, such as keyboard, mouse, joystick, and etc, have been widely used for computer interaction. Unfortunately, typical computer input device always requires hand's control. Controlling of cursor and selection of key must be done by using hand. To provide computer input for handicapped person, the possible solution is by utilizing camera as mouse and also keyboard. Mouse-keyboard utilizing web camera has been used to replace typical mouse and keyboard. It means that by utilizing web camera as mouse, it also can be used as keyboard by using screen keyboard. The method of controlling pointer in camera-based mouse-keyboard can be broadly classified into following categories: (1) Gaze-based (The pointer is controlled based on user's gaze) [2][3], (2) Face-based (The pointer follows center of face location) [1][6], (3) Head poses-based (The pointer is controlled based on head poses such as roll, tilt, and yaw) [4], (4) Hand gesture-based (The pointer is controlled by using hand gesture) [5], and (5) Laser based (Laser pointer mounted on user's glasses controls the pointer) [7].

Ref [2] developed eye mouse system based on user's gaze. After face is found and tracked, eye location is searched by projection of difference between left and right eye images. Output of this system is only left and right direction which used to control mouse pointer. No upward and downward directions are used. Ref [3] developed eye mouse system which user's gaze is obtained from pupil location. Also, blinking is used as left click mouse event. Ref [1] developed camera mouse using face detection and eye blink. Center position of face is detected by using Adaboost face detection method and tracked by using Lucas-Kanade method. This location is used as pointing value and blinking is used as left click mouse event. In ref [6], camera mouse is driven by 3D model based visual face tracking method. Human facial movement is decomposed into rigid movement (rotation and translation) and non rigid movement (facial expressions). Mouse cursor is controlled by face movement (x and y coordinates). Left click event is operated by opened mouth detection and right click event is operated by stretching mouth corner detection. Ref [4] developed camera mouse based on head poses. Head poses such as yaw, roll, and tilt are used to tune the mouse cursor. Rough position of mouse is estimated based on center of face. Right and left click mouse event are done based on roll angle. Ref [5] developed camera mouse based on hand gesture. Hand location is detected and tracked by using SIFT and CAMShift algorithm. Hand gesture is recognized by using PCA method. Mouse event is controlled by stretching the finger. Ref [7] developed unistroke keyboard and mouse by using image of a keyboard (is made from paper) and laser pointer mounted on user's glass. The keyboard image is monitored by a camera and illuminated key locations are recognized. According to the type of the click mouse event, it can be classified into (1) Left click event only [1] [3] and (2) Left and Right Click events [4] [5] [6]. There are several methods such as timer, blinking, roll movement of head, stretching of finger, and mouth condition are used to control mouse events. Refs [1] and [3] used blinking as left click event. Refs [4], [5], and [6] used roll movement of face, stretching of finger, and state of mouth to control right and left click mouse event. The comparison between click mouse events of each method is shown in table.1.

Ref	**Click Events**	
	Left Click	**Right Click**
[1] [3]	Blinking	None
[2] [7]	None	None
[4]	Roll movement of face	Roll movement of face
[5]	Stretched finger	Stretched finger
[6]	Opened mouth	Stretched mouth

TABLE 1: Comparison between existing camera mice according to their click events method

The existing camera mice only consider how to select and choose the desire point. There is no camera mouse system which provides complete click events (right and left click events), allows shortcut keys of keyboard, and also allows user do troubleshooting when not-responding program happens. Moreover, the uses of camera mouse during long period make user's eyes become dry moreover give neck fatigue effect. For the aforementioned reason, we propose camera-based mouse-keyboard system with troubleshooting ability, recovery, and complete click mouse events by using timer and blinking.

The objectives of this proposed system is to overcome click mouse events, shortcut keys of keyboard, troubleshooting, recovery, eyes dryness, and neck fatigues effect problems of previous camera mice system. Mostly, the existing camera mice only provide left click mouse event. Only Refs [5] and [6] provide both of left and right click mouse events. It utilize stretched finger and opened mouth to control left and right click mouse events. Because the use of camera mouse as keyboard always involve screen keyboard, it must allow all keyboard shortcut keys. Unfortunately, the original screen keyboard didn't provide hold function of CTRL, ALT, SHIFT, and CAPS LOCK keys and causes system eliminate shortcut keys such as CTRL+C as copy, CTRL+V as paste, and etc. When computer freezes, the typical solution is by press CTRL + ALT + DEL keys simultaneously to show task manager. Unfortunately, the use of camera mouse for handicapped person eliminates this shortcut because user cannot press CTRL + ALT + DEL keys simultaneously by hand. Also, when users want to break, the system should be able to recover and able works again when user comes back to front of computer. The use of camera mouse during long period makes user's neck and eyes become tired. Also the use of blinking as left click event makes user's eye become dry.

In this paper, we propose camera mouse system combining timer and blinking as right and click mouse events. Left click event is controlled by timer and right click event is controlled by blinking. Our proposed system is different with other methods because it provides complete click mouse events. This method allows users do all of click mouse events. Moreover, we modify windows screen keyboard layout by add drag/ drop, task manager buttons, and change the behavior of CTRL, ALT, SHIFT, and CAPS LOCK keys in order to allow shortcut key of keyboard. The drag/ drop button allows users drag and drop the desired object. Instance, when user want to move the window of program, users click "drag/ drop" button first and continue with left click. This first click will drag the object until the second click happens. This makes window program able to be moved to another place and drop will be automatically selected when the second left click happens. Also, task manager button allows users do troubleshooting by themselves. When system freezes, user can end task the active program by call the task manager. It will replace the use of "CTRL – ALT – DEL" combination keys. Moreover, the visibility of user face is monitored by using three tracked points. When users face is invisible, system will automatically reboot and back to the beginning process. This kind of recovery process will eliminate neck fatigue effect and eye's dryness.

This paper is organized as follows: section 2 describes our proposed system involving hardware configuration, modified screen keyboard, mouse click events, and recovery method. Section 3 describes our experimental results involving performance camera mouse when used for typing, blinking detection accuracy, and testing of recovery. Section 4 describes the conclusions.

2. PROPOSAL

The problem of existing camera mouse is how the use of camera mouse can replace all functions of typical mouse and keyboard. As we now that typical mouse has two click events (left and right click), left click is used for selection and right click is used to show the additional menus. The system should provide drag/ drop events without burden the user. Because the use of camera mouse automatically leaves the use of hand, it must be able to play keyboard's role as one of computer input device. Beside it can be used for typing, it must be able to do troubleshooting. When computer become freeze, by press "CTRL - ALT - DEL" user can end task the not-responding program. The use of camera mouse must cover whole functions of typical mouse and keyboard. Pointing, clicking events, typing, and troubleshooting must be done by camera mouse only. Moreover, the use of camera mouse should not burden the user (make user's neck fatigue).
When we use camera mouse as computer input, it should be considered about how the user do typing, drag/ drop event, troubleshooting, right and left click event, and others. To provide typing function, almost camera mouse use screen keyboard software. The screen keyboard software with keys button layout allows user type by clicking the key button. Actually, screen keyboard

software can replace the typical keyboard's function. Unfortunately, screen keyboard program doesn't allow user types "CTRL - ALT -DEL" simultaneously. It means that the use of screen keyboard eliminate troubleshooting keys because user cannot open the task manager program by pressing "CTRL - ALT -DEL" simultaneously.

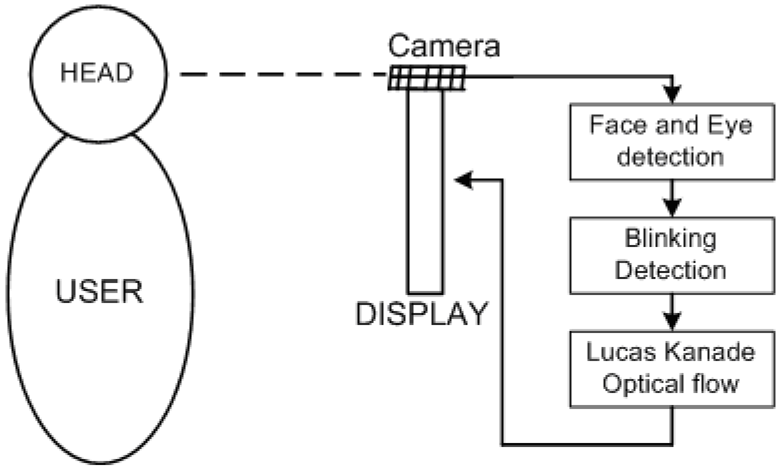

FIGURE 1: Block diagram of proposed system.

This paper proposes camera mouse with contribution involving modification of screen keyboard layout with task manager and drag/ drop buttons, left click is done by timer, right click is done by blinking, and the recovery method allowing user go away and go back from/ to front of camera in order to avoid neck fatigue and dryness of eye. By adding of task manager button, it allows user does the troubleshooting by him/ her self. Also, after drag/ drop button is clicked, once user click the object, this click is recognized as drag event and next click is recognized as drop event. The use of timer and blinking make camera mouse system acting as typical mouse (with left and right click events). To update the cursor position, once the face location is found by Adaboost based on Haar-like feature method (proposed by Viola and Jones), Lucas-Kanade Optical flow track this position and use to update cursor position. Last, we use normal flow to detect blinking. The block diagram of proposed system is shown in Figure 1.

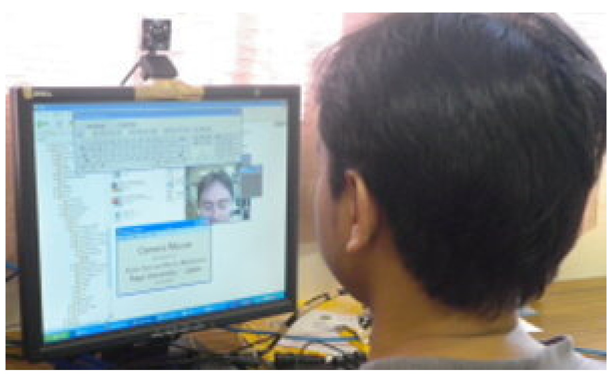

FIGURE 2: Proposed Camera Mouse System.

2.1 Hardware Configuration

We develop our system using Visual Studio 2005 and OpenCv Library downloaded as free on their website. Optiplex 755 dell computer with Core 2 Quad 2.66 GHz of CPU and 2G RAM is used. Web camera NetCowboy DC-NCR 131 with 1.3 M pixels resolution is used as main source. Web camera is placed on top of display. The distance between user and camera is 30 cm. Even though we used high speed computer, we have tested that our system works with ASUS EEEPC 1005HE netbook (Intel Atom N280 processor with 1.66 GHz, 1GB RAM, and built-in1.3 M pixels web camera). Also the use of our system is not limited for Infra Red camera but also invisible camera. It means that all type web cameras with minimum resolution 1.3 M pixels also can be used. The proposed camera mouse is shown in Figure 2.

2.2 Modified Screen Keyboard

The objective of this modification is to allow users do drag/ drop click events and troubleshooting. The existing camera mouse systems do drag/ drop events by utilizing blinking. The closed eye can be signed as drag event and when opened eye is signed as drop event. The weakness of this method is that drag/ drop events are always rely on eye while eye condition will become tired/dry during long use duration. When eye is used only in a few minute, it will not make eye become dry. Unfortunately, the use of camera mouse is usually in long duration depending on necessity. The modified screen keyboard layout is shown in Figure 3.

FIGURE 3: Modified Screen Keyboard

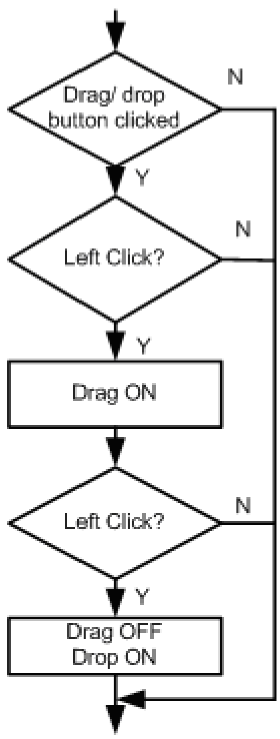

FIGURE 4: Process flow of drag and drop events.

We modify the screen keyboard layout by adding two buttons. The first button is task manager button assisting user when not-responding program happens. It means that the troubleshooting of computer which usually do by pressing “CTRL + ALT + DEL” keys, now is replaced with this button. When user uses the camera mouse and not-responding program appears, user can click this button to end task the not-responding program. The second button is drag/ drop button for assisting user do drag and drop the object. Instance, when user want to move the window position of running program, user can click this drag/ drop button and continue with click to the window and then the last click is automatically signed as drop event. The adding of this button replaces function of drag and drop mouse click events. The process flow of drag and drop events are shown in Figure 4.

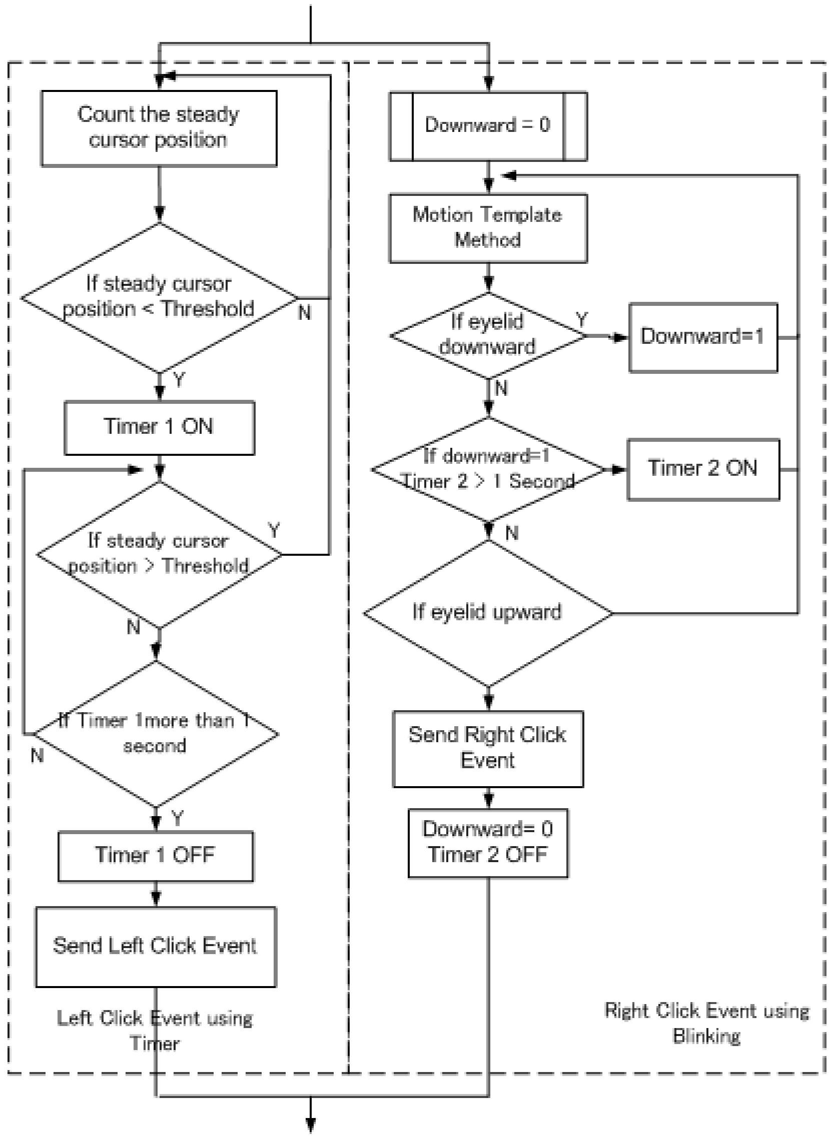

FIGURE 5: Click event process flow.

2.3 Mouse Click Events

To interact with computer, mouse has been used by user to give input on graphical user interface by using clicking. The stored procedure can be controlled by using mouse. It means that mouse assist user does interaction with computer by clicking the button or other graphical user interface tool. The typical mouse usually has two buttons: (1) left click button (performs a first action on object selection) and (2) right click button (performs a secondary action). After mouse cursor is controlled, the decision is done by these click events. Instance, when mouse is used for typing using screen keyboard software, user select the desired key by clicking the key. Typically this clicking is done by using left click. Also, when user wants to show the properties menu of icon, user does right click on this icon. Both of left and right clicks can be used simultaneously and it make user become easy to interact with computer. In this paper, we propose mouse click events using timer and blinking. When camera mouse is used to replace the role of typical mouse, it also must cover the complete mouse click events. Left and right click must be provided by camera mouse system. Our system use timer as left click event. By using timer, left click will automatically click when cursor position stops during specific duration. This method is easy to use than other method such as blinking, head rotation, opened mouth, and etc because there is no physiological effort is required. According to that left click event is more often used than right click event, the method must not burden the user physiology. We cannot use blinking method as left click because the use of blinking during long period causing the user's eye becomes dry. We use blinking as right click event. The click events process flow is shown in Figure 5.

The both click events run simultaneously. In left click event, system counts the steady of cursor position. If the steady of cursor position less than threshold, timer 1 is activated. However, when steady of cursor more than threshold, timer 1 is resetting. If timer 1 more than 1 second, system send left click event while timer 1 is turned OFF. The steady of cursor position is calculated as follow,

$$\mu(x, y) = \sum_{i,j} \frac{P(x_i, y_j)}{N}, \quad (1)$$

$$Steady = \sum_{N} \left| P_N(x) - \mu_N(x) \right| + \left| P_N(y) - \mu_N(y) \right|, \quad (2)$$

where P(x, y) is cursor location and N is number of locations. The left click event by using timer is shown in Figure 5 (left).

To provide right click event, we use blinking. Blinking is detected by using motion template method [8]. This method obtains magnitude and angle of total motion on image. After eye image is cropped, motion template method estimates the motion. We detect blinking if magnitude of motion on image is more than threshold and the first angle is 270º (eye become closed) and the opened eye is signed by angle is 90º. The blinking detection method is shown in Figure 5 (right). First, we detect the closed eye by using motion template method. When eye become closed, timer is turned ON. If eye become open when timer 2 bellow than 1 second, it will not be recognized as blinking. Otherwise, if eye become open and timer 2 exceed 1 second, it will be recognized as blinking. If blinking happens, system send right click event.

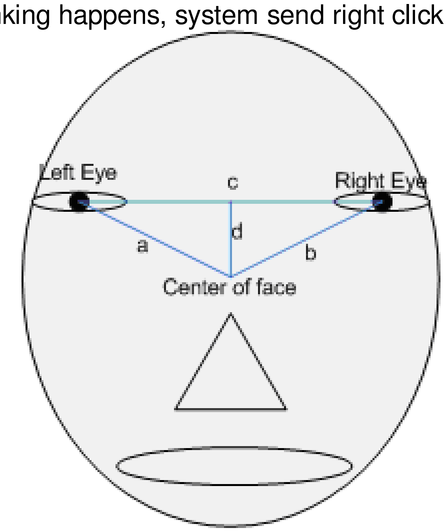

FIGURE 6: Detection of face dissipation caused by head movement.

2.4 Recovery Method

To avoid neck fatigue and dryness of eyes, we develop recovery method. This method prevents error when user's face is invisible. When user's neck becomes tired, user may take a rest by going away from camera. Because the face is invisible, error may happen due to this situation.

This kind of error can be avoided by resetting all variables and return to the beginning. The action of dissipation of user's face can be divided into: (1) out of region (when user shifts the position until out of the camera boundary) and (2) dissipation that caused by head movement (yaw and tilt). When user shifts the face until out of region, the all variables are reset and return to first process. However, disappearing of user's face that caused by head movement is detected by measuring the reasonable distance between center face and eyes. If the distances of face and eyes are more than threshold, then all variable is reset.

Figure 6 shows the detection of dissipation of face which caused by head movement. The over yaw movement happens when |a-b|> threshold. The over tilt movement happens when d > threshold. When the over head movements are detected, system will reset all variables and return to beginning.

3. EXPERIMENTAL RESULTS

To test the camera mouse performance, we conducted several experiments: measurement of blinking accuracy, testing of recovery (measuring the camera mouse performance when user go from front of camera and then come back again to front of camera), and testing of typing performance (use the camera mouse to type on MS Word). The detail experiments are explain bellow,

3.1 Blinking Accuracy

The objective of this experiment is to measure the accuracy of blink detection method. We measure the blinking accuracy by compare between manual detection and our blinking detection method. This experiment is done by users blink their eye 20 times. The accuracy of blinking detection is 90%.

3.2 Recovery Testing

The objective of this experiment is to know how our system can recognizes invisible user's face and also recover when user comes back again to front of camera. We counted the time of recovery process for ten times. The average time for recovery process is 8.7 seconds.

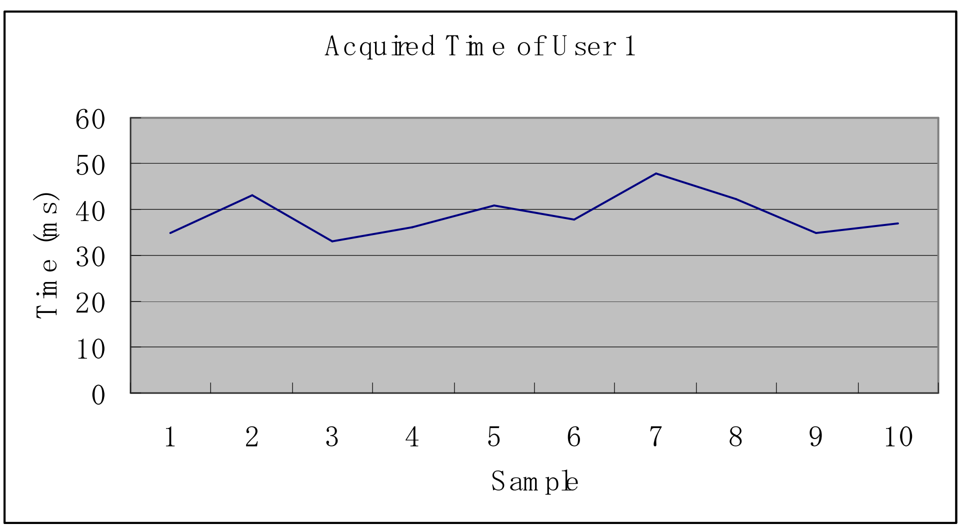

FIGURE 7: Acquired Time of User 1. This figure shows that the acquired time is stable and relative fast.

3.3 Typing Performance

To test the camera mouse usability, we conducted experiment with user typing on MS Word by using our camera mouse. This experiment involves four different users with each of them has different nationality. User sit down in front of computer with web camera mounted on the top of display and types “Saga University” on MS Word using our camera mouse. User types “Saga University” until ten times while the consumption times are recorded. The recorded times are shown in Figure 7, Figure 8, Figure 9, and Figure 10. The first user is expert user. He is Indonesian. He often used our software for typing. He can fast and easy control the cursor of our camera mouse. The acquired time of first user is shown in Figure 7.

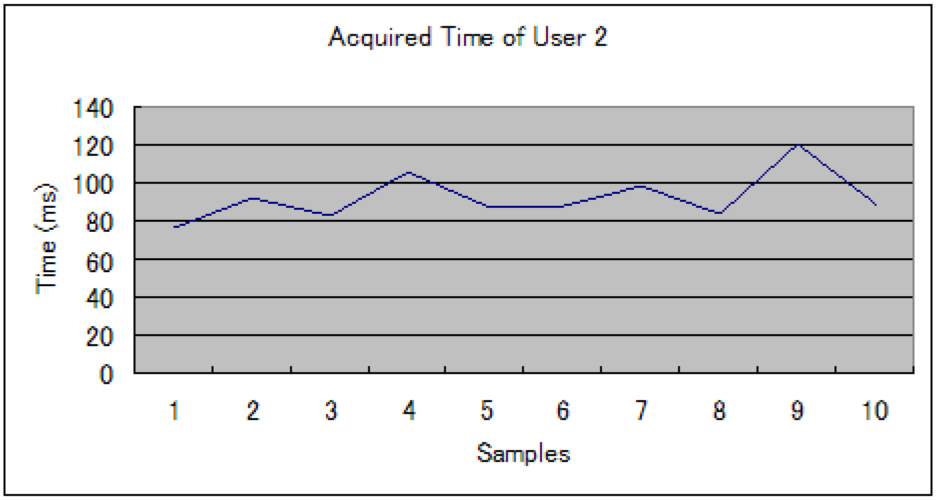

FIGURE 8: Acquired Time of User 2. This figure shows that the acquired time rise at last sample. At last sample show that user become tired on his head.

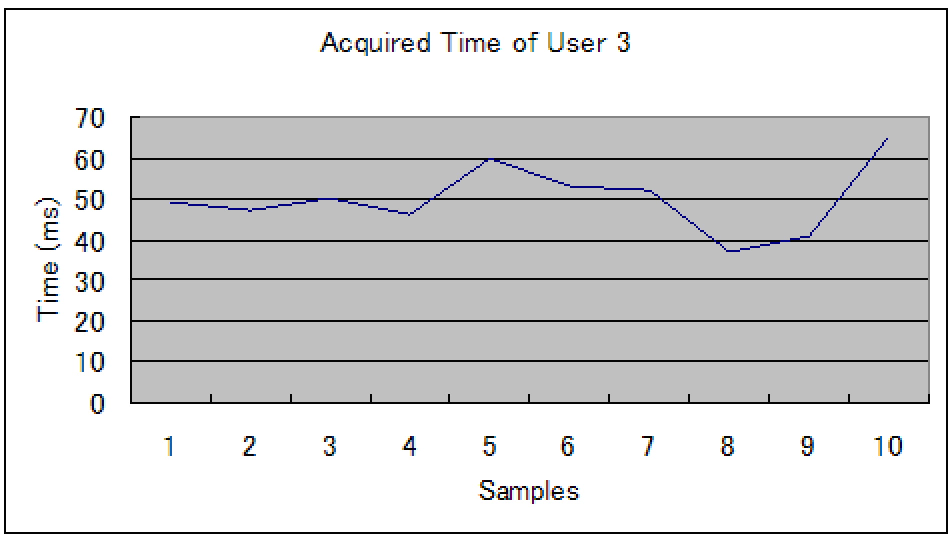

FIGURE 9: Acquired Time of User 3. This figure shows that between first samples until 7th sample, user able stable to type on MS Word. At 8th and 9th samples, user able type faster than previous because he is lucky and no mistake happen. Unfortunately, at 10th sample user become tired and he made several mistake and caused longer time to type.

The second user is beginner user. He is Sri Lankan. This experiment is the first time for him using our camera mouse. Before the experiment, he never used it before. Because of this, we explained how to use this and let him practiced for one minute. After one minute practiced, he began to type “Saga University” on MS Word by using our camera mouse. The recorded consumption time is shown in Figure 2. In this figure, we can see that the time will increase when user uses our camera mouse during long period. It is caused by he feel that his neck becomes tired. His neck will become tired because he never used his neck for controlling something before. If he uses it everyday then he will not feel tired again. The next user is also beginner user. He is

Indonesian. He never used our camera mouse before. After did practice for one minute, he began to type “Saga University” using our camera mouse. The acquired time is shown in Figure 3. The last user is also beginner user and never used it before. He is Bangladeshi.

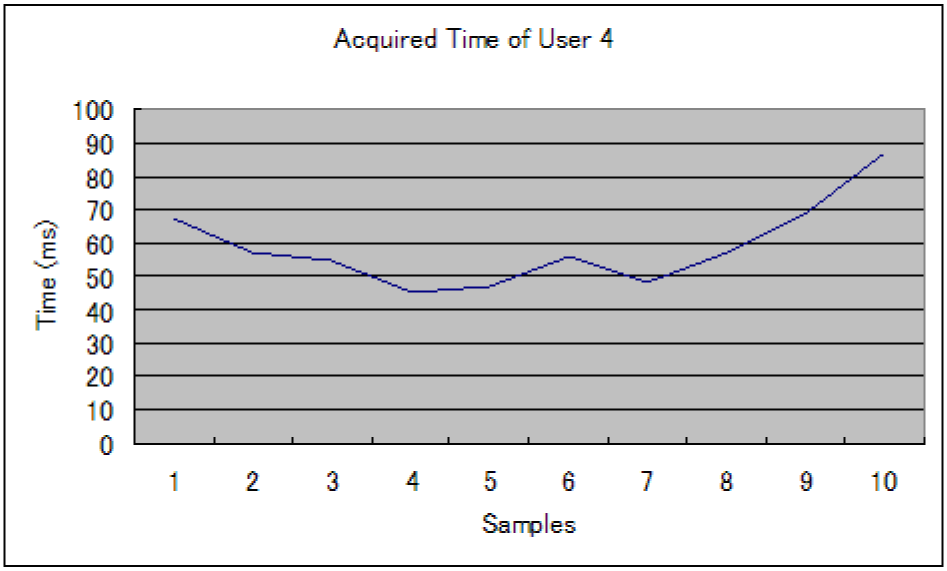

FIGURE 10: Acquired Time of User 4. This figure shows that the acquired time is decrease at four of first samples. It is caused that user need to adapt with our camera mouse. After 4th sample shows that it becomes stable because user becomes familiar with it. The two of last sample show that acquired time is increase. It is caused by user neck become tired.

3.4 Works with Others Windows Application

We used our camera mouse to control windows application such as MS Word (For typing), Soliter Game (Using drag and drop functions), selecting file from windows explorer, playing music by using windows media player, setting properties of file by using blinking as right click event, setting windows volume, and etc. User used our camera mouse system which acts as mouse and keyboard simultaneously.

4. CONSLUSIONS

The camera mouse with ability to replace all mouse and keyboard functions has been successfully implemented. By using timer and blinking, users successfully do left and right click events, drag and drop events, and also troubleshooting. Combination between timer and blinking, as left and right click events, has advantage in reducing fatigue because timer can be used more often than blinking, so timer is suitable to replace left click meanwhile blinking is used to replace right click event. The added button allows user did troubleshooting by clicking the task manager button from our modified screen keyboard layout. Another added button also allows user use shortcut combination keys such as “CTRL+A”, “CTRL+C”, and etc. By implement this system, handicapped persons will able to use computer completely.

5. REFERENCES

1. Zhu Hao, Qianwei Lei, "Vision-Based Interface: Using Face and Eye Blinking Tracking with Camera," iita, vol. 1, pp.306-310, 2008 Second International Symposium on Intelligent Information Technology Application, 2008.

2. John J. Magee, Matthew R. Scott, Benjamin N. Waber, Margrit Betke, "EyeKeys: A Real-Time Vision Interface Based on Gaze Detection from a Low-Grade Video Camera," cvprw,

vol. 10, pp.159, 2004 Conference on Computer Vision and Pattern Recognition Workshop (CVPRW'04) Volume 10, 2004.

3. ChangZheng Li, Chung-Kyue Kim, Jong-Seung Park, "The Indirect Keyboard Control System by Using the Gaze Tracing Based on Haar Classifier in OpenCV," ifita, vol. 2, pp.362-366, 2009 International Forum on Information Technology and Applications, 2009.

4. Yun Fu, Thomas S. Huang, "hMouse: Head Tracking Driven Virtual Computer Mouse," wacv, pp.30, Eighth IEEE Workshop on Applications of Computer Vision (WACV'07), 2007

5. Hailing Zhou, Lijun Xie, Xuliang Fang, "Visual Mouse: SIFT Detection and PCA Recognition," cisw, pp.263-266, 2007 International Conference on Computational Intelligence and Security Workshops (CISW 2007), 2007.

6. Jilin Tu, Thomas Huang, Hai Tao, "Face as Mouse Through Visual Face Tracking," crv, pp.339-346, The 2nd Canadian Conference on Computer and Robot Vision (CRV'05), 2005.

7. M.E. Erdem, I.A. Erdem, V. Atalay, A.E. Cetin, "Computer vision based unistroke keyboard system and mouse for the handicapped," icme, vol. 2, pp.765-768, 2003 International Conference on Multimedia and Expo - Volume 2 (ICME '03), 2003

8. Gary Bradski, Andrian Kaebler: ”Learning Computer Vision with the OpenCV Library”, O’REILLY, 341-348, (2008).

9. National Institute of Neurological Disorders and Stroke, http://www.ninds.nih.gov/disorders/sci/detail_sci.htm

10. Christopher & Dana Reeve Foundation, http://www.christopherreeve.org