FaceController: A Facial Feature Based Hands-free Interface for Computer Accessibility

Yuwei Jiao
University of Waterloo
Waterloo, On, Canada
jyuwei@uwaterloo.ca

ABSTRACT

FaceController is a face tracking based system that enables computer manipulations with simple facial expressions and enhances the computer and software accessibility, especially for the disabled. Beyond a camera on the top of a computer monitor, there is no other hardware requirement of the system. Supplemented by the power of face detection in OpenCV, FaceController provides a complete solution for building hands-free input system. Foremost, FaceController allows people to move the cursor, click on buttons, control scroll bar, and type text via virtual keyboard in the screen with facial movements. Users can also set shortcuts for commands like opening/closing file or switching between panels with expressions like smiling or sad face. FaceController has an innovative capability of sending system-wide click and keystroke signals. Thus, it is conceivable that FaceController would provide another solution for the disabled to manipulate any software, and it could become the next generation input device.

Author Keywords

face tracking, hands-free control, input device

INTRODUCTION

Mouse and keyboard have been principal computer input devices for several years. It is undeniable that they provide easy, accurate and sensible controls over computers. However, input accessibility is far from solved. Firstly, it is rather frustrating and exhausting if users need to switch between mouse and keyboard constantly for a period of time. Secondly, mouse and keyboard are not elegant solutions for direct manipulation as there remains a distance between hands and eyes. Therefore users have to handle it carefully on what they do and what they see. Thirdly, as input devices, mouse and keyboard are not accessible to the disabled. As a result, traditional input devices somehow impair the accessibility and user experience, and also prevent people from realizing the full potential of computers.

With the increase of machine power and decrease of camera cost, vision technology provides a way for a new generation input device. To be operational, vision based input interface requires to be "affordable, flexible, precise and robust"[5]. A few researchers have explored the possibility of using facial features to control computers and developed hands-free mouse replacements. However, most prototypes use the movement of eyes and nose, or markings on user's face to navigate the cursor.

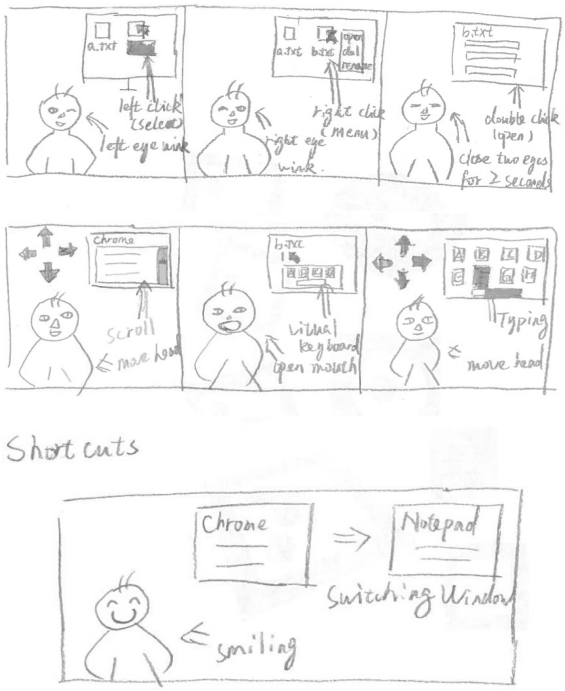

Figure 1. Sketch to illustrate FaceController

These solutions don't take fully advantage of the information delivered by human's facial expressions. There is still a large possible space unexplored. Furthermore, the functionality of conventional input devices is weakened.

In this paper, I present FaceController, a novel facial tracking based system that enables most software manipulations with basic human expressions (see Figure 1). The system-wide implementation allows users to move the cursor, click on buttons, control scroll bar and type characters directly via various facial expressions. Each facial movement is bound with an input command in the system. Currently FaceController is able to detect when users wink, raise either eyebrow, open their mouths and move heads around. And it can also recognize combinations of facial movements, which enlarges expressing space and improves the flexibility.

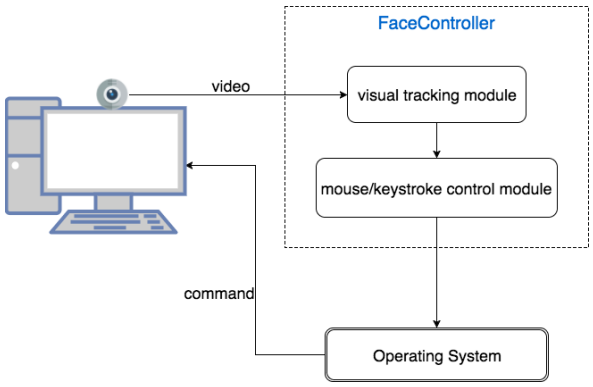

Figure 2. The framework of FaceController

Movement	
left / right eye wink	left / right button click
close two eyes (> 2s)	double click
open month	activate / deactivate virtual keyboard
move closer to screen	activate scroll mode
move head up / down / left / right	default mode: move the cursor
	scroll mode: scroll panel
	typing mode: typing alphabet
raise left / right eyebrow	switch left / right windows
Expression	
smiling face	zoom in
sad face	zoom out
angry face	return to desktop

Table 1. Facial movements and expressions commands

In my implementation, I first applied a variety of face tracking algorithms in an attempt to find the optimal method in regard to accuracy and complexity. Since the appearance of facial features differs among people, tracking performance varies among individuals. Besides, enviromental factors, for example, lighting, might affect the performance as well. Therefore it's necessary to take those factors into account. Next step was to project from 3D facial movement to 2D cursor position. FaceController provides three mouse cursor control modes: direct mode, joystick mode and differential mode. And I will explore more on hands-free typing in the future. I asked 10 students from University of Waterloo to play with the prototype and give their feedbacks. Then I did a timing comparison between regular input device and FaceController in scenarios of both gaming and working. Finally I present the discussions, as well as strengths and limitations of my prototype.

RELATED WORK

To be finished

[5] [10] [9] [8] [11] [2] [4] [1] [7] [3] [6]

IMPLEMENTATION

Framework

The framework of FaceController is shown in Figure 2. No other hardware requirement is needed beyond a commodity computer with camera on the top. There are at least three components in my system: the operating system to handle commands, the visual tracking module and the mouse/keystroke control module. At present, FaceController is built on Mac OS Sierra 10.12.3 and uses the built-in webcam (720p FaceTime HD camera).

Language, Libraries, Tookits

FaceController is developed using Python 3.5.1 and takes the advantage of OpenCV (3.2.0) and Numpy (1.11.2) to handle image signals. To control the mouse and keyboard programmatically, I adopt PyAutoGui (3.2.1) in FaceController. PyAutoGui is a cross-platform Python module which can simulate moving, clicking, dragging the mouse and pressing, releasing and holding keys. It even supports pressing keyboard hotkey combinations, which enables the functions of shortcuts in my implementation. See `http://pyautogui.readthedocs.io/en/latest/index.html` for more information about PyAutoGui.

Face Detection Algorithms

Face Detection is the first step in the system (see Figure 4). There are several face detection algorithms to locate a human face in the screen. Check out `http://www.face-rec.org/algorithms/`. In my implementation, I applied the following techiniques:

- Principal Component Analysis (PCA)
- Haar Feature-based Cascade Classifiers
- ...

To be finished

DISCUSSION

To be finished

FUTURE WORK

To be finished

REFERENCES

1. Kohei Arai and Ronny Mardiyanto. 2010. Camera as mouse and keyboard for handicap person with troubleshooting ability, recovery, and complete mouse events. *International Journal of Human Computer Interaction (IJHCI)* 1, 3 (2010), 46.

2. Margrit Betke, James Gips, and Peter Fleming. 2002. The camera mouse: visual tracking of body features to provide computer access for people with severe disabilities. *IEEE Transactions on neural systems and Rehabilitation Engineering* 10, 1 (2002), 1–10.

3. RL Cloud, Margrit Betke, and James Gips. 2002. Experiments with a camera-based human-computer interface system. In *Proceedings of the 7th ERCIM Workshop" User Interfaces for All," UI4ALL*. 103–110.

4. James Gips, Margrit Betke, and Peter Fleming. 2000. The Camera Mouse: Preliminary investigation of automated visual tracking for computer access. In *In Proc. Conf. on Rehabilitation Engineering and Assistive Technology Society of North America*. 98–100.

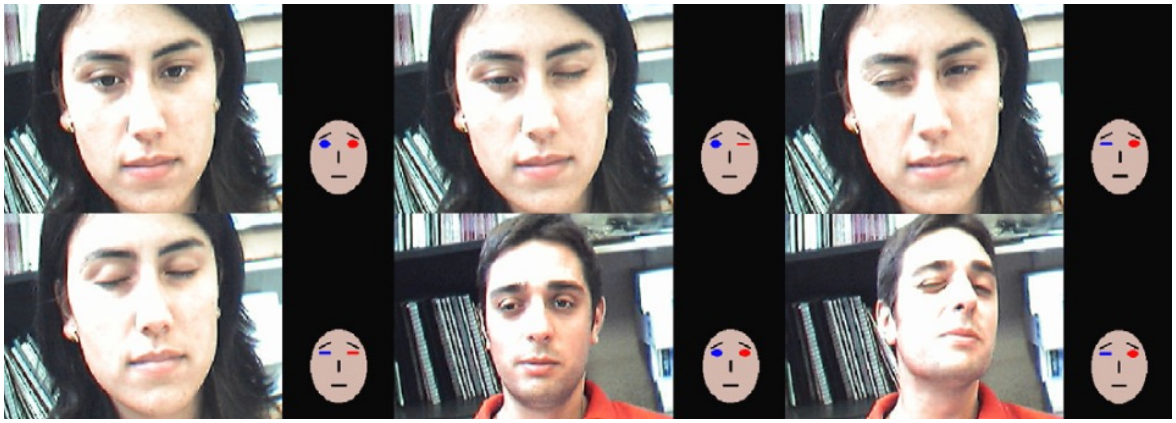

Figure 3. Wink detection [11]

5. Dmitry O Gorodnichy and Gerhard Roth. 2004. Nouse 'use your nose as a mouse' perceptual vision technology for hands-free games and interfaces. *Image and Vision Computing* 22, 12 (2004), 931–942.
6. Ricardo Jota and Daniel Wigdor. 2015. Palpebrae superioris: exploring the design space of eyelid gestures. In *Proceedings of the 41st Graphics Interface Conference*. Canadian Information Processing Society, 273–280.
7. John J Magee, Matthew R Scott, Benjamin N Waber, and Margrit Betke. 2004. Eyekeys: A real-time vision interface based on gaze detection from a low-grade video camera. In *Computer Vision and Pattern Recognition Workshop, 2004. CVPRW'04. Conference on*. IEEE, 159–159.
8. Tim Morris and V Chauhan. 2006. Facial feature tracking for cursor control. *Journal of Network and Computer Applications* 29, 1 (2006), 62–80.
9. Kaushik Parmar, Bhavin Mehta, and Rupali Sawant. 2012. Facial-feature based Human-Computer Interface for disabled people. In *Communication, Information & Computing Technology (ICCICT), 2012 International Conference on*. IEEE, 1–5.
10. Jilin Tu, Hai Tao, and Thomas Huang. 2007. Face as mouse through visual face tracking. *Computer Vision and Image Understanding* 108, 1 (2007), 35–40.
11. Javier Varona, Cristina Manresa-Yee, and Francisco J Perales. 2008. Hands-free vision-based interface for computer accessibility. *Journal of Network and Computer Applications* 31, 4 (2008), 357–374.

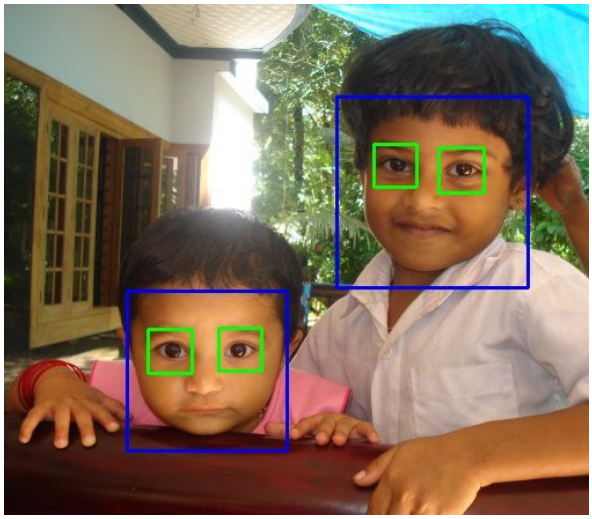

Figure 4. Face detection results using Haar Cascades in OpenCV