FaceSwitch - Low-Cost Accessibility Software for Computer Control Combining Gaze Interaction and Face Gestures

David Rozado
College of Enterprise and Development
Otago Polytechnic
Forth St. Dunedin, New Zealand
david.rozado@op.ac.nz

Jason Niu
DATA61
CSIRO
15 College Road, Hobart, Australia
jason.qniu@gmail.com

Andreas Duenser
DATA61
CSIRO
15 College Road, Hobart, Australia
Andreas.Duenser@csiro.au

ABSTRACT

We introduce here the FaceSwitch, an accessibility software system designed to facilitate computer interaction for users who are challenged in the mobility of their upper limbs. The FaceSwitch software tracks landmark features in a user's face using a deformable face tracker. The system lets the user map specific facial gestures to customized computer control commands such as left click, right-click, page down or escape through a convenient GUI. Hence, facial gestures act as substitutes of traditional mechanical switches. When combining facial gestures with gaze interaction, the emergent multimodal interaction paradigm improves the degrees of freedom offered by alternative accessibility software such as gaze only interaction, speech interaction or mechanical single-switch assisted gaze interaction. Furthermore, the FaceSwitch software improves the efficacy of traditional gaze interaction which has been traditionally limited by a high rate of false positives due to its dependence on target acquisition via dwell time activation. The FaceSwitch also reduces the latency to achieve a computer control task when compared to traditional accessibility software. We have made the FaceSwitch widely available in order for those in need to use it without restrictions and also to allow those with the right technical skills to potentially improve the software further.

Author Keywords

Multimodal interaction, Gaze interaction, accessibility, assistive technologies, Facial gestures

ACM Classification Keywords

H.5.2 User Interfaces

INTRODUCTION

Interaction with computers is challenging for users with motor impairment. Motor impairment can originate fromtraumatic injuries, congenital or degenerative conditions.

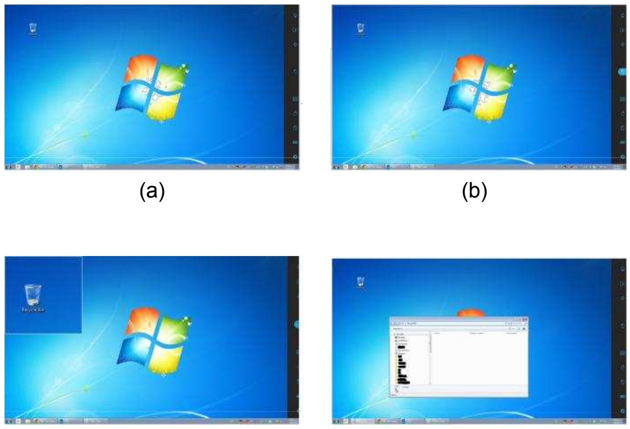
Mouse control for target acquisition is severely affected for users with motor impairment of the upper limbs. Assistive technologies for computer control strive to provide motor disabled subjects with practical solutions to interact with electronic devices. Unfortunately however, such technologies can often not match the information transfer rate between the human and the computer that traditional interaction with the mouse and the keyboard can achieve (Yuan et al., 2013). In this work, we present the FaceSwitch (https://github.com/Jason-NIU/Face-switch) system that combines the monitoring and recognition of facial gestures with gaze interaction to create a multimodal interaction modality that surpasses traditional gaze interaction or mechanical single switch assisted gaze interaction.

Gaze interaction has been very useful for subjects withmotor impairment to interact with computers via gaze. Unfortunately, this interaction modality suffers its own drawbacks such as an accuracy limit of about 0.5 degrees of visual angle (Rozado et al., 2013) at 60cm from the screen. Furthermore, gaze only interaction is afflicted by a high number of false-positives during target acquisition and low information transfer rates (Rozado et al., 2011) between the user and the computer. Mechanical switches to asynchronously signal click intention have been proposed to complement gaze interaction (Grauman et al., 2003). The usage of a switch however is not always feasible for users with very advanced degrees of motor disabilities. Furthermore, mechanical switches don't scale well, since motor impaired users are unlikely to be able to use several switches simultaneously. As a consequence, gaze only or single-switch complemented gaze interaction are significantly slower and more error prone for HCI purposes than standard mouse based interaction (Spakov et al., 2004).

The accuracy limitations of gaze interaction has often been circumvented in the past by the usage of customized software which is gaze responsive. This however severely limits the range of software available for users dependent on gaze for interaction. Traditionally, standard application control has been achieved by placing the mouse cursor on the point of regard of the user on the screen. This is inconvenient for the user since the offset

OzCHI '15 , December 07 - 10 2015, Melbourne, VIC, Australia
Copyright © 2015 ACM 978-1-4503-3673-4/15/12... $15.00
http://dx.doi.org/10.1145/2838739.2838809.

between actual gaze position and inferred gaze position is nerving for the user (Porta et al., 2010) to observe in such configuration. Alternatives such as using gaze gestures for control and interaction instead of just mapping gaze estimation coordinates to cursor position have been suggested (Rozado et al., 2012).



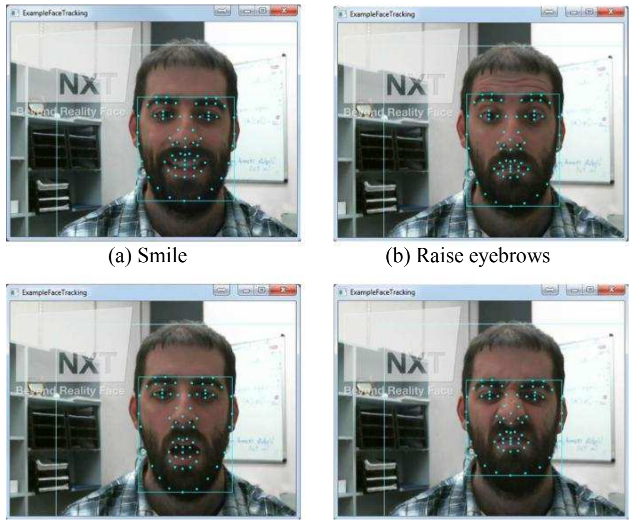
(a) (b) (c) (d)

Figure 1: Tobii Gaze Selection. In 1a, the user can freely inspect the screen with no risk of accidentally triggering a command via gaze. In 1b, the user has activate via gazed the left click action command. In 1c, the user gazes at its desired target location, using dwell time or a switch, a zooming mechanism is superimposed on the screen in order for the user to precisely acquire small targets. In 1d the action has been completed.

To make gaze interaction possible with traditional GUIs displaying small targets (Pomplun et al., 2001) proposed using a zoom mechanism. Commercial developers such as Tobii have quickly adopted this form of gaze interaction with for instance their Windows Gaze Control Software. This selection modality consists of the user selecting a desired interaction task (right click, left click, scroll, etc) by looking at an icon in a docked task bar on the edge of the monitor, see Figure 1, followed by the user gazing at the desired portion of the screen in which it wants to execute the previously selected task. A dynamic zoom towards the area surrounding the region of interest is then triggered by a switch or by using a dwell time threshold activation. The user can steer the dynamic zoom towards the desired target by using gaze corrections on the zoom trajectory to allow for fine grained target selection. This mode of gaze-based Windows control makes it possible for the user to control a standard Windows desktop operating system with a two step selection method. Although an improvement over traditional gaze only interaction, this multimodal approach even when combined with a single mechanical switch still provides only 1 degree of freedom, which constraints the speed of the interaction. The usage of additional mechanical switches to increase the degrees of freedom afforded by the user is severally limited by the motor impairment of the target users which often can hardly control several upper limb muscles efficiently. In this work, we propose to enhance gaze interaction by using the face as a multi-switch system and allowing the user to customize mappings of different gestures to different computer control commands. This eliminates the need of having a docked task bar which consumes screen real estate. Furthermore, eliminating the need for gazing at the taskbar to preselect an interaction command and allowing facial gestures to be generated asynchronously, speeds up interaction speed.

In this work we investigate the combination of gaze interaction with face gestures (Yuan et al., 2013) acting as a multi-switch system.

We use a deformable face tracker to track a user's face during interaction with the computer. The face tracker allows the system to track a number of landmark features on the face such as the eyebrow, nose or mouth contours. Time series of subsets of those landmarks can define specific facial gestures such as opening the mouth,



(a) Smile (b) Raise eyebrows (c) Open mouth (d) Twitch nose

Figure 2: Beyond Reality Face tracking software. The subfigures illustrate how the deformable face tracking software adapts the mesh of points to different facial configurations representing several facial gestures.

twitching the nose, smiling or raising the eyebrows. Specific gestures are mapped to specific control commands such as left mouse click, right mouse click, scroll, page down, enter or any combinations of mouse and keyboard that the end user specifies. When combining the face gestures with gaze tracking, a new multi-modal interaction method emerges that consists of using gaze for pointing at objects of interest and face gestures as switches that trigger specific interactions commands on the computer.

THE FACESWITCH

The FaceSwitch system is dependent on an eye tracker to carry out gaze estimation with a gaze estimation accuracy of around 0.5 degrees of visual angle at 60cm from the screen. A low cost web camera is also needed to provide a continuous video stream of subjects faces to the subsequently described face tracking software. In our work, we used a Tobii X2-30 Eye tracker and a Logitech HD 1080p WebCam. Tobii's Gaze Control software was used for gaze interaction purposes. This software provides gaze interaction capabilities by placing a taskbar docked on the right edge of the monitor, see Figure 1 that displays different interaction tasks (right click, left click, scroll, etc). The user activates a task by gazing at it. Then the user can gaze at the area of the screen with which it wants to interact using the previously gaze selected task. A dwell triggered zoom function facilitates the acquisition of small targets. Throughout the zooming process, the user can steer via gaze the direction of the zoom for fine grained control of target acquisition. After the zoom is completed, the pre-selected control command from the task bar is exercised in the zoomed in area. This gaze based mode of Windows Control makes it possible for the user to control a standard Windows desktop operating system with a two step selection method. The system can be extended with an external switch for users who are able to mechanically operate a switch. This eliminates the need for dwell based target acquisition. The FaceSwitch system uses the Beyond Reality deformable face tracker (https://www.beyond-reality-face.com) to track a user's face during interaction with the computer. The face tracker tracks several landmark features on the face. Time series of subsets of those landmarks define specific facial gestures such as a smile, opening the mouth, twitching the nose or raising the eyebrows, see Figure 2. We modified the face tracking software to stream the time series of points tracking facial landmarks using the Lab Streaming Layer, a system for the unified collection of time series that handles both the networking, time-synchronization, (near) real-time access as well as optionally the centralized collection, viewing and disk recording of the data. A gesture recognition engine is used to detect facial gestures in real-time and map them to specific control commands. The mapping of a facial gesture to a specific interaction command (keyboard or mouse events) can be defined by the user using the FaceSwitch user friendly Graphical User Interface, see Figure 3.

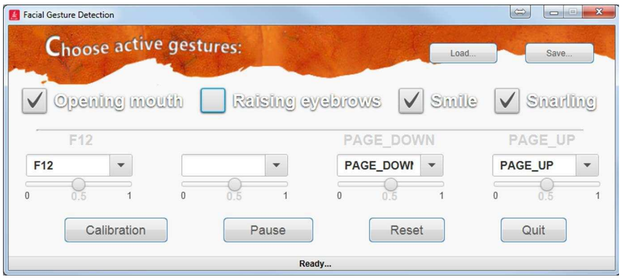


Figure 3: The Graphical user interface of the FaceSwitch allows the user to select which gestures are to be actively monitored as well as to map each gesture to a customize keyboard or mouse event.

The FaceSwitch system generates control commands when a facial gesture is detected by triggering the specific keyboard event that has been mapped to the aforementioned facial gesture. A video demo of the system at work is provided (https://www.youtube.com/watch?v=0vOrGOliwR8&feature=youtu.be). The video demo shows the high interaction speed achieved with the FaceSwitch interaction modality.

The facial gesture recognition engine of the FaceSwitch relies on monitoring facial landmark features in the stream of data generated by the face tracker. Our system compares distances between facial landmark points such as the top of the nose and the top of the mouth from previously established thresholds specific to a subject upon firing the application. This permits the detection of predefined facial gestures such as raising the eyebrows or a smile. The FaceSwitch software then maps the recognized gesture to a predefined control command such as left mouse click, right mouse click, page down or page up. The software does this by means of generating a keyboard event. To prevent unexpected multiple input commands, the application generates a key press and when the eyebrows return to a neutral position the software generates the keyboard event of releasing the key thus completing the control command. For detecting for instance the opening of the mouth in an o-like shape, three basic features are used: The Euclidean distance between predefined points specific to a gesture (i.e. point 51 and point 57 in the face tracking mesh), a threshold for judging mouth status (closed or open) and a baseline distance between those two points (51 and 57) while the user has a neutral face during a less than 1 second calibration procedure during program start up. This later number is used as a correction factor for monitoring gesture detection according to user head movement towards or away from the screen since the distance between points 51 and 57 on the image captured by the Web camera is heavily influenced by the distance of the face to the camera. The distance correction factor is estimated as a fraction of the current distance between the 2 reference points and the initial distance gathered during the calibration procedure. The distance between the 2 reference points for a gesture is continuously monitored in real-time only if that gesture is actively being monitored, i.e. activated by the user or assistant in the Application GUI. Additionally, since everyone's facial gestures differ in their intrinsic mechanics and metrics, a sensitivity function is used in order for different users to adjust the sensitivity of the software to their own characteristic facial gestures. In this way, users can also adjust for each gesture through the applications GUI the sensitivity of the system to gesture detection. For all the other gestures that can be monitored (eyebrows raising detection, smile detection and nose twitch), a similar procedure is followed.

DISCUSSION

We have presented a software system with the ability to detect facial gestures and map them to customized computer control commands. When combined with gaze estimation, a multimodal interaction modality emerges that provides more degrees of freedom than traditional accessibility software. The FaceSwitch allows any user to control a computer fast and reliably using only the head. The software is presented to the user via a user-friendly GUI that permits the customization of the software to particular user circumstances. The advantages of a video-based switch system over more traditional mechanical switches and even EMG switches that can be attached to muscles are obvious. EMG switches often involve a bundle of cables and require attachment to functioning muscles which can feel intrusive for the user and unfeasible for those who have no muscle control from the neck down. Even for users who can use facial musculature, placing EMG sensor electrodes over the face or making the user operate a mechanical switch with the head is uncomfortable and not scalable to more than one or a couple of switches. Our proposed video based FaceSwitch concept permits unobtrusive switch interaction and a scalable system where up to 4 facial gestures can be tracked simultaneously and mapped to different control commands hence augmenting the amount of degrees of freedom available to the user. A key component for the ability of the FaceSwitch to successfully recognize facial gestures is the robustness of the deformable face tracker monitoring facial features that we used. The FaceSwitch relies on the Next Generation Face Tracker, which showed to be a very robust and free deformable face tracker. It's only limitation being its inability to track more than a reduce number of facial landmarks which leaves out important landmarks such as the cheeks or the intrinsic forehead musculature which could be very predictive of complex facial gestures. Being able to interact fast and reliably with a computer is as important for users with special needs as it is for anyone else. The speed of this interaction can be captured and quantified in the form of the Information Transfer Rate statistic. It is of utmost importance, that accessibility software brings interaction speed for users with special needs to a standard as close as possible to traditional interaction modalities since many activities of work and leisure are nowadays performed through computers: banking, communication, entertainment, information acquisition and even working. An interaction paradigm that requires twice as much time to complete a standard task as another interaction paradigm means that a user that would carry out its usual computing activities in about 8 hours, would need 16 hours to carry out the same tasks. Hence, accessibility solutions that are half as fast as traditional interaction modalities, while useful, are often not enough in terms of accessibility performance. Hence, the importance of improving the communication bandwidth between the human and the computer is paramount.

One of the main limitations of the FaceSwitch is that facial gesture interaction with the computer has the potential to interfere with the natural expressiveness of the user's face. This could manifest itself in the form of the user expressing genuine surprise at some content presented in the computer display and the FaceSwitch system running in the background misrecognizing such pattern as one of the interaction gestures that is actively monitoring. Hence, the system imposes the constraint on the user of interacting with the computer using a neutral facial expression and only reserving facial gestures for active control. Every interaction modality involves a number of tradeoffs, and the FaceSwitch is no exception to that. However, due to the lack of efficient and fast interaction options for users paralyzed from the neck down, we believe that the benefits that the system offers surpass this relatively small cost.

We have made our software open source so anyone can benefit from it, assuming they can afford the cost of the eye tracker and the webcam. Fortunately, these devices have massively come down in price recently and there are already companies offering an eye tracker for only $99. Another reason for providing the software in an open source format is to invite those with the technical skills and the interest to push the boundaries of accessibility technologies further to take on our software and try to improve it in order to achieve higher information transfer rates by for instance increasing the number of gestures that can be robustly recognized simultaneously.

Future work should quantitatively explore the performance metrics of the FaceSwitch system against alternative accessibility options and standard HCI input modalities. This could include analyzing through a proper user study the accuracy metrics for different facial gestures, the maximum number of facial gestures that can be reliably and simultaneously tracked without taking a hit on performance as well as the influence of false positive activation on the reliability of the entire system. Furthermore, improving the system presented here to allow for the simultaneous monitoring of more than 4 facial gestures could augment the interaction possibilities of the system even further by allowing the additional gestures to be mapped to an expanded set of control commands. This would further enhance the interaction efficiency and the information transfer rate between the human and the computer.

REFERENCES

Grauman, K., Betke, M., Lombardi, J., Gips, J., and Bradski, G. R. Communication via eye blinks and eyebrow raises: video-based human-computer interfaces. *Universal Access in the Information Society* 2, 4 (Oct. 2003), 359-373.

Pomplun, M., Ivanovic, N., Reingold, E. M., and Shen, J. Empirical evaluation of a novel gaze-controlled zooming interface. *In Proceedings of HCI International* (2001).

Porta, M., Ravarelli, A., and Spagnoli, G. ceCursor, a contextual eye cursor for general pointing in windows environments. *In Proceedings of the 2010 Symposium on Eye-Tracking Research & Applications, ETRA '10*, ACM (New York, NY, USA, 2010), 333-337.

Rozado, D. Mouse and keyboard cursor warping to accelerate and reduce the effort of routine HCI input tasks. *IEEE Transactions on Human-Machine Systems 43, 5* (Sept. 2013), 487-493.

Rozado, D., Agustin, J. S., Rodriguez, F. B., and Varona, P. Gliding and saccadic gaze gesture recognition in real time. *ACM Trans. Interact. Intell. Syst. 1,* 2 (Jan. 2012), 10:1-10:27.

[Rozado, D., San Agustin, J., Rodriguez, F., and Varona, P. Gliding and saccadic gaze gesture recognition in real time. *ACM Transactions on Intelligent Interactive Systems* (2011).

[Spakov, O., and Miniotas, D. On-line adjustment of dwell time for target selection by gaze. In Proceedings of the *Third Nordic Conference on Human-computer Interaction*, NordiCHI '04, ACM (New York, NY, USA,
2004), 203-206.

Yuan, P., Gao, X., Allison, B., Wang, Y., Bin, G., and
Gao, S. A study of the existing problems of estimating the information transfer rate in online brain-computer interfaces. *Journal of Neural Engineering 10,* 2 (Apr. 2013), 026014.