FaceTech: An Operation of Using the Distance Between Face and Computer

Yu Ta, Chen
Taipei, Taiwan
R07922120@ntu.edu.tw

Wan-Ting Huang
Taipei, Taiwan
R07922024@ntu.edu.tw

ABSTRACT

Most people use the mouse and keyboard to operate the computer. However, in long-term use, the users' hand joints will be greatly hurt. In order to solve the hand soreness problems, we proposed an operation method - FaceTech, which operate on the computer according to the distance between users' face and the computer. Such an operation method can assist or combine the general operation method for reducing the time of using computers by users' hands, thereby it can reduce the soreness caused by long-term use of computers. The main contribution of this research is that we proposed a novel way to operate the computers which can reduce the degree of soreness of the hands when they input into the computers.

Author Keywords

Hand soreness, posture, face detection, assistive technology.

CSS Concepts

• **Human-centered computing~Interaction techniques**

INTRODUCTION

Traditionally, people usually use mouses to operate the computer, it can potentially damage their hands, like Carpal Tunnel Syndrome (CTS). Our motivation for this work is to explore a new way that assists the traditional way of using computers. We aimed at reducing hand soreness caused by using mouse and keyboard. Therefore, we proposed FaceTech, a novel way to operate the computer by detecting the distance between face and computer. It allows users to reduce their hand usage. We used OpenCV Face Detection [1] to detect the user's face. When the user leans toward the computer, the screen layout will zoom out; and when the user leans backward, the screen layout will zoom in.

The result of our user study has shown that by using our system, the soreness of the hand by using computer can reduce significantly. The contributions of this work are:

- Proposed an operation method that assists the traditional way.
- Proposed some tips to improve the accuracy of OpenCV Face Detection [5].
- Our approaches can reduce hand injuries caused by traditional methods.

RELATED WORK

The papers we refer to are divided into two types, one is to improve the accuracy of head and face tracking, and the other is to use the head and face gestures to operate.

Improve Recognition

There are many studies on how to make computers track the movements of the face or head more accurately. Morency et al. [4] used stereo camera to get depth information, and make head detection more accurately. Or used the learning method to make it more accurately [3]. These types of research will require users to have high-end hardware devices, such as stereo camera or high-level GPU. However, we hoped our method can be easily approached by general public. Therefore, we will explain later what the much simpler way we apply in our method.

Head Gesture

Many authors have proposed some head or face gestures. Harrison et al. [1] developed a system called "Lean and Zoom", when user leans toward the computer, the on-screen content will be magnified. Rick Kjeldsen [2] proposed using a nod or turning head to input to the computer. Yet, both of them did not analyze whether their method can reduce users' hand usage. Therefore, we designed a simple head gesture based on these papers. After users doing head gesture, we will evaluate whether they feel better than traditional operating way through their hand soreness.

SYSTEM IMPLEMENTATION

We designed two methods in total. Both of them were based on the distance between the user and the computer. The equipment and methods we used are easily to be approached by most people. In addition, we used some programs or web page layouts to demonstrate how we can use our operations to change these layouts. Next, we will introduce the steps we have taken.

Face Detection

First, we used the webcam of the general notebook to capture the frames of the users. Then used OpenCV Face Detection [6] to detect the bounding box of users' faces. The reason for using OpenCV was that we want our operation to be real-time, and the speed of OpenCV Face Detection can achieve this goal. In addition, if use the learning base to detect the

CHI 2020, April 25–30, 2020, Honolulu, HI, USA.

ACM ISBN 978-1-4503-6708-0/20/04...$15.00.
DOI: https://doi.org/10.1145/3313831.XXXXXXX
*update the above block and DOI per your rightsreview confirmation (provided after acceptance)

face [8], users may need a high-level GPU to complete the detection quickly. But most general users will not have this kind of high-end devices.

Compute Distance

After detecting the face, we used the bounding box of the face to calculate the distance from the computer. As we used webcam for shooting, we cannot get the depth information. Therefore, the method of calculating the distance we used is to calculate the area of the bounding box of the user's face. If the area of the bounding box is larger, the closer the user is to the computer, if not, then the further the distance.

Operation: Two Stage

The first operation mode is *Two Stage*. As the name implies, it has two states in total. First, the user will maintain the usual posture of using the computer before applying this operation method. Thus, the computer can capture the bounding box sizes of the first five frames. Then we averaged these five sizes to get an initial state, we called it *far*. Finally, a threshold α was given. When the size of the current bounding box is larger than the size of the far state multiplied by α, it is considered to be in the near state.

Operation: Three Stage

The second operation mode is *Three Stage*. This operation differs from *Two Stage* is that it has three states. First, the user will still maintain the usual posture of using the computer for capturing the bounding box sizes of the first five frames. Then we averaged these five sizes to get an initial state, we called it *middle*. Finally, two thresholds α and β were given. When the size of the current bounding box is larger than the size of the middle state multiplied by α, it is considered to be in the near state. When the size is smaller than the size of middle state multiplied by β, it is regarded as the far state. The thresholds we used will show in the section of User Study.

Layout

We wrote our demo layout on the web. For text layout, we took Visual Studio Code as an example. When the state is detected as *near*, the system changes the word size to 18px, and display the panel on the left (Figure 1 left). Thus, the user can select another file without clicking the menu button first. When the state is detected as *far*, the system changes the word size to 24px, and hide the panel on the left (Figure 1 right).

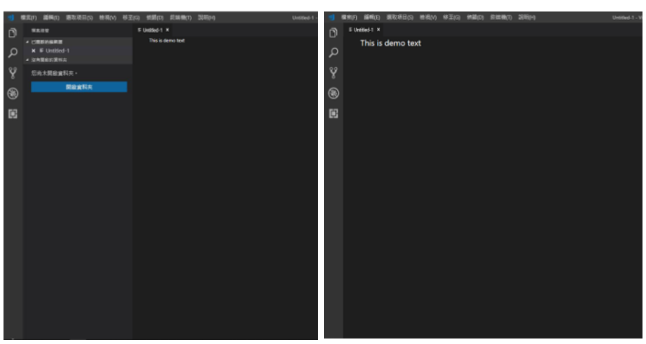

Figure 1. Visual Studio Code.

As for image layout, we imitated the form of the blog. When the state is detected as *near*, the image is small and the user can see many different pictures at once (Figure 2 left). Then the user can move the mouse to the image that he or she is interested in. Next, change his or her pose to *far*, the selected image will be enlarged and the information of the image will be shown (Figure 2 right).

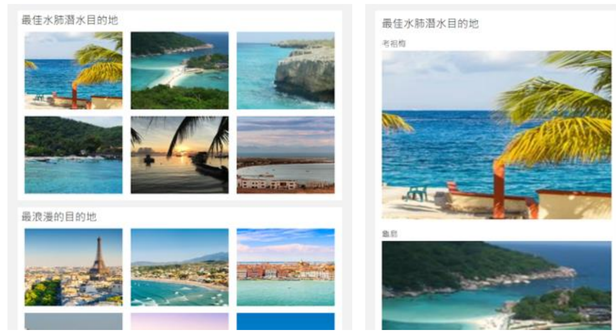

Figure 2. Blog.

We used YouTube as an example of video layout. When the state is detected as *far*, the video enters full screen (Figure 3 left). When the state is detected as *near*, the video returns to its original size, the user can select other videos or scroll down to see the comments as usual (Figure 3 right).

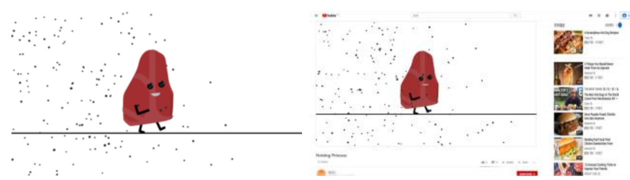

Figure 3. YouTube.

For websites that have different layouts and different appearances according to different content, we chose Facebook webpage as an example. In a post with many texts, a short paragraph of text will be displayed first and follow with "...more". The user needs to click "...more" to see the full article. But when he or she does not want to read till the end, he or she has to slide down until the end of this post to see the next post. With our system, the user can trigger *far* to see the full article (Figure 4 left), and when they don't want to read the article anymore, they can lean toward the computer to trigger *near*, the article will be closed (Figure 4 right), and they can scroll to another post faster.

Figure 4. Facebook - Text

For image-based post, the layout is as same as original, the user does not need to move the mouse to click on the image to enlarge it, he or she just need to lean backward to trigger *far* (Figure 5 left). To go back to the original post, the user just needs to lean forward to trigger *near* (Figure 5 right).

Figure 5. Facebook – Image.

For posts that contains video, the user needs to click on the play button to let the video start playing and then trigger *far* to enter full screen (Figure 6 left). To go back to the original post, the user just needs to lean forward to trigger *near* (Figure 6 right).

Figure 6. Facebook – Video.

SYSTEM IMPROVEMENT

The higher accuracy of face detection allows our system to perform better and make it easier for using. Therefore, we used some tips to improve accuracy. In this section, we will introduce the tips we use, and compare the accuracy of our tips with the original face detection.

Eyes Detection

Since the wrong face may be detected, as in Figure 7. When we detect the face of the image, there are two detected faces that are false positive, which will cause our face area calculation error. Therefore, we proposed a way to improve the accuracy of the predicted face, this method is using eyes detection with OpenCV [5] to detect the eyes of the face (Figure 8).

Figure 7. False Positive of Face Detection

Figure 8. Tips: Eyes Detection

Comparing with Previous Face

There is a disadvantage of Eyes Detection. If the user blinks his or her eyes or leans down slightly, it may cause the eyes to be undetectable, and we still cannot recognize the true face size of the user. Therefore, we proposed another method to compare all the faces in the frame with the face detected by the previous frame. The one with the highest similarity is regarded as the real face (Figure 9). The comparison method is comparing the bounding boxes of the two faces by calculated their Intersection of Union (IOU), and the higher the IOU, the more similar they are.

Figure 9. Tips: Comparing with Previous Face

Evaluation

As we want to use the highest accuracy method as our detection way, we will analyze the accuracy of the method proposed above. In addition, we will also evaluate whether the time spent on detection is fast enough for our system in real-time.

Generally, the accuracy analysis of face detection is done by calculating false positive and precision [7]. However, we not only care about precision but also want to have a high recall. Because higher recall can make our system detect the user's face much easier and avoid losing track of the user's face. Therefore, we adopt f1 score as the way to measure accuracy. To evaluate whether the face is correct or not, the detected face is used to calculate the IOU with ground truth. If IOU > 0.5, this prediction is to be considered as correct.

We used YouTube Faces Database for evaluation. The source of the videos was downloaded from YouTube, and

Detection Method	Accuracy				Time (ms)
	False Positive	Precision	Recall	F1 Score	Detection Time
General	117631	84.03%	99.64%	91.17%	14.06
Eyes Detection	21397	96.55%	96.28%	96.41%	17.01
Comparing with Previous Face	7106	98.85%	98.58%	98.72%	14.23

Table 1. Comparing the Tips of Improvement.

these videos were converted into frames, and the sizes of the videos were different. This dataset contains a total of 3425 movies (621,126 frames), and only contains one face in each frame.

Result

The results of our different methods for improving face detection accuracy and time cost is shown on Table 1. In terms of time evaluation, the two tips did not need to spend too much time. According to the results, both of them are enough for our system to be real-time. In the evaluation of accuracy, both performances are much better than not doing any tips, but the results of using the method of comparing with previous face is better than the method of making eyes detection. Finally, we decided to choose *Comparing with Previous Face* as our detection method. What if the face cannot be detected in previous frame? We choose *Eyes Detection*.

USER STUDY

Pilot Study

We wrote a web page which only contains a paragraph of text, when press ‘w’ button on the keyboard, the text will get bigger, when press ‘s’ button, the text will become smaller.

In pilot study, we recruited 3 participants, we asked them to maintain their usual posture of using the computer, then press the ‘w’ and ‘s’ button on the keyboard to change text size, till the size became that they can read the paragraph fluently. Then, we asked them to press the enter button so that we can get the text size. Next, we asked them to lean toward the computer, then do the same steps again. Finally, we averaged the results and get 18px for near word size, 24px for far word size.

After getting the font size, we will also research on the distance between users and computers. We asked the participants to maintain the usual posture for capturing the bounding box sizes of the first five frames. Then we averaged these five sizes to get an initial state. After that, we set the font size to 18px, let the users move their head to see the text most clearly, and get the value of bounding box size in this state. And then set the font size to 24px, also let the users move their head to see the text most clearly, and get the value of bounding box size in this state. Finally, we averaged the values of the three subjects to get $\alpha = 1.66$ and $\beta = 0.66$.

User Study

A user study with 12 participants was performed to evaluate the performance of our approach. We guided participants how to use the operations with our demo layout on the web, and asked them to do a speed test by using the three operations: general way of using the application, FaceTech - Two Stage, FaceTech - Three Stage. The order was randomized to compensate for training and fatigue effects in the combined results. During the process, we asked them to fill out SUS questionnaire to understand their satisfaction with the system, and also filled out current pain level questionnaire to see if our methods can really reduce the pain in the hands and make sure that our methods will not make their other body parts feel pain.

Speed Test

In this test, the mission for the users is that they need to move to the page we request by using Word as fast as possible. They need to move to the page as our setting order as 8, 3, 9, 5, 10, 6, 2, 7, 4, 1. When experiencing general way, they can use any method to move. But they need to use the following method when experiencing FaceTech.

The layout of our Word is very similar to the blog. When the state is detected as *near*, each page will be smaller, four pages will be displayed in one column (Figure 10 left). The users can see many pages at a time. It is similar to Word zooming in to 50%. When the users move the mouse to the page they want to see, they can trigger *far*, the page they chose will return to 100% (Figure 10 right) and the page will be shown in the screen.

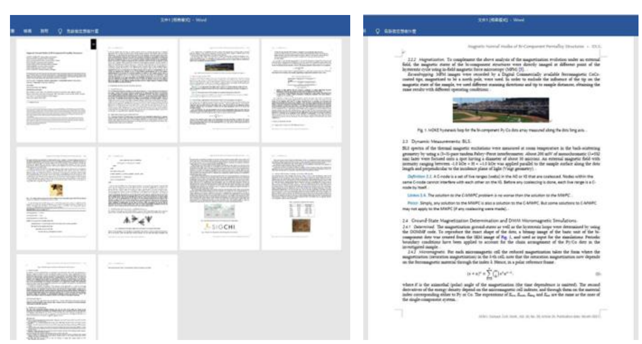

Figure 10. Word.

	FaceTech				
	(1) General	(2) Two Stage	(3) Three Stage	ANOVA	t-test
SUS	75.83	66.46	67.29	F=0.04, p=0.84	
Time	52.94	51.55	60.18	F=11.84, p<0.05*	2-3
Hand Soreness	0.92	0.17	0.25	F=6.75, p<0.05*	1-2, 1-3
Neck pain	0.17	0.67	0.92	F=2.24, p=0.13	
Waist Soreness	-0.17	0.42	0.67	F=2.64, p=0.09	
Back Pain	0.00	0.42	0.75	F=2.67, p=0.09	
Eye Soreness	0.58	0.42	0.08	F=1, p=0.38	

Table 2. The SUS score, Completion time in Word (sec), and Pain level

Procedure

In the beginning of the study, the participants filled out the questionnaire including their names and current pain level in hand, neck, waist, back and eyes. We then guided them to experience the operations, and told them what our motivation and concept are. After experiencing the operations, we asked them to do the speed test. They then filled out a System Usability Scale (SUS) questionnaire and a post-study questionnaire which is including the current pain level in hand, neck, waist, back and eyes.

Result and Discussion

Questionnaire results

The results are shown in Table 2 and Figure 11 through Figure 13. According to the results, we found that the speed of using Word with FaceTech - Two Stage is close to the speed of general method. However, users' operation speed in FaceTech - Three Stage is significantly slower than FaceTech - Two Stage. This is because the users need an extra move to change from *middle* to either *near* or *far* when using Three Stage. For hand soreness, there is significantly different between general method and the two methods we proposed, which supports our motivation to reduce hand soreness.

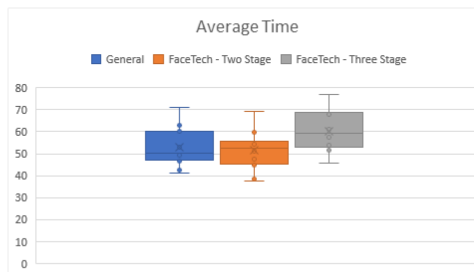

Figure 11. Average time that participants finished the task in Word

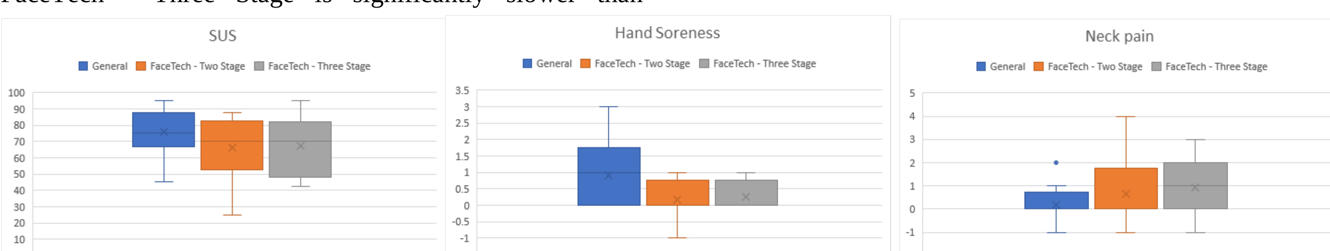

Figure 12. From left to right: Score of SUS, Score of Hand Soreness, Score of Neck pain

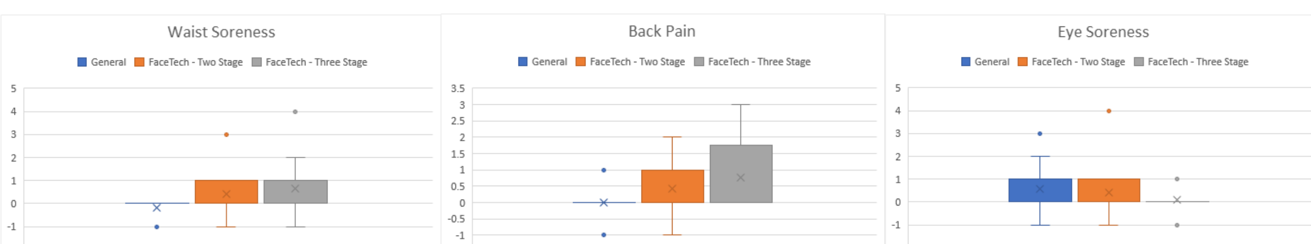

Figure 13. From left to right: Score of Waist Soreness, Score of Back Pain, Score of Eye Soreness

Feedback

Some participants thought our command and the results on displaying is not very intuitive. For example, most of participants thought that the font size should become bigger but not smaller when they lean toward the computers. Even some of them felt more convenient to use the mouse directly than using FaceTech. This is because we required them can only use FaceTech when experiencing it. However, our core concept is that FaceTech is only for assisting the traditional methods, not to replace them. Therefore, when it comes to real practice, our methods are playing an auxiliary role in real life.

Compared with FaceTech - Two Stage and FaceTech - Three Stage, some participants prefer to use FaceTech - Two Stage rather than FaceTech - Three Stage due to the faster operation speed. On the other hand, some participants like FaceTech - Three Stage better, because they can use the computers in a more comfortable posture.

CONCLUSION & FUTURE WORK

After completing the user study, we came to conclusion that the method we designed still needs to be improved. The problem is that the changing of information on the screen corresponding to FaceTech is not intuitive enough. Therefore, we need to improve the correspondence between the changing of layout and our operations.

However, after using our method, the users feel more comfortable than using the traditional method. Therefore, we can know that the operation which combined FaceTech and traditional method can really reduce the soreness of the hand. Thus, we expect that after proposing more gestures not used by hand in the future, it may be possible to reduce the number of cases of hand injuries caused by excessive use of the computer.

In the future, we will combine the advantages of the two methods that we proposed, and do more research on how to integrate FaceTech with general method. Moreover, we will conduct further research on how layout changes are more intuitive with our action.

REFERENCES

[1] Chris Harrison and Anind K. Dey. "Lean and zoom: proximity-aware user interface and content magnification." *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2008.

[2] Rick Kjeldsen. "Head gestures for computer control." *Proceedings IEEE ICCV Workshop on Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems.* IEEE, 2001.

[3] Thomas Maurer, and Christoph von der Malsburg. "Tracking and learning graphs and pose on image sequences of faces." In *Proceedings of the Second International Conference on Automatic Face and Gesture Recognition*, pp. 176-181. IEEE, 1996.

[4] Louis-Philippe Morency, Ali Rahimi, Neal Checka, and Trevor Darrell. "Fast stereo-based head tracking for interactive environments." In *Proceedings of Fifth IEEE International Conference on Automatic Face Gesture Recognition*, pp. 390-395. IEEE, 2002.

[5] OpenCV. 2014. OpenCV: Cascade Classifier. Retrieved June 18, 2019 from https://docs.opencv.org/2.4/doc/tutorials/objdetect/cascade_classifier/cascade_classifier.html

[6] OpenCV. 2018. OpenCV: Face Detection using Haar Cascades. Retrieved June 18, 2019 from https://docs.opencv.org/3.4.1/d7/d8b/tutorial_py_face_detection.html

[7] Rowley, Henry A., Shumeet Baluja, and Takeo Kanade. "Rotation invariant neural network-based face detection." (1998).

[8] Kaipeng Zhang, Zhanpeng Zhang, Zhifeng Li, and Yu Qiao. "Joint face detection and alignment using multitask cascaded convolutional networks." *IEEE Signal Processing Letters* 23, no. 10 (2016): 1499-1503.