Facial Tracking-Assisted Hand Pointing Technique for Wall-Sized Displays

Haokan Cheng(✉), Takahashi Shin, and Jiro Tanaka

University of Tsukuba, 1-1-1 Tennodai, Tsukuba, Ibaraki 305-8573, Japan
haokan@iplab.cs.tsukuba.ac.jp, {shin,jiro}@cs.tsukuba.ac.jp

Abstract. In this paper we propose a novel pointing technique leveraging the user's body motion to achieve smooth, efficient user experiences on wall-sized displays. Our proposal substantially consists of two parts: a graphical cursor controlled by the user's hand motions, and mechanisms to assist the cursor manipulation by tracking the user's face orientation. By interaction design associating the user's face and hand motions to different aspects of the cursor's movement, we aimed to bring swiftness to the interaction in large-display environments with necessary precision. A prototype was built to instantiate the concept, and two comparative experiments were conducted to evaluate the effectiveness of the proposal.

Keywords: Facial tracking · Input method · Pointing technique · Wall-sized display

1 Introduction

With decades of technological development, displays are becoming less expensive and more scalable in size, and this trend has stimulated the use of wall-sized displays (WSDs) as interactive spaces. Interactive tasks on a WSD involve basic manipulations, such as selecting and moving on-screen data representations (*e.g.,* icons, buttons, and other graphical user interface (GUI) elements), which are also common on smaller desktop or mobile displays. However, the increased amount and density of data potentially affect the task efficiency. In addition, large displays enable users to use various body parts (*e.g.,* head, hands, and feet) to control GUI elements on WSDs. In this study, we leveraged the user's face and hand motions to improve the pointing efficiency on WSDs. We focused on vertical WSDs so that the user can take full advantage of the large interactive space and the user's body, and used a conventional GUI cursor to achieve our design goal.

2 Related Work

Numerous studies have examined the means to improve the efficiency of pointing techniques. Some research has focused on optimizing the performance of conventional GUI cursors. Ninja Cursor [5] and Rake Cursor [6], for example, use multiple cursors to

N. Streitz and P. Markopoulos (Eds.): DAPI 2016, LNCS 9749, pp. 191–201, 2016.
DOI: 10.1007/978-3-319-39862-4_18

enhance the selection performance of the cursor. Multiple pointers have been used to reduce the actual distance the pointer must move, hence to improve the pointing efficiency. While sharing the same idea of shortening the cursor-to-target distance to improve the performance, we used a single cursor approach in our study.

Several research teams have also applied interaction techniques specifically designed for WSDs. Nakanishi et al. used facial tracking for manipulation on large and multiple displays [2, 3]. Nancel et al. [10] investigated multimodal, mid-air pointing techniques on very large displays, discussed on the unique requirements of pointing techniques on WSDs and provided solutions. Vogel and Balakrisham [4] developed a hand pointing technique for very large and high resolution displays. The idea of switching between direct and indirect pointing in this work inspired our interactive design. Liu et al. [9] investigated how screen size and data density would impact manipulation performance in their data classification task on a very large display. On the other hand, we strove to boost the performance by enhancing existing methods while maintaining the intuitiveness.

The use of eyes as an input source has long been discussed because of its intuitiveness and instantaneity. Many studies like ceCursor [7] worked on lever-aging gaze behaviors in GUI cursor manipulation. On the other hand, implementation of gaze-based pointing suffers from accuracy issues. Furthermore, it is difficult to use a perceptual device, such as the human eye, for direct manipulation while ensuring user comfort. Report from MacKenzie [8] comprehensively discussed problems in using eyes as input devices. Zhai et al. [1] provided an early concept of using eye gaze as an auxiliary means of controlling a GUI cursor and instantiated it with eye-tracking technology. The proposal technique to be introduced in this paper leveraged the user's gaze behavior in a less precision-demanding manner.

3 The Design of Facial Tracking-Assisted Hand Pointing

A typical GUI pointing task performed by a user can be divided into two steps: 'Focus' and 'Action'. In the Focus step, users quickly glance over the screen space and find the target to be manipulated; then, they perform the manipulation in the Action step. This division can be applied to conditions with displays of various sizes, yet has unique importance on a WSD compared with smaller desktop/mobile screens. Specifically, while the actual cursor movement happens in the Action step for both conditions, it takes considerably longer to finish the Focus step on larger displays due to the increased screen size, and also the Action step due to the greater cursor movement distance, all of which have an impact on the overall performance.

Our pointing technique consists of two operations, corresponding to the two steps mentioned above, as shown in Fig. 1. The user (1) orients his/her face onto the target, and then (2) moves his/her hand, which is associated with the cursor's movement, to adjust the cursor's position. Part of the cursor's movement is associated with the face orientation; by doing this, some of the cursor movement occurs during the Focus step, parallelizing the two steps to shorten the overall task time. In the following, we explain the details of our pointing technique.

3.1 The Role of the User's Face Orientation

To find the manipulation target (GUI element) on a WSD, the user needs to confirm the target's position visually, often by rotating his/her head due to the huge screen size. Therefore, the position of the user's intended target can be roughly, yet easily, estimated by tracking his/her face orientation. The estimated position is then used to shorten the distance between the cursor and the intended target.

In our proposed technique, we provide a manipulation window - a visible rectangular 'Frame' on the screen that follows the user's face orientation (Fig. 1(b)). This Frame represents the estimated manipulation area around the target position, in which the cursor can be moved in a manner similar to a conventional mouse pointer. The movements of the cursor and Frame are independent to each other when no contact occurs.

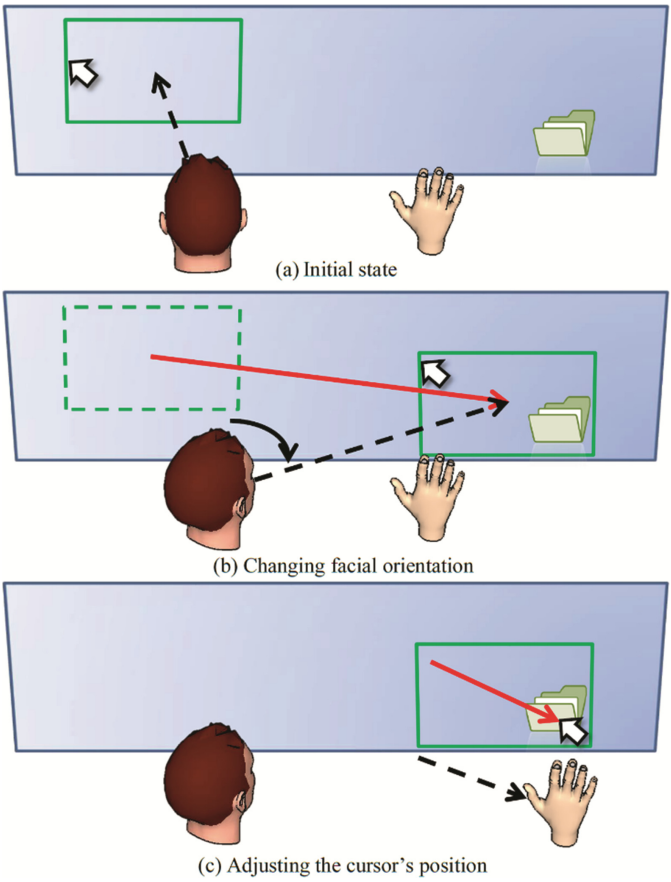


Fig. 1. Two steps involved in the proposed technique

The cursor is forced to be always inside the Frame; that is, the user can move the cursor freely inside the Frame, but the cursor will be "pushed" back into the Frame when reaches any of the Frame's edges (Fig. 2). Consequently, the cursor is always kept in the estimated manipulation area, which helps to shorten the distance between the cursor and target, substantially reducing the cursor movement time.

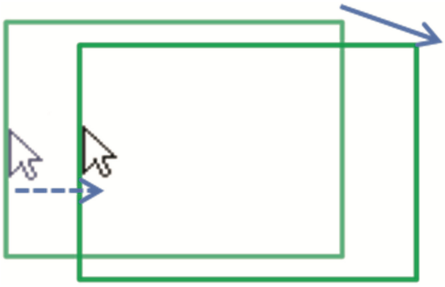


Fig. 2. Interaction between the 'Frame' and cursor

The size of the Frame is a key parameter that affects performance. During pilot experiments, we noticed that an oversized frame has a reduced ability to optimize the cursor position, while an undersized frame can unduly limit the cursor's movement, leading to manipulation errors. We hypothesized that the proper size varies with different hardware parameters (*e.g.,* screen size) and different interaction contexts. Rather than determining the proper size for each setting, we developed a dynamic, self-adaptive solution in which the size of the Frame is associated with the speed of the user's facial movement. Specifically, the Frame "shrinks" when the face orientation changes quickly, "expands" when the orientation tends to stabilize, and recovers its original size when the user focuses on the target (Fig. 3). The change in the size of the Frame during runtime is given by the following equation:

$$R_{rect} = max(R_{max} - \Delta s^k,\ R_{min}) \tag{1}$$

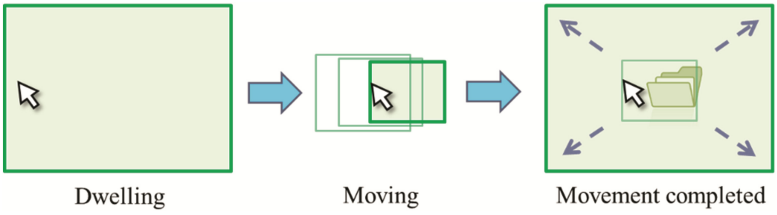


Fig. 3. The dynamic frame

In this equation, R_{rect} indicates the size of the Frame, R_{max} and R_{min} are the maximum and minimum size of the Frame, respectively, Δs is the moving speed of the Frame center, and the index k represents the steepness of the size change.

The purpose of the dynamic Frame size is to reduce the target-to-cursor distance in the Focus step, while providing a much larger range for the cursor movement in the Action step, to enhance the overall performance. This design also reduces the need for calibration when using the proposed method in different hardware settings.

3.2 Cursor Controlled by Hand Motions

While the cursor's moving range is constrained by the Frame as aforementioned, the user can control the cursor freely inside the Frame by using his/her hand. The basic cursor operations (i.e., moving and selecting) are associated with the user's hand motions in the following manner: (1) the cursor moves with the incremental movement of the hand in space; and (2) the user can switch the association between the hand's motions and the cursor on/off. The user's hand will not affect the cursor until it is moved forward a specified distance from his/her chest. Besides avoiding unexpected operations, this design also enables the user to move the cursor gradually for a longer distance by moving the hand in one direction repeatedly (Fig. 4), which provides more freedom handling the cursor.

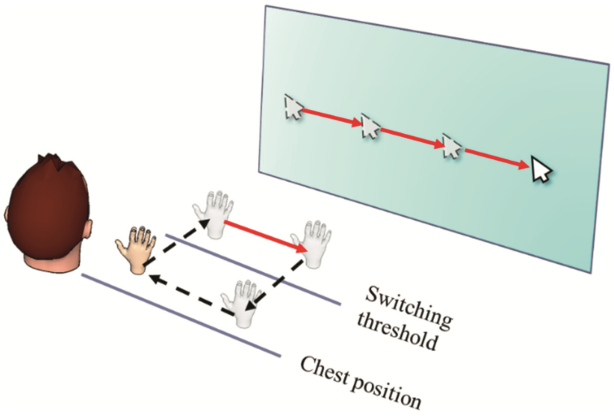
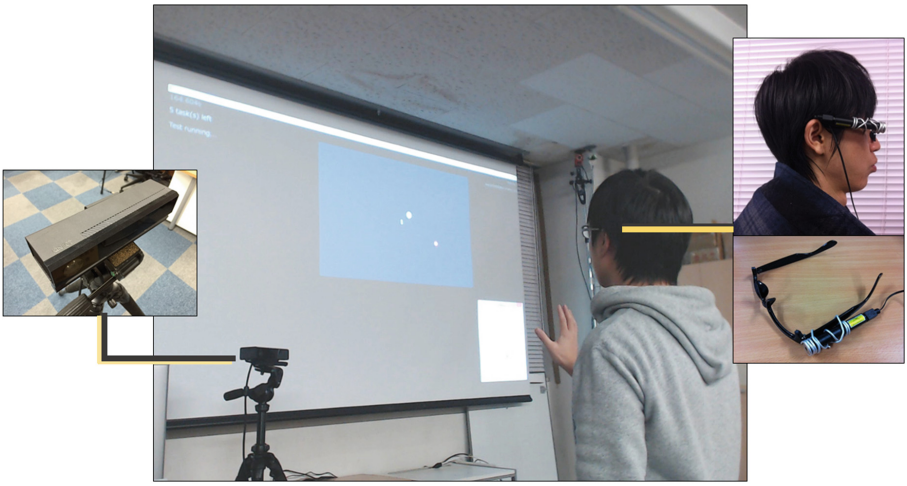


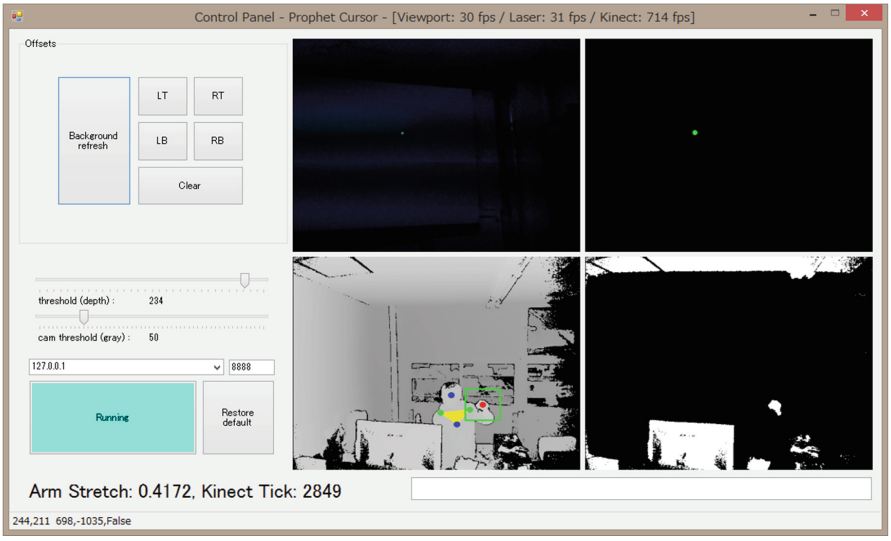
Fig. 4. Behavior of the GUI cursor

4 Implementation

We developed a prototype to evaluate the efficiency of our proposal. Our prototype system (Fig. 5) used a head-mounted laser pointer and an off-the-shelf webcam to track the user's face orientation. The user was asked to wear a pair of glasses with the laser pointer attached to the eyeglass frame. When the user looked at the WSD, the laser pointer projected a bright light dot on the screen. The webcam was placed in front of the WSD with its field of view covering the entire screen. The relative position of the light dot on the screen was extracted by processing the webcam image. The Frame was then centered on the laser dot on the screen, indicating the range within which the cursor can be moved.



(a) Tracking the user's face orientation and hand motions



(b) Processing data from cameras. The raw image from webcam, extracted light dot position, raw Kinect depth image and extracted hand position were monitored.

Fig. 5. Prototype system

We used a Microsoft Kinect which was capable of tracking multiple body parts of the user to control the cursor. The position of the user's two shoulders and center of the chest were tracked, and the plane formed by these three-dimensional positions was used as the reference plane. The distance between the user's hand and reference plane was calculated. When this distance exceeded a specified threshold, the movement of the user's hand parallel to the reference plane was measured to move the cursor.

5 Evaluation

We conducted user studies using our prototype system to evaluate the effectiveness of the proposal technique. Two comparative experiments focusing on the task completion time were conducted to determine whether the face orientation tracking approach improves the overall efficiency of manipulating the cursor.

5.1 Apparatus

Both experiments were conducted in a room with a 100-inch vertical projection screen capable of showing image content at a resolution of 1600 × 1200 pixels. A 1-mW green light laser pointer was used to cast a light dot onto the screen. A Logitech Webcam Pro 9000 system and OpenCV 2 library were used to capture and extract the position of the light dot to track face orientation. A Microsoft Kinect sensor was used to track the user's posture. The arrangement of these instruments is shown in Fig. 6.

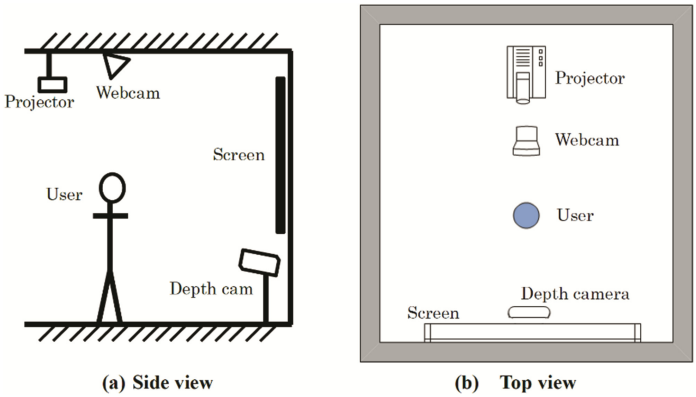


Fig. 6. Instrument layout used in the evaluation

5.2 Task

The tasks in both experiments involved the same basic design: a circular target was shown on the screen at random positions, and the participants were asked to point to each target with a GUI cursor as quickly as possible. Each participant repeated this simple task several times. A "selected" event was triggered when the cursor hovered on the target for 700 ms; the selected target was then replaced by a new target. The distances between two temporally adjacent targets were identical to maintain the same physical effort of moving the cursor for every single task.

5.3 Experiment 1: Fixed Frame

In the first experiment, we measured the performance of the proposed two-step technique using a fix-sized Frame, and compared it with a pointing method using only the hand, without facial tracking. Both of these conditions used the same aforementioned task

design, and each participant was asked to perform 50 consecutive tasks in each condition. In the facial-tracking-assisted condition, we used a 324 × 200 pixels Frame (1/6 of the screen size in height with a 1.618:1 aspect ratio). The task time and users' hand motions were recorded for further analysis. Ten participants from the university, one female and nine male, took part in this experiment. All participants were right-handed. Since each participant was asked to perform tasks in all two conditions, training for both conditions was provided before the experiment to alleviate the possible impact from learning effect.

The average task time performed by all participants was 4.21 s, which was 1.11 s less than its counterpart setting (Fig. 7, with standard error, $p < 0.01$). The maximum and minimum average task time of each participant when using the facial-tracking-assisted method were 6.25 and 3.17 s, respectively, compared with 8.81 and 3.91 s with the hand-only method, as shown in Fig. 8 (with standard error). All 10 participants showed a reduced task time using the facial-tracking-assisted method, indicating an overall performance boost.

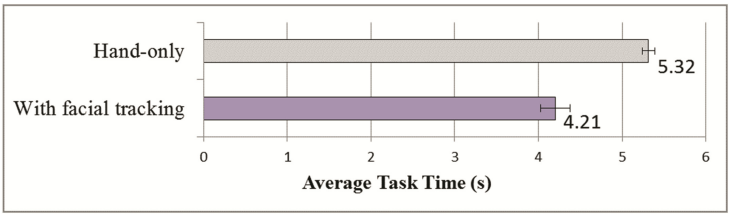


Fig. 7. Overall average task time in Experiment 1

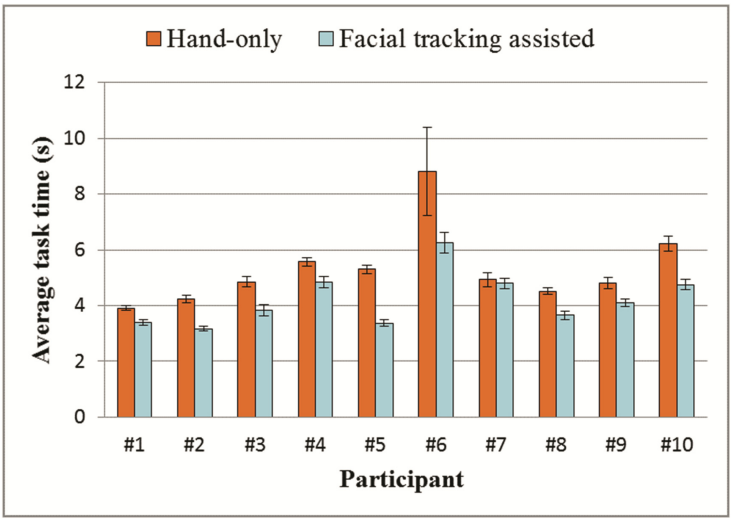


Fig. 8. Average task time of each participant

The proposed technique also showed a trend in that attempts to adjust the cursor's position (using the cursor on/off feature) were less frequent. Figure 9 shows the distribution of the number of attempts to "switch on the cursor" for all tasks performed. The average number of attempts using the proposed method was 1.26 compared with 2.34 using the hand-only condition. The figure also shows that with the help of facial tracking, 82.6 % of the tasks were done with a single attempt at cursor movement, while 74.4 % of the tasks in the hand-only condition required two or three attempts. This indicates that the use of facial tracking significantly reduced the eventual cursor movement distance and, hence, the overall task time.

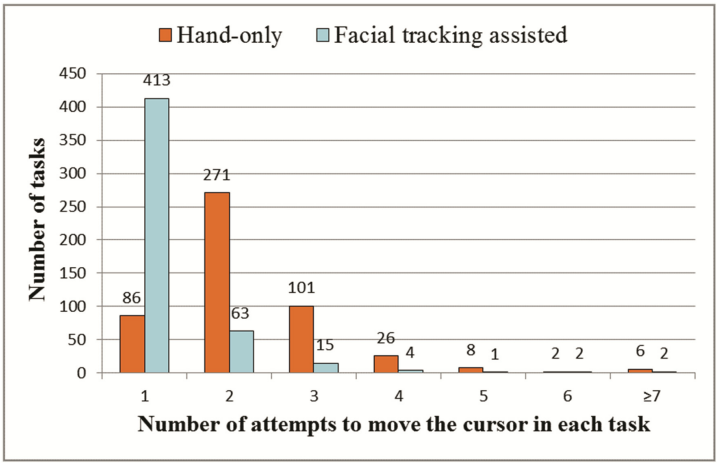


Fig. 9. Tasks cumulatively classified by the numbers of cursor-movement attempts

We observed situations in which the performance was affected by the Frame size setting during the first experiment. The cursor was occasionally hit by the Frame unexpectedly using our initial setting, and was sometime too far away from the target when we tentatively enlarged the frame. Both of the situations limited the performance. These were consistent with our observations in the pilot experiment. We attempted to solve these issues by adopting a dynamic frame approach. Experiment 2 was designed to test the effectiveness of this approach.

5.4 Experiment 2: Dynamic Frame

An experiment was conducted to determine the effectiveness of the dynamic frame. The experiment compared the dynamic frame with a fix-sized frame. The participants were asked to perform 20 consecutive tasks under both conditions. In both situation we used a Frame with an initial size of 971 × 600 pixels (1/2 of the screen size in height with a 1.618:1 aspect ratio) to guarantee the freedom of cursor movement. The minimum size of the dynamic Frame was set to zero. The index k was set to 1.6. The task time and user's hand motions were recorded. Eight university students (one female, seven male; all right-handed) participated in this experiment.

Figure 10 (with standard error bars) indicates that the task took an average of 3.99 s with the dynamic frame, which was 0.87 s faster than with the fix-sized frame ($p < 0.01$). Overall, for the 160 tasks performed in the experiment, the participants made an average of 1.93 attempts to finish the task using the dynamic frame method, compared with 2.16 attempts in the fixed frame condition, reflecting the slightly shorter cursor movement distances in the dynamic frame condition.

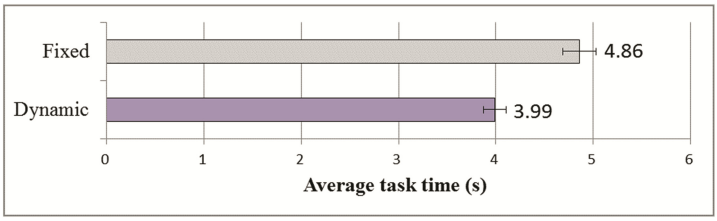


Fig. 10. Average task time comparison for Experiment 2

Using the dynamic frame approach, nearly two attempts were required on average, which was still high. Therefore, we postulated that other factors affected the performance. During Experiment 2, we noticed that the user's head motions slowed as his/her field of view approached the target, resulting in the Frame expanding instantly and limiting the effect of changing the frame size. To remedy this problem, we introduced a "delayed expansion" mechanism for the size-changing behavior. That is, when the Frame's speed slowed down, instead of expanding the Frame instantly, a timer was triggered, and the Frame started to expand only when its speed continued to decline for a certain time interval. We have found significant alleviation to the above issue during later observation, though further quantification experiments are needed to evaluate the effect of this approach.

6 Conclusions

In this paper, we proposed a pointing technique that enables efficient interactions on wall-sized displays that leverages human body motions. By tracking the user's face orientation, the user's focus on the screen can be roughly estimated to predict the position of the intended target. The user can then use hand motions for precise pointing. We conducted two experiments to evaluate the effectiveness of our approach.

The results of the first experiment illustrated the feasibility of boosting the pointing performance on a WSD by roughly tracking the user's field of vision. However, we also observed issues with the frame's behavior that affected the overall performance. The results of the second experiment supported our view that a self-adaptive approach can be leveraged as a remedy for these issues.

While the experiments demonstrated the effectiveness of our proposal in terms of overall performance, the manner by which internal variables (*e.g.,* size/shape of the Frame) affect the performance remains unclear. In the future, we plan to determine the role of each variable and develop a method to optimize the variable settings.

References

1. Zhai, S., Morimoto, C., Ihde, S.: Manual and gaze input cascaded (MAGIC) pointing. In: CHI 1999: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 246–253 (1999)
2. Nakanishi, Y., Fujii, T., Kiatjima, K., Sato, Y., Koike, H.: Vision-based face tracking system for large displays. In: Borriello, G., Holmquist, L.E. (eds.) UbiComp 2002. LNCS, vol. 2498, pp. 152–159. Springer, Heidelberg (2002)
3. Nakanishi, Y., Sato, Y., Koike, H.: EnhancedDesk and EnhancedWall: augmented desk and wall interfaces with real-time tracking of user's motion. In: Proceedings of Ubicomp2002 Workshop on Collaborations with Interactive Walls and Tables, pp. 27–30 (2002)
4. Vogel, D., Balakrisham, R.: Distant freehand pointing and clicking on very large, high resolution displays. In: UIST 2005: Proceedings of the 18th Annual ACM Symposium on User Interface Software and Technology, pp. 33–42 (2005)
5. Kobayashi, M., Igarashi, T.: Ninja cursors: using multiple cursors to assist target acquisition on large screens. In: CHI 2008: Proceeding of the Twenty-Sixth Annual SIGCHI Conference on Human Factors in Computing Systems, pp. 949–958 (2008)
6. Blanch, R., Ortega, M.: Rake cursor: improving pointing performance with concurrent input channels. In: CHI 2009: Proceedings of the 27th International Conference on Human Factors in Computing, pp. 1415–1418 (2009)
7. Porta, M., Ravarelli, A., Spagnoli, G.: ceCursor, a contextual eye cursor for general pointing in windows environments. In: ETRA 2010: Proceedings of the 2010 Symposium on Eye-Tracking Research and Applications, pp. 331–338 (2010)
8. MacKenzie, I.S.: An eye on input: research challenges in using the eye for computer input control. In: ETRA 2010: Proceedings of the 2010 Symposium on Eye-Tracking Research and Applications, pp. 11–12 (2010)
9. Liu, C., Chanpuis, O., Beaudouin-Lafon, M., Lecolinet, E., Mackay, W.: Effects of display size and navigation type on a classification task. In: CHI 2014: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 4147–4156 (2014)
10. Nancel, M., Pietriga, E., Chapuis, O., Beaudouin-Lafon, M.: Mid-air pointing on ultra-walls. ACM Trans. Comput.-Hum. Interact. (TOCHI) **22**(5), 1–62 (2015)