# Pointing in the air: Measuring the effect of hand selection strategies on performance and effort

Matthias Schwaller, Denis Lalanne
Department of Informatics, University of Fribourg,
Boulevard de Pérolles 90, 1700 Fribourg, Switzerland
{matthias.schwaller, denis.lalanne}@unifr.ch

**Abstract.** The research presented in this paper aims at measuring the effect of selection strategies on free-hand pointing performance and effort.
Different evaluations are presented which shows that the selection strategies and the feedback do influence the pointing performance. In the main evaluation presented in this paper, three selection strategies, namely dwell, thumb and pinching, are compared. There was no winning selection strategy, although there was a significant effect of the selection strategies on the pointing performance. Further, the paper shows that it is not enough to compare only the performance of the user, but also the effort, comfort and the selection errors must be taken into account.

**Keywords:** Deictic gestures, Gestural User Interfaces, Pointing performance, Fitts' Law.

## 1 Introduction

In recent years, gestural user interfaces have become very popular, mostly because they are fun (in games) and very adapted for teamwork in collaborative settings such as working around interactive walls. Since the launch of gestural consoles like the Wii, PlayStation Move and Microsoft Kinect, gestural interfaces have become accessible to everyone.

Nevertheless, free-hand deictic gestures (pointing and selection) still rely on basic strategies. Microsoft Kinect proposes a selection method based on a temporal threshold, i.e. you must stay on the target for a certain time (about a second) in order to select it. While it works and is reliable with a good visual feedback, we believe it is not an optimal strategy because of the time used for a selection.

This was the major motivation for the work presented in this article: to develop novel hand pointing and selection strategies that do not require holding a device or markers, nor calibration. We implemented three different selection strategies: using depth, temporal thresholding (dwell) and using the thumb. In the article, we focus our presentation on the selection strategies. Further, in order to augment the usability and precision of our deictic gestures, different kinds of visual feedback were implemented. The feedback not only indicates to users whether hands are detected or not, and if not why (too far, out of pointing range, etc.), it also shows how the system actually works.

Another contribution of our paper is the consideration of a general metric that combines comfort, accuracy and perceived quality as complementary factors, in addition to the index of performance.

The remainder of this paper is structured as follows: First, we give an overview of some related work. Next, we present two preliminary evaluations, one about selection strategies and one about selection feedback. Furthermore, the pointing and selection gestures with their recognition algorithms, as well as the feedback are illustrated. Finally, evaluations are presented along with their results, followed by conclusions and future work.

## 2 Related work

Concerning the recognition of pointing gestures, known as deictic gestures, research has been done in the past on which we based our implementation. Haker et al. [1] used a time-of-flight camera to acquire and recognize pointing gestures. Their first step for the gesture recognition was the segmentation of the person from the background. For this purpose, they used the intensity data which is similar to the depth map which we use. For the tracking of the hand they used a Kalman filter to do temporal smoothing to avoid jiggling. We also used a Kalman filter. The main difference is that they used the direction from the head to the hand as the pointing direction, while we consider only the hand for the pointing.

Harrison et al. [2] proposed an efficient algorithm for free-hand pointing gesture recognition using a depth camera. In their method, they first looked for fingers and then detected touching for selection. The finger detection started by computing the depth derivate using a sliding window and then looked for vertical slices of cylinder-like objects. However, the method is very sensitive to acquisition conditions (device/user positions) so that with their method the angle can neither be too steep nor too shallow.

Frati & Prattichizzo [3] proposed another approach in which they first calculated the convex hull of the overall hand and then searched for convexity defects to detect the fingertips. For the pinching selection strategy we also use the convex hull. We use the same polynomial approach to detect the thumb in the thumb selection strategy as Klompmaker et al. use in their dSensingNI framework [4] for detecting fingers.

Concerning selection strategies, various approaches have been proposed in recent research, taking into account or not ergonomic and physiological issues. Vogel et al. [5] interestingly noted that “since the hand is also pointing, the click or clutch action should be designed to minimize hand movement side effects, which can be tricky due to the interconnectedness of tendons and ligaments in the hand”. They implemented a thumbTrigger method on which preliminary tests led to the conclusion that this method is uncomfortable and tiring. Their thumbTrigger method was initially inspired by the trigger gesture of Grossman et al. [6]. As presented later in this article (section 4.3), in our adaptation of the selection strategy (S2), only the index finger is outstretched and the rest of the fingers (except for the thumb) are folded. The thumb can touch the middle finger or simply be partly hidden behind the palm. This requires only a little movement, and less effort than the thumbTrigger, and is thus less tiring. Such a thumb selection strategy is also presented by Moeslund et al. [7] although their system is wearable and has a head-mounted camera which make it intrusive. A thumb selection strategy was also presented by Gallo & Ciapi [8]. In their implementation, where they used Wiimotes and a data glove with IR LEDs, the thumb is bent for the pointing and outstretched for the selection which is the inverse of our implementation of the thumb selection. Banerjee et al. [9] adapted the thumbTrigger selection strategy of Grossman et al. (where the thumb can touch the middle finger) and instead the thumb has to be leaned towards the index finger, like in our implementation. They used this gesture to reach out-of-reach targets on a tabletop.

Wilson [10] and Gustafson et al. [11] both presented a detection of pinch gestures, but in different settings. Wilson used the pinch gesture over the keyboard in front of the screen. Gustafson et al. used the pinch gesture for drawing in the air detected through a wearable webcam. Benko & Wilson have extended the pinch gesture of [10] and used it in a dome [12]. Foehrenback et al. [13] also presented a pinching gesture, but which required the use of sensors on the fingers, which made it intrusive. In those four versions of the pinching gesture the fingers other than the thumb and index are outstretched. A modified version of the pinch gesture presented by Wilson was presented by Fukuchi et al. [14]. Their adaptation was that the pinch gesture can be done either with thumb and index, thumb and any finger or by thumb and all fingers. This system was used with a tabletop system. The advantage of the pinch gesture is that there is a non-ambiguous state, so either the thumb is touching the index or not and thus has an implicit feedback.

Finally, concerning visual feedback, Grossman et al. [15] presented the Bubble Cursor which is an area cursor which changes its size depending on the proximity of the targets. The bubble cursor does always “point” at a target. We do not have an area cursor but a part of our cursor changes its size according to a threshold (distance or time) the user has to reach to trigger a selection. The property, that a bubble cursor does always point to a target is not implemented in our research in order to compare the pointing performance of the user.

## 3 Evaluations 1 & 2: Preliminary tests of the effect of selections and feedback

We started by doing two preliminary evaluations to check whether (1) the selection strategy, and (2) the visual feedback might influence the pointing performance of the user. For this preliminary evaluation we used a time-of-flight (TOF) camera. We have chosen the TOF camera Swiss Ranger 4000, since this camera has a frame rate of about 50 frames per second. We thought to detect a movement it is better to have a higher frame rate than having a big resolution. To evaluate the impact of selection strategies and visual feedback, we first performed a user evaluation with 6 users. For this purpose we developed TargetCatching, an application that allows measuring the performance of pointing and selection gestures. In this application, the user had to click on several round targets of different sizes, located at different positions. The application then calculates, with the help of Fitts' Law, the index of performance (also called throughput) which is measured in bits per second (bps) IP = ID/MT, where ID is the index of difficulty = log2(D/W + 1), D is the distance from the initial point to the target, W the diameter of the target, and MT is the total time taken to select targets. In the final evaluation presented in section 6, we modified this application to be compliant with part 9 of the ISO 9241 standard for non-keyboard input devices, both in terms of target positioning and index of performance calculation.

### 3.1 Varying selection strategies

In this first evaluation we compared 3 different selection strategies: dwell, distance and thumb selection.

For a selection with the dwell selection strategy, the user has to keep the pointer (controlled with his hand) over the region she or he would like to select for approximately one second (similar to the one presented in section 4.3).

The distance strategy, as the name says, uses distance to perform a selection. The selection is performed when the user passes a certain distance (fixed distance towards the sensor). It is quite similar to a button press or a click in the air. In order to prevent a flickering effect the user has to augment the distance from the hand to the camera in order to make the next selection. The advantage of this distance selection is that the user can use any hand posture which helps to reduce fatigue.

The third selection strategy used in this evaluation was the thumb selection strategy. For this selection strategy the user deploys his thumb for pointing and hides it (or leans it to the middle finger) for selecting (similar to the one presented in section 4.3).

As pointing location we use the topmost point cloud of the contour of the detected hand (see **Fig. 1**).

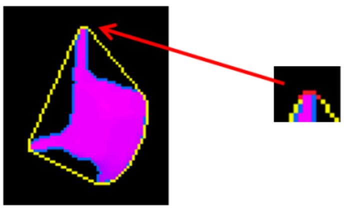

**Fig. 1.** Pointing with topmost location

The outcome of this evaluation is presented in **Table 1** (mean of all users). For each selection strategy, users had to select 20 targets, with varied target sizes and positions. The same set of 20 targets was used 3 times, but only the last 15 were used to compute the index of performance, leaving aside the first 5 selections due to the potential learning effect.

**Table 1.** Pointing efficiency with different selection strategies

| Selection strategy | Errors / selection | | Index of performance | |
|---|---|---|---|---|
| | M | SD | M | SD |
| Dwell | - | - | 1.30 bps | .07 |
| Distance | 0.62 | .34 | 0.75 bps | .18 |
| Thumb | 0.49 | .39 | 1.57 bps | .26 |

M = Mean, SD = standard deviation

There was a significant effect of the selection strategy on the index of performance $F(2, 10) = 310.286$, $p < .001$ (one-way repeated measures ANOVA). Three paired samples t-tests (Bonferroni corrected) were used to make post-hoc comparisons. Significance was found for the two pairs: distance vs. dwell; $t(5) = 8.96$, $p < .001$ and for distance vs. thumb $t(5) = 10.26$, $p < .001$. Between the thumb and dwell ($t(5) = 2.58$, $p = .049$), there was no significant difference when using the Bonferroni correction.

This evaluation measured the effect of selection strategies on the index of performance. With the evaluation, we found that the distance strategy is worse than the dwell strategy. Therefore we put the distance strategy aside and add another one for the final evaluation, the pinching strategy which is often used (see section 4.3). The errors were not measured for the dwell selection strategy since we detected a "pointer over" event (detection if the cursor is over a target) and ran a timer to detect a click. So if the cursor is next to the target, nothing will happen. Therefore we adapted the dwell strategy for the final evaluation to be able to detect false clicks. Further, the selection for the distance strategy was quite difficult since the pointing was also influenced by a movement in depth which made it difficult to "stay" on a target for a selection.

### 3.2 Visual feedback

The second preliminary evaluation aimed at comparing the pointing/selection performance using different visual feedback strategies. This permitted us to select a good visual feedback for the user evaluation presented in section 4.4.

In the experiment, the thumb selection strategy was used with three different types of visual feedback: using only the overview (**Fig. 2** on the left), using the overview together with a varying size circle cursor illustrating the distance between the thumb and index finger (**Fig. 2** on the right), and using the overview with an image of the detected and zoomed hand following the cursor (**Fig. 2** in the middle).

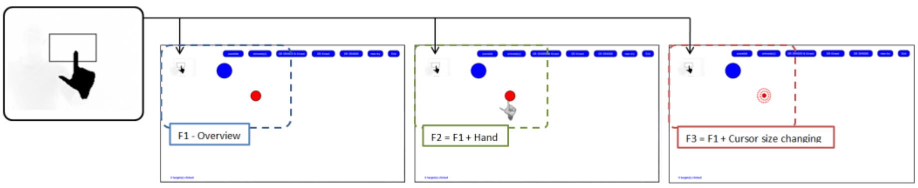


**Fig. 2.** Feedback: overview, hand and cursor

We found that it makes no sense to use the application without feedback at all. Without the overview it is hard to start because the user will not see where to place their hand so that it is in the camera range and if the hand is too far or too close. Further, if there would be no overview and the user goes sometimes out of the camera range and thus the system would not correctly recognize the movements; the system would probably not be well accepted by the user. The setup for this evaluation was the same as for the first preliminary evaluation. For this evaluation, 6 users (different from the first preliminary experiment) had to select 20 targets (5 training and 15 to measure the index of performance). This preliminary evaluation was done using the thumb selection strategy. The outcome of this evaluation is presented in **Table 2** (mean of all users).

**Table 2.** Pointing efficiency with different types of feedback

| Feedback | Errors / selection | | Index of performance | |
|---|---|---|---|---|
| | M | SD | M | SD |
| Only Overview | 0.21 | .20 | 1.41 bps | .16 |
| Overview with cursor | 0.17 | .14 | 1.51 bps | .16 |
| Overview with hand | 0.20 | .17 | 1.33 bps | .15 |

M = Mean, SD = standard deviation

There was a significant effect of the visual feedback on the index of performance $F(2, 10) = 4.85$, $p = .034$ (one-way repeated measures ANOVA). Post-hoc analysis (Bonferroni corrected) showed a statistically significant difference between the hand feedback and the cursor feedback: $t(5) = 4.84$, $p = .0047$ (between the other two conditions there was no statistically significant difference. For this reason we decided in the subsequent evaluation to use the cursor feedback, which we improved with an arrow indicating the center of the selection more precisely.

This user evaluation permitted us to evaluate that the feedback influences the pointing performance. Normally, one would think that a selection feedback in addition to the overview feedback would improve the performance, but in this evaluation we found out that this is not always the case. If a feedback attracts the user's attention too much, as it is the case with the hand feedback (significant effect on the pointing performance), the performance seems to decrease. For the final evaluation we decided to take the best feedback of this preliminary evaluation and adapted it a bit (see section 4.4).

# 4 Evaluation 3: Effect of selection strategies on user performance

## 4.1 Setup and recognition

In our setting, the user sits on a chair with armrests, in front of a big screen (52 inches). The gestures are recognized through a Kinect for Windows sensor from Microsoft® which is placed in front of the user below the screen (see **Fig. 3**). The user can put the elbow on the armrest; the forearm has to point towards the top of the screen. The wrist has to be moved slightly up in order that the palm points towards the Kinect. We used in this experiment a Kinect instead of the time-of-flight camera in the preliminary evaluations, because we remarked that 30 frames per second are enough and with the Kinect we have the higher resolution (640x480 pixels versus 176x144 pixels). The distance between the users hand and the screen is about 1.8 m. The Kinect stands on a box of 0.78 m and at a distance of about 1.05 m.

**Fig. 3.** Setup

The recognition of deictic gestures (pointing and selection) is done through several steps in our implementation: first pre-processing, then pointing and then selection. Next, there is a small tracking algorithm which considers the last 4 states. As state we consider the pointing location and if a selection occurred or not. Finally a Kalman filter is used to make the cursor movement smoother.

Concerning the pre-processing, we filter out the image using the distance. We assume the user's hand to be the closest object in front of the Kinect and take a predefined distance (10 cm) after the closest pixel to the sensor. The current restriction is that nothing can be closer to the sensor than the pointing hand. We then apply a blurring over the image. Next we use erosion and dilatation to prepare the image for the contour detection, which uses canny edge detection.

### 4.2 Pointing

In our current implementation we use the right hand for pointing, due to our selection methods. In the beginning, we used the highest point of the detected hand (contour) as the pointing position.

This obviously only works for selection strategies where the highest point of the hand does not move during the selection action. For the selection strategy S3 - pinching click - the pointing with the topmost position provokes a pointer moving when just a click is done. Thus, the user would have to move his or her hand up while clicking, which is not comfortable at all. Therefore, we had to find a pointing implementation which works for all the selection strategies in the same way, to be able to compare them equally. Therefore, we now use the center of mass of the detected hand. This works with any hand postures, which reduces fatigue.

To be able to detect the full hand in any position, we reduced the pointing window in which the user points. As a second advantage, this also reduces the movement which the user has to do to go from one side of the screen to the other side. We then directly map the pointing position in this window to the cursor on the screen. We will see later in the article how the visual feedback helps users position their hand in this pointing window.

Furthermore, since the resolution of the sensor is much smaller than the resolution of the screen in our setting, the cursor would have a flickering effect. Therefore, we used a Kalman filter which allow a fluid pointer movement and predict the location in case of a missing frame.

### 4.3 Selection

Pointing recognition and tracking permits to move a pointer on the screen. Still, a selection must be performed to “click” on a target. An important aspect for a reliable selection is that the movement which is necessary for selecting should not influence the pointing. For the selection, we implemented several strategies, for this research they were only implemented for the right hand with the palm towards the sensor (but of course, the thumb and pinch could be mirrored also be implemented for the left hand):

**S1 – Dwell strategy**

The user has to keep the pointer (controlled with his hand) over the region she or he would like to select for one second, which is approximately the time which is used in most Kinect games for Xbox at the time of writing.

The time for a click is a trade-off between speed and accuracy (minimal false clicks). We are currently using 5/6 of a second, which empirically gave the best results. The region in which the cursor has to stay for a click is 60 pixels, which represents the radius of the biggest targets used for the evaluation. Our hypothesis is that since this strategy does not require any additional movement for the selection, it requires less effort and is less tiring than other selection strategies but requires more time for a selection. For instance, we suppose that such a selection strategy is not well adapted for a menu with submenus. Furthermore, no drag-and-drop can be done with such a selection strategy.

**S2 - Thumb click**

The user deploys his thumb for pointing and hides it for a selection. In order to hide it, it is enough to lean the thumb to the middle finger. The posture of the thumb selection strategy can be seen in **Fig. 4 a**.

The hand segmentation for the thumb click starts by detecting a polynomial approximation over the contour (green lines in **Fig. 4 c-d**). The topology of the edges of this polygon permits to detect if the thumb is present. Therefore, we need to take the left most point and search for the two following points in clockwise order on the polynomial approximation. If the green triangle (left most point), blue cross and the yellow rectangle are in a clockwise order, it means that the thumb is present and if they are in a counterclockwise order, the thumb is hidden or leaned towards the middle finger.

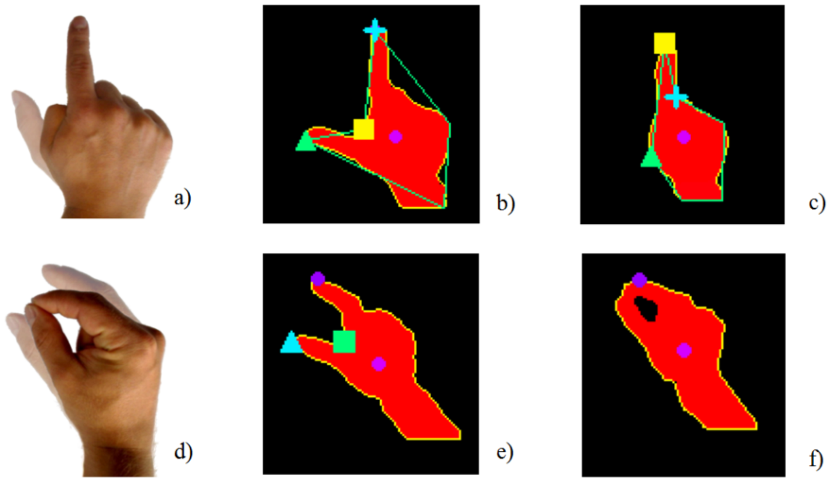


**Fig. 4.** a) thumb gesture, b) thumb detection with important points, c) Thumb detection with thumb leaned towards the middle finger, d) Pinching gesture, e) Pinch open detection with important points and f) Pinch closed detection with important points

**S3 - Pinching click**

The user performs a gesture like a zoom out on a smartphone. This gesture, where the thumb touches the index finger (**Fig. 4** d), is very similar to taking something small from a table. We decided to define that the three other fingers have to be retracted because we found in our tests that this is slightly less tiring than to have them outstretched as in [10–14].

The detection of a pinching gesture starts by detecting convexity defects in the contour of the hand. If the depth of such a defect is above a certain threshold (image width / 45) it means that the thumb does not touch the index. Additionally, the depth point (the farthest point from the convex hull point within the defect, see green rectangle in **Fig. 4 e-f**) has to be over a certain height (midpoint plus 1/5 of the height of the detected contour).

**Tracking**

The images of the Kinect sensor may contain noise. Therefore, it is possible that in a frame the contour cannot be detected or that the pointing locations are wrongly detected or selections falsely detected. For this reason, we implemented a simple history queue which tracks the four last locations and four last selections. With the help of the history information, missing locations can be predicted. In order to be able to configure and also change the selection strategy and feedback, a configuration frame was implemented.

### 4.4 Feedback

The visual feedback shows the user how the system works, meaning that it highlighted the detected hand and the feedback shows if the hand is within the camera range or not. This is important at the beginning until the user has some experience with the system. This facilitates the acceptance of the system by the user. The feedback is composed of three steps. First, there is the overview feedback. This feedback permits the user to see the scene overview in the top left corner. In the overview the user can see the depth image of the detected hand highlighted in black. Furthermore, there is a surrounding frame which shows the pointing zone to the user. The user can only point within this zone, which helps to reduce the size of the movement the user has to do, to go from one side of the screen to the other side. The goal of this overview is to help the user to understand what is detected and what not. This feedback is especially helpful at the beginning to place the hand at the right place or when the user puts the hand outside of the pointing range. This feedback is shown in **Fig. 5**.

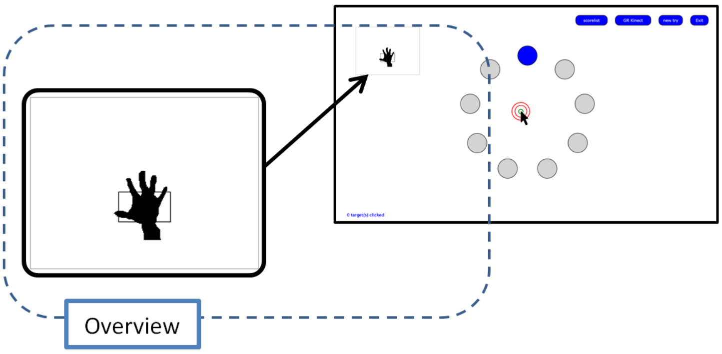


**Fig. 5.** Feedback

The second feedback provides users with additional information for the selection. The circle around the cursor (see **Fig. 6**) can have two different colors. If the circle is green, it means that a click occurred and in this case the circle is small. When the cursor is red, its size changes automatically depending on the distance to a click. For the dwell selection strategy the distance represents the time that the user has to wait for the click. At the beginning it is big and if the user stays on the target, its size reduces for 1 second. For the thumb selection strategy it represents the distance from the thumb tip to the index (see green triangle and blue cross in **Fig. 4** b). As the user approaches the thumb to the index, the circle gets smaller and smaller. For the pinching selection strategy the size of the cursor represents the distance between the tips of the thumb and the index finger (see the two left most points in **Fig. 4 e**). Additionally, a sound is played when a selection is made.

**Fig. 6.** Cursor changing size for the selection feedback

### 4.5 Evaluation and Results

We have changed our application (TargetCatching see **Fig. 5**) to measure the performance so that the evaluations are based on part 9 of the ISO 9241standard for the non-keyboard input devices multi-directional tapping test. So, now the user has to click on circular targets of the same size arranged in a circle on the screen. The size of the circular targets, as well as the circle on the screen, varies. The advantage of this arrangement in a circle is that movements in all directions are equally tested. If the user clicks next to a target, we consider this as an error and the target is not validated and thus the user has to click again on the current target.

**Participants.** Twelve volunteers, 2 female and 10 male, participated in this evaluation. The range in age of the participants was from 21 to 35 years. All were daily computer users.

**Apparatus.** For this evaluation we used a 52 inch TV with a 1920 x 1080 resolution as the screen. The user sat in a chair with arm rests about 2 m from of the screen. The Microsoft Kinect sensor was placed below the TV.

**Task.** The application now calculates the effective index of performance (or throughput), still based on Fitts' Law: IP = IDe/MT, where IDe = effective index of difficulty = log2((D + We) / We), D = radius of the circle, We = 4.133 Sx, Sx = standard deviation of the selection coordinates, MT = total time taken to select targets. This way, the evaluation can be compared to other evaluations based on part 9 of the ISO 9241 standard for non-keyboard input devices.

**Procedure.** To compare the three different selection strategies (S1: dwell, S2: thumb and S3: pinching) presented above, we conducted a within-subject design user evaluation with 12 users, in which all the users tested all the selection strategies. The selection strategies were counter-balanced to reduce carry over effects (mainly fatigue and learning). For each selection strategy there were 4 circles with 9 circular targets which the user had to click. The target circle used for practice had an ID of 2.807 (using D=600, W=100), the other three target circles had an ID of 3.285 (using D=700, W=80), 3.322 (using D=900, W=100), and 2.585 (using D=600, W=120). The topmost circular target had to be clicked at the beginning and at the end. The first circle was used only as training due to the potential learning effect. As radii of the circles we used 300, 350, 450, and only 300 pixels for training. The target sizes (diameters) were 80, 100, 120, and only 100 pixels for training. The effective index of performance was calculated per circle and we took the average of the second, third and fourth circle. The same set was used 3 times.

**Results.** The results obtained are presented in **Table 3** (mean of all users). There was a significant effect of the selection strategy on the effective index of performance F(2, 22) = 4.98, p = .016 (one-way repeated measures ANOVA). Three paired samples t-tests were used to make post-hoc comparisons. We did not found any statistical significance with the t-tests when using the Bonferroni correction (dwell vs. thumb p = .027, dwell vs. pinching p = .026 and thumb vs. pinching p = .68). The pinching selection strategy gave the best result (IP = 1.38 bps).

**Table 3.** Pointing efficiency with different selection strategies

| Selection strategy | Errors / selection | | Eff. index of performance | | Mean |
|---|---|---|---|---|---|
| | M | SD | M | SD | sel. time |
| Dwell | 0.22 | .14 | 1.19 bps | .14 | 2.67 s |
| Thumb | 0.38 | .18 | 1.36 bps | .21 | 2.37 s |
| Pinching | 0.30 | .15 | 1.38 bps | .21 | 2.29 s |

M = Mean, SD = standard deviation

Concerning the training, we noticed that the index of performance was stabilized after the first of the four circles (with a logarithmic trend line). We remarked during the preliminary evaluations (through the comments of the users), that the users are less stressed if the evaluation starts with training.

Beside the index of performance there was also a significant effect of the selection strategy on the errors per selection F(2, 22) = 4.71, p = .020 (one-way repeated measures ANOVA, values were normally distributed). For the post-hoc comparisons we used three samples t-tests. We did not found any statistical significance with the t-tests when using the Bonferroni correction (dwell vs. thumb p = .017, dwell vs. pinching p = .14 and thumb vs. pinching p = .12).

Furthermore, we used the questionnaire proposed in ISO 9241 part 9 annex C. Each question has a Likert scale from 1 to 7, where 7 is the best and 1 the worst. Also for the force, effort, and fatigue does a higher value mean that it is better (0 means very tiring and 7 means not tiring at all, for the fatigue questions for instance). This is because we followed the ISO. The results (median) of the questionnaire are presented in with a box plot in **Fig. 7**. Note that for our questions the higher the score, the better it is.

To compare the results we used the Friedman test (used for non-parametric one way repeated measures) since the results of the questionnaire were not normally distributed. There was a statistically significant difference of the selection strategy on the force required for actuation X2(2) = 10.65, p = .0049. Three Wilcoxon signed rank tests were conducted to make post-hoc comparisons. A significant difference was found for dwell (M = 5.08, SD = 1.38) vs. thumb (M = 3.75, SD = 1.54); Z = 49.5, p

= .011 and significant effect for dwell vs. pinching (M = 4, SD = 1.48); Z = 39, p = .023. Furthermore, we found significant effect of the selection strategy on finger fatigue X2(2) = 14.8, p = .0006 (Friedman test). Three paired samples Wilcoxon signed rank tests were conducted to make post-hoc comparisons. We also found significance for the two pairs dwell (M = 6.67, SD = 0.65) vs. thumb (M = 4, SD = 1.91); Z = 55, p = .002 and dwell vs. pinching (M = 4.5, SD = 1.78); Z = 36, p = .006.

Our hypothesis, that the dwell strategy is less tiring but needs more time for a selection was accepted since there is a significant effect on the pointing performance as well as on the finger fatigue.

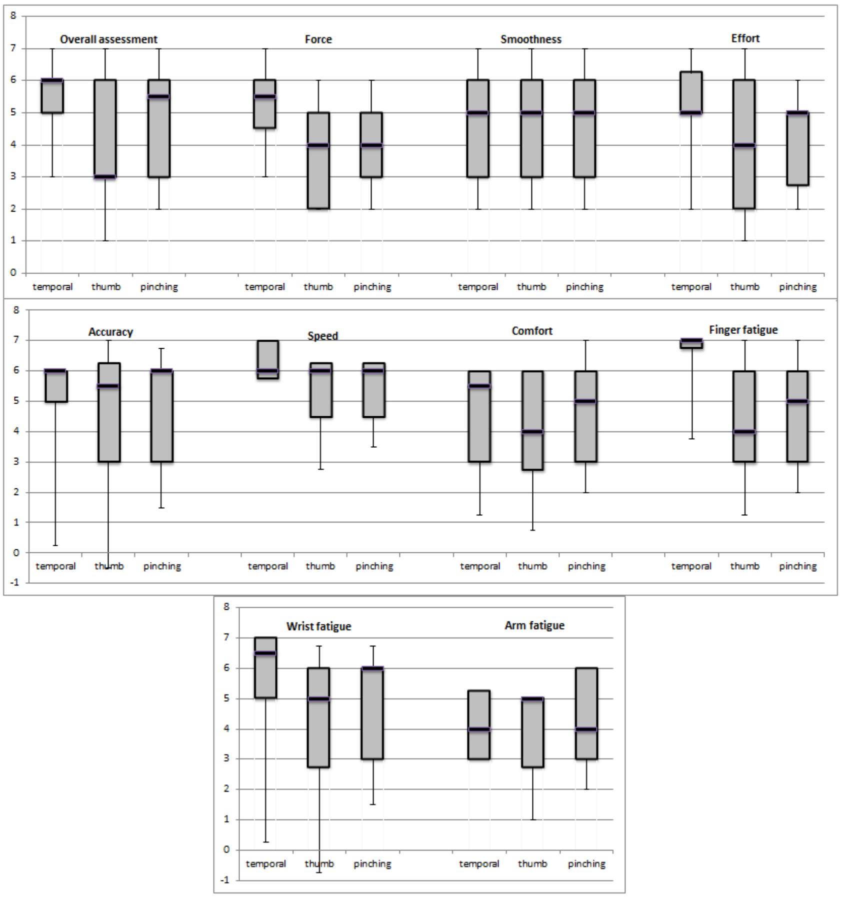


**Fig. 7.** Box plot summarizing answers to the questions

## 5 Discussion

To compare the three selection strategies presented in the third evaluation in this paper an overall performance graph was prepared as shown in **Fig. 8**. Each parameter (perceived quality, overall comfort, accuracy and performance) has a value between 0 and 1. The overall comfort is calculated by averaging the force, effort, finger, wrist and arm fatigue from the questionnaire. The higher the overall score the better the selection strategy. The performance (effective index of performance) of pinching is much better than the one of the dwell selection strategy, but the perceived quality by the user and the accuracy of pinching are worse than those of the dwell strategy. Overall, the pinching strategy is only slightly better than the dwell one. For the future it is clear that we have to work on the accuracy and try to get a better performance and fewer errors for the selection strategies for the users.

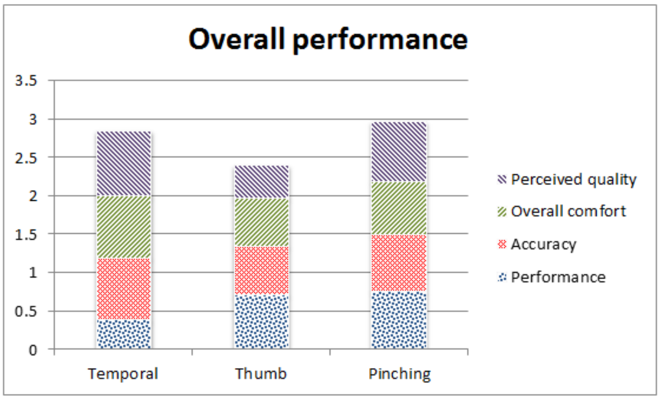


**Fig. 8.** Overall performance

## 6 Conclusion

This article presented various recognizers for hand pointing and selection which do not require any calibration or markers. It further presented the design and implementation of three selection strategies and of various visual feedback to support pointing and selection.

Secondly, the paper presented a user evaluation that showed that selection strategies influence the effective index of performance for selecting targets on a screen. Further, it describes two short user evaluations which helped to determine the best selection strategies and visual feedback mechanism for the later user evaluation. No winning selection strategy was found, and we argued that such selection strategy cannot be measure only with the index of performance. The errors, the perceived quality and the fatigue have also to be taken into account, in addition to the index of performance. By comparing the overall performance (see **Fig. 8**) we remark that the dwell selection strategy has almost the same performance as the pinching selection strategy. At the performance level, the selection strategy which gave the best index of performance, i.e. pinching, is far from being effective since the effective index of performance is not comparable with the performance of a Wiimote for instance (2.7 bps) [16], there are still numerous false clicks, and the perceived quality for the user is not good enough. More work is thus necessary to improve selection strategies.

In the experiments described in this article, gestures have to be performed towards the camera. For instance, when performing a pinching gesture, the hand has to be turned slightly to the left so that the system is able to recognize it. In order to improve the comfort of gestures we plan to use a second depth sensor on the side of the user to avoid occlusions and also to improve performance. We also plan to continue developing our gesture recognizer in order to enrich the gestural vocabulary with more than only pointing and selection. Our plan is to use the dominant hand for pointing and the non-dominant hand for gestural commands such as selection, zoom, rotation, etc.

Finally, using longitudinal studies, we plan to measure the ergonomics of the chosen gestures, meaning not only whether they support a high effective index of performance for pointing and selection, but also how tiring and usable they are when used in a long period of time.

## Acknowledgment

Grateful acknowledgement for proofreading and correcting the English goes to Agnes Lisowska Masson.

## References

1. Haker, M., Böhme, M., Martinetz, T., Barth, E.: Deictic Gestures with a Time-of-Flight Camera. In: Kopp, S. and Wachsmuth, I. (eds.) Gesture in Embodied Communication and HumanComputer Interaction International Gesture Workshop GW 2009. pp. 110–121. Springer (2009).
2. Harrison, C., Benko, H., Wilson, A.D.: OmniTouch. Proceedings of the 24th annual ACM symposium on User interface software and technology - UIST '11. pp. 441–450. ACM Press, New York, New York, USA (2011).
3. Frati, V., Prattichizzo, D.: Using Kinect for hand tracking and rendering in wearable haptics. 2011 IEEE World Haptics Conference. pp. 317–321. IEEE (2011).
4. Klompmaker, F., Nebe, K., Fast, A.: dSensingNI. Proceedings of the Sixth International Conference on Tangible, Embedded and Embodied Interaction - TEI '12. p. 217. ACM Press, New York, New York, USA (2012).
5. Vogel, D., Balakrishnan, R.: Distant freehand pointing and clicking on very large, high resolution displays. Proceedings of the 18th annual ACM symposium on User interface software and technology - UIST '05. pp. 33–42. ACM Press, New York, New York, USA (2005).
6. Grossman, T., Wigdor, D., Balakrishnan, R.: Multi-finger gestural interaction with 3d volumetric displays. Proceedings of the 17th annual ACM symposium on User interface software and technology - UIST '04. pp. 61–70. ACM Press, New York, New York, USA (2004).
7. Moeslund, T., Nørgaard, L.: Recognition of deictic gestures for wearable computing. Gesture in Human-Computer Interaction and Simulation. 112–123 (2006).
8. Gallo, L., Ciampi, M.: Wii Remote-enhanced Hand-Computer interaction for 3D medical image analysis. 2009 International Conference on the Current Trends in Information Technology (CTIT). 1–6 (2009).

9. Banerjee, A., Burstyn, J., Girouard, A., Vertegaal, R.: Pointable: An In-Air Pointing Technique to Manipulate Out-of-Reach Targets on Tabletops. Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces - ITS '11. p. 11. ACM Press, New York, New York, USA (2011).
10. Wilson, A.D.: Robust computer vision-based detection of pinching for one and two-handed gesture input. Proceedings of the 19th annual ACM symposium on User interface software and technology - UIST '06. 255 (2006).
11. Gustafson, S., Bierwirth, D., Baudisch, P.: Imaginary Interfaces: Spatial Interaction with Empty Hands and without Visual Feedback. Proceedings of the 23nd annual ACM symposium on User interface software and technology - UIST '10. p. 3. ACM Press, New York, New York, USA (2010).
12. Benko, H., Wilson, A.D.: Multi-point interactions with immersive omnidirectional visualizations in a dome. ACM International Conference on Interactive Tabletops and Surfaces - ITS '10. 19 (2010).
13. Foehrenbach, S., König, W. a., Gerken, J., Reiterer, H.: Tactile feedback enhanced hand gesture interaction at large, high-resolution displays. Journal of Visual Languages & Computing. 20, 341–351 (2009).
14. Fukuchi, K., Sato, T., Mamiya, H., Koike, H.: PAC-PAC: Pinching Gesture Recognition for Tabletop Entertainment System. Proceedings of the International Conference on Advanced Visual Interfaces - AVI '10. p. 267. ACM Press, New York, New York, USA (2010).
15. Grossman, T., Balakrishnan, R.: The bubble cursor: enhancing target acquisition by dynamic resizing of the cursor's activation area. Proceedings of the SIGCHI conference on Human factors in computing systems. 281–290 (2005).
16. McArthur, V., Castellucci, S.J., MacKenzie, I.S.: An empirical comparison of "wiimote" gun attachments for pointing tasks. Proceedings of the 1st ACM SIGCHI symposium on Engineering interactive computing systems - EICS '09. p. 203. ACM Press, New York, New York, USA (2009).