Gesture termination algorithm for continuous input techniques

Tatiana V. Evreinova, Grigori Evreinov, and Roope Raisamo
Dept. of Computer Sciences, University of Tampere Kanslerinrinne 1, FIN-33014 Tampere, Finland
Phone: +358 3 3551 8549, Fax: +358 3 3551 6070, e-mail: {etv, grse, rr}@cs.uta.fi

Abstract — Many designs have been proposed to improve head tracking interfaces. However, in most head tracking applications, only a click to select is commonly available interaction style. In this paper we present an algorithm which integrates the parameters of the continuous input stream (movement acceleration and the amount of movement) and information concerning the user choice. The results of the test demonstrated that people could control head movement acceleration and the amount of movement without significant effort. The method supports a multiple choice of the commands like a marking menu, as well as several ways to cancel a selection. The method could also improve usability of video-based input techniques for people who still have a sufficient degree of head control.

I. INTRODUCTION

People who cannot use regular input devices are searching for alternatives. In windows-based graphical user interfaces both discrete and continuous gestures can be used to trigger commands (command strokes, shortcut gestures) as well as for menu selection through a marking menu [11]. Pointing and selection can be performed through gestures made with the hand, head, finger, face, nose, mouth, tongue, respiration, lips, eyes, brows or the whole body movement [4], [12], [13], [20]. Head gestures can support computer access for disabled people who still have a sufficient degree of head control [7], [8], [17].

What is the difference between a conventional hand-operated mouse and a head tracker? The user can interrupt the mouse input at any moment and the cursor will wait for the next action at the same screen location. The clicking of a mouse button indicates the termination of movement and initiates a selection of the objects or execution of commands in a graphical user interface.

When a touch sensitive input device is used a stylus/finger landing and lifting can determine and delimit one gesture from another. There is no counterpart of such an action termination for continuous input techniques such as head tracking. In a camera mouse head movements are continuously translated into the cursor position. In other words, the start and end points of the intentional movements have to be detected employing the analysis of the input stream.

In general, a confirmation action to initiate the command can physically be separated from pointing and selection. The specific local human behavioral activity can be detected and used to confirm a selection and terminate a gesture. For instance, electrical muscle activity, teeth chattering, whistle, sipping and puffing, eye blinks and frowning have already been used to simulate a button click. Still, a click to select interaction style limits disabled users in performance and ability to control windows-based applications by using full mouse features in combination with pointing (right-click), as it was designed for able-bodied people.

Many designs have been proposed to improve gesture-based human-computer interaction as well as to adapt such techniques for people with special needs [7], [18]. For instance, in the last version of EyeTwig [7] the authors used dwell time to simulate a left mouse click, a head gesture (nod) to make a right click and speech recognition of predefined commands such as drag, drop, scroll, keyboard and internet. Kjeldsen and Hartman explored application of head gestures for pointing, continuous control of sound pitch and volume, spatial selection (simulation of mouse buttons through tilts of the head) and symbolic selection (by yaw/no and pitch/yes) [11].

Some authors have noted that gesture recognition which is based solely on processing video images can be less accurate and error prone. Morency et al. have recently reported on improvements in recognition performance of the head gestures based on contextual features taken from the windows state manager [16]. They found a subset of lexical, punctuation and timing features from the user interaction context, which could be anticipated at the moment of decision-making.

Besides the reliability of making selection, a possibility to produce a multiple choice to execute different commands is a great challenge in continuous input techniques. McGlaun et al. described a combined video-based approach to track the head and classify gestures [14]. The authors have applied color- and shape-based segmentation of the image, a template matching of the nose bridge, a stochastic algorithm to process the optical flow, and discrete Hidden Markov Models to classify six types of gestures, achieving a recognition rate of about 93-97%. Thus, multimodal integration is computationally
very expensive, some components (techniques) require customization (e.g., stereo-based alignment, classifier training), using special hardware (a high resolution stereo system) and, therefore, such a solution is often impractical.

Keates and Robinson have evaluated the multimodal input system when two gestural input channels, head and hand, were combined to increase functionality of the gesture-based interaction techniques [10]. The authors demonstrated that in such a case the physical and cognitive loads could quickly become extensive and hamper human-computer interaction.

There is a significant challenge in delimiting algorithms and methods to split continuous movements into particular gestures for real-time applications. Herewith, we have to especially emphasize that the cancel request is considered as inherent feature in user interface design. Any interaction method must provide the opportunity of deselection to break/cancel the user input for both completed and uncompleted gestures. This feature has a significant impact on user performance and satisfaction.

An interruption of the continuous movement can be done with a dwell time [8]. Dwell time termination is considered as the ending of action to make a selection in the dwell position. While small movements within a local area during the dwell time interval are possible, dwell time should have a sufficient duration to provide a possibility to cancel selection. On the contrary, to avoid an accidental selection even a dwell time of 400 ms is quite a short time for expert users to cancel a selection (to leave the selected area) with eye and head tracking systems [21]. For an average (able-bodied) user dwell time exceeds 600 ms.

User performance can also be improved by employing an adaptive dwell time duration. However, the main restriction is that the particular time interval (dwell time) allows the user to execute a single predefined choice. Different options (e.g. left-click, right-click or double-click) supporting the user input might be assigned to the time intervals in a sequence. Still, during each of the pieces of the waiting time it should be a possibility to cancel a selection. When the duration of the waiting time is greater than 900 ms the dwell time technique will not bring any benefits.

Zhai and colleagues considered different ways for object selection or command activation in crossing-based interfaces. Crossing the edge of the graphic object, e.g., when the cursor is entering and leaving a software button, could be applied to head/eye tracking as an alternative to selection by dwell or clicking. Depending on the spatial arrangement of the targets crossing gestures had been classified into discrete and continuous, collinear and orthogonal. The authors pointed out that multiple goals could be crossed in a cascaded manner and bidirectional movement could add functionality to the crossing. Also crossing techniques are mentioned as requiring less accuracy in comparison to pointing actions and that could be especially appropriate in some cases for motion impaired users [1].

By taking into account the shape and direction of the strokes, Apitz and Guimbretière in a sketching program CrossY [2] introduced five basic tools for crossing-based drawing applications such as a pen, an eraser, a lasso, a highlighter and a search tool. They also offered to combine CrossY tools and FlowMenu as a command selection method including file operations and application termination. To increase the number of alternatives available immediately after pointing, Hinckley et al. demonstrated a combination of circular (lasso) and directional gestures (marking-menu commands) [9]. To split the input stream into sub-streams the authors used different delimiters such as lifting a pen, dwell time, a button press, and an additional small loop (pigtail) sub-movement.

Finally, with the Curve dial technique [19] curvature of the tracks was determined by analysis of the consequent displacements (concatenated sub-movements) based on recording their speed and direction. Changing the parameter can be proportional to a curvature direction (clockwise and counterclockwise), while the curvature value depends on the circling radius. The curvature was dynamically estimated on each three points. However, the method was also originally developed for discrete hand movements, that is, using landing and lifting of a stylus or finger to initiate and terminate the gestures.

Excluding dwell time, most of the earlier methods have been designed for gesture-based interaction with touch sensitive input devices such as touchpads touch tablets and touch screens. Nevertheless, the techniques specifically designed for pen-based interaction are almost impossible to fit to head tracking input. Taking into account all what was mentioned above, we studied a possibility of gesture termination and making a selection of widgets with the video-based tracking technique using the analysis of the cursor movements. The goal of this empirical research was to find three values for the input system - two thresholds (one upper, one lower) for the magnitude of cursor movements within a gesture and the optimum number of gesture components to discriminate and classify the intentional gesture termination in continuous user activity such as head pointing and navigation.

The basic parameter to be evaluated was a variability of the movement displacement vector (its magnitude and direction). Mensvoort and Keuning suggested that the user intention (a conscious purpose) directs a voluntary action and behavioral activity and, therefore, the purpose can be inferred from cursor
movements [15]. Thus, based on the analysis of the amount of cursor movement - direction of movement, track length and velocity, the successive movements could be separated and interpreted. For instance, a deceleration of the cursor below some threshold value might indicate the area of the user interest (\(d_0\)); a predefined number of cursor displacement vectors (\(\{\hat{d}_1, \hat{d}_2, \hat{d}_3, \ldots\}\)) could be considered as a confirmation event of the specific action, while deviation from the previous direction (\(\hat{d}_i \neq \hat{d}_{i-1}\)) could indicate a change of intent. Equally, a full stop or fast movement to another location might be interpreted as a cancel of unconfirmed selection.

In the next section we will present the algorithm, which integrates the parameters of the continuous input stream (movement acceleration and the amount of movement) and information concerning the user choice.

II. THE GESTURE TERMINATION ALGORITHM

Input techniques, which are based on detection of continuous user movements (hand, finger, head, body), are usually following a particular interaction protocol that limits the cursor movements according to subtasks such as navigation, pointing, selection and confirmation of a selection. Navigation gestures have a significant variability of the velocity and direction of the cursor displacement vector (\(\hat{d}\)). However, when a user has a motivation to capture a target or to point out the cursor displacement vector (\(\hat{d}\)). Deceleration of the cursor below some threshold value might indicate the area of the user interest (\(d_0\)). A full stop or fast movement to another location might be interpreted as a cancel of unconfirmed selection.

To recognize the cursor displacement vector \(\hat{d}\) within continuous cursor movements, the computer system has to detect a starting point that might be associated with an area of the user's interest. The starting point in our algorithm was detected through deceleration of the cursor displacement (\(d_0\)). According to Card et al. (1983), the elementary movement cycle measured for able-bodied persons can vary in a range of 30-100 ms, while for persons with disabilities this value can increase up to 300 ms (Keates and Robinson, 1998). Therefore, a sampling rate to record cursor displacements was set at 100 ms. This time interval is appropriate even for fast head movements because the head-neck system acts as a low-pass filter with cut-off frequency higher than 10 Hz in a horizontal plane [3].

The algorithm of analyzing the continuous user input is shown in Fig. 1. As can be seen from Fig. 1, Timer1 sets a sampling rate of 100 ms when the “GestCtrl” application starts. First of all, the algorithm has to detect a cursor displacement which is less than some predefined value Threshold1 (\(d \leq \text{Threshold1}\)).

We cannot anticipate an absence of body movement (full stop) because of tremor-like micro-movements and possible drifting of the tracking system. On the other hand, the full stop is not necessary at all because it can slow down interaction. Being detected, the displacements which are still greater than a system 'noise' will start Timer2 and reset the index \(i (i = 0)\).

Timer2 is necessary to stop motion analysis when nothing is going on. Timer2 sets the time interval during which it is possible to detect gesture termination and make a selection, to execute assigned command, or to process an alternative movement to cancel the input. To confirm the intent, the user has to make a continuous (non-broken) directional gesture. To execute an action associated with the starting point, the system has to detect the predefined number \((k)\) of cursor displacements \(\hat{d}\). Nevertheless, the displacement vector (track length and velocity) should have values which are close to a regular navigation-like cursor movements (\(\text{Threshold1} < d < \text{Threshold2}\)).

To cancel the input initiated by directional gesture, the user can stop before a final displacement (confirmation) was done; the user can also make a fast movement (\(d > \text{Threshold2}\)) or change the direction of the previous cursor displacement (\(\hat{d}_i \neq \hat{d}_{i-1}\)). Thus, each 100 ms motion analysis makes a comparison of the cursor displacements, their value and direction. When a final sub-movement was processed, the user can change direction and continue interaction with the main application.

The algorithm was implemented as a software module and an application program GestCtrl. The general functionality of the application program was preliminarily tested with a regular mouse and camera-based head tracking software (camera mouse). Next, we had to carry out an empirical study in order to estimate a range for parameters of the algorithm (\(\text{Threshold1}, \text{Threshold2}\) and the index \(k\)).
III. AN EMPIRICAL EVALUATION OF THE ALGORITHM

A. The Application

The application GestCtrl that carried out the analysis of the cursor movements was written in Microsoft Visual Basic 6.0. The main form of the application is shown in Fig. 2 (picture on the right). When application is on desktop, the main form of the application presents to the user a minimum of information and controls. The size of the main form is only 110 pixels \( \times \) 90 pixels. After starting-up the application is hidden and stayed in the taskbar.

There are several commands and appropriate hotkeys to control the application. The form “Settings” (Fig. 2, picture on the left) can be activated by the corresponding label that become visible when the cursor is over the control. Each label has several functions: it is an indicator of the parameter and it is also a software button to change a value by a mouse click. Herewith, the left mouse button is used to increase the parameter and the right button to decrease any value with a predefined step. To assign action for a specific direction the user (carer) has to select a direction by clicking on “Dir. control” and “Dir. indicator” (red dot) will show the direction selected.

Then the user can click on “Action control” to make a selection from the list of actions (L_click, Dbl_click, R_click, exit). During the test assigned actions were non-functional and the taskbar on the Windows interface was hidden as well. As can be seen from Fig. 1, to cancel either selection movement the user can make a stop (until the waiting time of 2000 ms will expire that is indicated by the progress bar in Fig. 2), a fast movement to another location (when the velocity is more than 80 - “threshold 2” in Fig. 2) or can change the initial direction of the displacement vector.

B. The Experimental Setup

A desktop PC with Intel Pentium 4 CPU (3.2 MHz Cache 2MB) was employed in the experiment. The monitor used was a 19” AL1931 ACER on Matrox Millennium G550 that had a resolution of 1024 \( \times \) 768 pixels. Logitech QuickCam Pro 3000 was set to the black and white image mode with horizontal mirror and a frame rate of 30 fps. The camera-mouse module was implemented employing Microsoft Video for Windows (VfW) and API functions. An improved version of the cross-correlation algorithm for real-time template-based tracking was used [6]. The part of the entire image format of 320 \( \times \) 240 pixels where the face was located in a frame of 160 \( \times \) 120 pixels was centered and zoomed accordingly. The region of 48 \( \times \) 48 pixels near the eyebrows (the template) was used as a facial landmark being tracked. Usually, a transfer function of camera-mouse translates head displacements into coordinates of the pointer non-linearly, using the indexes of speed and acceleration [7]. To improve cursor control in a Windows application Kjeldsen and Hartman proposed a sigmoid-based transfer function for a parameter control task which combines head velocity and position [11]. Our camera-mouse module provided a relative pointing accuracy of screen resolution \( \pm 3,15 \) pixels (\( \pm 2 \) mm) with an acceleration factor of 2, velocity 2.5, and a moving average on 4 detected template locations. These indexes were chosen to make head movements smooth and convenient from the viewpoint of visually supported interaction on the monitor used.

Fig. 3 illustrates the layout of home targets and sample targets which are visible in adjusting mode. The position of each target could be changed as necessary before testing. The layout was designed in such a way that the participants should make head movements with different acceleration/deceleration. Screen distances used between the centers of the home target and the sample targets are shown in Table 1.

The goal of the experiment was to test whether the person can produce a subset of directional sub-movements to make gesture termination in a specific position without stopping and to evaluate a variability of the movement displacement vector (its magnitude and direction). According to the task, in each trial, two square targets of 30 \( \times \) 30 pixels appeared on the monitor: the home target and the sample target (Fig. 3(b)). Each of four sample targets (i-iv) in Fig. 3(a) had to be captured sequentially starting from the same

Fig. 2. The application program GestCtrl showing the settings adjusted.

Fig. 3. The experimental setup: the adjusting mode (a) and the typical position of the home target and the sample target (b). Black arrows (a) show the predefined directions for each target.
Table 1. The distances between the home targets and the sample targets used.

<table>
<thead>
<tr>
<th>The home target</th>
<th>Distance to the sample target, mm</th>
</tr>
</thead>
<tbody>
<tr>
<td>i</td>
<td>ii</td>
</tr>
<tr>
<td>1</td>
<td>95</td>
</tr>
<tr>
<td>2</td>
<td>135</td>
</tr>
<tr>
<td>3</td>
<td>155</td>
</tr>
<tr>
<td>4</td>
<td>175</td>
</tr>
<tr>
<td>5</td>
<td>185</td>
</tr>
<tr>
<td>6</td>
<td>180</td>
</tr>
<tr>
<td>7</td>
<td>125</td>
</tr>
<tr>
<td>8</td>
<td>95</td>
</tr>
<tr>
<td>9</td>
<td>70</td>
</tr>
<tr>
<td>10</td>
<td>70</td>
</tr>
</tbody>
</table>

The participants should move the cursor by head towards the home target. When the cursor crossed the home target the border color of the target had changed and the person heard a short sound feedback that indicated the starting position (p) in Fig. 4. Starting from this point, the participants were instructed to quickly move towards the sample target and to decelerate their head movement but by not stopping before crossing the edge, position (r) in Fig. 4. When cursor displacement vector was less than the predefined threshold \((\text{Thresh}1 = 10 \text{ pixels})\) in the sample target location, the person heard the sound feedback that indicated that the starting point for gesture termination was recorded, position (s) in Fig. 4. To accomplish the subtask the participant should move the cursor to cross the (blue) indicator of gesture termination direction and to produce the relevant directional gesture with consistent velocity until gesture termination was confirmed by sound feedback again.

The target selection occurred only when a predefined number of sub-movements (i.e. \(k\) in Fig. 1) was recorded: these were adjusted to be 2, 3 or 4 in different sessions. By sub-movements, we mean an amount of cursor movement, that is, the length of the track counted as an absolute displacement of the cursor recorded once in every 100 ms. By other words, it means that the user has to produce a different length of the directional gesture. The time limit was set to 2000 ms for gesture termination when no cursor displacements \((d > \text{Thresh}1)\) were recorded. The maximum displacement threshold \((\text{Thresh}2)\) was set to 80 pixels per 100 ms. Thus, the time limit and the upper bound of magnitude of displacement vectors had a minimum impact on continuous user activity such as head pointing and navigation. When the application detected significant cursor deviations from the predefined direction or direction of the previous sub-movement \((d_i \neq d_{i-1})\) before confirmation of the target selection or/and there was an angle of deviation of more than 22.5 degrees, the trial was considered as failed and an error was counted. Immediately after the error or cancel was detected, both targets were hidden. The error was accompanied by a negative sound. The new trial started automatically with randomly chosen the home target. Failed trials started again when the target number was randomly selected.

C. Participants

Still seven able-bodied volunteers from the local university took part in the experimental evaluation of the algorithm. All had normal or corrected to normal vision abilities, two participants wore glasses, and none had motor disabilities. Four of the subjects were males and three were females. The age of the subjects ranged from 26 to 52 years with a mean age of 37. The subjects used computer on a daily basis, reporting 6 to 8 hours of usage per day. Two of the subjects had a previous experience with a camera-mouse.

D. Procedure

Each session consisted of 40 trials starting from 10 home targets to make acquisition of 4 sample targets by directional gestures as indicated in Fig. 3(a) by the black arrows (Left, Right, Up and Down). Session duration was about 2-3 minutes. Still, the participants could rest at any time before the home target was activated. The participants were asked to repeat each session 10 times with different number of sub-movements randomly selected (2, 3 or 4). Thus, 8400 tracks were recorded: 7 persons × 10 sessions × 3 types of termination gestures × 10 home targets × 4 directions (sample targets). The experiment took a little more than one hour per participant. The data collected within each session were temporary stored in an array, automatically regrouped by the home targets and directions (sample targets), and finally were stored as MS Excel datasheet.

During preliminary testing of the algorithm with the head tracker we used eight directions and found that diagonal head-gestures have similar accuracy as orthogonal ones, but the speed of head movements (cursor displacements) was a little slower. We excluded diagonal gestures from the test because the goal of the experiment was to find the threshold values. The task was explained and demonstrated to the participants, and a warm-up session was given.
E. Results and Discussion

The test was similar to a shooting game enhanced with gestures. Being asked about issues, none of the participants reported neck strain or other usability problems regarding the experimental setup.

Fig. 4 illustrates the typical concatenated head gestures which were recorded during the sample target acquisition when the participant started from the home target. Both the cursor displacement \( (d) \) and the components \( dx \) and \( dy \) are presented in absolute (positive) values. The marker positions \( (p) \) and \( (r) \) in Fig. 4 indicate the moments when the cursor entered the home target \( 9 \) and the sample target \( 3 \) (Fig. 3) accordingly. The interval \( (q) \) illustrates a great variation of the displacements \( (dx \) and \( dy) \) of cursor position recorded with a sampling rate of 100 ms during the navigation towards the sample target. After getting the sample target the application is waiting for the deceleration of the displacement vector within the target, and less than the predefined threshold \( \text{Thresh1}=10 \) pixels). Cursor displacements made near the targets were more accurate as the subjects were asked to decelerate their head movement before entering the sample target. However, either directional gesture made during the navigational part could not cause an unintentional selection.

After the starting point \( (s) \) the application is waiting for the concatenated movements which satisfy to the condition: \( \text{Thresh1} < d \leq \text{Thresh2} \), as indicated in Fig. 1. As the person heard a short sound feedback at the point \( (s) \) s/he could start making the directional gestures immediately after deceleration. However, as can be seen from Fig. 4 and Fig. 5, a delay (about 400 ms) was systematically recorded between points \( (s) \) and \( (t) \). Because during the test each of the sample targets had to be selected using a specific directional gesture, we can only suppose that even a simple hint indicating the termination direction (the blue band) caused a decision-making delay, still without a stop. Deceleration after the first or second directional sub-movement did not cause the error. In this example, the person had to make a gesture termination by making a head gesture downwards, that is, displacements on y-axis \( (dy) \) mostly affected the results after the starting point – the cursor displacement \( (d) \).

The total average displacement recorded among all the participants throughout the test, within the epoch of analysis of \( (p-h) \) (as shown in Fig. 4) was about 37.8 pixels (STD = 25.8 pixels). Herewith, the total average maximum value was about 170.6 pixels (STD = 47.3 pixels). Three of the participants (2 women and 1 man) did the test more carefully. Their error rate at the beginning of the test was about 7.5%, their tracks were smooth and the total maximum average displacement did not increase to more than 136.9 pixels (STD = 23.8 pixels). The mean time for acquisition of the sample target \( (s-h) \) of these participants was of about 986 ms (STD = 157 ms). The other four persons (1 woman and 3 men) showed faster performance times, which were about 696 ms (STD = 121 ms), but they committed more errors \( (d_i \neq d_{i-1} \) before confirmation of the target selection), the error rate at the beginning of the test was about 23.2% and the total maximum average displacement was about 214.4 pixels (STD = 59.5 pixels). With practice all the participants significantly improved their skills and the error rate was decreased by about threefold.

In fact, the delimitation of four basic directions requires of 45 degrees of the deviation threshold. However, during the test there was no reason to cancel the trial and we could not receive any data concerning the cases of cancel selection. For able-bodied participants inaccurate control is not inherent, therefore the deviation threshold was decreased to 22.5 degrees. The errors had been detected when the displacement vector deviated more than 22.5 degrees from the initial displacement vector. The values of errors incurred indicated that the participants could manage by eight directions as well. Of course, a specific test could be performed concerning a false positive selection but not earlier than other parameters would be defined.

In Fig. 5 during one session the participant made a gesture termination by a head gesture downwards. The average displacement \( (d) \) recorded during 10 sessions by the same participant, within the epoch of analysis of \( (s-h) \) (see Fig. 4) was about 15.1 pixels (STD = 10.1 pixels). The average minimum displacement was about 3.38 pixels (STD=3.1 pixels); it was commensurable with the accuracy of the camera-mouse used. The average maximum displacement was about 59.6 pixels (STD = 25.3 pixels). The mean time

![Fig. 4. The typical concatenated gestures: (p) – getting the home target 9 (Fig. 3) and (q) – navigation towards the sample target 3 (Fig. 3); (t) – getting the sample target 3 (Fig. 3) and (r-s) – deceleration before a starting point; (s) – the starting point; (s-t) – decision making delay; (t, g, h) – performing the termination of selection by three sub-movements (samples 18-19, 19-20 and 20-21).](image-url)
Fig. 5. An acquisition of the same sample target starting from different positions of the home target using a termination of selection by 3 sub-movements. The starting point (s) in Fig. 4 was used to synchronize a superposition of tracks.

The starting point (s) for acquisition of the sample target was about 1078 ms (STD = 171 ms). The error rate was about 6.5%.

In comparison with dwell time used for making selection (600-900 ms), a gesture-based termination seems as less efficient technique in terms of the user performance. However usually with a head tracking device, the user can only make a single choice by dwell, while directional gestures can be associated with different commands.

Nevertheless, when navigational parts were excluded (Fig. 6), the paired samples t-test showed that there was not a statistically significant difference by direction between two groups of the participants ($t_2 = -2.34, t_3 = -2.33, t_4 = -2.63, df = 3, p > 0.07$) for 2, 3 and 4 sub-movements made to confirm a selection of the sample target accordingly. Difference was not found neither between the mean nor the maximum nor the minimum values. However, individual patterns recorded had greater variability (Fig. 5) both in duration and direction.

Three of the participants who did the test more carefully felt more satisfied, when they used longer gestures ($k = 3$). The paired samples t-test showed that there was a statistically significant difference between two conditions ($k = 2$) and ($k = 3$) of the gestures length $t(3) = -659, p < 0.01$, while no statistically significant difference was found between ($k = 3$) and ($k = 4$), $t(3) = 0.6, p > 0.5$. Similar results were obtained for the maximum and minimum values. However, the case ($k = 2$) seems impractical, as it is difficult to make cancel immediately after the first sub-movement was done. Other two conditions ($k = 3$ or $k = 4$) give more time to the user to make a decision. The total average time needed for confirmation of an acquisition of the sample target at $k = 3$ was about 972 ms (STD = 133 ms) and about 1090 ms (STD = 123 ms) at $k = 4$. But these results include a decision making delay (s-t) as shown in Fig. 4.

As can be seen from Fig. 6, when some people make head movements faster than the other participants, the length of four cursor displacements can significantly increase a possible range of movement and gesture termination will fail due to the screen-space limitation. For instance, the total average displacement for Down and Left direction was about 95.6 pixels (STD = 8.7 pixels) in the group of four participants against 61.3 pixels (STD = 8.8 pixels) within another group of three participants. Noteworthy that the camera-based head mouse had been used with acceleration 2 and velocity 2.5 during the test on 19” monitor with screen resolution of 1024×768 pixels. Thus, the amount of movement limits the use of a specific direction when the selected object is near to the bezel of the monitor. Depending on the task and cursor position, assigned options can adaptively be re-arranged as well as there is a possibility to make the cursor acceleration and velocity adaptive as well. This adaptation will also affect on minimum threshold ($\text{Thresh1}$).

IV CONCLUSION AND FUTURE WORK

We have proposed and explored a gesture termination algorithm for continuous video-based input techniques, such as head tracking. The method provides a multiple choice like a marking menu, as well as several ways to cancel a selection. The starting point of the directional gesture was detected using a threshold velocity of head movements.

The results of the test demonstrated that people could control head movement acceleration and an amount of head movement without significant efforts. Head gesture acceleration can be detected through analysis of the cursor displacement vector (magnitude and direction) and interpreted according to the application interaction protocol. The commands, which can be assigned for a specific direction, can be customized in advance using the settings of the application.

Based on the statistics obtained from the experimental data we concluded that the following parameters, $\text{Thresh1} = 10$ pixels, $\text{Thresh2} = 8 \times \text{Thresh1}$
and \( k = 3 \) or \( k = 4 \) (a number of the gesture submovements recorded with a predefined sampling rate of 100 ms), would be appropriate when the head tracker (camera-mouse) provides a relative pointing accuracy of screen resolution not worse than \( \pm 3 \) pixels (of \( 1024 \times 768 \) pixels). Herewith, an acceleration factor of 2, velocity 2.5 and moving average on 4 detected template locations were used. When another input device is being employed or the ability of the user is different, the critical parameters can be customized. Depending on movement ability of the user the number of directions for gesture termination can vary from 1 to 8 providing a suitable angular accuracy. A function of any mouse button can be associated with the particular direction or several ones. The thresholds (\( \text{Thresh}_1 \) and \( \text{Thresh}_2 \)) can be increased when user cannot produce accurate movements (due to tremor or other reason).

In the future research we plan that the particular features of the input device would be gathered by the application program during the game-like calibration process and they would be used to adaptively change the settings when necessary.

REFERENCES


