Buttonless Clicking:
Intuitive Select and Pick-release Through Gesture Analysis

Ali Choumane*
INRIA Lille
France

Géry Casiez †
LIFL & INRIA Lille
University of Lille, France

Laurent Grisoni ‡
LIFL & INRIA Lille
University of Lille, France

Abstract

Clicking is a key feature any interaction input system needs to provide. In the case of 3D input devices, such a feature is often difficult to provide (e.g. vision-based, or tracking systems for free-hand interaction do not natively provide any button). In this work, we show that it is actually possible to build an application that provides two classical interaction tasks (selection, and pick-release), without any button-like feature. Our method is based on trajectory and kinematic gesture analysis. In a preliminary study we exhibit the principle of the method. Then, we detail an algorithm to discriminate selection, pick and release tasks using kinematic criteria. We present a controlled experiment that validates our method with an average success rate equal to 90.1% across all conditions.

Index Terms: H.5.2 [User Interfaces]: Input devices and strategies

1 Introduction

Selection and positioning are two basic tasks of any 3D application that can be executed using buttons, gestures [3] or voice recognition [1]. Pinch to grab represents the most common use of glove for selection and positioning [2]. However most 3D input devices like 3D trackers have only three or six degrees of freedom (DOF) making them unsuitable for this interaction technique. Other devices, like the Gametrak$^{\text{TM}}$ and the EyeToy® have no button requiring the use of specific gestures or voice recognition to select options in 3D or move objects. Users are also reluctant to run the different calibration and training steps associated with gestures and voice recognition.

In this paper, we present a new approach based on trajectory and kinematic gesture analysis. We claim that selection, picking and releasing can be predicted based on the trajectory and kinematic patterns of the user's hand. The technique we propose requires no training or calibration and can be used with any 3D input device. In an initial study we analyze the trajectory and kinematic patterns of users using buttons to select and position 3D objects. Positioning starts with an object being picked and finishes with its release. We then propose an algorithm based on the invariants we found in the initial study to replace the use of buttons for selection, pick and release tasks. We last present a controlled experiment to evaluate our proposed algorithm.

2 Related Work

In 2D environment, the *dontclick*[1] project allows to navigate a website without using mouse buttons. This website principle is that when the cursor is on an item, it is selected. In 3D environment, Payne et al. [6] investigated issues affecting the usability and fun in the context of 3D gestures and video games. This work confirmed user benefits of 3D spatial gesture as a mean of interaction, such as intuitive movements linked to actions performed, as opposed to ”button bashing”. The shapewriter system [4] introduces an efficient way to combine keyboard stroke, using pen movement analysis. Stroke selection is identified by evaluating significant movement changes in pen movement. Although devoted to a specific task (stroke selection on a screen-displayed keyboard), this work can be seen as a major step in the direction of continuous gesture analysis for computer-human interaction simplification. Lank et al. provide estimators of gesture endpoint, using motion analysis [5].

None of the mentionned works proposed any method for combining selection and pick-release without button in the same application, which is the core contribution presented here. Next section presents the initial study we performed, that is at the origin of the proposed algorithm.

3 Initial Study

The purpose of this initial study is to investigate the trajectory and kinematic patterns associated to the select and pick-release tasks when they are performed using a button. We asked two participants

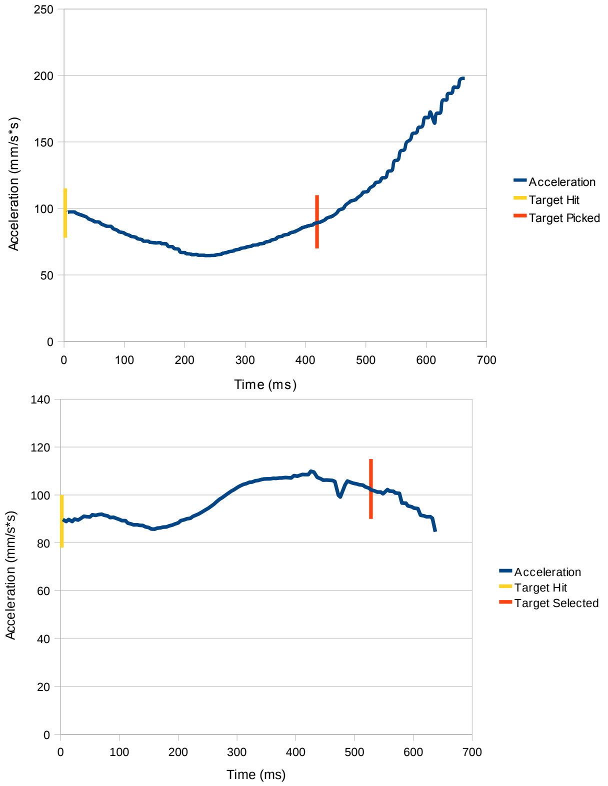

Figure 1: Acceleration profile for the pick (top) and selection (bottom) tasks when the cursor enters the target.

[1] http://www.dontclick.it/

*e-mail: ali.choumane@inria.fr
†e-mail: gery.casiez@lifl.fr
‡e-mail: laurent.grisoni@lifl.fr

to select, and pick-and-release spheres randomly positioned in 3D. The hand was tracked using a Gametrak device[2] and we used the button of a wireless mouse hold by the hand to perform the task. The Gametrak has a 3m cube workspace with a resolution ranging from 0.01 mm to 7 mm in all directions and a 125 Hz sampling rate. We collected a total of 160 trials for the analysis. For each condition (select, pick, release), we computed the mean velocity and acceleration profiles. This gave us a total of six graphs.

The velocity and acceleration profiles for the whole movement measured from the cursor first move to the target selection or picking did not show any specific pattern. Instead we observed discriminant patterns for the end of each movement when the pointer reaches the target.

Among the different profiles we computed, the most specific pattern we observed was for the picking. For this task, the device acceleration when entering a target first decreases until reaching a minimum and increases before the user presses the button. The corresponding profile is illustrated in Figure 1 (top). The time when the pointer enters the target is represented by the yellow bar and the red bar represents the time the button is pressed. Figure 1 (bottom) represents the corresponding acceleration profile for the selection task. We could not find a similar profile for the acceleration between the time the target is entered and the time the button is pressed.

This characteristic profile for the picking task could be explained by a planning of the movement trajectory before pressing the button. The trajectory is planned to enter the target and then exit it in a direction given by the expected drop position. Trajectory analysis confirmed that the distance between the pointer position and the target center increases as soon as the button is pressed.

For the selection task, the velocity profile (Figure 2) represents the most characteristic profile. The velocity keeps decreasing once the target is entered (represented by the yellow bar) until the button is pressed down (red bar) and up (end of the blue curve). The average speed when the button is pressed down is equal to $21 mm.s^{-1}$ (SD = $8mm.s^{-1}$) and the duration of the click (button down and up) was measured to be equal to 180ms.

For the release task, we observed that the average speed when the button up event occurs is equal to $14mm.s^{-1}$ (SD = $7mm.s^{-1}$). For both selection and picking tasks we observed that the average velocity when entering a target is equal to $75mm.s^{-1}$ (SD = $40mm.s^{-1}$).

In the next section we propose an algorithm taking into account the observations of the study presented in this section to predict the select, pick, and release tasks.

4 Proposed Algorithm

Based on the results of the preliminary study, we propose the state transition diagram as illustrated in Figure 3 to select, deselect, pick, and release objects.

As we can only discriminate the acceleration profiles when a target is entered, our algorithm first checks if the pointer is inside a 3D object that can be selected or picked (state 0). When the target is entered (state 1), the velocity, acceleration and trajectory profiles are analyzed.

In state 1, we check if the target is intended to be selected or picked by measuring the device speed. If it is above $V_{threshold1}$ we consider the user passing through a target without any intention of interacting with it. We used a value of $115mm.s^{-1}$ for $V_{threshold1}$ as the preliminary study showed that the mean velocity when entering a target is equal to $75mm.s^{-1}$ with a standard deviation equal to $40mm.s^{-1}$. If the device velocity remains below $V_{threshold1}$ we then discriminate between the select and pick tasks.

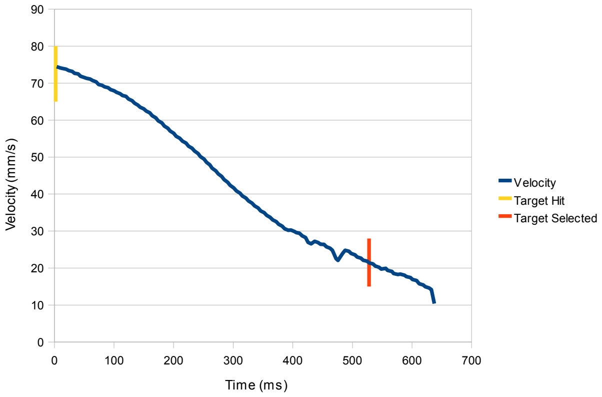

Figure 2: Velocity profile for the selection task when the cursor enters the target.

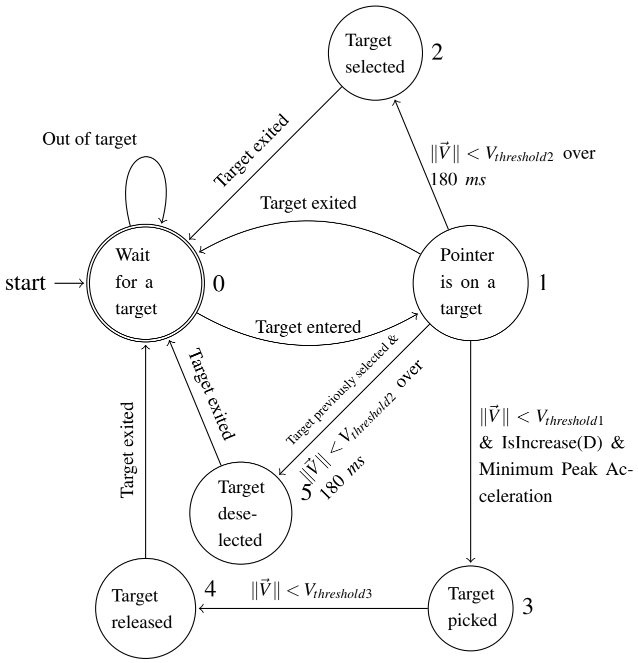
Target selected
2
Out of target
Target exited
$\|\vec{V}\| < V_{threshold2}$ over 180 ms
Target exited
start
Wait for a target
0
Pointer is on a target
1
Target entered
Target exited
Target previously selected & $\|\vec{V}\| < V_{threshold2}$ over 180 ms
5
Target deselected
Target exited
$\|\vec{V}\| < V_{threshold1}$ & IsIncrease(D) & Minimum Peak Acceleration
Target released
4
$\|\vec{V}\| < V_{threshold3}$
Target picked
3

Figure 3: The state transition diagram representing the algorithm to predict select, deselect, pick, and release tasks. D is the distance between the current pointer position and the center of the entered target.

The picking task is conditioned by the two criteria found in the initial study: a minimum peak in the acceleration profile and an increase in the distance to the target center (*isIncreased(D)*). The increase in distance is checked after the minimum peak acceleration is detected. If the two conditions are fulfilled the target is picked (state 3).

For the selection task, we check if the device speed remains below a speed threshold $V_{threshold2}$ over a period of time. According to the initial study we found an average velocity equal to $21mm.s^{-1}$ (SD = $8mm.s^{-1}$) when the button is pressed down. We use a speed equal $29mm.s^{-1}$ for $V_{threshold2}$ and use 180ms for the time period. The target state (selected or deselected) is then toggled.

The target release occurs when the device speed drops below a third threshold speed $V_{threshold3}$.We used a value of $7mm.s^{-1}$ as the mean value found in the initial study with the corresponding standard deviation.

[2] http://www.pdp.com

The detection of the pick task depends, in part, to the function *IsIncrease*(D). As mentioned above, this function allows to predict that the user is moving out from the target. In the formal experiment (cf. section 5) targets were spheres. Hence D is the distance between the sphere center to the pointer position. We assume that our algorithm is adapted for targets with a geometric form that can be easily enclosed in a sphere, for example, pyramid, cube, cylinder. For these forms, D is well defined.

5 Experiment

Our goal is to evaluate the effectiveness of our algorithm by measuring the percentage of correctly recognized actions for each subtask: selection, pick, and release. As our algorithm is based on kinematic gesture analysis, we also want to assess its robustness on parameters that can affect the kinematic profile: target size (the size of the object to select or pick), destination size (the size of the zone to release the picked target) and target orientation (measured as the angle formed by the movement starting point, the target position and the horizontal). The target size is known to affect the velocity profile with small sizes requiring more accuracy and thus reducing the corrective movement speed [7]. The target orientation affects the relative displacement of the hand which can make a difference in the velocity profile. Our goal is also to compare our technique to the button alternative in terms of performance (movement time and error rate) and subjective preference.

Participants

Four female and four male with a mean age of 26 (SD=1.8) participated. Participants had an average forearm length equal to 27 cm (SD=3), arm length equal to 29 cm (SD=2.5), and an average height equal to 173 cm (SD=9). All participants were right-handed and had normal or corrected-to-normal vision. Two participants had little experience and six had no experience with virtual reality and 3D applications but this is acceptable as we are observing a lower level physical behavior. Among the participants fours were computer scientists, three were electronic engineers and one was medical doctor. None of them participated in the initial study.

Apparatus

The experiment was conducted using a retroprojected 6500 mm (256 inch) large curved screen with stereoscopic display using a 2344 × 1050 pixels resolution (202 × 203 mm), 96 DPI pixel density, and 120 Hz refresh rate. The hand position was tracked using a DTrack device with an ARTtrack infrared-based optical tracking system[3]. The ARTrack system gives a 0.06 mm positional resolution in all directions, 0.4 mm positional accuracy and a 60 Hz sampling rate. The DTrack device was positioned on top of the hand. For the button condition, we used the same apparatus except participants hold a wireless mouse in the hand that was tracked. The left mouse button was used for interaction.

Task

We used two tasks in our experiment: a multi-directional pointing task and a multi-directional pick-and-release task. The two tasks were evaluated with and without button (figure 4). For each task we used four targets evenly distributed on a circle positioned at the center of the screen in a plane parallel to the screen. The pointer was represented as a pink sphere with a diameter equal to 2 mm measured at the center of the circle. We used a constant gain to map the hand position to the 3D cursor position.

For the selection task, the target to select appeared in red while the other targets remained grey. Upon successful selection of the target, it disappeared and the next target to select turned red. Picked targets were counted as errors and had to be released anywhere before the next target to select turned red.

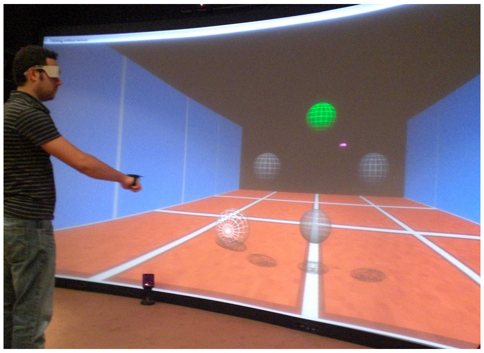

Figure 4: Picture of the experiment setup for the pick-and-release task showing our experimental hardware with the large screen, the DTrack device fixed to the hand of the participant, the targets to pick-and-release and the destination on the floor.

We used a similar scenario for the pick-and-release task except the target to pick appeared green and we used a sphere at a pseudo-random position on the floor representing the destination location to release it. The destination sphere appeared white and turned green when the target was fully inside. Targets that were inadvertently released were counted as error and had to be picked again until released at the correct destination. Targets inadvertently selected were counted as errors before the next target to pick turned green.

In addition to perspective and occlusion, we added shadow projection on ground for the pointer, targets and destination to improve depth perception. The camera remained fixed during the whole experiment. Participants were instructed to perform the tasks as accurately as possible. For all participants, the parameters of our model were set to the same values given by the initial study.

Design

A repeated measures within-subjects design was used. The independent variables were Technique (system used with or without button), Task (Select, Pick, Release), Orientation (0°, 90°, 180°, 270°), Target Size (6 mm, 12 mm and 24 mm - measured at the center of the circle) and Destination Size for the release task (10 mm and 20 mm greater than the target size). The orientation is measured from the horizontal axis counter-clockwise (the 0° is positioned at the right of the scene). The pick-and-release task is decomposed into two sub-tasks: the pick task and the release task.

Trials were organized in Blocks to measure the learning effect. We used six blocks for the no-button technique and two blocks for the button technique. Each Block was composed of the three Task evaluated each time with three Target size, four Orientation and the two destination Size for the Release task. Task and Technique were counter-balanced across participants. This gave us a total of 8 × 480 = 3,840 total trials. The experiment lasted approximately 60 minutes.

6 Results

The dependent variable are movement time and success rate.

6.1 Movement Time

Trials marked as error were removed from movement time analysis. Repeated measures analysis of variance found no significant effect ($F_{1,7}$ = 0.017, p = 0.9) for Technique, Task ($F_{2,14}$ = 3.426, p = 0.061) and no significant interaction between Technique and Task ($F_{2,14}$ = 2.91, p = 0.088) on movement time. These results

[3] http://www.ar-tracking.de/

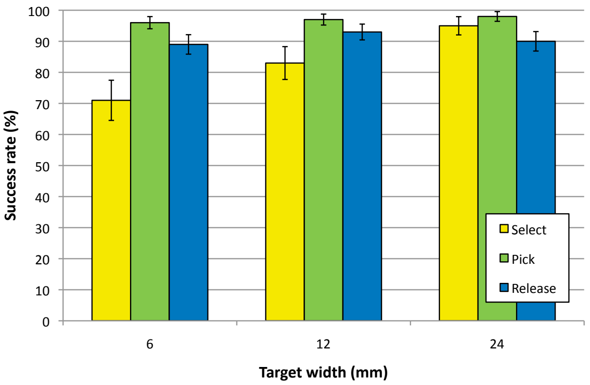

Figure 5: Mean success rate for each Task and Target size for the no-button technique. Error bars represent 95% CI.

lead us to conclude there is no degradation in the movement time for the no button technique. Further evaluations would be required to show that no small effect exists.

6.2 Success Rate

The success rate for the Release Task is computed from the successfully picked targets. Repeated measures analysis of variance found no significant effect for Block on the success rate showing no learning effect.

Repeated measures analysis of variance found a significant main effect for Technique ($F_{1,7} = 67.5$, $p < 0.0001$) and a significant interaction between Technique and Task ($F_{2,14} = 9.0$, $p = 0.03$) on success rate. The overall success rate is 96.2% with button and 90.1% without button. Pairwise comparison show significant differences for the Pick Task (p=0.004, 99.7% success rate with button and 96.7% without button) and Select Task (p=0.024, 96.2% success rate with button and 82.9% without button). The Release Task shows no significant difference (p = 0.37) with a 92.6% success rate with button and 90.7% without button. To better understand the factors influencing the success rate without button, we removed the button data for subsequent analysis.

Repeated measures analysis of variance found a significant main effect for Task ($F_{2,14} = 41.2$, $p < 0.0001$), Target Size ($F_{2,14} = 17.4$, $p < 0.0001$) on success rate and significant interaction between Task and Target Size ($F_{4,28} = 10.8$, $p < 0.0001$) on success rate. Pairwise comparison show significant differences ($p < 0.018$) between the different target sizes of the Select Task with 70.5% success rate for the 6 mm target, 82.8% for the 12 mm target and 95.4% for the 24 mm target. Repeated measures analysis of variance found a significant main effect for Destination Size ($F_{1,7} = 5.9$, $p = 0.045$) on success rate with 88.5% for the smallest width and 93% for the largest width.

7 Subjective Results

After each block, participants rated their fatigue level using a 5 points Likert scale (1:no fatigue, 5:high fatigue). The mean value across participants remained consistently around 3 throughout the blocks showing no increase in fatigue.

At the end of the experiment, we asked participants to give their favorite technique. Half the participants declared to prefer the technique without button and the other half with button. Participants who preferred the technique with button mentioned that it is closer to mouse-based interaction they are familiar with. From the informal qualitative feedbacks we noticed that participants were first surprised about the idea of selecting and moving object without button but they quickly adopted and enjoyed the technique. We observed that sometimes people intuitively closed their hand to pick the target. One participant said: ”it's as if the computer reads my mind”, another one said : ”it's much more fun without button”.

8 Discussion

Overall the experiment validated our algorithm with an average success rate equal to 90.1% across all conditions. This result is lower than the 96.2% success rate with the button technique but we were pleased to see that our algorithm requires almost no learning and was robust to all participants with the settings defined in the initial study. We evaluated the influence of the orientation, target size and destination size as confounding parameters. The experimental results show that the orientation has no influence on the recognition rate while the target size significantly affect the Selection success rate with smaller target sizes decreasing the recognition rate. In addition the results on movement time indicate that the no button technique is equivalent to the button technique and the results on fatigue did not show any specific fatigue associated to the no button technique.

These results show that our method represents a valid alternative to the use of buttons for selection and pick-and-release tasks in applications where the targets size is sufficiently large. Our results also show there is space for improvement in our algorithm to increase the recognition rate. An area for improvement is to take into account the target size in our algorithm.

9 Conclusion and Future Work

We presented a new approach for discriminating between selection and picking-and-releasing tasks based on trajectory and kinematic analysis, especially useful for tracking systems having no button. We detailed the algorithm and validated it in a controlled experiment that shows high recognition rates.

As future work we first plan to tune the algorithm for small target size. We then want to check if our algorithm can simply apply to object of any shape by taking their bounding sphere or if we need to refine the algorithm depending of the object shape.

Acknowledgments

This work was supported by the ANR project n°ANR-07-TECSAN-020 ”REACTIVE”.

References

[1] R. A. Bolt. “put-that-there”: Voice and gesture at the graphics interface. In *SIGGRAPH*, pages 262–270, New York, USA, 1980. ACM.

[2] D. A. Bowman, C. A. Wingrave, J. M. Campbell, V. Q. Ly, and C. J. Rhoton. Novel uses of pinch gloves for virtual environment interaction techniques. *Virtual Reality*, 6(3):122–129, 2002.

[3] S. Carbini, J. E. Viallet, and L. DelphinPoulat. Context dependent interpretation of multimodal speechpointing gesture interface. In *ICMI Doctoral Spotlight &Demo Papers*, pages 1–4. ACM, 2005.

[4] P. O. Kristensson and S. Zhai. Command strokes with and without preview: using pen gestures on keyboard for command selection. In *CHI*, pages 1137–1146, New York, NY, USA, 2007. ACM.

[5] E. Lank, Y.-C. N. Cheng, and J. Ruiz. Endpoint prediction using motion kinematics. In *CHI*, pages 637–646, New York, NY, USA, 2007. ACM.

[6] J. Payne, P. Keir, J. Elgoyhen, M. McLundie, M. Naef, M. Horner, and P. Anderson. Gameplay issues in the design of spatial 3d gestures for video games. In *CHI*, pages 1217–1222, New York, USA, 2006. ACM.

[7] D. Wisleder and N. Dounskaia. The role of different submovement types during pointing to a target. *Experimental Brain Research*, 176:132–149, 2007.