Clutching Is Not (Necessarily) the Enemy

Mathieu Nancel
Cheriton School of Computer Science
University of Waterloo, CANADA
mnancel@uwaterloo.ca

Daniel Vogel
Cheriton School of Computer Science
University of Waterloo, CANADA
dvogel@uwaterloo.ca

Edward Lank
Cheriton School of Computer Science
University of Waterloo, CANADA
lank@uwaterloo.ca

ABSTRACT
Clutching is usually assumed to be triggered by a lack of physical space and detrimental to pointing performance. We conduct a controlled experiment using a laptop trackpad where the effect of clutching on pointing performance is dissociated from the effects of control-to-display transfer functions. Participants performed a series of target acquisition tasks using typical cursor acceleration functions with and without clutching. All pointing tasks were feasible without clutching, but clutch-less movements were harder to perform, caused more errors, required more preparation time, and were not faster than clutch-enabled movements.

Author Keywords
clutching; pointing; performance; trackpad; touchpad

ACM Classification Keywords
H.5.2. Information interfaces (e.g., HCI): User Interfaces.

INTRODUCTION
Relative pointing is the mapping of physical motion to cursor displacement using a Control-to-Display (CD) transfer function. It is common with mice and trackpads and proven efficient for mid-air interaction [1, 6, 7]. One characteristic is that a discrepancy builds between the relative location of the cursor on the display and the position of the input device (e.g. finger on trackpad). To stay within an optimal range of input sensing or comfort, this discrepancy must be reset at regular intervals with a 'clutch'. This is a break in control-to-display mapping when the input device can be moved independently of the cursor (e.g. lifting the finger to reposition it on a trackpad). Based on observed correlations between increased pointing time with increased clutching [1, 3], the belief is that clutching impairs performance since input time is "lost" when clutching. For this reason, researcher and designers typically view clutching as the 'enemy', seeking to avoid it entirely or minimize its frequency or effect.

Avoiding clutching completely usually requires using absolute mappings [5–7] or rate-based mappings [3]. Since clutching is expected when there is not enough physical input space to perform a pointing movement [2,3], a commonly proposed solution is to increase the CD gain [2, 6] since low gains cause more clutches. However, the interaction between CD gain, clutching, and pointing performance is not that trivial. For example, Chapuis *et al.* observed performance decreased with low gains even without clutching [4]. Since clutching cannot be avoided with relative pointing, work has considered it when designing transfer functions and performance modeling [5] and methods have been proposed to improve the clutching mechanism itself [8] or make up for lost pointing time by gliding the cursor during a clutch [1].

Yet despite encouraging results for these reduced-clutching pointing techniques [1, 3, 6], what causes clutching and whether clutching is responsible for lost performance remains hypothetical: (i) Casiez *et al.*'s model of clutching [3] was originally evaluated in limited input and task conditions, and in [1] Beaudouin-Lafon *et al.* were unable to show that this model fits their data better than Fitts' law despite a significant number of clutches; (ii) the clutch-avoidance effectiveness of recent approaches [2, 6] remains to be quantified, and (iii) it remains unclear whether the performance gain in clutch-minimizing techniques is entirely due to reduced clutching, or simply due to faster cursor movements.

We investigate the cause and effect of clutching independently of the transfer function. We ran a controlled experiment using a laptop trackpad where all target acquisition tasks could be performed without clutching. In the first condition, clutching was permitted to observe if participants clutched by choice and how clutching was affected by the transfer function. In the second condition, clutching was not permitted to produce hypothetically ideal clutch-less motions to directly compare performance with and without clutching.

Our results indicate no clear time improvement between movements with "unnecessary" clutching and ideal clutch-less movements on a trackpad. Yet, clutch-less movements are harder to produce, cause twice as many errors, and require more planning time. Therefore clutch-less movements are not necessarily an optimal pointing strategy. This calls for a reconsideration of clutching and for deeper investigation of its causes and effects in relation to input devices, transfer functions, physical space, and input speed.

MOTIVATION

Causes of clutching
Clutching is usually attributed to insufficient physical space to make the cursor travel a given distance with a transfer function [3], whether the function is linear [2] or acceleration-based [6]. This is easy to verify with linear functions, as cursor motion with constant CD gain is only affected by the physical amplitude of the input movement (provided that input events and cursor updates have the same frequency). This

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
CHI 2015, April 18–23, 2015, Seoul, Republic of Korea.
Copyright is held by the owner/author(s). Publication rights licensed to ACM.
ACM 978-1-4503-3145-6/15/04...$15.00
http://dx.doi.org/10.1145/2702123.2702134

effect is less trivial to model with acceleration because the amplitude and speed of the input movement are co-dependent when considered in the resolution of cursor motion: user performance might vary depending on ability to perform fast input motions or to control wide ranges of speed, but approximations can be made using the highest accessible gains [2,6]. In these cases, clutching is unavoidable and prior research suggests minimum bounds for gain values to avoid it [2], based on the sensing or operating range of the input device. Another plausible purpose for clutching is to minimize the control-display discrepancy by readjusting the location of the user's limb relative to the input sensor or to the user's body. This cause is similar but depends on the *remaining* physical space available in the sensing range in the direction of the pointing motion, rather than its total size.

We suspect clutching is caused by more than mechanical limitations. This is based on informal observations: the acceleration settings of operating systems generally allow the cursor to go through the whole display with a short wrist or finger flick, often of less than half the operating range of the mouse or trackpad; yet users using such normal settings appear to clutch more often that would be accounted for by simply needing to minimize the control-display discrepancy. Similar observations were reported for mid-air pointing [1,6], where participants clutched even though the transfer functions theoretically allowed them to perform all tasks without clutching. We evaluate whether this informal observation holds with the more common laptop trackpad, and if so, whether clutching really reflects suboptimal pointing behavior.

Effects of faster transfer functions

Clutching is assumed to lower pointing performance and it is often advocated to avoid it when designing relative pointing techniques [2,3,6]. A common guideline is to use higher CD gain values, because these generally result in better performance. However it is unclear why. Less clutching and faster cursor motions are both trivial consequences of higher gains, but the exact causation from these intermediate effects to increased performance remain to be quantified. We need to control for clutching when evaluating the effects of faster transfer functions, but even more strictly than Chapuis *et al.* [4].

Effects of clutching on performance

The specific assumption that clutching lowers pointing performance remains to be formally asserted and quantified. Since time is spent clutching while the cursor remains static, it must be that performance is reduced (barring inertial pointing, like [1] or trackballs). However, there are other aspects to consider. For example, the effective operating range decreases, potentially lowering the number of joints and muscles in use. Also, people may unconsciously optimize the amount of effort or attention required to perform difficult tasks like selecting small targets over large distances. This requires both high and low input speeds, and clutching could be used to divide the pointing movement into sub-movements of more controllable speeds [5]. Therefore, to assess the effectiveness of the hypothetically optimal clutch-less behaviour, we need to evaluate conditions where participants are specifically instructed not to clutch.

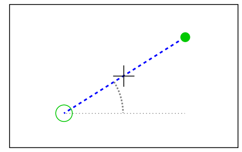

Figure 1. Pointing task: all tasks were centred around the screen centre; participants had to click the pre-target then the target. Targets scale increased for readability and in experiment, the background was black.

STUDY

Our experiment investigates four research questions:

Q1 Is clutching only triggered by transfer functions that are too slow or the physical range being too small?

Q2 What are the effects of transfer functions on clutching?

Q3 At equal difficulty and transfer function, is there a systematic advantage to performing clutch-less movements?

Q4 What are the effects of clutching on performance?

Participants

We recruited 12 participants (1 female), aged 23 to 39 (mean 26.8, SD 4.4), all right-handed and using computers more than 30 hours a week (mean 51.4, SD 15.9).

Apparatus

Participants used a MacBook Pro Retina with 115 × 85 mm trackpad and OSX 10.8 acceleration settings. In order to accurately differentiate tapping events from clutch events, we disabled tap-to-click and instructed participants to click using the embedded physical button on the trackpad. Thus, a clutch was detected any time participants lifted their finger. Software was developed in Java 7 using the Java Native Interface[1] to access low-level trackpad events.

Task and Stimuli

The task was consecutively clicking inside a *pre-target* (circle, ⌀10 mm) then a target (disc) of variable width and distance (Fig. 1). Both were displayed at the beginning of the task trial; the pre-target disappeared when it was clicked. We ignored the time taken to reach the pre-target in our analyses. We chose a randomized target presentation over a repetitive, reciprocal one for two reasons. First, people do not necessarily plan pointing trajectories far ahead, but trajectories can be artificially learned with repetitive tasks. Second, this ensures that pointing motion is not constrained by the physical location of the hand at the end of the previous task.

Design

We used three ACCELERATION levels: 5th, 7th and 9th notches on the slider of the Mac OS 10.8 trackpad speed settings. Participants first performed 50 trials with each ACCELERATION level as accurately and quickly as possible without any specific instruction about clutching (CLUTCH condition). The cursor was a white cross. After we briefly explained clutching, participants were instructed to perform the same trials without clutching between the pre-target and target (NOCLUTCH condition). In pilot studies, participants tended to overlook

[1] http://docs.oracle.com/javase/7/docs/technotes/guides/jni/

the NoClutch instruction after a while, so the cross cursor was surrounded with a white circle as soon as they clicked the pre-target, as a visual reminder. If a clutch was detected in the NoClutch condition, the whole cursor turned red and thicker until they resumed pointing. We discard NoClutch trials with clutches from analysis (i.e. when this extra feedback is shown). Clutches occurring before the pre-target was clicked are not considered.

The ordering of the CanClutch factor was not counterbalanced. Pilot tests showed that pointing on a trackpad is such a common task that no amount of training could prevent normal pointing habits (i.e. Clutch) to influence the NoClutch condition. Conversely, pilot tests confirmed that practising with NoClutch does affect further “normal” pointing behaviour. While interesting, the real-life implications of that effect do not answer our research questions. The first 15 trials of all CanClutch $\times$ Acceleration conditions were excluded as training, resulting in 2 CanClutch{Clutch, NoClutch}$\times$ 3 Acceleration {Acc5, Acc7, Acc9}$\times$ 35 trials = 210 data points per participant. The experiment lasted 40 min, participants did not report substantial fatigue.

Distributions of Independent Variables

To minimize loss of information or effect, we generate uniform distributions for Width, Amplitude, Fitts's ID, and direction to obtain an all-inclusive, equally-likely sample of the observed effect (rather than extrapolate from a regression line based on a small number of aggregates). Widths are chosen to be realistically hard to acquire, from the size of a desktop icon (12 mm) to a very small size visible with normal eyesight (1.5 mm). Amplitudes are chosen to reflect realistic pointing tasks on a 13” laptop (43 to 239 mm on a 285 $\times$ 175 mm screen) while limiting edge pointing, known to affect pointing behaviour and performance.

Since $ID = f(A, W)$, the uniformity of all three variables cannot be strictly maintained. We focus on Fitts's ID and relax the uniformity constraint on Amplitude and Width as follows. In all trials, Fitts's ID is chosen randomly between 2.2 and 7.32 bits (best- and worst-case scenarios). In odd trials, Width is chosen randomly between 1.5 and 12 mm and Amplitude is computed as $A = W(2^{ID} - 1)$. In even trials, Amplitude is chosen randomly between 43 and 239 mm and Width is computed as $W = \frac{A}{2^{ID}-1}$. In all trials, direction angle is chosen randomly: pre- and actual targets are centred on screen Amplitude mm apart on a random angle between -180 and 180 degrees (Fig. 1); if the resulting targets are partially or completely out of the screen, a new angle is chosen. Pilot tests revealed the trackpad button is much harder to use when near the top of the trackpad, especially problematic with NoClutch. To compensate, we added an additional condition to angle selection: the target had to be more than 15 mm from the top of the display.

This produced uniform distributions for Fitts's ID (M 4.74 bits, SD 1.45 bits). The distribution of angles was affected by the screen aspect-ratio, favouring horizontal angles for high Amplitude (M 7.4 deg, SD 106.7 deg). As expected distributions were skewed for Amplitude (M 130.6 mm, SD 63.5 mm, skewness .17) and Width (M 5.6 mm, SD 3.4 mm, skewness .41), which remain close to uniform distributions.

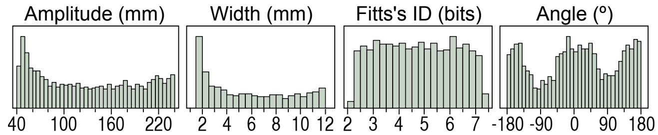

Figure 2. Distributions for A, W, ID and angle.

Results

We found no order effect of Acceleration on task time. All analyses are multi-way ANOVA: participant is a random variable using the REML procedure of the SAS JMP package. Post-hoc Tukey tests are used unless stated otherwise.

Amount of clutching

We found significant effects of CanClutch ($F_{1,11} = 36.5$, $p < 0.0001$), Acceleration ($F_{2,22} = 15.4$, $p < 0.0001$) and CanClutch$\times$Acceleration ($F_{2,22} = 14.6$, $p < 0.0001$) on the number of clutches. Post-hoc tests show that Acceleration did not significantly affect the number of clutches in the NoClutch condition (mean .2 clutches per trial), that NoClutch caused significantly less clutches than Clutch (mean 1.24 clutches per trial) overall and that, in the Clutch condition, Acc5 caused significantly more clutches (mean 1.6 clutches per trial) than Acc7 (1.18) and Acc9 (.94). A nominal logistic test showed that Fitts's ID had a significant (decreasing) effect on whether participants clutched or not in the Clutch condition ($\chi^2 = 57.8, p < .0001$). Overall, 14.7% of the NoClutch trials had one or more clutches: 14.9, 17.6 and 11.5% for Acc5, Acc7 and Acc9.

Error rate

We consider the Error rate as the ratio of trials where one or more clicks missed the final target. We found a significant effect of CanClutch on Error ($F_{1,11} = 7.1, p = .0222$) and no effect of Acceleration. A Student t-test revealed that participants made almost twice as many errors in the NoClutch phase (4.2%) than in the Clutch phase (2.6%). All trials with Errors are ignored in the following analyses.

Time performance

We measured Movement Time (Mt) as the time between clicking the pre-target and clicking the target, Preparation Time (Pt) the time between entering the pre-target and clicking it, and Start Time (St) the time between clicking the pre-target and leaving it. No aggregation was performed before running regressions with Fitts's ID, on account of our continuous Fitts's ID range. For other within-subject ANOVA, we used medians to compensate for skewed distributions.

We excluded 183 NoClutch trials where a clutch occurred (14.7%) in order to obtain a corpus of ideal clutch-less pointing motions regardless of the cursor transfer function. Clutchless motions also occurred in the Clutch condition, more often (but not exclusively) in the easier tasks, as reported above.

We found no significant effect of CanClutch and Acceleration on Mt, and an equivalence test with 200 ms threshold finds Mt equivalent for Clutch and NoClutch ($p = .0406$).

Fitts's law regressions are very similar: $\text{MT} = 461.6 + 394.8 \times ID (r^2 = .41)$ for NoClutch and $\text{MT} = 348.6 + 433.8 \times ID (r^2 = .39)$ for Clutch, see Fig. 3.

Neither CanClutch and Acceleration had any significant effect on ST, but PT was significantly affected by CanClutch ($F_{1,11} = 7.8, p = .0176$) and CanClutch×Acceleration ($F_{2,22} = 5.5, p = .0113$). Participants took significantly more time to click the pre-target in the NoClutch condition (834.7 ms) than in the Clutch condition (654.1 ms). More precisely, the condition Acc5×NoClutch (936 ms) was significantly slower than every other combination (mean 676.2 ms) except Acc7×NoClutch (825.6 ms). Fitts's ID also has a significant effect on PT for Clutch ($F_{1,11.85} = 6.42, p = .0265$, $\text{PT} = 590.8 + 25.9 \times ID$) and NoClutch ($F_{1,9.48} = 33.22$, $p = .0002$, $\text{PT} = 610.2 + 68.3 \times ID$), see Fig. 3.

We hypothesize that participants spent more time above the pre-target in the NoClutch condition to prepare for difficult tasks, either to relocate their finger on the trackpad or to plan a fast movement. This indicates that pointing actions started before the pre-target was clicked. We thus introduce another time measure, Task Time (TT=MT+PT). We found no significant effect of CanClutch or Acceleration on TT, and an equivalence test with 200 ms threshold finds TT equivalent for Clutch and NoClutch ($p = .0073$). Again, Fitts's law regressions are very similar: $\text{TT} = 1071.8 + 463.1 \times ID, r^2 = .38$ for NoClutch and $\text{TT} = 939.4 + 459.6 \times ID, r^2 = .31$ see Fig. 3. On average, TT was higher for NoClutch (Fig. 4).

Finally, CanClutch had a significant effect on cursor-moving time ($F_{1,11} = 8.2, p = .0152$), which was significantly higher in the NoClutch condition (3606.8 vs. 3391.5 ms) (Fig. 4).

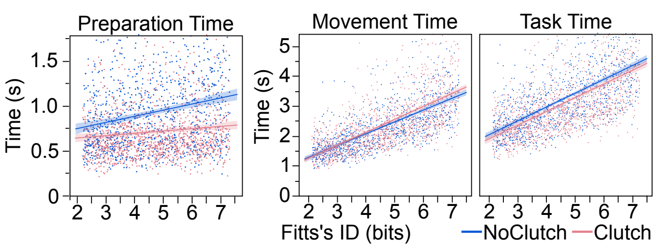

Figure 3. Fitts's law regressions for Preparation Time (PT), Movement Time (MT) and Task Time (TT). Outlines represent the confidence of fit.

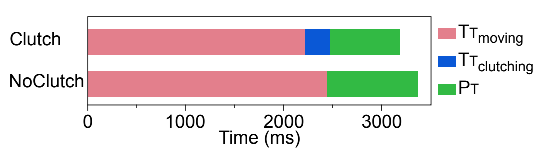

Figure 4. Mean Task Time when moving the cursor ($\text{TT}_{\text{moving}}$) and when clutching ($\text{TT}_{\text{clutching}}$), and Preparation Time (PT).

Summary

Almost all participants felt that the Clutch condition was faster (P9 felt otherwise), more precise (P2 felt otherwise), easier to use, less tiring (P2 felt otherwise), and preferred it over NoClutch. All aspects rated on 7-point Likert scales. One participant (P4) felt no difference at all since they always try to avoid clutching.

All tasks were feasible with a single finger motion, yet we found that participants clutched often when not instructed otherwise (M 1.24 clutch/trial, **Q1**). Unsurprisingly, lower pointer accelerations increased clutching (**Q2**), but faster accelerations did not cancel it (.94 clutch/trial with Acc9). Despite previous assumptions, we did not find explicitly clutch-less pointing motions to be significantly beneficial to pointing time. On the contrary, they caused twice as many pointing errors (**Q3**). Clutch-less movements were also harder to produce (14.7% failed) and necessitated more preparation time. Despite not "loosing" any time from clutches, the time spent actually moving the cursor ($\text{TT}_{\text{moving}}$ in Fig. 4) was about 200 ms longer in average with clutch-less pointing (**Q4**).

DISCUSSION AND FUTURE WORK

These results challenge the current belief that clutching is detrimental to pointing performance. Clutching is a favoured and effective trackpad pointing strategy, not just a method to compensate for limited physical space. The hypothetically optimal clutch-less movements turn out harder to perform, causing more errors and no time improvement. Possible causes, including optimization of motor resources (fewer joints and muscles put in use) or psycho-motor resources (splitting a difficult movement into more controllable chunks) will be investigated in future works. This work is a first step towards formulating a more comprehensive model of clutching and pointing. Given these encouraging first results, we are pursing expanded analysis of different input devices, transfer functions, and tasks where clutching is unavoidable.

REFERENCES

1. Beaudouin-Lafon, M., Huot, S., Olafsdottir, H., and Dragicevic, P. Glidecursor: Pointing with an inertial cursor. In *Proc. AVI'14*, ACM (2014), 49–56.
2. Casiez, G., Vogel, D., Balakrishnan, R., and Cockburn, A. The impact of control-display gain on user performance in pointing tasks. *HCI 23*, 3 (2008), 215–250.
3. Casiez, G., Vogel, D., Pan, Q., and Chaillou, C. Rubberedge: Reducing clutching by combining position and rate control with elastic feedback. In *Proc. UIST'07*, ACM (2007), 129–138.
4. Chapuis, O., and Dragicevic, P. Effects of motor scale, visual scale, and quantization on small target acquisition difficulty. *ACM ToCHI 18*, 3 (Aug. 2011), 13:1–13:32.
5. Nancel, M. *Designing and combining mid-air interaction techniques in large display environments*. Ph.D dissertation, Université Paris Sud - XI, Dec. 2012.
6. Nancel, M., Chapuis, O., Pietriga, E., Yang, X.-D., Irani, P. P., and Beaudouin-Lafon, M. High-precision pointing on large wall displays using small handheld devices. In *Proc. CHI'13*, ACM (2013), 831–840.
7. Vogel, D., and Balakrishnan, R. Distant freehand pointing and clicking on very large, high resolution displays. In *Proc. UIST'05*, ACM (2005), 33–42.
8. Woźniak, P., Fjeld, M., and Zhao, S. Limiting trial and error: Introducing a systematic approach to designing clutching. In *Proc. Chinese CHI*, ACM (2014), 35–39.