Limiting Trial and Error: Introducing a Systematic Approach to Designing Clutching

Paweł Woźniak
t2i interaction laboratory
Chalmers University of Technology
pawelw@chalmers.se

Morten Fjeld
t2i interaction laboratory
Chalmers University of Technology
fjeld@chalmers.se

Shengdong Zhao
NUS-HCI Lab
National University of Singapore
zhaosd@comp.nus.edu.sg

ABSTRACT

When designing new interfaces we are likely to be confronted with the problem of *clutching*—creating a way to switch between engaged and disengaged input modes. Using constraint analysis we propose a systematic approach for designing clutching. Based upon Buxton's three-state model, we introduce a design procedure for determining optimal clutching mechanisms. Using this procedure, designers of future interfaces can benefit from reduced time for trial-and-error in clutching design, since key candidates for clutching mechanisms can now be quickly identified. Through a case study of clutching for pen tilt input, we show how our method can be applied to a concrete design task.

Author Keywords

Input channels; design procedure; clutching;

ACM Classification Keywords

H.5.m. Information interfaces and presentation (e.g., HCI): Miscellaneous.

INTRODUCTION

Most graphical user interfaces featuring *continuous input* [1] will likely require a mechanism for *clutching*—a way for the user to turn the tracking mode of the device on and off. Lifting the mouse and going back over the surface of the mouse pad to move the cursor over larger distances is a form of clutching with which most users are familiar. Clutching is a particular case of mode switching, a fundamental operation for many established devices. In touch input, clutching is inherent, as lifting the finger from the touch surface stops input. As most input methods are constrained by their form factor and the extent of the human body, clutching can be used to extend input range, permitting the reuse of physical space for continuous input.

So far, most researchers have approached the design of clutching experimentally, testing a number of alternatives and then comparing performance, leading to heuristic choices usually not visibly accompanied by a clear design rationale. We believe that a clear design strategy to efficiently reach clutching solutions is useful, as more input methods will emerge in the future. Hence, we propose a systematic approach that will identify, filter, and detect a number of candidates for a particular clutching method, thereby reducing the need for experimental evaluation and help identify bad choices earlier in the design process.

This note elaborates and suggests two notions: 1) a general design procedure for clutching solutions and 2) an underlying theoretical analysis. To our knowledge, this note is the first attempt to systematically study clutching as a separate interface design issue.

RELATED WORK

Though clutching is as old as relative position control, the term "clutch" was introduced much later [5]. Researchers have also used alternative terms such as "ratcheting recalibration mechanism" [2] and "re-clutching" [3,10]. Despite a traceable history of the concept, our literature survey revealed clutching has often been designed without an underlying design motivation [7] and method choices have not been well motivated.

A number of more recent examples illustrate how recent interfaces may benefit from a clutching mechanism, but where implementation is still decided using trial and error These examples include Wiimote pointing by Pelling et al. [6], where an elaborate double-cursor setup was used to enable gesture-based clutching. In ArcheoTUI [7], pedals and buttons were compared as clutching solutions. Similarly, past experimental inquiries into mode switching (e.g. in wall-sized displays [9] or indirect touch systems [8]) have focused on choosing a number of arbitrary possibilities and then performing comparative experiments. This is why we see the need to approach clutching in a systematic way.

Chinese CHI '14, April 26 - 27 2014, Toronto, ON, Canada

ACM 978-1-4503-2876-0/14/04…$15.00.
http://dx.doi.org/10.1145/2592235.2592241

DESIGN CONSIDERATIONS FOR CLUTCHING

Firstly, clutching only concerns a subset of user interfaces—tasks featuring continuous position control [1]. Limiting the scope of our inquiry to continuous input is quite self-explanatory—binary (non-continuous) tasks can be performed more effectively without movement [4]. The other constraints illustrate the raison d'être of clutching—position control is preferred to rate control, offering a high degree of perceived usability [11]. However, this comes at the price of designing ways for users to reuse physical space and split movements into sub-movements. To enable this, the system should support repeating the same movements in the same space. Think of scrolling a long list with a mouse—you are sure to traverse your mouse pad several times before the end of the list. For the purposes of this note, we define clutching as: "The momentary recalibration of the device required to extend input range beyond the physical input space".

In order to establish a way to think about and discuss clutching, we propose a four-stage diagram explaining the phenomenon (Figure 1). This can serve as an aid in the proposed design procedure, marking out the key elements of clutching. The state diagram we propose is based on a state transition diagram heavily inspired by Buxton's three states [1]—which is general and powerful enough to be applied to nearly all kinds of interfaces.

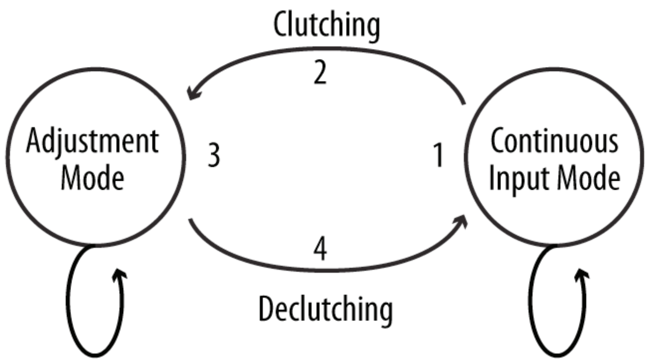

Figure 1: An adapted version of Buxton's three state model for continous input interfaces offers a state-transition diagram providing a better understanding of clutching where (1) and (3) are states, (2) and (4) are transitions.

The proposed approach helps us to conceptualise clutching tasks as sequences of operations, and to understand the relationships between those operations. It is essential to realise that input tasks consist of all four stages in a sequence (1, 2, 3, and 4). Furthermore, there is a complementary relationship between the states: the design of stage 1 always affects the design of stage 3, and the design of stage 2 always affects the design of stage 4. Operations in these pairs must always revert the effects of their counterparts and this affects their design by requiring reversible metaphors.

Clutching design needs to consider the following three constraints:

C1. Human constraints: the limitations on our body's physical movements and of our perceptual capabilities must be taken into account (e.g. reach, desk space, and fatigue).

C2. Device constraints: the limited technical capability of any given device plays a crucial role in the design (e.g. wire length, sensor sensitivity).

C3. Environmental constraints: limitations can also be imposed by the context of use (e.g. table surface size, lighting conditions, or social context).

Constraints are likely to vary over time and space (e.g. the context of use for mobile devices may change often), and a well-designed clutching mechanism must account for these changes.

In addition, two principles derived from practice are essential for effective clutching design:

P1. Engaging the clutch (stage 2) should not affect values in stage 1. That is, the clutching operation should be possible to perform at all times without affecting the input.

P2. The clutch should be seamlessly integrated in the input task to provide a minimal number of accidental triggers.

PROPOSED DESIGN PROCEDURE

We use the analysis presented above to introduce the proposed design procedure, using three steps of identifying, filtering, and detecting to distill viable candidates for a clutching solution.

Identify candidates: List the capabilities of the system and all the input possibilities it may offer. Include input channels (i.e. variables generated by the input devices) the system may support and the types of input they may generate. It may be beneficial to allocate other input channels to interaction tasks or to "reuse" the same input channel by dividing it into zones (spatially and temporally). Figure 2 provides some examples.

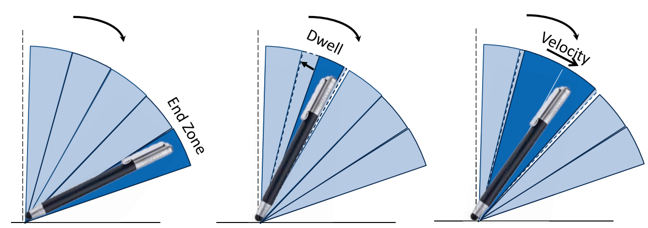

Figure 2. Designing clutch technique for pen tilt. Three alternative examples of dividing an input channel into zones to create clutching cadidates. Here, we use spatial, temporal and spatial-temporal zones (left to right).

Filter: Investigate clutching constraints (C1-C3) and principles (P1-P2) as listed above. Filter out any input channels highly dependent on either changing constraints or requiring excessive movement or cognitive capability.

Detect conflicts (T1-T3): Compare the list of input parameters for conflicts with the application task. Eliminate input parameters having considerable effect on steps 1 and 3. As candidates obtained by dividing channels into zones reuse input channels, they do not create an immediate conflict. We propose three approaches to detect conflicts depending on the type and complexity of the input task:

T1. Simple tasks. When a task uses only a single input channel, analyze possible clutching candidates by juxtaposing the used input channel with all other available input channels. Determine if there is any conflict between the input task and the possible clutching channel and estimate the duration of the conflict. A measure of the conflict is the product of the two. Choose the possibilities with the lowest scores for experimental evaluation. Example: Consider controlling the cursor with a mouse on the XY plane. If considered alone (i.e. buttons do not perform any functions) pressing a button can be a clutch for the input task. The user may reposition the mouse when the left or right button is pressed. On the other hand, we cannot use rotating the mouse (Rz) as a clutch, as this may affect the value of Position (X, Y)—there is a conflict.

T2. Compound tasks. Whenever an input task uses multiple output channels, possible candidates for clutching mechanisms must be examined for conflicts with all the input channels constituting a task. Perform the procedure described in *T1* for each of the input channels for the compound tasks. Sum conflict measures for each candidate. Experiment with a number of the lowest-scoring candidates. Example: Think of drawing using a mouse where a button press engages the virtual ink. This is a compound task consisting of the Left Button Press and Position (X, Y). We can no longer use the Left Button Press as a clutch for Position (X, Y).

T3. Sequential tasks. When a task is part of a sequence of tasks required to achieve a given outcome, one may need to take the sequence into account. Begin by performing the analysis required by *T1* and *T2*. Next, eliminate those input channel candidates directly preceding or following the sequential task. A state transition diagram may help identify the conflicting input channels. Example: Consider a mouse-based task where the user drags an icon (with Left Button Pressed) and then brings up a context menu with the Right Mouse Button. Here, Right Button Press is conflicted with Position (X, Y) and a different clutching solution must be found.

In the following case study section we will use a practical example to concretely show how one can apply the proposed procedure.

CASE STUDY

We will apply the procedure to design the optimal clutching mechanism for various types of pen input on a Wacom Intuos4 PTK-440 309x208mm tablet with the Wacom Pen. The device is presented in Figure 3. Clutching may be needed for many (task and input channel) combinations, for example the scenarios of panning a map by moving the pen tip on the horizontal surface, changing the width of the drawing line using pen tilt, select a song from a list using pen pressure. To identify the ideal clutching technique for each of the above scenarios, we follow the following steps.

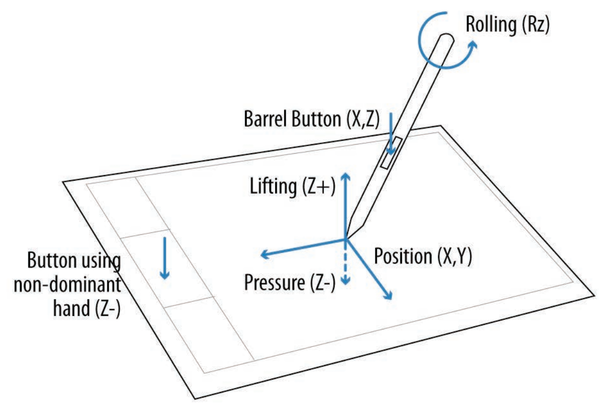

Figure 3. The tablet used in the case study, presented with a set of candidate input channels.

In the first step we *identify* available input channels which will we will study in more detail (Figure 3). Next, we eliminate potential clutching candidates dismissed by the three constraints. Rolling (Rz*) is eliminated by the *human constraint* (C1), as repeated rolling of the pen positions the hand awkwardly, fatiguing the user. We also notice how the tilt input channel may be split into zones as shown in Figure 2. We *filter* out the velocity zone-based clutch as it may cause fatigue (C1) and may be difficult to track (C2). To illustrate how to apply our procedure we study the input task at three levels of abstraction.

Consider pen tilting as an input channel. This action is performed in two directions, X and Z. We investigate the possibilities for a clutching mechanism for an input task consisting of tilting the pen. In *T1* we *detect* potential conflicts and their duration. Position (X, Z) is clearly conflicted with Tilting as both use movement in the X direction. Pressure (Z-) and Lifting (Z+) can easily affect the logical value of Tilting. Pressing the Barrel Button may require adjusting pen tilt (e.g. when the pen is close to the tablet surface), so these candidates are *filtered* out.

The analysis yields two possible candidates: Lifting (Z+) and Secondary Button. Additionally, we still consider two zone-based candidates.

To illustrate the application of *T2*, we now consider a different task—drawing a line of varying width. This input task uses two input channels: pressure (Z-) and position (X, Y). We repeat the *T1* analysis for both input channels. It is apparent that position (X, Y) cannot be a clutch for pressure (Z-), as it would conflict with the compound task despite the lack of conflict with pressure (Z-)—declutching pressure would result in drawing undesired lines. Analogously, while Lifting (Z+) is not conflicting with Position (X, Y), it poses a conflict with pressure (Z-). As a consequence, a button-based solution may be required.

Back to the pen tilt example. In the final step we look for input channels, which may be part of actions directly preceding or following the input tasks. We do not *filter* out any new input channels, because the previous steps have already eliminated many candidates. However, had we not previously filtered out the Barrel Button Press option, we could have done it here, because the button may be used to call up contextual menus. Consequently, in a task where the user was to call a menu after tilt control, a conflict would arise.

We have shown a method of how a list of potential clutching candidates may be narrowed. An overview of the process is presented in Table 1.

Input channel	Reason for filtering out / Candidate
~~Position (X, Y)~~	Conflict: Moving the pen in the X direction affects tilt
~~Lifting (Z+)~~	Conflict: Lifting the pen invalidates tilt
~~Pressure (Z-)~~	Conflict: Pressing the button may affect tilt
~~Barrel Button (X, Z)~~	Conflict: Pressing the button may affect tilt
Secondary button	**Candidate**
~~Rolling (Rz*)~~	C1—causes fatigue
Zone – End zone	**Candidate**
Zone – Dwell	**Candidate**
~~Zone – Velocity~~	C1—causes fatigue

Table 1. Design procedure for the pen tilt case: candidate input channel (left) and the result of the design process for each candidate (right).

The results of the design procedure are now to be used to conduct an empirical study that will provide final insights as to the suitability of each candidate. The baseline "no clutching" condition may be used to increase the validity of the study. As in most input interface studies, one should measure accuracy and speed and decide on the optimal mechanism using these parameters.

CONCLUSION AND FUTURE WORK

We have introduced a design procedure and a number of considerations that help in reducing the number of possible candidates for clutching solutions. We proposed a three-step procedure (identify, filter, detect) to help distill viable candidates for a clutching solution. We presented a way to conceptualize clutching that helps the understanding of the proposed procedure based on Buxton's 3 state model. The method supports empirical studies by limiting the number of trials required to achieve insight on the proper clutching technique thereby showing how it can be applied. The procedure was applied to a design task, investigating clutching possibilities for pen tilt.

We would like to see our procedure applied to new interfaces. We believe that it may significantly reduce the resources needed in designing clutching solutions, as well as limiting trial-and-error investigations in future inquiries.

REFERENCES

1. Buxton, W.A.S. A three-state model of graphical input. *Human-Computer Interaction 90*, (1990), 449–456.
2. Hinckley, K., Pausch, R., Goble, J.C., and Kassell, N.F. A survey of design issues in spatial input. *Proc. UIST '94 94p*, 2 (1994), 213–222.
3. Hinckley, K., Sinclair, M., Hanson, E., Szeliski, R., and Conway, M. The VideoMouse: a camera-based multi-degree-of-freedom input device. *Proc. UIST '99*, ACM Press (1999), 103–112.
4. Isenberg, T. and Hancock, M. Gestures vs . Postures : 'Gestural' Touch Interaction in 3D Environments. *3DCHI Workshop*, (2012).
5. Jacob, R.J., Leggett, J.J., Myers, B.A., and Pausch, R. An Agenda for Human-Computer Interaction Research: Interaction Styles and Input/Output Devices. *Behaviour and Information Technology*, (1993).
6. Pelling, C., Sko, T., and Gardner, H.J. Be careful how you point that thing: Wiimote aiming for large displays. *Proc. OZCHI '09*, ACM Press (2009), 397.
7. Reuter, P., Riviere, G., Couture, N., Mahut, S., and Espinasse, L. ArcheoTUI - Driving virtual reassemblies with tangible 3D interaction. *J. Comput. Cult. Herit. 3*, 2 (2010), 4:1–4:13.
8. Voelker, S., Wacharamanotham, C., and Borchers, J. An evaluation of state switching methods for indirect touch systems. *Proc. CHI '13*, ACM Press (2013), 745.
9. Vogel, D. and Balakrishnan, R. Distant freehand pointing and clicking on very large, high resolution displays. *Proc. UIST '05*, ACM Press (2005), 33.
10. Zhai, S., Milgram, P., and Buxton, W. The influence of muscle groups on performance of multiple degree-of-freedom input. *Proc. CHI 96*, (1996), 308–315.
11. Zhai, S. Human performance in six degree of freedom input control. (1995).

ACKNOWLEDGEMENTS

This work was supported by the EU FP7 People Programme (Marie Curie Actions) under REA Grant Agreement 290227 and the Swedish Foundation for International Cooperation in Research and Higher Education (STINT, grant 2013-019).