Flick-and-Brake: Finger Control over Inertial/Sustained Scroll Motion

Mathias Baglioni, Sylvain Malacria, Eric Lecolinet, Yves Guiard

HAL Id: hal-00705430
https://imt.hal.science/hal-00705430

Submitted on 7 Jun 2012

Flick-and-Brake: Finger Control over Inertial/Sustained Scroll Motion

Mathias Baglioni

Télécom ParisTech – LTCI-CNRS,

Alactel Lucent Bell Labs

46 rue Barrault

Paris, 75013 FRANCE

mathias.baglioni@enst.fr

Sylvain Malacria

Télécom ParisTech – LTCI-CNRS

46 rue Barrault

Paris, 75013 FRANCE

sylvain.malacria@enst.fr

Eric Lecolinet

Télécom ParisTech – LTCI-CNRS

46 rue Barrault

Paris, 75013 FRANCE

eric.lecolinet@enst.fr

Yves Guiard

Télécom ParisTech – LTCI-CNRS

46 rue Barrault

Paris, 75013 FRANCE

yves.guiard@enst.fr

Abstract

We present two variants of Flick-and-Brake, a technique that allows users to not only trigger motion by touch-screen flicking but also to subsequently modulate scrolling speed by varying pressure of a stationary finger. These techniques, which further exploit the metaphor of a massive wheel, provide the user with online friction control. We describe a finite-state machine that models a variety of flicking interaction styles, with or without pressure control. We report the results of a preliminary user study that shows that for medium to long distance scrolling the Flick-and-Brake techniques require less gestural activity than does standard flicking. One of the two variants of the technique is faster, but no less accurate, than state-of-the-art flicking. Users reported they preferred Flick-and-Brake over the standard flick and judged it more efficient. We indicate some pending issues raised by the results of this preliminary investigation.

Keywords

Flick and Brake, flicking, mobile devices, touch-screen interaction, pressure, pseudo-pressure, physical metaphors, kinetic scrolling.

ACM Classification Keywords

H5.2 [Information interfaces and presentation]: User Interfaces. - Graphical user interfaces.
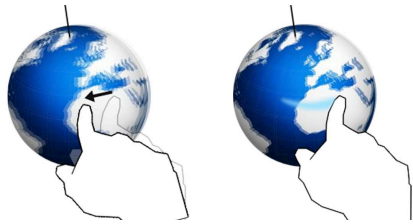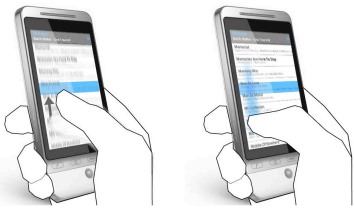
Figure 1: Real world Flick-and-Brake

Figure 2: Virtual Flick-and-Brake

General Terms
Design, Human Factors

Introduction
If touch-sensitive surfaces bring new interaction capabilities, they suffer some limitations of their own. For instance, tabletops and smartphones have usually no equivalent of the mouse buttons and wheel. These limitations are overcome through new interaction techniques. For instance, swipe, flick or pinch gestures have been proposed as substitutes. But the definition of these gestures is often floating. For example a flick is literally a quick throwing gesture that serves to impart motion, yet it is often used to denote the sequence of a flick gesture followed by automatic scroll with simulated inertia and friction. The term "flick" first appeared in Penpoint [5] to designate a straight pen gesture that could serve as the command to switch to the previous/next page. Then flick gestures served in various contexts of use: for instance [13] used them in association to the throwing-object metaphor on a tabletop, and [12] combined them with automatic friction-decelerated motion, completing the physical metaphor.

For the throwing metaphor to be ideally vivid, we need to simulate both inertia and energy dissipation via friction. One difficulty with the current application of the metaphor to scrolling (e.g., on the iPhone) is that if inertia is indeed simulated, the simulation of finger-added friction is overly simplified to the level of an all-or-none variable: touching the screen during scrolling motion stops it instantly.

The real world does not work like that. For example, when manipulating an earth globe (**Figure 1**) one can not only impart rotary motion to the globe with an impulse, but also, subsequently, regulate the globe's deceleration by applying on its surface subtle amounts of finger pressure. In this paper we propose to complete flicking with finger pseudo-pressure, so as to provide users with control over both the initial acceleration and the final deceleration of scrolling motion. With *Flick-and-Brake*, the technique we propose, the mechanical metaphor is more complete and, we will argue, more vivid.

We will describe two variants of the Flick-and-Brake technique, and then a generic finite-state machine that models Flick-and-Brake and flicking interaction in general. We will report the results of a preliminary experiment which compared Flick-and-Brake with state-of-the-art flicking for item acquisition in 1-D lists. We will then discuss some future work we plan to conduct based on these first results.

Related Work
A few techniques using pressure on mobile devices have already been described in the literature. For example pressure served as an alternative to multi-tap for text entry in [9] with different pressure levels for different letters on a key, and for the virtual keyboard in [4], where pressure allows the uppercase switch. The Pressure Widget [11] proposed several designs of widgets that exploit finger pressure, like for instance the pressure-based marking menu; this paper also reported an investigation of the number of pressure levels that users are able to control with a finger. GraspZoom [10] proposed to use pressure on a touchscreen for enabling one-handed continuous scrolling: when a pressure threshold is exceeded in the upper (resp. lower) part of the screen, the contents start to scroll upwards (resp. downwards), the direction being adjustable via tiny thumb gestures.

Many studies have focused on novel techniques for interacting with objects on touch-screens, including [7], which introduced one of the first applications of the object-throwing metaphor. Inertia and friction simulation was reported in [8] for panning and flicking
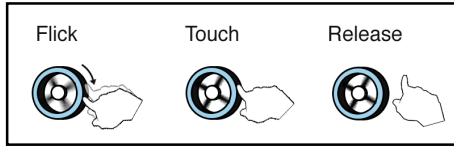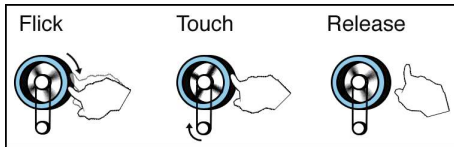
Figure 3: Inertial Flick-and-Brake.

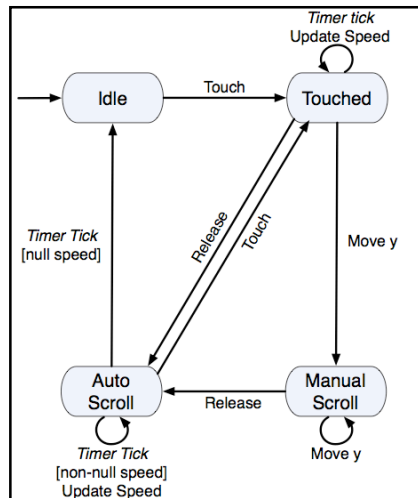**Figure 4**: Powered Flick-and-Brake.

Figure 5: Finite-state machine.

to give physical realism to the translation of sheets on interactive tabletops. Superflick [12] enhanced object flicking with special techniques to improve efficiency and precision. Multi-flick [1] introduced a new design space and evaluated techniques for list scrolling based on repetitive flick actions.

Much of the literature has focused on the issue of the gestural initiation of scrolling motion. Little research has been reported inquiring, as in [12], into the gestural techniques that might help control the progression and, most importantly, the termination of scrolling motion. Yet the ends of our actions are generally their most critical part—for example, it is the act of reaching a certain item that motivates any flick-triggered scroll. The work presented below aims at paving the way for improvements of the middle and final phases of scrolling actions.

Flick-and-Brake

Common implementations of flicking involve kinetic scrolling: once a graphical object, for instance a list, has been metaphorically thrown by the user, the system simulates inertia (the object continues to move) as well as friction (the speed progressively decreases down to rest). The deceleration is thus automatically controlled by the system. Flick-and-Brake, in contrast, lets the user control this deceleration by pressing a finger on the screen: the stronger the pressure, the stronger the deceleration, as in the real world (Figure 1). In fact our algorithm involves no friction other than that applied by the user's finger, meaning that the user has exclusive control over the deceleration (in the absence of a finger contact, the motion will continue for ever). We now describe our two variants of the Flick-and-Brake technique.

Variant 1: Inertial Motion Metaphor
This variant uses the metaphor of a frictionless wheel (**Figure 3**) whose rotary motion, once triggered by a

flick, would continue forever if the user did not operate the brakes. Importantly, the control is not all-or-none: the user can finely modulate finger pressure to modulate the rate of (negative) change of speed (second-order control mode). With a constant pressure level, the speed will decrease at a constant rate down to rest.

Variant 2: Powered Motion Metaphor
The other variant relies on a frictionless wheel metaphor too, but with an engine that will oppose finger friction (**Figure 4**). Finger pressure will still decrease the speed created by the initial flick, but control here turns to the first-order kind: as long as the finger is in contact with the wheel, scrolling speed will be proportional to finger pressure, pressure saturation stopping the motion. Removing the finger will bring the speed to its initial value, unless speed has zeroed out

Generic Finite State Machine
In the Finite State Machine (FSM) of **Figure 5** finger pressure is a user-controlled variable. The model identifies four states, Idle, Touched, Auto-Scroll, and Manual-Scroll. All modeled techniques, either the standard flick or the Flick-and-Brake technique need the four states but they use different stop cases. Standard flicking techniques stop the scrolling instantly upon screen contact, pressure being binary. In contrast the Flick-and-Brake (F&B) techniques consider the pressure factor as a continuous input variable in the 0-1 range, under user control.

To illustrate, let us describe standard flicking with a list. At the start (Idle state), scrolling speed is zero. As the user touches the screen (Touched state), she starts scrolling the list by moving her finger quickly (Manual-Scroll state) then releases her finger (Auto-Scroll state). The list keeps on scrolling, its speed decreasing down to zero due to system friction (back to the Idle state).

The *Auto-Scroll* function updates the scrolling speed when the finger is off the screen. Usually, this function affects the current speed by a fixed *k* parameter corresponding to system friction, with $k \in [0;1]$ (for k=1 speed remains constant, for k=0 the motion stops instantly). The FSM goes to the Idle state if the speed zeroes out, and to the Touched state if the user touches the screen.

The *Touched* function updates the speed when the user touches the screen and keeps her finger stationary. For standard flicking (SF) techniques, the binary pressure (*p*) factor sets the scrolling speed to zero at the first timer tick. With the Flick-and-Brake (F&B) techniques, *p* varies continuously between 0 and 1 so that the stronger the press, the stronger the scrolling deceleration. Noting V_n the speed at sample n,

$$V_{n+1} = V_n*(1-p), p \in \{0,1\} \quad \text{Standard flick} \quad (1)$$
$$V_{n+1} = V_n*(1-p), p \in [0;1] \quad \text{Inertial F\&B} \quad (2)$$
$$V_{n+1} = V_0*(1-p), p \in [0;1] \quad \text{Powered F\&B} \quad (3)$$

Common mobile devices have no hardware for measuring the actual finger pressure on their screen. It must thus be estimated from *pseudo-pressure* (*pp*), a value returned by the operating system (Android 2.1 in our case), which relies on the measurement of the contact area between the finger and the touch-screen [3]. This area depending on finger size and shape, *pp* is user-dependent. Calibration was obtained by simply subtracting the initial pp_0 value (obtained when the user starts pressing the screen) from the current *pp* value. A pilot study showed that the *pp* value changed in a non-linear fashion, unsurprisingly as *pp* depends on the surface area of the contact. We used a quadratic function to estimate *p* from *pp*, so that the *p* of Equations 2 and 3 was computed as

$$p = (pp - pp_0)^2.$$

Figure 6: Experiment software screen capture.

Experiment

The main goal of this experiment was to evaluate the efficiency of user-controlled deceleration, compared with system-controlled deceleration, in users asked to reach a specified song in a long alphabetically-ordered list. We deliberately chose a task that is common on mobile devices (creating a playlist on a music player). We were chiefly interested in performance speed and accuracy, but we also wanted to learn about the users' utilization of the techniques, in particular in terms of the number of flicks (an open variable in this sort of task), as well as about their subjective ratings of the competing techniques.

Method

APPARATUS

We used a HTC Hero Smartphone under Android 2.1 with a 3.5' capacitive touch screen and a 320x480px display with pseudo-pressure detection. Items being 50px high, 8.5 of them were visible in the viewport.

TASK AND PROCEDURE

The name of each target item was shown at a fixed position at the top of the screen. The trial was validated when at least half of the specified item appeared, stationary and finger off, within a fixed 50px-high rectangle located in the middle of the screen (**Figure 6**). The text at the top of the screen then changed to indicate the next target, at a distance of 25, 75, 150, or 250 items above or below. Note that because no target was displayed before the previous target had been successively selected, in this task the error rate (selecting a wrong item) was a forced 0%.

The distance and the direction were randomized, with each combination presented exactly once (8 targets per block). To avoid learning effects, each block used a unique song list with different song names.
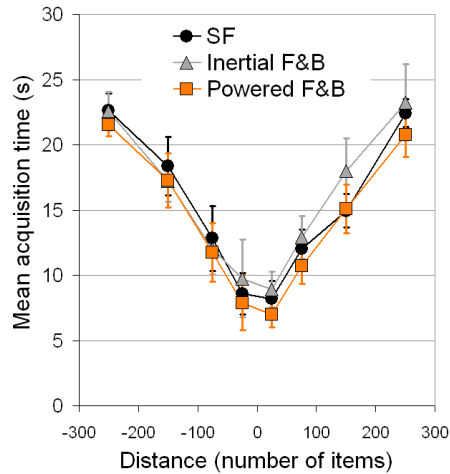
Figure 7: Target-acquisition time (error bars show 95% confidence intervals).
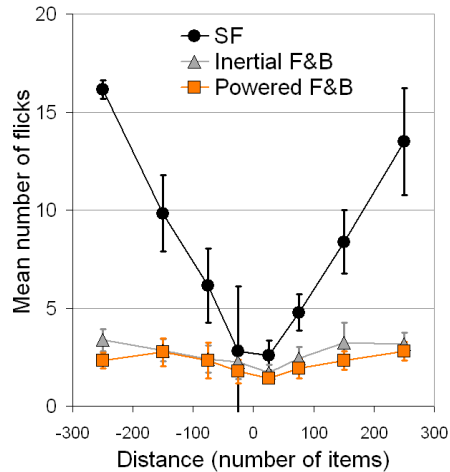

Figure 8: Number of flicks (error bars show 95% confidence intervals).

We investigated three techniques, the standard flick (SF) as well as the inertial and the powered Flick-and-Brake. For each technique, the maximum scrolling speed was limited to 2,100 px/s, in keeping with the suggestion of [2]. Order of presentation of the techniques was balanced using a Latin-square design.

Twelve adult participants (four female) completed two training trials and four test trials for each technique. The experimental design was a fully within-participant 3 techniques × 4 distances × 2 directions.

After the experiment the participants were asked to rank the techniques by order of preference and to evaluate each with four criteria (how fast, precise, pleasant, and simple), using a 5-point Likert scale.

Results
TARGET ACQUISITION TIME (FIGURE 7).
For performance speed the powered version of F&B surpassed both SF (Student's t-test: $t_{11}=1.84$, $p=.046$) and the inertial version of F&B ($t_{11}=2.99$, $p=.006$), which failed to outperform SF ($t_{11}=1.18$, $p=.131$). These are consistent but small effects, perhaps because of our 2,100px/s speed limit.

NUMBER OF INDIVIDUAL FLICKS PER TARGET ACQUISITION (FIGURE 8)
The participants made far fewer flicks per target acquisition with either of the two F&B techniques than with SF ($F_{1,11}=72.61$, $p<.001$). The figure shows that while the number of flicks increased about linearly with target distance in either direction, the slope of this relation was considerably shallower with the two F&B techniques than with SF: hence the farther the target, the larger the saving of finger activity with F&B, as reflected by a strong interaction between technique and target distance (both F&B techniques being pooled, $F_{3,33}=54.73$, $p<.001$).

SUBJECTIVE PREFERENCES (FIGURE 9)
The participants reported they felt faster when using F&B, whether powered (Wilcoxon test, $p<.0003$) or inertial ($p<.0005$), than when using SF. F&B was judged generally more pleasant than SF ($\chi^2=6.983$, $p<.0305$), but this preference was especially clear between powered F&B and SF ($p<.0176$). Judging by the verbal explanations we received, these results seem to mostly reflect the fact that SF requires of users a problematic number of flicking gestures when target distance becomes really large.
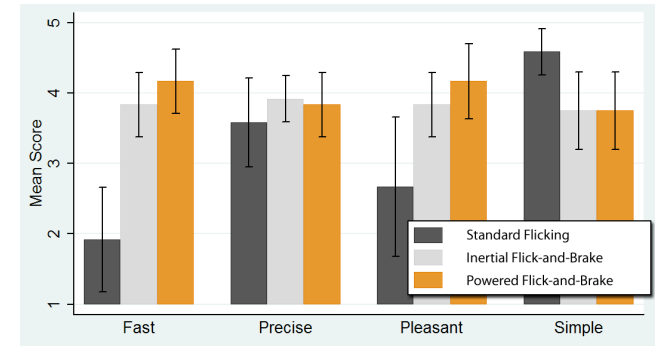

Figure 9: Subjective preferences

During the experiment the participants enjoyed the braking-control facility available in the F&B techniques. As shown in Table 1, both variants of the Flick-and-Brake technique were well received, the participants always ranking one of them first.

	1st	2nd	3rd	Mean
SF	0	1	11	2.92
Inertial F&B	4	7	1	1.75
Powered F&B	8	4	0	1.33

Table 1: Technique ranking.

Conclusion and future work

We have presented two variants of Flick-and-Brake, a novel technique that completes the physical metaphor that underlies flicking interaction. F&B allows users to continuously control the scrolling speed by pressing the finger on the screen, down to final target acquisition. We also presented a finite state machine that describes all flicking techniques. Our data show that users make fewer flicks to cover long distances, and that they perform and feel significantly faster with user-controlled, rather than system-controlled friction. And, importantly, F&B was well accepted by participants.

Among the pending issues raised by this research, we plan in our future work to focus on the efficiency of Flick-and-Brake for 2D-scrolling interaction and on its usability in real-life contexts. We also plan to try to optimize the form of the visual feedback for friction and the transfer function between pressure and friction.

Acknowledgments

References

[1]   Aliakseyeu, D., Irani, P., Lucero, A., and Subramanian, S. (2008). Multi-flick: an evaluation of flick-based scrolling techniques for pen interfaces. Proc. CHI '08. ACM, 1689-1698.

[2]   Andersen, T. H. (2005). A simple movement time model for scrolling. Proc. CHI '05, ACM, 1180-1183.

[3]   Blasko, G. and Feiner, S. (2004). Single-handed interaction techniques for multiple pressure-sensitive strips. Proc. CHI EA '04, ACM, 1461- 1464.

[4]   Brewster, S. A. and Hughes, M. (2009). Pressure-based text entry for mobile devices. Proc MobileHCI '09. ACM, 1-4.

[5]   Carr, R.K, and Shafer, D. (1991). The Power of Penpoint (1st ed.). Addison-Wesley Longman Publishing Co., Inc..

[6]   Cechanowicz, J., Irani, P., and Subramanian, S. (2007). Augmenting the mouse with pressure sensitive input. Proc. CHI '07. ACM, 1385-1394.

[7]   Geißler, J. (1998) Shuffle, throw or take it! Working efficiently with an interactive wall. Proc. CHI EA '98. ACM, 265-266.

[8]   Kruger, R., Carpendale, S., Scott, S., Tang, A. (2005). Fluid integration of rotation and translation. Proc. CHI 2005, ACM, 601-610.

[9]   McCallum, D. C., Mak, E., Irani, P., and Subramanian, S. (2009). PressureText: pressure input for mobile phone text entry. In Proc. CHI EA '09. ACM, 4519-4524.

[10] Miyaki, T. and Rekimoto, J. (2009). GraspZoom: zooming and scrolling control model for single-handed mobile interaction. In Proc. MobileHCI '09. ACM, 1-4.

[11] Ramos, G., Boulos, M., and Balakrishnan, R. (2004). Pressure widgets. Proc. CHI '04. ACM, 487-494.

[12] Reetz, A., Gutwin, C., Stach, T., Nacenta, M., and Subramanian, S. (2006). Superflick: a natural and efficient technique for long-distance object placement on digital tables. Proc. Graphics. ACM, 163-170.

[13] Mike Wu and Ravin Balakrishnan. 2003. Multi-finger and whole hand gestural interaction techniques for multi-user tabletop displays. Proc. UIST '03. ACM, 193-202.