# LibPad: Analyzing the Impact of OS-Specific Transfer Functions on Touchpad Pointing Performance


Been An
Chung-Ang university
Seoul, Republic of Korea
dksqls08@cau.ac.kr

Hojoon Lee
Sungkyunkwan University
Suwon, Republic of Korea
hojoon.lee@skku.edu

Eunji Park*
Chung-Ang University
Seoul, Republic of Korea
eunjipark@cau.ac.kr



## Abstract

Touchpads are essential input devices commonly integrated into laptops. Some users have pointed out that touchpad usability may vary depending on operating system (OS), with MacBook trackpads often cited as offering superior user experiences. Despite this strong user interest, prior research has primarily focused on hardware factors (e.g., surface friction, touchpad size), while software-level factors remain underexplored. To address this gap, we present `LibPad`, a toolkit for applying OS-specific transfer functions to different hardware. We investigate how transfer function differences across OS affect pointing performance across laptop hardware. Our findings reveal significant performance differences both across hardware under identical transfer functions and within same hardware under different configurations. These results highlight the importance of software-level considerations in enhancing touchpad usability.



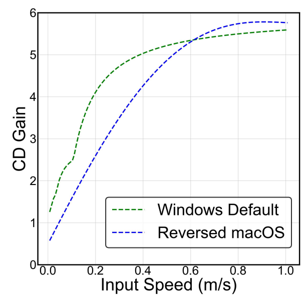
## CCS Concepts

• **Human-centered computing** → *Pointing devices.*

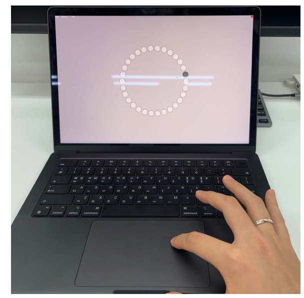
## Keywords

Transfer function, Touchpads, Pointing Performance, CD gain

**ACM Reference Format:**
Been An, Hojoon Lee, and Eunji Park. 2025. LibPad: Analyzing the Impact of OS-Specific Transfer Functions on Touchpad Pointing Performance. In *The 38th Annual ACM Symposium on User Interface Software and Technology (UIST Adjunct '25), September 28–October 01, 2025, Busan, Republic of Korea.* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3746058.3758391


**(a) Transfer function of Windows default and macOS**



**(b) Pointing task program with touchpad**

**Figure 1: Comparison of transfer functions and task program**

## 1 Introduction

Touchpads are nowadays de facto standard input devices for mobile computers. Interestingly, opinion on their user experience is often polarized. Some users adamantly refuse to use a mouse along with their laptops, while external touchpads for desktop use are also common. In online communities, users express conflicted opinions on touchpad usability and it is notable that the MacBook touchpad users often swear by superior user experiences of Apple's Trackpad [6, 17–19]. However, as to why Apple's trackpad implementation is generally preferred is yet to be scientifically understood.

Besides, while input devices have been a key element in Human-Computer Interaction (HCI) research, underlying mechanisms and usability optimization of touchpads have been relatively underexplored compared to those of mice [2, 9, 14, 15]. For instance, *transfer functions* that play a pivotal role in pointing device usability are largely unexplored for touchpads. The effect of transfer functions in mouse usability has been studied through several works [3, 8]. A dedicated study is necessary to understand and be able to model transfer functions of touchpads. This is because touchpads must optimize their transfer functions within the bounded surface area, unlike mice. Also, we found that touchpad hardware reports absolute coordinates, unlike mice that measure relative motion counts [5, 7]. Most existing studies on touchpads have focused on hardware aspects, such as surface friction [13, 16] and physical size [1]. This gap in research motivated us to conduct a study to understand the role of transfer functions in touchpads, and reveal how OS-specific implementations and the underlying hardware together form transfer functions. As such, we address following research questions:

- **RQ1**: How are the transfer functions implemented in different OSes, and how do they affect touchpad pointing performance?
- **RQ2**: How does the different laptop touchpad hardware influence pointing performance when using identical transfer functions?

## 2 Implementation

### 2.1 Reverse engineering macOS transfer function

We reverse engineered the macOS transfer function through a set of tools and techniques. This is because the transfer function of macOS touchpad is not publicly documented unlike that of Windows [12]. Using a custom tool built on a publicly undocumented and private macOS framework (i.e., MultitouchSupport), we recorded each touch event with its timestamp, coordinates, and touch state (e.g., contact, clicking). We then implemented a pointing task program in Unity based on `FittsStudy` [21], that can log cursor positions


*Corresponding author




*UIST Adjunct '25, Busan, Republic of Korea*



ACM ISBN 979-8-4007-2036-9/25/09
https://doi.org/10.1145/3746058.3758391

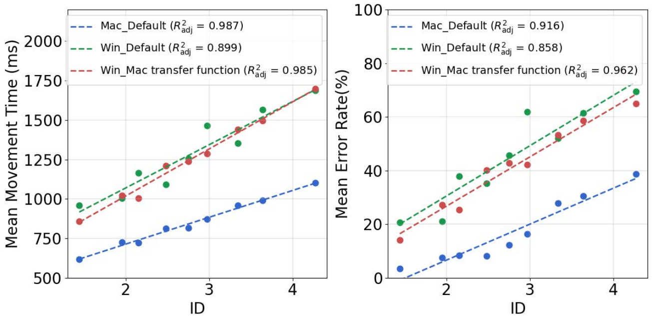



**Figure 2: Regression results of Movement Time and Error Rate**

at millisecond resolution. These cursor logs were aligned with the corresponding touchpad inputs at same timestamp, enabling us to calculate displacements in both spaces (i.e., touchpad input, cursor positions) over time. Based on these data, we computed Control-Display gain (C-D gain) as follows:

$$\text{Control-Display Gain} = \frac{\sqrt{(x_d - x_{d-1})^2 + (y_d - y_{d-1})^2}}{\sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2} \cdot \frac{\sqrt{W^2+H^2}}{C}}$$

where $(x_i, y_i)$ are touchpad inputs, $(x_d, y_d)$ are cursor positions, $W$ and $H$ denote touchpad size, and $C$ is the pixel-meter converter based on display's DPI. Plotting C-D gain against input speed revealed the macOS transfer function curve, as shown in Figure 1a. To apply this transfer function, we developed `LibPad` based on Libpointing [2], a library originally designed for mouse input. Libpad is a touchpad-specific toolkit that accepts absolute input coordinates of the touchpad and supports the application of a custom transfer function for touchpad input.

# 3 Pilot Study

***Participant.*** Six participants from a local university (5 males, 1 female) were recruited. Participants' average age was 24.17 years ($\sigma$ = 2.23). 3 participants primarily used a MacBook with OS X, and others used LG and Lenovo laptops with Windows 11. MacBook users in our study typically used the touchpad in daily tasks, whereas Windows laptop users typically used a mouse.

***Design.*** The experiment followed a 3x3x2x3 within-subject design with the following independent variables:

- **Target Width**: 0.28 cm, 0.56 cm, 1.12 cm
- **Amplitude**: 1.92 cm, 3.20 cm, 5.12 cm
- **Time Limit**: 1000 ms, 2000 ms
- ***Configuration.*** Windows-default transfer function on ThinkBook (***WD***), macOS-default transfer function on MacBook Pro (***MD***), and macOS-default transfer function on ThinkBook (***WM***)

***Apparatus.*** The application was implemented on a Lenovo ThinkBook 14 G6 running Windows 11, and a MacBook Pro 14 (2024) running macOS. ThinkBook's touchpad measures 12 cm × 7.5 cm, while the MacBook's touchpad measures 13 cm × 8 cm. The pointing device was an integrated touchpad.

***Task and Procedure.*** Participants were instructed to click 30 circularly positioned targets (See Figure 1b) per condition and asked to perform a task as quickly and accurately as possible. After a brief task explanation, Participants completed practice trials until familiarized. During the experiment, we recorded the coordinates of the touchpad and cursor on the display.

| | MD | | WD | | WM | | F value | | |
|---|---|---|---|---|---|---|---|---|---|
| | Mean | SD | Mean | SD | Mean | SD | Config | ID | Config × ID |
| MT | 840.87 | 285.59 | 1300.41 | 495.61 | 1256.19 | 469.06 | 3262.4$^\dagger$ | 884.75$^\dagger$ | 33.66$^\dagger$ |
| ER | 17.01 | 25.34 | 45.03 | 37.18 | 40.99 | 37.64 | 734.81$^\dagger$ | 310.65$^\dagger$ | 11.60$^\dagger$ |
| TE | 1.08 | 0.37 | 1.23 | 0.57 | 1.17 | 0.49 | 51.67$^\dagger$ | 17.55$^\dagger$ | 6.20$^\dagger$ |
| OC | 0.02 | 0.13 | 0.10 | 0.13 | 0.10 | 0.13 | 879.88$^\dagger$ | 43.10$^\dagger$ | 70.27$^\dagger$ |
| SC | 1.39 | 1.62 | 2.38 | 1.83 | 1.94 | 1.67 | 396.05$^\dagger$ | 313.53$^\dagger$ | 12.48$^\dagger$ |

**Table 1: Descriptive statistics and ART ANOVA F-values. MD: Mac Default, WD: Win Default, WM: Win Mac transfer function. † denotes $p < 0.001$.**

| | MD – WD | | | MD – WM | | | WD – WM | | |
|---|---|---|---|---|---|---|---|---|---|
| | Est. | SE | $z$ | Est. | SE | $z$ | Est. | SE | $z$ |
| MT | -3650.62 | 50.61 | -72.14$^\dagger$ | -3421.90 | 50.61 | -67.61$^\dagger$ | 228.72 | 50.54 | 4.53$^\dagger$ |
| ER | -2319.39 | 64.98 | -34.70$^\dagger$ | -1948.85 | 64.98 | -29.99$^\dagger$ | 370.54 | 64.89 | 5.71$^\dagger$ |
| TE | -556.09 | 55.05 | -10.10$^\dagger$ | -224.33 | 55.05 | -4.07$^\dagger$ | 331.76 | 54.98 | 6.03$^\dagger$ |
| OC | -1429.83 | 39.54 | -36.16$^\dagger$ | -1444.34 | 39.54 | -36.53$^\dagger$ | -14.51 | 39.49 | -0.37 |
| SC | -1801.41 | 64.54 | -27.91$^\dagger$ | -1104.53 | 64.54 | -17.11$^\dagger$ | 696.87 | 64.45 | 10.81$^\dagger$ |

**Table 2: Pairwise contrast estimates, SEs, and z-values between *Configurations.* † denotes $p < 0.001$.**

## 3.1 Results

We compared *Movement Time* (MT), *Error Rate* (ER) [3], *Overshooting Count* (OC) [4], *Target Entry* (TE) [10], and *Submovement Count* (SC) [11] as performance metrics. Since the Shapiro-Wilks test indicated non-normal distributions ($p < .05$) for all conditions, we applied Aligned Rank Transform (ART) [20] to examine statistical significance as shown in Table 1, including interaction effects. All comparisons between *Configuration* groups showed further significant differences ($p < .001$) shown in Table 2, except for the post hoc comparison of *Overshooting* between *WD* and *WM* ($p = .92$).

***Movement Time.*** Average Movement Time was shortest on *MD*, followed by *WM* and *WD*. Post-hoc showed *MD* was significantly faster than both *WD* and *WM* ($p < .001$), as shown in Figure 2.

***Error Rate.*** Error rate was lowest for *MD* and highest for *WD*. Post-hoc analysis revealed $p < .001$ for all comparisons, indicating that *MD* was significantly more accurate than *WM* and *WD*.

***Target Entry.*** Participants showed an average of 1.08 entries on *MD*, followed by 1.17 entries on *WM* and 1.23 entries on *WD*. Post-hoc analysis indicated that participants made significantly fewer entries on *MD* compared to both *WD* and *WM*.

***Overshooting Count.*** Average Overshooting Count was 0.018 counts on *MD*, significantly higher than both *WD* and *WM*, though the difference between *WD* and *WM* was not significant.

***Submovement Count.*** Average Submovement Count was lowest on *MD*, with Post-hoc comparison showing all effects significant.

# 4 Conclusion

In this study, we examined how OS-specific transfer functions of touchpads affect user experience across different laptop hardware. Our results demonstrates that variations in transfer functions significantly affect pointing performance, even on identical laptop hardware. Moreover, hardware differences paired with identical transfer function also affect on pointing performance. These results indicate the importance of considering both hardware and software factors when evaluating and optimizing usability of touchpads.

## References

[1] Angie Avera, Christy Harper, Natalia Russi-Vigoya, and Stephen Stoll. 2016. Effects of touchpad size on pointing and gestural input area and performance. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, Vol. 60. SAGE Publications Sage CA: Los Angeles, CA, 825–829.

[2] Géry Casiez and Nicolas Roussel. 2011. No more bricolage! Methods and tools to characterize, replicate and compare pointing transfer functions. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*. 603–614.

[3] Géry Casiez, Daniel Vogel, Ravin Balakrishnan, and Andy Cockburn. 2008. The impact of control-display gain on user performance in pointing tasks. *Human–computer interaction* 23, 3 (2008), 215–250.

[4] Edward Robert FW Crossman and PJ Goodeve. 1983. Feedback control of hand-movement and Fitts' Law. *The Quarterly Journal of Experimental Psychology Section A* 35, 2 (1983), 251–278.

[5] Hetty Dillen, James G Phillips, and James W Meehan. 2005. Kinematic analysis of cursor trajectories controlled with a touchpad. *International Journal of Human-Computer Interaction* 19, 2 (2005), 223–239.

[6] Hacker News users. 2023. Ask HN: What makes the MacBook touchpad so good? https://news.ycombinator.com/item?id=36608840

[7] N Kargar, AR Choobineh, M Razeghi, S Keshavarzi, and N Meftahi. 2018. Posture and discomfort assessment in computer users while using touch screen device as compared with mouse-keyboard and touch pad-keyboard. *Work* 59, 3 (2018), 341–349.

[8] Seonho Kim, Munjeong Kim, Jonghyun Kim, Donghyeon Kang, Sunjun Kim, and Byungjoo Lee. 2025. Hardware-Embedded Pointing Transfer Function Capable of Canceling OS Gains. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*. 1–15.

[9] Byungjoo Lee, Mathieu Nancel, Sunjun Kim, and Antti Oulasvirta. 2020. Auto-Gain: Gain function adaptation with submovement efficiency optimization. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. 1–12.

[10] I Scott MacKenzie, Tatu Kauppinen, and Miika Silfverberg. 2001. Accuracy measures for evaluating computer pointing devices. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. 9–16.

[11] David E Meyer, Richard A Abrams, Sylvan Kornblum, Charles E Wright, and JE Keith Smith. 1988. Optimality in human motor performance: ideal control of rapid aimed movements. *Psychological review* 95, 3 (1988), 340.

[12] Microsoft Community. 2018. Differences of mouse pointer velocity between Windows 7 and 10. https://answers.microsoft.com/ko-kr/windows/forum/all/%EC%9C%88%EB%8F%84%EC%9A%B0-7-%EA%B3%BC-10/37599356-bc33-4634-8a2c-ad255cad6532 Accessed: July 8, 2025; Published: July 28, 2018.

[13] Kazuyuki Mizuhara, Hiroyuki Hatano, and Katsutoshi Washio. 2013. The effect of friction on the usability of touchpad. *Tribology International* 65 (2013), 326–335.

[14] Eunji Park and Byungjoo Lee. 2020. An intermittent click planning model. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. 1–13.

[15] Nicolas Roussel, Géry Casiez, Jonathan Aceituno, and Daniel Vogel. 2012. Giving a hand to the eyes: leveraging input accuracy for subpixel interaction. In *Proceedings of the 25th annual ACM symposium on User interface software and technology*. 351–358.

[16] Sameerajan Suresh, David Kaber, and Michael Clamann. 2014. Effects of laptop touchpad texturing on user performance. *International Journal of Human-Computer Interaction* 30, 6 (2014), 470–479.

[17] Justin Turner. 2020. When programming on a laptop: mouse or trackpad? https://dev.to/turnerj/when-programming-on-a-laptop-mouse-or-trackpad-4065/comments

[18] Reddit Users. 2017. Why is the touchpad of the MacBooks so good? https://www.reddit.com/r/apple/comments/6c4frp/why_is_the_touchpad_of_the_macbooks_so_good/ Discussion thread on Reddit.

[19] Reddit users. 2022. What makes MacBook's touchpad so good? https://www.reddit.com/r/apple/comments/wjofc2/what_makes_macbooks_touchpad_so_good/

[20] Jacob O Wobbrock, Leah Findlater, Darren Gergle, and James J Higgins. 2011. The aligned rank transform for nonparametric factorial analyses using only anova procedures. In *Proceedings of the SIGCHI conference on human factors in computing systems*. 143–146.

[21] Jacob O Wobbrock, Kristen Shinohara, and Alex Jansen. 2011. The effects of task dimensionality, endpoint deviation, throughput calculation, and experiment design on pointing measures and models. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 1639–1648.