Computational Cognitive Modeling of Touch and Gesture on Mobile Multitouch Devices: Applications and Challenges for Existing Theory

Kristen K. Greene[1], Franklin P. Tamborello[2], Ross J. Micheals[1]

[1]National Institute of Standards and Technology[*1], Gaithersburg, MD
{kristen.greene, ross.micheals}@nist.gov

[2]Cogscent, LLC, Washington, DC
Frank.tamborello@cogscent.com

Abstract. As technology continues to evolve, so too must our modeling and simulation techniques. While formal engineering models of cognitive and perceptual-motor processes are well-developed and extensively validated in the traditional desktop computing environment, their application in the new mobile computing environment is far less mature. ACT-Touch, an extension of the ACT-R 6 (Adaptive Control of Thought-Rational) cognitive architecture, seeks to enable new methods for modeling touch and gesture in today's mobile computing environment. The current objective, the addition of new ACT-R interaction command vocabulary, is a critical first-step to support modeling users' multitouch gestural inputs with greater fidelity and precision. Immediate practical application and validation challenges are discussed, along with a proposed path forward for the larger modeling community to better measure, understand, and predict human performance in today's increasingly complex interaction landscape.

Keywords: ACT-R, ACT-Touch, cognitive architectures, touch and gesture, computational cognitive modeling, modeling and simulation, movement vocabulary, gestural input, mobile handheld devices, multitouch tablets, model validation, Fitts' Law

1 Introduction

Research in Human-Computer Interaction (HCI) demonstrates that the design of tools and procedures significantly impacts total human-system performance. Formal engineering models of cognitive and perceptual-motor processes can aid system design and evaluation. ACT-R is a formally specified theory of human cognition, perception,

---

[*] Disclaimer: Any mention of commercial products or reference to commercial organizations is for information only; it does not imply recommendation or endorsement by the National Institute of Standards and Technology nor does it imply that the products mentioned are necessarily the best available for the purpose.

and action that enjoys wide scientific support and includes relatively rich perception and motor action modeling capabilities [1] and [2]. However, ACT-R and other modeling frameworks have historically assumed the modeled user is seated at a desktop computer with a monitor, keyboard, and mouse. This assumption was both necessary and appropriate given the basic research from which said models were developed; decades of behavioral research in cognitive science and HCI were conducted in the traditional desktop computing environment. Due to recent advances in pervasive computing technology, this previously dominant interaction paradigm is rapidly giving way to mobile touchscreen devices [3].

Our project focuses on advancing modeling methods for multitouch tablet devices, motivated largely by their use in NIST's Biometric Web Services (BWS) project [4], which enables remote control of biometric devices by handheld touchscreen computers. In mission-critical systems, it is often difficult or impossible to gain access to the appropriate users in sufficient numbers and contexts. For example, it would be inadvisable—even dangerous—to deploy a large experimental biometrics system simply for research purposes at customs and border control. Instead, computational cognitive modeling can significantly augment existing usability testing methods by simulating human performance in these types of complex, dynamic systems. The current work will allow NIST's BWS project to model human operators using small handheld tablets and networked biometric sensors to capture biometric data, ultimately exploring and measuring effects of operator error, network delays, and sensor failure on total system performance. More importantly, the current work provides the basic functional foundation upon which to advance general modeling theory and practice across a variety of existing and emerging mobile task domains.

1.1 ACT-R

ACT-R is a computational cognitive architecture, a general theory of human cognition instantiated in an open-source modeling and simulation software package [2]. This means that it is a formally specified framework for constructing models of how people perform tasks, and these models generate quantitative predictions. ACT-R incorporates a motor planning and execution module adapted from another cognitive architecture, EPIC [5]. According to this model of motor planning and execution, cognition has a vocabulary of simple movement styles such as ply and other, more complex styles composed from those, such as to move the mouse cursor. Each movement is specified by features such as which hand, which finger(s), movement direction and movement distance. Movements are requested by central cognition and processed in stages by a motor module. However, ACT-R and other modeling frameworks still widely assume the modeled user is seated at a traditional desktop computer with a monitor, keyboard, and mouse. By extending the manual motor capabilities of ACT-R to include such new command vocabulary as swipe, pinch, and rotate gestures, we can better simulate the more complex user interactions that exist in the milieu of novel mobile touchscreen devices today.

2 ACT-Touch

ACT-Touch extends ACT-R's existing motor module by providing it with a simulated multitouch display device and multitouch display gesture motor vocabulary; ACT-Touch is implemented as Lisp code that is meant to load with ACT-R's software. ACT-Touch can be downloaded as a single archive from Cogscent, LLC's website [6]. The architectural component to ACT-Touch is a library of manual motor request extensions, including assumptions and changes specific to the multitouch task environment. One such difference between ACT-R and ACT-Touch is the default starting hand position. ACT-Touch assumes that in the mobile touchscreen domain, a user's hands no longer start at traditional home row positions (left and right index fingers positioned over the F and J keys respectively) because there is no longer a desktop keyboard present. Instead, a user's hands are assumed to start on both sides of the tablet (left hand on left side of tablet, right hand on right side of tablet).

The units of measurement for distance specifications in motor movement requests also differs between ACT-R (distance in "keys") and ACT-Touch (distance in pixels, at 72 ppi). Motor movement distance in ACT-R was measured in "keys" (the distance between two adjacent keys in the same row or the same column on the simulated desktop keyboard). This works well for modeling typing tasks in the desktop computing environment given consistency in key sizes and spacing for traditional QWERTY physical keyboards. However, in the newer mobile touchscreen computing environment, virtual keyboards can vary significantly. Virtual keyboard sizes vary across mobile devices based on physical differences in the maximum available touchscreen real estate. Virtual keyboards may even vary within a single device, depending on device orientation (landscape versus portrait mode) and whether the virtual keyboard is split; in addition to reducing key sizes, splitting the keyboard also changes the relative distance between some keys more so than others. For these reasons, ACT-Touch uses pixels to specify distances for motor movement requests.

ACT-Touch introduces a z-axis for motor movements, which represents vertical distance between the surface of the multitouch display device and the model's finger above it. ACT-Touch adds several basic motor movement styles (tap, swipe, pinch, and rotate gestures) that are commonly used across a variety of today's handheld mobile devices. As with ACT-R, ACT-Touch's basic movement styles are then combined to form more complex movements. The specific multitouch gestural commands currently implemented in ACT-Touch are: tap, peck-tap, peck-recoil-tap, tap-hold, tap-release, tap-drag-release, swipe, pinch, rotate, and move-hand-touch.

A tap gesture in ACT-Touch simulates the model's finger moving toward and momentarily contacting the surface of the multitouch display directly under the finger's current location; this is analogous to ACT-R's punch command for a traditional desktop keyboard, where the key to be pressed is already directly below the finger. A peck-tap gesture in ACT-Touch simulates moving the model's finger to a new location and tapping that location on the multitouch display (as a continuous movement). Peck-recoil-tap is similar, except the model's finger returns to its starting location after having tapped the display. Tap-hold simulates the model tapping and holding a finger on the surface of the multitouch display until a tap-release movement is requested. Tap-drag-release simulates a drag-and-drop movement (e.g., the model will tap-hold the display surface under its finger, move said finger to a new location without breaking contact with the display surface, then release its finger from the display). For a swipe, the model moves the specified number of fingers (1-5, incrementing from index to pinkie and thumb) onto the display, moves them the specified distance and direction, then releases them from the display. For a pinch/reverse pinch gesture, the model moves the specified finger and thumb onto the display, moves them together/apart by the difference between the specified start- and end-widths (in pixels), then releases them from the display. A rotate gesture is similar (move finger and thumb to display, move them on display surface, release from display), but the distance between the model's digits remains constant as they are moved rotationally; direction of rotation is specified in radians.

The ACT-Touch distribution includes a simulated virtual multitouch display device (1,024 pixels wide x 768 pixels tall) with which a model can interact using the new motor movement commands described above. Default hand positions for the model are at either side of the display, centered approximately vertically. The ACT-Touch distribution also includes a virtual experiment window, a derivative of Mike Byrne's Experiment-Window ACT-R experiment instrumentation library [7]. The modified virtual experiment window allows modelers to build a multitouch display-based task environment and collect data for ACT-Touch. ACT-Touch, together with ACT-R, outputs a time-stamped series of predicted user behaviors. The instrumentation built into ACT-Touch supports capture of user latencies and errors within the simulated mobile touchscreen computing environment.

3 Model Validation

A critical next step is to compare ACT-Touch's motor movement predictions with human behavioral data. After testing and validating ACT-Touch's predictions for single-finger discrete tapping input (i.e., tap, peck-tap, and peck-recoil-tap), we will move on to testing predictions for more complex, continuous single-finger gestures, then progress in an orderly manner through two-, three-, and four-fingered multitouch gestures. As ACT-Touch predicts both movement latencies and their associated XY touch coordinates, we are collecting human data at a similarly fine level of granularity. These model validation efforts are currently under way, starting with implementation of customized touch-logging for tapping tasks on mobile handheld iOS devices (currently Apple iPads). We record a timestamped log of user touch events (XY screen coordinates) and corresponding system responses (e.g., button press detected, popover dismissed) via the actual mobile device with which they are completing the task.

3.1 Touch-Logging

We have already identified several touch-logging challenges of particular interest for model validation efforts; some pose immediate issues, while others may not apply until the tasks we model become more complex and representative of real-world activities. Of immediate relevance is the question of sampling rates during scrolls: is there a single ideal sampling rate for these purposes or does it vary based on the complexity of the gesture and task being modeled? Sampling rates that are too low may not provide sufficiently detailed data for investigating more subtle effects in users' movement trajectories. On the other hand, higher sampling rates cause touch-logging files to become very large, very quickly, especially when system events and notifications are also logged.

Touch-logging for certain native iOS interface elements can be particularly nuanced. In native iOS applications with unusually small buttons (i.e., smaller than the minimum iPhone button size of 44x44 pixels recommended in Apple's Human Interface Guidelines [8]), the active touch area is automatically extended invisibly beyond the button. These invisible button extensions make the effective target size larger than the visible target, as is the case with the "Detail Disclosure" button or the standard "Back" button in navigation-based iPhone applications. Another example of enlarged hit areas occurs on the inner edges of a split iOS virtual iPad keyboard. Although it is fairly easy to log notifications of when the keyboard appears and disappears, capturing the actual user input event (i.e., timestamp and XY touch coordinates for a tap on the "hide keyboard button") that triggered the keyboard hide is not possible without additional custom code.

Finally, the native iOS magnifier loupe may also pose challenges for modeling certain tasks, as there are no system notifications to log when the magnifier loupe appears or disappears. Attempting to detect it by listening for long presses in a text field has not worked thus far, as the act of listening disables the loupe. Current options include adding invisible controls on top of the text field to detect the long press, or re-implementing the entire functionality of the magnifier loupe. It may be the case that the programming effort required to implement these types of iOS work-arounds would be better spent elsewhere. Additional work is necessary to compare and contrast relative touch-logging capabilities between different mobile development platforms and devices.

4 Future Work

Assumptions borne from directly adapting EPIC's model raise some interesting questions in the touchscreen domain. Fitts' Law requires a target width, but what exactly is the target of a swipe or a pinch? A long fast swipe used for scrolling quickly through multiple pages versus one intended to scroll to a more precise location within a single page? The short quick flick required to turn a virtual book page? Does it make more sense to construct two-, three-, and four-fingered swipes as different movement styles, as opposed to ACT-Touch's current implementation of a single movement style with a feature specifying the number of fingers?

The preceding questions pertain specifically to movement predictions for the gesturing hand, but what about the stabilizing hand? Recent anthropometric research suggests that people will cycle between different stabilizing hand positions when completing extended tasks in an unsupported environment [9]. The authors suggest future work to examine how input gestures may alter the stabilizing hand [9], but we are more interested in the reverse question: how does the stabilizing hand impact input gestures? Furthermore, do different stabilizing postures facilitate or inhibit motor learning and fatigue for the gesturing hand? Although motor learning and fatigue have not been widely modeled in the ACT-R community to date, this may change as the body of HCI/HF literature in this area continues to grow.

5 Discussion

The current work, ACT-Touch, successfully augments the existing ACT-R cognitive architecture for modeling touch and gesture on mobile devices. Specific multitouch gestural commands currently implemented include tap, peck-tap, peck-recoil-tap, tap-hold, tap-release, tap-drag-release, swipe, pinch/reverse pinch, rotate, and move-hand-touch. A critical next step is to compare model predictions with basic behavioral data, starting with simple psychophysical tasks to examine variance between people and gesture types, and examine motor learning and manual fatigue. Among other research questions, what is the target of a swipe? A pinch? Fitts' Law requires a target width, and recent literature suggests that traditional application of Fitts' Law may not always be sufficient for 3-D gestures and small touchscreen smartphones [10], [11]. Future research will focus on understanding when and how traditional HCI methods need modification to accurately model users interacting with novel technologies. To do so, modeling tools themselves must continually evolve to incorporate new interaction paradigms. The current work provides a technical foundation for modeling the evolution of touch and gesture on novel mobile devices.

Acknowledgements

This work is funded in part by a Measurement Science and Engineering grant from the National Institute of Standards and Technology's Information Technology Laboratory (ITL) Grant Program. Federal Funding Opportunity 2012-NIST-MSE-01. Grant 60NANB12D134, "Formal Model of Human-System Performance," awarded to Cogscent, LLC.

References

1. Anderson, J. R., Bothell, D., Byrne, M. D., Douglass, S., Lebiere, C., & Qin, Y. (2004). An integrated theory of the mind. *Psychological Review, 111(4)*, 1036-60.
2. Carnegie Mellon University, http://act-r.psy.cmu.edu/about/
3. Dediu, H. (2012). Apple sold more iOS devices in 2011 than all the Macs it sold in 28 years. Retrieved from http://www.asymco.com/2012/02/16/ios-devices-in-2011-vs-macs-sold-it-in-28-years/.
4. Biometric Web Services, http://bws.nist.gov/
5. Kieras, D. E., & Meyer, D. E. (1997). An overview of the EPIC architecture for cognition and performance with application to human-computer interaction. Human-Computer Interaction, 12(4), 391-438.
6. Cogscent, LLC, http://cogscent.com/
7. CHIL, http://chil.rice.edu/projects/RPM/index.html
8. Apple. (2012). iOS Human Interface Guidelines. Retrieved from *http://developer.apple.com/library/ios/documentation/userexperience/conceptual/mobilehig/MobileHIG.pdf*
9. Feathers, D. J., & Zhang, H. (2012). Holding a Multi-touch Tablet with One Hand: 3D Modeling and Visualization of Hand and Wrist Postures. *Proceedings of the HFES 56th Annual Meeting.*
10. Jo, J. H. and Kim, I. (2012). Adjusting Fitts' Paradigm for Small Touch-Sensitive Input Device with Large Group of Users. *Proceedings of the HFES 56th Annual Meeting.*
11. Park, D. and Myung, Rohae. (2012). Prediction of a Three-Dimensional Pointing Task through Extending the Motor Module of ACT-R. *Proceedings of the HFES 56th Annual Meeting*.