Potential Limitations of Multi-touch Gesture Vocabulary: Differentiation, Adoption, Fatigue

Wendy Yee

User Centric, Inc.
2 Trans Am Plaza Dr, Ste 100,
Oakbrook Terrace, IL 60181 USA
wyee@usercentric.com

Abstract. The majority of gestural interactions in consumer electronics currently represent "direct" gestures related to the direct manipulation of onscreen objects. As gestural interactions extend beyond consumer electronics and become more prevalent in productivity applications, these gestures will need to address more abstract or "indirect" actions. This paper addresses some of the usability concerns associated with indirect gestures and their potential limitations for the typical end-user. In addition, it outlines a number of considerations for the integration of abstract gestures with productivity workspaces.

Keywords: Interaction design, multi-touch, software design, gestures.

1 Introduction

With the advent of the iPhone, Surface, and other consumer-oriented multi-touch devices, gestural interactions have become an increasingly common mode of interacting with software environments [1]. Since gestures tap into a very basic part of the human syntax [2], these gestural interactions may be an especially appropriate way to make technology accessible to a broader audience.

The majority of gestural interactions in consumer electronics currently represent "direct" gestures related to the direct manipulation of onscreen objects, such as the standard "dragging" action used to move onscreen objects on the Apple iPhone and Microsoft's Surface table. As gestural interactions extend beyond consumer electronics and become more prevalent in productivity applications, these gestures will need to address more abstract or "indirect" actions, such as acting on a group of onscreen objects or navigating through an information hierarchy. This paper addresses some of the usability concerns associated with indirect gestures and their potential limitations for the typical end-user. In addition, it outlines a number of considerations for the integration of abstract gestures with productivity applications.

2 Review of Current Criteria for Effective Gestures

Gestural interaction has its roots in the 1980s and 1990s [3], but has become increasingly mainstream with the integration of gesture recognition with operating systems

J.A. Jacko (Ed.): Human-Computer Interaction, Part II, HCII 2009, LNCS 5611, pp. 291–300, 2009.

[4, 5], including applications supported by recent-model 2008 MacBooks with multi-touch trackpads [6, 7]. Although gestural manipulation of items and files have been commercially available through iGesture keypads [8] and graphic design tablets that recognize direction and stroke pressure [9], gestural interactions have not yet achieved integration with the daily activities associated with typical productivity applications (e.g., writing, analyzing data in spreadsheets, manipulating images, editing video).

Across existing literature [10, 11, 12, 13, 14], five major criteria are thought to contribute to the effectiveness of gestural interactions:

A. **Provide a high degree of interaction context:** The application or system interface should make it clear to the user that gestures can be used. From the user perspective, gestures should be obvious and intuitive in the context of relevant tasks.
B. **Allow users to gesture with minimal effort:** Effective gestures should be simple to perform and should not require unusual dexterity.
C. **Use appropriate metaphors:** Gestures should have a logical relationship with the application functionalities that they represent, both in the type of movement and type of interaction with objects that they are acting on.
D. **Be designed for repetitive use and minimal muscle stress:** Given the repetitive nature of most activities associated with productivity, consumer, and gaming applications, gesture fatigue is likely to be a common issue among end-users.
E. **Facilitate accurate recognition by the application**: In the ideal world, gestures would be sufficiently unambiguous to applications so that minor variations of the same gesture can be recognized by software as having the same function. In addition, applications would ideally recognize completely different gestures with a minimum of false-positives or cross-positives.

These criteria have been especially relevant when gestures have been used to replace basic navigation and selection actions that were formerly associated with point-and-click selections (mouse and trackpad) or game controller/joystick selections (gaming systems). Here, they represented "direct" gestures with one-to-one correlation to vector-based movement (e.g., length of the gesture determines the extent of the software action, such as zooming) or single-movement-to-object associations (e.g., dragging a photo across the screen). This direct type of interaction has benefited consumers by reducing their learning curve and focusing their attention on direct interactions with the digital representations of objects.

Now, however, gesture vocabularies are poised to expand beyond basic navigation tasks into productivity applications. Indirect or "abstract" gestures are increasingly common and can now be used to initiate, manipulate, and complete activities that are not associated with direct one-to-one visual representations [15]. For example, the "three-finger swiping" gestures allows users of recent-model MacBooks to gesture on their trackpads and rapidly advance through documents. A related "four-finger swiping" gesture allow users to quickly locate an open window or to hide all windows

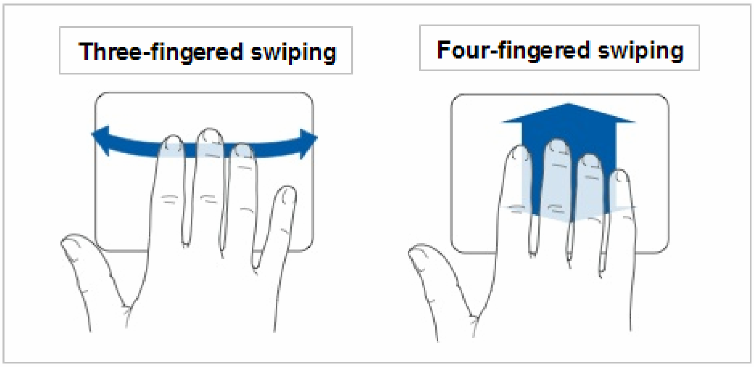

Fig. 1. Examples of indirect gestures on recent-model MacBook trackpads. Source: Apple Support Article HT3211 (February 20, 2009).

without clicking on any specific target. Neither of these indirect gestures require the user to select an object before taking action.

3 Potential Limitations of Using Gestures in Productivity Applications

By their nature, abstract gestures can be more difficult for users to discover and adopt. For example, Fingerworks has developed gestures that will allow users to initiate and execute abstract actions using different combinations of multi-finger gestures and movements [8], which Apple has incorporated into a multi-touch gesture library [16].

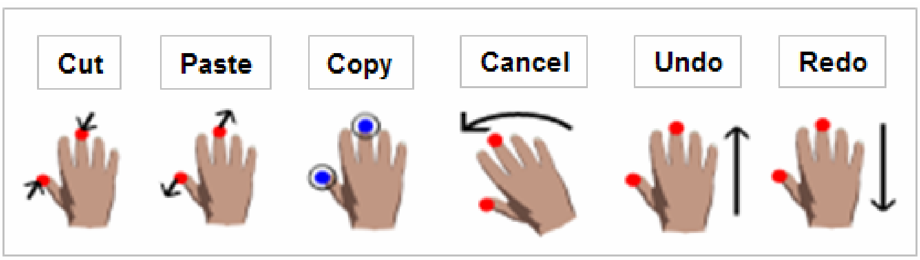

Fig. 2. Finger chords from the Fingerworks gesture keymap for editing. Source: http://www.fingerworks.com/gesture_keymap.html

There are also additional commercial examples of abstract multi-touch gestures that can be used to trigger contextual menus and navigate between and within applications [17, 18]. These are typical events associated with productivity applications.

The relationship between abstract gestures such as finger chords and the requirements of multi-touch gestural interactions is an interesting one because it reflects basic tensions between the need to satisfy two potentially disparate requirements:

- The need to develop gestures that are sufficiently **distinct that they can be recognized by software.** For example, finger chording has been sufficiently commercialized and has a proven level of recognition by software. Finger chording also allows combinations of different finger movements in a consistent and modular way.
- The need to develop abstract gestures that are sufficiently **distinct so they can be effectively cued in end-user behavior and executed by the end-user in repetitive situations**. While the universe of finger and palm gestures that can be recognized by software is somewhat finite, the subset of gestures that consumers can consistently remember and use with minimal cueing by the user interface is likely to be significantly smaller. In the case of finger chording, substantial practice is required to become facile and the learning curve is high.

The use of abstract gestures in productivity applications is further complicated by the likelihood that gestures are likely to occur on a larger physical scale due to the size of multi-touch displays (multi-touch tablet screens, laptop screens, and monitors). To accommodate the space between objects and the general screen real estate, gestures may no longer be limited to finger spans.

Eventually, there will need to be workable guidelines outlining abstract gestures that are intuitive and ergonomic for the end-user but also distinct enough to facilitate recognition by the system. Although the visual user interface context is critical for helping users understand what they can do when faced with abstract tasks, there are obvious limitations for cueing abstract gestures.

4 Additional Criteria for Usable Gestures in Productivity Applications

Based on a meta-analysis of previously published gesture research studies and our exposure to different types of multi-touch interfaces, we propose three additional criteria that should be considered in the ongoing design and testing of gestural interactions when applied to productivity applications:

F. **Minimize the learning curve among users/ increase differentiation among gestures:** One of the concerns that interaction designers have about abstract gestures is reduced discoverability, or a reduced likelihood that users will easily discover an abstract gesture, understand its functionality, and adopt it. Unlike scrolling for navigation or using two fingers to zoom in, there is a much weaker causal (and visual) association between (for example) three-fingered swiping and page navigation. Although abstract gestures have been compared to right-clicking - which also has some discoverability issues - the potential variety of abstract gestures is likely to increase the learning curve much more than right-clicking. (Most mice and trackpads only have one right button but gestures are theoretically limited only by flexibility.)

G. **Cue efficient gestures:** There are currently a limited number of gesture-related UI cues used to prompt users' movements in multi-touch environments. For example, partial-height rows at the top and/or bottom of touchscreen lists and menus are often used to let the user know than they can scroll further up or down the list. On the Surface table, onscreen items or areas that can recognize physical objects (such as a tagged glass or tagged mobile device) will appear to pulsate or glow and will display a simacrulum of the target object. However, these cues typically rely on one-to-one correlations between an onscreen object and the use of a single gesture.

The use of one-to-one gestures may be too repetitive since multiple instances of the same gesture may be required to sort, select, or organize multiple items. Instead, a good gesture ecosystem should help the user identify gesture "shortcuts" and prevent the user from making needless repetitive gestures. If the user should be using a more efficient gesture, there should be consistent cues that alert them to this option.

H. Focus abstract gestures on finger movements: Although full-hand gestures that involve movement of entire hand and forearm might be useful for extending the specific actions, these gestures require more physical effort that limited-range finger movements. Repetitive use of full-hand gestures (where the entire hand moves) is likely to lead to unnecessarily arm fatigue. There may be cases where large-scale gestures may more accurately map to an action type than a small-scale gesture (such as dragging fingertips across a screen to navigate to a new area of a map), but repetitive use of this gesture may be counterproductive. This argues for representation of large-scale actions using equivalent gestures with finger movements.

5 Gestural Considerations

Our hypothesis is that applying these additional usability criteria would yield a more cohesive and self-evident set of abstract gestures in productivity applications. To facilitate further discussion in the interaction design and research community, we suggest the following specific points for the three criteria discussed earlier.

5.1 Minimize Learning among Users and Increase Differentiation among Gestures

- **Examine the gesture ecosystem for an application:** We urge our colleagues in interaction design, HCI, and software development to think of multi-touch gestures as part of a clearly defined "ecosystem" of gestures. As with all ecosystems, the presence of redundant gestures or gesture-related UI elements can make it difficult for the overall ecosystem to thrive.
- **Evaluate and test related gestures:** Are gestures that are not related to one another sufficiently distinct? Can the user differentiate between related gestures and non-related gestures?

- **Apply consistent semantic guidelines to gestures:** Does the overall approach to the design of gestures reflect a consistent approach?
  - For example, are sweeping gestures used for large group selections while small, nudging gestures are used to move single items? Or are small circular movements to bring up context menus vs. large circular movements to act on a group of objects?
  - Full-hand gestures may be best reserved for actions that represent a definitive starting or stopping action. Body language associated with broad gestures suggests that broad strokes should be reserved for formal transitions or actions with a certain degree of completion.
- **Increase discoverability by focusing on the design of the user interface:** The visual cues used to prompt gestures needs to be self-explanatory and provide abundant affordances for prompting and confirming gestures.
  - Well-designed visual cues and affordances in the interface will help users better understand the context, which will in turn help them select the appropriate set of gestures for a set of items or tasks.
  - Users will gauge their success with a gesture interface based on the overall behavior of the interface and its responses to gestures.
  - Based on observations made during user research studies, gestural interfaces that provide **visual and audio cues** for abstract gestures are more likely to be discovered, such as:
    - Solid visual metaphors for relationships (groups, linkages)
    - Affordance for navigating between different views of the same object sets (grouped or sorted by date, name or another attribute)
    - Self-evident and persistent modes for canceling an action and/or undoing the immediate previous action
    - Indication of the type of gestural action initiated
    - Indications of which items can be acted on (or not)
    - Indications of which items are selected (or not).

5.2 Design Efficient Gestures to Increase User Adoption

- Ensure the extensibility of gestures:
  - Make sure that an abstract gesture can be applied to a single item as well as multiple items, regardless of the size of the group of items
  - Separate different types of gestures for initiating new events
  - Support navigation gestures for non-visible items (or navigation access via a preview of items)
- **Select the mode of gesture interaction (with menus or without menus):** The use of abstract gestures will typically require increased context which might take the form of onscreen confirmations and/or controls. For example, depending on an interaction design team's gesture philosophy, one of the following approaches might be used to act on a group of files.

Option A: Close out the task using an indirect gesture.

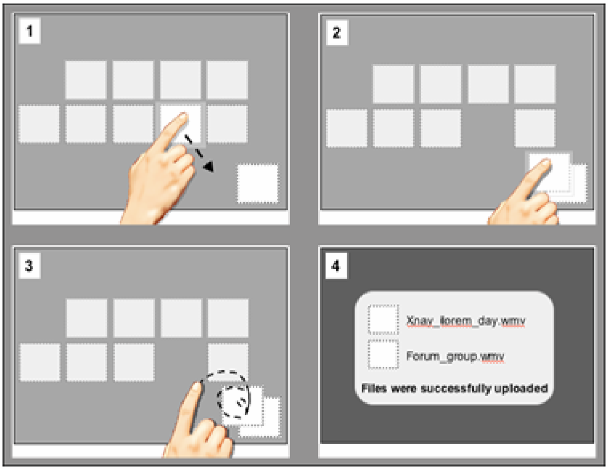

Fig. 3. Use of a direct gesture to select files, followed by use of a fictional indirect gesture to close out the task (upload the selected files)

Option B: Close out the task with an indirect gesture (right hand) while simultaneously using a screen control (left hand) to confirm the selection and prevent the gesture from being applied in error.

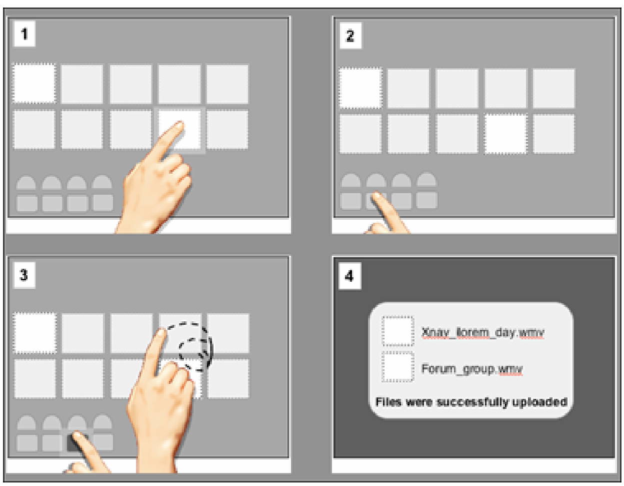

Fig. 4. Use of a direct gesture to select files, followed by **simultaneous** use of an onscreen control and a fictional indirect gesture to close out the task

Option C: Select and group items using direct gestures, then close out the task with screen controls.

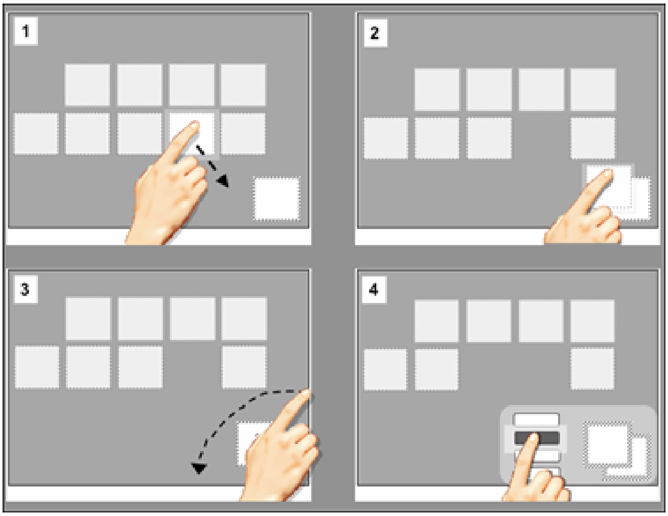

Fig. 5. Use of direct gesture to select and group files, followed by use of onscreen controls to close out the task

- **Leverage the use of zones:** Minimize the number of user errors caused by the accidental positioning or use of gestures in the incorrect areas by limiting recognized gestures to defined active zones:
  - Active zones could be highly context-dependent and appear in close proximity to active or selected objects. Certain gestures might only trigger actions occurring in zones around active or selected objects (e.g., delete, crop, erase, apply filter, run macro, replace, etc)
  - To prevent unrecoverable errors, provide the user with persistent ways to undo gestures using either gestures or menus.

5.3 Considerations: Maximize the Value of Finger Gestures

- **Play to users' strengths:** Users' fingers tend to be very accurate at pointing and are less accurate at dragging large object sets.
  - Optimize more gestures for pointing-related accuracy.
  - Increase error margins for gestures involving dragging.
  - Consider adding elasticity and rebounds to individual and group selections and movements associated with dragging gestures.
- **Do not assume gestures need to scale:** Allow users to act on a group of items by using smaller scale finger gestures once that group is selected. In other words, gestures do not need to be "to scale" once a group selection is made - regardless of the "size" of the group.

6 Evaluating Gesture Ecosystems

We would argue that (a) the strength of gestural interactions is derived from the overall functionality and usability of a complete gesture set and (b) these "gesture ecosystems" are subject to the same enduser expectations for consistency and usability. Given the relatively new state of commercial gestural interactions, it is reasonable to expect that these interactions (both the gestures and the UI context for gestures) will take decades to evolve, similar to the way point-and-click GUIs have evolved over the last twenty years.

The only data-based way to evaluate the effectiveness of multi-touch gestures in productivity applications is to test potential gestures with users at different stages of interaction design and development. The usability adage to "test early and test often" is definitely applicable to the design and development of gestural interactions.

We anticipate that major technology organizations already have specific plans to formalize their own distinct flavors of gestural interactions. Given the supposed advantages of intuitiveness and self-evidency (and discoverability) of gestural interactions, we urge all interaction designers to consider the additional usability criteria for abstract gestures we have discussed in this paper when designing both direct and indirect gestures for productivity applications.

References

1. Green, K.: Touch screens for many fingers. Technology Review, January 18 (2007), `http://www.technologyreview.com/computing/18079/`
2. Goldin-Meadow, S.: The resilience of language: What gesture creation in deaf children can tell us about how all children learn language, p. 262. Psychology Press, New York (2003)
3. Buxton, B.: Multi-touch systems I have known and loved, `http://www.billbuxton.com/multitouchOverview.html`
4. Apple iPhone User Guide (2008), `http://manuals.info.apple.com/en_US/iPhone_User_Guide.pdf`
5. Microsoft: Introduction to Windows Touch, `http://download.microsoft.com/download/a/d/f/adf1347d-08dc-41a4-9084-623b1194d4b2/Win7_touch.docx`
6. Apple: MacBook Air and MacBook Pro (early 2008): Useful keyboard and trackpad tips and shortcuts. Support article HT1115, `http://support.apple.com/kb/HT1115`
7. Apple: MacBook: Using the trackpad Support article HT3211. kb/HT3211, `http://support.apple.com/`
8. Fingerworks iGesture Keypad, `http://www.fingerworks.com/userguides.html`
9. Wacom, `http://www.wacom.com`
10. Keates, S., Robinson, P.: The use of gestures in multimodal input. In: FACM SIGACCESS Conference on Assistive Technologies archive. Proceedings of the third international ACM conference on Assistive Technologies, pp. 35–42 (1998)
11. Hummels, C., Staplers, P.J.: Meaningful gestures for human computer interaction: Beyond Hand Posters. In: Proceedings of the 3rd International Conference on Automatic Face & Gestures Recognition (FG 1998), pp. 591–596. IEEE Computer Society Press, Los Alamitos (1998)

12. Nielsen, M., Storring, M., Moeslund, T.B., Granum, E.: Procedure for developing intuitive and ergonomic gesture interfaces. Technical Report CVMT 03-01 (2003)
13. Karam, M., Schraefel, M.C.: Investigating user tolerance for errors in vision-enabled gestural interactions. In: Proceedings of the working conference on Advanced Visual interfaces table of contents, 225–232 (2006)
14. Pavlovic, V., Sharman, R., Huang, T.: Visual Interpretation of Hand Gestures for Human-Computer Interaction: A Review. IEEE Transactions on Pattern Analysis and Machine Intelligence 19(7), 677–695 (1997)
15. Saffer, D.: Designing Gestural Interfaces, 1st edn. O'Reilly Media, Sebastopol (2008)
16. Elias, J.G., Westerman, W.C., Haggerty, M.M.: Multi-Touch Gesture Dictionary. US Patent Application 20070177803 (2007)
17. Perceptive Pixel, `http://www.perceptivepixel.com`
18. Tse, E., Shen, C., Greenberg, S., and Forlines, C.: Enabling Interaction with Single User Applications through Speech and Gestures on a Multi-User Tabletop. Advanced Visual Interfaces (AVI) (2006)