WritLarge: Ink Unleashed by Unified Scope, Action, & Zoom

Haijun Xia[1,2], Ken Hinckley[1], Michel Pahud[1], Xiao Tu[1], and Bill Buxton[1]

[1]Microsoft Research, Redmond, WA, United States, {kenh, mpahud, xiaotu, bibuxton}@microsoft.com

[2]Dynamic Graphics Project Lab, University of Toronto, Canada, haijunxia@dgp.toronto.edu

ABSTRACT

WritLarge is a freeform canvas for early-stage design on electronic whiteboards with pen+touch input. The system aims to support a higher-level flow of interaction by 'chunking' the traditionally disjoint steps of *selection* and *action* into unified *selection-action phrases*. This holistic goal led us to address two complementary aspects:

- SELECTION, for which we devise a new technique known as the *Zoom-Catcher* that integrates pinch-to-zoom and selection in a single gesture for fluidly selecting and acting on content;

plus:

- ACTION, where we demonstrate how this addresses the combined issues of navigating, selecting, and manipulating content. In particular, the designer can transform select ink strokes in flexible and easily-reversible representations via *semantic*, *structural*, and *temporal* axes of movement that are defined as conceptual 'moves' relative to the specified content.

This approach dovetails zooming with lightweight specification of scope as well as the evocation of context-appropriate commands, at-hand, in a location-independent manner. This establishes powerful new primitives that can help to scaffold higher-level tasks, thereby unleashing the expressive power of ink in a compelling manner.

Author Keywords

electronic whiteboard; pen+touch; bimanual input; toolglass.

ACM Classification Keywords

H.5.2 [Information Interfaces and Presentation]: Input

INTRODUCTION

Electronic whiteboards remain surprisingly difficult to use in the context of creativity support and design. A key problem is that once a designer places strokes and reference images on a canvas, actually doing anything useful *with a subset of that content* involves numerous steps. Hence, ***scope***—that is, SELECTION of content—is a central concern, yet current techniques often require switching modes and encircling ink with a lengthy lasso, if not round-trips to the edge of the display [18,51]. Only then can the user take ACTION, such as to copy, refine, or re-interpret content.

CHI 2017, May 06 - 11, 2017, Denver, CO, USA

ACM 978-1-4503-4655-9/17/05…$15.00
DOI: http://dx.doi.org/10.1145/3025453.3025664.

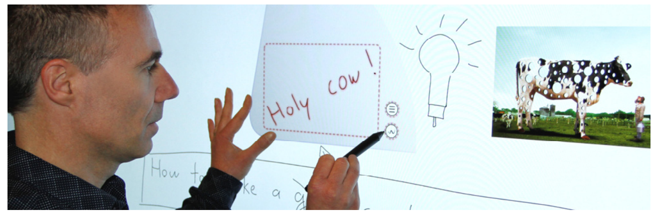

Figure 1. *WritLarge* uses bimanual input to integrate *selection* by a multi-touch framing gesture with *action* via the pen.

Such is the stilted nature of selection and action in the digital world. But it need not be so. By contrast, consider an everyday manual task such as sandpapering a piece of woodwork to hew off its rough edges. Here, we use our hands to grasp and bring to the fore—that is, *select*—the portion of the work-object—the wood—that we want to refine. And because we are working with a tool—the sandpaper—the hand employed for this 'selection' sub-task is typically the non-preferred one, which skillfully manipulates the frame-of-reference [22] for the subsequent 'action' of sanding, a complementary sub-task articulated by the preferred hand.

Therefore, in contrast to the disjoint subtasks foisted on us by most interactions with computers, the above example shows how complementary manual activities lend a sense of flow that "chunks" [11] selection and action into a continuous *selection-action phrase* [27]. By manipulating the workspace, the off-hand shifts the context of the actions to be applied, while the preferred hand brings different tools to bear—such as sandpaper, file, or chisel—as necessary.

The main goal of *WritLarge*, then, is to demonstrate similar continuity of action for electronic whiteboards. This motivated free-flowing, close-at-hand techniques to afford unification of selection and action via bimanual pen+touch interaction. To address SELECTION, we designed a lightweight, integrated, and fast way for users to indicate scope, called the *Zoom-Catcher* (Fig. 1), as follows:

> *With the thumb and forefinger of the non-preferred hand, the user just* ***frames*** *a portion of the canvas.*

This sounds straightforward, and it is—from the user's perspective. But this simple reframing of pinch-to-zoom affords a transparent, toolglass-like palette—the Zoom-Catcher, manipulated by the nonpreferred hand—which floats above the canvas, and the ink strokes and reference images thereupon. The Zoom-Catcher elegantly integrates numerous steps: it dovetails with pinch-to-zoom, affording multi-scale interaction; serves as mode switch, input filter, and an illumination of a portion of the canvas—thereby doubling as a lightweight specification of scope; and once latched-in, it sets the stage for ACTION by evoking commands at-hand, revealing context-appropriate functions in a location-independent manner, where the user can then act on them with the stylus (or a finger).

Building from this key insight, our work contributes unified SELECTION and ACTION by bringing together the following:

- Lightweight specification of scope via the Zoom-Catcher.
- In a way that continuously dovetails with pinch-to-zoom.
- Thus affording unified, multi-scale selection and action with pen+touch, and both hands, in complementary roles.
- These primitives support flexible, interpretation-rich, and easily-reversible representations of content, with a clear mental model of **levels** spatially organized along ***semantic*, *structural*, and *temporal* axes of movement**.
- Our approach thereby unleashes many natural attributes of ink, such as the position, size, orientation, textual content, and implicit structure of handwriting.
- And in a way that leaves the user in complete control of *what* gets recognized—as well as *when* recognition occurs—so as not to break the flow of creative work [14].
- A preliminary evaluation of the system with users suggests the combination of zooming and selection in this manner works extremely well, and is self-revealing for most users.

The structure of the paper that follows is somewhat unusual because our goals are holistic, seeking to establish a new approach to unified selection and action through higher-level selection-action phrases, rather than (for example) following the formula of a classic technique paper focusing on just the selection step. At first, we focus on the Zoom-Catcher, unpacking its properties and detailing its design evolution in depth. In the latter half of the paper, we then switch gears and show how this foundation evokes a rich set of in-context commands close at-hand, particularly by transforming select ink strokes in easily-reversible representations along our semantic, structural, and temporal axes of movement. Collectively these contributions aim to reduce the impedance mismatch of human and technology, thus enhancing the interactional fluency between a creator's ink strokes and the resulting representations at their command.

AN EXAMPLE OF INK UNLEASHED—HOLY COW!

Figure 2 shows how our approach unleashes latent attributes of ink. The user just frames the desired handwriting with thumb and forefinger, and uses one of the radial menus thus revealed to move one step up our 'semantic' representational axis. The recognized text *with the location, point size, and baseline orientation intact* then appears on the canvas.

The user can just as easily back out of this representation, by stroking the radial menu in the reverse direction to move one level down the semantic axis—that is, to 'unrecognize' the text and revert to the original hand-drawn strokes. This is just one example of the flexibility of representation [23,54,71] and re-interpretation [58] afforded by *WritLarge*.

Note how the style of interaction afforded by this example—particularly the simplicity of expressing all of these attributes simultaneously through the ink itself—contrasts with much of current practice. Presentation software, for example, requires many steps: create a text box, place it approximately in the right place, size it, type in the text string, orient it, select the text, adjust the point size, and then re-position the rotated text box to its final desired location.

But *WritLarge* achieves this in fewer steps, and in a direct-at-hand manner. The Zoom-Catcher indicates what to recognize, and when to recognize it—all in a single unified selection-action phrase. The remarkable economy and continuity of gesture thus achieved harkens back to the dovetailing, and sense of flow, in our early motivating example of sandpapering rough edges off a piece of wood.

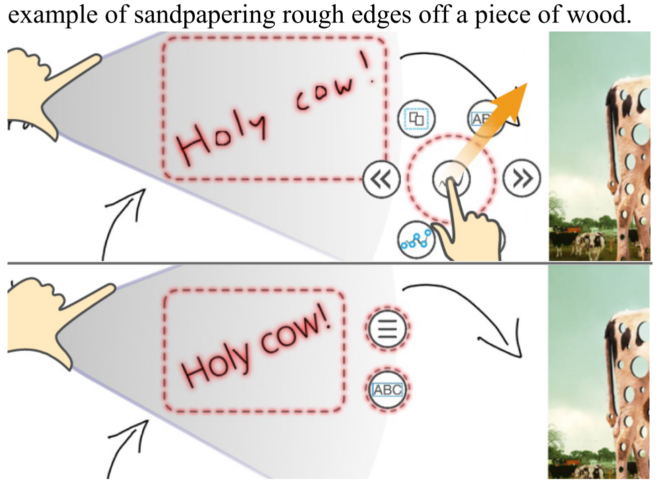

Figure 2. The *Zoom-Catcher* indicates an area of the canvas via the nonpreferred hand (Top). The user can then act on it with the preferred hand, such as to recognize handwriting (Bottom)—including position, point size, and orientation.

Next, we unpack the Zoom-Catcher in more depth, which will allow us to contrast its properties with Related Work.

ZOOM-CATCHER: UNIFIED SCOPE, ACTION & ZOOM

The Zoom-Catcher offers a new way to select multiple objects with the nonpreferred hand through multitouch, using a *framing gesture* that indicates the area to select by the relative orientation of thumb and forefinger. 'Framing' content in this manner is a naturally-occurring human behavior to focus attention and reference annotations to a portion of a page [30]. Yet to our knowledge, leveraging both the position and orientation of the off-hand to specify the scope of a selection on a large electronic whiteboard is novel.

While framing is a natural behavior, it posed an interesting design challenge because at the outset it seemed likely to conflict with pinch-to-zoom, which of course is also a two-finger gesture. But as we ultimately discovered, reconciling framing with pinch-to-zoom affords a novel unification, such that the Zoom-Catcher co-exists—and indeed is continuous with—pinch-to-zoom. The twist is that both *zoom* and *select* interpretations of the gesture exist simultaneously as the interaction gets underway. At any subtle hesitation while zooming, selection feedback starts to fade in. Users encounter this while navigating, and quickly grasp that it affords selection, thus leading them to discover this capability without further explanation. Or they can ignore it, and continue to zoom without difficulty.

That is the high-level view. But the Zoom-Catcher is the result of careful iterative design that puts several desirable properties into play all at once, in a way not fully realized before. We can unpack these properties (P0–P12) as follows:

P0. ***Grounded in Inking as the neutral state.*** By default, the pen lays ink on the canvas, making writing and sketching the foundation of the creative experience.

Yet, when user brings *the nonpreferred hand* to task, our system unifies selection, action, and zoom in a way that:

P1. ***Leverages a natural human gesture***, two-finger framing, to focus attention on an area [30]; which in turn

P2. ***Dovetails with pinch-to-zoom***, its electronic heir-apparent, such that framing and zooming interpretations of the gesture are *simultaneously* active; and

P3. ***Self-reveals by piggybacking on a familiar gesture,*** which leads users to discover the Zoom-Catcher while navigating the canvas; and by this unification also

P4. ***Affords multi-scale interaction,*** which allows users to access large, small, or out-of-reach areas alike.

These approachable transactions get users started without explanation. The Zoom-Catcher then reveals its presence and affords interaction with select content via:

P5. ***Fading in a semi-transparent palette***, which resembles a shadowy projection of the cleft between thumb and forefinger, and floats above the canvas; and thus

P6. ***Illuminates objects in an* orientable *canvas region***, making them 'pop' from the surround, and resulting in

P7. ***A lightweight scoping tool*** with clear feedback of the objects selected.

These attributes put the scaffolding in place for the higher-level actions we set out to support. The Zoom-Catcher then:

P8. ***Evokes commands close at-hand***, yielding location-independent interaction [51], such that the interface

P9. ***Reveals only context-appropriate functions***, with multiple radial menus that support further interesting functionalities (as discussed later in this paper); and also

P10. ***Provides a nonpreferred-hand, spatial mode switch*** that is spring-loaded by muscular tension; wherein

P11. ***An input filter acts on strokes from the stylus***, or subsequent preferred-hand touches, in a 'gesturing through the looking glass' manner [7]; and finally

P12. ***Always returns to the neutral state on release of the nonpreferred hand,*** such that the entire spring-loaded transaction—despite its richness—comes full circle, and the user is back to the primary task of design-sketching.

Some aspects of these properties (P0-P12) do appear in previous systems. However, the Zoom-Catcher is the first technique to articulate and bring together all of these benefits for fluid pen+touch interaction on electronic whiteboards.

RELATED WORK

WritLarge is a creativity support tool that draws from interaction techniques for ink, and large displays. Framing as a gesture to indicate scope—as well as integrated commands, modes, selection, and action—are all important ingredients.

Creativity Support Tools for Design

A key challenge for design and creativity support tools lies in balancing the rapid capture of ideas at the speed of thought, versus the ability to explore and refine those ideas once captured [10,26,61,63]. The former is well-suited to analog, freeform techniques such as writing and sketching with a stylus. The latter often involves adding structure, or semantics, as reinterpretations of the underlying freeform ink strokes and reference images [26,58,61]. What is needed are multiple interpretation-rich representations that allow one to back up, reinterpret moves, and shift directions [23,54,71]. Yet added structure (i.e. spatial rearrangement, recognition) can be harmful [39,62] unless "smart but silent" [34]—that is, applied selectively, and fully under the user's control.

Interaction Techniques for Ink and Large Displays

Electronic whiteboards often transform digital ink, such as to group, list, or tabulate. Avoiding fixed locations for interface elements is a critical issue often tied up with mode switching, gestures, and the interface metaphor itself [4,48,51].

Tivoli [46] organizes ink via lasso selection, long a go-to method for ink [12]. Yet drawing lassos [21,45] is tedious for large or hard-to-reach areas. Harpoon's velocity-sensitive area cursors offer one strategy to address this [41]. Automatic clustering of strokes, as in Suggero [43] and cLuster [49], offer another. But all of these approaches still rely on the pen, and thus the preferred hand, to select objects. This reliance on the pen serializes SELECTION: it adds an extra step that can't overlap with ACTION. By contrast, our approach of *framing with the non-preferred hand* unifies selection and action into the same continuous workflow.

However, we must also point out that the Zoom-Catcher is not intended for complex selections with low target cohesiveness; this is an intentional trade-off that optimizes for the common case of straightforward selections in our whiteboard scenarios. The Zoom-Catcher thus affords an intermediate level of control between lassoing—which is slow—and tapping on objects—which relies on the system to properly group ink marks. Our way of indicating general areas via multi-touch thus complements other approaches.

Musink [65] is an ink-based tool for music composers that has many synergies with our work, particularly in the way it lets users specify multiple levels of interpretation, and how it uses multiple representations (to move interaction between digital-paper and computer). Another strategy is *translucent patches* [35] that define persistent areas, via lasso selection, that can layer recognition behaviors. Flatland [47] similarly uses 'segments' that expand as the user writes, but this requires users to be constantly aware of (and potentially distracted by) how their strokes are being segmented. NotePals [15] supports sharing notes to a group, while TEAM STORM [23] affords working with multiple ideas in parallel across whiteboard and tablet. *WritLarge* supports similar capabilities but shows how selecting and working with portions of an electronic canvas might be made more fluent by combining pen and touch, with both right hand and left.

The Spotlight [32] illuminates large areas (P6) from a distance, but does *not* select, nor integrate subsequent actions; it focuses the attention of others on an area. Drag-and-pop highlights a distant target [3], but it is an indirect way of pointing—not a selection method. The Vacuum explores selection and manipulation of remote objects [5], but does not combine multiple-object selection with action. And while various refinements of zooming have been suggested (e.g. [2,18,50]), none dovetail with selection. Indeed, the Zoom-Catcher sidesteps many issues of action at-a-distance, or indication of large areas, through integration of pan and zoom continuously within selection itself.

Uses of Multi-Touch, and Framing, to Indicate Scope

People manipulate paper with the nonpreferred hand while writing [22]. Framing the work-area between thumb and index finger is a natural "behavior to focus attention" [30] on part of the page. VIDEOPLACE [36] and ActiveDesk [9,13] offered early examples of grasping an object in this manner.

In Hands-on-Math [72], users frame part of a page to fold it; the nonpreferred hand indicates scope for *one* at-hand action, *fold*. GatherReader [28] also uses framing to select part of a page—partly echoing some of our design properties (P5,P6 and P10-12), but critically not the ability to indicate scope in an *orientable* manner (P7) to allow coming at a desired selection from different suitable angles, nor evocation of commands (P8-9). The eTab [6] acts as an at-hand input filter that reveals context-appropriate commands (P8-11), but requires use of an additional tangible device on top of a tablet. Our approach extends and generalizes such benefits through a novel unification with pinch-to-zoom (P2-P4).

Pen+touch [22] uses a framing gesture with the nonpreferred hand to hold down a *single object*, such as a photo. To include multiple objects, the user must tap-select them in sequence or resort to lasso selection. By contrast *WritLarge* is consistent with *pen+touch = new tools* but shows a much richer way of indicating a scope containing multiple objects.

Integration of Commands, Modes, Selection, and Action

ToolGlass [7] provides a mobile palette, manipulated via trackball and nonpreferred hand. The preferred hand then 'clicks through' a tool to apply it to the object below. This integrates the two actions. However, the 'scope' is limited to a click—a single point—rather than area-selection via multi-touch. And the inputs are indirect—with trackball and mouse—as opposed to direct—with pen and touch on the display itself. T3 extends the toolglass paradigm to pen input, with a puck for the nonpreferred hand, and even a thumb-wheel for interleaving zoom with other interactions [38]; indeed, T3's use of bimanual input and its holistic approach to designing interactions are foundational to our work. However, it remains unclear how to achieve this kind of fluidity with direct touch (and pen), as opposed to T3's indirect setting using desktop digitizers without touch.

A single pen gesture can combine multiple actions, including selection, action, and direct manipulation [1,37]. Gestures convey an additional benefit when they select an object *and* evoke an action (P8) close at-hand [7,37,51]. However, overburdening a pen stroke complicates gestures and may require a *delimiter* [27] as an extra step. Furthermore, because a single modality (the pen) provides all of these functions, mode-switching also becomes necessary [42,60].

Holding a spring-loaded mode with the non-preferred hand [55,60,64] is a fast way to switch modes [40,42]. While this approach can scale to more than one mode [29,57], and to multi-touch [24,67,68], the literature offers few examples of techniques that combine modes with selection, action, manipulation—and now zoom. This consideration hints at a deeper concern, namely that 'selection' is oft-treated as a fundamental operation, with its own mode. For example, Foley et al. [19] list *Select* as one of six elemental tasks. But selection is a sub-task: it comprises only part of the user's task-flow, a prelude to *action* on the indicated scope. The higher-level task is a compound one of *selection-action*.

Our objective is to obviate 'selection' as a separate mode, and extra step, by integrating this sub-task into nonpreferred-hand mode switching *and* the performance of the action itself. The result is a continuous motor-sensory act that takes place at this higher conceptual level via the Zoom-Catcher.

This resonates with chunking [11], and phrasing of selection-action [7,27,38], for pen+touch in particular [8,24,28,30,44]. We show how this can foster a new approach to creativity support on electronic whiteboards, and in particular our analysis articulates numerous desirable properties (P0-P12) that we then proceed to integrate under a common umbrella.

WRITLARGE: FREEFORM CREATIVITY SUPPORT

To explore the potential of these directions we implemented a creativity support tool that runs on a large electronic whiteboard, the 84" Microsoft Surface Hub, which supports both pen and multi-touch. The same application runs on the Surface Pro 4 tablet, which also has pen+touch.

Overall UI Architecture and Metaphor

In accordance with our desire to provide a freeform creativity support tool for early-stage design, ideation, and informal discussions, inking is the center of the experience. By default *WritLarge* simply acts as a whiteboard, where bringing the pen to the screen leaves ink. The user is free to sketch, mark-up, and handwrite however desired without worrying about accidentally triggering pen gestures or recognition features.

The basic unit of work in *WritLarge* is a drawing canvas. The user can pan and zoom the canvas, but canvases are delineated into individual screens. The metaphor is thus similar to a large paper flip-chart, with subsequent sheets (canvases) arranged horizontally. If the user zooms out far enough, the series becomes visible, in a slide-sorter type of view. This allows the user to quickly grab a clean whiteboard and storyboard a sequence of ideas. Canvases themselves are selectable with the Zoom-Catcher and may thereby be duplicated or rearranged through direct-manipulation. A horizontal swipe flips through the sequence of 'charts.'

The application has no permanently visible UI. The entire screen is devoted to the user's content on the canvas. The canvas supports ink strokes as well as freeform layout of reference images. All actions arise from the canvas itself, through the framing gesture and evocation of the Zoom-Catcher. We do not use any barrel-buttons on the pen. To act on ink strokes and other objects, the user brings the Zoom-Catcher over them with the nonpreferred hand, and gestures through the semi-transparent palette or its associated radial menus. Because *WritLarge* achieves this solely using pinch-to-zoom, rather than relying on an out-of-band gesture or button press, our approach has potential utility for many other applications. Also, as a proof-of-concept for small-group design sessions, a single tablet (also running *WritLarge)* can pair and share selected content with the Hub.

Implementation

WritLarge is a Universal Windows App (UWA) built on Windows 10 with (simultaneous) pen and multi-touch reported through standard system events. We use the built-in Windows handwriting recognition components since such algorithms are not our focus. As such, our recognition results of course aren't perfect—but the user may easily back out of them by reverting to ink strokes. Graphics effects and feedback are rendered using the Win2D SDK. The Surface Hub and Surface Pro 4 communicate over wireless TCP.

DESIGN OF THE ZOOM-CATCHER

So far we've discussed a number of properties exhibited by the Zoom-Catcher, but we have not yet detailed its design. Of particular importance is the simultaneity of selection and zooming, and some of the considerations and alternatives we iteratively worked through to arrive at this design.

As a starting point, we envisioned transactions such as the *Holy cow!* of Fig. 2. But this raised two key design questions:

1. *What is the scope* of strokes to recognize, or re-structure?
2. *When is recognition triggered*—and how can the user indicated other desired shifts between representations?

To bring good answers to these questions and make the resulting transactions fast and fluent for users, we started exploring ways to use the non-preferred hand to select areas in a lightweight way, and particularly via two-finger framing [30]. We realized immediately this would collide with pinch-to-zoom, but our objective was to first see how the interaction felt (through informal pilot tests and expert opinion), and determine whether it was even worth pursuing.

Dwell Time to Distinguish Selection?

One of the first things we experimented with was triggering selection based on dwell time. Touching down and holding the framing gesture in place for a fixed timeout triggered selection. But, after trying several different time-outs ranging from 1500, 1000, 750, 350, or even 150ms we arrived at the conclusion that any choice of time-out interval would raise problems. The very short time-outs tended to trigger by accident while the user was thinking, or fine-tuning a pinch-to-zoom—and yet even then, they slowed down interaction when selection was desired. But longer time-outs were annoying because they slowed selection to a crawl—echoing previous results showing that dwell time is an inefficient mode-switching technique [42]. Not satisfied, we kept searching for other approaches.

Two Fingers to Zoom, Three Fingers to Select?

We also tried making pinch-to-zoom vs. selection a predictable and learnable manual skill, based on the number of fingers brought to the canvas. We reserved two fingers for zoom, and instead used three fingers to switch to area-based selection. This avoided the ambiguity of a time-out, but switching between two- and three-finger postures stifled the interaction in other ways. It still wasn't very fast. It didn't feel comfortable. And it lost much of the natural appeal of framing with thumb and forefinger. After a pilot test with four users, one participant summed it up as follows: "*I don't like [three finger framing], because I have to always remember three finger is framing, and sometimes I do use three fingers to drag the canvas or zoom it. I also don't like that I always have to keep all the fingers down to control the region. It felt pretty stiff to rotate with three fingers.*"

Speculative Execution of both Zooming and Selection

Eventually we realized that pinch-to-zoom and selection aren't necessarily mutually incompatible, so we asked: *Why not perform both?* This is reminiscent of *speculative execution* in pipelined microprocessor design, as well as handling multiple inputs with uncertainty [59]. This insight led us to consider a design where the interface proceeds with *both* selection *and* zooming interpretations of a two-finger gesture, simultaneously.

At first, the pinch-to-zoom interpretation of two-finger gestures dominates selection: panning and zooming responds immediately, and initially no selection feedback is shown. That is, the "semi-transparent palette" of the Zoom-Catcher, per property (P5), is still fully faded out, and invisible. But if the user hesitates while zooming, we start to fade in *only the illumination of the selected objects (P6)*—the highlighting feedback—for all objects contained fully within the selection region, in a nuanced and understated way. And if pinch-zoom motion resumes, all selection feedback fades out fully.

But not until the user holds the framing gesture stationary—defined as less than 10mm of finger movement over a 500ms window—does the selection feedback (and radial menus) fade in fully. At this point the Zoom-Catcher "latches in" and the semi-transparent palette of (P5) appears as in Figure 2.

However, critically, the user does *not* have to wait 500ms to act on the highlighted objects. Since the radial menus start fading in with the object highlighting, the user can short-circuit the fade-in transition by touching down on a menu immediately—or by grabbing the objects, such as to drag them (as a unit) to a new position. Indeed, from the instant two fingers touch down on the canvas, the invisible 'selection area' is active, and can be acted upon by the user.

As users acquire skill with the technique, they can learn to anticipate this, and interact without waiting for the timeout.

Once the Zoom-Catcher latches in, the pan/zoom interpretation is no longer active and the user can move, reorient, and adjust the angle subtended by the Zoom-Catcher without disturbing the canvas. To return to pan/zoom, the user must lift their nonpreferred hand—which dismisses the Zoom-Catcher—and then reengage.

This approach allows zooming to be freely combined with selection highlighting (P2,P4) until the Zoom-Catcher latches in. The user can zoom in close to select a single ink stroke (Figure 3a), or zoom far away to select large areas or objects that would otherwise be out-of-reach (Figure 3b).

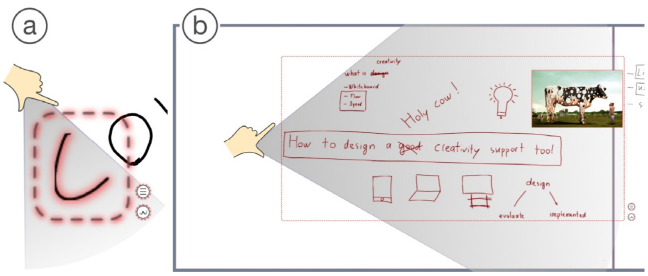

Figure 3. Unification of multi-scale selection with the Zoom-Catcher allows the user to access (a) small objects by zooming in, and (b) large or out-of-reach areas by zooming out.

Nuances of the Selection Feedback

We carefully designed the visualization of the semi-transparent palette to foster its perception as a natural extension of the hand. Rather than a sharp-angled cone for the selection area (e.g. as used by the ActiveDesk [9,13]), we give it a soft shape that echoes the cleft between thumb and forefinger (P5)—thereby fitting into the hand, so that it feels like a projection of one's reach. We also use this rounded area at the apex of the palette as a place where the user can tuck away objects and carry them 'in the fold' of the thumb-forefinger cleft. This serves as a temporary clipboard known as *The Fold*, discussed in more detail later.

The palette is rendered with a semi-transparent gradient that trails off with the distance from the apex, much as the light cast by a flashlight beam fades away. This fits the "illumination" metaphor (P6) were trying to establish and also lends the selection area a soft-edged feel, in keeping with the informal, relaxed use-cases of the application itself.

To highlight objects in the selection, we render them with a bright-red outer glow. We also show a dashed-line roundtangle around all the selected objects, with two radial menus at the lower right. We render all these objects *on top of* the semi-transparent palette so that they visually pop (in contrast to other objects on the canvas that may be nearby, but not included in the selection).

State Machine for the Zoom-Catcher

We implement the Zoom-Catcher as a simple state machine (Figure 4). The interaction starts in the *Neutral* state, with zero or one finger on the display, and transitions to *Speculating*—with both pinch-to-zoom and selection active—as soon as a second finger touches down. When selection feedback starts to fade in during the *Speculating* state, it does so proportional to the system's confidence in the degree of movement vs. hesitation. Once the user holds the framing posture stable for long enough—or the user short-circuits the fade-in animation by engaging directly with the selected area, or the radial menus—the *Framing* interpretation latches in.

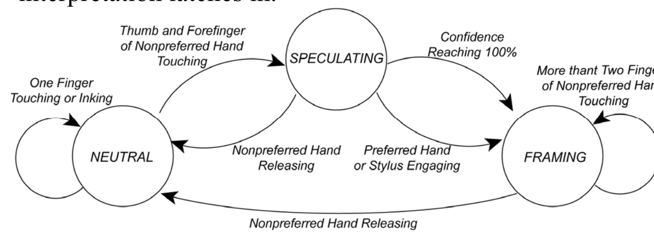

Figure 4. State machine for the Zoom-Catcher. While 'speculating,' both pinch-to-zoom and the proportional fading of selection feedback are simultaneously active.

Flexibility of Posture and Adjustment of the Framing Gesture

Note that in the *Speculating* and *Framing* states, more than two fingers can rest on the canvas if desired. This relaxes the posture required, and makes the gesture more flexible, forgiving, and natural because it doesn't depend on a specific number of fingers [70]. In our own use we find this helps to reduce arm fatigue because the whole hand can rest comfortably on the screen. This flexibility applies to zooming, selection, and the Zoom-Catcher itself.

Adjusting the Selection Area

By re-orienting thumb and forefinger the user can steer the Zoom-Catcher around, such as to come at a cluttered area from another angle where it is easier to select only the objects desired (e.g. Figure 3a). The distance between fingers also controls the angle subtended by the Zoom-Catcher, making it easy to cast it wide, or focus it into a narrow beam, depending on the surround of other objects nearby. Currently, we only support right-handers, so the Zoom-Catcher initially projects its selection region to the right. We plan to add a setting for left-handed users in the near future.

By default the Zoom-Catcher extends 1/5 of the way across the canvas. If it is necessary to adjust this, the user can simply swipe the Zoom-Catcher with the preferred hand to directly adjust its extent. The user just touches down and drags the semi-transparent palette to modify it (Figure 5).

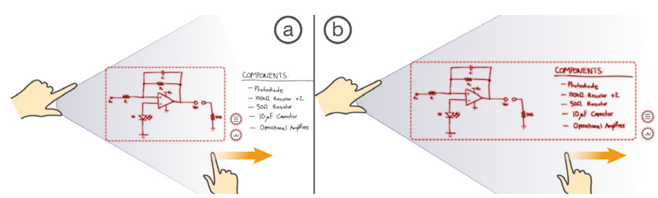

Figure 5. Modifying the extent of the selection (a) via a bimanual swipe with the preferred hand (b).

Framing Region as Mode Switch and Input Filter

The framing region also serves as a nonpreferred-hand mode switch (P10) that allows the user to gesture-through the semi-transparent palette, thus integrating a further interesting capability into this simple hand posture. Pen strokes within the palette are interpreted as a gesture, rather than leaving ink. For example, the user can lasso specific ink strokes as a fallback for refining the selection in tricky cases (e.g. for dense areas of the canvas). Likewise, touches within this region are treated differently. For example, the user can drag the selected ink strokes to a new position on the canvas, or dab a finger on an image to blend its colors. Hence, the framing region doubles as an input filter (P11) that modifies the interpretation of pen or touch gestures in this area.

Evoking Context-Appropriate Commands Close At-Hand

To support transition to ACTION, the Zoom-Catcher evokes commands at-hand (P8). In addition to gesturing-through the palette (such as lasso selection with the pen), two radial menus that reveal only context-appropriate functions (P9) fade in with the selection, and appear at its lower right. These are the *Actions Menu* (Fig. 6, top) and the *Representations Menu* (Fig. 6, bottom). Note the user need not wait for these menus to fully fade-in before accessing their functions.

The *Actions Menu* includes four commands: *Duplicate*, *Share*, *Search*, and *The Fold*. *Duplicate* allows a downward swipe on the menu to copy the selection and place it in a single uninterrupted movement [27,37,53]. *Search* performs a web image search from the selected phrase, while *Share* allows passing select content to a collaborator via a paired (co-located) tablet. We'll further detail *The Fold* shortly.

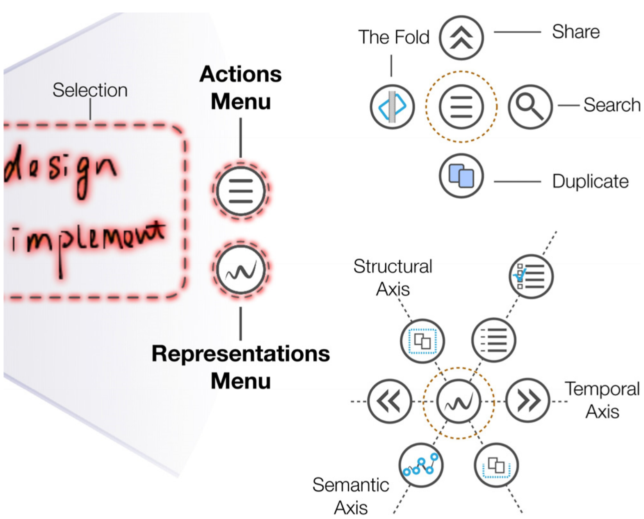

Figure 6. Exploded view of radial menus on selection. (a) The *Actions Menu* (top) always contains generic *Share, Search, Duplicate,* and *Fold* commands. (b) the *Representations Menu* (bottom) provides spatial, temporal, and semantic axes of movement defined as conceptual moves relative to the content.

The *Representations Menu* is also radial. It encapsulates three conceptual axes of movement: *semantic*, *structural*, and *temporal*. These conceptual axes offer a key contribution because they afford facile re-interpretation of the selected objects in a variety of ways, including 'un-interpreting' moves in a manner analogous to a local undo operation. As we will show, this richness of expression complements the at-hand and in-context nature of the Zoom-Catcher, and thereby demonstrates some benefits of more holistically phrasing together *selection* and *action* in *WritLarge*.

The Fold: An At-Hand Place to Tuck Things Away

"The Fold" is a special area at the apex of the Zoom-Catcher's semi-transparent palette where the user can tuck objects for later use. This serves as a lightweight clipboard for multiple objects. But unlike Fix-and-Float [56], the Attic [16], Toolspaces [52], or Pocket [28]—all of which require stashing objects at a particular area of the screen, such as the bottom edge—the Fold is always at hand, and therefore supports this capability in a location-independent manner.

From the Actions Menu, Selecting *The Fold* animates the selection into the curved apex of the Zoom-Catcher. When the user later makes the framing gesture again, any items in the Fold are visible and can be dragged out one-by-one. This is convenient for transporting objects long distances, or across canvases. And by zooming out, entire canvases (even multiple canvases) can be selected and pulled into the Fold, which makes it easy to storyboard alternative sequences.

Summary of the Zoom-Catcher

As we have seen, the Zoom-Catcher integrates quite a few capabilities, but it is important not to lose sight that from the user's perspective, it is astonishingly simple. One simply brings thumb and forefinger to screen and nearby objects light up. Its design carefully integrates numerous desirable properties (P0-P12). Yet by tight integration and continuity of action, they all essentially collapse into a single step that follows from the familiar gesture of pinch-to-zoom.

With this foundation firmly in place, we now show some interesting ways that *WritLarge* leverages this powerful scoping tool. In particular, it affords a user experience with an unusually flexible notion of moves between multiple representations, arranged along three conceptual axes of movement: semantic, structural, and temporal.

REPRESENTATIONAL AXES OF MOVEMENT

We argued earlier that creativity support tools can benefit from flexible moves among multiple interpretation-rich representations. This affords the back-and-forth re-interpretation typical of many design, creation, and collaborative ideation scenarios [26,58,61,63] where one wishes to nimbly explore a variety of alternatives—or multiple ideas in parallel [23,54,71].

WritLarge achieves this through *three conceptual axes of movement* that are organized into *levels*. These axes include:

- **Semantic**: the *meaning* that the system ascribes to the selected objects. The default semantic, of course, is that of the ink stroke. But the semantics can be elevated to recognize text, or to identify list structure—as well as drop down a level, to the individual points of the stroke.
- **Structural**: the spatial arrangement of the objects, which can be freeform (i.e. no structure), grouped, or arranged in a grid. Objects in a structure revert to their original arrangement if the user drops down to the freeform level.

- **Temporal**: the time-ordering of the objects, which allows scrubbing back and forth in time to see (for example) the selected ink strokes in a previous state of completion.

Each of these axes may offer multiple gradations of representation. The user can progressively add formality and structure to a representation by moving ***up a level***, or revert to a less-structured representation by moving ***down a level*** in the opposite direction. Hence, adding different types of structure takes on a spatial metaphor that is well-suited to "the intelligent use of space" [33], and it furthermore makes the notion of reversing course intuitive by expressing this as a simple movement in the opposite direction.

While one can envision a very wide range of functionalities along these three axes, as a technology probe [31] and proof of concept, *WritLarge* at present explores a small number of levels for each axis. However, exercising restraint on the number of operations possible also helps to afford our simple spatial arrangement of these features into linear axes that offer just a few levels. While this approach might not scale to a much larger number of representations, going too far in terms of adding features and complexity would move *WritLarge* away from being the freeform creativity support tool that it is, and closer to a full-blown "computer program" and the burdens of formality [62] that tend to come along with that. As such, we chose to favor keeping things simple.

Representations as 'Patches'
The user can gradually instill or remove representations from content on the canvas. Moving up a representational level creates a new type of object, which is inserted on the canvas in place of the lower-level objects. These behave much like *patches* [35,47]: they filter input events, allowing objects to interpret pen and touch inputs in a way that suits the data type involved. This adds to the richness of techniques available.

The Semantic Axis
The semantic axis of movement is arranged diagonally on the Representations Menu from lower left, to upper right. It supports semantics at the level of points, ink strokes, recognized text, lists, checkboxes, diagram nodes, and tables.

Note that the levels available depend on the selection. Reference images, for example, currently only support two levels: the image itself, or dropping down a semantic level to the pixels, which allows a small set of image-editing operations. And for ink strokes, the *list* semantic is only available if the standard Windows handwriting recognizer can extract a list structure from the selected strokes. Likewise, the *diagram node* semantic only applies to handwriting that is circled or surrounded by a hand-drawn box. As stated earlier, our focus is not on the recognition technology itself; that is not our contribution here. Rather, it is the fluid and simple way in which we can surface this variety of capabilities on the scope indicated by the user.

From Strokes to Recognized Text
Zoom-catching a short handwritten phrase and moving up one semantic level leads to our early motivating *Holy cow!* example (Fig. 2) that recognizes not only the text itself, but also preserves the position, point size, and orientation of handwritten strokes. But by selecting the recognized text and dropping down a semantic level, the user can revert to the original ink strokes to make corrections or add a few words. This reversal of representations—essentially, a localized undo function [25] that operates only on the selection—is extremely simple to express in *WritLarge*, yet demonstrates a capability rarely encountered in inking applications.

From List to Diagram
If the recognizer detects a list structure in the zoom-caught strokes, the list representation becomes available on the semantic axis. The system provides feedback that this semantic is available by changing the icon at the center of the Representations Menu. Moving up a semantic level then recognizes the text and formats it as a bulleted list. The user can drag individual list items around with a finger to re-order them, which shows an example of how the 'patch' can filter input events. Similarly, if the user moves up one more level to the checkbox representation, a pen stroke can check off items as the user completes them, for example.

From Hand-drawn Grids, to Tables, and Back Again
WritLarge also includes a simple heuristic for recognizing a hand-drawn grid of rows and columns as a formatted table. If the selection includes such a grid, the table representation becomes available along the semantic axis and the user can move up a level to recognize it. Strokes drawn across the formatted table add new rows or columns. But rather than offering a complex UI for merging cells, for example, the user can instead drop down to the semantic level of the points making up the ink strokes, use the pen's eraser to remove divisions, and then move back up to the table level to re-recognize it (Fig. 7). Handwriting on top of a table cell is automatically recognized as text.

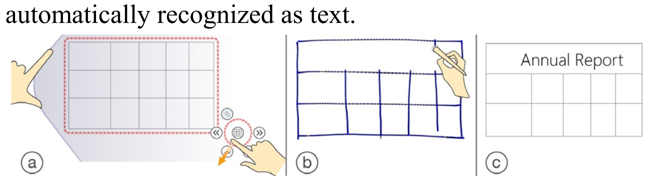

Figure 7. (a) Moving a recognized table down to the points level (b) allows editing the strokes, such as to merge cells. (c) Handwriting on tables is immediately recognized as text.

The Structural Axis
By default, objects such as ink strokes or reference images are treated at the *freeform* level—that is, without any inherent structure. Moving up the structural axis affords a number of spatial rearrangements of the selected objects.

Grouping Objects and Arranging in Grids
Moving up one structural level groups the selected objects. This does not change their visual appearance, but allows them to subsequently be directly manipulated as a unit. Swiping one further level up from the group representation rearranges the objects in a fixed grid (Fig 8). Once arranged in the grid, the user can drag items between grid positions to re-order them. Pen strokes drawn on the grid snap to the grid positions as well. Dropping back down to the freeform level

reverts the objects to their original spatial arrangement. And of course, the user can intermingle semantic and structural shifts of representation to achieve various effects (see video).

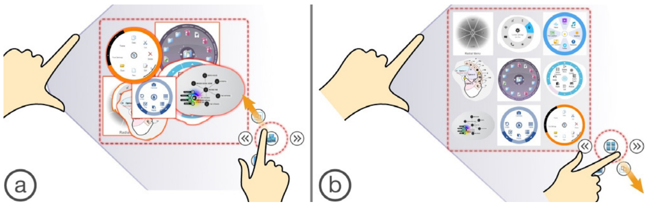

Figure 8. Structural axis. (a) A freeform pile of images can be reorganized into a grid layout by moving up the structural axis. (b) Dropping back down one level of structure reverts to the original freeform layout.

The Temporal Axis

The temporal axis is easy to explain but rich in capability. It provides an easily expressed notion of pulling a particular area of the canvas back in time, essentially serving as a local undo function, scoped to a region readily indicated by the Zoom-Catcher. The user simply scrubs forward and back—left and right from the Representations Menu—to roll back their ideas in time (Fig. 9). Note that this is articulated through continuous sliding, and does not require repetitive swiping motions to move further along the temporal axis.

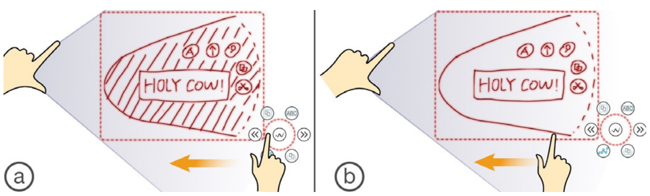

Figure 9. Temporal axis. (a) sliding to the left (b) rolls *only* the content in scope back to a previous state.

PRELIMINARY EVALUATION WITH TEST USERS

We conducted an informal evaluation to gain feedback from participants about our interaction techniques. We were particularly interested in how users would discover and use the Zoom-Catcher, their reactions to the four conceptual axes of movement, and whether users would feel that *WritLarge* affords a continuous and fluid flow of ink-based creation.

Participants and Procedure

We recruited 8 participants (4 female) aged between 24-32 years. All had used touch devices for at least 3 years. Before being introduced to the interaction techniques, participants were asked to briefly explain their ongoing work to the experimenter by inking using *WritLarge* on an 84" Microsoft Surface Hub. This allowed us to introduce the interaction techniques with participant-created content. The study focused on the use of the Zoom-Catcher, as well as the *semantic* and *structural* axes, as we felt those were the most provocative—and in need of user feedback. After trying each technique, participants filled out a questionnaire, and responded to a 10-15 minute interview. The study took ~1hr.

Results

Results are reported in a 7-point Likert scale with 1 labelled "strongly disagree" and 7 labelled "strongly agree".

Discoverability of Zoom-Catcher

As the Zoom-Catcher layers a new interpretation onto pinch-to-zoom, we sought to explore whether participants would discover this. Once participants finished inking on the canvas, they were asked to manipulate the canvas with their hands. All participants were able to activate and discover the Zoom-Catcher, as visual feedback starts to fade in when more than one finger touches the screen. All participants were able to immediately associate the interaction's purpose with selection because of the visual feedback.

At this point, participant behavior diverged. Five participants continued moving their fingers to manipulate the scope, as our design intended. The other three released their fingers, thinking this would lock in the selection, and then tried to touch the screen again. This suggests that adding a way to pin the selection in place—perhaps by pressing harder [20]—would be interesting to try. Furthermore, 6/8 participants discovered that they could directly manipulate the selected content with their preferred hand, such as to drag ink strokes to a new location on the canvas. Overall, this validates that piggybacking the Zoom-Catcher on "framing" and the familiar pinch-to-zoom gesture (as in properties P1-P3) afforded self-revelation of its core capabilities.

Scope-to-Select with Zoom-Catcher

Participants responded very positively to using the framing gesture to select content. They found the technique easy to learn (four 7/7 ratings, the rest 6/7). Participants responded to "I could use this technique (framing to select content) to quickly select the content I wanted" with four 7/7, three 6/7, and one 5/7. Participants acknowledged that "*most of the time you don't need the granularity of a lasso to select objects* and *forcing people to go around and do the lasso is just painful.*" With Zoom-Catcher, they only had to "*put the two fingers there and just get it*." The advantages of zoom-catching over lasso selection was most salient when selecting a large area: "*I just kind of shrink it, zoom it out, and then select it.*" This further validates that our design successfully allowed the participants to dovetail pinch-to-zoom with selection as intended; users perceived no conflict.

Participants also reported the *"arc-radius model"* allowed "*an intuitive feel how it's gonna appear, even without explicitly knowing beforehand and seeing it on the screen.*" This shows that users could successfully anticipate the selection region, and plan their actions accordingly, even before the selection feedback fully faded in. Participants also felt that, despite its simplicity, the Zoom-Catcher covered most of their selection needs well. They also appreciated our inclusion of lasso selection as a fallback for difficult cases.

Semantic and Structural Axes

Participants found changing representations by moving along the conceptual axes easy to learn. They rated both the semantic axis (one 7/7, six 6/7, and one 5/7) and the structural axis (four 7/7, three 6/7, and one 5/7) fairly highly. Participants liked the ability to easily reinterpret the content they created: "*I like that you can easily convert analog things* *to digital, and that the objects can have different levels of abstraction*." They immediate saw the benefits of using the axes to "*prepare presentation*, *brainstorm*, *and help my design*." As one participant described, "*sometimes when things are digitized*, *I would like to keep a copy of physical one, so that I can go back to the raw sketches and add more things to it*. *Your tool really has the potential to just let me to going back and forth from analog and digital*."

The movement and feedback of the axes was also easy to understand; as one participant noted "*It's very intuitive. You don't even need to explain them to me. Are they standard?*" Participants who had used similar features in other applications found spatially organizing these capabilities in the axes "*makes much more sense than listing the commands out, as the axis movement matches what I have in mind."*

Participants found the concept of axes was "*extensible and universal."* As one participant noted, *"any idea, essentially, will start from something scratch and simple, and then you refine it and build upon it."* Several participants felt that these capabilities held great promise, and encouraged further development of the concept, as they believed one "*can go very very far with it."* But several participants did suggested showing *all* the levels available on each of the axes to make it easier to move directly to the level they wanted, especially "*if there are many levels associated with the selected content."* Since the radial menu currently only shows adjacent levels, not all of them, the design could clearly benefit from enhancing the visibility of any additional levels.

Fluid Workflow and Maintenance of User Control

The overall workflow of the *WritLarge* application was well-received. Participants found framing with one hand, and executing actions with the other, "*played together very well*" and allowed working "*in a really fast way*; *it feels very natural and fast that I can use both of my hands; you don't even need to think about it*." Participants also appreciated that the interface allowed them to "*always focus on sketching stuff"* by giving them full control of when and how the content should shift representations—if at all—making users feel "*more productive"* in the flow of creativity.

DISCUSSION

We have argued that to elevate the fluidity of whiteboard interaction, SELECTION should *not* be considered as an elemental task [19] that is a separate step (or mode, or tool), but rather should be regarded as part of a continuous "chunking" of manual activity that specifies a higher-level command—the ACTION. And, while we believe *WritLarge* provides a convincing demonstration this possibility, that is not to say that techniques cannot go further still in the quest for full unification as SELECTION-ACTION phrases.

One key issue is the manner in which the Zoom-Catcher emerges from the canvas, via speculative execution of both *select* and *zoom* interpretations of two-fingered gestures. Earlier, we motivated and justified this design in detail, and all evidence from our preliminary evaluation suggests it works extremely well. However, we have not yet established quantitatively how (or even *if*) this differs from a simple tap-and-hold gesture. We designed our fade-in such that expert users can short-circuit the animation, by immediately manipulating the selection, or targeting the radial menus even before they've fully faded in. But it is not yet clear to what extent users can take advantage of this to realize performance faster than a dwell time. Future work should probe this by subjecting variations of our technique to a "true cost" analysis of the mode-switching time [17,42].

A second issue is the way the Zoom-Catcher dovetails with zoom. Early on, one can adjust the zoom to afford multi-scale selection, but once the Zoom-Catcher latches in, zooming is no longer possible. It may be possible to enhance the fluidity of these transitions—in both directions—such as by using pressure, or perhaps other nuances of multi-touch.

Third, ideally the Zoom-Catcher should scale to a larger set of commands. We support a few gestures directly on the selection, but not as many as Pen+Touch [30], for example. And for most of our contextual actions, we resort to radial menus, which can only comfortably handle 8 commands. At present we have two such menus, and of course more could be added, but are there other approaches, such as those explored by the Hotbox [37] or HandMark [66]? And for that matter, although we believe having a small number of rich representations is a sweet spot in the design, we have not yet explored how far one might push this: how many more levels could be handled along these conceptually simple axes?

Fourth, and finally, it remains unclear how to adapt the Zoom-Catcher to multiple, collaborating users. Indeed, there may be opportunities to extend the representations we make available with a collaborative "social axis" as well, since social interaction plays such a key role in support of creativity [63]. Another key technical issue is identifying which user is touching the whiteboard, and with what hand, where wearable sensing (for example) may be helpful [69]. Also, zooming is a global mode that affects the entire canvas, so some way to handle zoom (such as a fisheye lens) that does not disrupt the other user's work is needed.

CONCLUSION AND FUTURE WORK

Taken as a whole, the design of *WritLarge* sought to open new vistas for freeform content on electronic whiteboards. The system achieves this in a unique way, using carefully crafted input techniques that afford unified scope, action, and zoom, with pen-plus-touch—and both hands—in natural and complementary roles. While much yet remains to be done to extend and prove out this approach, the results so far appear quite promising. Our hope is that these contributions can help to reduce the impedance mismatch between the creator and their strokes on the canvas, thus bringing the rich, actionable, and flexible representations of *ink unleashed* readily to hand.

RIGHTS FOR FIGURES

REFERENCES

1. Anand Agarawala and Ravin Balakrishnan. 2006. Keepin' it real: pushing the desktop metaphor with physics, piles and the pen. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*, 1283-1292. http://doi.acm.org/10.1145/1124772.1124965.
2. Jeff Avery, Mark Choi, Daniel Vogel, Edward Lank. 2014. Pinch-to-zoom-plus: an enhanced pinch-to-zoom that reduces clutching and panning. In *Proceedings of the 27th annual ACM symposium on User interface software and technology (UIST '14)*, 595-604. http://dx.doi.org/10.1145/2642918.2647352
3. P. Baudisch, E. Cutrell, D. Robbins, M. Czerwinski, P. Tandler, B. Bederson, A. Zierlinger. 2003. Drag-and-Pop and Drag-and-Pick: Techniques for Accessing Remote Screen Content on Touch- and Pen-operated Systems. In *Interact 2003*, 57-64.
4. Benjamin B. Bederson, James D. Hollan, Allison Druin, Jason Stewart, David Rogers, David Proft. 1996. Local tools: an alternative to tool palettes. In *Proceedings of the 9th annual ACM symposium on User interface software and technology (UIST '96)*, 169-170. http://dx.doi.org/10.1145/237091.237116.
5. Anastasia Bezerianos and Ravin Balakrishnan. 2005. The vacuum: facilitating the manipulation of distant objects. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '05)*, 361-370. http://dx.doi.org/10.1145/1054972.1055023.
6. Andrea Bianchi, So-Ryang Ban, Ian Oakley. 2015. Designing a Physical Aid to Support Active Reading on Tablets. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15)*, 699-708. http://dx.doi.org/10.1145/2702123.2702303.
7. Eric A. Bier, Maureen C. Stone, Ken Pier, William Buxton, Tony D. DeRose. 1993. Toolglass and Magic Lenses: The see-through interface. In *Proceedings of the 20th annual conference on Computer graphics and interactive techniques (SIGGRAPH '93)*, 73-80. http://dx.doi.org/10.1145/166117.166126
8. Peter Brandl, Clifton Forlines, Daniel Wigdor, Michael Haller, Chia Shen. 2008. Combining and measuring the benefits of bimanual pen and direct-touch interaction on horizontal interfaces. In *Proceedings of the working conference on Advanced visual interfaces (AVI '08)*, 154-61. http://dx.doi.org/10.1145/1385569.1385595.
9. B. Buxton. *The Active Desk*. Retrieved from: http://www.billbuxton.com/ActiveDesk.html.
10. Bill Buxton. 2007. Sketching User Experiences: Getting the Design Right and the Right Design. Morgan Kaufmann.
11. W. Buxton. 1986. Chunking and Phrasing and the Design of Human-Computer Dialogues. In *Proceedings of the IFIP World Computer Congress*, 475-480.
12. W. Buxton, E. Fiume, R. Hill, A. Lee, C. Woo. 1983. Continuous hand-gesture driven input. In *Proceedings of Graphics Interface '83*, 191-195.
13. William Buxton. 1997. Living in augmented reality: Ubiquitous Media and Reactive Environments, in *Video Mediated Communication*, K. Finn, Sellen, A., Wilber, S., Editor. Erlbaum, Hillsdale, NJ, 363-384.
14. M. Csikszentmihalyi. 1991. Flow: The Psychology of Optimal Experience. HarperCollins.
15. Richard C. Davis, James A. Landay, Victor Chen, Jonathan Huang, Rebecca B. Lee, Frances C. Li, James Lin, III Charles B. Morrey, Ben Schleimer, Morgan N. Price, Bill N. Schilit. 1999. NotePals: lightweight note sharing by the group, for the group. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems (CHI '99)*, 338-345. http://dx.doi.org/10.1145/302979.303107.
16. Steven J. DeRose and A. van Dam. 1999. Document structure and markup in the FRESS hypertext system. *Markup Languages* 1, 1 (January): 7-32.
17. R. F. Dillon, Jeff D. Edey, Jo W. Tombaugh. 1990. Measuring the true cost of command selection: techniques and results. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '90)*, 19-26. http://dx.doi.org/10.1145/97243.97247
18. George Fitzmaurice, Azam Khan, Robert Piek\, \#233, Bill Buxton, Gordon Kurtenbach. 2003. Tracking menus. In *Proceedings of the 16th annual ACM symposium on User interface software and technology*, 71-79. http://doi.acm.org/10.1145/964696.964704.
19. J. D. Foley, V. L. Wallace, P. Chan. 1984. The human factors of computer graphics interaction techniques. *IEEE Computer Graphics and Applications* 4, 11: 13-48.
20. Clifton Forlines, Chia Shen, Bill Buxton. 2005. Glimpse: a novel input model for multi-level devices. In *CHI '05 Extended Abstracts on Human Factors in Computing Systems (CHI EA '05)*, 1375-1378. http://dx.doi.org/10.1145/1056808.1056920.
21. Tovi Grossman, Ken Hinckley, Patrick Baudisch, Maneesh Agrawala, Ravin Balakrishnan. 2006. Hover widgets: using the tracking state to extend the capabilities of pen-operated devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'06)*, 861-870. http://doi.acm.org/10.1145/1124772.1124898.
22. Yves Guiard. 1987. Asymmetric division of labor in human skilled bimanual action: The kinematic chain as a model. *Journal of Motor Behavior* 19, 4: 486-517.
23. Joshua Hailpern, Erik Hinterbichler, Caryn Leppert, Damon Cook, Brian P. Bailey. 2007. TEAM STORM: demonstrating an interaction model for working with multiple ideas during creative group work. In *Proceedings of the 6th ACM SIGCHI conference on*

Creativity & cognition (C&C '07), 193-202. http://dx.doi.org/10.1145/1254960.1254987

24. William Hamilton, Andruid Kerne, Tom Robbins. 2012. High-performance pen + touch modality interactions: a real-time strategy game eSports context. In *Proceedings of the 25th annual ACM symposium on User interface software and technology (UIST '12)*, 309-318. http://dx.doi.org/10.1145/2380116.2380156.
25. Gary Hardock, Gordon Kurtenbach, William Buxton. 1993. A marking based interface for collaborative writing. In *Proceedings of the 6th annual ACM symposium on User interface software and technology (UIST '93)*, 259-266. https://doi.org/10.1145/168642.168669.
26. Thomas T. Hewett. 2005. Informing the design of computer-based environments to support creativity. *Int. J. Hum.-Comput. Stud.* 63, 4-5 (October 2005): 383-409. http://dx.doi.org/10.1016/j.ijhcs.2005.04.004.
27. Ken Hinckley, Patrick Baudisch, Gonzalo Ramos, Francois Guimbretiere. 2005. Design and analysis of delimiters for selection-action pen gesture phrases in scriboli. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, 451-460. http://doi.acm.org/10.1145/1054972.1055035.
28. Ken Hinckley, Xiaojun Bi, Michel Pahud, Bill Buxton. 2012. Informal Information Gathering Techniques for Active Reading. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '12)*, 1893-1896. http://dx.doi.org/10.1145/2207676.2208327.
29. Ken Hinckley, Francois Guimbretiere, Patrick Baudisch, Raman Sarin, Maneesh Agrawala, Ed Cutrell. 2006. The springboard: multiple modes in one spring-loaded control. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*, 181-190. http://doi.acm.org/10.1145/1124772.1124801.
30. Ken Hinckley, Koji Yatani, Michel Pahud, Nicole Coddington, Jenny Rodenhouse, Andy Wilson, Hrvoje Benko, Bill Buxton. 2010. Pen + Touch = New Tools. In *Proceedings of the 23nd annual ACM symposium on User interface software and technology*, 27-36. http://doi.acm.org/10.1145/1866029.1866036.
31. Hilary Hutchinson, Wendy Mackay, Bo Westerlund, Benjamin B. Bederson, Allison Druin, Catherine Plaisant, Michel Beaudouin-Lafon, Stéphane Conversy, Helen Evans, Heiko Hansen, Nicolas Roussel, Björn Eiderbäck. 2003. Technology probes: inspiring design for and with families. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '03)*, 17-24. http://dx.doi.org/10.1145/642611.642616.
32. Azam Khan, Justin Matejka, George Fitzmaurice, Gordon Kurtenbach. 2005. Spotlight: directing users' attention on large displays. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '05)*, 791-798. http://dx.doi.org/10.1145/1054972.1055082.
33. David Kirsh. 1995. The intelligent use of space. *Artificial Intelligence* 73: p. 31-68.
34. Scott R. Klemmer, Mark W. Newman, Ryan Farrell, Mark Bilezikjian, James A. Landay. 2001. The designers' outpost: a tangible interface for collaborative web site. In *Proceedings of the 14th annual ACM symposium on User interface software and technology*, 1-10. http://doi.acm.org/10.1145/502348.502350.
35. Axel Kramer. 1994. Translucent patches-dissolving windows. In *Proceedings of the 7th annual ACM symposium on User interface software and technology (UIST '94)*, 121-130. http://dx.doi.org/10.1145/192426.192474.
36. Myron W. Krueger, Thomas Gionfriddo, Katrin Hinrichsen. 1985. VIDEOPLACE-an artificial reality. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '85)*, 35-40. http://dx.doi.org/10.1145/317456.317463.
37. Gordon Kurtenbach and William Buxton. 1991. Issues in combining marking and direct manipulation techniques. In *Proceedings of the 4th annual ACM symposium on User interface software and technology*, 137-144. http://doi.acm.org/10.1145/120782.120797.
38. Gordon Kurtenbach, George Fitzmaurice, Thomas Baudel, Bill Buxton. 1997. The design of a GUI paradigm based on tablets, two-hands, and transparency. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, 35-42. http://doi.acm.org/10.1145/258549.258574.
39. James A. Landay and Brad A. Myers. 2001. Sketching Interfaces: Toward More Human Interface Design. *Computer* 34, 3 (March 2001): 56-64. http://dx.doi.org/10.1109/2.910894.
40. Edward Lank, Jaime Ruiz, William Cowan. 2006. Concurrent bimanual stylus interaction: a study of non-preferred hand mode manipulation. In *Proceedings of Graphics Interface 2006 (GI '06)*, 17-24.
41. Jakob Leitner and Michael Haller. 2011. Harpoon selection: efficient selections for ungrouped content on large pen-based surfaces. In *Proceedings of the 24th annual ACM symposium on User interface software and technology (UIST '11)*, 593-602. http://dx.doi.org/10.1145/2047196.2047275.
42. Yang Li, Ken Hinckley, Zhiwei Guan, James A. Landay. 2005. Experimental analysis of mode switching techniques in pen-based user interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '05)*, 461-470. http://doi.acm.org/10.1145/1054972.1055036.
43. David Lindlbauer, Michael Haller, Mark Hancock, Stacey D. Scott, Wolfgang Stuerzlinger. 2013. Perceptual grouping: selection assistance for digital sketching. In *Proceedings of the 2013 ACM international conference on Interactive tabletops and*

surfaces (ITS '13), 51-60. http://dx.doi.org/10.1145/2512349.2512801.

44. Fabrice Matulic and Moira C. Norrie. 2013. Pen and touch gestural environment for document editing on interactive tabletops. In *Proceedings of the 2013 ACM international conference on Interactive tabletops and surfaces (ITS '13)*, 41-50. http://dx.doi.org/10.1145/2512349.2512802.
45. Sachi Mizobuchi and Michiaki Yasumura. 2004. Tapping vs. circling selections on pen-based devices: evidence for different performance-shaping factors. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '04)*, 607-614. http://dx.doi.org/10.1145/985692.985769.
46. Thomas P. Moran, Patrick Chiu, William van Melle. 1997. Pen-based interaction techniques for organizing material on an electronic whiteboard. In *Proceedings of the 10th annual ACM symposium on User interface software and technology*, 45-54. http://doi.acm.org/10.1145/263407.263508.
47. Elizabeth D. Mynatt, Takeo Igarashi, W. Keith Edwards, Anthony LaMarca. 1999. Flatland: new dimensions in office whiteboards. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems (CHI '99)*, 346-353. http://dx.doi.org/10.1145/302979.303108
48. Ken Perlin and David Fox. 1993. Pad: an alternative approach to the computer interface. In *Proceedings of the 20th annual conference on Computer graphics and interactive techniques (SIGGRAPH '93)*, 57-64. http://dx.doi.org/10.1145/166117.166125.
49. Florian Perteneder, Martin Bresler, Eva-Maria Grossauer, Joanne Leong, Michael Haller. 2015. cLuster: Smart Clustering of Free-Hand Sketches on Large Interactive Surfaces. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology (UIST '15)*, 37-46. http://dx.doi.org/10.1145/2807442.2807455.
50. Ken Pfeuffer, Jason Alexander, Hans Gellersen. 2016. Partially-indirect Bimanual Input with Gaze, Pen, and Touch for Pan, Zoom, and Ink Interaction. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16)*, 2845-2856. http://dx.doi.org/10.1145/2858036.2858201.
51. Ken Pier and James A. Landay, *Issues for Location-Independent Interfaces*. 1992: Technical Report ISTL92-4, Xerox Palo Alto Research Center.
52. Jeffrey S. Pierce, Matthew Conway, Maarten van Dantzich, George Robertson. 1999. Toolspaces and glances: storing, accessing, and retrieving objects in 3D desktop applications. In *Proceedings of the 1999 symposium on Interactive 3D graphics (I3D '99)*, 163-168. http://dx.doi.org/10.1145/300523.300545.
53. Stuart Pook, Eric Lecolinet, Guy Vaysseix, Emmanuel Barillot. 2000. Control menus: excecution and control in a single interactor. In *CHI '00 Extended Abstracts on Human Factors in Computing Systems (CHI EA '00)*, 263-264. http://dx.doi.org/10.1145/633292.633446.
54. A. T. Purcell and J. S. Gero. 1998. Drawings and the design process. *Design Studies* 19, 4: 389-430.
55. J. Raskin. 2000. The Humane Interface: New Directions for Designing Interactive Systems. ACM Press.
56. George G. Robertson and Stuart K. Card. 1997. Fix and float: object movement by egocentric navigation. In *Proceedings of the 10th annual ACM symposium on User interface software and technology (UIST '97)*, 149-150. http://dx.doi.org/10.1145/263407.263535.
57. Jaime Ruiz and Edward Lank. 2007. A study on the scalability of non-preferred hand mode manipulation. In *Proceedings of the 9th international conference on Multimodal interfaces (ICMI '07)*, 170-177. http://dx.doi.org/10.1145/1322192.1322223.
58. Donald A. Schön. 1992. Designing as reflective conversation with the materials of a design situation. *Research in Engineering Design* 3, 3: 131-147. http://dx.doi.org/10.1007/BF01580516.
59. Julia Schwarz, Scott Hudson, Jennifer Mankoff, Andrew D. Wilson. 2010. A framework for robust and flexible handling of inputs with uncertainty. In *Proceedings of the 23nd annual ACM symposium on User interface software and technology*, 47-56. http://doi.acm.org/10.1145/1866029.1866039.
60. A. Sellen, G. Kurtenbach, W. Buxton. 1992. The prevention of mode errors through sensory feedback. *Human Computer Interaction* 7, 2: 141-164.
61. Jami J. Shah, Noe Vargas-Hernandez, Joshua D. Summers, Santosh Kulkarni. 2001. Collaborative Sketching (C-Sketch)--An Idea Generation Technique for Engineering Design. *Journal of Creative Behavior* 35, 3 (3rd Qtr 2001): 168-198
62. F. Shipman and C. Marshall. 1999. Formality Considered Harmful: Experiences, Emerging Themes, and Directions on the Use of Formal Representations in Interactive Systems. *Computer Supported Cooperative Work (CSCW)* 8, 4: 333-352.
63. Ben Shneiderman. 2007. Creativity support tools: accelerating discovery and innovation. *Commun. ACM* 50, 12 (December 2007): 20-32. http://dx.doi.org/10.1145/1323688.1323689.
64. L. Tesler. 1981. The smalltalk environment. *Byte* 6, 8: 90-147.
65. Theophanis Tsandilas, Catherine Letondal, Wendy E. Mackay. 2009. Musink: composing music through augmented drawing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '09)*, 819-828. http://dx.doi.org/10.1145/1518701.1518827
66. Md. Sami Uddin, Carl Gutwin, Benjamin Lafreniere. 2016. HandMark Menus: Rapid Command Selection and Large Command Sets on Multi-Touch Displays. In *Proceedings of the 2016 CHI Conference on Human*

Factors in Computing Systems (CHI '16), 5836-5848. http://dx.doi.org/10.1145/2858036.2858211.

67. Julie Wagner, Stéphane Huot, Wendy Mackay. 2012. BiTouch and BiPad: designing bimanual interaction for hand-held tablets. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '12)*, 2317-2326. http://dx.doi.org/10.1145/2207676.2208391.
68. Julie Wagner, Eric Lecolinet, Ted Selker. 2014. Multi-finger chords for hand-held tablets: recognizable and memorable. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '14)*, 2883-2892. http://doi.acm.org/10.1145/2556288.2556958.
69. Andrew M. Webb, Michel Pahud, Ken Hinckley, Bill Buxton. 2016. Wearables as Context for Guiard-abiding Bimanual Touch. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology (UIST '16)*, 287-300. https://doi.org/10.1145/2984511.2984564.
70. Jacob O. Wobbrock, Meredith Ringel Morris, Andrew D. Wilson. 2009. User-defined gestures for surface computing. In *Proceedings of the 27th international conference on Human factors in computing systems*, 1083-1092. http://doi.acm.org/10.1145/1518701.1518866.
71. Yasuhiro Yamamoto and Kumiyo Nakakoji. 2005. Interaction design of tools for fostering creativity in the early stages of information design*. Int. J. Hum.-Comput. Stud.* 63, 4-5 (October 2005): 513-535. http://dx.doi.org/10.1016/j.ijhcs.2005.04.023.
72. Robert Zeleznik, Andrew Bragdon, Ferdi Adeputra, Hsu-Sheng Ko. 2010. Hands-on math: a page-based multi-touch and pen desktop for technical work and problem solving. In *Proceedings of the 23nd annual ACM symposium on User interface software and technology*, 17-26. http://doi.acm.org/10.1145/1866029.1866035.