Gaze-Shifting:
Direct-Indirect Input with Pen and Touch Modulated by Gaze

Ken Pfeuffer, Jason Alexander, Ming Ki Chong, Yanxia Zhang, Hans Gellersen

Lancaster University
Lancaster, United Kingdom
{k.pfeuffer, j.alexander, m.chong, y.zhang5, h.gellersen}@lancaster.ac.uk

ABSTRACT

Modalities such as pen and touch are associated with direct input but can also be used for indirect input. We propose to combine the two modes for direct-indirect input modulated by gaze. We introduce gaze-shifting as a novel mechanism for switching the input mode based on the alignment of manual input and the user's visual attention. Input in the user's area of attention results in direct manipulation whereas input offset from the user's gaze is redirected to the visual target. The technique is generic and can be used in the same manner with different input modalities. We show how gaze-shifting enables novel direct-indirect techniques with pen, touch, and combinations of pen and touch input.

ACM Classification Keywords

H.5.2. Information interfaces and presentation: User Interfaces: Input devices and strategies

Author Keywords

Gaze; touch; pen; direct input; indirect input; eye tracking

INTRODUCTION

Any direct input device can also be operated in indirect input mode, by redirecting input to a target that is offset from the physical location at which the input is sensed [18]. Support of indirect input alongside direct input has been motivated to overcome problems that are inherent with direct input, such as occlusion of the target during manual input [35], effort and time required for reaching targets located further from the user [5], and precision limitations due to the low "resolution" of fingers [2] and static control-display ratio [22]. A range of techniques have been designed to address these issues by extending direct input with indirect input without changing device or modality. For example, multi-touch as well as pen interfaces have been extended for users to be able to dynamically switch between absolute positioning and relative positioning [4, 12, 36]. The combination of the two modes can enable users to perform the same task interchangeably with direct or indirect input [12], or to switch tasks associated
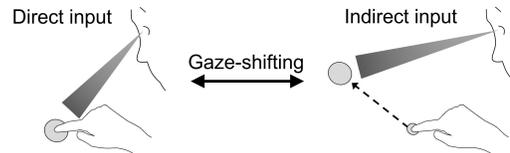
Figure 1: Gaze-shifting: manual input by the user is modulated by their gaze. Input is direct when the user's gaze is on the touch location (left), but redirected to the gaze location if gaze and touch are separate (right).
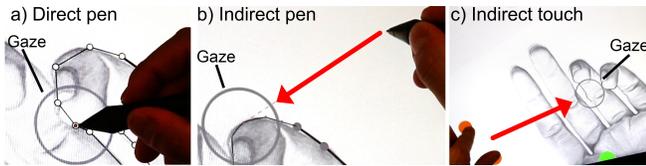
with either mode (e.g., writing/drawing versus command input [14]). A dynamic redirection of input generally increases expressiveness of surface interactions as it allows the same target to be manipulated from different input positions [27].

In this work, we explore the combination of direct and indirect modes of input with pen and touch, and propose to use gaze to facilitate the combination. We focus on hybrid *direct-indirect input* techniques that provide an integrated user experience of direct and indirect input. Direct-indirect input requires a switching mechanism, for users to dynamically transition between the two modes, and a redirection mechanism for input to be shifted to an offset target when the user enters indirect mode. Mode switching has been studied extensively for pen interfaces [21] and a host of sensory mechanisms have been explored [17]. Mode switching of manual touch has been explored in more specific contexts (e.g., to shift input to resolve "fat finger" target ambiguity [35]), but can generally be facilitated by dedicated gestures (e.g., [36]). Input redirection is commonly supported by a cursor that appears in indirect input mode, but the redirection can also be context-dependent (e.g., indirect manipulation of a previously selected target) or facilitated by special-purpose widgets (e.g., for target reaching [1, 5]).

We introduce *gaze-shifting* as a novel technique for direct-indirect input, embracing the user's gaze to modulate their manual input. Figure 1 illustrates the principle of gaze-shifting. When a user initiates manual input in the area under their gaze, then it is processed as direct input at the touch location. When manual input is initiated while the user looks at an offset target, the input is redirected to the target. Gaze-shifting is generic and can be used in the same manner with different input modalities, and specifically pen input, touch input, and combinations of both. The technique provides a natural mechanism for mode switching that is based on the alignment of manual input and visual attention for direct versus indirect input. Implicitly, the technique provides a consistent mechanism for shifting the user's input from a touch

Figure 2: Direct-indirect input enabled by gaze-shifting: a) direct pen input is used to outline an image with straight lines; b) indirect pen input is used for curving of gaze selected lines; c) indirect touch is used to zoom in and out relative to the user's gaze. The grey circles indicate the user's gaze, and the arrows the redirection of input.

point to a gaze-identified target, and vice versa from an indirectly controlled target back to the touch point.

Gaze-shifting enables the design of novel direct-indirect techniques that combine direct and indirect input with pen and/or touch in cohesive workflows. Figure 2 illustrates this with a scenario of tracing an image. A user can mark outline points with direct pen input, and the points are connected by straight lines to create a rough outline. The lines can be edited in detail with indirect pen input for which the user moves the pen off the visual target so not occlude any detail. The user can zoom in and out with indirect touch input relative to the visual target. All three tasks (outlining, curving, zooming) have a distinct mapping to an input mode, and the user can easily transition between these modes.

To explore gaze-shifting, we have integrated a pen and touch display with an eye tracker, and created a design application as playground for the development of direct-indirect interaction techniques. The application provides a meta-interface in which gaze-shifting is used to dynamically configure the four modes of direct pen, indirect pen, direct touch and indirect touch. We have used this application environment to develop novel direct-indirect techniques that leverage gaze-shifting with touch input, pen input, and a combination of both pen and touch.

Our contributions are as follows. First, we introduce gaze-shifting as a generic concept and technique to facilitate direct-indirect input. Second, we describe a prototype system and application that supports dynamic configuration of pen and touch modalities for use with gaze-shifting. Third, we contribute novel direct-indirect input techniques that leverage pen, touch, and gaze in different configurations and illustrate the design space opened up by gaze-shifting. Finally, we report on first insights on the user experience of direct-indirect interaction with gaze-shifting.

RELATED WORK

Combined Direct and Indirect Input
The properties and advantages of direct versus indirect input have been explored widely [13, 11, 29, 23]. A range of works have extended direct input with indirect input to improve speed [13, 22], precision [2, 4, 36], or reach [1, 5, 12, 26]. Most existing work has focussed on overcoming specific limitations by extending direct with indirect input. For example, TractorBeam and Pointable extend direct touch interaction with indirect means to select targets beyond the user's

reach [3, 26]. We focus more generally on the design opportunities provided by combining direct and indirect input modes. Examples of prior pen and touch work that has been exploratory include HybridPointing and Rock & Rails. HybridPointing is a pen interface that enables users to switch from direct pen input to indirect input by tapping a trailing widget; this provides the user with a cursor they can be positioning with relative input motion [12]. Rock & Rails exemplifies extension of direct input with indirect input in a multi-touch context and introduces specific hand gestures that enable users to create proxies to offset targets, and to manipulate targets with a combination of direct and indirect input [36].

Gaze for Multimodal Input
Gaze has been widely studied as explicit input but it has also been shown that gaze can be effective as contextual input in support of other modalities. This has been exemplified in MAGIC, where gaze modulates mouse input by warping the mouse cursor to the gaze area to eliminate large portions of the cursor movement [38]. In LookPoint, gaze is used to implicitly switch input of mouse and keyboard between multiple displays [10]. Rake Cursor [7] and Ninja Cursors [20] show multiple cursors on the UI, and use eye gaze to implicitly select the active cursor. The gaze-shifting technique is designed in the same spirit of exploiting gaze as a context to modulate the input provided by other modalities. Contrasting earlier work, we focus on the use of gaze to facilitate transition between direct and indirect uses of other input modalities.

Gaze is also explored in conjunction with multi-touch. A variety of works have considered gaze for redirection of multi-touch input to remote displays. Stellmach and Dachselt investigated drag and pinch gestures performed on a handheld multi-touch device to position and manipulate gaze-selected targets on a remote display [30, 31]. Turner et al. investigated variants of indirect RST (rotate, scale, translate) gestures for gaze-selected object manipulation across large screens [32], as well as gaze-assisted techniques for transferring content across displays [33, 34]. The motivation for use of gaze in these works was to facilitate interaction across devices and with displays beyond the user's reach. In contrast, Pfeuffer et al. recently explored how gaze can complement multi-touch on the same surface. Their work introduced Gaze-touch, a technique based on a *gaze selects, touch manipulates* division of labour [27]. While the work focused on techniques enabled by touch redirection to gaze targets, it also showed that conventional gaze-touch and conventional direct touch can be interleaved for interaction, which directly inspired our development of the gaze-shifting concept.

Pen and Touch Interaction
As reviewed above, hybrid direct-indirect input has been studied with both pen and touch interfaces. Here we review additional aspects of relevance in prior pen and touch research. Many pen interfaces complement inking with additional modes such as editing, moving, or drawing [21, 24], and research has investigated a wide range of methods to switch pen mode. Explicit mode switching techniques have employed sensors around the pen, e.g. with pen rolling [6],

hover [14], pressure, button, dwell-time [21], as well as auxiliary sensors [16]. Modes can also be implicitly detected, e.g. by inferring the mode from the pen trajectory [28], or from grip and other context sensed [17]. Gaze as proposed in our work can be considered as a complementary pen context.

Our development of novel techniques is grounded in the pen and touch literature. Hinckley et al. argued that both pen and touch work well in tandem based on the division of labour *pen writes, touch manipulates* [19]. Brandl et al. proposes use of the pen in the dominant hand, and touch input with the the non-dominant hand, based on Guiard's principles [9, 15]. Our work leverages these insights in the design of techniques that combine pen and touch with gaze as context.

GAZE-SHIFTING

Gaze-shifting is conceived as a generic mechanism for switching between direct and indirect modes of manual input. The technique enables direct and indirect input to be combined in novel and dynamic ways. We first review the design opportunities afforded by concept, and then discuss system design considerations.

Design Opportunities

Direct-indirect Transitions. Designers can leverage gaze-shifting to create techniques that leverage not only direct versus indirect modes but also the different transitions between the modes. There are four possible transitions for input to be provided from one position to another: *direct to direct*, *direct to indirect*, *indirect to indirect*, and *indirect to direct*. A designer can assign different behaviours to any of these transitions. Figure 3 illustrates the four transition within a line drawing example. A user touch the first and second point directly, and as the user looks at the touches the system interprets touches as direct (a). The user then looks at a point offset from their hands and triggers indirect touch, which extends the line from the direct to the indirect position (b). From this point, the user looks at another remote point and indirectly selects it as well (c). Lastly, the user looks back at their hand, which the system interprets as intention for direct input, and a touch directly selects the point (d). In this example, each transition results in the same effect, i.e. the creation of a line point. However, each transition can also be associated with different behaviour, and we will illustrate this further below with different copy & paste semantics depending on the direct-indirect transitions (Fig. 12).

Interaction Models: Direct interaction is based on a 1:1 mapping between input and output, but indirect input can have variable mappings that we can consider as interaction models (Figure 4). For example, indirect touch can adopt a model similar to a laptop's touchpad: a cursor becomes visible after shifting to indirect mode, (1) an indirect touch dragging moves the cursor, (2) double tap selects a target, and (3) dragging after double tap manipulates the target. A distinct cursorless indirect model is Gaze-touch [27], where indirect touches issue the same 'direct touch' effect on the gaze-identified target: (1) look to indicate the target, (2) indirectly touch down to select it, and (3) touch drag to manipulate the target. In comparison, cursor based models increase precision
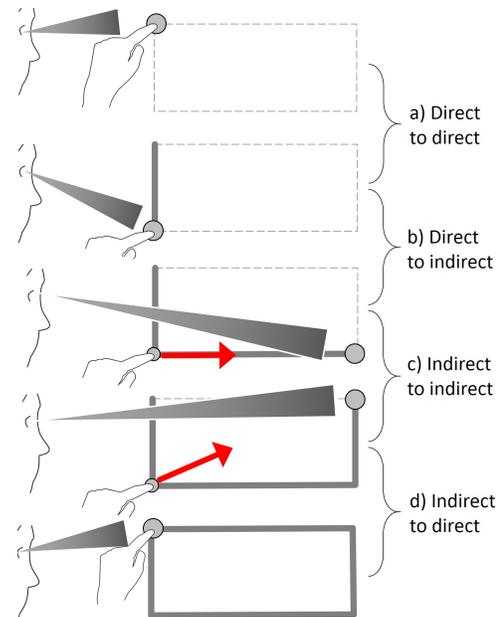
Figure 3: Direct-indirect transitions: successive input can involve four types of transition. In this example, the same task is completed while transitioning across different modes. Alternatively, a designer can associate each type of transition with different semantics.

as users precisely position the cursor. Cursorless models only use gaze to select which is less precise, but faster as users immediately manipulate without prior cursor dragging. Notably, a hover-enabled device such as a pen can adopt a digitiser or mouse like model: (1) hover drags the cursor, (2) pen down clicks, and (3) pen drag manipulates an object.
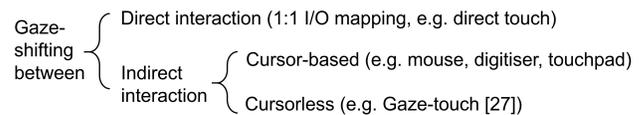
Figure 4: Interaction models for direct and indirect input

System Design Considerations

From a system's perspective, gaze and manual input are two positions on the input surface. As illustrated in Figure 5, the system evaluates whether the input is within the user's gaze area, by applying a threshold on the distance between gaze point and manual input position.
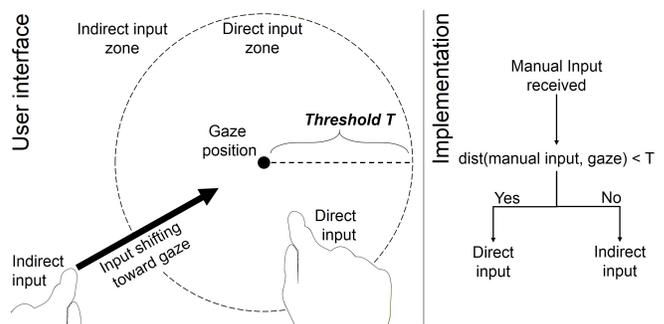
Figure 5: Gaze-shifting from a system's perspective

Threshold Configuration: The threshold defines the boundaries of direct and indirect input. Larger thresholds make indirect input difficult as users need to move their hand far away from their gaze, and smaller thresholds make direct input difficult as users need to keep their input close to where they look. As a starting point, the threshold can be based on human vision. The direct input zone includes the foveal vision, an area of higher acuity, and the indirect input zone aligns with the peripheral vision where vision is generally poor.

Context-awareness: The threshold should consider the task's context and the ratio between needed direct and indirect interaction. Principally, activation can trigger at any input event, e.g. at pen down, drag, or up, as users can move their gaze arbitrarily. In practice, the context of the task at hand needs to be carefully considered. For example, gaze-shifting can be disabled when users draw with a pen, to avoid unexpected behaviour when users briefly look away. On the other hand, when users looked at a menu and issue manual input from clearly far off, it is likely that the user intends to trigger indirect input to interact with the menu. When user interaction is mainly based on direct manipulation, a large threshold is recommended to avoid false positive errors (detect indirect instead of direct).

APPLICATION

We now describe our prototype design application. The idea is to provide basic drawing, design, and manipulation tools and with them explore gaze-shifting across different contexts. Our application is based on the following components:

- Input modalities: pen and touch.
- Input modes: direct pen, indirect pen, direct touch, indirect touch.
- Toolbars: 4 input mode toolbars (one for each), 3 pen specific toolbars (brush size, shapes, colour palette).
- Modes: Each toolbar offers a set of modes. A mode enables a specific drawing or object manipulation operation with an input mode.
- Configuration: a state where all input modes are mapped to a mode.
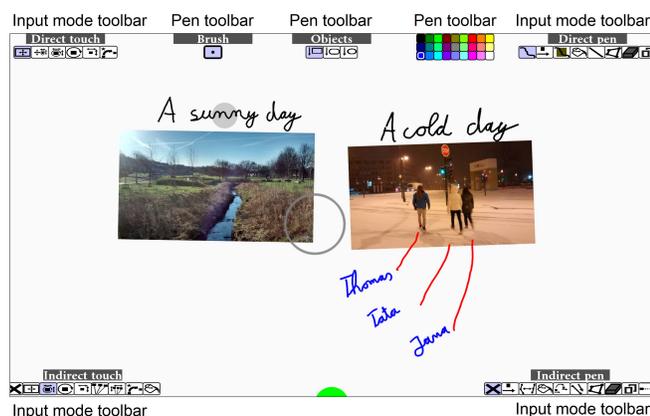

Figure 6: The user interface and its toolbars

Toolbars

Figure 6 shows an instance of the UI and its toolbars. Horizontally, both types of input device are shown (left touch *versus* right pen), and vertically, both types of input mode are shown (top direct *versus* bottom indirect). In light of pen and touch application principles [9, 19], we used inking modes for pen, manipulation modes for touch, and some commonly used modes:

Pen Toolbars: Both direct and indirect pen toolbars include conventional operations such as free inking, move object, fill object, draw line, draw shape, eraser, copy & paste of objects, and a default no-mode mode. Modes exclusive for indirect pen are object editing, where users edit vector points of a shape, and object association, which allows users to draw a connection between two objects with a dotted line.

Touch Toolbars: Both direct and indirect touch toolbars include Five RST based modes. These enable single-point dragging and two-point pinching gestures. They include one mode for global navigation, where the whole canvas is pan & zoomed, and one mode for single object RST that affects one object. Additional modes are pinch gestures that only affect R, S, or T dimension to support separability of touch gestures [25]. Lastly, the 'transfer' mode allows instant object transfer from a direct touch position to the user's gaze position; also a default no-mode mode is available.

The additional brush, objects, and colour toolbars allow users to configure the brush size, create primitive shapes, and set a colour for the drawing modes, respectively. Notably, each toolbar dynamically changes its opacity so that the user's focus is not detracted from the workspace. A toolbar fades in quickly when looked at (.2s), and fades out slowly when looked away (1.6s). 'Looked' is detected when the user's gaze is within 150px around the toolbar. The minimum opacity is 5%, so that users can easily find the toolbars (Figure 6 shows all toolbars at 100% for descriptive purpose).

Users can employ direct and indirect input to interact with the toolbars. First, when looking at a toolbar and issuing direct pen/touch input on it, the mode is directly selected by pen/touch. Second, users look at the toolbar, put the pen or finger down on the canvas (Figure 7a), and indirectly drag left or right to move the toolbar's selection mask to the corresponding direction (b).

System Implementation

Our system consists of three main parts: a *pen and touch display* (Wacom Cintiq 22HD touch), an *eye tracker* attached to the bottom of the display (Tobii EyeX, 30 hz, ≈1° of visual angle accurate), and a *laptop* (Windows 7, 8GB RAM, dualcore i7 2.9GHz CPU). The display is 1080p, 21.5" (475 x 267mm), and supports 10 finger multi-touch and a Wacom Grip Pen. Default Windows 7 settings distinguish pen and touch, and disable touch at pen hover/down. The display is oriented at 45° toward and ≈ 60cm away from the user. Hand occlusion of the tracker's view to the eyes is mostly avoided as hands are usually held around the view. The software is implemented with Java using MT4J (Multitouch for Java v0.9, https://code.google.com/p/mt4j/, 08/04/2015). We

use dynamic gaze sampling, using raw data at fast eye movement (>75px between two samples), and otherwise sampled data for 1s (30 samples) to enable quick travel of distances, but a smooth gaze indication when fixating an area.

Gaze-shifting implementation is based on Figure 5 with the following thresholds. For toolbars, indirect is detected when users look <150px (37mm) close to the menu's border, and issue input from outside. When no toolbar interaction is detected, on default the system continuously checks direct or indirect mode based on a 350px (86mm) radius. For touch input, the detection occurs at touch down, drag, and up events. For the pen, hover events are used in addition. When users hover into 'indirect' mode, techniques can show a mouse cursor that initially appears at the user's gaze in order to provide feedback before users issue input.

DIRECT-INDIRECT INTERACTION EXAMPLES

We now describe examples of gaze-shifting based direct-indirect interaction. Each consists of a fixed mode configuration of our application. We categorise them into tasks for pen, touch, and combined input. Table 1 overviews the examples and used modalities. Each example demonstrates a specific use of gaze-shifting, and collectively, they depict the flexibility of gaze-shifting's application across tasks, techniques, and modalities. The figures indicate the user's gaze with a grey circle.

		Pen		**Touch**	
	Fig.	Direct pen	Indirect pen	Direct touch	Indirect touch
Draw directly, configure pen indirectly	7	✓	✓		
Trace line directly, edit line indirectly	8	✓	✓		
Trace line directly and indirectly	9	✓	✓		
Select colour directly, set colour indirectly	10			✓	✓
Direct local zoom, indirect global zoom	11			✓	✓
Direct and Indirect Copy & Paste	12	✓	✓	✓	✓
pen edits, touch zooms	13	✓	✓		✓
Select directly, transfer indirectly, Draw directly, associate indirectly	14	✓	✓	✓	✓

Table 1: Overview of the direct-indirect interactions (*rows indicate the task, columns the integrated modalities*)

PEN

The following examples describe how gaze-shifting aids pen-specific functionality such as inking, shape design, and hover based techniques.

Draw Directly, Configure Pen Indirectly

This example demonstrates rapid pen mode switching. Direct pen input enables standard inking, and indirect pen interaction with toolbars. We added three toolbar variations specifically designed for the pen. In general, the menus are controlled by using gaze to select the overall menu, indirect pen down confirms menu selection, and indirect pen dragging moves the selection mask across the menu.

In particular, we implemented three variations of this interaction, each demonstrated in a specific pen menu type. First, the 'brush' toolbar displays only one mode which visual icon changes size according to the dragging. Pen dragging from left (c) to right (d) increases brush size. Second, the 'objects' toolbar has extended functionality: dragging left/right changes the type of a shape, but dragging from up (e) to down (f) creates the selected shape at the pen's direct physical position. This allows users to quickly create new notes of various shapes. Then, the user can further reposition the object (here: a rectangle) with direct pen. Lastly, the 'colour palette' toolbar demonstrates how users can easily interact with a menu that involves many small-sized modes. A grid of 9x3 colours is provided (g-h), on which users perform indirect 2D dragging to select a desired colour.
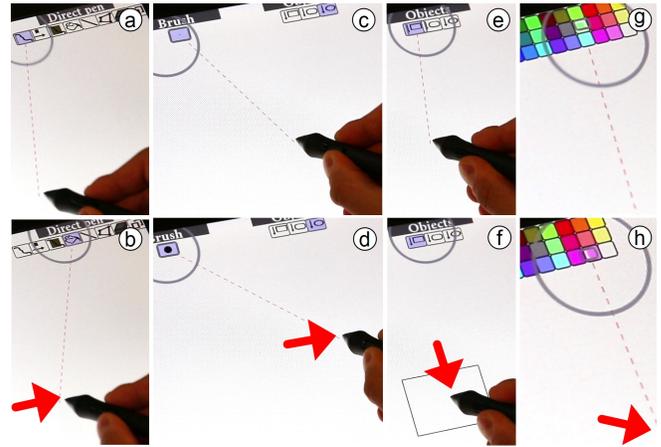
Figure 7: Draw directly, configure pen indirectly: users switch a toolbar's mode by look and indirect touch down (a), and vertical drag (b). Special toolbars are 'brush' that displays one adjustable widget (c-d), 'objects' where a drag down creates the selected shape at the pen's position (e-f), and 'color palette' where 2D dragging switches colours (g-h).

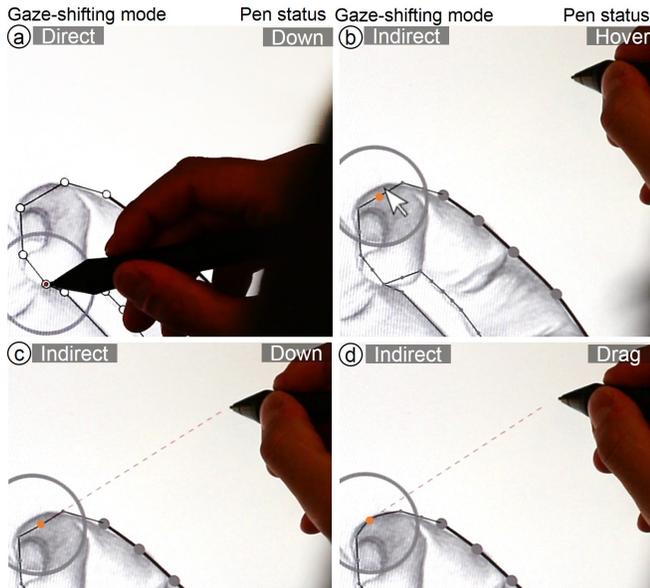### Trace Line Directly, Edit Line Indirectly

This example demonstrates how users shift between two different modes with a pen. Tracing is a common task, e.g. to digitise hand-drawn figures. A method to accomplish this is drawing straight lines at the outline, and then rounding them off to align them to the figure. These two tasks can be interleaved into pen input: direct pen input allows straight line drawing, and indirect pen input allows rounding off the lines.

For example, when looking close to the pen, users draw straight lines (Figure 8a). In particular, users perform pen down on each point, which get interconnected to a shape. A shape finishes when users connect the current point to an existing point, or perform double click. When pen hovers out of the user's sight, the system switches to line edit mode, where users curve lines with indirect pen input (b).

Hover into the indirect pen mode changes the visualisation of the shapes and enables cursor control. Each straight line of a shape shows three points (start, center, end) enlarged to indicate their editability. Hovering into indirect makes a mouse cursor appear at the user's gaze (b). Indirect pen hover moves the cursor to precisely select pixels. Pen down selects a point

(c), and pen drag moves the point 1:1 to adjusts the roundness of the line (d). Pen up releases the selection.

The cursor appearance is delayed by 300 ms to avoid unwanted cursor movement when users just moved the pen to indirect mode. To avoid pen hover clutching for long distances, the cursor automatically jumps to the gaze position (based on MAGIC [38]) if the mouse cursor is far away from the gaze position (>250px for 300ms).
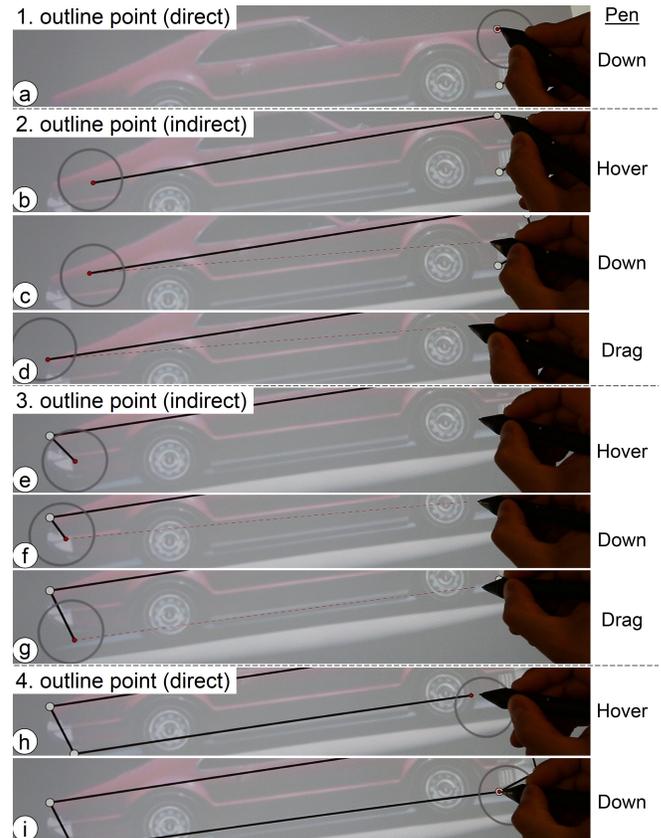

Figure 8: Trace line directly, edit line indirectly: direct pen input enables drawing straight lines from point to point (a). Transitioning from direct to indirect pen input makes a mouse cursor appear at the user's gaze. Indirect pen hover moves this cursor (b), indirect pen down selects a specific point of a line (c), and pen dragging moves this point which adjusts the roundness of the line (d).

Trace Line Directly and Indirectly

This examples demonstrates how users can facilitate direct and indirect pen input for the same outlining task. It can be useful for tracing shapes of larger size, creating lines that stretch across the display, and for general dot-to-dot designs. The main benefit is that this technique reduces the need to reach across distant points on the display. Tracing involves creating multiple points that become connected to form an outline. Creating points is consistent across direct and indirect input: a pen down denotes trace point creation, pen drag refines its position, and a pen up confirms it.

Figure 9 illustrates this technique. A tap of direct pen input creates a new point at its position (a). When looking far away from the pen's position, the next point gets previewed at the gaze position (b). Putting the pen down (indirectly) will confirm the current gaze point as a desired position for the outlining point (c), followed by indirect pen dragging to refine this point's position (d). Pen up confirms the point and begins the next point. Again, this point snaps to the gaze position (e) until pen down (f), followed by pen drag to refine this point's position (g), and a pen up to confirm this point. Users shift back to direct mode by looking close to the pen, where the outlining point snaps to the pen's hover position (h). A direct

tap will create the next shape point (i). Notably, users can also indirectly begin the shape in a remote point, or complete a whole shape without using direct pen input.

Figure 9: Trace line directly and indirectly: direct tap sets an outline point (a). The user lifts the pen and looks at the next point, which extends the line to the gaze (b). Indirect pen down takes over (c), and pen dragging refines the point's position (d). Pen up confirms the point and initiates a new point snapped to the user's gaze (e). Indirect pen down (f) and drag refines the next point (g). Pen up and look back shaps the line to the hover position (g), and direct pen down sets the next outline point (i).

TOUCH

The following examples describe gaze-shifting interactions specifically for finger tapping and pinch-to-scale touch gestures.

Select Colour Directly, Set Colour Indirectly

Our pen mode switching examples (Fig. 7) use direct input for primary manipulations, and indirect input for mode switching. In this example, we want to demonstrate the other direction: direct input for mode switching (colour selection), and indirect input for primary manipulations (colour set). When looking at the toolbar (Fig. 10a), users can directly select a colour (b), and when looking at an object in the canvas, the same touch will apply the selected colour to the gaze-identified object (c). This enables a dynamic interplay between selecting and setting colours, with potential application for formatting text, adjusting shapes, or filters on images.
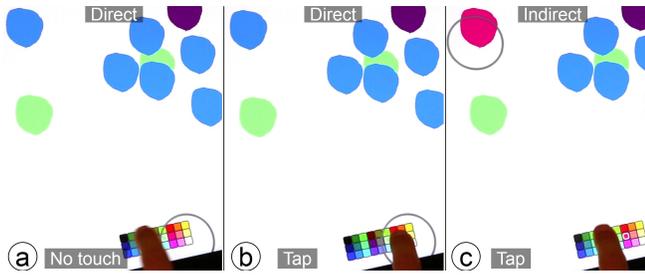
Figure 10: Select colour directly, set colour indirectly: select the colour by looking at the toolbar (a) and directly tapping on the colour (b), and set the colour by looking at the target and an indirect tap (c).

Locally Zoom Directly, Globally Zoom Indirectly

This technique demonstrates how users rapidly shift between local and global pinch-to-zoom operations. To get both operations with direct manipulation, touch on objects is typically used for single object RST, and global zoom is active when touching blank space. This can be difficult with many objects, or in case blank space is hard to acquire. To approach this, we use single object RST when interacting with direct touch, and global canvas zoom when interacting with indirect touch.

Figure 11 illustrates this technique. Global manipulation is active when users indirectly pinch far apart from their gaze position (a), which will zoom the canvas into the user's gaze location (b). When a desired image is found, users directly manipulate this image with direct touch (c-d). Users can immediately return to global manipulation by moving the hand off the gaze position (e-f).
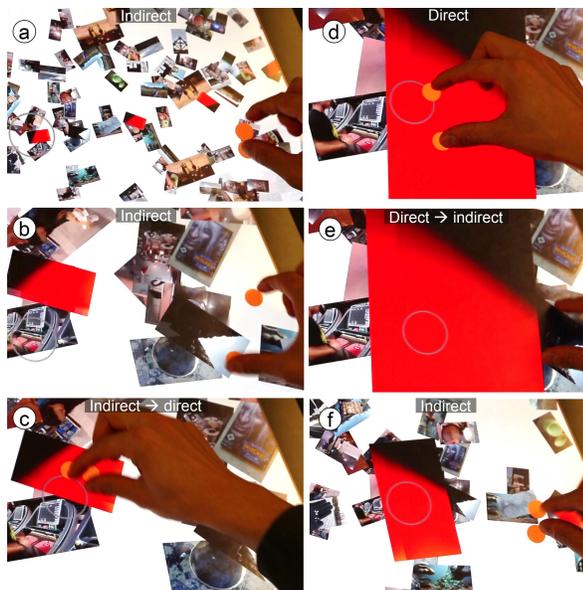

Figure 11: Locally zoom directly, globally zoom indirectly: users can globally zoom the canvas with indirect touch (a-b), locally zoom objects with direct touch (c-d), and immediately return to global zooming (e-f).

PEN AND TOUCH

We now describe interactions that use both modalities. In the first example, pen or touch is interchangeable, and the other examples divide the labour between the modalities.

Copy & Paste Directly and Indirectly

This example demonstrates how the addition of gaze-shifting can principally enable five variations of the same task, on the example of copy & paste. This is possible as each technique is based on a different gaze and manual input behaviour, yet the copy object and paste destination position are consistently located within the user's gaze. Each copy or paste can be direct or indirect. Pen down selects the copy target, and a pen drag gesture will paste it. Figure 12 illustrates the techniques:

- *direct-to-direct:* direct input copies and pastes at the pen's physical position (Figure 12a). This is the standard method as used in conventional applications.
- *Direct-to-indirect:* direct input copies the object, but after looking at a remote target, the indirect drag gesture pastes the object at the gaze position (b). This is useful when users easily reach the object, but hardly the paste destination.
- *indirect-to-direct:* Indirect pen copies the object, and by looking at the pen's position, direct drag pastes the object at this position (c). This can be useful to bring remote objects close to the user, to then perform further manipulations directly on it.
- *indirect-to-indirect (same):* indirect pen copies and pastes the object at the same gaze-identified position (d), with the aim to quickly replicate remote objects.
- *indirect-to-indirect (different):* indirect pen copies the object, and by looking at a remote target, an indirect pen drag gesture pastes it there (e). In this method, users need to look at two remote positions, providing the user with the freedom to copy anywhere located objects to any position.
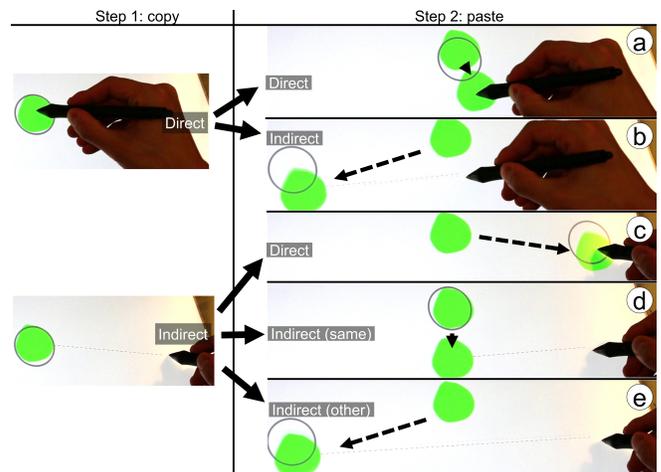

Figure 12: Five techniques for copy & paste: direct copy & paste (a), direct copy & indirect paste (b), indirect copy & direct paste (c), indirect copy & indirect paste at the same position (d), and indirect copy & indirect paste at another position (e).

Pen Draws and Edits, Touch Zooms

This example demonstrates how indirect pan & zoom complements pen activities, as illustrated in Figure 13. A user directly traces the outline of a figure (a), then lifts the pen to perform zooming. The user indirectly touches down (b) and performs pinch-to-zoom that affects their gaze position (c). Afterwards, the user continues editing the line with indirect pen input (d). This example supports users when pen and

379
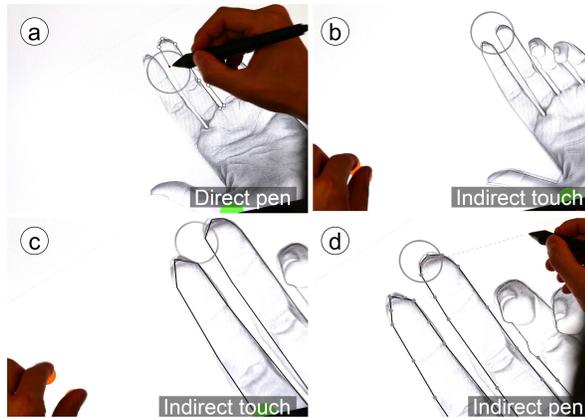
Figure 13: Pen writes and edits, touch zooms: A user first directly traces with the pen (a), performs indirect gaze-directed zooming with touch (b-c), and then performs line editing with indirect pen input (d).

touch physically interfere on the same target, as both modalities can be temporally used indirecty, from remote.

Draw Directly, Associate Indirectly (Pen),

Select Directly, Transfer and Manipulate Indirectly (Touch)

Lastly we present an example where gaze-shifting supports more complex combinations of techniques. A modeling scenario includes the following techniques:

- *Direct pen*: allows free drawing/writing (Figure 14a).
- *Indirect pen (on toolbar)*: creates notes at the direct pen's position as illustrated in Figure 7e-f.
- *Indirect pen (on note)*: allows to establish associations between notes. A user looks at a note, performs pen down (Fig. 14e), then looks at a second note (f), and performs pen up (g). This will create a visual line between both notes. Users can delete it by drawing another association between the same notes.
- *Direct/indirect touch*: direct or indirect touch is set to RST, and thus is used to rescale or move note objects (Figure 14h).
- *Direct+indirect touch*: combined direct and indirect touch input enables object transfer. Directly touch a note (Fig. 14b), look at a remote position (c), and the note is transferred to this position (d). The transfer is delayed by 300 ms, so that the object won't jitter along the user's gaze.

Figure 14 illustrates a whole scenario: a user first labels the notes with direct pen input (a), then uses touch to transfer the objects into a spatially arranged model (b-d). After this step, the user establishes logical connections between the notes using indirect pen input (e-g), and lastly users adjust the notes' positions with RST manipulation of indirect touch input(h).

USER STUDY

The study goal is to gather insights about the actual act of gaze-shifting, and what potential issues occur during its use. We designed a task for pen, touch, and their combination:

Pen (Figure 16a): The pen tracing task investigates shifting between direct line drawing and indirect line editing in an image outlining task (c.f. Figure 8). Users traced six shapes
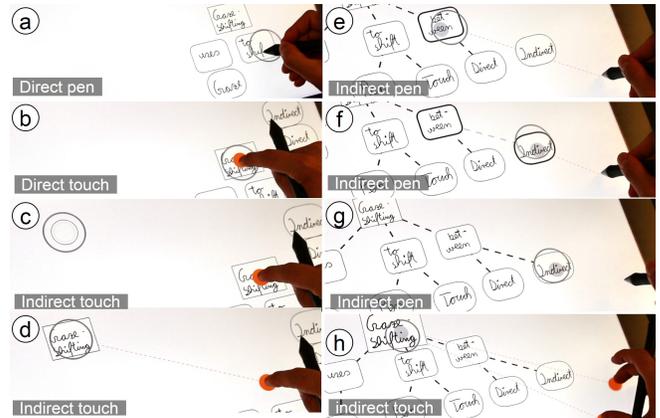

Figure 14: Users can directly draw on notes with the pen (a). Then, users can transfer notes by directly touching the note (b), looking at the transfer destination (c), where it is transferred to (d). Next, users can create associations between notes, by looking at a note and indirect pen down (e), looking at the second note (f), and pen up (g). Lastly, users can perform indirect touch manipulation on notes (h).

sequentially as presented in Figure 15. Users were instructed to use few lines, e.g. four lines for the first shape (2 straight lines, 2 lines to round). Each shape appeared at the screen center with an average size of 373x329px. User instructions included what users can do with direct/indirect pen, and how they can switch between them.

Figure 15: Used shapes in the user study for tracing and puzzle tasks

Touch (Figure 16b): The puzzle task investigates shifting between direct touch for RST and indirect touch for object transfer. The reverse transfer technique as shown in Figure 14 was used: look at the remote target (d), touch down locally (c), look at this touch point and the object snaps to it (b). Each shape of Figure 15 was cut in six pieces, randomly placed and oriented, and users had to solve the puzzle (one shape at a time). The six pieces were scattered at the screen's top area, which users transferred to the screen's bottom area to solve the puzzle with direct touch RST gestures.
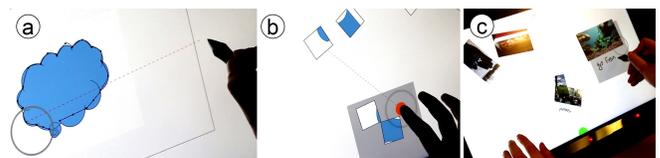
Figure 16: Study tasks for pen (a), touch (b), and both (c)

Pen and Touch (Figure 16c): The image annotation task investigates the shifting between direct pen and indirect touch input. Five images were presented, piled together in the display's top left area. Indirect touches trigger RST image manipulation on the gaze target (like Fig. 13, b-c). The task includes: indirect touch to move an image to the pen, direct pen to annotate it, and then indirect touch to drag it back to

the pile. Users did this successively for five images, fostering frequent switching between direct pen inking and indirect image movement. Users repeated this task four times.

Overall, the study started with the two individual pen/touch tasks to get familiar with the single modalities, and ended with the combined pen and touch task. After each task, users rated six Likert scale questions (c.f. Fig. 17) about the act of shifting between direct and indirect input. A short interview followed up about occuring errors, users' (dis-) likes, and potential applications ideas. Toolbars and non-relevant input modes were disabled. Each user session lasted ≈30 minutes.

12 right-handed users from 23 to 34 years old participated (M=29.6, SD=5.2, 3 female). On a scale between 1 (None) to 5 (Expert), on average users were experienced with multi-touch (M=4.1, SD=1.1), moderately experienced with stylus (M=3.2, SD=1.2) and gaze interaction (M=3.1, SD=1.5), and less experienced with combined pen and touch (M=2, SD=1.3). 4 users wore glasses, and 2 used contact lenses.

Results

The results of our study indicate that despite some accidental gaze-shifting activations users can quickly employ gaze-shifting as they successfully completed all tasks. On average, users rated gaze-shifting as easy to use, fast, moderately accurate, easy to learn, not mentally demanding, and useful across the tested tasks (Figure 17). Seven users had eye tracking accuracy problems when leaning forward to ink, for which users had to adjust back.
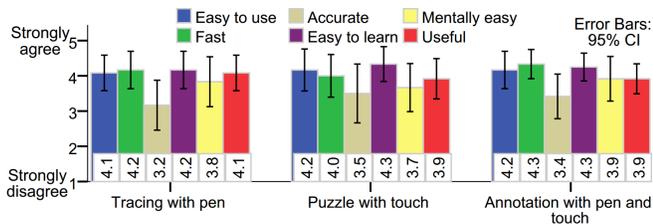

Figure 17: Likert ratings of gaze-shifting across the study tasks

Tracing with the pen
Most users needed 2-5 trials to understand the mechanism, and two users required more trials. Users liked shifting between the two modes (*"its practical, its fast and makes sketching very easy"*), and saw potential for pen mode switches (*"it would be really useful for drawing with graphics, and to change the tool or brush"*) and large displays (*"you can manipulate the points by positioning the stylus at one point, so you don't have to move the stylus over there"*). Initially, errors occurred when users held the pen just between direct and indirect input, which became less frequent with increasing experience (*"sometimes when I came closer, I was confused, but once I got used to it, it made sense"*).

Touch puzzle
Users were used to direct manipulation and initially reached out, but although users reported that they had to *"think different, you look at it and bring it with the eyes"*, all users got quickly used to it within 2-5 trials (*"you can adapt to it easily"*). One user found gaze-shifting counter-intuitive (*"it is*

disturbing as it is not what you would naturally do"), seven users did not notice any effort (*"I wasn't paying a lot of attention on switching, so it was quite fluid"*), and four users found it easier than direct touch (*"It feels like an easy way instead of dragging all across the screen"*). Users suggested use in design tools, e.g. to bring effects or files down to the designs they are working on, or add them to layers.

Pen and touch annotation
Similarly, users initially reached out but got quickly used to indirect touch manipulation. 4 users experienced erroneous image selection at overlapping images, and users suggested clear highlighting which image will be gaze-selected. Otherwise, users were positive (*"It's just easy to use, it's a kind of way that simplifies such work"*), and would find it useful in applications for browsing, categorising, and archiving of photos, potentially in collaboration with a partner.

Across tasks, users got quickly used to gaze-shifting after a short training, and experienced two types of errors. First, when users attempted to lean forward to use the pen for inking, eye tracking accuracy decreased as the eye tracker positioned at the lower bezel was partly occluded by the arm, and users moved away from the original calibration position. Multiple eye trackers (e.g., another one positioned at the top bezel) covering a larger tracking range can alleviate this problem. Second, ambiguity of the direct and indirect zone can confuse users. We expect improvements by providing users with additional feedback to make users more aware which interaction zone they are currently working with, e.g. subtle icon highlighting when users enter the indirect zone.

DISCUSSION

Gaze-shifting enables users to shift between direct and indirect input by a glance, as demonstrated by the direct-indirect interaction techniques that we developed for validation of the concept. On a pen and touch display, gaze-shifting enhances mode switching, large shape drawings, precise point editing, or pinch-to-zoom operations. These techniques highlight the flexibility of gaze-shifting's integration across task, technique, modality, and application level. In this work, we have explored gaze-shifting for pen and touch interaction, but the technique is generic and readily extensible to other direct input devices. For example, it could be used in conjunction with tangibles to switch between direct input to manipulate the tangible's state, and indirect input where the tangible devices become a proxy for remote input.

Exposure of our techniques to users showed that gaze-shifting is easy to understand, and users were able to complete direct-indirect interaction tasks after only a few trails. User feedback also indicated limitations which in part related to general eye tracking issues, but also showed challenges arising from ambiguity of the direct and indirect zone. Users however quickly adapted to these problems, and with this successfully used gaze-shifting and found it easy to use.

In particular, specific conceptual and technical limitations need to be considered in the design of gaze-shifting. Gaze-shifting is highly dynamic, which makes hybrid interactions

possible, but also introduces potential false-positive activations. Context-awareness can help, through support of users in selecting the right mode based on their input techniques and task. For instance with static menus, gaze-shifting is only enabled when users look at the fixed menu position on the UI, and otherwise (when not looking at the menu) gaze-shifting remains inactive (Fig. 7). For dynamic objects such as graphics, the system constantly determines direct/indirect input based on proximity of gaze and input position (Fig. 8) — which in principle can be used as an overarching direct/indirect mechanism over default UIs. In other cases however, interactions might not be necessarily coupled to the user's gaze (e.g. file dragging), and then gaze-shifting should be inactive. A technical factor to consider is eye tracking imprecision that affects target acquisition, particularly for overlapping targets. Hover-enabled pens counter this issue: gaze selects the area, pen hover selects the detail position, and pen tap manipulates the target (Fig. 8). Hoverless devices such as touch can integrate additional precision techniques [30, 39].

We have focused on enabling direct input devices to shift to indirect input, however, we also consider generalising gaze-shifting to indirect input devices. While indirect input devices are normally used for interaction with remote displays, the act of looking at the input device can enable additional device-specific interactions. For example, a laptop's touchpad can become a touchscreen when looked, providing an additional menu to quickly switch between applications, conceptually similar to our colour select & set example (Figure 10). For a mouse, when users look at the device, displays like those in the LensMouse [37] can be toggled interactive, and enable mouse-specific configurations. For a virtual or physical keyboards, users can shift between standard key input when looking at the normal screen, and when looking at the keyboard, they interact with on-board displays (e.g., a display like in Touch-Display-Keyboards [8]). Therefore in principle, gaze-shifting can apply beyond direct input devices, and raises potential interaction possibilities on both ends of the direct/indirect input device spectrum.

CONCLUSION
We presented gaze-shifting as a generic and implicit method to enable combined direct and indirect interaction. With a single manual input (e.g. a tap or pen click), users can perform direct manipulation, indirect control, and transition in between them. This leads to new interactive experiences where direct and indirect inputs are not separate, but continuous within the interaction. Our line tracing example has the closest fit between both inputs: users can directly start a line, extend the line to any remote gaze point, return to continue with direct manipulation, and so on. Yet the closer we bring direct and indirect together, the less is input distinguishable by this categorisation. Eventually, we could arrive at a point where direct or indirect becomes irrelevant, as any device or object is usable for both inputs, and it all coalesces into one interaction driven by our gaze.

REFERENCES
1. Martha Abednego, Joong-Ho Lee, Won Moon, and Ji-Hyung Park. 2009. I-Grabber: Expanding Physical Reach in a Large-display Tabletop Environment Through the Use of a Virtual Grabber. In *ITS*. ACM, 61–64.

2. Pär-Anders Albinsson and Shumin Zhai. 2003. High precision touch screen interaction. In *CHI*. ACM, 105–112.

3. Amartya Banerjee, Jesse Burstyn, Audrey Girouard, and Roel Vertegaal. 2011. Pointable: an in-air pointing technique to manipulate out-of-reach targets on tabletops. In *ITS*. ACM, 11–20.

4. Hrvoje Benko, Andrew D. Wilson, and Patrick Baudisch. 2006. Precise selection techniques for multi-touch screens. In *CHI*. ACM, 1263–1272.

5. Anastasia Bezerianos and Ravin Balakrishnan. 2005. The vacuum: facilitating the manipulation distant objects. In *CHI*. ACM, 361–370.

6. Xiaojun Bi, Tomer Moscovich, Gonzalo Ramos, Ravin Balakrishnan, and Ken Hinckley. 2008. An Exploration of Pen Rolling for Pen-based Interaction. In *UIST*. ACM, 191–200.

7. Renaud Blanch and Michaël Ortega. 2009. Rake Cursor: Improving Pointing Performance with Concurrent Input Channels. In *CHI*. ACM, 1415–1418.

8. Florian Block, Hans Gellersen, and Nicolas Villar. 2010. Touch-display Keyboards: Transforming Keyboards into Interactive Surfaces. In *CHI*. ACM, 1145–1154.

9. Peter Brandl, Clifton Forlines, Daniel Wigdor, Michael Haller, and Chia Shen. 2008. Combining and Measuring the Benefits of Bimanual Pen and Direct-touch Interaction on Horizontal Interfaces. In *AVI*. ACM, 154–161.

10. Connor Dickie, Jamie Hart, Roel Vertegaal, and Alex Eiser. 2006. LookPoint: An Evaluation of Eye Input for Hands-free Switching of Input Devices Between Multiple Computers. In *OZCHI*. ACM, 119–126.

11. Clifton Forlines and Ravin Balakrishnan. 2008. Evaluating Tactile Feedback and Direct vs. Indirect Stylus Input in Pointing and Crossing Selection Tasks. In *CHI*. ACM, 1563–1572.

12. Clifton Forlines, Daniel Vogel, and Ravin Balakrishnan. 2006. HybridPointing: Fluid Switching Between Absolute and Relative Pointing with a Direct Input Device. In *UIST*. ACM, 211–220.

13. Clifton Forlines, Daniel Wigdor, Chia Shen, and Ravin Balakrishnan. 2007. Direct-touch vs. Mouse Input for Tabletop Displays. In *CHI*. ACM, 647–656.

14. Tovi Grossman, Ken Hinckley, Patrick Baudisch, Maneesh Agrawala, and Ravin Balakrishnan. 2006. Hover Widgets: Using the Tracking State to Extend the Capabilities of Pen-operated Devices. In *CHI*. ACM, 861–870.

15. Yves Guiard. 1987. Asymmetric Division of Labor in Human Skilled Bimanual Action: The Kinematic Chain as a Model. In *Journal of Motor Behavior*. 486–517.

16. Khalad Hasan, Xing-Dong Yang, Andrea Bunt, and Pourang Irani. 2012. A-coord Input: Coordinating Auxiliary Input Streams for Augmenting Contextual Pen-based Interactions. In *CHI*. ACM, 805–814.

17. Ken Hinckley, Michel Pahud, Hrvoje Benko, Pourang Irani, François Guimbretière, Marcel Gavriliu, Xiang 'Anthony' Chen, Fabrice Matulic, William Buxton, and Andrew Wilson. 2014. Sensing Techniques for Tablet+Stylus Interaction. In *UIST*. ACM, 605–614.

18. Ken Hinckley and Daniel Wigdor. 2012. The Human-computer Interaction Handbook. Taylor & Francis, Chapter Input Technologies and Techniques, 151–168.

19. Ken Hinckley, Koji Yatani, Michel Pahud, Nicole Coddington, Jenny Rodenhouse, Andy Wilson, Hrvoje Benko, and Bill Buxton. 2010. Pen + Touch = New Tools. In *UIST*. ACM, 27–36.

20. Masatomo Kobayashi and Takeo Igarashi. 2008. Ninja Cursors: Using Multiple Cursors to Assist Target Acquisition on Large Screens. In *CHI*. ACM, 949–958.

21. Yang Li, Ken Hinckley, Zhiwei Guan, and James A. Landay. 2005. Experimental Analysis of Mode Switching Techniques in Pen-based User Interfaces. In *CHI*. ACM, 461–470.

22. David C. McCallum and Pourang Irani. 2009. ARC-Pad: Absolute+Relative Cursor Positioning for Large Displays with a Mobile Touchscreen. In *UIST*. ACM, 153–156.

23. Anne Collins McLaughlin, Wendy A. Rogers, and Arthur D. Fisk. 2009. Using Direct and Indirect Input Devices: Attention Demands and Age-related Differences. *ACM Trans. Comput.-Hum. Interact.* 16, 1, Article 2 (April 2009), 15 pages.

24. Thomas P. Moran, Patrick Chiu, and William van Melle. 1997. Pen-based Interaction Techniques for Organizing Material on an Electronic Whiteboard. In *UIST*. ACM, 45–54.

25. Miguel A. Nacenta, Patrick Baudisch, Hrvoje Benko, and Andy Wilson. 2009. Separability of Spatial Manipulations in Multi-touch Interfaces. In *GI*. CIPS, 175–182.

26. J. Karen Parker, Regan L. Mandryk, and Kori M. Inkpen. 2005. TractorBeam: Seamless Integration of Local and Remote Pointing for Tabletop Displays. In *GI*. Canadian Human-Computer Communications Society, 33–40.

27. Ken Pfeuffer, Jason Alexander, Ming Ki Chong, and Hans Gellersen. 2014. Gaze-touch: Combining Gaze with Multi-touch for Interaction on the Same Surface. In *UIST*. ACM, 509–518.

28. Eric Saund and Edward Lank. 2003. Stylus Input and Editing Without Prior Selection of Mode. In *UIST*. ACM, 213–216.

29. D. Schmidt, F. Block, and H. Gellersen. 2009. A comparison of direct and indirect multi-touch input for large surfaces. In *Proc. INTERACT*. 582–594.

30. Sophie Stellmach and Raimund Dachselt. 2012. Look & Touch: Gaze-supported Target Acquisition. In *CHI*. ACM, 2981–2990.

31. Sophie Stellmach and Raimund Dachselt. 2013. Still Looking: Investigating Seamless Gaze-supported Selection, Positioning, and Manipulation of Distant Targets. In *CHI*. ACM, 285–294.

32. Jayson Turner, Jason Alexander, Andreas Bulling, and Hans Gellersen. 2015. Gaze+RST: Integrating Gaze and Multitouch for Remote Rotate-Scale-Translate Tasks. In *CHI*. ACM, 4179–4188.

33. Jayson Turner, Jason Alexander, Andreas Bulling, Dominik Schmidt, and Hans Gellersen. 2013. Eye Pull, Eye Push: Moving Objects between Large Screens and Personal Devices with Gaze & Touch. In *INTERACT*. Springer, 170–186.

34. Jayson Turner, Andreas Bulling, Jason Alexander, and Hans Gellersen. 2014. Cross-device Gaze-supported Point-to-point Content Transfer. In *ETRA*. ACM, 19–26.

35. Daniel Vogel and Patrick Baudisch. 2007. Shift: A Technique for Operating Pen-based Interfaces Using Touch. In *CHI*. ACM, 657–666.

36. Daniel Wigdor, Hrvoje Benko, John Pella, Jarrod Lombardo, and Sarah Williams. 2011. Rock & Rails: Extending Multi-touch Interactions with Shape Gestures to Enable Precise Spatial Manipulations. In *CHI*. ACM, 1581–1590.

37. Xing-Dong Yang, Edward Mak, David McCallum, Pourang Irani, Xiang Cao, and Shahram Izadi. 2010. LensMouse: Augmenting the Mouse with an Interactive Touch Display. In *CHI*. ACM, 2431–2440.

38. Shumin Zhai, Carlos Morimoto, and Steven Ihde. 1999. Manual and gaze input cascaded (MAGIC) pointing. In *CHI*. ACM, 246–253.

39. Xinyong Zhang, Xiangshi Ren, and Hongbin Zha. 2008. Improving Eye Cursor's Stability for Eye Pointing Tasks. In *CHI*. ACM, 525–534.