Gaze-touch: Combining Gaze with Multi-touch for Interaction on the Same Surface

Ken Pfeuffer, Jason Alexander, Ming Ki Chong, Hans Gellersen
Lancaster University
Lancaster, United Kingdom
{k.pfeuffer, j.alexander, m.chong, h.gellersen}@lancaster.ac.uk

Figure 1: Users select by gaze, and manipulate with multi-touch from anywhere (a). This can enable seamless switching between indirect (a) and direct manipulation (b), implicit mode switching during direct-touch tasks (c), zooming into map locations the user looks at (d), and dragging multiple targets that are out of the hand's reach (e). The gray cursor indicates the user's gaze.

ABSTRACT

Gaze has the potential to complement multi-touch for interaction on the same surface. We present gaze-touch, a technique that combines the two modalities based on the principle of "gaze selects, touch manipulates". Gaze is used to select a target, and coupled with multi-touch gestures that the user can perform anywhere on the surface. Gaze-touch enables users to manipulate any target from the same touch position, for whole-surface reachability and rapid context switching. Conversely, gaze-touch enables manipulation of the same target from any touch position on the surface, for example to avoid occlusion. Gaze-touch is designed to complement direct-touch as the default interaction on multi-touch surfaces. We provide a design space analysis of the properties of gaze-touch versus direct-touch, and present four applications that explore how gaze-touch can be used alongside direct-touch. The applications demonstrate use cases for interchangeable, complementary and alternative use of the two modes of interaction, and introduce novel techniques arising from the combination of gaze-touch and conventional multi-touch.

Author Keywords

Gaze input; multi-touch; multimodal UI; interactive surface

ACM Classification Keywords

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

UIST'14, October 05 - 08 2014, Honolulu, HI, USA

http://dx.doi.org/10.1145/2642918.2647397

INTRODUCTION

As eye tracking is maturing, gaze input can become as widely available for interaction with surfaces as multi-touch is presently. In past HCI research, gaze has often been considered as an alternative to default modalities [6, 10, 14, 16] although it has also been argued that gaze might have greater potential as an addition to other modalities [25]. In this work, we explore how gaze can complement multi-touch to advance interaction on the same surface.

We present gaze-touch, a technique that integrates the gaze and touch modalities with a clear division of labour: *gaze selects, touch manipulates*. Gaze-touch is best explained in comparison with direct-touch interaction, which normally involves: (i) moving the hand to the target, (ii) touching down on the target to select it, and (iii) direct manipulation with the fingers on the surface. Gaze-touch, in contrast, is based on (i) looking at the target, (ii) touching down anywhere on the surface to select it, and (iii) manipulation with the fingers on the surface (but displaced from the target, Figure 1a).

Gaze-touch spatially separates the hand from the target. The potential utility of this separation can be considered from two viewpoints:

- More expressive input from the same touch position (Fig. 2): finger touches in the same position can resolve to selection of any point on the surface. Without moving their hands out of position, users can reach and select any position on the surface, and rapidly switch context using their gaze.
- More expressive input to the same target (Fig. 3): the same target can be manipulated from different positions on the surface. Users can move their hands off an object but continue to manipulate it with their hands "out of the way". This can help address occlusion, and also enable novel in-

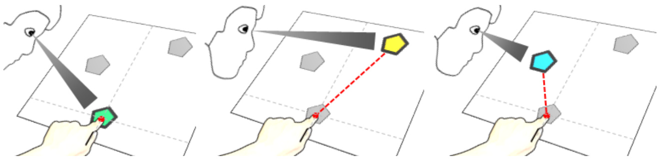

Figure 2: More expressive input from the same touch position: three examples of users touch on the same touch position, but each time manipulate a different target.

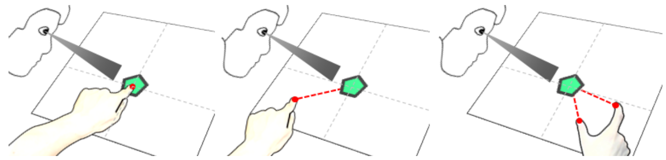

Figure 3: More expressive input to the same target: three examples of manipulating the same target that the user sees, but each time with different touches on the surface.

direct manipulation techniques, for instance with variable control-display gains to adjust precision of input.

The idea of gaze-touch is to complement direct-touch. Our focus in this work is therefore to understand how these two modes of interaction compare, and how gaze-touch can be employed alongside direct-touch. For this reason, we first characterize gaze-touch in comparison to direct-touch input through an analytical discussion of their interaction properties. The second part of the paper demonstrates four different applications that explore how gaze-touch can be used in relation to direct-touch:

1. Gaze-touch *or* direct-touch. The Image Gallery application allows users to manipulate the same image indirectly by gaze-touch (gaze and touch are separate (Figure 1a)), or directly with direct-touch (users look and touch at the same target (b)).
2. Gaze-touch *and* direct-touch. The Paint application allows users to draw and manipulate primitive shapes with direct-touch on the main canvas, and switch e.g. colour mode on the menu through gaze-touch (Figure 1c).
3. Gaze-touch *instead of* direct-touch. The Map Navigation application allows users to zoom into their gaze location instead of where they touch (Figure 1d).
4. Gaze-touch *extends* direct-touch. The Multiple Objects application allows users to quickly select and drag multiple targets anywhere on the surface (Figure 1e).

Our work makes four contributions. First, we introduce gaze-touch as a novel mode of interaction to complement direct interaction on the same interactive surface. Second, we analyse the design space of gaze-touch in comparison to default direct-touch interaction. Third, we demonstrate how gaze-touch complements direct-touch in four application examples. Fourth, we present nine interaction techniques that are based on gaze-touch and introduced with the applications.

RELATED WORK

Related work of gaze-touch can be regarded from three perspectives: multimodal gaze based interaction, gaze and touch based interaction, and indirect multi-touch interaction.

Although gaze has shown efficient pointing speed faster than any other input device [11, 16], it suffers from not having a natural mechanism to confirm a selection ('Midas Touch', [10]). To approach this issue, gaze is often complemented with a second modality in order to add selection confirmation. The second modality can be, for example, voice [12], mouse and keyboard (e.g., [10, 25]), hand gestures [12, 15], or touch [19, 20, 21, 22]. Notably, Zhai et al.'s gaze and mouse hybrid presented work where gaze was firstly used to improve performance of manual pointing, in essence making the point that gaze may have a better part to play in advancing other modalities than in replacing them [25]. Our approach follows in the same spirit: looking to enhance multi-touch with gaze, rather than pursuing gaze as an alternative.

While multi-touch has emerged as a new dominant paradigm on a wide range of devices from phones and tablets to tabletops and interactive walls, there has been little work on integration with gaze. Stellmach and Dachselt employed touch on a handheld device to assist with gaze acquisition and manipulation of targets on a remote screen [19, 20]. Turner et al. studied the same combination of devices, and combined gaze selection on remote screens with touch gestures on the handheld device to support transfer of content [21, 22]. Our work is distinct from these prior works on gaze and touch in four aspects. First, we use gaze to advance established direct interaction, e.g. by providing solutions for occlusion or fatigue issues. Prior work focused on interaction over distance where these issues do not occur. Second, we present techniques that leverage gaze and multi-touch on one large surface, that affords flexible multi-touch input with both hands, and seamless transitions between gaze-touch and direct-touch modes of interaction. In contrast, prior work was based on separated input (handheld) and output (remote display) where touch was constrained to single-point and two-point input (two thumbs, [20]). Third, our techniques consistently use the division of *gaze selects, touch manipulates*, while prior work applied gaze for positioning of targets. Forth, our techniques are grounded in a design space analysis of gaze-touch in comparison to conventional direct interaction.

Previous research on multi-touch surfaces has contributed techniques that complement the default direct-touch with means for providing indirect input in order to address problems of reach and occlusion. For example, Albinsson and Zhai, and Benko et al. proposed dual finger techniques to select targets more precisely [2, 4]. These techniques can improve the acquisition of small targets, and increase the precision of their manipulation. Banerjee et al. used in-air pointing above the surface to reach remote targets on tabletops [3]. Further research suggested widgets that are specifically designed for remote selection with touch, such as The Vacuum [5], or I-Grabber [1]. In Rock & Rails, proxies to the target were created where the first hand as a fist selects the proxy's position, and the second hand selects the target [23]. In general, these approaches require management of these indirect handles, augment the user interface, or require multi-finger or bimanual input for single target selections. Conse-

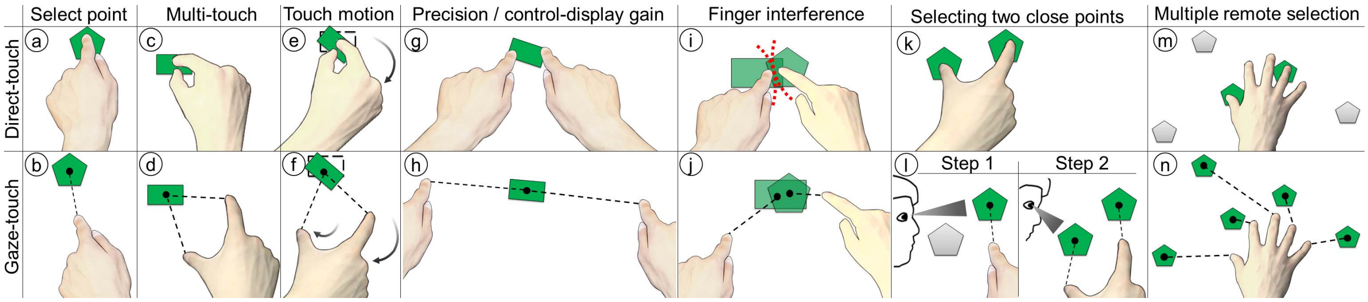

Figure 4: Illustrated differences between gaze-touch and direct-touch.

PROPERTY	DIRECT-TOUCH		GAZE-TOUCH	
Manipulation start time	**Direct** (manipulate the moment of touch down)			
Manipulation location	**Direct**	(touch point is point of manipulation)	**Indirect**	(point of manipulation is remote from touch)
Manipulation motion	**Similar** (manipulate with similar hand motion)			
Remote targets	**Low**	(only targets in physical reach)	**High**	(reach any target by look)
Occlusion	**Moderate**	('fat-finger') to **Large** (palm, pinch, hand)	**Low**	(object separate from touch)
Precision of selection	**Moderate**	(precise, but 'fat-finger')	**Moderate**	(no 'fat-finger', but gaze imprecision)
Precision of manipulation	**Moderate**	(usually control-display ratio of 1)	**High**	(control-display ratio through finger distance, that user can adjust)
Physical feedback	**High**	(finger/hand indicate current manipulation)	**Low**	(finger/hand separate from manipulation point)
Physical fatigue	**Moderate**	(move hand / arm)	**Low**	(look, and little hand / arm movement)
Physical interference	**High**	(multiple fingers/hands in same location)	**Low**	(fingers/users can be remote)
Acquisition time	**Moderate**	(move finger to position then touch down)	**Low**	(look and touch down anywhere)
Speed of selection of multiple objects within hand's reach	**High**	(select multiple objects at once)	**Low**	(Must sequentially select each object by gaze & touch)
Selection of multiple objects out of hand's reach	**—**	(needs two hands or other indirect method)	**High**	(multiple remote targets can be selected with sequential gaze & touch)
Degrees of freedom per point	**Low**	(1 touch per point)	**High**	(multiple touches can map to one gaze point)

Table 1: Summary of the differences of direct-touch to gaze-touch.

quently the point in time when manipulation starts is delayed, and effort increased. In comparison to these indirect methods, gaze-touch is also spatially indirect as the touch is separated from the object position. However, gaze-touch is different as manipulation can start directly at touch down, similar to direct-touch input. This enables the speed of direct-touch selection, while at the same time gaining indirect properties such as minimizing hand movement, enabling remote manipulation, or avoiding occlusion.

DESIGN SPACE: DIRECT-TOUCH VS. GAZE-TOUCH

To gain a deeper understanding of the conceptual differences between direct-touch and gaze-touch, we analyse the two techniques. We provide a design space analysis under the following headings, without claiming completeness: *similarities*, *occlusion*, *precision of selection*, *precision of manipulation*, *physical aspects*, *multiple object selection*, and *multi-touch to one point*. Table 1 provides a summary of the comparison and Figure 4 illustrates the conceptual differences.

Similarities. Both gaze-touch and direct-touch are temporally direct, as manipulation of an object starts as soon as users touch the surface. Both techniques accept a single touch point for 'clicking' an object (see Figure 4a & 4b), and two touch points for manipulating an object (Figure 4c & 4d). Gaze-touch uses the same hand motion for object manipulation; e.g. rotating two touch points to rotate a selected object (see Figure 4e & 4f), and pinch gestures to scale (Figure 4g & 4h). These similarities enable ease of learning and preserve consistency, as users can transfer their knowledge of direct-touch for operation of gaze-touch.

Occlusion. A direct-touch gesture causes surface occlusion, because users place their hands on top of an object for selection. As users place more fingers on an object, the area of occlusion increases (see Figure 4c). Researchers have suggested indirect methods that avoid occlusion, like creating proxies to the objects [23]. However, these methods can lead to additional effort for users, and can delay the manipulation task. Gaze-touch prevents occlusions by enabling spatially indirect manipulation (Figure 4d). Since touch actions are disjoint from the gaze-selected object, users can touch down on any surface location while looking at the object.

Precision of selection. Using direct-touch for target selection can be problematic when the target's size is smaller than the user's finger [9]; known as the 'fat-finger' problem. Although researchers suggested techniques to alleviate this problem by using multiple touch points (e.g. [2, 4]), the use of multiple fingers or hands hinders the selection process. Using gaze for selection in principle can overcome this issue. However, our eyes naturally jitter, and inaccuracy of eye trackers can cause imprecision [26]. Touch is still more precise for single-finger taps on large objects, but gaze-touch is potentially more suitable when the interaction requires placement of multiple fingers on an object (see Figure 4c & 4d).

Precision of manipulation. The precision of manipulation differs between gaze-touch and direct-touch. The standard direct-touch model is based on a 1:1 control-display ratio, so fine-grained manipulations can become difficult as they require tiny and precise movements. In practice, the size of objects has a limit; an object becomes difficult to manipulate if its size is too small to be selected or manipulated with fingers (Figure 4g). The standard touch technique could be improved by having users first select a target and then put their fingers elsewhere to manipulate (like the Rock&Rails technique [23]). The necessity to select and deselect the object complicates the interaction and delays the manipulation. In contrast, gaze-touch allows users to draw their fingers as far apart as the screen allows, and to immediately start manipulation at the moment of touch down (see Figure 4h).

Physical aspects. In gaze-touch, the finger touch positions are detached from the gaze position. Users only see digital *feedback* in their sight radius, i.e. on the selected object. However, the fingers are probably out of the users' sight. This contrasts direct-touch, where users can see physical feedback to the selected objects, because their fingers are placed on the object. Further, detaching the touch and gaze reduces muscle *fatigue*. Users' can keep their hands within their comfortable regions and still able to manipulate gaze-selected objects. On the other hand, the active use of gaze to select targets could lead to eye fatigue, as the eyes, a channel to perceive visual content, should not be overloaded with motor tasks [25]. Another benefit of detaching gaze and touch is that it avoids finger interference. Interference can occur when multiple fingers or hands collide within the same location, which interrupts the task (Figure 4i). With gaze-touch, the objects can be separate from the finger's position, so physical collision is prevented (Figure 4j).

Multiple object selection. Gaze is a single-point input, while multi-touch supports simultaneous input from multiple points (Figure 4k). With gaze, users must select multiple targets by looking at each object and placing a touch down (Figure 4l). Although conceptually gaze selection of multiple targets is slower than direct-touch, gaze-touch yields a benefit that users can select scattered objects on a surface. Selection of multiple objects with direct-touch is limited by the distance that a hand can reach and users can only select multiple objects that are near by each other (Figure 4m). Gaze-touch in contrast eliminates this restriction (Figure 4n).

Multi-touch to one point. Gaze-touch can map multiple touch points to a single gaze point (Figure 4d). This contrasts with direct-touch where one finger can be physically mapped to one point on the screen (Figure 4a & 4k). Furthermore, a gaze-touch is invariant of the hand's posture. In a rotation gesture with direct-touch, a user fits their hand to the object's shape to then perform the rotation from this hand posture (Figure 4c & 4e). Prior work has shown that there are several occasions where rotation or scaling postures and motions can be difficult [7, 8]. Using a gaze-touch, target acquisition is more comfortable as users only look at the object and touch down remotely with any hand posture (Figure 4d & 4f).

APPLICATIONS

In the following we describe four applications that each demonstrate a specific use of gaze-touch. Each application is described in its own section. Within each application, we describe concept, interaction techniques, and implementational details. The first three applications were also part of a preliminary user study which design and setup are described once, and which task and results are described within each application section. Notably, the gray circle indicates the user's current gaze point in all figures.

APPLICATION: IMAGE GALLERY

This application demonstrates that gaze-touch *or* direct-touch can be used for the same task. Users can browse through their set of images. They can scale them up for a detailed view, rotate the images to correct the aspect ratio, and drag images across the surface for sorting, grouping, or other object manipulation tasks. In essence, users can perform two types of touch gestures: single-touch dragging, and multi-touch rotate, scale, and translate (RST). Multiples of these gestures can be performed at the same time, when using multi-finger and bimanual input.

Switching between Gaze-Touch or Direct-Touch

The switching between direct-touch and gaze-touch is accomplished through using the user's coordination between gaze and touch position. When a user looks at an image and at the same time touches on it, direct-touch is enabled. This means the touch point is used as input, and not the gaze point (Figure 5a). However, when the user looks at a target but touches down somewhere else, gaze-touch is enabled (b, c).

Interaction Techniques

In addition to standard direct-touch translate, rotate, and scale gestures, the user can perform the following gaze-touch techniques:

Accelerated-Object-Dragging

When users look at an image and touch down once remotely, they can drag the image with their finger. While the selection is similar to previous techniques for interaction over distance [20, 21, 22], this technique only uses touch dragging for positioning. This dragging of images uses a dynamic control-display gain. We implemented a dragging acceleration similar to windows XP mouse acceleration, which amplifies the speed of the dragging finger. This enables to overcome larger distances with shorter movement, and be more precise when moving the finger slowly.

Indirect-Rotate-Scale-Translate (RST)

This technique is the gaze-touch counterpart for the RST gesture. Users touch down two fingers while looking at the same image (similar to [20], however without mode-switching). It has some characteristics that are distinct to direct-touch. Users only need the gaze point to be on the image, enabling manipulation of images that are too small to directly lay multiple fingers on it (Figure 5b), and when high precision is required (c). The further the user draws apart their fingers at touch down, the more precise is the manipulation. This provides the user with a choice of how precise they want to manipulate the image: users can place their fingers very close for fast manipulation (b), or very far apart for high precision (c).

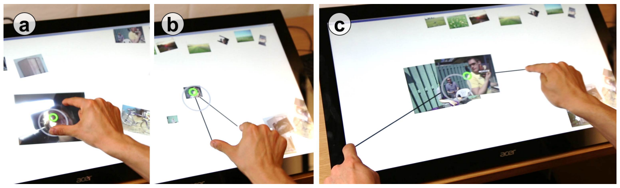

Figure 5: Indirect-RST: in addition to direct image manipulation (a), users can indirectly manipulate images for easy acquisition of small targets (b), or more precision (c).

Multi-Image-Dragging

While users can sequentially drag multiple images with the *Accelerated-Object-Dragging* technique, they can also drag multiple objects at once (Figure 6). The user first selects each image by looking at each image and each time touching down, to then perform one drag gesture. This is particularly interesting as, in contrast to direct-touch, users can simultaneously drag objects that would be out of the hand's reach.

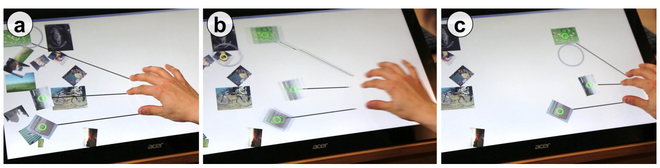

Figure 6: Multi-Image-Dragging: after multiple gaze-touch selections, users can drag them out of the pile using a single dragging gesture. Through a dynamic control-display gain, small movements can overcome large distances.

Implementational Details

The moment the user touched down, the system decides if it is a gaze-touch or a direct-touch. If the user touches on an image and does not look at another image, direct-touch is triggered. Else, gaze-touch is active. The gaze point is set as the target of manipulation of a touch input session until the user lifts their finger. Intermediately received touch events of this session (touch_update) are executed on the point of gaze that was received at the moment of touch_down (for gaze-touch, respectively). To counter inaccurate gaze data, we used target-assistance. The image is highlighted as 'looked', when the system's gaze estimate is close to the image.

An interesting case is the control-display gain for multi-touch gestures, such as two-finger scaling. In direct-touch, this case is clear as the distance between the two fingers can be mapped to the same distance on the screen, thus an absolute 1:1 control-display gain. RST with gaze-touch relates two-touch input to one gaze point, and therefore it is unclear to what display-distance it should be mapped to. In our application instance, the distance between the fingers of a two-touch gesture is mapped to the radius of the target's size.

Figure 7: The system consists of a 27" 1080p multi-touch sensitive surface (a), and the 120hz Eye Follower eye tracking device (b).

Study Design

We conducted a preliminary user study to demonstrate the feasibility of and to gather user opinions about the applications. 13 volunteers between 22 and 33 years took part in the study (M=27.9, SD=3.73, 4 female). On a scale between 1 (no experience) to 5 (very experienced), users perceived themselves as well experienced with multi-touch (M=4.3, SD=0.9), and as less experienced with eye based interaction (M=2.5, SD=1.4). After a brief introduction, users were once calibrated to the eye tracking system. Users then interacted with the applications (counterbalanced). Each application began with a short training session where the experimenter explained the interaction techniques, and ended with an interview session. Each application test and interview lasted approximately 5-10 minutes. Users were not bound to a specific performance goal of the tasks to keep it to natural usage of the interactions.

Apparatus

We use an LC Technology Eye Follower with a touchscreen that is tilted 30° toward the user to enable convenient touch reaching (Figure 7). The user's eyes were approximately 50cm in front of the screen's center. Occlusion of the eye tracking camera could occur during the use. In practice, however, mostly users bend their arms around the tracking camera's view because of the close proximity of the touchscreen. As touchscreen we used an Acer t272 27" 1080p display that allows up to 10-finger multi-touch input. The system is running at a frame rate of 60hz, on a quadcore i7 2.3GHz CPU computer. The applications are written in Java using the Multitouch For Java library[1]).

User Feedback

Users were provided with ten images and were trained using both direct-touch and gaze-touch techniques. They performed two tasks of sorting images into groups (e.g. indoor/outdoor), and two tasks of searching for an image with a specific element in it (e.g. a bus). Before each task, the images were randomly placed, rotated, and sized. Users could scale the images between 50 and 750px.

[1] Used library available at http://www.mt4j.org (16/04/2014)

All users got quickly used to the techniques in this applications. Users did not have difficulties to switch between the direct and indirect counterpart. The study showed that most users stick to one technique for each particular task:

Single-Touch Dragging. Twelve users kept on using gaze-touch after the training. Interviews revealed that their reasons were speed, ease, and less physical effort. This was considered important with multiple images, where moving back and forth for each image is avoided, as one user stated: *"you do not always have to go back with your hand, but [you] keep it [the hand] stationary while your gaze goes back to the imagepool"*. Users emphasized that gaze-touch has less physical fatigue (*"You just move your arms, not your whole body"*). Users also liked the speed of dragging (*"It is effortless to move, as you can accomplish more with less movement"*). Some users were also positive about less occlusion through their fingers (*"My fingers sometimes obscure the pictures [with direct-touch]."*).

Two-Touch RST. Seven users kept on using direct-touch and four users gaze-touch. The user who preferred direct-touch found it to be easier and more intuitive (*"It is more intuitive, the movement"*). They also stated prior knowledge of direct-touch (*"I prefer on the picture [...] based on how I use my phone"*). An interesting case occurred when these users wanted to acquire small images with two fingers. They tried to put their fingers directly on it, yet in a failed attempt they put their fingers only close to the image as it was too small. This triggered gaze-touch on the very image (users looked and touched close to it) with which users scaled it up, without being aware of a gaze-touch.

Errors. Three users stated some difficulties with overlapping images. Inaccurate gaze tracking by the hardware we used lead to false positive image selections (*"When pictures overlapped sometimes, it did not jump at the picture that I wanted"*). Another issue occurred when selecting an image to drag. The user already looked away to the dragging destination during touch down, which lead to a wrong selection (*"I already looked at where I wanted to move it before I touched, so it moved something else"*).

Specific Findings. Two users stated they used direct in front (user's comfort zone), and gaze-touch in the remaining area. They intuitively use direct-touch in close proximity, however to avoid reaching out, gaze-touch became convenient (*"When it is far from me, then I can drag it from distance. If it is close to me, I can use the picture itself"*). One user emphasized an interesting feature of gaze-touch: users can manipulate an image, even though touching on another (*"If I look at a picture, I can go anywhere with my fingers. Even if I have my fingers on another picture"*).

Summary

Our evaluation showed that having direct and indirect manipulation within the same application is feasible. The majority of users kept using gaze-touch for single-touch dragging, and direct-touch for two-touch scaling and rotation. Users acknowledged the speed, reachability, reduced movement and reduced fatigue of gaze-touch in comparison to direct-touch. However, many users preferred using direct-touch for RST gestures. They perceived it easier to perform this gesture directly on the image.

APPLICATION: PAINT

This application demonstrates how gaze-touch *and* direct-touch are used. The user is provided with standard tools of a drawing application. With direct-touch, users can draw on the main canvas of the interface. In the menu, users can create three types of primitive shape (rectangle, circle, triangle), that initially have a size of 100x100px. After creation, they can be dragged and scaled using direct-touch input. Thus the user can create figures based on individually drawn lines and these primitive shapes.

The menu is completely gaze-touch enabled (but can also be directly touched). The menu provides the functions select colour, create primitive, and copy existing primitive. To trigger a menu mode, users look at a menu icon, and select it by touching down anywhere on the surface. We believe this can have an advantage for drawing tasks, as users do not need to remove hands from their current target. And after a mode is switched on the menu with gaze-touch, users do not need to relocate the previous position of the hand to continue the task. Users can keep their hand at the drawing position, and from there perform gaze-touches to the remote menu. This concept can be applied to many applications that involve a main interactive area and remotely positioned menus, such as ribbon menus in office tools, tabs in browsing, etc.

Interaction Techniques

Remote-Colour-Select

Most actions of the user are around the main canvas, where the figure is drawn directly. From here, users can quickly change the colour through gaze-touch (Figure 8). The user looks up at the colour (a), and touches down at their current position to confirm (b). Once done, the user can continue the drawing task (c). This technique can be easily extended to multiple finger use. Users can touch down many fingers, and each time look at a different colour, to simultaneously draw with several colours. In direct-touch, the user would have to reach out to the canvas or use a second hand to apply different colours to each fingers.

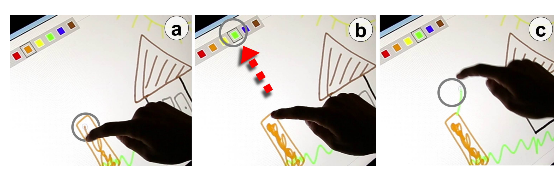

Figure 8: Remote-Colour-Select: a user draws the tree stem directly (a). The user then changes to the 'green' colour by a look at the corresponding menu icon, and a tap (b). The user directly continues drawing (c).

Create-Object

Contrary to mode changes, this technique creates a new element into the canvas. When users perform a gaze-touch on a graphical primitive icon of the menu, the primitive is created at the position of the user's touch. From here, the user can directly drag it to a more precise position, or perform direct

RST manipulation. The operation of this technique is similar to drawing (Figure 8), but instead of a colour it adds an object.

Copy-Paste-Object

Graphical primitives are direct-touch enabled in our application, thus users can drag them with single-touch on it. However, a single-touch can also be used for copy-paste of the primitive. The system switches to this special mode when users touch on the object, while they look at the copy-paste icon in the menu (Figure 9). This creates a copy directly under the user's finger, that can then be dragged elsewhere. This technique is distinct as the user is required to coordinate both the touch and gaze point. This requires more mental effort. However, this technique allows the user to perform two different tasks (dragging or copying) with a single-touch on the object, that are distinguished by where the user looks at. The technique also scales to multi-touch. Users can instead touch down two fingers to create two copies simultaneously.

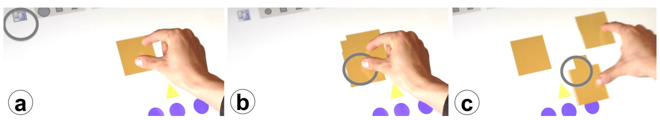

Figure 9: Copy-Paste-Object: the user can copy an existing object with a single-touch. Usually, a touch on the object leads to dragging. However, when the user looks at the copy icon in the menu (a), and then touches down on the object, the user obtains a copy of the touched object under her finger (b). Then, the user can directly drag the new copy to a desired position (c).

Implementational Details

The moment the user touched down, the system determines whether the gaze position is on one of the icons of the menu. If true, a gaze-touch is triggered, otherwise direct-touch is kept. To aid a potential inaccurate gaze position, we used target-assistance for the icons. If the gaze cursor is close to the menu, it attaches to the closest icon. No gaze cursor is shown, but the icons in the menu are highlighted when the user looks at them.

User Feedback

For the purpose of this study that investigates the switching between direct-touch and gaze-touch, we limited the interactions to direct-touch drawing on the canvas, and gaze-touch selection of colours in the menu. The task of the users was to draw a house, a tree, a car, a sun, and their name with various colours. All users were able to perform the drawing tasks.

The interviews revealed that seven users were positive, three users negative, and the other participants had mixed opinion about the application. Most users commented that the gaze-touch menu is easy to use, fast, and convenient (*"It goes quicker to select the colour [...] than by hand"*). Also it was noticed that it helps to focus on the main drawing task (*"It indirectly saves interaction, you can focus on the draw surface"*), and that it reduces mental effort (*"There is less thinking involved"*).

Negative user comments were mainly based on false positive colour selections. This had two reasons: (1) inaccuracy of eye tracking hardware, and (2) eye-hand coordination of the system. Often, users looked at a colour, but already moved on before touch down. It occurred that users passed close to other colours when looking back to the canvas, which the target-assistance wrongly interpreted as the colour of choice. This has been reported as 'Late-Trigger errors' and can be addressed by delayed selection [13].

Two users stated that they disliked the gaze-touch menu, because of mental demand (*"I feel like I have to focus"*) and non-normal behaviour (*"Often your eyes move without you knowing [that] they are moving"*).

Summary

The evaluation showed users can use direct-touch in conjunction with gaze-touch. Both techniques are used for separate areas on the screen, and therefore give the user a clear separation of input. Users recognized that gaze-touch is useful for menus that are often out of reach. They also confirmed that it is easy to use, comfortable, and contributes to better focus on the main drawing task. On the downside, our implementation led to false positive colour selections for some users (further discussed by Kumar et al. [13]).

APPLICATION: MAP NAVIGATION

This application demonstrates where gaze-touch can be used *instead of* direct-touch. The application begins with a world map, that the user can then explore with direct single-touch dragging gestures to pan the whole map, and gaze-touch based zooming to zoom in locations. To complement previous work that used gaze for interaction on maps [18], we use gaze implicitly as the target of a two-finger zooming gesture.

Gaze-Focused-Zooming

To perform zooming, the user looks at the location of interest, and then performs a pinching gesture anywhere on the surface. This triggers zooming into the user's gaze point. This yields several benefits over the direct counterpart. First, users can keep their hand on the same position for multiple zooms that reduces hand movement, occlusion, and fatigue, as only the user's gaze is used for target selection (Figure 10). Second, the user's gaze is faster than the hand for the selection of a zooming target. Third, users are able to change the zooming target during the gesture. With direct-touch, the target is fixed to the touch position once touched down. With gaze-touch, users can change the position by a glance. This becomes useful for corrective zooming: if a user zoomed into the wrong area, the user can zoom out, look at a different location, and zoom in again; all within a single pinch gesture.

Figure 10: Gaze-Focused-Zooming: users can change their zoom-in position during several zooms without changing the pinching position.

Implementational Details

Within the touch input manager, we changed the zooming target from the touch center position to the gaze position. During zooming gestures, the system receives gaze events on-line to enable dynamic changing of the zooming focus. We also added a gaze cursor for this application. To avoid distracting behavior and gaze jittery, the cursor is set large (width=250px) and we average jittering gaze samples for 150 ms when only short eye movements occur.

User Feedback

In this part of the study we let user compare direct-touch against gaze-touch zooming. Users performed both conditions (counterbalanced). In each condition, users searched for five capital cities starting from a world view. Users did not have any difficulties finding the cities. Four users stated they had to get used to the gaze-based approach within the first or first two city tasks.

Preferences. Nine users favored map navigation with gaze-touch, two users thought they were equal, and the remaining two preferred direct-touch zooming. Users preferred gaze-touch zooming because of ease, speed, less physical effort, precision, and reachability. Users commented that it is more precise and reliable, as with direct-touch *"You often zoom in a bit too close, [...] and you have to zoom out again to correct"*. Interaction with gaze-touch was perceived as easy and intuitive, since users already look where they want to zoom anyway (*"I always look at the area where I expect the city"*). A user mentioned that it is much less fatiguing in comparison to her own touch-enabled device: *"Because sometimes with the IPad you always use your hands, you get tired"*. In addition, users were positive about no occlusion through hands and less body movement (e.g. *"[With direct-touch] I cover what I see with my hand and when the area is further away I have to lean forward to zoom in with the hand"*). Two users favored direct-touch zooming. The first user thought it was more precise with direct-touch (*"It is a little vague with the eyes"*). The other user stated the gaze-cursor that is used is confusing, as it moved constantly with the user's gaze.

Gaze-Touch Experience. While some users did not notice any difference, other users perceived a different map experience with gaze-touch. For example, users stated that gaze-touch helps map navigation (*"It helps you on what you are searching, you are not distracted"*). Another user mentioned increased zooming awareness (*"I was more aware of where I zoom"*) and another user perceived it as being guided (*"It is like you are guided"*).

Summary

The majority of users preferred gaze-touch over direct-touch for zooming. Reasons were speed, less effort, precision, and reachability. Further discussions with users showed that the map navigation experience is altered; users felt it is more helpful, and increases location-awareness.

APPLICATION: MULTIPLE OBJECTS

This application demonstrates how gaze-touch *extends* direct-touch interactions. The application allows users to manipulate a large number of objects spread across the surface. It is configurable with regards to number, shape, size, and colour of objects. Users can quickly select multiple objects, and reposition them by dragging gestures. Users can touch down with up to ten fingers, that would lead to 10 object selections. This allows us to experiment with gaze-touch's capability of fast and simultaneous manipulation of objects. To overcome the physical friction of the screen and gain fluent and reliable multi-touch, we used multi-touch gloves in our demonstrations. Because of its experimental state, this application was not included in the user study. These techniques can be useful, for example, in visual analytics that commonly involve sorting, searching, or grouping of many objects [24, 17].

Implementational Details

Our goal was to optimize object dragging. Therefore a touch down will always map to the target that is closest to the user's gaze point on the screen. Further, one touch will only map to a single target. This allows to quickly select multiple objects, e.g. when touching down two fingers at once, the two objects closest to the user's gaze are selected. In addition, the dragging acceleration from the Image Gallery application is integrated.

Interaction Techniques

Instant-Multi-Object-Dragging

Users can instantly select up to five objects to a hand (Figure 11). When the user touches down, the system binds the closest object to the finger. If multiple fingers are downed, each finger will get one object associated (a). This can be useful, for example, when sorting a large amount of objects. The user can sort out all selected objects at once by a single dragging gesture (b, c). Immediately after this, the user can continue to sort out the next objects as the user only needs to look at the next objects.

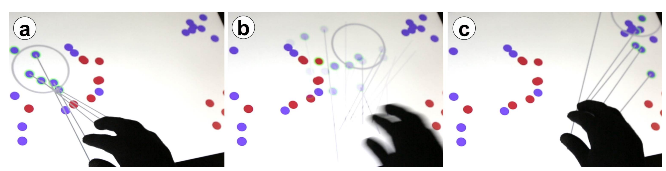

Figure 11: Multi-Finger-Dragging: users can select the five closest objects to their gaze by touching down five fingers (a). Users can then sort them out at once with a single dragging gesture (b, c).

Multi-Object-Pinching

We implemented a variant of this application where pinching leads to relative movement of objects toward the user's hand. When the user selects multiple objects as explained above, the user can perform a pinching gesture to move all objects to the hand's position (Figure 12). The distance between each finger and the center of all fingers is mapped to the distance between the object and the fingers' center. Thus this technique allows continuous movement of objects toward the hand, but moreover, it can also be used for positioning anywhere on the screen. To move close objects far apart, the user can start with a small distance between the fingers. By expanding the fingers (pinch-out), the objects would be drawn away (Figure 12, from (b) to (a)).

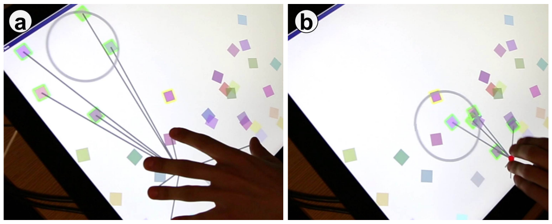

Figure 12: Multi-Object-Pinching: when multiple objects were selected (a), a pinching gesture moves the objects to the hand's position (b).

DISCUSSION

Starting from our conceptual analysis we outlined the differences between gaze-touch and direct-touch. The beneficial differences that we identified, such as reachability, no occlusion, speed, less fatigue, and less physical movement, were confirmed in our user study. Besides differences, a key characteristic of gaze-touch is its similarities to direct-touch. Users can manipulate objects at the moment they place a touch down, they can perform the same multi-touch gestures to manipulate content, and they look at the target before they initiate manipulation. These similarities greatly reduce learning effort as users are already familiar with touch-based interaction, so that they can apply the established knowledge to gaze-touch. This was further shown in our user study. Our participants required little training, and were able to get used to gaze-touch interaction very quickly.

The similarities between gaze-touch and direct-touch enable users to switch seamlessly between the techniques. Users can use direct-touch to interact with objects that are within their immediate comfort zone, while they can seamlessly switch to gaze-touch for reaching distant objects or mode switching as illustrated in our paint application. Furthermore, direct-touch enables single target manipulation by simply touching an object, and users can employ gaze-touch for multi-target operations. Our images application allows the use of both techniques; which led many participants to choose direct for single-target scaling and rotation, and gaze-touch for multi-target dragging. Our participants confirmed that these kind of divisions improve the interaction within the applications.

Our work shows many and varied potentials and examples of using gaze and touch for interactive surfaces of combined input and output. While we can confirm prior work that this combination allows to efficient reaching of remote targets [19, 20, 21, 22], we discovered additional benefits for surface interaction. A single-touch is now more expressive as it can have many different meanings – users can drag an object like in direct-touch, but also copy, delete, add, or any other task depending on which mode the user looks at. Users can perform the same task either directly or indirectly with gaze-touch, in essence providing more expressive input to the same target. Techniques can take advantage of both gaze and touch point, e.g. drag objects to the close touch position, or copy the object that is under the touch. Multiple target manipulations are more efficient. Users look at each target and perform manipulation on the same position, such as zooming into different locations, or manipulate all targets in sight at once, such as sorting of multiple images across the surface.

Limitations

Eye Tracking

In our setup, the position of the eye tracker is non-trivial because users can occlude the camera's view. When users positioned their arms in front of the eye tracker, the action can block the tracking of the users' eyes. Another problem is eye tracking inaccuracy by hardware limits and natural eye jittering, that can increase with a larger surface space [11]. We approached this issue individually for each application: e.g. target assistance when objects were involved (e.g. the menu of Paint application), or by filtering gaze noise (Map Navigation application), however further improvements can allow a smoother gaze-touch experience.

Inappropriate Tasks

A conceptual limitation of gaze-touch is that it requires the user to look at a target of interest. For many tasks the user's gaze is already at the target of interest, but there are cases where users do not need to look at the target. For example, when users are familiar with the input position, they simply use their muscle memory for input (e.g. PIN entry). This example, however, only applies to input targets that are fixed in location, and in this case gaze-touch can simply be disabled. In other cases however, where content is dynamic e.g. image aligning, video editing, or multi-view interfaces, the use of gaze-touch might become difficult. In these cases gaze-touch is more of benefit when used complementary to direct-touch, e.g. as shown in our Paint application (gaze-touch for mode switching, direct-touch for primary task).

Eye-Hand Coordination

Eye-hand coordination plays a relevant role in gaze-touch. Often users already gaze away from the target before acquisition. Known as the 'Late-Trigger errors' [13], it can be approached by selection delay or intelligent eye fixation detection, however a deeper understanding might be needed.

Multiple Selection and Eye Overload

A gaze-touch selection is completely based on the single-channel gaze modality. This principally disallows simultaneous selection of multiple targets. One approach is selecting as many objects close to the user's gaze as the user touches down fingers (c.f. our 'Multiple Objects' application). However, when sequences of tasks require users to visually fixate many points over time, the users' cognitive or visual abilities might get overloaded. While our principle 'gaze selects, touch manipulates', reduces gaze usage to the moment when users touch down, it is yet unknown how much it affects the user's mental and physical abilities. In this context, it has to be considered that the utility of gaze-touch is its complementary nature, in cases direct-touch is limited.

CONCLUSION

In this paper we introduced gaze-touch as a novel interaction technique that faciliates gaze and multi-touch on the same surface. The technique makes existing direct interactions more flexible, as it allows for implicit mode switching by a glance, and manipulation of many targets without directly touching them. This leads to novel application designs where gaze-touch can be used complementary or alternately to existing direct manipulation, and even can replace or extend tasks that previously belonged to the territory of direct input. Gaze-touch enhances touch interactions with seamless and efficient interaction techniques, as reachability, physical movement and fatigue are overcome, while the speed and familiarity with common multi-touch gestures prevail. Gaze-touch is simple in its core technique, but lends itself to extend surface interactions with dynamic and effortless capabilities.

REFERENCES

1. Abednego, M., Lee, J.-H., Moon, W., and Park, J.-H. I-grabber: Expanding physical reach in a large-display tabletop environment through the use of a virtual grabber. In *ITS '09*, ACM (2009), 61–64.
2. Albinsson, P.-A., and Zhai, S. High precision touch screen interaction. In *CHI '03*, ACM (2003), 105–112.
3. Banerjee, A., Burstyn, J., Girouard, A., and Vertegaal, R. Pointable: an in-air pointing technique to manipulate out-of-reach targets on tabletops. In *ITS '11*, ACM (2011), 11–20.
4. Benko, H., Wilson, A. D., and Baudisch, P. Precise selection techniques for multi-touch screens. In *CHI '06*, ACM (2006), 1263–1272.
5. Bezerianos, A., and Balakrishnan, R. The vacuum: facilitating the manipulation distant objects. In *CHI '05*, ACM (2005), 361–370.
6. Hansen, J. P., Tørning, K., Johansen, A. S., Itoh, K., and Aoki, H. Gaze typing compared with input by head and hand. In *ETRA '04*, ACM (2004), 131–138.
7. Hoggan, E., Nacenta, M., Kristensson, P. O., Williamson, J., Oulasvirta, A., and Lehtiö, A. Multi-touch pinch gestures: Performance and ergonomics. In *ITS '13*, ACM (2013), 219–222.
8. Hoggan, E., Williamson, J., Oulasvirta, A., Nacenta, M., Kristensson, P. O., and Lehtiö, A. Multi-touch rotation gestures: Performance and ergonomics. In *CHI '13*, ACM (2013), 3047–3050.
9. Holz, C., and Baudisch, P. The generalized perceived input point model and how to double touch accuracy by extracting fingerprints. In *CHI '10*, ACM (2010), 581–590.
10. Jacob, R. J. K. What you look at is what you get: eye movement-based interaction techniques. In *CHI '90*, ACM (1990), 11–18.
11. Jacob, R. J. K. Eye movement-based human-computer interaction techniques: Toward non-command interfaces. In *Advances in Human-Computer Interaction*, Vol. 4, Ablex Publishing (1993), 151–190.
12. Koons, D. B., Sparrell, C. J., and Thorisson, K. R. Intelligent multimedia interfaces. American Association for Artificial Intelligence, Menlo Park, CA, USA, 1993, ch. Integrating Simultaneous Input from Speech, Gaze, and Hand Gestures, 257–276.
13. Kumar, M., Klingner, J., Puranik, R., Winograd, T., and Paepcke, A. Improving the accuracy of gaze input for interaction. In *ETRA '08*, ACM (2008), 65–68.
14. Mateo, J. C., San Agustin, J., and Hansen, J. P. Gaze beats mouse: Hands-free selection by combining gaze and emg. In *CHI EA '08*, ACM (2008), 3039–3044.
15. Pouke, M., Karhu, A., Hickey, S., and Arhippainen, L. Gaze tracking and non-touch gesture based interaction method for mobile 3d virtual spaces. In *OzCHI '12*, ACM (2012), 505–512.
16. Sibert, L. E., and Jacob, R. J. K. Evaluation of eye gaze interaction. In *CHI '00*, ACM (2000), 281–288.
17. Stasko, J., Görg, C., and Liu, Z. Jigsaw: Supporting investigative analysis through interactive visualization. *Information Visualization 7*, 2 (Apr. 2008), 118–132.
18. Stellmach, S., and Dachselt, R. Investigating gaze-supported multimodal pan and zoom. In *ETRA '12*, ACM (2012), 357–360.
19. Stellmach, S., and Dachselt, R. Look & touch: gaze-supported target acquisition. In *CHI '12*, ACM (2012), 2981–2990.
20. Stellmach, S., and Dachselt, R. Still looking: investigating seamless gaze-supported selection, positioning, and manipulation of distant targets. In *CHI '13*, ACM (2013), 285–294.
21. Turner, J., Alexander, J., Bulling, A., Schmidt, D., and Gellersen, H. Eye pull, eye push: Moving objects between large screens and personal devices with gaze & touch. In *INTERACT '13*, Springer (2013), 170–186.
22. Turner, J., Bulling, A., Alexander, J., and Gellersen, H. Cross-device gaze-supported point-to-point content transfer. In *ETRA '14*, ACM (2014), 19–26.
23. Wigdor, D., Benko, H., Pella, J., Lombardo, J., and Williams, S. Rock & rails: Extending multi-touch interactions with shape gestures to enable precise spatial manipulations. In *CHI '11*, ACM (2011), 1581–1590.
24. Wise, J. A., Thomas, J. J., Pennock, K., Lantrip, D., Pottier, M., Schur, A., and Crow, V. Visualizing the non-visual: Spatial analysis and interaction with information from text documents. In *INFOVIS '95*, IEEE (1995), 51–58.
25. Zhai, S., Morimoto, C., and Ihde, S. Manual and gaze input cascaded (magic) pointing. In *CHI '99*, ACM (1999), 246–253.
26. Zhang, X., Ren, X., and Zha, H. Improving eye cursor's stability for eye pointing tasks. In *CHI '08*, ACM (2008), 525–534.