Combining Gaze with Manual Interaction to Extend Physical Reach

Jayson Turner
Lancaster University
j.turner@lancaster.ac.uk

Andreas Bulling
University of Cambridge
Lancaster University
andreas.bulling@acm.org

Hans Gellersen
Lancaster University
hwg@comp.lancs.ac.uk

ABSTRACT
Situated public displays and interactive surfaces are becoming ubiquitous in our daily lives. Issues arise with these devices when attempting to interact over a distance or with content that is physically out of reach. In this paper we outline three techniques that combine gaze with manual hand-controlled input to move objects. We demonstrate and discuss how these techniques could be applied to two scenarios involving, (1) a multi-touch surface and (2) a public display and a mobile device.

Author Keywords
Gaze Interaction, Manual Interaction, Pervasive Display, Interactive Surface, Mobile.

ACM Classification Keywords
H.5.2 [Information interfaces and presentation]: User Interfaces, Input Devices and Strategies, Interaction Techniques.

General Terms
Human Factors

INTRODUCTION
As computing moves away from the desktop and in to our daily environment, public displays and interactive surfaces give rise to content that can become inaccessible to users. This can be due to the lack of an appropriate input modality, or because the display is too large, causing content to be located out of physical reach. Several researchers have addressed this issue by devising techniques to remotely access content, move content or bring it closer for manipulation. Their work has incorporated modalities such as touch [4, 5, 1], free-hand gestures [9] and mobile devices [1].

We have identified two tasks that users may undertake with objects that are out of their physical reach; *Retrieve & Return* and *Relocate*.

Retrieve & Return is a two-stage sequence of actions to bring

a distant object closer to manipulate it, then return it to its original location. In the first stage the user selects an object, chooses a location near to them and performs an action to move the object to that position. Once the user has the object close, they can manipulate it however they wish. In the second stage the user selects the object again and performs an action to return it to its original location.

Relocate is the task of moving an object from one location to another. This task includes the same actions as the first stage of *Retrieve & Return* but instead of selecting a location nearby, the final location of the object could be anywhere.

To aid performing these tasks in pervasive situations we propose three techniques that utilise the combination of gaze with manual interaction. In this paper we outline these techniques and demonstrate how they could be applied to two different scenarios. The advent of wearable eye trackers promises continuous eye movement monitoring and mobile eye-based interaction to become a reality in the near future [3]. Our eyes are a potentially fast input modality and provide a reliable method of determining visual attention [6, 10].

RELATED WORK
Eye-based Interaction
Jacob showed that the eye can provide a pointing accuracy of around one degree [6]. He also described the "Midas Touch" problem that refers to the fact that users need a method of confirming their actions upon an object in view. To address this issue Jacob suggested the paradigm of dwell-time, i.e. applying a fixed length of time (150ms - 250ms) that the user's gaze must remain over an object before it is activated [6]. Dwell-time has an inherent latency, to overcome this Jacob suggested allowing users to bypass the time-out and confirm their actions instantly using a keyboard.

Zhai *et al.* combined manual and gaze input for pointing and selection [10]. Their system showed how gaze can be used for pointing to assist manual control to speed up object selection and reduce the need for manual cursor movement. The following systems focus on eyes only interaction with displays. Mardanbegi *et al.* demonstrated a head-worn eye-tracking system to map gaze to displays in the environment for interaction [7]. Blinking was used to perform commands and they described the issues around parallax error. Agustin *et al.* focused on informative displays upon which users could gaze to activate them for reading [8]. Breuninger

used a head-worn mobile eye-tracking device to help people with Quadriplegia to control devices around the home [2].

Extending Reach

There is a great deal of work that focuses on extending the physical reach of the user using touch interaction. For example, Boring *et al.* demonstrated interaction with remote screens through a live video image [1]. Their system allowed touches on the live video image of a mobile phone camera to be mapped to a public display that is in view. Von Hardenberg *et al.* showed that bare-hand computer interaction has advantages when used at a distance [9]. Objects could be moved and selected easily, though a hand-based dwell-time to simulate clicks added latency to their techniques.

Baudisch *et al.* investigated different techniques for dragging and dropping objects [4]. Drag-and-Pop and Drag-and-Pick used proxies to shorten large distances and to provide access to distant icons. Drop-and-Drag by Doeweling *et al.* focused on a more precise Drag & Drop technique that allowed the user to suspend the interaction mid-drag to perform fine-grained navigation [5]. All three techniques were found to be faster than traditional Drag & Drop for sufficiently distant targets.

COMBINING GAZE WITH MANUAL INTERACTION

It is clear from the related work that manual input allows for a multitude of unique commands to be defined. It is also clear that the eyes can be used to point over a distance. In particular in this paper we focus on interaction with surfaces and displays that are touch enabled, we therefore combine manual touch commands with gaze to obtain and relocate objects that are out of physical reach.

When moving objects there are two stages: (1) object selection and (2) target selection. Each of these stages requires two parameters: *Location* and *Action*. For object selection, *Location* defines where the object is and *Action* is the input to actually select it. For target selection, *Location* defines where the object will finally be placed and *Action* is used to confirm the location. In our case, gaze and multi-touch are used to define these parameters. Table 1 shows these two stages and dwell-time along with a break down of how the parameters for each are obtained.

Here we propose three new techniques; *Eye Cut & Paste*, *Eye Drag & Drop* and *Eye Summon & Cast*. Each technique is designed to fulfil the Retrieve & Return and Relocate tasks we've defined. We use a different touch command for each technique, the commands are described below, refer to Table 1 for an explanation of which technique uses which command.

- *Tap* consists of two touch events: a touch down and a touch up. These are performed in quick succession.

- *Hold/Release* uses two touch events which are more sustained than the tap command. Hold performs a touch down event. Release performs a touch up event.

- *Swipe* has three touch events in the order of touch down,

	Object Sel.		Target Sel.	
	Loc.	Action	Loc.	Action
Dwell-time	G	Dwell	G	Dwell
Eye Cut & Paste	G	Tap	G	Tap
Eye Drag & Drop	G	Hold	G	Release
Eye Summon	G	Swipe	Swipe	Swipe
Eye Cast	Swipe	Swipe	G	Swipe

Table 1. Input used for the stages of each technique. Summon and Cast are actually classed as one technique but are separated here to show the different stages, the swipe for each is only performed once. Key: G=Gaze, Tap/Hold/Release/Swipe=Touch Gesture, Dwell=Dwell-time

touch moved, touch up. The user places their fingers on the display and moves them for a short distance before lifting them off.

Eye Cut & Paste

This technique is like traditional Cut & Paste as it enables the user to pick up an object, remove it from the display and keep it before dropping it at a new location. The user is free to perform other tasks before dropping the object. As shown in Table 1 the object and target locations are set by the gaze of the user. Both actions are defined by the same touch command, each in turn is performed to move the object. To relocate an object this technique only needs to be performed once. To fulfil Retrieve & Return this technique must be performed twice, once to bring the object close and then again to move it back.

Eye Drag & Drop

Based on Drag & Drop, this technique allows the user to pick up the object and hold it in their grasp until they drop it at a new location, unlike Eye Cut & Paste the object is always held by the user and not stored temporarily. Referring to Table 1 it is shown that the object and target locations are set by the users gaze. To move the object, two actions must be performed to confirm each location, these are obtained through the touch modality. As with Eye Cut & Paste, the user may perform this technique again to either return the object or move it to another location.

Eye Summon & Cast

This technique allows for an object to be summoned (brought closer) using one touch command. Object location is defined by gaze whereas target location is defined by the location of a touch command. The same command fires both actions and causes the object to move. When the user wishes to return the object or move it to another location, they must cast it away. The target location is first defined by gaze, after which a touch command defines the object location and fires both actions to move the object.

This technique's two operations more explicitly fulfil the Retrieve & Return task than the other techniques. To relocate an object it must first be summoned then cast to its new location. In Table 1 Summon & Cast is shown on two separate rows, this is to clearly show how the roles of each modality change for the two functions of the technique. Swipe is

highlighted to clearly express that the command is only performed once to fire both actions and define one of the two locations needed.

APPLICATION SCENARIOS

In the following we will show how the proposed techniques can be applied to two different pervasive interaction scenarios. In both scenarios, users want to perform the tasks *Retrieve & Return* and *Relocate*.

Surface Interaction

This scenario demonstrates a situation where the user wishes to manipulate content that is out of physical reach on an interactive surface. We use a separate touch command for each technique (cf. Table 1) and two fingers to perform each command.
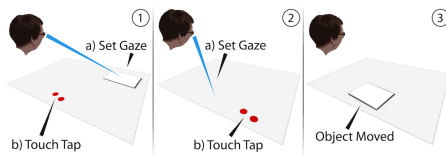
Eye Cut & Paste

Figure 1. Eye Cut & Paste on an interactive surface. (1) a) Look at object. b) Select with tap. (2) a) Look at target b) Confirm with tap. (3) Object moves to the target location.

Figure 1 shows three steps: (1) Object Selection: The user defines an object using gaze then performs a tap command to select it. (2) Target Selection: To move or retrieve the object the user directs their gaze to an area of the surface that is unoccupied and performs another tap command. (3) The object is moved to the gaze location defined in step 2. To return the object, the user would perform this technique again, but in step 1 they would select the object to return and in step 2 they would select an approximate location of where the object originated.
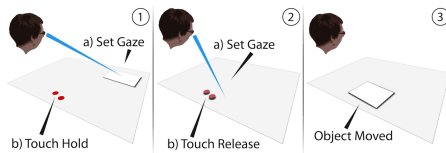
Eye Drag & Drop

Figure 2. Eye Drag & Drop on an interactive surface. (1) a) Look at object. b) Select with hold. (2) a) Look at target b) Confirm with release. (3) Object moves to the target location.

Figure 2 follows the same steps as Eye Cut & Paste but differs as (1) shows the hold command used to select the object and (2) shows the release command that drops the object. Similarly, these step can be performed again to return or relocate the object.

Eye Summon & Cast

Figure 3 shows: (1) (Summon) Object Selection: The user defines the object to move using their gaze and performs a swipe command towards themselves, Target Selection: The
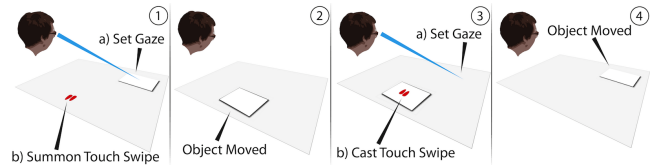

Figure 3. Eye Summon & Cast on an interactive surface. (1)(Summon) a) Look at object. b) Swipe sets target location and confirms move. (2) Object moved to swipe location (3)(Cast) a) Look at target. b) Swipe selects object and confirms move. (4) Object moved to gaze location.

same swipe command also defines the target and finalises the technique. (2) The object is moved to the location where the swipe command occurred. The next two stages show the cast operation: (3) (Cast) Target Selection: The target location is first set using the eyes. Object Selection: The user then does a swipe away from themselves on top of the object they wish to move. (4) The object touched moves to the location set by the eyes. To relocate an object using Eye Summon & Cast the object to be moved is summoned then cast away to the new location.

Tablet & Situated Public Display

Users may encounter a large public display that is not touch enabled or that is out of physical reach. In this case users could interact with this display in combination with a nearby surface or device, e.g. a tablet device. This could allow for several methods of manual input including, touch, hardware based buttons or movement detection using an accelerometer. The touch commands we've defined could easily be adapted or replaced by one of these modalities. For this example we use touch on a tablet surface (see Figures 4, 5 & 6).
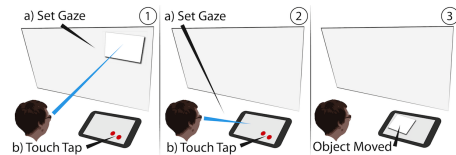
Eye Cut & Paste

Figure 4. Eye Cut & Paste on a tablet/display. (1) a) Look at object. b) Select with tap. (2) a) Look at target b) Confirm with tap. (3) Object moves to the target location.

Note in Figure 4 that in (2) the target location is defined on the mobile phone screen. When the object is brought closer in (3) it is delivered to the location defined on the tablet.
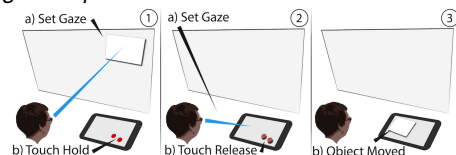
Eye Drag & Drop

Figure 5. Eye Drag & Drop on a tablet/display. (1) a) Look at object. b) Select with hold. (2) a) Look at target b) Confirm with release. (3) Object moves to the target location.

See Figure 5. The same points apply here as with Eye Cut & Paste above, but the Hold/Release command is used instead.
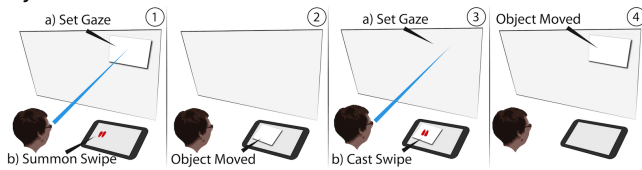
Eye Summon & Cast

Figure 6. Eye Summon & Cast on a tablet/display. (1) Summon a) Look at object. b) Swipe sets target location and confirms move. (2) Object moved to swipe location (3) Cast a) Look at target. b) Swipe selects object and confirms move. (4) Object moved to gaze location.

In this case, when moving an object, it must always come to the tablet display before being placed elsewhere, Figure 6.

DISCUSSION

The techniques we describe are at this point conceptual, but they could feasibly be implemented along with the systems that we have described within a short amount of time using current technology. *Dikablis* by Ergoneers and *iView X HED* by SensoMotoric Instruments (SMI) are portable wearable head-worn eye tracking systems capable of real-time output. Others like the *Tobii Glasses* by Tobii and *SMI Eye Tracking Glasses* by SMI demonstrate how eye-tracking technology is becoming more lightweight and potentially wearable in everyday life for pervasive interaction. To implement a working system, detection of screens in the environment would be required. Marker tracking technology such as *ARToolkit* could be used to register displays in the scene video, another method was described by Mardanbegi *et al.* involving detection of screen contours [7]. Both methods have the potential to suffer from error due to parallax [7].

It is also important to consider possible accuracy and usability issues concerned with using our proposed techniques. We presuppose that the eyes are a method of input that users will not have major issues utilising. It should be considered however, that many users are accustomed to established modalities. Touch is the more familiar form of input in this instance. Our techniques, *Eye Cut & Paste* and *Eye Drag & Drop* both receive their location information from the eyes alone. *Eye Summon & Cast* receives some of this information from touch. While gaze direction can be estimated with an accuracy of about one degree of visual angle accuracy may be lower for real-world applications. In addition to this, the hardware used to track the eyes could introduce further error.

Dwell-time is the most common method for eye-based interaction. Due to its inherent latency, it could be assumed that it is slow in comparison to our new techniques that allow the user to instantly confirm their actions through touch. *Eye Cut & Paste* and *Eye Drag & Drop* use gaze to get location information but receive their actions from touch. This is similar to using a keyboard to speed up selection as described in [6]. *Eye Summon & Cast* could be considered even quicker given that both locations are known and both

actions are fired with the execution of one touch command instead of two in sequence. The techniques will be compared against their speed and accuracy in future studies.

CONCLUSION

We have shown how gaze and manual interaction can be combined to develop techniques for interaction at a distance. We have outlined three techniques that would enable users to interact with content that is out of physical reach on public displays and interactive surfaces. We defined two tasks Retrieve & Return and Relocate and demonstrated how each of these could be performed in two different scenarios. In our discussion we've contemplated the potential issues of eye-based interaction, the possible advantages of our techniques over the existing dwell-time paradigm and the technologies that could be used for implementation.

REFERENCES

1. S. Boring, D. Baur, A. Butz, S. Gustafson, and P. Baudisch. Touch projector: mobile interaction through video. *Proc.* CHI '10, pages 2287–2296, 2010. ACM.

2. J. Breuninger, C. Lange and K. Bengler. Implementing gaze control for peripheral devices. *Proc. of the 1st International Workshop on Pervasive Eye Tracking and Mobile Eye-Based Interaction*, PETMEI. ACM, 2011.

3. A. Bulling and H. Gellersen. Toward Mobile Eye-Based Human-Computer Interaction. *IEEE Pervasive Computing*, 9(4):8–12, 2010.

4. R. S. Content, P. Baudisch, E. Cutrell, D. Robbins, M. Czerwinski, P. T, B. Bederson, and A. Zierlinger. Drag-and-pop and drag-and-pick: techniques for accessing. pages 57–64, 2003.

5. S. Doeweling and U. Glaubitt. Drop-and-drag: easier drag & drop on large touchscreen displays. *Proc.* NordiCHI '10, pages 158–167, 2010. ACM.

6. R. Jacob. The use of eye movements in human-computer interaction techniques: what you look at is what you get. *ACM Transactions on Information Systems (TOIS)*, 1991.

7. D. Mardanbegi and D. W. Hansen. Mobile gaze-based screen interaction in 3d environments. *Proc.* NGCA '11, pages 2:1–2:4, 2011. ACM.

8. J. San Agustin, J. P. Hansen, and M. Tall. Gaze-based interaction with public displays using off-the-shelf components. In *Proc.* Ubicomp '10, pages 377–378, 2010. ACM.

9. C. von Hardenberg and F. Bérard. Bare-hand human-computer interaction. In *Proceedings of the 2001 workshop on Perceptive user interfaces*, PUI '01, pages 1–8, New York, NY, USA, 2001. ACM.

10. S. Zhai, C. Morimoto, and S. Ihde. Manual and gaze input cascaded (magic) pointing. In *Proc.*, CHI '99, pages 246–253, 1999. ACM.