HOBS: Head Orientation Based Selection in Physical Spaces

Anonymous for submission

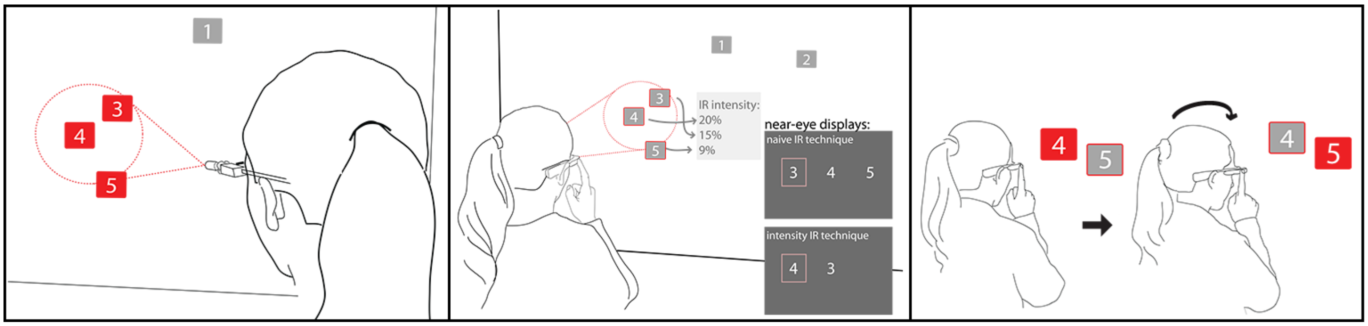

Figure 1. ***Left*: Our head orientation-based selection techniques use an IR emitter – multiple targets may fall within its illumination area. *Center*: We offer two list-based refinement techniques – *Naive IR* uses alphabetical ordering; *Intensity IR* orders targets by IR intensity. *Right*: Alternatively, users can refine their selection through head orientation refinement in a quasi-mode.**

ABSTRACT

Emerging head-worn computing devices can enable interactions with smart objects in physical spaces. We present the iterative design and evaluation of HOBS – a head-orientation based selection method for interacting with these devices at a distance. We augment a commercial wearable device, Google Glass, with an infrared (IR) emitter to select targets equipped with IR receivers. Our first design shows that a naive IR implementation can outperform list selection, but has poor performance when refinement between multiple targets is needed. A second design uses IR intensity measurement at targets to improve refinement. To address the lack of natural mapping of on-screen target lists to spatial target location, our third design infers a spatial data structure of the targets enabling a natural head-motion based disambiguation. Finally, we demonstrate a universal remote control application using HOBS and report qualitative user impressions.

Author Keywords

Wearable computing; selection; glass; infrared

ACM Classification Keywords

H.5.2. Information Interfaces and Presentation (e.g. HCI): Interaction styles (e.g., commands, menus, forms, direct manipulation).

INTRODUCTION

The number of smart objects in our environment with embedded computation and communication has grown rapidly. These objects are all potential targets for interaction. To initiate such an interaction, a user needs to first acquire the target object – a fundamental task in an interaction process that has been extensively studied in the digital world, but is not yet well-explored in physical spaces.

Past research has introduced techniques of augmenting hand-held mobile devices with accessories like laser pointers to enable direct aiming at target devices in space [1, 18]. While promising, some drawbacks of using hand-held devices are that the device first has to be retrieved (e.g., from a pocket) and cautiously aimed; that the user's hands are required to be free for operation; and that the user's visual attention is split between looking down at a screen and out at targets in the world.

Emerging head-worn computing devices have the potential to overcome some of these limitations: they do not require retrieval since the devices are already worn; they may enable hands-free or uni-manual interactions; and they offer near-eye or see-through displays to present information in the wearer's field of view. We thus investigate how such computing devices may be used for the selection of physical targets.

Head-worn devices can naturally exploit the user's head orientation, an important but less precise indicator of the user's *locus of attention* [15]. It suggests the general direction, but not the particular point of focus. While gaze tracking could provide further information about a user's focus, wearable gaze tracking cannot by itself reliably determine the identity of objects in focus (i.e. it also needs to know what the user is seeing). Equipping all devices with stationary gaze trackers is

Submitted for review to UIST 2014.

possible [21] but much costlier than our approach. We therefore study how to leverage head-orientation alone for target selection in physical spaces.

The imprecision of human head movement suggests adapting area cursor techniques developed for assistive devices [6, 24, 5]. Such techniques employ a two-step selection process: a *coarse* selection of an area of interest, followed by a *refinement* to select a target within that area.

In this paper, we describe the iterative development and evaluation of HOBS, an area-selection technique that can be readily implemented with small hardware changes to emerging head-worn devices. We augment Google Glass[1] to enable infrared (IR) communication between Glass and target appliances. In our *Naive IR* technique, the cone of light emitted by an IR LED provides the *coarse* selection area (illustrated in Figure 1 *left*), where targets that have received an IR signal light up. To *refine* selection when multiple targets have received IR signals, the *Naive IR* solution starts with an ordering based on their names. This is shown in the center of Figure 1, where targets in range are ordered numerically on the near-eye display. A study with 14 participants compares acquisition times for physical targets in a room for our technique and an alternative list selection interface without any IR targeting . We find that target acquisition through head orientation is preferred by users and is faster than list selection, but refinement is still time consuming.

In response, we designed an improved refinement technique, *Intensity IR*, in which target objects compare received IR signal strength. This value allows the system to eliminate some distracting targets on the peripheral and to re-order the refinement interface's list by their intensity values. For example, in Figure 1B of *Intensity IR* technique, device 5 is eliminated first and the list is re-ordered based on the intensity readings. A second study with 10 participants proves that *Intensity IR* successfully reduces both the chance of needing to do refinement as well as the time spent in list navigation when compared to *Naive IR*.

A final design addresses the lack of a natural mapping when users select a target in the refinement step using their device's touchpad - the axes of motion do not map directly to the spatial layout of target devices in a room. Our *Head-motion Refinement* technique first learns the relative spatial structure of the targets using Glass' orientation sensors. In a future target acquisition stage, users can perform head movements to change selections to spatially adjacent targets (see the right of Figure 1). For example, nodding down to select the target below current selection, or tilting right to select the next target on the right. We present preliminary feedback from participants on this technique.

We also demonstrate an example application of our technique used as a remote control of smart appliances: a user looks at the appliance he wishes to control and confirms selection by tapping. An appliance-specific user interface is then shown on the user's near-eye display for further interactions.

[1] `http://www.google.com/glass/start/`

BACKGROUND AND RELATED WORK

Our approach is related to head- and eye-controlled interfaces, area cursors and prior work on hardware devices for pointing in physical spaces.

Head and Gaze Input

Head movement has long been used for virtual camera control in VR applications [12] and as an assistive input technology for cursor control of desktop applications [13]. However, human neck muscles have a much lower bandwidth than other muscle groups, e.g., the wrist [3].

Prior work often focused on head orientation for controlling graphical interfaces; in contrast, we apply this modality to selection in physical spaces.

Gaze can also be used to control graphical user interfaces [8]. While there are wearable gaze trackers [2], turning information about a concrete point in space where a user is looking into a selection requires a map of each space with known target locations. Our system works through point-to-point IR communication and does not require an *a priori* map. Target objects in the environment can also be equipped with individual cameras that watch the user [19, 21]. Such an approach can enable similar benefits as our approach, but is computationally more expensive than our single-pixel sensor solution and may not work at greater distances or angles, because it relies on finding the user's pupils in a camera image.

Area Cursors

In GUI area cursors, the activation area of the cursor is enlarged, which facilitates acquiring smaller targets [6]. Area cursors are especially appropriate for individuals with motor control impairments or difficulties [24, 5]. We argue that head orientation pointing has similar challenges (limited pointing performance and accuracy). In all area cursors, the bigger cursor activation area can lead to multiple targets being selected and disambiguation is needed.

We conceptualize area pointing as a two-stage process: in the *coarse* phase, which we call *scanning*, users move so the activation area intersects with the target object (and possibly other, unintended targets). In the *refinement* phase, they adjust so only the intended target will be selected. Many disambiguation techniques are possible for refinement – this paper describes the trade-offs between several of them.

Pointing in Physical Spaces

Rukzio studied alternative methods for selecting devices in physical spaces and found that users strongly preferred either tapping target appliances with a mobile device or pointing at a distance to browsing a list [16].

Several approaches to spatial selection with handheld devices exist for controlling appliances [1, 18, 23, 17, 7] and exchanging information with smart infrastructure sensor networks [9, 11, 4]. In some techniques, users select objects of interest with laser pointers. The laser dot provides immediate visual feedback to the user about what is being selected; however, its small target area makes it poorly matched to head orientation input.

Other approaches rely on virtual room models in which a user's location is estimated using IMU-based orientation sensing [23, 9] – in contrast, our technique does not require a static map ahead of time.

Handheld projectors can both display a user interface in space and communicate control information optically, e.g., by encoding information temporally (using Gray codes in Picontrol [17] and RFIG [14]) or spatially (using QR codes in the infrared spectrum in SideBySide [22]). Our solution is similar in spirit but relies on simple low-cost IR emitters and detectors.

Other targeting systems use IR with handheld pointers [20] as well as wearable devices such as rings and Bluetooth audio earpieces [10] to connect to smart devices. Our system tackles an unresolved issue of such IR-based approaches – navigating an area dense with potential targets.

IR FOR HEAD ORIENTATION SENSING

Hardware

A central hypothesis of this paper is that the area cursor paradigm [6] is well matched to head orientation selection. Head orientation input is imprecise because it does not capture eye movement and has to rely on a low-bandwidth muscle group. Therefore, point selection techniques like laser pointers are not appropriate.

Infrared emitters are a good technology match since they emit light within a given angle, resulting in a cone in front of the emitter where the light is visible. LEDs with many different fields of view are commercially available. Each individual target that is equipped with IR detector can react to the signal. Such point to point communication enables us to achieve targeting without needing a map.

Another benefit is the low cost and easy integration of both emitters and detectors - detectors often have support for common IR protocol decoding built in and are inexpensive, making it economically feasible to add detectors to dozens or hundreds of devices in an environment.

Our augmentation to Google Glass adds a 940nm 5mm IR emitter (SFH 4545 from OSRAM Opto Semiconductors Inc.). The emitter is controlled by an additional microcontroller which communicates with Google Glass through Bluetooth radio (see Figure 2). In the following description, we use the term "Glass" to refer to this augmented hardware.

While we use IR signal for targeting, wireless communications are needed to efficiently transmit the reception status of IR signal from targets to Glass. We have chosen a commercial off-the-shelf ZigBee implementation (XBee based on 802.15.4 radio) for this purpose.

Interaction Model

From the user's perspective, interaction with HOBS proceeds in two stages (Figure 3):

Scan: The user first scans the environment to locate the position of the target. During this stage, Glass constantly sends out IR signals, and targets offer immediate feedback when they receive a signal. The user confirms his desire to connect to a target by tapping on the Glass touchpad. Glass collects the responses from targets that have received IR reception through the backchannel of XBee. If there is only one single target in IR range, it is automatically selected. However, in a dense environment where multiple targets are within range, the user needs refine his selection.

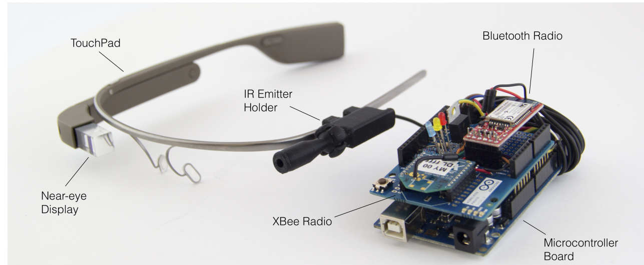

Figure 2. Our Glass hardware: Google Glass augmented with a repositionable IR holder, and an additional microcontroller that communicates with Google Glass and controls IR emitter.

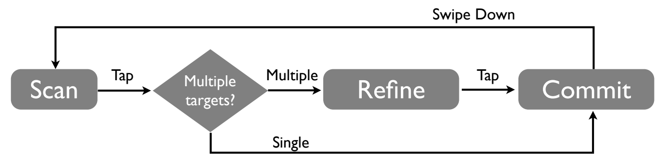

Figure 3. *scan* and *refine* – two main stages during the interaction with HOBS. For completeness, we have also added the final state of *commit*.

Refine: When disambiguation is needed, the user must make an explicit selection among the targets within his view range. We have designed multiple refinement mechanisms – all of which enable the user to select one from a subset of targets. The user confirms a current selection with a tap. Since the purpose of this stage is to disambiguate among potential targets, we will also use *disambiguation* to refer to this stage. Finally, a tap confirms a decision.

The overall target acquisition time thus depends on scan and refine times, the probability that refinement is needed, and the time to commit an action (tap):

$$t_{total} = t_{scan} + P(refine) * t_{refine} + t_{commit} \quad (1)$$

In the following sections, we describe our iterative design and evaluation process to minimize the overall target selection time.

ITERATION 1: NAIVE IR

Our initial research question is: *Can IR-based targeting reduce the selection time compared to the case where only UI list navigation is used on a head-worn device?*

Technique

In our first implementation, we use IR for scanning as described in the previous section. For the refinement stage, we simply show a list of the subset of targets that have received IR signals on the Glass display. Users swipe to select from that list and tap to confirm the intended target.

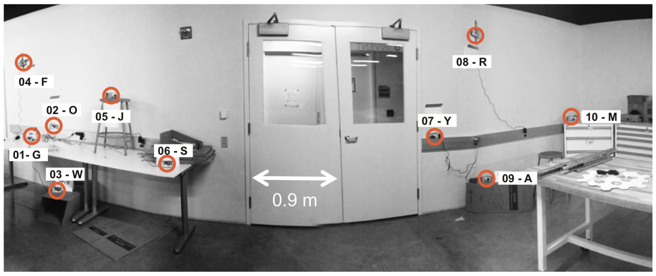

Figure 4. In the targeting study, participants were asked to find and select one of 10 targets in the lab environment.

A natural point of comparison is an interface that does not use any head orientation information - it always shows a complete list of all targets. We implemented a list view where users swiped forward and backward on the Glass touchpad to navigate incrementally through the list. To quantify the benefit of using IR for the scanning step, we carried out a target acquisition study which compares the *Naive IR selection* and *list selection*.

Method

We deployed 10 wireless nodes in an indoor environment (Figure 4). Each of them has a number as ID and a letter representing the name. We recruited 14 participants from our institution for this study. 13 had never used Google Glass before, so we offered a tutorial before the experiment in order to introduce the device. Four participants wore prescription glasses, which makes Glass more cumbersome to use and adjust, and may have affected their task performance. In this study, the IR LED was fixed and not repositionable as shown in Figure 2.

In the within-subject study, half of the participants performed *IR selection* first and the other half used *list selection* first. For each selection condition, we conducted 15 target selections by randomly choosing from all the targets. During the study, we measured the **target acquisition time** for each target selection. Afterward, participants were asked to complete a survey of primarily open-ended answers about their experience.

Results

Our results indicate that *IR selection* outperforms *list selection*. The average target acquisition time for *IR selection* is 6.67 seconds while *list selection* took 8.86 *list* seconds (see Figure 5). A t-test shows a significant difference ($t(279) = -3.81, p = 0.00017$).

To understand how scanning and refinement contribute to total selection time, we split the data from *IR selection* into two parts – trials that required refinement and ones that did not. It takes 6.40 seconds (on average) to complete a selection without any refinement, but 9.16 seconds with refinement, indicating that an additional 2.76 seconds are needed for disambiguation (see Figure 6). This difference is significant ($t(19) = -2.7827, p = 0.012$) using t-test. Because many targets were spaced far apart, refinements were only necessary in 10% of total *IR selection* trials in this study.

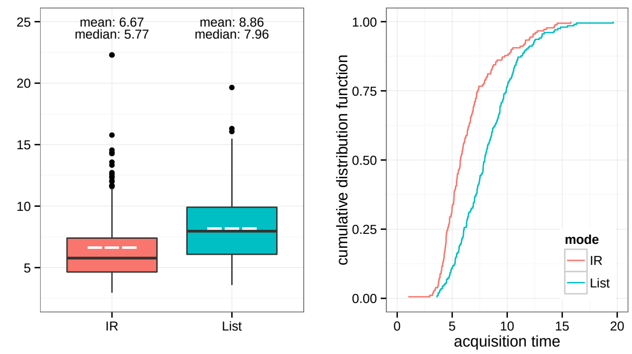

Figure 5. Boxplot of target acquisition time for *IR selection* and *list selection* is shown on the left. The center is the median value, and the mean value is shown using white dashed lines. The cumulative distributions are on the right.

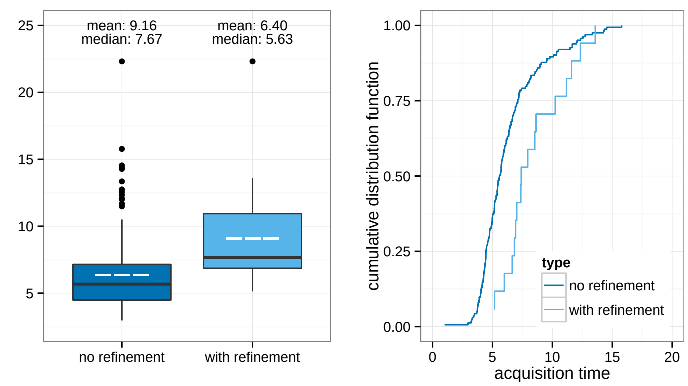

Figure 6. Boxplot and CDF of target acquisition time when refinement is needed or not in *IR selection*

To further generalize the results, in Figure 7, we show the acquisition time for each individual target in *list selection* condition, ordered by their relative position in the list. Since with IR technique, the acquisition time is invariant from each target's order, we use solid lines to represent the average performance of *IR selection*. From this figure, we can see that once there are more than 6 targets, the average acquisition time will be larger than IR selection, even if disambiguation is needed.

As for the qualitative feedback, 11 out of 14 participants preferred IR selection over list mode (two preferred list, one was undecided). While both interfaces were judged similarly on overall ease of connecting, list selection was perceived to be cumbersome. One user noted benefit of *IR selection* being *"more direct [than list mode]"*, allowing users to focus on the targeted objects instead of the screen. One subject called it *"natural to interact with things just by looking at them"*. Another mentioned that *"it's really convenient that what I'm looking at is what I'm targeting"*, and many participants compared the overall time of target acquisition. One subject pointed out that "IR selection *doesn't requires you to swipe a lot of times like list mode"*.

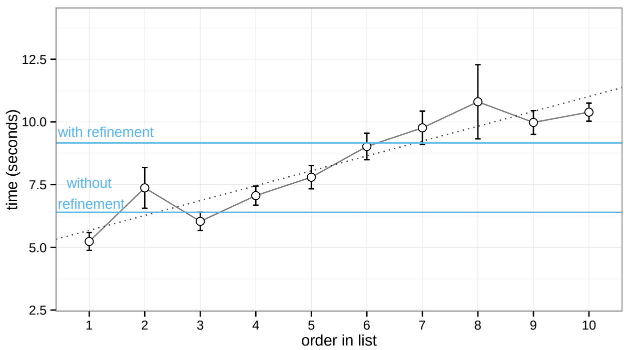

Figure 7. Times taken to select a device vs. its order in the *list selection*. The dotted line is a linear fit between the average time and target orders in the list. Two horizontal solid lines are the average target acquisition times in *Naive IR* when refinement is needed or not.

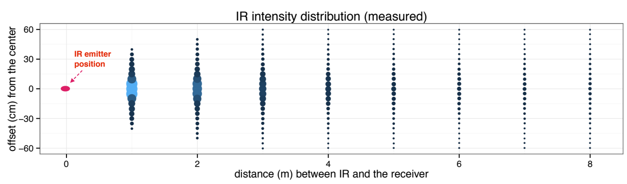

Figure 8. Empirical measurement of IR intensity at different position. We measure the intensity at a plane every meter away from the IR emitter. On that plane, we obtain a sample every 5 centimeters. The size and color brightness in this graph represents the intensity readings.

In summary, *Naive IR* can outperform linear list selection, and most users prefer this head-orientation based targeting. From our quantitative result, however, we found that the refinement step detracts significantly from the efficiency of the technique.

ITERATION 2: INTENSITY IR

Performance of the *Naive IR* technique will degrade as target density in an environment increases, as increased density will require refinement steps. We therefore ask a follow-up research question: *How might we improve selection time in a dense environment?*

Technique

Previously we only used IR reception as a binary signal for identifying potential targets. We hypothesize that IR intensity at the receiver side can provide more information about the likelihood that a user intended to select a particular target. Received IR intensity falls off with distance between IR emitter and receiver as well as with the angle between the emitter and receiver. To measure intensity, we add an IR light-to-voltage converter TSL267-LF by AMS-TAOS USA Inc.

We have empirically measured the intensity distribution at the receiver for this configuration in Figure 8. Our measurements confirm that angular difference has a large effect on the intensity readings (see how the IR intensity distribution changes when the offset from the center increases).

The intensity information is used in two ways:

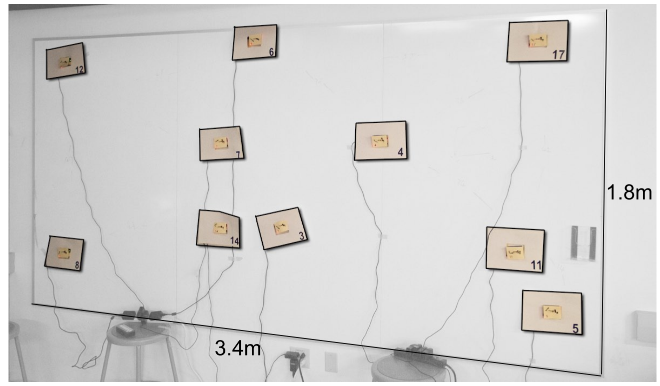

Figure 9. The environment setup for our second study. In comparison to our first study, we have deliberately increased the target density.

1. When multiple targets have received IR signals and reported the intensity readings, we discard those whose intensities are significantly lower than the largest reading[2]. Therefore, when there is only one target within line of sight, the IR intensity approach has the same behavior as the previous iteration - no disambiguation is needed. When the environment turns denser, the new design can filter some peripheral targets out, reducing $P(refine)$, the likelihood of entering the refinement stage.
2. When refinement is still needed, meaning that multiple targets have relatively close intensity values, the system sorts the disambiguation list according to the IR intensity, from strongest to weakest. We hypothesize that this will reduce t_{refine} significantly by minimizing extra navigation steps, as the first list item will generally match the intended target.

Method

To quantify the improvement in this design we performed a second study to compare the *Naive IR* and *Intensity IR* approaches. Because we were interested in discovering performance differences in denser environment, we reposition the 10 nodes and set them up in a smaller area (see Figure 9). We recruited 10 participants for this study. Each user performed 30 target acquisition tasks for each approach. As in our first within-subject study, half of them perform *Naive IR* first and the other half *Intensity IR* first.

Results

Intensity IR reduces the number of trials in which refinement dialogs are needed from 225 of 300 in *Naive IR* to 167 of 300 trials. A Chi-square test shows this difference is significant ($\chi^2(1) = 24.755$, $p < 0.0001$). This demonstrates that the new approach successfully reduces the probability that a disambiguation dialog is needed.

In the cases where disambiguation is inevitable, *Intensity IR* sorts the list based on the intensity reading, while *Naive IR* sorts alphabetically. *Intensity IR* reduces the fraction of refinement trails in which additional list navigation is necessary

[2]In our current implementation, we empirically set it to be half of the ADC resolution, which is frequently used as it indicates a 3dB loss in the signal strength.

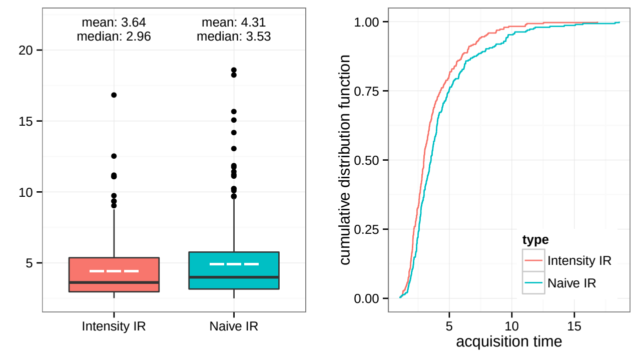

Figure 10. Boxplot and CDF of the target acquisition time in *Naive IR* and *Intensity IR* conditions.

(i.e., the first, already selected element is incorrect). *Intensity IR* sorted the desired target as the first one in the list in 55% of cases (93 of 167). In comparison, for *Naive IR*, only 35% of trials sorted the desired target as the first one in the list (80 out of 225). A Chi-square test show that this difference is significant (with $\chi^2(1) = 15.758$, $p < 0.0001$).

From Figure 10, we can see that the overall target acquisition time has decreased from 4.31 seconds for *Naive IR* to 3.64 second for *Intensity IR*. This difference is also significant ($t(555) = 3.2945$, $p = 0.001$.)

One side effect that we have observed in this approach is that the *Intensity IR* sometimes eliminates the desired target during the *scanning* stage. Out of 300 trials, 13 of them (4.3%) has been accidentally eliminated. This is higher than the error rate for *Naive IR* (only 5 out of 300, 1.6%).

ITERATION 3: HEAD MOTION REFINEMENT

Both *Naive IR* and *Intensity IR* rely on list navigation on the near-eye display for the refinement step. The list interface has a few clear shortcomings: navigation actions map poorly to real world results, and the users must switch their focus back and forth between the physical scene and the near-eye display. These problems motivated us to design an approach that harnesses the head orientation during the *refinement* stage.

In our third iteration, we ask the question *can we build a system purely using head orientation and visual feedback from the environment for target selection?*

Technique

We introduce a third technique that uses a combination of motion sensors and IR to learn the relative orientations of targets in a room and intelligently suggest targets during refinement (see Figure 12).

In the learning stage, the user scans over the targets, and the system attains the absolute orientation of each device from IR and motion sensors. From this information it can abstract out their relative positions and build an *adjacency map*.

This absolute orientation cannot be applied to all indoor environments, since the users movements through the space could change the relationships between targets. However, with the

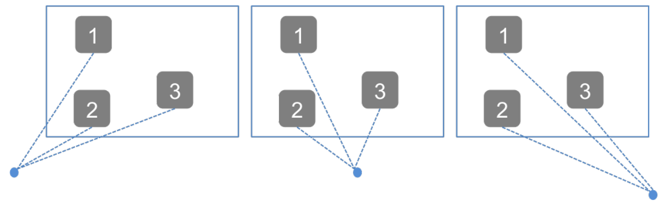

Figure 11. This illustrates that when the change of user's absolution position doesn't change the relative relationship of physical targets.

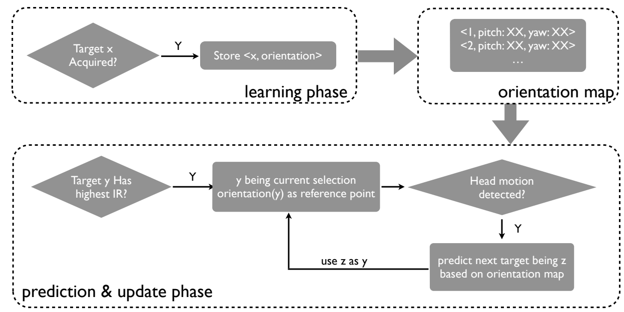

Figure 12. Our third technique learned each target's absolute orientation and construct the adjacency map. During the *refinement* stage, the prediction is based on relative changes to a reference point.

constraint that the targets are spread around the periphery, their relative orientations are stable (see Figure 11).

After the map is created, the user can hold down on the touchpad to enter a quasi-mode for refinement. In this quasi-mode, one device lights up at a time. When the user turns his head in the direction of another device, the light switches to that device. Therefore, the user can move between devices one at a time with slight head movements. We implement this interaction by calculating the user's direction of motion using a low-passed history of sensor measurements and searching through the adjacency map for the nearest device in that direction.

Evaluation

We evaluate the head motion refinement method by holding an informal study and collecting qualitative feedback from a subset of 4 users from the Iteration 2 study. In the study, we asked users to try cycling through the targets using the new quasi-mode.

The users had strong preferences for the new method of refinement. On a scale from 1-7, 1 being the least mental effort and 7 being the most mental effort users rated the old technique 4.25 and the new technique 2 on average. 100% indicated a preference for head movement to list navigation. One user referenced the issue of naming targets that the list necessitates, preferring the experience of *"matching visual cues rather than numbers"*. Another participant remarked that it *"just made more sense"* and was a *"more natural way for demonstrating intentionality"*. The users preferred the new mapping in relation to the whole environment: *"it leveraged the spatial sense that I already had just by using the system"*. They were also delighted to avoid list navigation, which they now called *"difficult"* and *"painful"*.

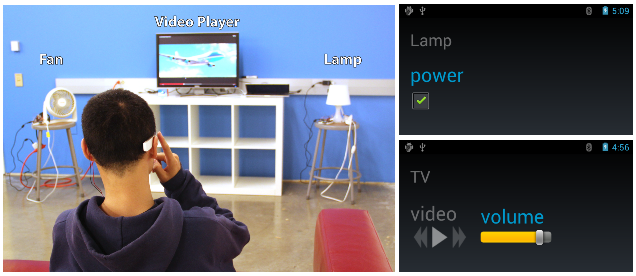

Figure 13. In the smart home scenario, we have built three smart appliances for a user experience study. The interface supports both simple on/off appliances (lamp or fan) and multi-functional appliances (TV, for example).

APPLICATIONS

Head orientation targeting can enable a wide range of context-aware applications. We implement one particular demonstrative application: a universal smart appliance controller. Users select a smart device (e.g., light fixture, TV, or home appliance) with Glass — upon confirming the selection, an appliance-specific UI is shown on the user's near-eye display, and they can control the application (without having to continually look at it) through their device touchpad.

We implemented a prototype of this remote controller for three physical devices: a lamp and fan that could be switched on or off, and a smart TV control with playback, volume and navigation controls (see Figure 13). We switched discrete appliances using relays, and implemented the smart TV with a laptop connected to a 30" display. User interfaces for each were pre-defined in our application.

We set the devices up in a simulated living room environment and invited 14 users to step through a predefined set of tasks to control the appliances at a distance.

All participants successfully completed the list of tasks. They commented positively on the universal remote control functionality (e.g., *"I didn't have to search for different remote controllers for different appliances"*) and stated it was easy to target and connect to appliances, in line with the findings of the previous study procedure. Participants saw potential benefits of the device for families; one user remarked that he could imagine people using the system *"while keeping an eye on their children at the same time"*.

DISCUSSION

The rapid development of sensing technologies has created many opportunities for new ways to interact with smart objects. In our exploration of the design space of head orientation-based target selection, we carefully selected sensing techniques that are readily available and easy to deploy; and we added complexity to our system only when necessary.

Interpretation of results

A primary goal of this paper was to provide an effective and efficient method for target selections in physical spaces. Targeting is a fundamental building block across many interaction tasks - it has a significant impact on user experiences collectively and can provide seamless interaction when designed well.

We formalized a scan and refine model of head orientation-based selection (Equation 1). We first introduced head orientation as an alternative to list selection and showed that *scanning* can outperform list selection. Our redesigns then focused on the case where refinement is needed. The two ways to reduce refinement time are 1) to reduce $P(refine)$, the probability that a manual refinement is necessary; and 2) to reduce t_{refine}, the time required to perform the refinement interaction itself. Using IR intensity readings addresses both these terms, as it can be used to both avoid showing refinement dialogs, and to optimize their display when they are needed.

Our final head-orientation technique improves the nature of the mapping between items in the refinement dialog and the layout of targets in space. Informal testing suggests that users prefer using this spatial mapping.

Limitations

We introduced new refinement operations based on IR intensity and spatial adjacency tracking that improve the performance of head orientation in environments with dense targets. These techniques have several limitations.

First, IR intensity measurements only work within the dynamic range of our sensor. Additional strong IR sources like direct sunlight may saturate the sensor and make discrimination impossible. Second, our adjacency map is built assuming stable relative target locations. If targets move, the map will have to be recalculated. This may be done incrementally during everyday interactions, but we have not yet tackled this challenge.

Finally, we acknowledge several limitations of our study design: we have not yet systematically studied target density variation, our study was performed in a lab environment, and only measured first use. Future work should study how the technique applies in realistic settings over longer periods of time.

CONCLUSION

In this paper, by presenting our iterative design process in head-orientation based targeting, we have learned that IR alone can help reduce the overall acquisition time by reducing the chances when we need to perform refinement. With IR intensity added, the targeting can work better in a relative dense environment. However, a more natural approach is to combine IR with head motion. Through our preliminary user studies, we learned that this is a more intuitive way of performing refinement in comparison to the menu-based selections. We leave a more comprehensive technical solution of using motion sensors and its evaluation as the future work.

REFERENCES

1. Beigl, M. Point & click-interaction in smart environments. In *Handheld and Ubiquitous Computing*, H.-W. Gellersen, Ed., no. 1707 in Lecture Notes in Computer Science. Springer Berlin Heidelberg, Jan. 1999, 311–313.

2. Bulling, A., Roggen, D., and Tröster, G. Wearable eog goggles: Seamless sensing and context-awareness in everyday environments. *Journal of Ambient Intelligence and Smart Environments 1*, 2 (2009), 157–171.
3. Card, S. K., Mackinlay, J. D., and Robertson, G. G. A morphological analysis of the design space of input devices. *ACM Trans. Inf. Syst. 9*, 2 (Apr. 1991), 99–122.
4. Costanza, E., Panchard, J., Zufferey, G., Nembrini, J., Freudiger, J., Huang, J., and Hubaux, J.-P. SensorTune: a mobile auditory interface for DIY wireless sensor networks. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '10, ACM (New York, NY, USA, 2010), 2317–2326.
5. Findlater, L., Jansen, A., Shinohara, K., Dixon, M., Kamb, P., Rakita, J., and Wobbrock, J. O. Enhanced area cursors: reducing fine pointing demands for people with motor impairments. In *Proceedings of the 23nd annual ACM symposium on User interface software and technology*, ACM (2010), 153–162.
6. Kabbash, P., and Buxton, W. A. The prince technique: Fitts' law and selection using area cursors. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, ACM Press/Addison-Wesley Publishing Co. (1995), 273–279.
7. Kemp, C. C., Anderson, C. D., Nguyen, H., Trevor, A. J., and Xu, Z. A point-and-click interface for the real world: laser designation of objects for mobile manipulation. In *Proceedings of the 3rd ACM/IEEE international conference on Human robot interaction*, HRI '08, ACM (New York, NY, USA, 2008), 241–248.
8. Kumar, M., Paepcke, A., and Winograd, T. Eyepoint: practical pointing and selection using gaze and keyboard. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, ACM (2007), 421–430.
9. Lifton, J., Mittal, M., Lapinski, M., and Paradiso, J. A. Tricorder: A mobile sensor network browser. In *Proceedings of the ACM CHI 2007 Conference-Mobile Spatial Interaction Workshop* (2007).
10. Merrill, D., and Maes, P. Augmenting looking, pointing and reaching gestures to enhance the searching and browsing of physical objects. In *Pervasive Computing*. Springer, 2007, 1–18.
11. Mittal, M., and Paradiso, J. Ubicorder: A mobile device for situated interactions with sensor networks. *Sensors Journal, IEEE 11*, 3 (2011), 818–828.
12. Pausch, R., Shackelford, M. A., and Proffitt, D. A user study comparing head-mounted and stationary displays. In *Virtual Reality, 1993. Proceedings., IEEE 1993 Symposium on Research Frontiers in* (1993), 41–45.
13. Radwin, R. G., Vanderheiden, G. C., and Lin, M.-L. A method for evaluating head-controlled computer input devices using fitts' law. *Human Factors: The Journal of the Human Factors and Ergonomics Society 32*, 4 (1990), 423–438.
14. Raskar, R., Beardsley, P., van Baar, J., Wang, Y., Dietz, P., Lee, J., Leigh, D., and Willwacher, T. RFIG lamps: interacting with a self-describing world via photosensing wireless tags and projectors. In *ACM SIGGRAPH 2004 Papers*, SIGGRAPH '04, ACM (New York, NY, USA, 2004), 406–415.
15. Raskin, J. *The Humane Interface: New Directions for Designing Interactive Systems*. ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, 2000.
16. Rukzio, E., Leichtenstern, K., Callaghan, V., Holleis, P., Schmidt, A., and Chin, J. An experimental comparison of physical mobile interaction techniques: Touching, pointing and scanning. In *UbiComp 2006: Ubiquitous Computing*. Springer, 2006, 87–104.
17. Schmidt, D., Molyneaux, D., and Cao, X. PICOntrol: using a handheld projector for direct control of physical devices through visible light. In *Proceedings of the 25th annual ACM symposium on User interface software and technology*, UIST '12, ACM (New York, NY, USA, 2012), 379–388.
18. Shwetak N. Patel, G. D. A. A 2-way laser-assisted selection scheme for handhelds in a physical environment. 200–207.
19. Smith, B. A., Yin, Q., Feiner, S. K., and Nayar, S. K. Gaze locking: passive eye contact detection for human-object interaction. In *Proceedings of the 26th annual ACM symposium on User interface software and technology*, ACM (2013), 271–280.
20. Swindells, C., Inkpen, K. M., Dill, J. C., and Tory, M. That one there! pointing to establish device identity. In *Proceedings of the 15th annual ACM symposium on User interface software and technology*, UIST '02, ACM (New York, NY, USA, 2002), 151–160.
21. Vertegaal, R., Mamuji, A., Sohn, C., and Cheng, D. Media eyepliances: using eye tracking for remote control focus selection of appliances. In *CHI'05 Extended Abstracts on Human Factors in Computing Systems*, ACM (2005), 1861–1864.
22. Willis, K. D., Poupyrev, I., Hudson, S. E., and Mahler, M. SideBySide: ad-hoc multi-user interaction with handheld projectors. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*, UIST '11, ACM (New York, NY, USA, 2011), 431–440.
23. Wilson, A., and Shafer, S. XWand: UI for intelligent spaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '03, ACM (New York, NY, USA, 2003), 545–552.
24. Worden, A., Walker, N., Bharat, K., and Hudson, S. Making computers easier for older adults to use: area cursors and sticky icons. In *Proceedings of the ACM SIGCHI Conference on Human factors in computing systems*, ACM (1997), 266–271.