

HOBS: Head Orientation-Based Selection in Physical Spaces

Ben Zhang[†], Yu-Hsiang Chen[†], Claire Tuna[†], Achal Dave[†],
Yang Li[‡], Edward Lee[†], Björn Hartmann[†]

[†]: UC Berkeley EECS & CITRIS Invention Lab [‡]: Google Research
{benzh,clairetuna,achal,eal,bjoern}@berkeley.edu, sean.yhc@gmail.com, yangli@acm.org

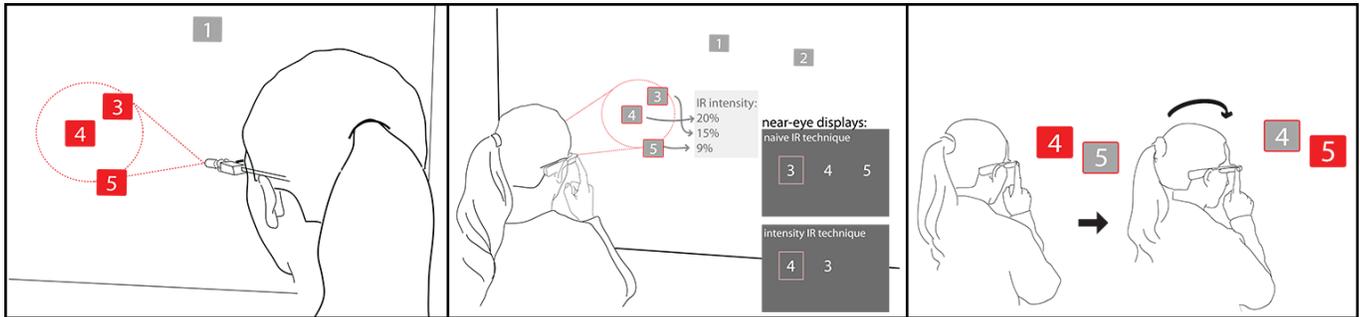


Figure 1. *Left*: Our head orientation-based selection techniques use an IR emitter – multiple targets may fall within its illumination area. *Center*: We offer two list-based refinement techniques – *Naive IR* uses alphabetical ordering; *Intensity IR* orders targets by IR intensity. *Right*: Using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

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 technique 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; Spatial Interaction; Selection; Glass

ACM Classification Keywords

H.5.2. Information Interfaces and Presentation (e.g. HCI): Interaction styles

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the Owner/Author. Copyright is held by the owner/author(s).
SUI '14, Oct 04-05 2014, Honolulu, HI, USA
ACM 978-1-4503-2820-3/14/10.
<http://dx.doi.org/10.1145/2659766.2659773>

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 *spatial interactions*, a user needs to first acquire the target object – a fundamental task that has been extensively studied in graphical user interfaces, but not yet well-explored in *physical spaces*.

Today, companies like Samsung and Whirlpool are making smart appliances with companion applications that use smartphones as *universal remote controls*. With these applications, the user can select a device from a list in order to control it with a device-specific user interface. However, this method faces *naming* issues (i.e. “what do we name the lamp on the left?”) and *scaling* issues as the number of controlled devices increases. These solutions also present a necessarily flawed mapping from the positions of the appliances in the rich, 3-dimensional world to their place in a 1D or 2D list presented on the screen. Past research has used direct aiming at target devices in space with phones to overcome these problems [2, 14]. Such techniques have a few drawbacks: the aiming device first has to be retrieved; the user’s hands have to be free for operation; and 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 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 and control of devices in physical spaces. Head-worn devices can naturally exploit the

user's head orientation, an important (but imprecise) indicator of the user's *locus of attention* [18]. It suggests the general direction, but not the particular point of focus. We draw an analogy to assistive area cursors and adapt area cursor techniques [7, 26, 5] for physical selection. 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 Glass¹ to enable infrared (IR) communication between Glass and target appliances. We contribute and evaluate new methods for addressing selection ambiguity in this context. In all our techniques, the emitted IR beam provides an initial *coarse* selection area (illustrated in Figure 1 *left*). To *refine* selection when multiple targets have received IR signals, we describe and evaluate three techniques:

Our *Naive IR* technique shows an alphabetically ordered disambiguation list on the near-eye display (Figure 1 center). A study with 14 participants finds that target acquisition with naive IR targeting is preferred by users and is faster than pure list selection without IR, but refinement is still time consuming.

Our *Intensity IR* technique improves refinement as target objects compare IR received signal strength (RSS). This value allows the system to eliminate some peripheral targets and to re-order the refinement interface's list by their intensity values. For example, in Figure 1 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 shows that *Intensity IR* successfully reduces both the probability of needing to do refinement as well as the time spent in list navigation when compared to *Naive IR*.

Our final *Head-motion Refinement* 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. We first learn the relative spatial structure of the targets using Glass' orientation sensors. Users can then 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 such as lighting and TV sets: 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.

BACKGROUND AND RELATED WORK

Our approach is related to head- and eye-controlled interfaces, area cursors, pointing in physical spaces and computer vision-based selections.

Head and Gaze Input

Head movement has long been used for virtual camera control in VR applications [15] and as an assistive input technology for cursor control of desktop applications [16]. However, it is notable that human neck muscles have a lower bandwidth than other muscle groups, e.g., the wrist [4]. 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 [9]. While there are wearable gaze trackers [3], turning information about a concrete point in space where a user is looking into a selection requires a map with known target locations. Our system works through point-to-point IR communication and does not require an *a priori* map or markers. Target objects in the environment can also be equipped with individual cameras that watch the user [22, 23]. Such an approach can enable similar benefits as our approach, but is computationally more expensive 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 2D area cursors for GUIs, the activation area of the cursor is enlarged, which facilitates acquiring smaller targets [7]. We argue that head orientation pointing has analogous characteristics (limited pointing performance and accuracy). Area cursors are especially appropriate for individuals with motor control impairments or difficulties [26, 5]. Similar ideas have also been extended into 3D to provide selection with progressive refinement in 3D scenes [1]. All area and space cursors necessitate disambiguation when there are multiple targets and no clear winner. This paper describes the trade-offs between several disambiguation approaches.

Pointing in Physical Spaces

Rukzio et al. [20] 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. Several other approaches to spatial selection with handheld devices [2, 14, 25, 21, 8] or finger-worn devices [11] exist.

In some techniques, users select objects of interest with laser pointers; however, the laser dot's 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 [25, 10] – in contrast, our technique does not require a static map ahead of time. In the FreeMote system, [6] an IR camera in a handheld controller interprets readings from IR emitters on target appliances. In contrast, we explore a lower cost alternative, using only IR emitters and receivers. Our system tackles an unresolved issue of prior approaches – navigating an area dense with potential targets and resolving selection ambiguity.

¹<http://www.google.com/glass/start/>

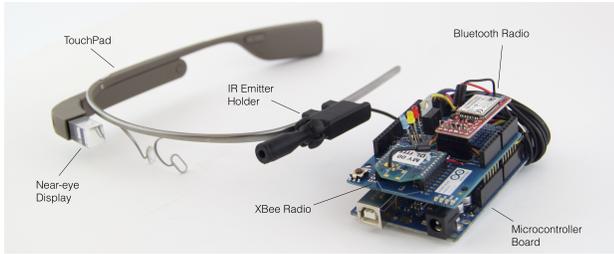


Figure 2. Our Glass hardware: Google Glass augmented with a repositionable IR holder, and an additional microcontroller that communicates with Google Glass and controls IR emitter.

Vision- and Projection-Based Target Selection

Many alternative solutions for detecting devices in contained spaces rely on computer-vision recognition of printed tags on devices. Unfortunately, these methods either impose significant constraints on the camera used for detection [12], require large or obtrusive tags [13], or are designed to work specifically at short distances [19]. Passive markers also cannot show visual feedback in the environment. 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 [21] and RFIG Lamps [17]) or spatially (using QR codes in the infrared spectrum in SideBySide [24]). Our solution is similar in spirit but requires only small, low-cost IR emitters and detectors.

IR FOR HEAD ORIENTATION-BASED TARGETING

In this section, we present our hardware platform (IR targeting) and interaction model (head orientation-based) that we will use throughout our iterative designs.

Hardware

We hypothesize that infrared (IR) emitters are a good technology match for head orientation selection, since they emit light within a given angle, resulting in a cone in front of the emitter where the light is visible. IR LEDs with many different beam angles are commercially available. We augment Google Glass with a 940nm 5mm IR emitter with 10° beam angle (OSRAM SFH 4545). The emitter is controlled by an additional microcontroller which communicates with Google Glass through Bluetooth radio, since Glass does not directly support hardware modification (see Figure 2). Target devices use Vishay TSOP38238 IR receivers. Data is encoded using standard IR remote protocols at 38.0kHz.

IR signals are used for initial line-of-sight targeting; subsequently, we use bidirectional wireless communication to send information such as target IDs and signal strength from targets back to Glass. We have chosen the commercial off-the-shelf ZigBee implementation (XBee based on 802.15.4 radio) for this purpose (see Figure 3). This architecture was mostly chosen for reasons of expediency and we do not claim optimality for prototyping decisions. Future head-mounted devices could clearly integrate IR emitters; other wireless techniques (WiFi, Bluetooth) can also be used.

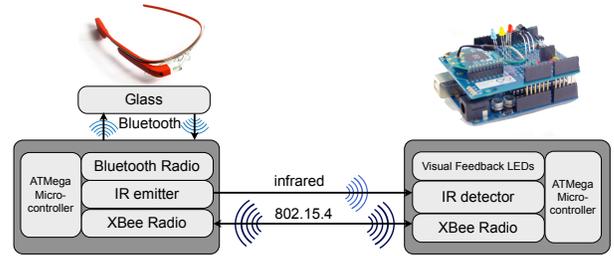


Figure 3. Our System architecture diagram. The selection is initiated through infrared but confirmed over 802.15.4.

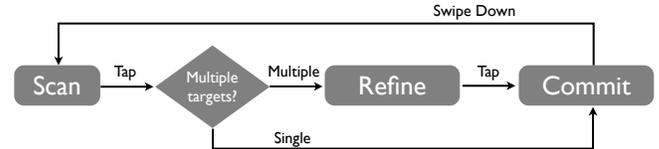


Figure 4. *scan* and *refine* – two main stages during the interaction with HOBS. For completeness, we have also added the final state of *commit*.

Interaction Model

From the user’s perspective, interaction with HOBS proceeds in two stages (Figure 4):

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 visual 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 to refine his selection.

Refine: When disambiguation is needed, the user must make an explicit selection among the targets within their 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 the 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?*

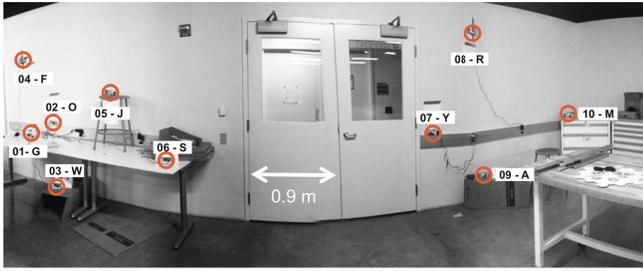


Figure 5. In the targeting study, participants were asked to find and select one of 10 targets in the lab environment.

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.

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 5), each with a number ID and a letter representing the name. For the study, we recruited 16 participants from our institution (9 males and 7 females) by email. Participants included undergraduate and graduate students, as well as university staff. Their educational backgrounds included Information Science (6), Engineering (4), Math and Science (4), Design (1), Others(1). 14 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. At the beginning of each acquisition, participants were asked to stand at a fixed position approximately 10-12 feet away from the nodes, looking down until a target is announced. Participants were allowed to physically move towards targets if they decided that this would help them with the task. After each task, they were asked to move back to the starting position. Afterwards, participants were asked to complete a survey of primarily open-ended questions about their experience.

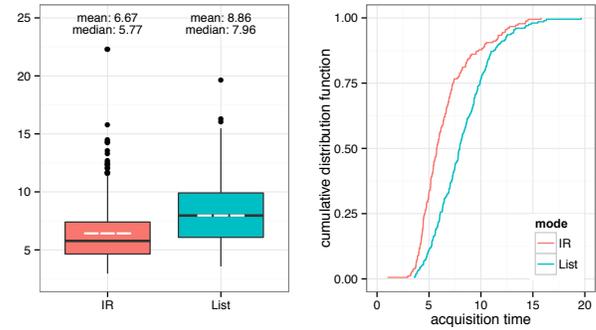


Figure 6. Boxplot of target acquisition time for *IR selection* and *list selection* is shown on the left. The center is the median value, and the mean value is shown using white dashed lines. The cumulative distributions are on the right.

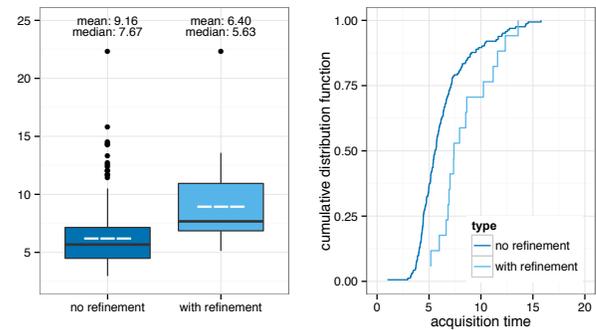


Figure 7. Boxplot and CDF of target acquisition time when refinement is needed or not in *IR selection*.

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 seconds (see Figure 6). A t-test shows a significant difference ($t(279) = -3.81, p < 0.001$).

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 7). This difference is significant ($t(19) = -2.7827, p < 0.05$). Because many targets were spaced far apart, refinements were only necessary in 10% of total *IR selection* trials in this study.

To further generalize the results, in Figure 8, we show the acquisition time for each individual target in the *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.

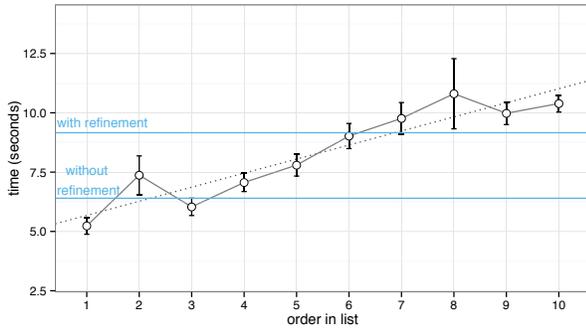


Figure 8. Times taken to select a device vs. its order in the list selection. The dotted line is a linear fit between the average time and target orders in the list. Two horizontal solid lines are the average target acquisition times in *Naive IR* when refinement is needed or not.

In the elicited qualitative feedback, 11 out of 16 participants preferred IR selection over list mode (3 preferred list, 2 were undecided). When asked: “When you have multiple candidate targets within sight, which method for selecting between them did you use?”, 81% chose *adjust head so that only one candidate is signaled*, 56% chose *tap (to enter refinement stage) and then choose between list*, and 25% chose *walk closer to the candidate for a better result* (multiple choices were allowed). While we observed more participants walking one or two steps during some tasks, this suggests that participants viewed their moving as a followed-through action of adjusting their heads instead of an intentional to walk closer to the targets.

While both interfaces were judged similarly on overall ease of connecting, IR selection was perceived to be more direct and pleasant. One user noted benefit of *IR selection* “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”.

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

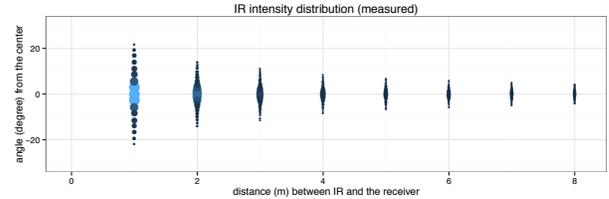


Figure 9. Empirical measurement of IR intensity at different positions. We measure the intensity on a horizontal plane at one-meter intervals from the IR emitter. On that plane, a sample is taken every 5 centimeters, and we stop when the intensity reading is around the level of ambient noise. To match our assumption about the relationship between angles and intensities, the vertical axis shows the relative degree of the angle rather than the distance to the center of the plane. The size and color brightness represent the intensity of the readings for visualization.

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 9. Our measurements confirm that angular difference has a large effect on the intensity readings, with rapid fall-off at increasing angles.

The intensity information is used in two ways:

1. When multiple targets have received IR signals and reported the intensity readings, we discard those whose intensities are significantly lower than the largest reading². Therefore, when there is only one target within the line of sight, the IR intensity approach has the same behavior as the previous iteration - no disambiguation is needed. When the environment becomes more populated, the new design can filter some peripheral targets out, reducing $P(\text{refine})$, 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 re-positioned the 10 nodes and set them up in a smaller area (see Figure 10). 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

²In 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.

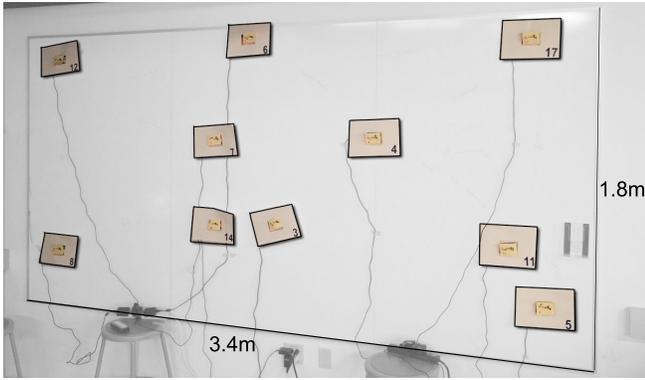


Figure 10. The environment setup for our second study. In comparison to our first study, we have deliberately increased the target density.

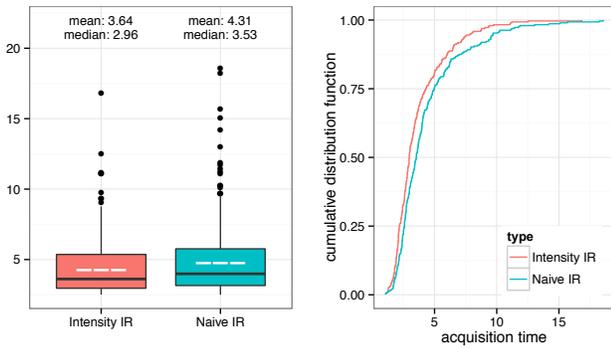


Figure 11. Boxplot and CDF of the target acquisition time in *Naive IR* and *Intensity IR* conditions.

300 trials. A Chi-square test shows this difference is significant ($\chi^2(1) = 24.755, p < 0.001$). 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 (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 $\chi^2(1) = 15.758, p < 0.001$).

From Figure 11, 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, the target was accidentally eliminated in 13 (4.3%). This is higher than the error rate for *Naive IR* (5 out of 300, 1.6%). Even though the higher error rate increases the chance of multiple attempts in

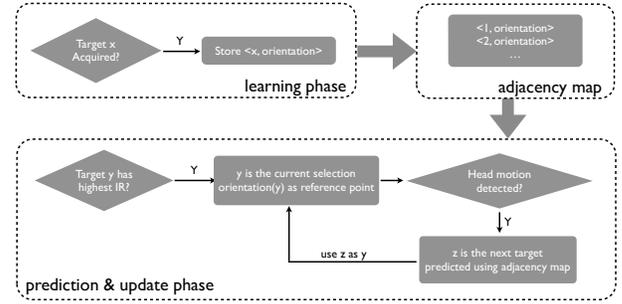


Figure 12. Our third technique learned each target’s absolute orientation and construct the adjacency map. During the *refinement* stage, the prediction is based on relative changes to a reference point.

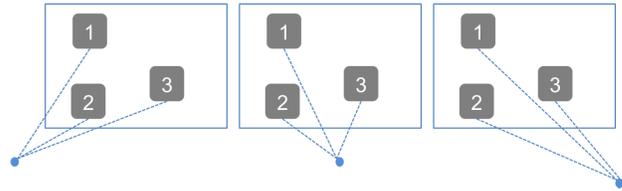


Figure 13. This illustrates that the change of a user’s absolute position doesn’t change the relative relationship of physical targets.

target selection, our analysis above illustrates that overall performance of *Intensity IR* is still better than *Naive IR*.

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 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 research 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, as the user scans over targets in their normal use, the system attains the absolute orientation of each device from IR and motion sensors. From this information it can abstract out the relative positions of targets and construct an *adjacency map*. The absolute orientations in the map cannot be applied to all indoor environments, since the user’s movements through the space could change the relationships between targets. However, with the constraint that the targets are spread around the periphery at similar distances, their relative orientations are stable (see Figure 13). This learning process can be transparent to the user and the map can be built without an explicit calibration phase.

After the map is created, the user can enter a quasi-mode for refinement by holding down on the touchpad. In this quasi-mode, a single selected device lights up at a time. When the user turns his head in the direction of another device, the selection (and indicator light) switches to that device. Therefore, the user can move between devices one at a time with slight head movements. This prediction is implemented by first calculating the user's direction of head motion, and searching through the adjacency map for the nearest device in that direction. To calculate the direction, we maintain a circular buffer of the last 10 sensor measurements, passed through a low pass filter. The direction is then calculated as the difference between the last sample and the first sample. We use a hysteresis method to avoid spurious selection changes — if the variance of the buffer is below a threshold, we assume no movement has occurred and do not select another device.

Evaluation

We evaluated the head motion refinement method through an informal study and collected qualitative feedback from a subset of 4 users from the Iteration 2 study. In this evaluation, we asked users to cycle through multiple targets using the new quasi-mode.

The users had strong preferences for the new method of refinement. Our observations suggest that each trial became much easier than previous iterations. We conducted a survey to collect qualitative feedback after the experiment. 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. All users 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”*.

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 device-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.

Our prototype, includes three smart devices: a lamp and a fan that could be switched on and off; and a smart TV with playback, volume and navigation controls (see Figure 14). The prototype used *Naive IR*, as the small number of target devices made selection without disambiguation possible. The appliances are switched with 120V AC relays. The smart TV is a 30” display connected to a laptop. User interfaces for each were pre-defined in our application.

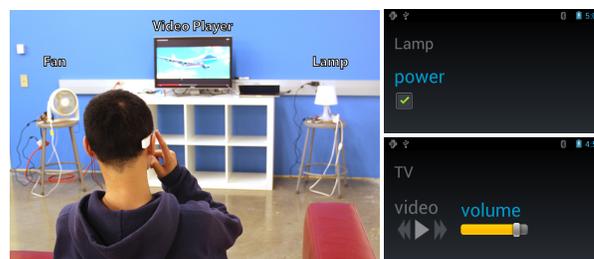


Figure 14. In the smart home scenario, we have built three smart appliances for a user experience study. The interface supports both simple on/off appliances (lamp or fan) and multi-functional appliances (TV, for example).

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. The tasks included turning off a lamp, playing a movie and turning up the volume.

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 studies. 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”*.

While our system enables users to select and control the example devices successfully, work remains in devising appropriate interfaces for complex devices. Users rated ease of use higher for the lamp and fan which had simple, discrete on/off actions, and lower for the TV control, which had more options. Multiple participants remarked that the difficulty was based on the affordances of Glass: *“most of the difficulty I had with Glass came from having to navigate the interface on the tiny screen with the touch pad”*. We leave addressing this fundamental usability challenge of wearable devices with near-eye displays to future work.

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 HOBS, 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 ori-

entation 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 $P(\text{refine})$, 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-based 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

Our system faces a few 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.

We also 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.

ACKNOWLEDGMENTS

This work was supported in part by the TerraSwarm Research Center, one of six centers supported by the STARnet phase of the Focus Center Research Program (FCRP) a Semiconductor Research Corporation program sponsored by MARCO and DARPA. Additional support was provided by a Sloan Foundation Fellowship and a Google Research Award.

REFERENCES

1. Bacim, F., Kopper, R., and Bowman, D. A. Design and evaluation of 3d selection techniques based on progressive refinement. *International Journal of Human-Computer Studies* 71, 7 (2013), 785–802.
2. 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.
3. 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.
4. 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.
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 23rd annual ACM symposium on User interface software and technology*, ACM (2010), 153–162.
6. Hipp, M., Mahler, T. D., Spika, C., and Weber, M. Universal device access with freemote. In *Intelligent Environments* (2009), 311–318.
7. 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.
8. 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.
9. 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.
10. 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).
11. 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.
12. Mohan, A., Woo, G., Hiura, S., Smithwick, Q., and Raskar, R. Bokode: Imperceptible visual tags for camera based interaction from a distance. In *ACM SIGGRAPH 2009 Papers, SIGGRAPH '09*, ACM (New York, NY, USA, 2009), 98:1–98:8.
13. Moran, T. P., Saund, E., Van Melle, W., Gujar, A. U., Fishkin, K. P., and Harrison, B. L. Design and technology for collaborage: Collaborative collages of information on physical walls. In *Proceedings of ACM UIST*, ACM (New York, NY, USA, 1999), 197–206.

14. Patel, S. N., and Abowd, G. D. A 2-way laser-assisted selection scheme for handhelds in a physical environment. In *UbiComp 2003: Ubiquitous Computing*, Springer (2003), 200–207.
15. 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.
16. 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.
17. 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.
18. Raskin, J. *The Humane Interface: New Directions for Designing Interactive Systems*. ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, 2000.
19. Rekimoto, J., and Ayatsuka, Y. Cybercode: Designing augmented reality environments with visual tags. In *Proceedings of DARE 2000 on Designing Augmented Reality Environments*, DARE '00, ACM (New York, NY, USA, 2000), 1–10.
20. 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.
21. 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.
22. 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.
23. 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.
24. 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.
25. 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.
26. 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.