Exploring Gestural Interaction in Smart Spaces using Head Mounted Devices with Ego-Centric Sensing

Barry Kollee
University of Amsterdam
Faculty of Science
Amsterdam, Netherlands
barrykollee@gmail.com

Sven Kratz
FX Palo Alto Laboratory
3174, Porter Drive
Palo Alto, USA
kratz@fxpal.com

Tony Dunnigan
FX Palo Alto Laboratory
3174, Porter Drive
Palo Alto, USA
tonyd@fxpal.com

ABSTRACT

It is now possible to develop head-mounted devices (HMDs) that allow for ego-centric sensing of mid-air gestural input. Therefore, we explore the use of HMD-based gestural input techniques in smart space environments. We developed a usage scenario to evaluate HMD-based gestural interactions and conducted a user study to elicit qualitative feedback on several HMD-based gestural input techniques. Our results show that for the proposed scenario, mid-air hand gestures are preferred to head gestures for input and rated more favorably compared to non-gestural input techniques available on existing HMDs. Informed by these study results, we developed a prototype HMD system that supports gestural interactions as proposed in our scenario. We conducted a second user study to quantitatively evaluate our prototype comparing several gestural and non-gestural input techniques. The results of this study show no clear advantage or disadvantage of gestural inputs vs. non-gestural input techniques on HMDs. We did find that voice control as (sole) input modality performed worst compared to the other input techniques we evaluated. Lastly, we present two further applications implemented with our system, demonstrating 3D scene viewing and ambient light control. We conclude by briefly discussing the implications of ego-centric vs. exo-centric tracking for interaction in smart spaces.

ACM Classification Keywords

H.5.2 User Interfaces: Input devices and strategies

Author Keywords

head-mounted device (HMD);smart spaces;interaction techniques;modalities;ego-centric;hand gestures

Permission to make digital or hard copies of all or part 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 components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions@acm.org.
SUI '14, October 04 - 05 2014, Hololulu, HI, USA
Copyright is held by the owner/author(s). Publication rights licensed to ACM.
ACM 978-1-4503-2820-3/14/10. . .$15.00.
http://dx.doi.org/10.1145/2659766.2659781

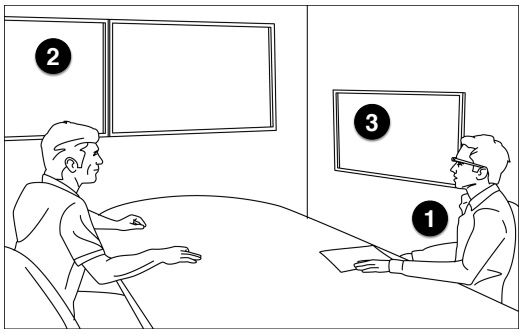


Figure 1. An experimental setup of a "smart" office environment. Users, such as the person on the right (1) can use their HMD to interact with the displays (2, 3).

INTRODUCTION

Wearable devices are currently increasing in popularity. An indication of this the increase of the number of head-mounted devices (HMD) and other wearables such as smart watches that are becoming available commercially. At the same time, many spaces within office buildings are becoming "smart", i.e., spaces such as meeting rooms are being equipped with a proliferation of devices, such as displays, wireless speakerphones and remotely-controlled lighting, that can be interconnected via network over wireless or wired connections.

In this paper, we explore the use of wearable devices, specifically HMDs, for interaction in such *smart spaces*. Most currently available commercial HMDs, such as Google Glass, have a relatively limited array of input options. The Google Glass devices, for instance, rely mainly on voice commands and a rim-mounted touch pad, although there is also support for head-tilt gestures using third party libraries[1]. In the context of smart space interaction this poses usability problems, as, for instance, it can be difficult to select external devices in the environment. Previous works have already tried to address this problem [6], although the interactions presented there still seem cumbersome as they still rely on the limited existing input mechanisms provided by the HMD they used. A further issue which we touch tangentially is the question whether smart spaces should be instrumented to (externally) track users, or if the users

[1]E.g., "Head Gesture Detector", `https://github.com/thorikawa/glass-head-gesture-detector`

themselves should be instrumented for input tracking in an ego-centric manner, which is possible using HMDs with embedded sensors, e.g., depth cameras, that are able to track gestural input. Although, in this paper, we do not make a direct comparison between ego-centric and external (i.e., exo-centric) tracking, we do highlight some of the interaction possibilities for ego-centrically tracked users working in smart spaces.

We propose the use of in-air hand gestures to increase the input possibilities for HMDs. We believe that hand gestures have the potential to make interaction easier not only with the HMD itself, but also with external devices and displays in a smart space scenario. However, since there are several modalities besides gestures available to user interface designers of HMDs or other types of wearables, e.g., touch pad, voice commands, head gestures and hand gestures, we are also interested in evaluating the usability of these different modalities for input tasks on HMDs with a focus on interaction in smart spaces.

Specifically, we contribute an interaction scenario that we use as a test bed for gestural interactions with external displays, using an HMD that allows ego-centric sensing of gestural input by the user. To realize this scenario, we contribute a prototype HMD system that uses ego-centric tracking to enable in-air hand gestures as input, in addition to the other modalities mentioned previously. Furthermore, we present the results of two user studies centered around a smart-space interaction scenario. In the studies, we examine the usability of a range of input techniques available through our prototype for smart space interaction.

In the first study, we gather qualitative feedback through an elicitation study on the usability of several gestural input techniques for an interaction scenario involving and HMD and a smart space. Based on the insights gained in the first study, we conducted a second user study using the prototype HMD system we developed. In this study, we intended to find out if the results of the elicitation study are reflected in an user evaluation of an actually running software and hardware prototype.

Lastly, we describe two further applications of our prototype application: a 3D scene viewer on a large display, and abient light control.

RELATED WORK

In the following, we provide an overview of related work, with a focus on the interaction techniques we study in this paper.

Head Orientation

Chen et al. [6] showed ways how physical home appliances could be controlled by using head orientation as input for their prototype. They compared head orientation selection versus the device's touchpad (i.e. list-view selection) to control these home appliances. They found that if more than 6 home appliances are selectable, head orientation input outperformed the list view of the device's touchpad. Meaning that home appliances could be selected and controlled faster with head orientation input than with touchpad input. We thus incorporated head orientation as a candidate technique for our prestudy.

Hand Tracking and Gestures on Wearable Devices

Hand tracking is not yet implemented in mainstream HMDs (such as Google Glass) and is still an ongoing field of research as seen, e.g., in the MIME prototype by Colaco et al. [7]. In contrast to MIME, the sensing setup used by our system allows dual-handed applications as we can track both the user's hands. In contrast to the present paper, MIME does not address how external devices could be controlled using in-air gestures through HMD-based tracking.

Bailly et al.'s ShoeSense [2] is a shoe-mounted wearable system that explored gestural interactions in the space in front of the user. While ShoeSense did solve many problems associated with minimum range limitations of the depth sensors at that time, we believe that a head-mounted sensor might be more practical and, in certain situations, more socially acceptable. One reason for this is that shoe-mounted cameras can be perceived as obtrusive, due to their "upskirt" orientation.

Mistry et al. presented a chest-worn projector/camera system that could track the user's hand gestures [14]. However, this system required colored markers on the user's fingers for tracking, which in many cases cumbersome and should be avoided, according to the design considerations proposed by Colaco et al. [7].

A further, highly related work was carried out by Piumsomboon et al. They conducted an elicitation study for hand gestures in AR [15]. In exploring HMD-based gestures in smart spaces, the current work address a somewhat different domain, nevertheless many of the gestures elicited the authors could be incorporated into our system.

Touch Control

The most common form of input on current mobile devices of the phone or tablet class is touch input. Multi-touch allows users to interact using multiple fingers and enables relatively complex gestures to be entered and also provides the opportunity to provide physical analogies (i.e., *Reality-Based Interfaces* [11]) in the interfaces to improve the usability. Thus, it is not surprising that current HMDs have moved towards using touch as the main input modality, seen, e.g., in the rim-mounted touchpad of the Google Glass device [10]. We argue, however, that touch might not be the best way of interacting, as, for example, the size of the touchpad on head-worn devices such as Glass is limited by the form factor of the device, and can only support relatively coarse input. Lastly, Malik et al. studied multi-touch input for control of distant displays [13].

Interaction with External Devices

HMDs can communicate with external devices via a wired or wireless connections. Examples of such devices could be small embedded wireless devices, e.g., Chen et al. [6] or and smartphones [10].

A large body of work has been completed on the topic of interaction across multiple devices. In this paper, we present a prototype that uses interactions similar to Rekimoto's "Pick and Drop" [16]. This allows the user, through a selection and a deselection technique, to "pick" a digital item on one device or display, and "drop" it on another device or display.

Multi-Display Environments and Smart Spaces

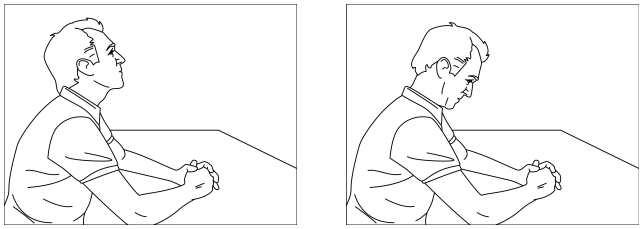
A large body of related work has discussed interaction in multi-display environments. For instance *PointRight* [12] discusses the implementation of input redirection such environments. We believe that many issues in input direction could be solved by using direct manipulation via gestures, as suggested in this paper. With the *LightSpace* project, Wilson et al. [19] realized the direct manipulation concept in a highly instrumented space. Our current research works in the direction of realizing such interactions through ego-centric tracking.

PROTOTYPE ENVIRONMENT AND SCENARIO

Modern work environments contain spaces such as conference rooms that are equipped with a large amount of technology, such as displays, projectors, audio and telecommunication systems, climate control and lighting systems. However, the interface to this technology is often cumbersome and heterogeneous. Thus, we believe that new and improved ways of interacting with infrastructure that is present in such *smart* spaces need to be found. Therefore, we devised a scenario that allows us to prototype and evaluate HMD-based gestural interactions in smart work spaces.

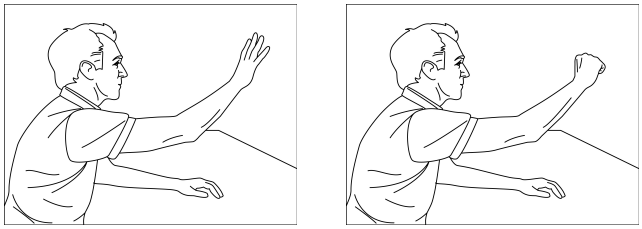
Specifically, our scenario describes a brainstorming session in a modern architect's office. The architects are trying to decide what type of stone pattern they want to use in an interior design project. Our proposed smart office space allows the architects to organize and compare the types of stone patterns by means of placing images of the patterns next to each other on multiple displays in a "smart" conference room. Instead of a normal desktop-based interface, they make use of an HMD that can track mid-air hand gestures and also control the displays in the conference room (Figure 1). Our implementation of this scenario consists of a conference room with multiple displays that allow interaction using a HMD (as in Figure 1).

Boring et al. [3] previously showed how visual content could be selected, controlled and transferred from one screen to another by using a smartphone and the embedded camera from the device. Our prototype environment replicates this setup in some aspects. However, we have replaced the mobile device by an HMD, and input on the phone's touch screen with in-air gesture input. In this paper, we investigate whether a set of input tasks as required by our scenario can be accomplished by a HMD using two types of hand gestures as well as voice and HMD-based GUI input.



(a) Head Gestures (**T1**)

(b) Hand Tracking (**T2**)



(c) Hand Gesture (**T3**)

Figure 2. Proposed gestural techniques for gestural interaction with external devices using an HMD with ego-centric tracking.

Interaction Techniques and Tasks

In our scenario, the task (i.e. moving an image thumbnail from one screen to another) that needs to be performed by the user consists of three input steps: (1) Select the display, which holds the image thumbnail that needs to be transferred, (2) select the desired image thumbnail on the selected display and (3) select the display to which the image thumbnail needs to be transferred to.

To accomplish these tasks, we propose two mid-air gestural input techniques, *hand gestures* and *hand tracking* as well as *head gestures*, *voice commands* and the use of the rim-mounted *touchpad* on a Google Glass device. In the following, we will will explain the proposed interaction techniques in more detail:

Head Gestures

When using head gestures in our scenario, the user gazes towards the display or image thumbnail that needs to be selected (Figure 2(a)). Selection takes place by means of a "nod" gesture (i.e., moving the head up and down).

Hand Tracking

This interaction technique consists of the tracking of the hand's position with respect to the HMD (Figure 2(b)). By means of a push movement in space (i.e., a rapid shift in z-axis position) a selection takes place.

Hand Gestures

By hand gestures we refer to detecting certain poses of the hand. Colaco et al., proposed detecting hand gestures for interaction [7]. For our tasks, we chose to use a grasp gesture (opening and closing the hand) to select items on the displays or the displays themselves (Figure 2(c)). We believe that grasp gestures are appropriate for content manipulation on an external display as this follows the Natural User Interface (NUI) paradigm [18]. A typical interaction using hand gestures would work as follows:

1. The user selects the display by looking at the display and grasping it in space
2. The user opens up his hand and hovers over the thumbnails. When the desired image thumbnail is in the view of the user, the user grasps the thumbnail by holding his or her hand as a fist.
3. The user looks towards the display where the image thumbnail needs to be transferred to and opens his or her hand again.

Voice Commands

We assume that many future HMDs will incorporate voice command functionality. Thus, we decided to allow voice commands in our scenario. Because we cannot predict the ability of future devices to recognize complex grammatical constructs, e.g., "Select item 3 on display 1 and transfer it to display 2", we chose very simple commands that follow a verb and noun structure, e.g., "Select display 1". Furthermore, the choice of voice commands allows the user to replicate the same (atomic) interaction steps as in the other modalities we propose, thus making a comparison easier.

Touchpad

As a baseline technique, we use the existing touchpad input capabilities of Google Glass. The touchpad is used to interact with a selection list that contains all the selectable screens and display items. The list is displayed in the Glass device's display. The selectable displays are displayed (in numerical order) at the beginning of the list, and the selectable items are displayed (in numerical order) in the remainder of the list.

ELICITATION STUDY

Before implementing the previously discussed scenario as a software prototype, we conducted a prestudy to elicit qualitative user feedback on the gestural input techniques we discussed in the previous section, i.e., head gestures (**T1**), hand tracking (**T2**) and hand gestures (**T3**). The reason we wanted qualitative feedback was to gain an insight into which gestural input technique might be acceptable to users for the proposed scenario.

Methodology

To elicit qualitative feedback from test subjects, we explained the scenario to them and then, using a within-subjects design, showed each test subject three videos[2] demonstrating the use of **T1**–**T3**, respectively.

After viewing the video for each technique, participants were asked to fill out a questionnaire. The questionnaire consisted of three parts. In the first part, participants were asked to fill in five Likert-Scale questions with respect to *Effectiveness*, *Learnability*, *Practicality*, *Intuitiveness* and *Comfort*.

In the second part of the questionnaire, we elicited emotional reponses to the interaction examples shown using a custom variant of the *EmoCard* approach [1]. In this method, we allowed the test subjects to distribute up to three "points" to any emotion on the EmoCard scale. We believe that this approach allows users to either express a wider range of emotions or strongly express single emotions while rating a technique.

Finally, we asked users, after viewing the demonstration video and filling out the questionnaires described previously, to provide three preference rankings of the current gestural technique vs. baseline[3] commands, which were *Voice Command* (**B1**) and *Touchpad* (**B2**), for the following tasks: "Select a display", "Select a visual item on the display" and "Transfer the visual item to another display".

In total, 16 participants from an industrial research lab participated in the study. The age range was 22–55 years, 4 participants were female.

Hypotheses

Prior to our study, we assumed that the techniques using in-air hand gestures would be superior to head gestures, and that hand gestures would possibly be rated superior to voice commands or using the HMD's touchpad. Thus, we formulated the following hypotheses:

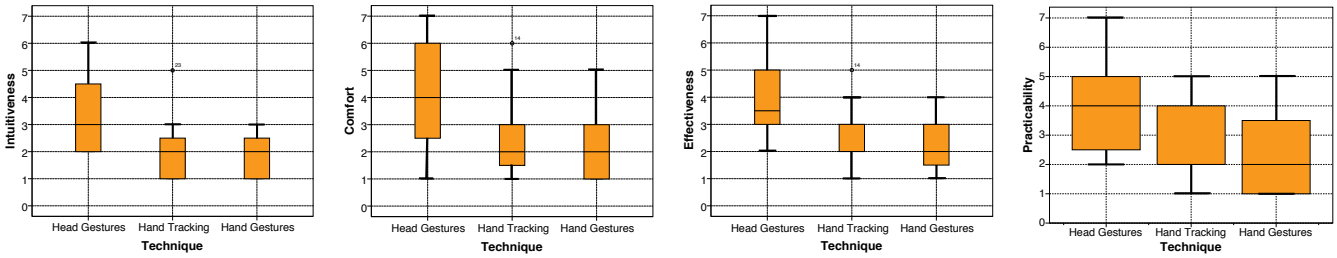
- **H1:** the input techniques using in-air hand gestures (**T2** and **T3**) are rated more favorably in the Likert Scale questions as well as receive more positive EmoCard emotion ratings than head gestures **T1**.
- **H2:** gestural input techniques are ranked first more often than the baseline techniques **B1** and **B2**.

Findings

The results of the Likert-scale questionnaires indicate that head gestures were rated less favorably than hand gestures in terms of *intuitiveness*, *comfort* and *effectiveness* (Figure 4). A nonparametric Friedman test on the rating data shows a significant difference within these measures with $p = 0.001$ for *intuitiveness* and $p < 0.001$ for *comfort*, $p = 0.001$ for *effectiveness* and $p < 0.001$ for

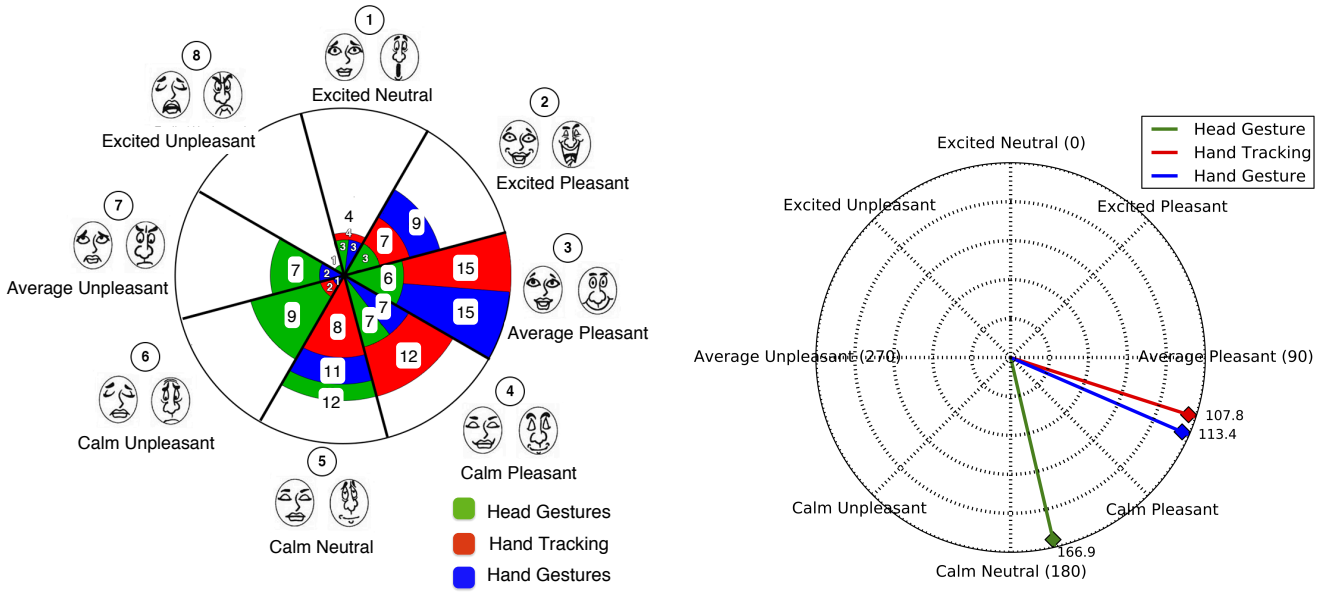
[2]The videos shown to the test subjects can be viewed online at `https://vimeo.com/102943456`.

[3]We derived the baseline techniques from the set of input capabilities currently available in Google Glass HMDs.



(a) intuitiveness (b) comfort (c) effectiveness (d) practicability

Figure 4. Box plots of the results of the prestudy Likert scale questionnaires rating the *intuitiveness*, *comfort*, *effectiveness* and *practicability* of the interaction techniques. A *lower* rating is better.



(a) Frequency Slice Chart of EmoCard Ratings (b) Mean EmoCard Ratings

Figure 5. (a) Frequency slice chart of ratings on EmoCard sectors. (b) Mean EmoCard Ratings plotted as angles on the EmoCard scale.

practicability. We did not obtain significant results for *learnability*.

We received a total of 48 emotion ratings for head gestures using the custom method we described previously (i.e., 16 participants×3 distributed emotions for head gestures). 28 ratings (58%) were given in the "Average Unpleasant" to "Calm Neutral" sectors (Figure 5(a)). The EmoCard results (Figure 5) are consistent with the results for intuitiveness, comfort and effectiveness of the Likert scale questionnaires. Figure 5(a) shows a frequency slice chart that presents a histogram of the test subjects' ratings plotted on the EmoCard sectors. We can see that head gestures (green color) were rated more towards the neutral/unpleasant side of the EmoCard scale.

In an alternative analysis, we assigned an angle value to each EmoCard emotion (e.g., "Excited Neutral" = 0, "Calm Neutral" = 180, etc.) and calculated the average angle. Figure 5(b) shows the average EmoCard angles plotted on polar coordinates. It is clear that head gestures (green) are centered mostly around the "Calm Neutral" sector and the hand gesture techniques are on average in the sector between "Calm Pleasant" and "Average Pleasant". An ANOVA on the converted EmoCard ratings shows a significant difference ($f_{1,44} = 4.666$, $p = 0.036$). A Bonferroni post-hoc comparison shows a significant difference between Head Gestures (**T1**) and **T2** ($p = 0.030$) as well as a marginally significant difference between (**T1**) and **T3** ($p = 0.056$).

In summary, the elicitation study indicates that input via head gestures (**T1**) was rated less favorably by the test subjects in terms of *intuitiveness, comfort* and *effectiveness* as well as EmoCard-based emotional response than hand tracking (**T2**) or hand gestures (**T3**). Furthermore, gestural techniques (**T1**–**T3**) were ranked first more often than the baseline techniques **B1** and **B2**. We can thus confirm our hypotheses **H1** and **H2**.

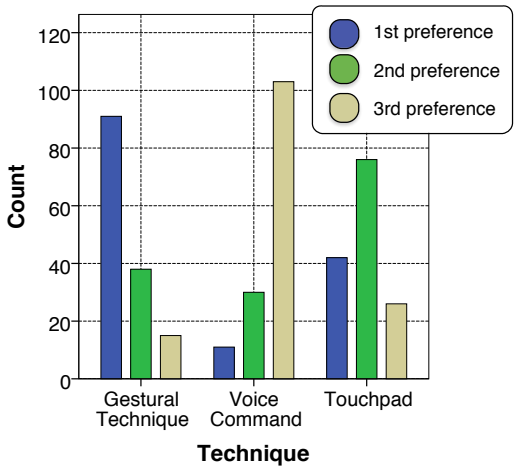


Figure 3. Ranking frequency of gestural techniques vs. basline techniques.

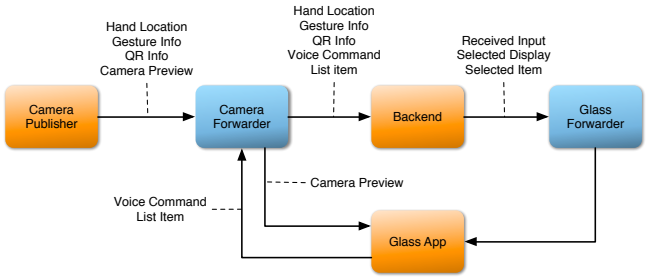


Figure 6. This system diagram shows the direction and type of messages that are sent between the three software components that comprise the prototype.

PROTOTYPE

We developed a software and hardware prototype to implement the scenario and interaction techniques proposed earlier. Because head gestures (**T1**) were rated unfavorably in the elicitation study, we decided not to implement them in our prototype and also not to study head gestures as an input technique in the subsequent evaluation. The prototype software consists of three intercommunicating applications: a C++ application that publishes depth sensor tracking data, a Java backend application for state management and graphical output on large displays and an Android application running on a Google Glass device. Figure 6 provides an overview of how the system components communicate with each other. In the following, we describe each component and associated hardware of the prototype in more detail:

Gestural Input using Head-Mounted Depth Camera

Currently available HMDs, such as Google Glass, do not have the capability to capture depth information or provide spatial hand tracking information. Therefore, we used a Creative SENZ3D short-range depth camera[4] to fetch the spatial position (i.e., real-world x, y and z coordinates) of the user's hands. We developed a C++ application using the Intel Perceptual Computing SDK[5] (PCSDK), OpenCV[6] and ZBar[7] libraries to publish hand tracking information, a low-resolution camera preview image stream and QR Code information for use by the backend application.

[4]The maximum IR depth resolution of this sensor is 320x240 and its diagonal field of view is 73°. The sensor captures depth and RGB images at a rate of 30 Hz.

Figure 7. Depth camera head strap as a substitution for embedded hand tracking in the HMD.

The PCSDK is used to fetch finger tracking information from the depth sensor and to provide simple gesture recognition capabilities, e.g., "hand close", "hand open" or "thumbs up" gestures. OpenCV is used to generate a low-resolution preview of a 256 × 256 camera center region shown in the Glass Device's display (shown in Figure 8), which is intended to ease targeting of QR Codes for display or item selection, as the camera view can be slightly misaligned with respect to the user's gaze direction. ZBar is used to detect QR codes based on the depth sensor's RGB image stream.

We engineered a customized strap that enables the user to head-mount the depth sensor in order for hand tracking and gesture recognition to function (Figure 7). This is intended to be a substitute for future HMD hardware with built-in sensing capabilities for user hand tracking. Users are able to wear the head strap at the same time as a Google Glass device, which completes the hardware of our prototype.

Backend Application

The backend application is implemented in Java. It makes use of the Processing Library[8] for graphical output and manages the interaction state. The backend application receives hand tracking and gesture events from the camera publisher module, voice command and selection events from the HMD.

In addition to processing user input events, the backend produces the graphical output shown on the displays of the smart office environment. On these displays the backend displays visual identifiers (QR Codes) that can be observed by the depth sensor. This is used as a simple way of determining the item the user wants to select on the display. As previously mentioned, as a targeting

[5]`https://software.intel.com/en-us/vcsource/tools/perceptual-computing-sdk/home`
[6]`http://opencv.org/`
[7]`http://zbar.sourceforge.net/`
[8]`http://www.processing.org/`

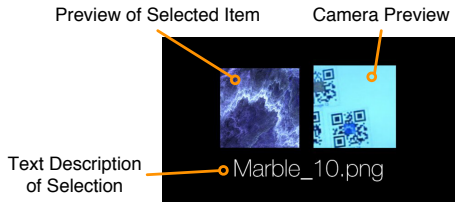


Figure 8. Screenshot of the HMD application.

aid for the user, a 256×256 pixel image of the center region of the RGB image captured by the depth sensor is shown in the HMDs display (Figure 8).[9]

HMD Application

To realize the HMD portion of our proposed scenario, we implemented an Android-based HMD application for Google Glass that provides visual feedback to the user while interacting in the smart office. In addition to a visual and textual description of the currently selected item, the HMD application shows a camera live view to aid target selection using hand gestures.

The HMD application also allows the user to use voice commands or a list-based selection mechanism that uses Glass' rim-mounted touchpad. Voice commands use Google's voice recognition service, which samples user audio, sends it to a cloud service and returns the recognized voice input as a string. To improve the detection accuracy for our voice command set, we use a *Hamming Distance* measure to enable somewhat more relaxed string matching for detected voice commands. We defined a Hamming Distance of 2 to still be acceptable for matching voice commands. The list-based selection mechanism presents the user with a vertical list that contains all known displays and selection items. The user can swipe through this list and on each tap event a message is forwarded to the Backend application containing the unique identifier of the selected item.

Messaging

Since the three software components of our prototype are required to communicate with each other, we use ZeroMQ[10] (ZMQ) to implement messaging via sockets. We designed the messaging architecture in our prototype using publish/ subscribe semantics. The blue boxes in Figure 6 represent ZMQ "forwarder devices" that implement the messaging semantics. Our system uses a local WiFi hotspot to transmit messages.

USER STUDY

We conducted a second user study to obtain quantitative usability information on an actual prototype that implements the scenario outlined in Section . Specifically, we wanted to find out if there was a difference between the two proposed gestural techniques **T2** (Hand Tracking) and **T3** (hand gestures) and the baseline techniques **B1** (voice command) and **B2** (list view), and generally, if the gestural techniques would be feasible alternatives to **B1** and **B2**. We decided against comparing our proposed gestural techniques with a more traditional baseline technique, such as mouse input. Mouse input on large and distant displays has been extensively studied in the literature. Robertson et al. illuminate a number of user experience issues of mouse input on large displays in [17]. As an alternative, Forlines et al. as well as Malik et al. argue for using touch input when interacting with tabletop-size or distant displays in situations where more than one user is using the displays [9, 13]. Lastly, we are specifically interested in comparing input techniques that are supported by current or potential future HMDs, so studying mouse our touch pad input would not gain us further insight into interaction tasks with HMDs.

Participants and Apparatus

We recruited a total of 12 participants (2 female) from an industrial research lab. Their age range was 23-44 years old.

The study was set up using the prototype described in the previous section. The backend application was used to visualize the application output on two external 60 inch HD displays, placed in a modern conference room environment. In keeping with the "architect" scenario, the study application allowed participants to move visual items (i.e., pictures of different stone patterns) between the two displays. The list view for **B2** was set up to maximize its effectiveness: we listed the two displays as the first two items on the list and the selectable items following those. Thus, the list had only a single selection level, and the number of items (12 stone patterns and two displays) was low enough such that the users could scroll the list effectively.

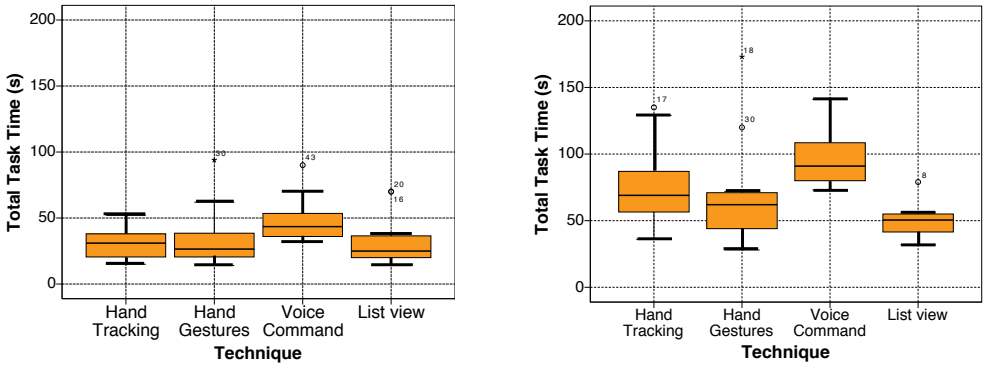
Procedure and Design

To simulate a typical interaction sequence of our scenario, participants were asked to perform two tasks using each interaction technique, thus following a within-subjects design. The two tasks were (1) move a visual item from display 1 to display 2, and (2) to move a visual item from display 1 to display 2 and then back to display 1. The reason for only asking for one task repetition was that the two task were set up in such a way such that each interaction technique was repeated several times per task (e.g., repeated selection of visual items during a task). This way, the users performed the same action per interaction technique multiple times.

We measured (total) task completion time and error count (i.e., the number of missed selections) per task as performance measured and also asked participants to answer a System Usability Scale (SUS) [4] questionnaire for each input technique.

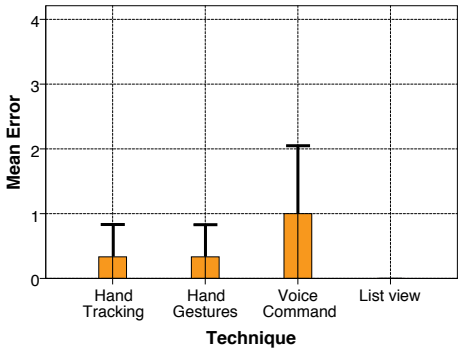
[9]We note here that with complete camera-hand-display calibration, which might be a complex task to achieve, a direct item selection based solely on pointing towards an item would be possible. We believe, however, that the setup we use in this paper is sufficient for the goals of our user study.

[10]`http://zeromq.org`

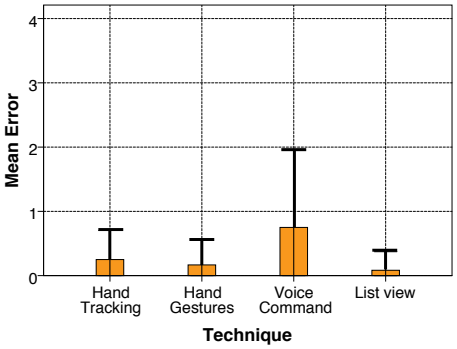


(a) Total Time Task (1)

(b) Total Time Task (2)



(c) Errors Task (1)



(d) Errors Task (2)

Figure 9. Quantitative results of our main user study. Note: the error bars on boxplots (a) and (b) show the 95 % confidence interval, whereas the bars on plots (c) and (d) show one standard deviation from the mean.

Before starting the main tasks, each participant was instructed to practice with the system for about 15 minutes until he or she became familiar with all the proposed input techniques.

Hypothesis

Our initial hypothesis (**H3**) was that **T2** and **T3** would have shorter task execution times and lower error counts than the baseline techniques **B1** and **B2**.

Findings and Discussion

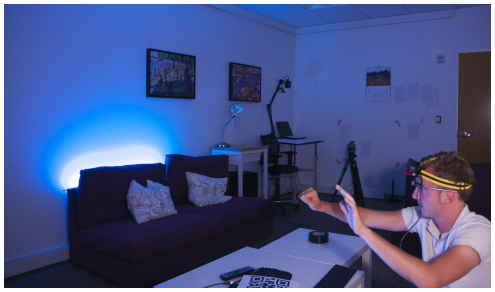
Figures 9(a) and 9(b) show the total task time for tasks (1) and (2). An ANOVA shows that there was a significant difference in the task completion times for task (2) ($F_{3,44} = 5.516,\ p = 0.003$). For task (2), list view (**B2**) was the technique with the lowest (average) total execution time of 49.92 s, followed by hand gestures (**T3**) at 69.33 s and hand tracking (**T2**) at 75.92 s. Voice commands (**B1**) had the highest total execution time at 95.92 s. However, a Bonferroni pairwise comparison showed only a significant difference between techniques **B2** and **B1**.

We conducted an ANOVA on the error count measure. The only observable statistically significant difference ($F_{3,44} = 5.359,\ p = 0.003$) was for task (1), where **B1** and **B2** differed significantly according to a Bonferroni post-hoc test (p = 0.002).

We obtained an average SUS score of 51 for **B1**, **B2** and **T3** as well as 49 for **T2**. Not surprisingly, an ANOVA showed that there was no statistical significant difference for the SUS score. We should note, however that the score for all techniques was rather low on the SUS scale (which ranges from 0 to 100). The somewhat diffuse SUS results suggest that this questionnaire might not be entirely suitable to evaluate the usability of the post-desktop interactions we discuss in this paper.

Our results do not validate our initial hypothesis **H3**. Although **T2** and **T3** did show a lower average execution time for tasks (1) and (2) than **B1**, **B2** had the lowest average task execution time. We should note, however, that there was no significant difference to **B2**. Although not provable with our statistical analysis, we can hypothesize that **T2** and **T3** performed more or less on par with **B2**. This is also reflected by the significantly higher error rate observed for **B1** (see Figures 9(c) and 9(d)). Although we chose a very simple vocabulary and syntax for the speech-based interface, and allowed for recognition errors in the speech commands, there were still a lot of errors stemming from misrecognized voice commands. Nevertheless, voice recognition on current devices like Google Glass is not instantaneous, as voice recognition is performed in the cloud, which causes a delay that may have contributed to the somewhat longer average task execution times, an effect which we have

(a) Interaction with 3D content on large display.



(b) Ambient light control through in-air gestures.

Figure 10. Two demonstration applications we implemented using our prototype.

considered to be an intrinsic property of voice-based interfaces for the purposes of this study.

We should also note that, as mentioned previously, the list view of **B2** was set up in a very effective way. We believe however, that the list view interface will not scale effectively if the number of selectable targets grows. Scrolling actions will get proportionally longer. Furthermore, the time to choose an appropriate item (i.e., a list item that corresponds to a desired selection item in the environment) in the list view will increase as the number of selectable items increases (Hick's Law [5]). Here, we believe that a gestural approach would be the most effective as users can directly select items they see rather than searching for a correspondence between the desired item on an external screen and an item on their HMD's display. Future work will need to asses under which conditions selection via hand gestures will become more effective than the current list view interface.

FURTHER SPATIAL INTERACTION APPLICATIONS

Apart from the application and scenario we have already discussed in this paper, there are numerous possibilities of leveraging the spacial input properties afforded by our prototype. With ego-centric tracking, users can perform spatial interactions with any type of connected device (i.e., reachable via TCP/IP or other communication channel) in the environment, without requiring further instrumentation of the device to track user inputs. In the following, we describe two demonstration applications we have developed based on our prototype.

As our prototype allows full 3D capture of the user's hand, we can, for example, implement user interfaces for 3D scene viewing on ubiquitous displays very easily (see the demo application in Figure 10(a)). A further advantage for ego-centric tracking here is that, e.g., for very large displays (e.g., in public spaces) supporting interaction by larger numbers of users there is no sensor scalability or coverage problem as each user is equipped with a personal sensing device.

Another application example for our prototype is controlling smart room appliances such as ambient lighting (see Figure 10(b)). In contrast to previous work in this domain [6], we believe that interacting directly through gestures can be more effective than using GUI-based controls on the HMD. Gestures, for example, allow more possibilities to fine tune certain parameters that are of interest when controlling ambient lighting, for instance, fine-tuning the lights' brightness and hue settings. Also, targeting the ambient devices can be accomplished by visual selection, i.e., turning the head towards the device, or direct pointing via a gesture.

CONCLUSION AND FUTURE WORK

In this work we presented a prototype scanario for HMD-based gestural interactions in a smart office environment. We described our implementation of a prototype system for use in this environment. The development of the prototype was informed by a preliminary study that elicited user emotions towards several proposed gestural input techniques and also provided user preference ranking feedback of HMD gestural input techniques vs. a set of baseline techniques. In a second study implementing the proposed scenario we quantitatively evaluated the usability our prototype.

The results of our first study indicate that gesture-based interaction techniques were accepted by the user both in preference, as shown by the ranking and Likert-scale results, and emotional feedback, where the input techniques using in-air hand gestures elicited a positive emotional response.

The results of our second study, however, are more difficult to interpret due to the unclear statistical results. However, the statistically significant results for audio do indicate that this modality (used by itself) is at a disadvantage vs. gestural input and the baseline GUI "list view" technique, both in terms of execution speed and error count. The comparison between gestures and the baseline technique does not show a clear winner. Nevertheless, we believe that the list view may at a certain point (e.g., at a certain level of available targets, or for more complex nested selection tasks) show disadvantages vs. gestural input. A reason for this is that using gestures, users potentially have a larger space to perform inputs, and furthermore, when not using a HMD that covers the user's entire field of view, using gestures in conjunction with the external display provides a much larger space for displaying information, which might also speed up selection tasks.

A future study that varies the number of targets on the display and that also implements nested selection tasks will be needed to answer detailed questions comparing the list view to gestural interactions. However, the knowledge gained in this study may be limited, as the insights would be applicable mostly to HMDs with limited display areas, such as the Google Glass device we used in this work. It is likely that the view area of future devices will be larger, and thus allow the implementation of more effective menu structures, such as drop-down menus or pie menus. Therefore, it would arguably be more interesting to conduct a future study with a slightly more capable HMD, such as the Epson Moverio Series of devices [8], which, in contrast to Google Glass, have a significantly wider viewing angle and a handheld large touch pad for interaction.

Lastly, we presented two demonstrator application that make use of our ego-centric tracking technique for HMDs. We believe that ego-centric tracking has advantages over exo-centric tracking as spaces do not need to be specially instrumented with sensors. Furthermore, the input capabilities of a space can scale with the number of users when using ego-centric tracking. However, one issue to address is how to detect and interface with external interactive devices in a seamless way. In this paper, we used visual codes as a helper to address devices and content. Using natural image features, BLE beacon technology, ultrasound, IR or visual light signaling, etc., could be used to address this problem in more elegant ways.

In the future, we are interested in more directly studying the difference between exo- and ego-centric tracking in terms of tracking performance, e.g., making a comparison of our prototype vs. a standard depth sensor such as the Kinect. We also wish to explore further applications for controlling ambient devices, and, correspondingly, investigate what further gestures may be useful given those applications.

REFERENCES

1. Agarwal, A., and Meyer, A. Beyond usability: evaluating emotional response as an integral part of the user experience. In *CHI 2009 Extended Abstracts*, ACM (2009), 2919–2930.
2. Bailly, G., Müller, J., Rohs, M., Wigdor, D., and Kratz, S. Shoesense: a new perspective on gestural interaction and wearable applications. In *Proc. CHI 2012*, ACM (2012), 1239–1248.
3. Boring, S., Baur, D., Butz, A., Gustafson, S., and Baudisch, P. Touch projector: mobile interaction through video. In *Proc. CHI 2010*, ACM (2010), 2287–2296.
4. Brooke, J. Sus-a quick and dirty usability scale. *Usability evaluation in industry 189* (1996), 194.
5. Card, S. K., Newell, A., and Moran, T. P. The psychology of human-computer interaction.
6. Chen, Y.-H., Zhang, B., Tuna, C., Li, Y., Lee, E. A., and Hartmann, B. A context menu for the real world: Controlling physical appliances through head-worn infrared targeting. Tech. rep., DTIC Document, 2013.
7. Colaço, A., Kirmani, A., Yang, H. S., Gong, N.-W., Schmandt, C., and Goyal, V. K. Mime: compact, low power 3d gesture sensing for interaction with head mounted displays. In *Proc. UIST 2013*, ACM (2013), 227–236.
8. Epson. Epson Moverio BT-200. `http://ow.ly/Ag2Y4`, Aug. 2014.
9. Forlines, C., Wigdor, D., Shen, C., and Balakrishnan, R. Direct-touch vs. mouse input for tabletop displays. In *Proc. CHI 2007*, ACM (2007), 647–656.
10. Google. Google glass. `http://www.google.com/glass/start/`, July 2014.
11. Jacob, R. J., Girouard, A., Hirshfield, L. M., Horn, M. S., Shaer, O., Solovey, E. T., and Zigelbaum, J. Reality-based interaction: a framework for post-wimp interfaces. In *Proc. CHI 2008*, ACM (2008), 201–210.
12. Johanson, B., Hutchins, G., Winograd, T., and Stone, M. Pointright: experience with flexible input redirection in interactive workspaces. In *Proc. UIST 2002*, ACM (2002), 227–234.
13. Malik, S., Ranjan, A., and Balakrishnan, R. Interacting with large displays from a distance with vision-tracked multi-finger gestural input. In *Proc. UIST 2005*, ACM (2005), 43–52.
14. Mistry, P., Maes, P., and Chang, L. WUW-wear Ur world: a wearable gestural interface. In *CHI Extended Abstracts 2009*, ACM (2009), 4111–4116.
15. Piumsomboon, T., Clark, A., Billinghurst, M., and Cockburn, A. User-defined gestures for augmented reality. In *Human-Computer Interaction–INTERACT 2013*. Springer, 2013, 282–299.
16. Rekimoto, J. Pick-and-drop: a direct manipulation technique for multiple computer environments. In *Proc. UIST 1997*, ACM (1997), 31–39.
17. Robertson, G., Czerwinski, M., Baudisch, P., Meyers, B., Robbins, D., Smith, G., and Tan, D. The large-display user experience. *Computer Graphics and Applications, IEEE 25*, 4 (2005), 44–51.
18. Wigdor, D., and Wixon, D. *Brave NUI world: designing natural user interfaces for touch and gesture*. Morgan Kaufmann, 2011.
19. Wilson, A. D., and Benko, H. Combining multiple depth cameras and projectors for interactions on, above and between surfaces. In *Proceedings of the 23nd annual ACM symposium on User interface software and technology*, ACM (2010), 273–282.