A User Study on Augmented Virtuality Using Depth Sensing Cameras for Near-Range Awareness in Immersive VR

Ghassem Alaee*
Department of Computer Science
Memorial University of Newfoundland

Amit P. Desai†
Department of Computer Science
Memorial University of Newfoundland

Lourdes Peña-Castillo‡
Department of Computer Science
Department of Biology
Memorial University of Newfoundland

Edward Brown§
Department of Computer Science
Memorial University of Newfoundland

Oscar Meruvia-Pastor¶
Department of Computer Science
Memorial University of Newfoundland

Abstract

A common drawback of fully immersive Virtual Reality (VR) Head-Mounted Displays (HMDs) is the visual and in some cases aural isolation of the user. This isolation, which comes as a result of the goal to provide a better VR experience, makes it extremely difficult for users to look at and interact with their physical environment, reducing the convenience of such systems. In particular, smartphone access within immersive VR displays is severely limited by the fact that when users need to make use of their smartphone, they usually have to remove their HMDs. One way of dealing with the issue of visual isolation is to move towards an Augmented Virtuality (AV) design. Typically, AV incorporates part of the physical world into the virtual world with the assistance of one or more cameras. In our case, we use a portable RGBD camera mounted on an HMD to provide near-range awareness, facilitating access to the users' personal mobile device within the field of view (FOV) of the scene inside the VR. We call this solution Near-Range Augmented Virtuality (NRAV). To validate NRAV, we compare it, through a user study, against SDSC, a Smartphone Detector based on a Statistical Classifier developed by Desai et al. The study covers a variety of tasks associated with the daily operation of a smartphone within a VR context, including reading and typing messages, answering phone calls, and navigating within the VR environment. Results of the user study with 25 participants show that NRAV allows users to perform the tasks presented and is preferred by a majority of users in most of the cases.

Index Terms: Head-Mounted Displays—Augmented Virtuality—Immersive Virtual Reality—Mixed and Augmented Reality; Depth Sensing and RGBD Cameras—Near-Range Awareness—Smartphones and Mobile Devices in VR—; Human-Computer Interaction—User Studies—User Experience Evaluation—;

1 Introduction

Affordable Head Mounted Displays (HMDs) such as the Oculus Rift [18], HTC Vive [10], and similar technologies [19], have extended the accessibility of immersive Virtual Reality (VR) to a wide segment of the population [12] and the research community [7, 13, 20]. VR HMDs provide a rich, interactive and immersive VR environment by offering low-cost, high-resolution graphics, stereoscopic imagery, and the possibility to look around in the virtual environment (VE). Since VR HMDs are designed to provide an immersive environment where users feel they are part of the virtual environment, a natural consequence is that users of these systems cannot see (visual isolation) and sometimes cannot hear (aural isolation) their actual physical environment, which might be unsafe of otherwise inconvenient [2, 14]: simple tasks such as reaching out for a mouse or looking at an incoming message on the phone, become impossible unless the users remove their HMDs.

Augmented Reality (AR) and Augmented Virtuality (AV) are parts of the Mixed Reality spectrum that deal with the combination of real-world and computer-generated visuals [15]. In this research, we are interested in exploring solutions to facilitate access to a user's personal mobile phone (smartphone) while immersed in VR using AV. Mobile phones are an important conduct for people to communicate with each other and keep track of important events and social media. According to recent statistics, there are currently 5 billion mobile users, and more than 3 billion people are active social media users.

To address the temporary isolation that takes place while wearing a VR HMD, we present a method that enables users to interact with their smartphone devices so that they can read messages, place calls, and use other apps, while still being immersed in the VR environment. Through Augmented Virtuality, additional devices such as an RGBD camera can be used as a window to a user's smartphone. In this research, we use Near-Range Augmented Virtuality (NRAV), which provides users of fully immersive VR HMDs with awareness of items in their close proximity using an Intel RealSense RGBD sensing camera mounted on top of an Oculus Rift HMD. With this implementation, users do not need to remove their HMD's to use their smartphone while immersed in VR.

To find out the extent to which this solution allows a user to operate a smartphone while immersed in VR, and the degree of user acceptance of this solution, we designed a study where 25 participants were evaluated in their ability to do some basic tasks involving smartphones under two different AV implementations, the one we presented here, and the one implemented by Desai et al. [6].

The rest of this article is structured as follows: In the next Section, we discuss related work. In Section 3 we describe the system setup and implementation. In Section 4 we explain the methodology used to validate the implementation. In Section 5 we present the results of the user study, in Section 6 we discuss our findings, and in Section 7 we summarize the main conclusions of this research and describe future work.

2 Related Work

To address the issue of visual isolation, researchers have proposed to import a portion of reality into the virtual environment (VE) using either fixed cameras in the environment or by virtualizing a part of the reality into the VE using mobile cameras. Examples of the

*e-mail: galaee *at* mun *dot* ca
†e-mail: apd525 *at* mun *dot* ca
‡e-mail: lourdes *at* mun *dot* ca
§e-mail:brown *at* mun *dot* ca
¶e-mail:oscar *at* mun *dot* ca

first approach include the work by Nahon et al. [16, 17], whereas examples of the second approach include the works of Bruder et al. [4], Tecchia, et al. [21] and Desai et al. [6], among others [11]. In all of the cases above, at least one camera was used to capture part or all of the user's body or the environment and some portions of the physical environment were imported into the VE. Some of the early implementations of Augmented Virtuality were referred to as "see-through video" and implemented by mounting cameras on HMDs [3]. Bruder et al. used two separate cameras, an IR camera, and a USB camera, to capture the user's body and incorporate it into a VE using a skin and body classifier. They used the resulting environment to test whether the users liked their augmented presence in the VE and the results showed a higher sense of presence. Tecchia et al. used an RGBD camera mounted on the HMD to create the same effect by having a texturized geometric mesh of the hands and body rendered as a 3D model within the VR. They also argue that by having an egocentric view of the VR and augmenting the user's body into it, the user would have a better virtual experience. As opposed to the previous two systems, Nahon et al. proposed a method to not only support a 1st. person point of view of the environment, but also a 3rd. person view using multiple Kinects mounted in a room, and claimed that the resulting environment would solve some of the safety issues such as hitting something or falling. The focus of Desai [6] was to address the visual isolation by augmenting the VE with an image of the smartphone. In that work, the Leap MotionSensor and Oculus Rift DK2 were combined. Using a Smartphone Detector based on a Statistical Classifier (SDSC, for brevity), the approach proposed a solution specifically designed to detect and track a smartphone that is being held within a certain range and orientation in front of the Leap Motion controller. This resulted in a real-time system capable of detecting a smartphone with 90% accuracy when it is held in front of the Leap Motion controller. An Android app was used to send a stream of screenshots and orientation data to display the tracked smartphone within the VR application [6].

Recent solutions have been explored where certain physical objects can be augmented into the VR using built-in trackers. However, those solutions are limited in that they are restricted to particular objects, such as the keyboard,or game props, and do not provide awareness of other items in the near-range of the user, such as smartphones, people, or hazards.

All approaches above deliver a certain level of awareness about the user's body, a part of the body, or the user's environment. Previous work have shown that having an egocentric AV environment increases awareness of the users and that they feel more present in the environment [4, 21]. To the best of our knowledge, with the exception of the work by Desai et al., no research has focused on the specific question of using a smartphone within the VR. We used this view as an opportunity to implement an alternative to such an AV environment, and to compare this alternative to the approach of Desai [6], shown on the right panel of Figure 2. In this research, we examined each of these approaches to see how operable they would be in everyday life and how much users would like them. More details about the implementations and the tools used in this research will be discussed in the next section.

3 Implementation

NRAV uses a combination of a video see-through HMD prototype with depth-based video segmentation to allow for near-range segmentation. To implement depth-based video segmentation from a first-person perspective, we mounted an Intel RealSense RGBD Camera on an Oculus Rift HMD, as shown in Figure 1. Using the depth field of the camera and its SDK, we selected only the objects within a certain distance (from 10 to 40 cm) from the camera and augmented the VE with the segmented video stream, thus ensuring that only near-range objects are embedded within the VR. The RGBD camera has a limited field of view (FOV); thus, in order for users to see their smartphones in VR, they need to hold the smartphone within the FOV of the camera, i.e., in front of the HMD. In this way, the operator's hands and the smartphone become visible when the phone is held in front of the user's face, eliminating other items that might be further away. This approach is similar to that of [6], but more general, in that it doesn't track specifically for smartphones, but any object within the proximity of the camera. In Figure 2 we show an image of how the user's hand and smartphone appear within the VR environment.

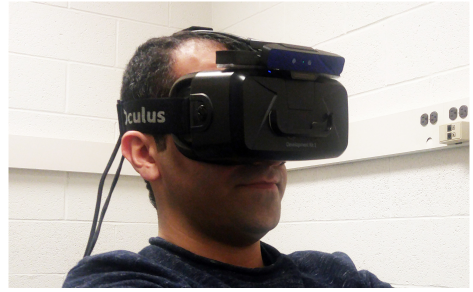

Figure 1: User wearing a RealSense RGBD camera on an Oculus HMD.

The following components are used in this implementation: a high-performance PC with an NVIDIA GTX 970 GPU, an Oculus Rift with an Intel RealSense camera mounted on top. The Oculus DK2 provides a 1920x1080 (960x1080 per eye) resolution and a maximum refresh rate of 75Hz. The Oculus Legacy Runtime for Windows 0.8.0.0-beta is the only required software package needed for integration of the Oculus Rift with Unity. Intel's RealSense Developer Kit SR300, was used in this work [5]. It is a depth-sensing camera that can be used for close-range depth perception. This version of the camera requires an Intel 6th generation (i7 6700 or newer) processor and would only work on a machine with a USB 3.0 port and Microsoft Windows 10 as the operating system. Its highest color resolution is 1920x1080 at 30 frames per second (fps) and its highest depth resolution is 640x480 with an optimal distance between 20 cm to 1.5 meters for best depth perception. For dynamic background segmentation, which comes as a module in its SDK, this resolution is decreased to 1280x720, which reduces legibility. The SDK used at the time of implementation was version 2016 R3. The smartphone device used was a Nexus 5 with 2 GB of RAM, Qualcomm Snapdragon 800 Quad-core processor, 4.95 inches of display and 1080 x 1920 pixels (445 PPI pixel density) screen resolution.

3.1 Software

There are three main steps to build an AV environment with the components mentioned above:

- Initialization: The RealSense SDK provides a C# interface for Unity. The two required namespaces in the RealSense SDK are 'Intel.RealSense' and 'Intel.RealSense.Segmentation'. In this step, we initialize the interfaces and the GameObject that shows the segmented image. Since the input image has a lower resolution than the maximum available resolution, reconfiguring the StreamProfile object to a higher resolution is necessary. The same process is repeated for the depth image.

- Segmentation: In segmentation, we select the objects that are close to the camera from the captured frames. This process is repeated for every single frame that is being sent from the camera to the previously initialized object in the start() function. The segmentation module comes with an event subscription mechanism that a function/method subscribes to, an event

Figure 2: Near-Range Augmented Virtuality: On the left, a user holding a phone in front of the HMD. In the middle, an image of the users's hand and phone as it appears within the default virtual environment. On the right, Augmented Virtuality as it appears when using SDSC.

called 'FrameProcessed'. The subscriber/observer gets an instance of the image for each frame and then within the function, we segment the image in a way that only the foreground object being selected is shown.

- Integration: The application displayed on the HMD is implemented through a Unity program that displays a segmented image as a GameObject along other GameObjects in the scene. For educational purposes, a simple version of this implementation is provided as open source code in a GitHub repository [1]. With this implementation, the user is able to see the augmented smartphone and the operator's hands. For optimal results, the smartphone should be held within the FOV of the camera, roughly at 30 cm distance in front of the HMD. The result is an egocentric view of the smartphone within the VE, as shown in Figure 3.

To study the user's capacity to operate a smartphone within a VR context, we chose to implement VR navigation using the smartphone. The VR navigation application was designed to communicate with a Unity project with an Android app. The Android App was developed to capture user input through the touchscreen as a virtual keypad, with four arrows indicating the desired direction of motion (forward &backward) and panning (left & right), as shown in Figure 3.1. The Android App also acts as a TCP/IP server and transfers the user's commands to the VR application. For the purposes of demonstrating this research, the VR environment where the user is immersed is a publically available model of a Viking village, provided by Unity. Figure 3 shows NRAV with the Viking village in the background, and the accompanying video (available at [1]) shows the look and feel of both NRAV and Desai's SDSC, and further describes other aspects of the study.

4 Methodology

To find out whether the proposed implementation allows a user to operate the smartphone for a variety of tasks within the VR, we performed a user study. Three conditions were tested in the study: the implementation proposed here, the implementation proposed by Desai et al, and the option of simply removing the HMDs, which we used as a baseline. The baseline condition corresponds to what users would normally have to do in the absence of any option to use the smartphone within the VR. These three conditions will be identified with the following names in the rest of this article: a) Removing the headset as "baseline", b) using Near-Range Augmented Virtuality (NRAV), and c) using Smartphone Detection based on a Statistical Classifier (SDSC) [6]. In the last two conditions, the user is not going to remove the HMD and instead will interact with the app using the smartphone, therefore, an android app was designed to send directions to the Unity VR application (see Figure 3.1). The Unity application needs the IP address and the port number from which the smartphone is sending its messages. To avoid transmission delays, both the smartphone and the machine running the Unity project should be using the same network.

Foley et al. identified six fundamental interaction types which are independent of application and hardware: Select (choosing between alternatives), Position (indicating a position on the display), Orient (rotating a symbol on the screen), Path (combination of Position and Orient), Quantify (entering a numeric value), Text (entering a text string) [8]. A high-level task like answering a phone call or answering a text message may involve the combination of two or more of these basic interactions. To test the operability of two implementations, 5 high-level tasks were selected. These high-level tasks are representative of the most likely tasks that a user may perform during a VR experience. We assume the users will be able to perform all other similar high-level tasks if they can successfully execute these 5 tasks. Here are the descriptions of the 5 high-level tasks included in our experiment:

- Task 1: Answering a phone call. This included swiping up the screen when it is ringing, answering, and putting the call-in-progress (or on speaker mode) by tapping on the speaker icon. The latter step is added to this task to make it easier for the participant to hear the other side of the conversation while wearing an HMD. The participant starts the task by hearing the ringtone and ends the task when the call is on speaker mode. This task includes two basic interaction modalities: Select and Position.

- Task 2: Using the calculator. This involves doing some calculations with the built-in calculator app in the smartphone. The participants start the task by unlocking the phone (swiping up the screen) and launching the calculator app. Before starting the task, the participants know what calculation they are expected to do, and during each attempt, they can request the calculation to be read to them whenever they want. The task ends with the participant reading out the result. This task includes three basic interaction forms: Position, Select, and Text (or Quantify).

- Task 3: Reading a text message. There are different ways to read an unread text message on an Android phone. The participants were asked to read the text message only by launching the messages app through the list of applications and not by using the notification bar. The task starts with the text tone, then the participant unlocks the phone and launches the messages app and ends with the participant reading out the text message. This task includes the Position and Select basic interaction forms.

- Task 4: Placing a call. In this task, for every attempt, the participant gets a name and should call that person by launching the contacts app. The task starts with participant unlocking the phone and ends with the ringtone of the recipient's phone. This task also includes two interaction types: Position and Select.

- Task 5: Navigating through a VR environment. In this task, the participants start the task from a starting point in the Viking

Figure 3: Dynamic background segmentation in NRAV integrated within Unity's Viking village, used as the VR environment for the user study. On the righ: User Interface of the Android App for navigation within the VR environment.

village, (described previously) and end the task by reaching another point in the village. In baseline, the participants use a keyboard to navigate, but in conditions NRAV and SDSC, they use a custom Android app. At the beginning of the task, the app is launched and ready to be used by the participant. This task includes the Position and Select interaction forms. To simulate a gaming condition, we used the Viking village, a free sample 3D environment published by Unity technologies that requires Unity 5.0 or newer. Figure 3 shows a screenshot of this environment in the background.

We performed a pilot study to evaluate the tasks presented to the users. Initially, we had included a task where users were asked to send a message using the built-in keyboard of the mobile phone. We found that users were unable to complete this task, as they struggled with the small buttons on the phone display. We believe that, at this time, neither of the evaluated methods is suitable for accomplishing this task, and plan to return to this challenging question in the future.

4.1 Demographics

Participants were all undergraduate and graduate level students, mainly from the Computer Science and Electrical Engineering Departments. Initially 26 participants were recruited, however 1 participant was not able to complete the experiment due to motion sickness and this data was removed from the analysis.

4.2 Data Collection

We collected the following data in this experiment: 1. Questionnaires: Two questionnaires were involved: a pre-questionnaire and a post-questionnaire. The pre-questionnaire collected personal information about the participants including the amount of time they spend on social media and different types of games on a weekly basis. The post-questionnaire was answered at the end of the experiment before the interview. The post-questionnaire collected the participant's preferences for the conditions in different tasks. 2. Timing measurements and pass/fail rates: All of the sessions were videotaped for measuring how much time the participant spent on each of the task attempts. At the same time, successful and unsuccessful attempts of the participants were recorded on result forms. 3. Interview: As a part of debriefing, we asked the participants about their performance and how they felt about each of the tasks and the conditions, the reasons behind their questionnaire answers and about the possible ways to improve the conditions.

5 Results

5.1 Completion Time Measurements

The mean time for each task is shown in Figure 4. To achieve data consistency, each participant was asked to repeat each task 3 times for each condition.

The followings observations are made based on the analysis of mean task completion times:

- For all tasks, ANOVA results indicate that the most significant factor (pvalue $<$ 2e-16) to explain the variance in completion times is the condition.

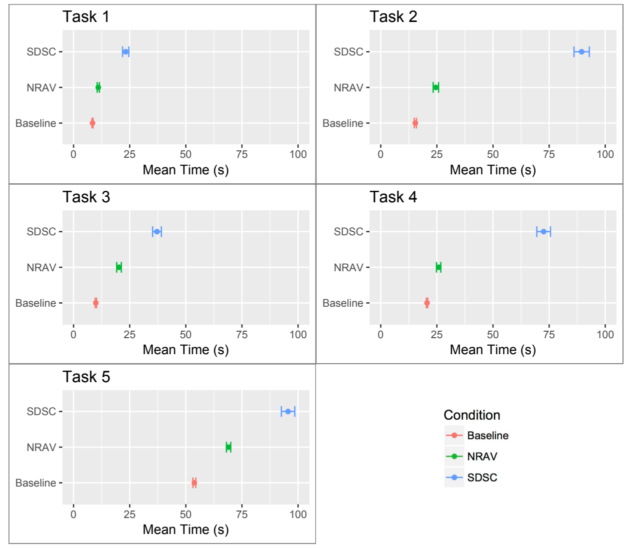

Figure 4: Mean completion times and error bars per task. Task 1: Answering a phone call; Task 2: Using the calculator; Task 3: Reading a text message; Task 4: Placing a phone call; Task 5: Navigating through the Viking village.

- For Tasks 1 (answering a call) and 4 (placing a call), the mean time of SDSC is statistically significantly different than the mean time of baseline and NRAV (pvalue $<$ 10e-7). That is, NRAV allows users to complete these tasks faster than SDSC. For both of these tasks, there were even some participants who used NRAV and were able to finish the tasks sooner than the baseline.

- For Tasks 2 (using the calculator), 3 (reading a text message) and 5 (navigation in VR): The mean completion time of the users using SDSC is (statistically) significantly longer than their mean completion time using NRAV and the baseline, and the mean completion time of NRAV is also statistically significantly longer than the mean completion time of the baseline (pvalue $<$ 0.001).

We hypothesize that the reason that NRAV takes longer than the baseline might have to do with two factors: the loss in visual acuity that takes place from having the video camera and the HMD as intermediaries between the real world and the users' visual system and the potential adaptation effects from the sensory rearrangement of both NRAV and SDSC, as noted in previous literature [3].

To analyze the distribution of user completion times for all conditions we produced violin plots [9] of the completion times (see Figure 5). Based on the analysis of these plots, the followings observations were made:

- Across all tasks, users' completion time are more consistent (i.e., have less variance) under the baseline condition and fol-

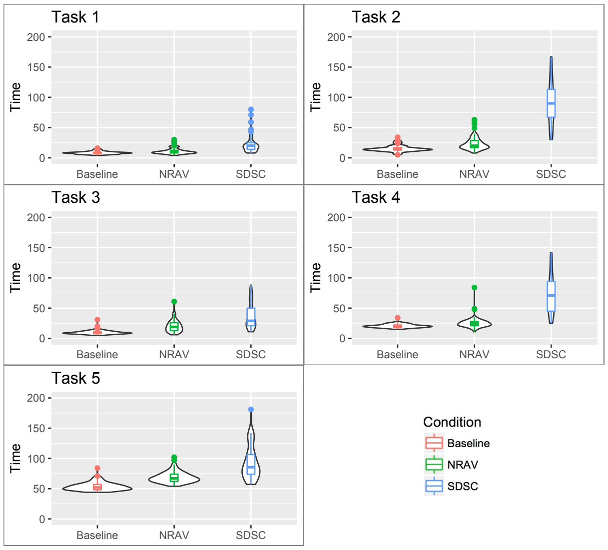

Figure 5: Violin plots based on the average completion times per task. Task 1: Answering a phone call; Task 2: Using the calculator; Task 3: Reading a text message; Task 4: Placing a phone call; Task 5: Navigating through the Viking village

low a Gaussian distribution. NRAV also shows a normal distribution with few outliers, and there are less outliers in NRAV than in SDSC.

- For Task2 (using the calculator) and Task 4 (placing a call) SDSC shows the highest spread in completion times, meaning some users did the tasks quickly while others took too long.

- Across all other tasks, SDSC also shows a higher variability in user response, and an increased number of outliers when compared to the two other conditions.

In addition, ANOVA results indicate an interaction between condition and attempt number for the VR navigation task (Task 5), shown in Supplementary Figure 1 [1]. From the statistical analysis of the mean completion times, we observed that the mean time of attempt 3 is significantly different than the mean time of attempt 1 (p-value = 0.013): users were faster in attempt 3 than in attempt 1, suggesting that, as expected, there is a learning effect that comes from practice across all conditions, but which takes a little longer in the case of SDSC.

5.2 Participants' preferences (Post-Questionnaire)

As a part of the post-questionnaire, participants were asked to choose the best and the worst conditions based on five different factors: ease of use, ease of learning, no frustration, fun, and speed of use. The only statistical difference between the baseline and NRAV was the fun factor. 75% of the participants chose NRAV as the most fun condition out of the three conditions. In addition, SDSC was considered the worst condition for all factors. Supplementary Figure 2 shows the results of these questions.

In the last part of the post-questionnaire, participants were asked to choose their desired condition for each of the tasks. They were asked specifically that if they were playing a game and they had to choose one of the 3 conditions to do a certain task, which one would they choose? Supplementary Figure 3 summarizes the responses to these questions.

The NRAV was preferred over the other two conditions in 3 out of 5 tasks. In Task 2 (calculations), baseline and NRAV each with 50% were equally preferred. The baseline was preferred by a slight majority only for Task 3 (reading an sms). Task 3 is the only task where NRAV was voted with less than 50% preference. For this same task, SDSC with 17% preference reached its peak of preference among all tasks, indicating that the users found some advantage of SDSC for reading. We hypothesize this is because SDSC renders frames grabbed directly from the phone display into the VR application, producing sharper images of the phone display within the VR.

5.3 Interview

One of the questions that were asked of the participants during the interview was to choose the easiest and the hardest tasks considering only conditions NRAV and SDSC. Supplementary Figure 4 shows that Task 1 (answering a call) was perceived as the easiest task and Task 2 (using the calculator) was perceived as the hardest task. Interestingly, Task 5, navigating through the Viking village was found the second easiest task, although it was actually the longest task. We believe that this is the case because even when navigating might have taken the most time, it involved a simpler interface and less fundamental interaction types than using the calculator.

6 Discussion

Task 1 was about answering a phone call and involved only one swipe up and one tap on the speaker icon on the screen. Task 5 involved the use of only four buttons on the phone's virtual keypad that the participants had to choose from to navigate through the Viking village. On the other hand, Task 2 was about doing some calculations, and it involved different digits and operators to be selected to complete the task. According to the instructions given to the participants, in only tasks 2, 3 and 4, the participants had to pick up the smartphone and launch the appropriate app from the list of applications, however, in tasks 1 and 5, there was no need to launch any apps. This can be one of the reasons that caused more work for the participants, especially in SDSC where the participants spent more time than in the other two conditions. During the interview, participants were asked about the reasons why they were unhappy about the augmented phone in the resulting environments. Based on the conversations and our informal evaluation, the following observations emerge:

- There are some major factors that might explain why users did not prefer SDSC. First, the participants could not quickly register their fingers with what the phone image within the VR. Second, the delay on what participants could see as the virtualized smartphone on the screen was one of the reasons that caused more human errors and eventually spending more time on that condition. On the other hand, the better legibility of the smartphone screen was a positive aspect of SDSC.

- In the case where participants did not prefer the NRAV, some participants reported that they could not read the texts clearly, especially on Task 3, and some of them did not like to see small parts of background environment on the edges of the augmented smartphone (due to imperfect segmentation).

- The blurriness of the texts in NRAV and the difficulty of selecting and launching the apps in SDSC was noted as one of the reasons that some of the participants would prefer to remove the headset on some of the tasks. As noted previously, the loss in visual acuity that takes place from having the video camera and the HMD as intermediaries between the real world and the users' visual system might have been an issue, in particular for reading text.

- One of the things that participants had difficulty with was the smartphone holding posture required during task 5, the navigation task. They did not consider it as intuitive as using the keyboard or other gaming controllers because they had to hold the phone in a sweet spot in front of them, and that was uncomfortable for them if they had to continue holding it in that position for a long period of time. This is understandable because people do not usually hold their phone in front of their face when looking towards the front and instead look down or lower their gaze temporarily.

- The context of the tasks is one of the important factors on how the participants would react to a condition. Users are not willing to perform all kinds of tasks in the middle of the game. It depends on the game and the importance of the tasks, and if a task is going to take twice as much time and energy as it normally should, then removing the headset, would not be considered a problem by the users.

7 Conclusions

This research characterizes the effectiveness of users trying to operate their smartphones while wearing an HMD. We compare two conditions: NRAV, which performs near-range augmented virtuality using depth-based video segmentation, and SDSC, which performs smartphone detection and tracking using a statistical classifier, against each other and against the baseline condition, where users need to remove their HMDs to operate the smartphones.

The main conclusions of this research are:

- Using NRAV, it is possible to operate the smartphone within the VR. NRAV allows users to perform several tasks on the smartphone as effectively as in the baseline condition while wearing an HMD.
- There is a time penalty when using NRAV compared to the baseline. However, users find that using NRAV is more fun than the baseline.
- Users performed all the presented tasks significantly faster when using NRAV rather than SDSC.
- The smartphone can be used as a touch keypad controller for VR Navigation using either NRAV or SDSC.

While some users still prefer to remove their HMDs to interact with the smartphone, user acceptance might increase with technological improvements in the capture resolution of RGBD cameras and display resolution of the HMD's. In fact, text legibility, as well as providing stereoscopic see-through video in HMDs to improve user awareness of their physical environment, have both been identified as a major target by leading HMD developers. As resolutions of displays and RGBD cameras increase, this approach is likely to have better performance and might even support typing on the mobile's touchscreen keyboards. This work also highlights the viability of using the smartphone touchscreen and display as an input device in immersive VR environments. Further work in this area involves exploring additional functionality that comes from the combination of smartphones with immersive VR systems, enabling users to perform more complex input tasks (such as typing, measuring, and sending messages with the phone), finding ways to reduce the gap of mean completion times with respect to the baseline, and characterizing in more detail the factors that prevent user acceptance of these technologies to improve user experience.

References

[1] G. Alaee, A. P. Desai, E. Brown, L. Pena-Castillo, and O. Meruvia-Pastor. A user study on augmented virtuality using depth sensing cameras for near-range awareness in immersive vr: Supplementary video, figures and code. `http://www.cs.mun.ca/~omeruvia/research/research.html`, 2018. Accessed: 2018-02-20.

[2] M. Billinghurst and H. Kato. Collaborative mixed reality. In *Proceedings of the International Symposium on Mixed Reality*, pp. 261–284. Yokohama, Japan, March 1999.

[3] F. A. Biocca and J. Rolland. Virtual eyes can rearrange your body: Adaptation to visual displacement in see-through, head-mounted displays. *Presence: Teleoperators and Virtual Environments*, 7(3):262–277, 1998. doi: 10.1162/105474698565703

[4] G. Bruder, F. Steinicke, K. Rothaus, and K. Hinrichs. Enhancing presence in head-mounted display environments by visual body feedback using head-mounted cameras. In *2009 International Conference on CyberWorlds*, pp. 43–50, Sept 2009. doi: 10.1109/CW.2009.39

[5] I. Corp. Intel realsense sdk 2016 r3 documentation. `https://software.intel.com/sites/landingpage/realsense/camera-sdk/v2016r3/documentation/html/index.html`, 2018. Accessed: 2018-02.

[6] A. P. Desai, L. Pena-Castillo, and O. Meruvia-Pastor. A window to your smartphone: Exploring interaction and communication in immersive vr with augmented virtuality. In *2017 Computer And Robot Vision (CRV) Conf. Proc.* Edmonton, AB, March 2015.

[7] C. Donalek, S. G. Djorgovski, A. Cioc, A. Wang, J. Zhang, E. Lawler, S. Yeh, A. Mahabal, M. Graham, A. Drake, S. Davidoff, J. S. Norris, and G. Longo. Immersive and collaborative data visualization using virtual reality platforms. In *Big Data (Big Data), 2014 IEEE International Conference on*, pp. 609–614, Oct 2014. doi: 10.1109/BigData.2014.7004282

[8] J. D. Foley, V. L. Wallace, and P. Chan. The human factors of computer graphics interaction techniques. *IEEE Computer Graphics and Applications*, 4(11):13–48, Nov 1984. doi: 10.1109/MCG.1984.6429355

[9] J. L. Hintze and R. D. Nelson. Violin plots: A box plot-density trace synergism. *The American Statistician*, 52(2):181–184, 1998. doi: 10.1080/00031305.1998.10480559

[10] HTC. HTC Vive. `http://www.htcvive.com/`, 2016. Accessed: 2016-05-01.

[11] D. Korsgaard, N. C. Nilsson, and T. Bjrner. Immersive eating: evaluating the use of head-mounted displays for mixed reality meal sessions. In *2017 IEEE 3rd Workshop on Everyday Virtual Reality (WEVR)*, pp. 1–4, March 2017. doi: 10.1109/WEVR.2017.7957709

[12] D. Kushner. Virtual reality's moment. *IEEE Spectrum*, 51(1):34–37, January 2014. doi: 10.1109/MSPEC.2014.6701429

[13] A. S. Mathur. Low cost virtual reality for medical training. In *2015 IEEE Virtual Reality (VR)*, pp. 345–346, March 2015. doi: 10.1109/VR.2015.7223437

[14] M. McGill, D. Boland, R. Murray-Smith, and S. Brewster. A dose of reality: Overcoming usability challenges in VR head-mounted displays. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI 2015, pp. 2143–2152. ACM, New York, NY, USA, 2015. doi: 10.1145/2702123.2702382

[15] P. Milgram and H. Colquhoun. A taxonomy of real and virtual world display integration. *Mixed reality: Merging real and virtual worlds*, 1:1–26, 1999.

[16] D. Nahon. Never blind in vr. In *ACM SIGGRAPH 2015 Computer Animation Festival*, SIGGRAPH '15, pp. 194–194. ACM, New York, NY, USA, 2015. doi: 10.1145/2745234.2810208

[17] D. Nahon, G. Subileau, and B. Capel. Never blind vr - enhancing the virtual reality headset experience with augmented virtuality. In *2015 IEEE Virtual Reality (VR)*, pp. 347–348, March 2015. doi: 10.1109/VR.2015.7223438

[18] Oculus. Oculus Rift. `https://www.oculus.com/en-us/rift/`, 2016. Accessed: 2016-05-01.

[19] Samsung Electronics Co. Ltd. Samsung Gear VR. `http://www.samsung.com/global/galaxy/gear-vr/`, 2016. Accessed: 2017-01-21.

[20] L. K. Simone, M. T. Schultheis, J. Rebimbas, and S. R. Millis. Head-mounted displays for clinical virtual reality applications: pitfalls in understanding user behavior while using technology. *Cyberpsychol Behav*, 9(5):591–602, Oct 2006. doi: 10.1089/cpb.2006.9.591

[21] F. Tecchia, G. Avveduto, R. Brondi, M. Carrozzino, M. Bergamasco, and L. Alem. I'm in vr!: Using your own hands in a fully immersive mr system. In *Proceedings of the 20th ACM Symposium on Virtual Reality Software and Technology*, VRST '14, pp. 73–76. ACM, New York, NY, USA, 2014. doi: 10.1145/2671015.2671123