Virtual Reality Meets Smartwatch – Intuitive, Natural, and Multi-Modal Interaction

Franca Alexandra Rupprecht
Computer Graphics & HCI
University of Kaiserslautern
Kaiserslautern, Germany
rupprecht@cs.uni-kl.de

Andreas Schneider
Computer Graphics & HCI
University of Kaiserslautern
Kaiserslautern, Germany
andreas.schneider@cs.uni-kl.de

Achim Ebert
Computer Graphics & HCI
University of Kaiserslautern
Kaiserslautern, Germany
ebert@cs.uni-kl.de

Bernd Hamann
Department of Computer Science
University of California
Davis, CA 95616, USA
hamann@cs.ucdavis.edu

CHI'17 Extended Abstracts, May 6–11, 2017, Denver, CO, USA.
ACM ISBN 978-1-4503-4656-6/17/05.
http://dx.doi.org/10.1145/3027063.3053194

Abstract

Despite the fact of increasing popularity virtual environments still lack useful and natural interaction techniques. We present a multi-modal interaction interface, designed for smartwatches and smartphones for fully immersive environments. Our approach enhances the efficiency of interaction in virtual worlds in a natural and intuitive way. We have designed and implemented methods for handling seven gestures and compare our approach with common VR input technology, namely body tracking using a 3D camera. The findings suggest our approach to be very encouraging for further developments.

Author Keywords

Intuitive and natural interaction; Low budget interaction devices; Mobile devices; Virtual Reality; Body movement gestures; Gesture recognition

ACM Classification Keywords

H.1.2 [User/Machine Systems]: Human factors; H.5.1 [Multimedia Information Systems]: Artificial, augmented, and virtual realities, Evaluation/methodology; H.5.2 [User Interfaces]: Input devices and strategies (e.g., mouse, touchscreen), Interaction styles (e.g., commands, menus, forms, direct manipulation), Evaluation/methodology, Theory and methods, User-centered design; I.3.6 [Methodology and Techniques]: Interaction techniques

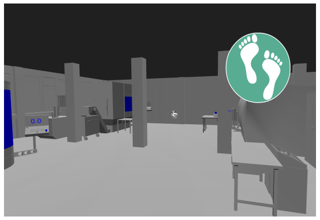

Figure 1: Virtual plant floor as seen through the HMD during the user study.

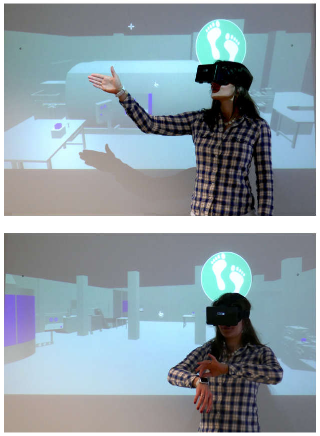

Figure 2: User performing swipe gesture (top) and value setting gesture (bottom). User's view is mirrored to wall in background in order to provide positional information to experiment instructor.

Introduction

Virtual Reality (VR) visual interaction environments make possible the sensation of being physically present in a non-physical world [9]. The value of this experience is a better perception and comprehension of complex data based on simulation and visualization from a near-real-world perspective [4]. A user's sense of immersion, the perception of being physically present in a non-physical world, increases when the used devices are efficient, intuitive, and as “natural" as possible. The most natural and intuitive way to interact with data in a VR environment is to perform the actual real-world interaction [7]. For example, gamers are typically clicking the same mouse button to swing a sword in different directions. However, the natural interaction to swing a sword in a VR application is to actually swing the arm in the physically correct direction as the sword is an extension of the user's arm. Therefore, intuitive and natural interaction techniques for VR applications can be achieved by using the human body itself as an input device [2].

Common technologies – like a flight stick, 3D mouse, or 3D controller with joystick and buttons – do not support body movement gestures and require one to invest a significant amount time for learning. A DataGlove supports the detection of finger movements, position tracking via body suits with in-sewed trackers. Exo-skeletons even make possible full body tracking, but restricting a user when performing interactions, as the user is tethered to the system, cannot walk around, and might fear to damage hardware" [8]. Position tracking done with 3D cameras is relatively cheaper, but it restricts a user's natural behavior as the tracking area is limited and the user must face the camera to avoid occlusion. VR devices are usually specialized to support one interaction modality used only in VR environments. Substantial research has been done in this field, yet VR input devices still lack highly desirable intuitive, natural, and multi-modal interaction capabilities, offered at reasonable, low cost.

We introduce a multi-modal interaction interface, implemented on a smartwatch and smartphone for fully immersive environments. We use a head-mounted display (HMD) for a high degree of immersion. Our approach improves the efficiency of interaction in VR by making possible more natural and intuitive interaction. We have designed and implemented methods for seven gestures and evaluated them comparatively to common VR input technology, specifically body tracking enabled by a 3D camera. We present our approach initially from an application-independent perspective. Later, we demonstrate and discuss its adaptation and utilization for a real-world scenario, as shown in Figure 1 and Figure 2.

Related Work

Bergé et al. [3] state that Mid-Air Hand and Mid-Air Phone gestures perform better than touchscreen input implying that users were able to perform the tasks without training. Tregillus et al. [13] affirm that walking-in-place as a natural and immersive way to navigate in VR potentially reduce VRISE (Virtual reality induced symptoms and effects [12]) but they also address difficulties that come along with the implementation of this interaction technique. Freeman et al. [5] address the issue of missing awareness of the physical space when performing in-air gestures with a multi-modal feedback system.

In order to overcome the lack of current display touch sensors to equip a user with further input manipulators, Wilkinson et al. utilized wrist-worn motion sensors as additional input devices [14]. Driven by the limited input space of common smart watches, the designs of non-touchscreen gestures are examined [1]. Houben et al. consider the chal-

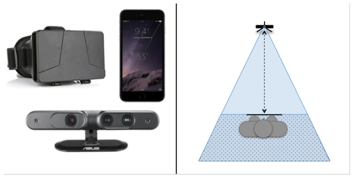

Figure 3: 3D camera setup: Asus Xtion Pro Live 3D camera and VR viewer fixing iPhone 6Plus (left). Viewing angle of camera limits user's movement ability (right).

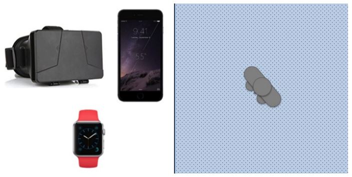

Figure 4: Watch setup: Apple watch sport 38mm and VR viewer fixing iPhone 6Plus (left). Allows usage of the entire physical space for a user's movement (right).

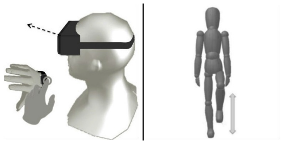

Figure 5: Walk gesture for watch setup(l) and 3D camera setup(r).

lenging task of prototyping cross-device applications with a focus on smartwatches. In their work, they provide a toolkit to accelerate this process with the help of hardware emulation and a UI framework [6].

Current research covers many aspects of interaction in VRs, being of great interest to our work. Similarly, there have been several investigations concerning interaction techniques with wrist-worn devices such as smartwatches and fitness trackers. However, present literature does provide very little insights about eyes-free interaction in VR as well as combination of VR technology, which is of crucial importance when it comes to the utilization of HMDs as an interface to the virtual world. With this paper, we go one step further in closing this gap, employing everyday available low-budget hardware.

Concept

Our approach uses common technologies, at relatively low cost, supporting intuitive, basic interaction techniques already known. A smartphone fixed in an HD viewer serves as fully operational HMD and allows one to experience a virtual environment in 3D space. The smartphone holds the VR application and communicates directly with a smartwatch. Wearing a smartwatch with in-built sensors "moves" the user into the interaction device and leads to a more natural interaction experience. In order to support control capabilities to a great extent, we consider all input capabilities supported by the smartphone and the smartwatch. In addition to touch display and crown, we considered accelerometer, gyroscope and magnetometer, as they are built-in sensors. In discussions with collaborating experts, we determined what types of interaction could and should be realized with the input devices and their capabilities. As the smartwatch has a small display and a user cannot see it, touch input is only used for inaccurate gestures (tap). Most smartwatches have several integrated sensors, e.g., to trace orientation and motion. To obtain platform independence, we decided to focus on accelerometer data as feature of all smart devices during design and implementation of our system. We designed seven distinct gestures dedicated to VR modes of orientation, movement, and manipulation. We have built two setups to enable body gesture interaction. While the first setup relies on body tracking based on a 3D camera, the second one features a smartwatch and its built-in sensors as basic interaction component. To make the approaches fully comparable, the underlying concept of both setups is the same: while hands-free gestures are used to interact within the virtual environment (VE), an HMD provides visual access to the virtual world. The input devices used to capture gestures differ in flexibility and have different limitations discussed in the following sections.

Setups

For both setups we decided to use a smartphone, the Apple iPhone 6+, in combination with a leap HD VR viewer. The smartphone is fixed in the viewer, which, in combination, is fully operational as HMD and allows one to experience a virtual environment in 3D space.

Camera Setup - Our 3D camera-based configuration essentially requires two components: (1) A 3D camera, an Asus Xtion Pro Live, tracks a user's skeleton posture and provides the system with a continuous stream of RGB color images and corresponding depth images and (2) an HMD. The 3D camera is tethered to the main system. A user must remain in small distance to and in field of view of the camera, to be tracked entirely. The tracking radius and the minimal distance of the user enforce a narrowed range of allowable movement, see Figure 3. More specifically, the camera features a 58° horizontal and 45° vertical field-of-view while the tracking distance ranges from .8m to 3.5m.

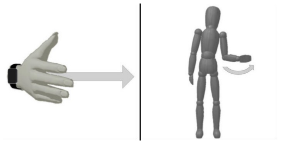

Figure 6: Swipe-left gestures. (L) Watch display faces wall; moving arm horizontally, first in left and then in right direction. (R) Perform swipe gesture with left arm in right direction.

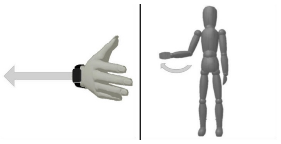

Figure 7: Swipe-right gestures. (L) Watch display faces wall; moving arm horizontally, first in right and then in left direction. (R) Perform swipe gesture with right arm in left direction.

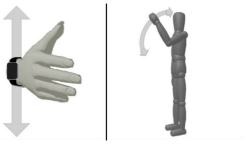

Figure 8: Vertical shaking gesture. (L) Watch display faces wall; fast arm movement in vertical direction. (R) Fast arm movement in vertical direction with right arm.

Another limitation must be applied to a user's orientation to ensure accurate tracking. A user must face the camera to avoid occlusion, preventing the possibility of misinterpretation of body parts or gestures.

Watch Setup - Our watch setup consists of two components: (1) A smartwatch, the Apple Watch Sport 38mm Generation 1 and (2) an HMD. The watch's dimensions are 38.6mm x 33.3mm x 10.5mm. Neither watch nor HMD are tethered, and there is no technical limitation to the tracking area. Also the battery is no limiting factor in our investigation. A user's range of movement is defined by the actual physical space, see Figure 4. One considerable limitation is the fact that body movement gestures are limited to one arm. This limitation implies that all other body parts cannot be utilized for gesturing. Body movements and gestures involving more body parts, like legs, both arms, or torso, would enable a more natural user interface experience.

Software Design and Implementation

Camera Gesture Recognition - In order to enable the system to detect gestures, a framework combining OpenNI 2 with NiTE 2 was designed. While OpenNI handles low-level image processing requirements, NITE serves as a middleware library for detecting and tracking body postures. It supports an easy-to-extend gesture detection framework. Gesture recognition is algorithmically handled via a finite state machine (FSM). Each detectable gesture is represented by a corresponding sequential FSM. In order to trigger the detection of a particular gesture one or more of a user's detected joints are tracked in a certain absolute position and/or relative position to one another. When a body posture indicates a starting condition of an implemented gesture, the system continuously checks for subsequent satisfaction of additional states of the underlying FSM. Once the FSM reaches its final state, the associated gesture is considered as complete. In addition to the gestures available in NITE, we expanded the system by adding several new gestures to satisfy additional needs. For detection, it was crucial to design the additional gestures in such a way that they do not interfere with each other.

Accelerometer-based pattern recognition - Smart watch and smartphone are connected in our framework via Bluetooth, making possible a continuous communication. Accelerometer data collected by the watch are communicated to the phone that computes and detects defined gestures, making use of the smartphone's computation power. It is challenging to devise an algorithm to transform the raw stream of accelerometer data into explicit gestures. Gestures should not interfere with each other, and the system must compute and detect gestures in real time. The resulting data stream to be transmitted and the resulting computation time required for data processing can lead to potential bottlenecks. Applying a low-pass filter to the data stream and dedicated gesture patterns makes it possible to detect necessary changes and to greatly reduce “jittering" of the watch. Thus, the system can effectively distinguish between gestures, which are described in the following.

Interaction Mechanisms

Both setups support the same application, but they differ in input mechanisms. The application is created with Unity3D, which is a cross-platform game engine. VR interaction modes can be grouped into movement, orientation, and manipulation modes. **Orientation** is implemented through head-tracking. A user can look around and orientate oneself. The smartphone uses built-in sensors, like accelerometer and gyroscope, to determine orientation and motion (of the devices), permitting translation, done by the game engine, into the user's viewpoint in a virtual scene. **Movement** is implemented by two interaction techniques: (1) In the watch setup, a user looks in walking direction, and single-touch taps the watch to indicate begin or end of movement. (2) In the 3D Camera setup, a user “walks on the spot," see Figure 5. **Manipulation** refers to the interaction with objects in a scene. For example, we designed and implemented six additional body movement gestures: swipe, in left and right direction; vertical shaking; circle gesture; slider-value setting; and button push, see Figures 6 - 11.

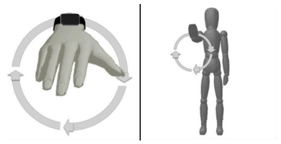

Figure 9: Circle gesture. (L) Watch display faces ceiling; arm movement in small circles clock-wise. (R) Arm movement in big circles clock-wise.

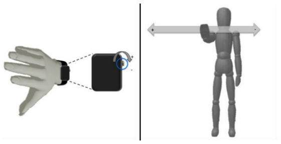

Figure 10: Gesture for value setting. (L) Using scroll wheel of watch; confirming/accepting value by tapping on watch display. (R) Sprawling out right arm; moving in horizontal direction sets value; holding position for three seconds.

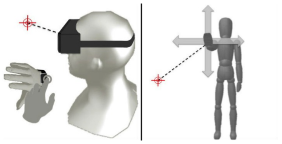

Figure 11: Push gesture. (L) Small point symbolized center of viewpoint; position point on object; approve by tapping on watch display. (R) Sprawl out right arm; Cursor symbolized hand position; position hand on object; hold position for 3 seconds.

User study

In order to find out to what extend working with the 3D Camera-based environment compared to the watch-enabled setup has an effect on a user’s task performance, we conducted a preliminary user study. While performing the experiment, the user is located in a VE constituting a factory building. Latter is an accurate 3D model of a machine hall existing in real world.

Design - There was a total of 20 participants (five of which were female, 15 male, accordingly) taking part in the evaluation and the subject’s age was ranging from 20 to 32. While all of them were used working on a computer on a regular basis, only a few of them had any prior experience concerning HMDs and VRs. Each participant performed the experiment in both of the given setups. Half of the user group began evaluating the 3D-Camera setup while the other half firstly started in the smart watch environment in order to cancel out learning effects while the assignment occurred randomly. Subsequent to the experiment, the participants were asked to fill out a questionnaire consisting of 24 questions considering their user satisfaction.

Realization - In the course of the experiment the subjects were asked to perform several authentic tasks in VR all of which are performed by actual field experts in real life on a daily basis. In total, we considered five machines (stations) in the virtual factory and realistically mapped their control to a sequence of gestures to be performed by the evaluation participant (see Figure 12 and Table1).

Table 1 describes the tasks and gestures of all stations excepting station 5 that is the most comprehensive one and therefore will serve as an illustration of the tasks to perform in this user study. At this station, the machine of interest is a virtual model of a WALTER Helitronic Vision, which is a tool grinding machine. When users reach the machine, they are asked to perform the following tasks in sequence:

1. **swipe left** to open the machine’s sliding door
2. **winding gesture** to rotate the workpiece inside the machine
3. **hammer gesture** to clamp the workpiece
4. **swipe right** to close the sliding door
5. **sliding gesture** to set a specific value at the control panel of the machine
6. **push button gesture** to start the machine

After having all gestures recognized in the correct way and order, the station is considered as finished.

For the purpose of having the whole experimental scenario as realistic as possible, the subjects had to virtually walk to the next station in the sequence before they were able to perform the gestures necessary. Hence it was possible to perform the whole experiment in one go, without having the users distracted or lowered their level of immersion.

As soon as the user reaches a specific station, they are standing in front of the corresponding machine in VR. Since we wanted the distraction and external input to be as low as possible, the users were supported by a pictogram illustrating the gesture currently to be performed. After completion of a sub-task, the pictogram instantly displays the upcoming task. In order to investigate possible differences between the two setups in terms of task performance, we

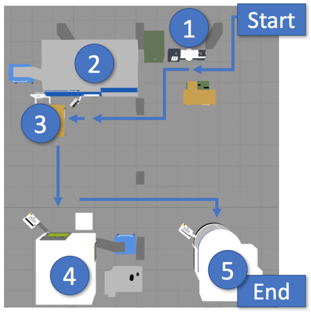

Figure 12: Top view of plant floor consisting of 5 stations.

Station 1 - Wending machine
1.Circle gesture positions workpiece
Station 2 - Turning machine
1.Circle gesture positions workpiece 2.Swipe right closes door 3.Set value gesture sets machining speed 4.Push button starts machine
Station 3 - Hammer
1.Vertical shaking gesture hammering cube in position
Station 4 - Milling machine
1.Swipe left opens door 2.Hammering to clamp piece 3.Swipe right closes door 4.Push button starts machine

Table 1: Gestures translated in user tasks for stations 1 - 4.

documented the time a participant needed to complete a station (i.e. completion of all corresponding gestures). Note that the measured times do not include walking from one station to another.

Results - Since each participant contributed to both of the experiments (within-subject design), we performed a paired t-test on the measured times of both, each station separately and cumulated execution. We then tested the null hypothesis (H_0: there is no significant difference between the given setups) for its tenability with each condition. Regarding the times of stations 1, 2 and 5 exclusively, we found no significant difference in task performance, meaning that the task performance in both setups are equally good, therefore we can not reject the hypothesis H_0. However, we found a significant effect considering stations 3 and 4 solely, as well as total time, with the watch setup outperforming the 3D-Camera setup.

- Station 3: t(19) = 6.70, $p < 0.05$, Cohen's d = 1.50
- Station 4: t(19) = 2.87, $p < 0.05$, Cohen's d = 0.64
- Total time: t(19) = 2.40, $p < 0.05$, Cohen's d = 0.54

As a result, we have a significant difference in task performance in the above cases, which allows us to legitimately reject the null hypothesis H_0. Therefore we can state that the interactions performed with the watch setup are equally good or better than the performance within the 3D camera setup. We could not found a significant difference at stations where the circle gesture was performed. A possible explanation could be found in the questionnaire: the only gesture subjects preferred within the 3D-camera setup over the watch setup was this particular circle gesture. Referring to the questionnaire, there were some interesting findings. Although, it has been assured, that the gestures for both setups are equally comfortable, natural, and intuitive for the users, 5 gestures were more preferred to perform with the watch setup (walk, push, value setting, swipe left, and vertical shaking). The swipe right gesture performance is nearly identical in both setups, which is also confirmed by the questionnaire. Overall, there was a low degree of motion sickness with no significant difference in both setups. These findings lead to the justified assumption that the novel approach presented in this paper is at least as good as currently used techniques.

Conclusions and Future Work

We introduced a combination of smartphone and smartwatch capabilities, outperforming a comparable common VR input device. We have demonstrated the effective use for a simple application. The main advantages of our framework for highly effective and intuitive gesture-based interaction are:

- Location independence
- Simplicity-of-Use
- Intuitive usability
- Eyes-free interaction capability
- Support for several different inputs
- High degree of flexibility
- Potential to reduce motion sickness
- Elegant combination with existing input technology

The demonstrated interaction techniques would be a significant enhancement to existing systems like the collaboration framework like presented from [10] or single user systems like shown in [11]. We plan to enhance the algorithms ad system by also translating accelerometer data into gestures. We will combine device motion data of the smart watch and smartphone to distinguish between more gestures and enable movement in a more natural manner.

Acknowledgements

This research was funded by the German research foundation (DFG) within the IRTG 2057 "Physical Modeling for Virtual Manufacturing Systems and Processes"

References

[1] Shaikh Shawon Arefin Shimon, Courtney Lutton, Zichun Xu, Sarah Morrison-Smith, Christina Boucher, and Jaime Ruiz. 2016. Exploring Non-touchscreen Gestures for Smartwatches. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. ACM, 3822–3833.

[2] Robert Ball, Chris North, and Doug A Bowman. 2007. Move to improve: promoting physical navigation to increase user performance with large displays. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM, 191–200.

[3] Louis-Pierre Bergé, Marcos Serrano, Gary Perelman, and Emmanuel Dubois. 2014. Exploring smartphone-based interaction with overview+ detail interfaces on 3D public displays. In *Proceedings of the 16th international conference on Human-computer interaction with mobile devices & services*. ACM, 125–134.

[4] Steve Bryson, Steven K Feiner, Frederick P Brooks Jr, Philip Hubbard, Randy Pausch, and Andries van Dam. 1994. Research frontiers in virtual reality. In *Proceedings of the 21st annual conference on Computer graphics and interactive techniques*. ACM, 473–474.

[5] Euan Freeman, Stephen Brewster, and Vuokko Lantz. 2016. Do That, There: An Interaction Technique for Addressing In-Air Gesture Systems. In *Proceedings of the 34th Annual ACM Conference on Human Factors in Computing Systems-CHI'16*. ACM Press.

[6] Steven Houben and Nicolai Marquardt. 2015. Watchconnect: A toolkit for prototyping smartwatch-centric cross-device applications. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. ACM, 1247–1256.

[7] Werner A König, Roman Rädle, and Harald Reiterer. 2009. *Squidy: a zoomable design environment for natural user interfaces*. ACM.

[8] Christoph Maggioni. 1993. A novel gestural input device for virtual reality. In *Virtual Reality Annual International Symposium, 1993., 1993 IEEE*. IEEE, 118–124.

[9] Randy Pausch, Dennis Proffitt, and George Williams. 1997. Quantifying immersion in virtual reality. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*. ACM Press/Addison-Wesley Publishing Co., 13–18.

[10] Franca A Rupprecht, Bernd Hamann, Christian Weidig, Jan Aurich, and Achim Ebert. 2016. IN2CO-A Visualization Framework for Intuitive Collaboration. *Eurographics Conference on Visualization (EuroVis) 2016* (2016).

[11] Andreas Schneider, Daniel Cernea, and Achim Ebert. 2016. HMD-enabled Virtual Screens as Alternatives to Large Physical Displays. In *Information Visualisation (IV), 2016 20th International Conference*. IEEE, 390–394.

[12] Sarah Sharples, Sue Cobb, Amanda Moody, and John R Wilson. 2008. Virtual reality induced symptoms and effects (VRISE): Comparison of head mounted display (HMD), desktop and projection display systems. *Displays* 29, 2 (2008), 58–69.

[13] Sam Tregillus and Eelke Folmer. 2016. VR-STEP: Walking-in-Place using Inertial Sensing for Hands Free Navigation in Mobile VR Environments. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. ACM, 1250–1255.

[14] Gerard Wilkinson, Ahmed Kharrufa, Jonathan David Hook, Bradley Pursgrove, Gavin Wood, Hendrik Haeuser, Nils Hammerla, Steve Hodges, and Patrick Olivier. 2016. Expressy: Using a Wrist-worn Inertial Measurement Unit to Add Expressiveness to Touch-based Interactions. In *Proceedings of the ACM Conference on Human Factors in Computing Systems:*. Association for Computing Machinery (ACM).