

PROCEEDINGS OF SPIE

[SPIDigitalLibrary.org/conference-proceedings-of-spie](https://spiedigitallibrary.org/conference-proceedings-of-spie)

Possible applications of the LEAP motion controller for more interactive simulated experiments in augmented or virtual reality

Peter Wozniak, Oliver Vauderwange, Avikarsha Mandal, Nicolas Javahiraly, Dan Curticaean

Peter Wozniak, Oliver Vauderwange, Avikarsha Mandal, Nicolas Javahiraly, Dan Curticaean, "Possible applications of the LEAP motion controller for more interactive simulated experiments in augmented or virtual reality," Proc. SPIE 9946, Optics Education and Outreach IV, 99460P (27 September 2016); doi: 10.1117/12.2237673

SPIE.

Event: SPIE Optical Engineering + Applications, 2016, San Diego, California, United States

Possible applications of the LEAP Motion controller for more interactive simulated experiments in Augmented or Virtual Reality

Peter Wozniak^{*a}, Oliver Vauderwange^a, Avikarsha Mandal^a, Nicolas Javahiraly^b, Dan Curticeapean^a

^a Offenburg University, Badstr. 24, 77652 Offenburg, Germany

^b University of Strasbourg / ICube, Boulevard Sébastien Brant, BP 10413, F-67412, Illkirch France

ABSTRACT

Practical exercises are a crucial part of many curricula. Even simple exercises can improve the understanding of the underlying subject. Most experimental setups require special hardware. To carry out e. g. a lens experiments the students need access to an optical bench, various lenses, light sources, apertures and a screen. In our previous publication we demonstrated the use of augmented reality visualization techniques in order to let the students prepare with a simulated experimental setup. Within the context of our intended blended learning concept we want to utilize augmented or virtual reality techniques for stationary laboratory exercises.

Unlike applications running on mobile devices, stationary setups can be extended more easily with additional interfaces and thus allow for more complex interactions and simulations in virtual reality (VR) and augmented reality (AR). The most significant difference is the possibility to allow interactions beyond touching a screen. The LEAP Motion controller is a small inexpensive device that allows for the tracking of the user's hands and fingers in three dimensions. It is conceivable to allow the user to interact with the simulation's virtual elements by the user's very hand position, movement and gesture.

In this paper we evaluate possible applications of the LEAP Motion controller for simulated experiments in augmented and virtual reality. We pay particular attention to the devices strengths and weaknesses and want to point out useful and less useful application scenarios.

Keywords: Education, Practical Lectures, Blended Learning, Human-Computer Interaction, LEAP motion, Augmented Reality, Virtual Reality

1. INTRODUCTION

Augmented and virtual reality technologies can be used for various visualization purposes. Our previous publication demonstrated the usage of augmented reality as a supplement for physics laboratory exercises. We showed how to utilize regular Android or iOS mobile devices to perform light and optical experiments in augmented reality. For our demonstration we picked a simple lens experiment, which is meant to give the students a basic understanding for the laws of optics and its applications. In many scientific studies this type of experiment is already part of the curriculum. To carry out this lens experiments the students need special hardware, thus it is not possible to prepare other than theoretically for the experiments at school. The possibility to simulate essential parts of the real experiment and the widespread availability of handheld mobile devices allows it to perform simulated virtual experiments principally everywhere.

In our previous publication we concentrated on this specific field of applications. The aim of this publication is to discuss a new kind of input device and its use for augmented reality based virtual experiments. The LEAP motion allows for a spatial tracking of a user's hands. It is specifically designed to interact with computers by using natural hand movements and gestures. We aim to describe the possibilities of this device and to outline suitable areas of applications, in particular for augmented reality based applications. Further we want to highlight advantages and disadvantages emerging from a gesture based human-computer-interaction.

*peter.wozniak@hs-offenburg.de; phone +49 781 / 205 – 350

2. THEORETICAL BACKGROUND

Virtual and augmented reality are closely related technological siblings. They both share common technologies, both are defined as visualization technologies and as such they characterize the applications they allow.

Azuma defines augmented reality (AR) independently of the used technology as a method to combine real and virtual elements interactively, in real-time and in three dimensions [1]. Another popular definition is Milgram's reality-virtuality continuum [2]. This continuum spans between the reality and virtuality and allows every form of mixed reality in between (Figure 1). While AR is closer to reality, augmented virtuality is closer to virtuality, which is also referred to as virtual reality (VR). In VR the user is completely immersed into a virtual world without any real elements and therefore represents the opposite of reality.

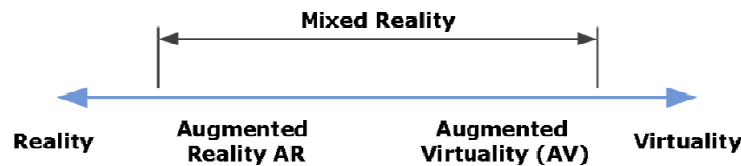


Figure 1 Milgram's reality-virtuality continuum.

Our publication mainly refers to augmented reality, but where necessary we will also refer to virtual reality applications. Augmented reality describes a concept of extending the perceived real world by additional computer generated content. Ideally virtual and real elements combine into an extended, an “augmented” experience. Basically augmented reality applications visualize any suitable sort of information by superimposing it on top of the seen. Popular areas of AR applications are education, military, medical, navigation, automotive sector and of course entertainment. Depending on the type of application the requirements and implementations vary a lot. However the details may look like, most AR systems are at least composed of the following three components: a display, a tracking system and content generation. In addition most AR application setups require some kind of human-computer-interaction and therefore a suitable interface.

2.1 AR displays

Although an AR system may incorporate other senses than the sense of sight, vision is generally considered to be the main sense addressed by almost all AR systems. Thus a display which allows presenting some sort of computer generated content to a user is usually mandatory and fundamental for an AR system. Generally it is possible to categorize most systems into visible-see-through (VST) display or optical-see-through (OST) display based systems. Both principles have their specific advantages but also disadvantages.

OST displays allow a direct view of the real world. The superimposition is achieved via a translucent display. The natural view on the world remains mostly intact, which is usually preferable for critical applications like car navigation or surgery aids. With an OST display it is not trivial to cover bright real objects with darker virtual content [3]. OST based displays are more complicated due to the need for calibration, as every user has a different anatomy. The missing possibility to directly check the outcome and quality of the augmentation process, as the system has no direct feedback. It is rather the user that can evaluate the superimposition process and thus is usually required to perform a calibration process in the beginning. As image processing and tracking cannot perform in real-time, delay issues are common for OST displays. Fast head rotations or fast object movements lead to the problem that the virtual superimposed objects are not correctly aligned and are shown not at the correct position but where the real object was shortly before. The illusion of a coexisting real and virtual world quickly falls apart with an incorrect image registration.

VST displays use a video imaging system that captures the real environment, whose images are processed and superimposed before the user perceives them on some kind of a display system. The user is incapable to perceive the real world directly while using a VST display, which could raise possible security issues and hazards with certain applications. VST are better suited for Mediated Reality applications as it is easier to remove unwanted objects from a

video image then try to occlude them with an OST display. Delays are also present but don't represent such a big problem as the video stream and rendered overlay can be synchronized easily. With VST systems the user perceives the reality by the use of a video system, thus the resolution of the camera and the display determines the possible visual impression. It is reasonable to assume that no technology will recreate a natural visual impression in the near future and thus VST systems will lack realism in this field. In comparison to OST systems, where the system has no direct feedback, VST systems can use computer vision methods, independently of the used tracking technique, to analyze the video image directly and to verify the outcome of the superimposition, which gives them the possibility to automatically perform necessary corrections to the image registration process.

AR displays can also be further categorized by the following characteristics:

- Mobile or stationary
- Head-mounted, handheld or permanently mounted

These characteristics generally derive from or rather relate to the type of the application. Mobile devices like smartphones or tablet computers with integrated cameras can be easily used as mobile VST AR systems as they include all necessary components to run AR applications. Those almost ubiquitous devices are fast enough to allow tracking, content rendering and image registration in real-time.

Head-mounted OST or VST devices are becoming smaller and smaller and are thus better suited for everyday use. On the other hand head-mounted devices are still uncommon and attract attention. So a breakthrough cannot be expected until the devices are as unobtrusive as regular glasses. Stationary AR displays are more common in industry projects or tech demos like e.g. Microsoft's Holodesk, where a user sees through a stationary half transparent OST and can interact with real and virtual objects [4].

2.2 Content rendering and image registration

AR applications are usually meant to visualize some kind of data or information that is related to the real scenery. E.g. a car navigation system showing the planned route and highlighting important navigation details. Additionally the car computer system could emphasize important information (augmented vision), like the current speed, outdoor temperature and additional road users. On the other hand medical AR applications are meant to visualize completely different information. Thus it is clear that depending on the aim of the application, not only the setup and implementation but also the augmented content varies a lot.

For our virtual lens experiment application we needed to simulate simple lenses, light sources and a bunch of light rays passing through. Although our simulation was not very sophisticated it was sufficient to allow the students to understand the underlying physical principles. The same applies to the processed and visualized data for any AR system.

Generally the content needs to be visualized three-dimensionally in order to be superimposed perspectively correct on the real world view (Figure 2).

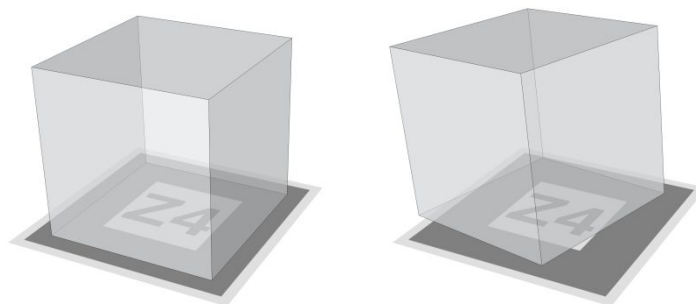


Figure 2 Visualization of a successful and failed image registration of a superimposed cube on top of an ARToolkit marker.

This so called image registration process depends heavily on the quality and precision of the tracking. For a realistic looking outcome it might also become necessary to not only consider the positional and orientational tracking. Light conditions, shadows and occlusion of real and virtual elements does matter as well to get a nice and realistic augmentation. Different focus planes of real and virtual elements could interfere with the user's experience. With video based AR systems the resolution, color temperature and motion blur could become an issue. With optical-see-through displays transparency, occlusions and the contrast of bright and dark elements become a matter. All current stereoscopic head-mounted glasses lack realistic field of depth. Generally the content is presented on one fix depth layer, causing the vergence-accommodation conflict and leading to discomfort [5]. Current research tries to solve this conflict of vergence and accomodation by developing and utilizing volumetric, multi-focal-plane or light-field display technologies. Although single techniques prove to be suitable there are still plenty of unsolved problems [6] [7] [8] [9] [10] [11].

2.3 Tracking

In order to superimpose virtual onto real content, the AR systems needs to be able to match the users viewing perspective onto the scenery and the virtual content that shall be shown. The recognition of space and objects placed in the scenery appears to be trivial and natural to a human being, but is a complex task for a computer system. In order to superimpose realistically virtual images on the view of the real world, AR systems rely on various tracking methods. Tracking allows for a constant determination of the position and orientation of parts of the AR system, the user and/or objects in the environment in relation to each other. Depending on the AR application the requirements for the tracking system differ significantly, but generally the tracking should have the following ideal characteristics:

1. Flawless and precise.
2. Robust.
3. Zero latency.
4. 6-Degrees of freedom.
5. Universally applicable.
6. Easy to deploy and use.
7. Affordable.

It is possible to categorize tracking systems by the means of their principal tracking concept into Outside-In and Inside-Out systems. E.g. a system that uses a marker placed on a head-mounted-display that is captured with one or multiple cameras placed in a room is categorized as Outside-In. The camera images are used to determine the position and orientation of the marker and thus the display on the users head. The objects are basically tracked from the "outside".

Inside-Out systems reverse this principle. The objects are equipped with sensors that allow the objects to be tracked from the "inside". E.g. the head-mounted-display could be equipped with camera sensors capturing markers that are placed within the room. Beside these basic categorization it is possible to further categorize tracking systems on the basis of the used physical principle. Tracking systems are based on:

- creating and measuring artificial magnetic fields,
- inertial sensors measuring the orientation and positional change,
- attached mechanical arms determining the position and orientation,
- runtime based methods and
- optical or image processing based methods.

In the field of AR especially the optical tracking methods are popular. Mobile handheld AR applications mostly rely on printed marker, GPS position, magnetic- and inertial-sensing. Recently Microsoft Hololens and Google Project Tango entered public beta phase, which are relying on markerless optical tracking methods [12] [13].

3. HUMAN-COMPUTER-INTERFACES FOR AR AND VR APPLICATIONS

A user interface that allows for a human-computer-interaction is not strictly mandatory for AR systems. Nevertheless most AR applications offer some kind of interaction. Depending on the aim of the application the interface differs greatly. Medical surgeons, who try to avoid any unnecessary contact with non-sterile items but still need to interact with more and more computerized equipment, possibly prefer adequate controls like gestures and voice commands. Mobile

handheld device users usually rely on the ubiquitous and well-known touchscreen, at least as long as they hold their device in the hands. Touchscreen based devices lack precision and tactical feedback. Nevertheless the users got used to its drawbacks and learned to live with them.

As soon as the mobile device is mounted in a VR viewer [14] the touch interface gets useless and new forms of interaction become necessary. Google introduced a magnetic trigger and, with the newer cardboard v2 design, a simple mechanical button touching the upper corner of the touch display. The user interacts by gazing onto onscreen user interface (UI) elements that gets activated after some visualized delay (Figure 3). This principle is quite straight forward and easy to perform, but also very limited in its functionality as it only allows for triggering. The same applies for the optionally available magnetic trigger. The newer mechanical button on the other hand allows long press actions and more sophisticated interactions like e.g. drag-and-drop functionality. As soon as the cardboard VR-viewer gets equipped with head straps it becomes a very low-cost HMD.

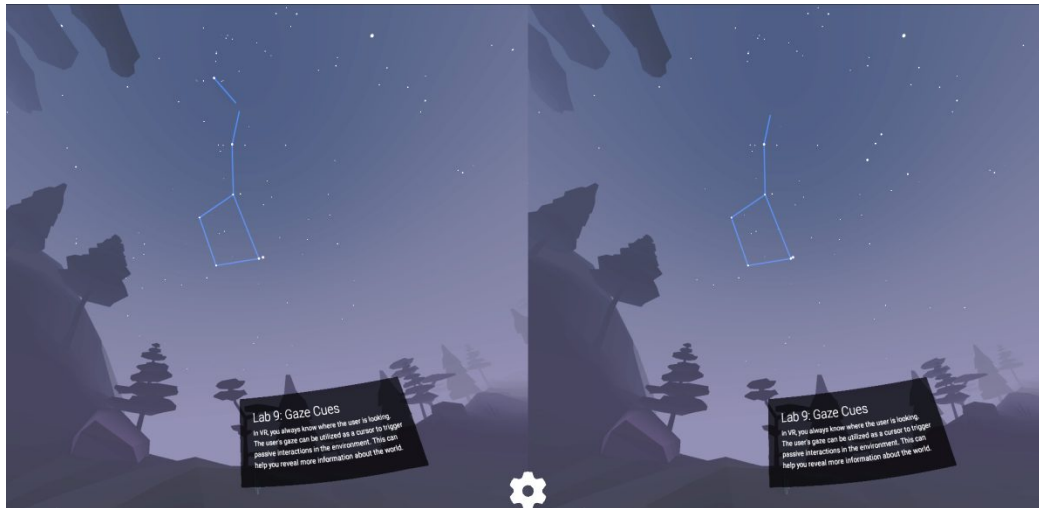


Figure 3 Screenshot from Google Cardboard Design Lab app. Demo of gaze activated UI elements [15].

It is uncommon for HMD based applications to rely on a keyboard and mouse interface, as they are not directly visible to the user. Thus HMD based applications are ideally suited for motion controller interaction. Motion sensing controllers allow an interaction by capturing the motions performed by a user and translating them into some information that can be further processed. Particularly computer-games utilize these devices to allow novel forms of interaction. A popular device is the Microsoft Kinect that is able to capture sceneries in three dimensions [16]. The three dimensional imaging allows for a skeletal tracking and gesture recognition and thus a hands-free computer interaction. Another device is the handheld Nintendo Wii Remote controller [17]. Built-in accelerometer and optical sensors allow gesture recognition and pointing for interaction. The HTC Vive VR HMD is sold together with two separately tracked handheld controllers. The positional and orientational tracking along with the buttons allow the user to interact in VR [18]. The designated Google cardboard successor Google Daydream specifies an Android HMD based VR viewer and a separate handheld controller [19]. The controller can track its rotation and orientation with high accuracy and probably allows a gesture and button based interaction, but no positional tracking like the HTC Vive controller. Another affordable consumer input device is the LEAP motion controller, which is going to be described in detail in the following chapter.

In any case the human-computer-interface needs to be well planned and designed for any specific application. To be accepted, the interface must be easy to learn and to use. For this new standards and guidelines of UIs have to be established. As usual a generalization is difficult. The professional user might be willing to spend more time on learning and mastering a complicated and complex user-interface than the casual user that wants to run a quick test of the newest AR game.

4. THE LEAP MOTION CONTROLLER

The LEAP motion controller is designed to track hand and finger movements in space and can be used for human-computer-interaction. The company of the same name as the device has been founded in 2010. By end of 2012 the first tracking devices has been send to developers and by mid-2013 the device has been freely available. With the emerging boom in VR HMDs the company in 2016 released a new SDK with added VR support [20].

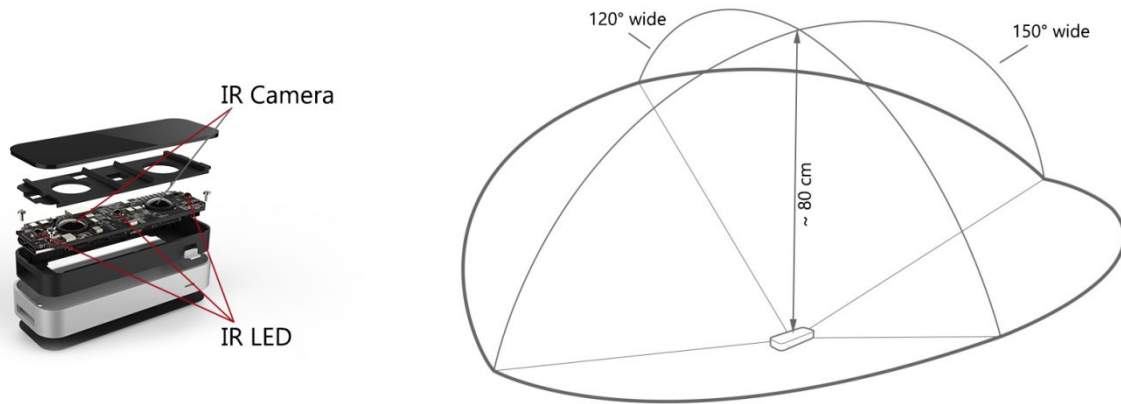


Figure 4 Exploded view of the LEAP motion device and visualization of action area [21].

The LEAP motion controller is a small USB device that is meant to be placed on the desktop, e.g. next to the keyboard. For VR usage the controller can be attached onto the HMD facing in the users viewing direction.

The Leap motion device can be used with Microsoft Windows, Mac OS and Linux. The newest VR ready software is windows only. The device is made for PC use and thus only as mobile as a PC is. There is however a closed alpha SDK promising a future of LEAP motion based mobile applications [22].

The LEAP motion controller is an infrared light based stereoscopic camera. By illuminating the close space near to the cameras with infrared light it is able to capture a user's hands and fingers. By utilizing some sort of tracking algorithm it is able to estimate the position and orientation of the hand and its fingers. The maximum infrared illumination is limited by the maximum current that can be drawn over the USB connection. Strong sunlight or poor lightning conditions can interfere with the tracking quality. The controller's action area is about 150° by 120° wide and spans about 80 cm originating from the device (Figure 4) [21]. The tracking precision has been proved to be remarkably good. The standard deviation is less than a millimeter, although an accuracy drop with higher distances has been noted [23] [24]. Although the camera can't see occluded parts of the hands, the algorithm can estimate to some extend typical hand gestures and thus provide astonishing useful tracking. Nevertheless the software can't fulfill wonders and thus sometimes fails. The device performs best with the whole hands visible.

Positional and orientational tracking

The basic functionality of the LEAP motion is the ability to track a user's hands. The position and orientation of the hand and its fingers are being tracked and can be directly mapped onto a virtual skeleton model. The SDK provides easy access to data of all fingers and its bones (Figure 5).

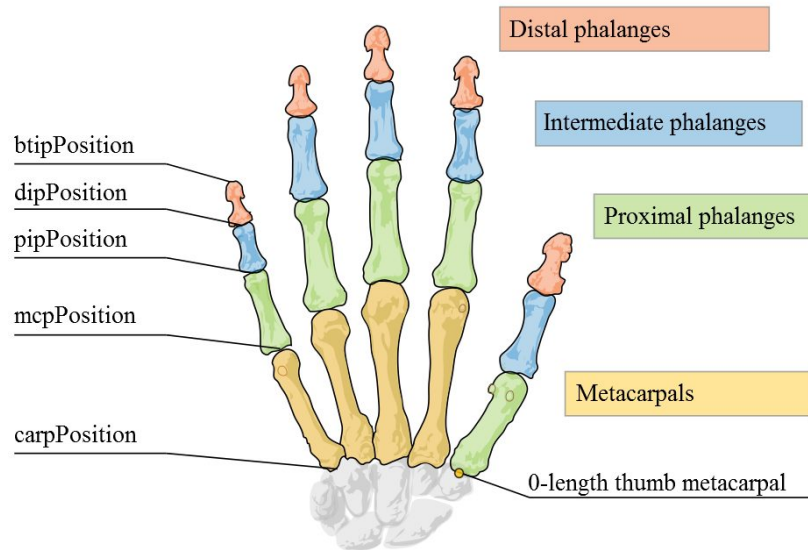


Figure 5 LEAP motion SDK: Skeletal tracking model [25].

It is therefore straight forward to anticipate using the tracking data to control a virtual representation of the user's hand (Figure 6). The user's hands can be visualized in VR applications, to increase the presence in the virtual world. In addition to the tracked data it is possible to fade-in the image data of the LEAP motion IR-camera.



Figure 6 Virtual and real hand of a user interacting with a VR demo [27].

Gesture recognition

To improve interaction possibilities the LEAP SDK is able to detect hand gestures. Gestures can be used to trigger events or can be combined with the positional and orientational tracking data. The SDK knows the following four basic gestures (Figure 7):

- Circle - A single finger tracing a circle.
- Swipe - A long, linear movement of a finger.
- Key Tap - A tapping movement by a finger as if tapping a keyboard key.
- Screen Tap - A tapping movement by the finger as if tapping a vertical computer screen.

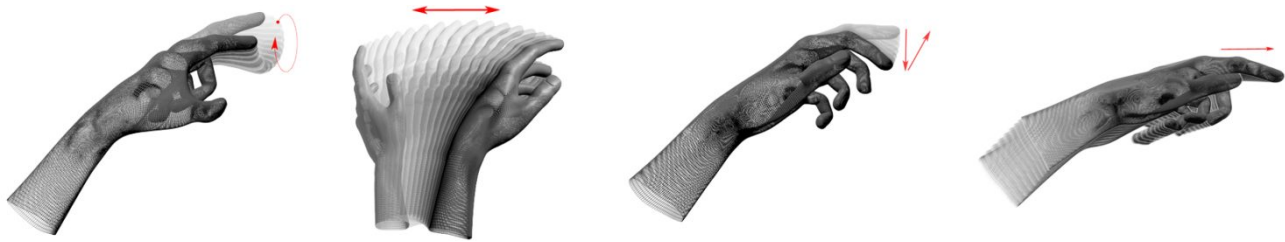


Figure 7 Basic LEAP motion gestures (f.l.t.r circle, swipe, key tap, screen tap) [28]

Further the LEAP motion software is able to detect pinching and grabbing gestures that can be used context-sensitive to directly manipulate the virtual elements. E.g. it is possible to grab, rotate and reposition objects. Two-handed pinch-gestures can be used to allow zoom actions, similar to the well-known zoom-pinch gestures on touchscreens, or to select and manipulate virtual dial- or slider-controllers.

These gestures are relatively simple and easy to learn. In order to be understood by the user they need to be explained first. It is considered good practice, to introduce the key concepts of the user interface with an interactive tutorial and to give the user in-app cues and hints. The current state of the application and the next possible actions should always be clear to a user. User interfaces should be designed to be forgiving in the case of operating errors. Possible errors should be made impossible or if not avoidable there should be an undo option.

The LEAP motion app “Blocks” demonstrates the design principles of a VR gesture based UI by the means of a simple game (Figure 8). The app incorporates all important elements for a pleasant user experience. A short interactive tutorial explains the interaction possibilities to the user. Immersed in the virtual world the user can see a virtual set of hands and is able to create and interact with virtual blocks. The UI is gesture based and visual cues allow the user to interact absolutely intuitively [31].

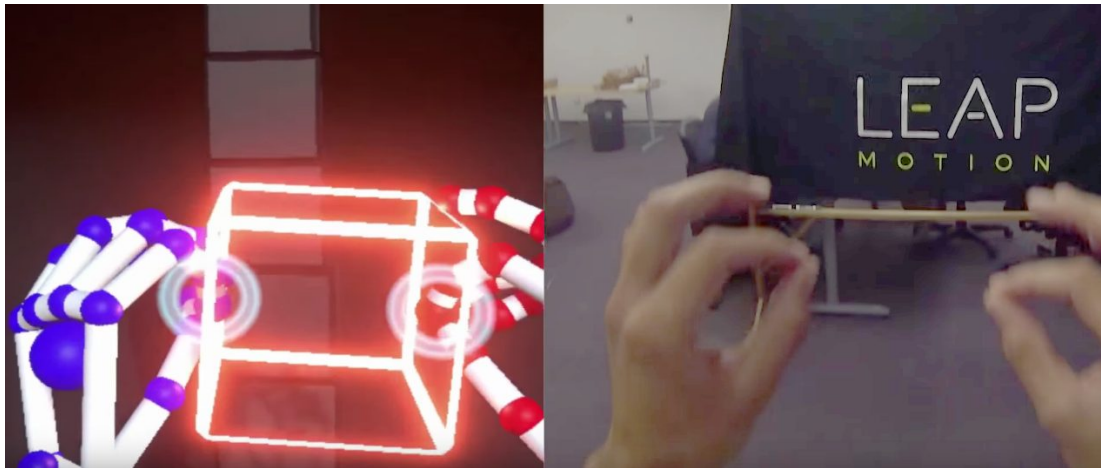


Figure 8 Screenshot from LEAP motion VR game Blocks demonstration [31].

Another very interesting project is “AR screen”. The application is also based on a VR HMD and a LEAP motion successor prototype called “Dragonfly”. Unlike the LEAP motion, the Dragonfly controller is able to capture high-resolution color images and thus converting a VR HMD into a VST AR HMD. The AR screen application demonstrates the extension of a common Windows desktop into AR (Figure 9) [32].

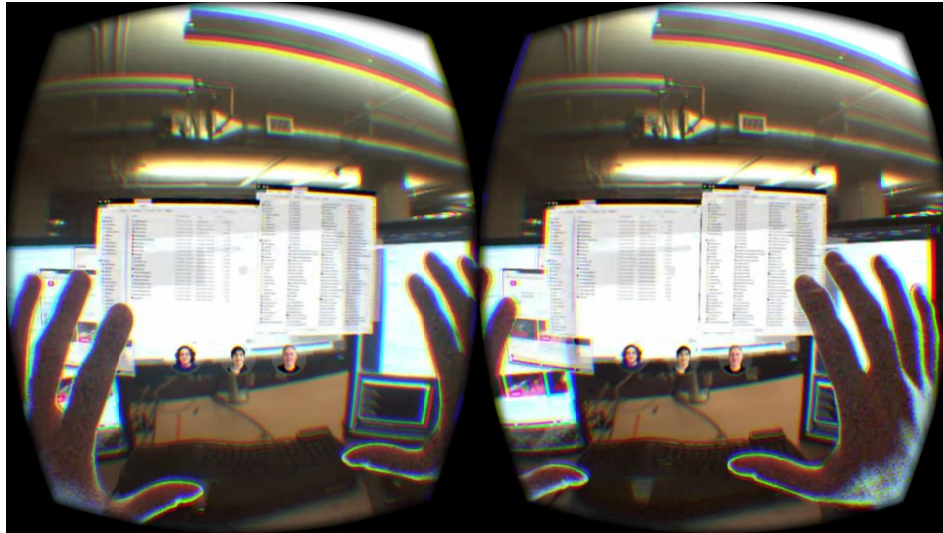


Figure 9 Screenshot from "AR screen" demonstration [32].

Tool tracking

The SDK also allows the tracking of pen-shaped tools. These tools can be used to perform arbitrary tasks in the air. Due to the uniformity of a pen it is not possible to track longitudinal rotations, although some people managed to work around that restriction (Figure 10).

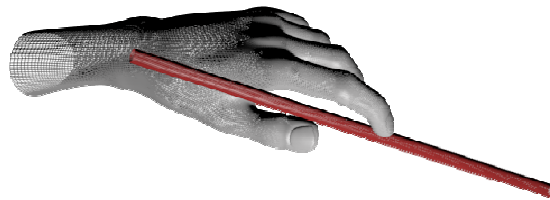


Figure 10 LEAP motion tool tracking [29]

Unfortunately there is no official support yet that allows a tracking of custom defined objects that could be used as additional tools. This restriction probably arises due to the used algorithm, which prioritizes fast hand tracking but lacks the flexibility to detect custom objects [26]. Also the newest VR capable SDK doesn't support tool tracking yet [29]. A short video from Michael Quigley demonstrates the tracking of a custom extended pen-shaped medical surrogate tool [30]. Interestingly enough the tool has been equipped with an extension that allows the additional tracking of longitudinal rotations of the tool.

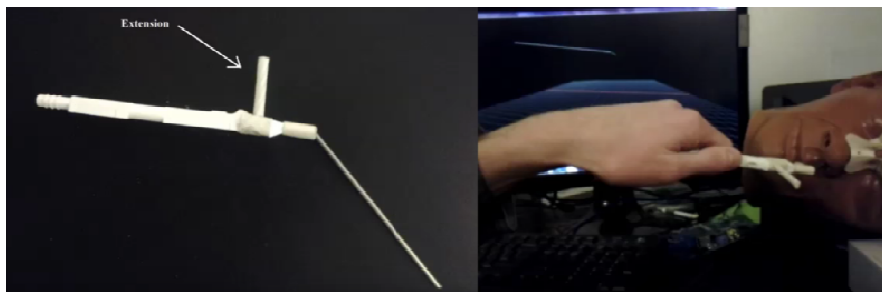


Figure 11 Tracking of a pen-shaped tool with an extension [30].

A virtual representation of the tool can be seen on a regular computer display (Figure 11). More sophisticated simulations including more realistic interaction could help to visualize the human anatomy and to understand surgeries better. The same technique could be applied for various other educational simulators. A similar functionality can also be realized with VST AR systems and the current LEAP motion controller. The project VoxelAR (Figure 12) demonstrates a hybrid AR interface for finger-based interaction and occlusion detection. An additional webcam is used to capture a color image of a user's hand. The system was used for physical rehabilitation [33].

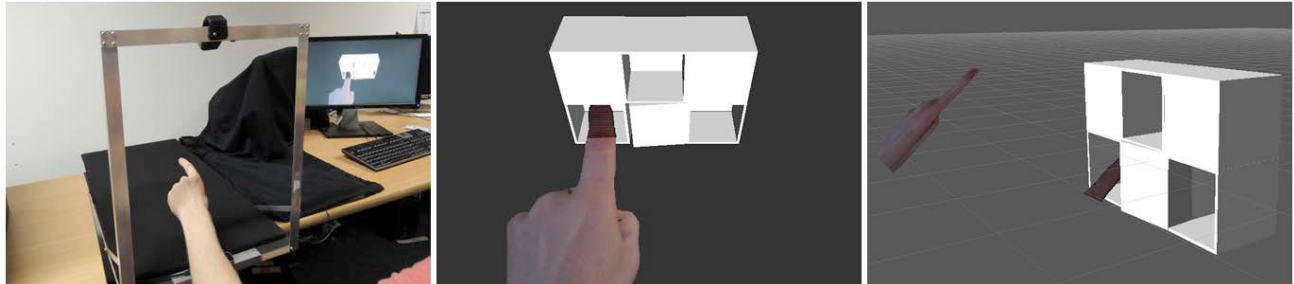


Figure 12 VoxelAR hybrid AR interface [33].

5. DISCUSSION

The previous chapter showed the possibilities of the LEAP motion controller and also how it's used by some interesting applications. Criteria for a meaningful use of the LEAP motion controller for interactive virtual experiments can be various and it is necessary to consider them in relation to the applications aim. Solely because the LEAP motion can be used to control any conceivable application, it is not necessarily a good idea to do that. Gesture-based user interfaces are still new and therefore the developers cannot rely on long-term user experience. In general it can be stated that the LEAP motion is not a replacement for a keyboard and mouse interface in connection with a desktop application. A gesture based interaction is rather an extension to the existing interface, broadening the possible user experience. Under all circumstances the UI should remain coherent to the user. The number of used gestures should remain low in order to keep the UI simple. The LEAP motion is best suited for interactive tasks where the user needs three-dimensional hand interaction. The software shouldn't rely on the exact positioning of the user's hands in space, in order to keep it simple and less frustrating. Also one should also keep in mind that the user has no tactile feedback from the interaction with virtual elements. Therefore it is necessary to give the user enough visual cues, in order that the brain can associate the performed interaction with the seen one on the screen.

For VR interaction the premise is similar except for the redundancy of keyboard and mouse interaction, because it is not easy to use them with a VR HMD. In fact the LEAP motion can show one's pluses with VR applications. The stereoscopic view makes the spatial coordination easier and more complex gesture based interfaces are possible. Of course the above mentioned criteria, like e.g. the highly important visual cues, remain crucial for a successful interaction.

6. CONCLUSION

We described typical components of AR and VR systems and outlined the most familiar interfaces for human-computer-interaction in VR and AR. We described the possibilities of the LEAP motion controller and highlighted some interesting already existing applications. In the following discussion we outlined and described criteria for best practice in designing a meaningful LEAP motion based interaction for interactive applications like simulated experiments.

The LEAP motion controller is a small, inexpensive and easy to setup device for hand tracking. The device can be used for common desktop based, but also novel VR applications. The SDK allows various interaction forms, from simple gestures up to complex gesture-based user interfaces. Especially VR applications can benefit greatly from natural motion interaction, whereby the biggest advantage is the visibility of its own hands in virtual reality. The main challenge is the design of a smart user interaction concept that utilizes the strengths of a gesture based interaction without overtaxing the user. The use of the LEAP motion controller for simulated experiments in AR and VR can be a useful supplement, but there is also the risk of great distraction from the main educational task.

REFERENCES

- [1] Azuma, R., T., "A Survey of Augmented Reality," *Teleoperators and Virtual Environments* 6, 4 (August 1997), 355-385 (1997).
- [2] Mehler-Bicher, A., Reiß, M. and Steiger, L., „Augmented Reality : Theorie und Praxis,“ München : Oldenbourg Verlag (2011).
- [3] Kiyokawa, K., Kurata, Y., & Ohno, H., „An Optical See-through Display for Enhanced Augmented Reality,“ *Poster Proceedings of Graphics Interface*, 10-11 (2000).
- [4] Wilson, A., "HoloDesk: Direct 3D Interactions with a Situated See-Through Display," Microsoft, 24 February 2012, <<https://www.microsoft.com/en-us/research/project/holodesk-direct-3d-interactions-with-a-situated-see-through-display/>> (22 July 2016).
- [5] Shibata, T., Kim, J., Hoffman, D. M., Banks, M. S., "Visual discomfort with stereo displays: effects of viewing distance and direction of vergence-accommodation conflict," *Stereoscopic Displays and Applications XXII* (2011).
- [6] Sawada, S., Kakeya, H., "Integral volumetric imaging using decentered elemental lenses," *Opt. Express Optics Express* **20**(23), 25902 (2012).
- [7] Banks, M. S., Kim, J., Shibata, T., "Insight into vergence/accommodation mismatch," *Head- and Helmet-Mounted Displays XVIII: Design and Applications* (2013).
- [8] Love, G. D., Hoffman, D. M., Hands, P. J., Gao, J., Kirby, A. K., Banks, M. S., "High-speed switchable lens enables the development of a volumetric stereoscopic display," *Opt. Express Optics Express* **17**(18), 15716 (2009).
- [9] Huang, F.-C., Chen, K., Wetzstein, G., "The light field stereoscope," *TOG ACM Trans. Graph. ACM Transactions on Graphics* **34**(4) (2015).
- [10] Stern, A., Yitzhaky, Y., Javidi, B., "Perceivable Light Fields: Matching the Requirements Between the Human Visual System and Autostereoscopic 3-D Displays," *Proceedings of the IEEE Proc. IEEE* **102**(10), 1571–1587 (2014).
- [11] Huang, H., Hua, H., "Design of an Optical See-through Multi-Focal-Plane Stereoscopic 3D Display With Eye-tracking Ability," *Imaging and Applied Optics* 2016 (2016).
- [12] "Microsoft HoloLens," Microsoft HoloLens, <<https://www.microsoft.com/microsoft-hololens>> (27 July 2016).
- [13] "Google Tango.," Google, <<https://get.google.com/tango/>> (27 July 2016).
- [14] "Cardboard.," Google – Google VR, <<https://vr.google.com/cardboard/>> (27 July 2016).
- [15] "Cardboard Design Lab - Android Apps on Google Play.," Google Play, <<https://play.google.com/store/apps/details?id=com.google.vr.cardboard.apps.designlab>> (28 July 2016).
- [16] "Kinect hardware.," Microsoft Dev Center, <<https://developer.microsoft.com/en-us/windows/kinect/hardware>> (27 July 2016).
- [17] "Wii Remote.," Wikipedia, <https://en.wikipedia.org/wiki/wii_remote> (27 July 2016).
- [18] "HTC Vive SteamVR SDK.," HTC Dev, <<http://www.htcdev.com/devcenter/opensense-sdk/htc-vive-steamvr-sdk>> (27 July 2016).
- [19] "Daydream - Google VR.," Daydream - Google VR, <<https://vr.google.com/daydream/>> (27 July 2016).
- [20] Feltham, J., "Leap Motion Announces Orion for Faster, More Accurate VR Hand Tracking," *VRFocus*, 17 February 2016, <<http://www.vrfocus.com/2016/02/leap-motion-announces-orion-for-faster-more-accurate-vr-hand-tracking/>> (31 July 2016).

- [21] Colgan, A., "How Does the Leap Motion Controller Work?," Leap Motion Blog, 9 August 2014, <<http://blog.leapmotion.com/hardware-to-software-how-does-the-leap-motion-controller-work/>> (27 July 2016).
- [22] Colgan, A., "Android SDK Hardware Requirements," Leap Motion Community, 4 April 2015, <<https://community.leapmotion.com/t/android-sdk-hardware-requirements/2845>> (27 July 2016).
- [23] Weichert, F., Bachmann, D., Rudak, B., Fisseler, D., "Analysis of the Accuracy and Robustness of the Leap Motion Controller," *Sensors* **13**(5), 6380–6393 (2013).
- [24] Guna, J., Jakus, G., Pogačnik, M., Tomažič, S., Sodnik, J., "An Analysis of the Precision and Reliability of the Leap Motion Sensor and Its Suitability for Static and Dynamic Tracking," *Sensors* **14**(2), 3702–3720 (2014).
- [25] "Introducing the Skeletal Tracking Model.," Leap Motion JavaScript SDK v2.3 documentation, <https://developer.leapmotion.com/documentation/javascript/devguide/intro_skeleton_api.html> (27 July 2016).
- [26] "Is it possible to get raw point cloud data?," Leap Motion Support, <<https://support.leapmotion.com/entries/40337273-is-it-possible-to-get-raw-point-cloud-data->> (27 July 2016).
- [27] "Orion Beta.," Leap Motion Developer, <<https://developer.leapmotion.com/orion>> (27 July 2016).
- [28] "Gestures.," Leap Motion C# SDK v2.3 documentation, <https://developer.leapmotion.com/documentation/csharp/devguide/leap_gestures.html> (27 July 2016).
- [29] "Tool.," Leap Motion Java SDK v3.1 documentation, <<https://developer.leapmotion.com/documentation/java/api/leap.tool.html>> (27 July 2016).
- [30] "Optical surgery tool tracking.," Unofficial LEAP Motion tech demos experiments collection, 16 October 2013, <<http://leap.quitebeyond.de/optical-surgery-tool-tracking/>> (27 July 2016).
- [31] leapmotion., "Leap Motion Blocks for HTC VIVE Playthrough," YouTube, 22 January 2016, <<https://www.youtube.com/watch?v=vvlojsswbx8>> (28 July 2016).
- [32] Bedikian, R., "The Leap Motion Hackathon's Augmented Reality Workspace," Leap Motion Blog, 21 July 2015, <<http://blog.leapmotion.com/hood-leap-motion-hackathons-augmented-reality-workspace/>> (27 July 2016).
- [33] Regenbrecht, H., Collins, J., Hoermann, S., "A leap-supported, hybrid AR interface approach," *Proceedings of the 25th Australian Computer-Human Interaction Conference on Augmentation, Application, Innovation, Collaboration - OzCHI '13* (2013).