VR-Interactions for Planning Planetary Swarm Exploration Missions in VaMEx-VTB

Rene Weller, Christoph Schröder, Jörn Teuber, Philipp Dittmann and Gabriel Zachmann
Computer Graphics and Virtual Reality
University of Bremen
Bibliothekstraße 5
28359 Bremen
Germany
{weller,schroeder.c,jteuber,dittmann,zach}@cs.uni-bremen.de

Abstract—**Virtual testbeds (VTBs) are essential for researchers and engineers during the planning, decision making, and testing phases of space missions because they are much faster and cost-effective than physical models or tests. Moreover, they allow to simulate the target conditions that are not available on earth for real-world tests, and it is possible to change or adjust mission parameters or target conditions on-the-fly. However, such highly specialized and flexible tools are often only available as desktop tools with limited visual feedback and a lack of usability. On the other hand, VR is predestinated for easy, natural interaction even in complex decision making and training scenarios, while simultaneously offering high fidelity visual feedback and immersion. We present a novel tool that combines the flexibility of virtual testbeds with an easy-to-use VR interface. To do that, we have extended a VTB for planetary exploration missions, the VaMEx-VTB (Valles Marineris Exploration-VTB), to support sophisticated virtual reality (VR) interactions. The VTB is based on the modern game engine 'Unreal Engine 4', which qualifies it for state-of-the-art rendering. Additionally, our system supports a wide variety of different hardware devices, including head-mounted displays (HMDs) and large projection powerwalls with different tracking and input methods.**

Our VR-VTB enables the users to investigate simulated sensor output and other mission parameters like lines-of-sight or ground formations for a swarm of different spacecraft, including autonomous ground vehicles, flying drones, a humanoid robot, and supporting orbiters. Moreover, the users can directly interact with the virtual environment to distract the swarm units or change environment parameters, like adding boulders or invoking sand storms. Until now, we have used our system for three different scenarios: a swarm-based exploration of the Valles Marineris on planet Mars, a test scenario of the same swarm units on the Canary Islands, and the autonomous building of a moon base. An expert review shows the general usability of our VR-VTB.

TABLE OF CONTENTS

1. INTRODUCTION 1
2. RELATED WORK 2
3. RECAP VAMEX-VTB 2
4. VR COMPONENTS 3
5. USE CASE: VAMEX 5
6. EVALUATION 7
7. CONCLUSIONS AND FUTURE WORKS 8
ACKNOWLEDGMENTS 9
REFERENCES 9
BIOGRAPHY 10

978-1-7281-7436-5/21/$31.00 ©2021 IEEE

1. INTRODUCTION

The goal of the VaMEx initiative, funded by the German Aerospace Center (DLR) as part of the Explorer Initiatives, is to investigate new technologies for the exploration of the Valles Marineris on planet Mars. This Martian region is the largest connected canyon landscape in the solar system, with a length of more than 4000 km. In these canyons' deep and protected areas, it is possible to find valuable resources like water or even signs of extraterrestrial life. However, due to the canyons' rugged nature, the development of new technologies is required to pursue exploration tasks in a robust, reliable, and autonomous manner.

The VaMEx initiative proposes to use a swarm of different autonomous robots that complement each other, including UAVs, wheeled ground vehicles, and walking robots, supported by a satellite in Mars orbit. In the first phase, we focus on developing concepts, the hardware, and algorithms, e.g., to allow flawless cooperation of the individual elements. A key feature for a mission consisting of a heterogeneous and autonomous swarm is a stable real-time communication system.

The validation and verification of such a complex mission, consisting of several interdisciplinary teams with many communication interfaces to exchange different kinds of data, is nontrivial. Real-world field tests for the individual parts are expensive, time-consuming, and not very realistic because the environments on earth differ significantly from the environmental conditions on Mars. The logistical effort in performing real-world field tests to evaluate swarm performance is considerable and out of reach in terms of financial resources.

In order to identify design gaps and inconsistencies at an early stage of mission planning, we have developed a *virtual* testbed (VaMEx-VTB) [1],[2] that simulates the communication interfaces, sensor input, and important physical properties of the local topography in a virtual environment. This allows the project partners to test their systems' software components before a real-world field test, diagnose flaws, and correct them already at the initial research stages. Moreover, our VTB allows rebuilding the Martian environmental conditions in a recreated 3D model of $40km^2$ of the Martian surface based on digital terrain models (DTMs) provided by HiRISE [3].

Major challenges when working with such virtual testbeds are the accessibility of the generated data produced during the simulation runs such as sensor or actuator data or, on a higher level, general information about the success of the prospection or exploration mission and the ability to easily change parameters in the scenario and observe the system's behavior. Virtual reality technologies offer a new way for interactions with VTBs. For instance, sophisticated immersive 3D visualizations allow new perspectives on the mission planning, and natural interaction metaphors offer easy access for changing environmental or vehicle parameters.

In this paper, we present several VR extensions to our VaMEx-VTB that realize precisely these features. This includes 3D visualizations for several sensor types, e.g., RGB(D) images, point clouds from lidar sensors, uncertainty visualizations to analyze the difference between observed and real data, etc. Moreover, we have implemented several tools to interact with the 3D environment and the simulated vehicles directly. For instance, the user can invoke a sandstorm to disturb the camera sensors, it is possible to place obstacles to stress the path planning, and we also offer the possibility to directly interact with the vehicles pushing them to simulate unexpected gusts of wind.

In order to achieve a natural interaction with the VTB, all data has to be processed in real-time. We present the integration of the different communication systems where we guarantee a real-time simulation of the data exchange between the individual VaMEx components. Moreover, it is often necessary to observe the scenario changes not alone but discuss them with colleagues or decision-makers. Therefore, we have implemented a multiplayer mode into our VTB. This allows the simultaneous inspection of the simulations in either VR via head-mounted displays or large projection walls - so-called powerwalls - but also via desktop in any combination.

We have asked several experts, both experts in VR who have experience in space-related VR systems and space engineers, for feedback. This also included standardized questionnaires to test the usability of our implementation.

We will start this paper with a brief overview of our verification and validation platform, the VaMEx-VTB, and will then focus on our VR extensions. As a use case serves the actual VaMEx initiative that we will briefly sketch before we finally present the results of our small expert centered user study in the results section.

2. Related Work

In general, virtual testbeds are software solutions that enable the validation and verification of arbitrary simulation models in user-definable virtual environments. They mainly help to reduce the need to build expensive physical prototypes by moving, especially early testing, into a pure virtual simulation environment. Consequently, VTBs reduce development time and cost significantly. Moreover, VTBs can be used as a common development and evaluation platform [4].

Virtual testbeds are already successfully used in many engineering fields like autonomous automotive development [5], physically-based automotive control [6], functional testing of smart ships [7] and supply chain planning [8].

Several virtual testbeds related to space and robotic applications already exist. For instance, Rossmann et al. have developed a virtual testbed with a database-driven architecture to minimizes the modeling effort required over a product's lifecycle in robotics [9], but there also exist testbeds for planetary exploration [10], [11].

A major challenge of virtual testbeds, especially in a space

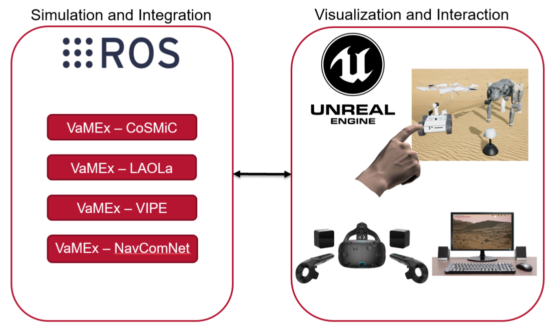


Figure 1: High-level overview of VaMEx-VTB structure.

context, is the realistic synthesization of sensor data. Consequently, there has been a lot of research in this direction, for instance, the real-time simulation of cameras that are not yet available as hardware products [12], other optical sensors [13] or more generalized frameworks that also support other kinds of sensors [14]. Beyond the sensors, a realistic physical behavior is essential, and thus, a research topic in the testbed domain [15].

While most virtual testbeds in the past concentrated on simulating single aspects of particular systems, there is today an ongoing trend to simulate larger systems and their environments [16]. This usually requires effective management of engineering processes from different domains [16]. Space agencies have a vital interest in supporting open-source frameworks for the verification and validation of space missions [17].

However, most virtual testbeds are still desktop programs. On the other hand, immersive virtual reality environments can offer interesting new spatial insights and accelerate the mission planning or review, especially the availability of affordable virtual reality devices that make this direction appealing. For instance, Sagardia et al. have presented a virtual reality simulator for teleoperation, a robotic arm with haptic feedback [18]. Even if such immersive visualization techniques are applied to the testbed, they often remain single user applications. However, there is also a trend to include multi-user capabilities: Garcia et al. [19] presented a collaborative visualization platform that allows even distributed groups of scientific and engineering experts to analyze and interpret combined datasets collectively. However, the main focus of this application was visualization. In our VR-extended version of VaMEx-VTB, we also include an underlying physical simulation and a physically plausible interaction with the environment and the simulated vehicles.

3. Recap VaMEx-VTB

The main goal of VaMEx-VTB is to serve as a virtual testbed for multimodal planetary space missions with a focus on swarm communication and navigation. Initially, it was developed as a common validation and verification platform for the VaMEx initiative (see Figure 5 for more details). As such, the VaMEx-VTB aims at

- simulating all relevant environmental aspects, including sensor synthesis, distribution of resources such as methane or water sources, collision detection,
- providing a highly detailed graphical feedback,

- and allowing extensions and exchangeability of the individual parts of the system.

In the following, we will briefly sketch some design details and discuss the features of VaMEx-VTB with respect to the requirements defined above.

General Design

Figure 1 provides a broad overview of the design of our virtual testbed. One core element of our VaMEx-VTB is a high-end visualization combined with the possibility of virtual reality (VR) interaction. We decided to use a state-of-the-art game engine, the Unreal Engine 4, that supports the most modern visualization effects and supports many VR hardware devices.

It is easy to import scenario-specific terrain data. For instance, we have manually created a $40km^2$ terrain of the Valles Marineris based on data available from NASA for our use case. However, the data's accuracy is limited; hence we included the possibility to add surface details easily. For instance, in the desktop version of our system, it is possible to simply paint the specific terrain type (e.g., sandy, rocky, etc.) directly on the surface. This also includes different texturing and even different physical properties for the simulation depending on the terrain type.

In order to connect the individual simulation components, such as vehicles or even individual sensors, we decided to support the widely used robot operating system (ROS) [20] because such components, especially in the navigation and communication domain, are predominantly implemented in this system. ROS is an open-source robotics middleware for software development in a robotic context. It provides services such as hardware abstraction but also low-level device control, and it supports a message-based communication protocol for the individual components. In general, we integrated and extended an interface to ROS to our VTB. More specifically, the VTB establishes a connection to a ROSbridge server [21] via a WebSocket to which the components can register to receive and send data. This fast ROS interface allows a simple modular design of practical relevance while maintaining the real-time capability of our VTB.

The system architecture was designed according to the *component-based software architecture*, which is also favored by the Unreal Engine. This makes it possible to create the swarm units as a plug-and-play system; sensors or even the unit's behavior can be easily assigned to them as a component.

High-Level Architecture

Our VTB consists of two separate parts (see Figure 1): first, an installation of ROS, containing all algorithms and software-components of the partners packaged in self-contained ROS-nodes and second, an Unreal Engine-project that contains the visualization, the interaction, and the simulation of the swarm units including the virtual sensors. We call this part the *simulation*.

The two parts are connected by `ROSbridge` which provides an interface for ROS that is accessible via a network connection. This means that the two parts of the VTB can be housed on two entirely different computer systems: for instance, the ROS system can be set up on a central computer accessible to all partners in the VaMEx-initiative, and every partner can run the Unreal-part on their computers to visualize and

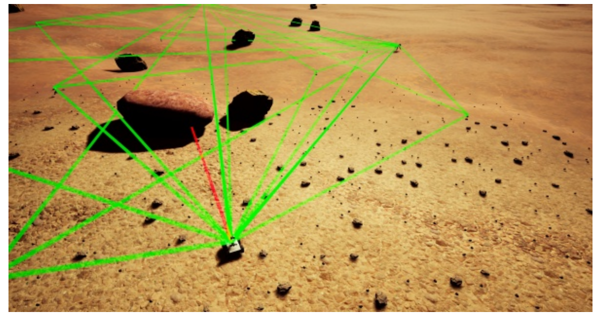


Figure 2: The line-of-sights show the visibility of the robots from communication beacons as part of the VaMEx-LAOLa project. The red line remarks an obstacle for a particular connection.

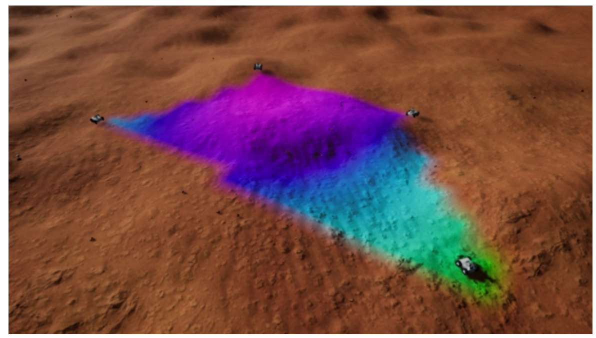


Figure 3: Visualization of the environmental process, i.e. basically methane measurements, as observed by the UGVs.

interact with this central ROS system. This is also important for security and supports intellectual property management. For development and testing purposes, a setup using a virtual machine running ROS on a Windows system running the simulation is the easiest option.

4. VR COMPONENTS

The VR extension of our VaMEx-VTB can visualize all implemented types of sensors directly in the stereoscopically projected 3D environment. Furthermore, we have implemented several types of interaction metaphors and tools to interact naturally with the simulated environment and the vehicles. However, the visualization must be adjusted to match the particular requirements for HMD displays and other immersive display technologies such as powerwalls.

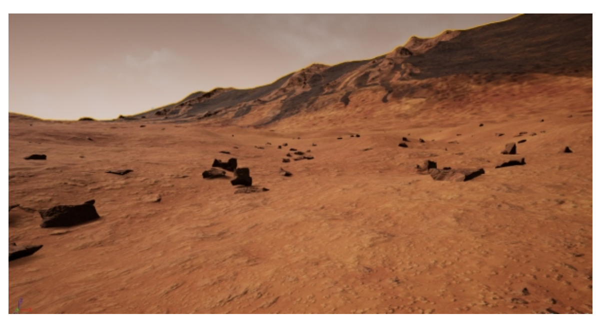


Figure 4: Highly detailed model of the terrain of Valles Marineris.

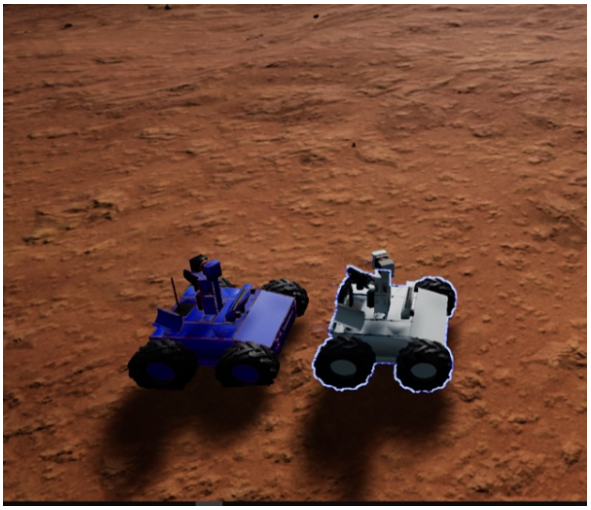


Figure 5: A blue ghost model shows the expected position of the real rover.

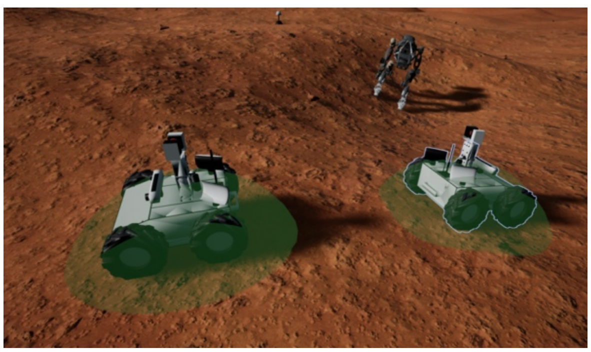


Figure 6: Ellipsoidal error visualizations help to identify uncertainties of the rover's pose.

Moreover, we have included a multiplayer mode that allows several users to share the same 3D environment and collaboratively discuss particular mission parameters.

Visualizations

The main visualization of our VR-VTB shows a 3D model of the planetary environment and a user specifiable arbitrary number of vehicles, e.g., UGVs, drones, UAVs, and hominid robots. For instance, for our use case (see Section 5), we have created a large exploration area of 40^2 km. The primary surface was derived directly from the HiRISE data that offers an accuracy of only 1m/pixel [3]. We further populated the surface with high-resolution textures and different physical surface properties. The textures were created manually. However, we offer a simple desktop tool to adjust the terrain by simply painting different pre-defined terrain types (sandy, rocky, etc.) and clutter on the surface. Our model also includes overhangs and caves (see Figure 4) and a semi-realistic sky-box based on different pictures taken by existing Mars rovers. The VTB simulates a real-time day-night-cycle in Mars time; however, the simulation can be accelerated by a factor of up to 4096. Additionally, for another use case, we have created a prototype of the Moon surface that is required for the EFRE innovation project *3D4Space* (see Figure 7). The paths of all swarm members can be visualized as lines in the 3D environment.

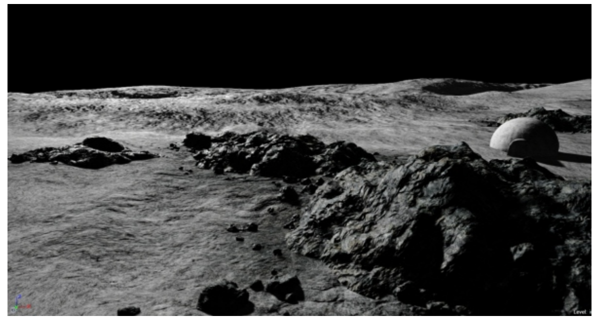


Figure 7: As a second use case for our VTB we have implemented a very preliminary mission to explore the moon.

Robots are fully physically simulated (e.g., the hominid robots or the UGVs can turn over) and directly react to the surface model. Moreover, our VTB supports the synthesis of Lidar and inertial measurement unit (IMU) data as well as RGB(D)-images (see Figure 18). The synthesis is implemented accurately: for instance, for the Lidar used by the CoSMiC rovers (see Section 5), we do not recycle RGBD-images because of the distortions that are typically produced by converting quadratic images to global-shaped scans. Instead, we use ray tests, including the simulation of the actuator rotation. RGBD-images support the same resolution and distortions as the original sensors.

Furthermore, it is possible to show the line-of-sights of the VaMEx-LAOLa beacons (see Figure 2) and to display environment color-coded environmental information gathered by the robots, e.g., the methane measured close to the ground by the rovers (see Figure 3). In order to evaluate the accuracy, it is often helpful to get visual feedback inside the VTB: our VaMEx-VTB supports, e.g., the colored ghosts that show differences of the expected pose by the path-planning of the vehicles to the ground truth (in the simulation)(see Figure 5). Incoming poses can be shown with color-coded ellipsoids (see Figure 6). Further uncertainty visualizations are implemented for different values like orb-slam [22] and IMUs. These additional visualizations (paths, ghosts, etc.) are invisible to the ROS-camera synthesis, i.e., the camera images that represent the sensor input for the vehicles do now show this data. The user can turn on and off the visibility of the visualizations in the VR environment.

The advantage of using VR for these kinds of visualizations is that they are directly embedded in the 3D environment and provide the user with a natural spatial understanding of the values.

Interactions

The most natural interaction metaphor is to touch objects directly. This is also supported by our VTB and the underlying physically-based simulation: e.g., the user can directly push the objects to simulate accidents (see Figure 8). In order to move around, we have implemented a simple teleporting mechanism because this locomotion metaphor causes relatively few symptoms of sickness [23]. The vehicles can be selected via a traditional ray selection metaphor. One reason is that all currently affordable VR devices like Oculus and the HTC Vive are delivered with 6-DOF pointing devices for which ray-based selection techniques are considered very intuitive, and they do not break any feeling of presence. Moreover, there is evidence that ray-based metaphors yield better user performance than, for instance, image-plane tech-

Figure 8: The user can physically interact in VR with our VTB e.g. knock over a rover to simulate an accident. (green screen image edited)

Figure 9: A VR menu can by opened by simply touching the wristwatch of the left virtual hand.

niques [24]. However, we have also included a menu structure that allows specifically selecting a vehicle and directly teleporting to its position. This is essential in such large areas to quickly travel to points of interest. Moreover, a menu is good for toggling several visualizations on and off.

Even when relying on a menu, we decided not to break the VR immersion totally. Consequently, we aimed at implementing the menu structure in a ”natural” way. To do that, the user calls the menu by pressing the wristwatch on his left hand by his right hand (see Figure 9). This enables a hologram projected above the watch to further dive into the menu structure (see Figure 10). The menu offers sub-menus for selecting specific vehicles, switching the visualizations, or enabling tools.

Currently, we have implemented several tools to interact with the environment directly. In detail, these are:

1. Stone Thrower: this tool enables the user to generate stones or rocks with user-definable sizes and place them in the environment using a simple ray metaphor.
2. Gravity Gun: this tool allows the user to remotely pick up objects of the environment or the vehicles and push or pull them towards a user-definable target using a ray metaphor (see Figure 11).
3. Sandstorm Generator: this tool allows the user to invoke a sand storm or dust devil and control its size and position (see Figure 18).

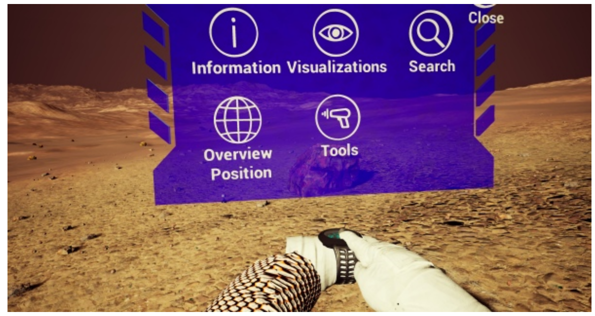


Figure 10: The main VR menu is presented as a hologram projected above the wristwatch.

Figure 11: The gravity gun tool allows users to remotely pick or push objects and vehicles in the environment, like the hominid robot in this example.

The tools can either be discarded via the menu or by a gesture that simulates putting the tool in a virtual backpack. All menu items contain a help button to obtain further information. Moreover, we have included a companion avatar that provides context-sensitive feedback to the user if required.

Multiplayer Mode

In addition to the interaction metaphors and tools, we have integrated a multiplayer mode that enables our VR-VTB to serve as a common platform for collective mission planning discussions. Our multiplayer mode supports different input and output devices, e.g., all HMDs that are compatible with the Unreal Engine 4 together with their respective controllers (see Figure 12), traditional desktop environments with classical mouse and keyboard input, and our powerwall with Optitrack optical tracking systems (see Figure 13). Other users are visible as virtual avatars in the scene. The systems can be combined in several ways, e.g., several powerwall users, several HMD users, and several desktop users may interact together in the same simulation, and the actions of the other users are reflected accordingly. This is one of the big advantages when basing the VTB on a modern game engine with support for multiplayer games: the synchronization of the data between the users can be easily integrated with the built-in multiplayer capabilities, with only little implementation overhead for special hardware like the powerwall.

5. Use Case: VaMEx

As a use case for our VR-VTB, we choose the VaMEx initiative. It consists mainly of four parts to explore the unknown terrain of the Valles Marineris: a swarm of unmanned aerial vehicles (UAVs) and wheeled rover that can cover

Figure 12: Interaction of two users in the simulated martian environment with HTC Vive HMDs. The view of the right user is also displayed on the screen behind the users.

Figure 13: Two users collaborating in front of our multi-user powerwall. The tracking is realized via an Optitrack system.

Figure 14: An overview of the VaMEx mission: autonomous wheeled rovers, UAVs, and hominid robots supported by a ground-based localization and navigation network and orbiters explore the Valles Marineris on Mars (not to scale).

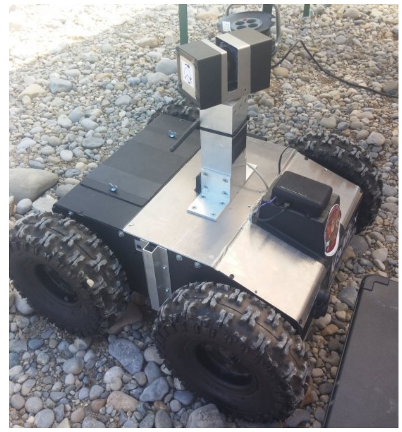


Figure 15: A physical model of the wheeled rover developed as part of VaMEx-CoSMiC.

Figure 16: A physical prototype of the UAV developed as part of VaMEx-VIPe.

large distances (VaMEx-CoSMiC), a hominid robot platform to explore also hardly reachable places like caves (VaMEx-VIPe), a ground-based localization and navigation network (VaMEx-LAOLa) and orbital support for global localization and communication (VaMEx-NavComNet) (see Figure 14).

VaMEx-CoSMiC

The VaMEx *Co*operative *S*warm Navigation, *Mi*ssion and *C*ontrol (VaMEx-CoSMiC) project focuses on the swarm exploration using autonomous rovers (see Fig. 15) and UAVs (see Fig. 16). The main goals are the development of efficient algorithms for surveying large areas without human supervision. The different vehicles are equipped with different sensor types, such as inertial sensors and monoscopic and stereoscopic cameras. Swarm communication is used for the distributed simultaneous localization and mapping (SLAM). Beyond the goal of using the sensor output for the navigation of the VaMEx-CoSMiC vehicles, it is used to create a map of the explored terrain and made available to other members of the VaMEx swarm.

VaMEx-VIPe

For an extensive exploration of the Valles Marineris, a robotic platform that can move within the fissured rock formations and navigate in caves and crevices that are unreachable by the rovers of VaMEx-CoSMiC is desired as part of the heteroge-

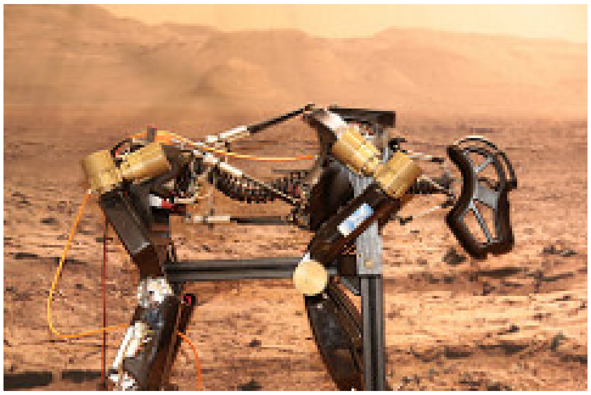


Figure 17: The physical model of the hominid robotic platform Charly developed in VaMEx-VIPe[2].

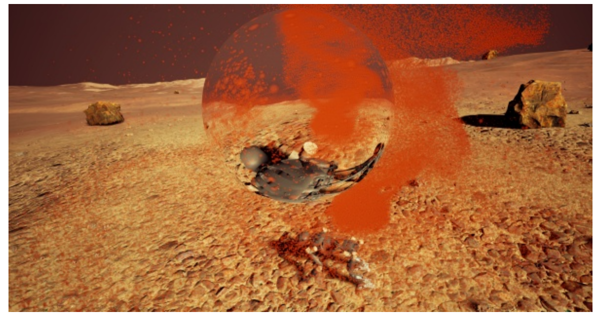


Figure 18: The third task in our experiment: the user has to invoke a sand storm and look at how it influences the image of a particular robot's RGB camera. The sensor image is displayed directly above the robot.

neous team. The hominid robot Charlie [25](see Figure 17, developed by DFKI, closes the remaining gap in the swarm (see Fig. 14).

VaMEx-LAOLa

The goal of the VaMEx-LAOLa (*L*okales *A*d-hoc *O*rtungs- und *La*ndesystem[3]) project is to provide systems for the communication between the individual members of the swarm, as well as enabling a localization to determine their positions relative to other swarm members. The local position is essential for the coordination of the swarm members and to find the way back to the lander. The accuracy of the local reference frame is higher than that of the global reference frame. The system is based on a set of beacons equipped with Frequency Modulated Continuous Wave (FMCW) secondary radar. For communication, the beacons contain a 2.4 GHz module additionally.

VaMEx-NavComNet

The VaMEx-NavComNet (*Nav*igation and *Com*munication *Net*work) has the concrete aim of serving as a science data, telemetry, and telecommand relay between earth and the in-situ users, as well as a cross-communication relay between users, but also providing a near real-time positioning system for surface, aerial and (potential future) space-based users. An ideal solution would consist of four satellites dispersed at different altitudes [26], ranging between 800 and 1200 km, and orbital inclinations up to 35 degrees, allowing for data exchange volumes of up to 300 Mbits per Sol (or Martian day). We are currently investigating more cost-efficient solutions consisting of a single satellite or nanosatellites.

6. EVALUATION

We have implemented our VTB with the Unreal Engine 4, mostly in C++, and we have included all necessary parts for our use case scenario. We decided to avoid a general broad user study with normal people to test the usability of our VTB because of the very specific target audience of such a relatively complex VR-VTB, i.e., mainly space engineers and space scientists. Instead, we have asked several experts in the field of scientific visualization and HCI in space and robotics context and an experienced space engineer for feedback.

[2]Courtesy of the German Research Center for Artificial Intelligence (DFKI), Germany
[3]German for: local ad-hoc localization and landing system

Due to the limited availability of such experts, especially in times of restricted traveling due to the pandemic, we provide mainly qualitative results instead of sophisticated quantitative analysis. However, we have also included the NASA TLX questionnaire to measure the cognitive load of our VTB [27]. We used the RAW version without the individual pairwise weighting because there is some evidence that this shorter version might increase the experimental validity [28]. Additionally, we used the System Usability Scale (SUS) [29] to evaluate general usability. However, the quantitative results have to be considered with care due to the small number of participants.

Experiment

We mainly concentrated on feedback on the user interface, the chosen interaction metaphors, and the visualizations' quality and usefulness. Hence, we only evaluated the single-user version. Due to the pandemic situation, it was not possible to invite all the experts into our lab to test the powerwall version of our VR-VTB; instead, we traveled to the experts carrying our complete setup, which includes the HMD version based on the HTC Vive Pro in combination with two traditional HTC Vive controllers.

In the beginning, we started with an open test to make the participants familiar with the general user interface and interactions of our VTB. This includes, e.g., that they get familiar with the teleporting system, the usage of the VR menu systems, and the tools. To save time, we gave direct oral feedback in case of questions instead of relying on our companion. This initial phase ended when the participants felt able to solve some simple tasks after about 10 minutes of training.

After the training, we asked the participants to solve three typical tasks that often appear in mission planning scenarios. We concentrated on tasks where the VR-VTB offers a real benefit over simple desktop- or even database-based applications. In detail, these are:

1. The first task was to check whether the vehicles are visible by the beacons during their movement. This task can be solved by beaming up to the top-view-position to get an overview (this can be called from the main menu) and select the line-of-sight visualization for the three kinds of vehicles from the visualization sub-menu.
2. The next task was to see how the second hominid robot (there were three robots of this kind in the test scenario) reacts if suddenly an obstacle appears in front of it. This task can be

solved by selecting the respective robot from the search menu to teleport to a close-by position, then selecting the stone-generation-tool from the tools sub-menu, and placing a stone in front of the robot.
3. The final task was to investigate how the third rover's RGB-camera (there were three rovers in total) is influenced in case of a sand storm. The solution requires the user again to teleport to the required rover, selecting the visualization of the RGB camera from the rover sub-menu, and finally, invoking a sand storm by selecting the according tool and placing the position of the sand storm in the range of the sensor (see Figure 18).

We did not measure the exact times that were required to finish the individual tasks. On average, it required about 20 minutes for all experts to master all three challenges.

Participants

As mentioned above, we asked six experts in different related research fields for feedback. Four of them were male, two female, five are computer scientists, one of them in combination with expert knowledge in HCI, the sixth expert has a background in space engineering. Their age reaches from 33 to 54 years (M=41.8, SD=1.7), and they have experience in their domain between 6-20 years (M=13.8, SD=5.4). Two participants rated their knowledge in using space simulations as expert knowledge, two as above average, and two as average on a 4-point scale reaching from beginner to expert. All participants have experience in some kind of space simulation software, mainly on their own development platform, and in other programs such as ParaView or Virtual Satellite or more database-related software for space simulations. Four of the participants rated their VR experience level as expert, one as above average, and one as beginner on the same scale as for the experience with space simulations.

To summarize, all of our participants can call on many years of experience in relevant fields for appropriate expert feedback. Currently, our VR-VTB is optimized for right-handed users, mainly because of the clock's placement to invoke the menus on the left hand and because of the placement of the tools in the users' right hand. However, it is easy to mirror the models of the hands to support also left-handed users. Fortunately, the dominant hand of all of our experts is their right hand.

Results

The NASA TLX RAW measures a task's workload with six subjective subscales on a 20 point scale. Higher values on a particular scale, mean a higher demand for this subscale. Regarding the cognitive load measures by NASA TLX RAW, the VR interactions for the tasks achieved average results with respect to mental demand (M=11.8, SD=3.2), physical demand (M=9.5, SD=2.6), performance (M=13.8, SD=4.7), and effort (M=10.8, SD=2.8). Only the temporal demand (M=6.8, SD=5.0) and the frustration (M=6.8, SD=3.6) were rated much lower and, thus, better. This indicates that, especially for such relatively demanding tasks that benefit from a natural interaction and direct visual feedback, VR seems to be a good direction to investigate further. However, there is still room for improvements, e.g., lowering the physical demands that are typically higher in VR environments than desktop interactions (see Figure 19).

Regarding the usability of our VR-VTB, we achieved a good score (M=67.9, SD=8.5). The total scale for SUS reaches from 0-100, where 100 means perfect usability, and in general, systems scored 68 or higher are considered above average and thus, usable (see Figure 20). However, regardless of all the participants being experts, this rating and all other quantitative measures we have presented here have to be considered as preliminary results due to the small sample size and the relatively high spread of the values reflected by the standard deviation.

Besides the standardized rating from the questionnaires, we also recorded qualitative feedback from the experts in interviews. As a primary reason for the questionnaires' average results, they all mentioned the limited training time that we had to set due to time constraints. They all reported that remembering the up to three levels of depth in the menu hierarchy requires significantly more training. Also, the number of options in the sensor menu was mentioned as too large by one expert (see Figure 21). Additionally, four out of the six experts complained about the tasks' description and a missing motivation. This is something we will improve in future evaluations. Two experts with more visual feedback for the ray selection metaphors, e.g., when selecting a vehicle but also during the sand storms' placement tasks and the stone obstacles. One expert recognized a performance lag when evoking a sand storm that slightly reduces the frames per second because of the high particle count. Other optimizations could be an improved top-view position that currently requires looking through the floor and using a "sprint" teleportation instead of the "blink" teleportation that we implemented. "Blink" teleportation typically fades momentarily black for the movement, while "sprint" teleportation uses a fast animation between the positions. However, we did not include a simulator sickness questionnaire to investigate which version performs best in space simulations, but we will consider this for future developments. However, the metaphor to remove the tool by performing a gesture to put it into a virtual backpack achieved a good rating by the experts. Also, the idea to use a clock for the menu source was well-received (even though it currently does not show the Martian time). Especially the space engineer mentioned that such VR-enabled VTBs could improve mission planning enormously in the future. However, regarding the low number of participants, our user study should be considered a preliminary study.

7. Conclusions and Future Works

We have presented a VR extension of our VaMEx-VTB for planetary swarm-based exploration missions, including novel visualizations of several kinds of data in VR, VR interaction tools, and metaphors for such space simulations like evoking sand storms or placing obstacles, and finally, we have implemented a sophisticated multiplayer mode. As a first test scenario, we have incorporated the VaMEx initiative, a swarm-based exploration of the Valles Marineris on Mars with three different kinds of ground vehicles and support by orbiting satellites. Our VR-VTB supports the simulation of several sensor types in real-time, enhanced visualization modes, including different camera views, point clouds, and uncertainty measures. Finally, we gathered qualitative feedback from a diverse group of relevant experts. The resulting SUS score indicates the good usability of our system.

We are confident that the idea of extending space simulators by immersive visualizations and natural interactions in VR will improve mission planning in the space domain enormously. While investigating sensor data in real-time and interacting with the environment or the vehicles, the direct

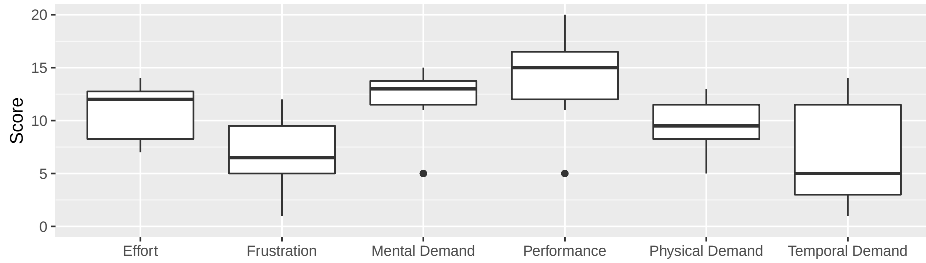


Figure 19: The results for the individual items of the NASA TLX RAW questionnaire. The experts recognized an average mental demand for all questions.

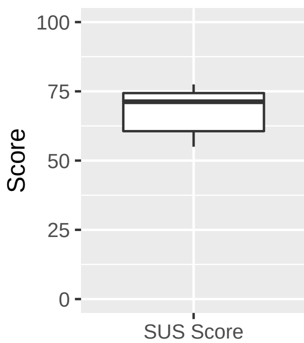


Figure 20: The SUS score derived from the SUS questionnaire. Overall, the experts rated our VR-VTB as usable.

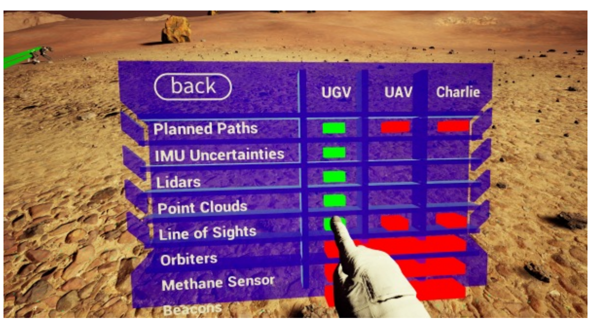


Figure 21: Some sub menus of our VR menu system have a lot of options, e.g. the menu for selecting visualizations. Actually, finding a good trade-off between single menu complexity or hiding the complexity in the depth of the menu structure is challenging.

visual feedback will give a much better understanding of the missions and their constraints. The modular and future-proof design of our VaMEx-VTB qualifies it to serve as a testing platform for other space projects, especially for planetary surface exploration scenarios. The first application is already planned for 3D4Space. Additionally, we also want to include more advanced features like dust- and sunflare-effects for the ROS-camera model, a basic weather simulation of Mars. Moreover, we plan to enhance the basic physics simulation to support also effects such as battery drain, weather effects, or wheel tracks. Finally, we plan to enhance the VR features by the expert feedback, e.g., by including a rewind option to rewind the time and restart the simulation with slightly changed parameters.

ACKNOWLEDGMENTS

The authors thank the DLR Space Administration and all VaMEx partners for supporting this project. This project was partly funded by the German Federal Ministry for Economic Affairs and Energy (BMWi) under grants 50NA1712 and 50NA1916.

REFERENCES

[1] J. Teuber, R. Weller, L. Buinhas, D. Kühn, P. Dittmann, A. Srinivas, F. Kirchner, R. Förstner, O. Funke, and G. Zachmann, "Virtual validation and verification of the vamex initiative," in *Proc. of the Interplanetary Probe Workshop (IPPW) 2019*, Oxford, England, 7 2019, poster.

[2] J. Teuber, R. Weller, L. Buinhas, D. Kuehn, P. Dittmann, A. Srinivas, F. Kirchner, R. Foerstner, O. Funke, and G. Zachmann, "Vamex-vtb – a modular virtual testbed for multimodal autonomous planetary missions," in *Proceeding of the 70th International Astronautical Congress. International Astronautical Congress (IAC-2019), October 21-25, Washington DC, DC, United States*, 2019.

[3] A. S. McEwen, E. M. Eliason, J. W. Bergstrom, N. T. Bridges, C. J. Hansen, W. A. Delamere, J. A. Grant, V. C. Gulick, K. E. Herkenhoff, L. Keszthelyi *et al.*, "Mars reconnaissance orbiter's high resolution imaging science experiment (hirise)," *Journal of Geophysical Research: Planets*, vol. 112, no. E5, 2007.

[4] P. Lange, R. Weller, and G. Zachmann, "Multi agent system optimization in virtual vehicle testbeds," in *EAI SIMUtools*. Athens, Greece: EAI, Aug. 2015.

[5] F. Dion, J. Oh, and R. Robinson, "Virtual testbed for assessing probe vehicle data in intellidrive systems," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 3, pp. 635–644, Sep. 2011.

[6] D. Davis and D. Brutzman, "The autonomous unmanned vehicle workbench: Mission planning, mission rehearsal, and mission repay tool for physics-based x3d visualization," *14th International Symposium on Unmanned Untethered Submersible Technology (UUST)*, 2005.

[7] F. Yang, J. Liu, S. Li, and F. Ma, "Virtual-real interaction tests for functional testing of smart ships," in *Proccedings of the 30th International Ocean and Polar Engineering Conference*, 08 2020.

[8] S. Chick, P. Sanchez, D. Ferrin, and D. Morrice, "A simulation test bed for production and supply chain

modeling," in *Proceedings of the 2003 Winter Simulation Conference, 2003.*, vol. 2, Dec 2003, pp. 1174–1182 vol.2.

[9] J. Rossmann and B. Sommer, "The virtual testbed: Latest virtual reality technologies for space robotic applications," in *International Symposium on Artificial Intelligence, Robotics and Automation in Space*, 2008, pp. 133–138.

[10] J. Rossmann and B. Sommer, "The virtual testbed: Latest virtual reality technologies for space robotic operations," *9th International Symposium on Artificial Intelligence, Robotics and Automationin Space (i-SAIRAS)*, 2005.

[11] J. Rossmann, B. Sondermann, and M. Emde, "Virtual testbeds for planetary exploration: The self-localization aspect," *11th Symposium on Advanced Space Technologies in Robotics and Automation (ASTRA)*, 03 2019.

[12] T. Stel and J. Roßmann, "A virtual reality testbed for camera simulation in aerospace applications," in *2015 3rd International Conference on Artificial Intelligence, Modelling and Simulation (AIMS)*, 2015, pp. 129–134.

[13] J. Rossmann, N. Hempe, M. Emde, and T. Steil, "A real-time optical sensor simulation framework for development and testing of industrial and mobile robot applications," in *ROBOTIK 2012; 7th German Conference on Robotics*, 2012, pp. 1–6.

[14] J. Rossmann, M. Emde, B. Sondermann, and N. Hempe, "Advanced sensor simulation in virtual testbeds: A cost-efficient way to develop and verify space applications," in *AIAA SPACE 2011 Conference & Exposition*, 2011. [Online]. Available: https://arc.aiaa.org/doi/abs/10.2514/6.2011-7153

[15] J. Rossmann and M. Schluse, "Virtual robotic testbeds: A foundation for e-robotics in space, in industry - and in the woods," in *2011 Developments in E-systems Engineering*, 2011, pp. 496–501.

[16] G. Jochmann, F. Blümel, O. Stern, and J. Roßmann, "The virtual space robotics testbed: Comprehensive means for the development and evaluation of components for robotic exploration missions," *KI-Künstliche Intelligenz*, vol. 28, no. 2, pp. 85–92, 2014.

[17] D. Seider, M. Litz, A. Schreiber, P. M. Fischer, and A. Gerndt, "Open source software framework for applications in aeronautics and space," in *2012 IEEE Aerospace Conference*, 2012, pp. 1–11.

[18] M. Sagardia, K. Hertkorn, T. Hulin, S. Schätzle, R. Wolff, J. Hummel, J. Dodiya, and A. Gerndt, "Vr-oos: The dlr's virtual reality simulator for telerobotic on-orbit servicing with haptic feedback," in *2015 IEEE Aerospace Conference*, 2015, pp. 1–17.

[19] A. S. García, D. J. Roberts, T. Fernando, C. Bar, R. Wolff, J. Dodiya, W. Engelke, and A. Gerndt, "A collaborative workspace architecture for strengthening collaboration among space scientists," in *2015 IEEE Aerospace Conference*, 2015, pp. 1–12.

[20] M. Quigley, B. Gerkey, K. Conley, J. Faust, T. Foote, J. Leibs, E. Berger, R. Wheeler, and A. Ng, "Ros: an open-source robot operating system," in *Proc. of the IEEE Intl. Conf. on Robotics and Automation (ICRA) Workshop on Open Source Robotics*, Kobe, Japan, May 2009.

[21] C. Crick, G. Jay, S. Osentoski, B. Pitzer, and O. C. Jenkins, "Rosbridge: Ros for non-ros users," in *ISRR*, 2011.

[22] R. Mur-Artal and J. D. Tardós, "ORB-SLAM2: an open-source SLAM system for monocular, stereo and RGB-D cameras," *CoRR*, vol. abs/1610.06475, 2016. [Online]. Available: http://arxiv.org/abs/1610.06475

[23] P. J. Líndal, K. R. Jóhannsdóttir, U. Kristjánsson, N. Lensing, A. Stühmeier, A. Wohlan, and H. H. Vilhjálmsson, "Comparison of teleportation and fixed track driving in vr," in *2018 10th International Conference on Virtual Worlds and Games for Serious Applications (VS-Games)*, 2018, pp. 1–7.

[24] D. A. Bowman and L. F. Hodges, "Formalizing the design, evaluation, and application of interaction techniques for immersive virtual environments," *Journal of Visual Languages & Computing*, vol. 10, no. 1, pp. 37 – 53, 1999.

[25] D. Kuehn, M. Schilling, T. Stark, M. Zenzes, and F. Kirchner, "System design and field testing of the hominid robot charlie," *Journal of Field Robotics*, vol. 34, no. Issue 4, pp. 666–703, 4 2016.

[26] L. Buinhas, G. Peytaví, and R. Förstner, "Navigation and Communication Network for the Valles Marineris Explorer (VaMEx)," in *Proceedings of the 69th International Astronautical Congress (IAC)*, 2018.

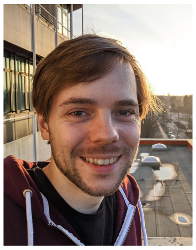
[27] S. G. Hart and L. E. Staveland, "Development of nasa-tlx (task load index): Results of empirical and theoretical research," *Human mental workload*, vol. 1, no. 3, pp. 139–183, 1988.

[28] E. A. Bustamante and R. D. Spain, "Measurement invariance of the nasa tlx," *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, vol. 52, no. 19, pp. 1522–1526, 2008. [Online]. Available: https://doi.org/10.1177/154193120805201946

[29] J. Brooke, *"SUS-A quick and dirty usability scale." Usability evaluation in industry*. CRC Press, June 1996, iSBN: 9780748404605. [Online]. Available: https://www.crcpress.com/product/isbn/9780748404605

BIOGRAPHY

Rene Weller *is a postdoctoral research assistant at the Computer Graphics Group at the University of Bremen, where he also received his Ph.D. degree in computer science in 2012. He is working in the field of computer graphics and virtual reality with a specialization in collision detection algorithms.*



Christoph Schröder *received his MSc degree in informatics from the University of Lübeck, Germany, in 2015. He is currently pursuing a Ph.D. at the Computer Graphics and Virtual Reality group of the University of Bremen. His research interests include VR projection techniques and machine learning on eye-tracking as well as 3D data.*

Jörn Teuber *graduated with a Masters degree in computer science from the Technical University of Clausthal in Germany and is currently pursuing a Ph.D. at the University of Bremen. His research interests are computer vision and graphics, machine learning, concurrent algorithms, and virtual testbeds and how to combine all of these. He is working on his Ph.D.*

Intentionally left blank for privacy reasons.

Philipp Dittmann *finished his MSc degree in informatics at the University of Bremen, Germany, in 2015. He is working on his Ph.D. at the Computer Graphics and Virtual Reality group at the same place. His research fields are VTBs, Remote Driving, and Acceleration Data Structures in VR.*

Gabriel Zachmann *is a full professor with the computer science department at the University of Bremen, Germany. He is the head of the visual computing group, focusing mostly on computer graphics and virtual reality. Prior to that, he was a professor for 7 years at the Technical University of Clausthal, Germany, where he established a research group for computer graphics. He has worked in virtual reality, geometric computing, and visual computing in general for 25 years.*