IAC–19–IAC-19,D1,4A,6,x53973

VaMEx-VTB – A Modular Virtual Testbed for Multimodal Autonomous Planetary Missions

Jörn Teuber[1], Rene Weller[1], Luisa Buinhas[2], Daniel Kühn[3], Philipp Dittmann[1], Abhishek Srinivas[1], Frank Kirchner[1,3], Roger Förstner[2], Oliver Funke[4], Gabriel Zachmann[1]

[1] University of Bremen, Germany
[2] Bundeswehr University Munich, Germany
[3] German Research Center for Artificial Intelligence, Germany
[4] German Aerospace Center (DLR), Space Administration, Germany

The "VaMEx - Valles Marineris Explorer" initiative is part of the DLR Explorer Initiatives. As such it is an interdisciplinary research program funded by the DLR Space Administration aimed at developing new concepts, algorithms and hardware for swarm-based exploration of the Valles Marineris on Mars. This includes a hominid robotic platform (project VaMEx-VIPe), autonomous swarm navigation including ground vehicles and UAVs (project VaMEx-CoSMiC) that rely on a local positioning and landing system(project VaMEx-LAOLa) and orbital support (VaMEx-NavComNet) serving as a science data, telemetry and telecommand relay between Earth and the in-situ elements and providing near real-time position updates to the other elements.Real validation and verification tests for such complex navigation and exploration systems are difficult, expensive and time-consuming because they require the availability of hardware as well as realistic environments.In this paper, we present VaMEx-VTB, a virtual testbed (VTB) that has been developed in the project of the same name. The VaMEx-VTB enables the verification and validation of such large and complex interdisciplinary research projects during very early phases. The basic idea of VaMEx-VTB is toprovide a common platform for all modules in combination with a sophisticated user definable computer simulation. Consequently, it can serve as an integration and discussion hub during the development process, thereby reducing expensive and time-consuming physical testing. The VTB allows users to configure various aspects of the test scenarios and the test environment, such as physical parameters,atmospheric conditions, or terrain features. This is essential especially for extraterrestrial planetary missions that are difficult to reconstruct on earth. Finally, a sophisticated graphical feedback, based ona state-of-the-art game engine, allows an easy and direct interaction of the engineers with the test case sin the VTB.As a first use case, the VTB is adopted to serve as testing and integration platform for the aforementioned projects of the VaMEx initiative. Our modular design based on ROS supports consistent data access for all components. So far, we have implemented a realistic simulation of the relevant environmental parameters and created an adjustable model of the Valles Marineris terrain, based on the HiRISE data.Additionally, the VTB synthesizes realistic sensor input for several algorithms running on the swarm elements. The modular design concept also qualifies the VTB to serve as a testing platform for other extraterrestrial missions in the future.

keywords: Virtual Testbed, Validation & Verification, Virtual Reality, Simulation, Swarm Exploration, Navigation

1. Introduction

The goal of the VaMEx initiative, funded by the German Aerospace Center (DLR) as part of the Explorer Initiatives, is the investigation of new technologies for the exploration of the Valles Marineris on Mars. This Martian region is the largest connected canyon landscape in the solar system with a length of more than 4000 km. In the deep and protected areas of these canyons, it is possible to find valuable resources like water or even signs of extraterrestrial life. However, due to the ragged nature of the canyons, the development of new technologies is required in order 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 a first phase, we focus on the development of concepts, the hardware but also algorithms, e.g. to allow a flawless cooperation of the individual elements. A key feature for a mission consisting of an 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 already 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 the financial resources.

In order to identify design gaps and inconsistencies at an early stage of mission planning we have developed a novel *virtual* testbed (VaMEx-VTB) 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 the software components of their systems before a real-world field test, diagnose flaws and correct them already at initial research stages. Moreover, our VTB allows to rebuild the Martian environmental conditions and it supports a user-adjustable modification of the terrain.

A main challenge for VaMEx-VTB are the large amounts of data that have to be handled. Actually, we have recreated a 3D model of $40km^2$ of the Martian surface based on digital terrain models (DTMs) provided by HiRise.[13] Moreover, we will present the integration of the different communication systems where we guarantee a real-time simulation of the data exchange between the individual VaMEx components. Finally, the VaMEx-VTB is not only a desktop application, but the integration of virtual reality technologies allows the engineers a more natural and immersive interaction with the systems.

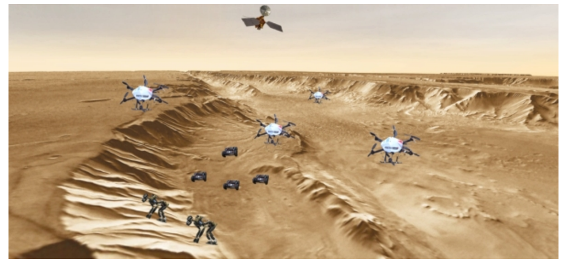

Fig. 1: An overview of the VaMEx mission: autonomous wheeled rovers, UAVs, and hominid robots supported by a ground-based localization and navigation network and orbiters explore the Valles Marineris on Mars.

We will start this paper with a brief overview on the individual parts of the VaMEx initiative, then focus on our novel verification and validation platform, VaMEx-VTB, and finally, shortly highlight some of the main features of VaMEx-VTB in the Results Section.

2. VaMEx Overview

The VaMEx initiative 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 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 1).

2.1 *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. 2) and UAVs (see Fig. 3). 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

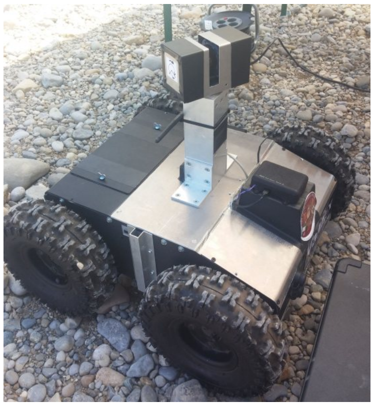

Fig. 2: A physical model of the wheeled rover developed as part ofVaMEx-CoSMiC.

Fig. 3: A physical prototype of the UAV developed as part of VaMEx-VIPE.

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.

2.2 *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 heterogeneous team. The hominid robot Charlie[10](see Figure 4, developed by DFKI, closes the remaining gap in the swarm (see Fig. 1).

To put the project into practice efficiently and cost-effectively, expertise and hardware built in previous projects from different areas, such as deep-sea

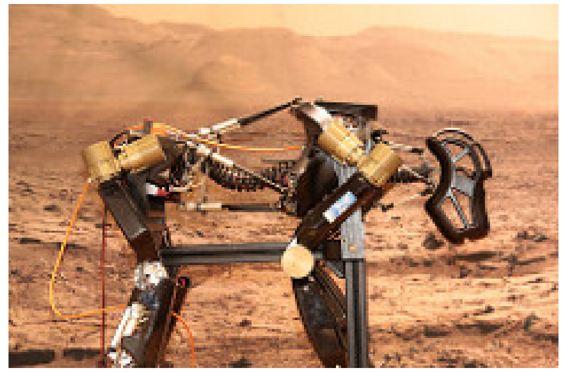

Fig. 4: The physical model of the hominid robotic platform Charly developed in VaMEx-VIPE.

robotics,[2,9,12] was used. Charlie, a four-legged walking robot, is well suited to overcome difficult terrain due to its light and highly integrated construction and agility. In addition, the tactile sensors are powerful tools in many applications.[1] In order to keep the weight of the robot platform low (important for the agility and transport costs of the system), only light sensors were used.

Visual navigation is a very suitable technology based on light, passive sensors, which allows a reliable position determination due to the high redundancy but usually requires a robust semantic representation of environmental objects and features like described in.[8] In contrast to radio-based positioning, no (visual) connection to other swarm participants is necessary. The position determination based on continuous visual odometry using a stereo camera, as it is used for the rovers and flight systems, can only be applied to a limited extent to Charlie. Especially in areas with low brightness, the exposure times would be too long or would require a continuous, and thus resource-intensive, illumination.

Particularly when overcoming a rugged terrain, it is crucial to place the legs of the walking robot precisely on stable surfaces.[17] In order to make this possible, a proprioceptive approach was researched that uses tactile sensors to record body position and movement in space and converts them into position information. This is a prerequisite for motion planning and reactive motion control, which makes it possible to overcome obstacles. By merging tactile data with visually-perceived surface structures such as edges and gaps, these two technologies complement each other to form a very promising approach for reliably pursuing exploration tasks in topographically-challenging areas.

The reactive motion control in Charlie was extended by further behavior modules, so that a safe locomotion over leveled and uneven ground as well as the overcoming of obstacles with the robot could be shown. In addition, an algorithm for optimal foot placement was developed. This adaptive foot-placement algorithm makes it possible to find an optimal foot contact point for each leg either between or including various obstacles with the help of a local map. It has to be mentioned that this is not a purely planning-based control of the robot. The reactive walk control is maintained, the planning level is only allowed to write offsets on the respective walking pattern of the different legs. This procedure takes place in real time and extends the robot's mobility in that it is not necessary to stop on uneven ground or in front of obstacles in order to plan the next steps. Even if the ground does not behave as expected (e.g. due to the flexibility of an obstacle, where a contact between foot and obstacle has been planned into the step cycle), the robot is able to continue its locomotion stably due to the permanently active reactive control level.

2.3 *VaMEx-LAOLa*

The goal of the VaMEx-LAOLa (*L*okales *A*d-hoc *O*rtungs- und *La*ndesystem*) 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 important 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 that are equipped with Frequency Modulated Continuous Wave (FMCW) secondary radar. For the communication, the beacons contain additionally a 2.4 GHz module.

2.4 *VaMEx-NavComNet*

The VaMEx-NavComNet (*Nav*igation and *Com*munication *Net*work) has the concrete aim of serving as 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,[3] 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 nano satellites.

*German for: local ad-hoc localization and landing system

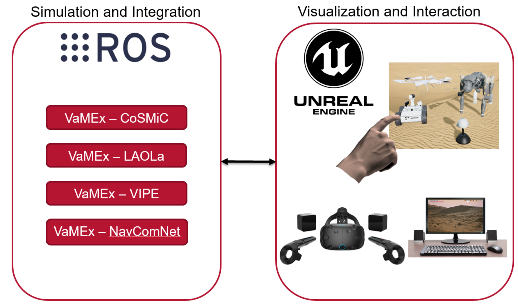

Fig. 5: High-level overview of VaMEx-VTB structure.

3. VaMEx-VTB

Virtual testbeds are already successfully used in fields like autonomous automotive development,[7] physically-based automotive control,[6] supply chain planning[4] but also planetary exploration.[15,16] 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.[11]

The main goals of VaMEx-VTB are:

- to serve as a common validation and verification platform for the VaMEx initiative,
- to simulate all relevant environmental aspects, including sensor synthesis, distribution of resources, collision detection,
- to deliver immersive natural interaction with the system and to provide a highly detailed graphical feedback,
- and to allow 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.

3.1 *General Design*

Figure. 5 provides a broad overview on the design of our virtual testbed. One core element of our VaMEx-VTB is a high-end visualization in combination with the possibility of virtual reality (VR) interaction. We decided to use a state-of-the-art game engine, the Unreal 4 Engine, that supports the most modern visualization effects and has an integration for a large amount of VR hardware devices.

Moreover, we have manually created a $40km^2$ terrain of the Valles Marineris based on data available from the NASA. However, the accuracy of the data is limited, hence we included the possibility to easily add surface details. For instance, our systems supports simply painting the specific terrain type (e.g. sandy, rocky, etc) directly on the surface. This includes also different texturing and even different physical properties for the simulation depending on the terrain type.

In order to connect the individual VaMEx components, which are predominantly implemented in the widely used robot operating system (ROS),[14] to the VTB we integrated and extended an interface to ROS. More specifically, the VTB establishes a connection to a ROSbridge server[5] 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 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, which each sensor or even behavior as a component that can easily be attached to it.

3.2 *High-Level Architecture*

Basically, our VTB consists of two separate parts (see Figure 5): 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 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 which is running the simulation is the easiest option though.

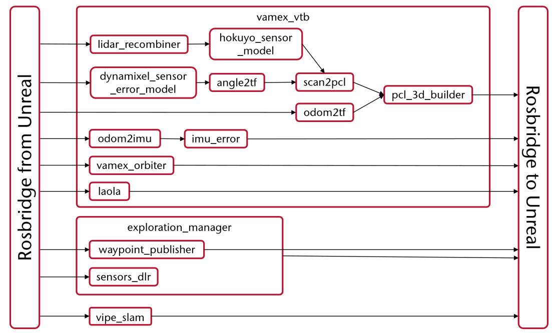

Fig. 6: Components and data-flow in the ROS-system.

3.3 *Components in ROS*

Figure 6 shows all ROS-nodes and their interaction with the other nodes, including the simulation via the `ROSbridge`. All rectangles in the center represent ROS-nodes, which can also contain other ROS-nodes, like the `vamex_vtb-node`. The `vamex_vtb-node` contains all nodes that can are accessible from the Unreal part. That includes nodes to convert the depth-measurements sent by the virtual Lidar into point clouds (all nodes up to and including `pcl_3d_builder`), a node that converts ground-truth odometry send by the simulation into imu-data and adds some noise to it, a node containing the SPICE-kernels simulating the orbits of the orbiters, and a node containing the algorithms developed in VaMEx-LAOLa.

The `exploration_manager-node` contains the exploration strategy developed by VaMEx-CoSMiC, which has to be started after the simulation. It supplies an environmental process for the swarm-units to explore and the simulation to visualize, and goal-points for the swarm-units to make measurements at.

The bottom-most node, `vipe_slam`, contains the ORB-SLAM2 algorithm that is used in VaMEx-VIPE with a few modifications. This is not included in the `vamex_vtb-node` even though it needs to be started before the simulation as it needs to be run as its own application (it is not a ROS-node in the sense that it cannot be started by ROS, but it communicates with

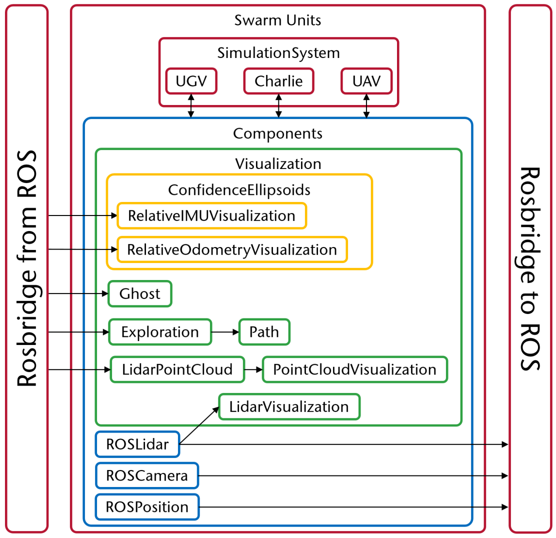

Fig. 7: Actors, Components and data-flow of the swarm-units in the simulation. Data flows from left to right along the arrows. The swarm-units (UGV, Charlie and UAV) use components to add functionality.

ROS).

3.4 *Architecture of the Simulation*

The simulation can be split into three parts: the swarm-units, the support-units and the environment (see Figure 7). In the following sections we will detail the architecture of the swarm-units and the support-units. The environment, i.e. the terrain, rocks and other features of the landscape, and the sky, is handled completely by the Unreal Engine and is therefore not part of the architecture itself.

3.4.1 *Architecture of the Swarm-Units*

The swarm-units and their components are designed with re-usability in mind. Most of a swarm-unit's abilities are encapsulated in components which can be programmed once and then used on any swarm-unit. This results in a plug-and-play-like architecture where new functionality, like virtual sensors or visualizations, can be added, removed or just moved on a robot whenever needed.

The `SimulationSystem` is implemented as a `GameInstance`, which is a central component of the Unreal Engine. When the simulation is started, the Unreal Engine creates an instance of the `SimulationSystem`, which in turn spawns all swarm-units and arranges them around the origin of the scene. The visualization-components (that are located in the green rectangle of Figure 7) have a common interface to make it easy to hide and show them and make them invisible to all `ROSCamera`-components, which simulate certain kinds of cameras and should not picture, for example, the point-clouds generated by the ROSLidar-component and shown by the `PointCloudVisualization`-component.

The `ROSLidar`-, `ROSCamera`-, and `ROSPosition`-components synthesize ground-truth sensor outputs, specifically of a Lidar, a RGB(D)-camera, and an odometry-sensor respectively. The Lidar can also visualize the plane it is currently scanning, which is encapsulated in the `LidarVisualization`-component.

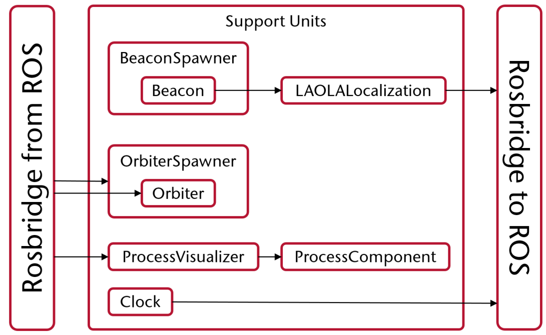

Fig. 8: Actors and components of the support-units.

3.4.2 *Architecture of Support-Units*

The support-units fulfill a variety of tasks within the VTB and the VaMEx-swarm in general. Instead of one class spawning all these like the swarm-units, each type of support-unit has its own `spawner-class`. This provides more flexibility and encapsulates the individual subsystems better. The beacons, which are spawned according to the settings in the `BeaconSpawner`, are used by LAOLa as static landmarks for their localization algorithms. They are used by the `LAOLALocalization`-actor to generate the messages that are send to the LAOLa-algorithms in ROS.

The `OrbiterSpawner`-actor spawns a set of `Orbiter`-actors with settings that can be specified in the spawner, for example a scaling factor. The orbiters then get their position from their respective ROS-topic. The `ProcessVisualizer`-actor uses two `ProcessComponents` to visualize the ground-truth and estimated environmental process as sent by the `ExplorationManager` in ROS.

Finally, the `Clock`-actor is a very simple class that sends out the simulations "official" timestamp every frame. This timestamp needs to be used by

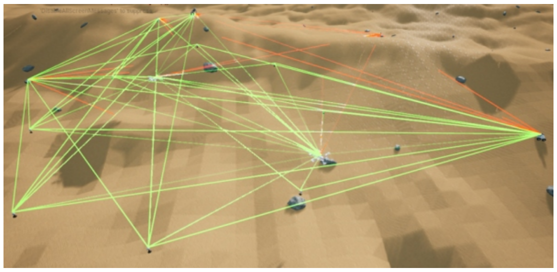

Fig. 9: The line-of-sights connecting the VaMEx-LAOLa beacons.

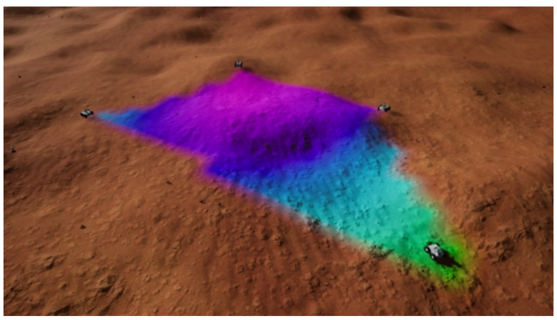

Fig. 10: Visualization of the process to be measured by the swarm in VaMEx-VTB.

ROS-nodes that, for their measurements, rely on the current time, for example the `vamex_orbiter-node`. This makes it possible to accelerate the time (time-warping) in the simulation and see the orbiters following their orbits much quicker.

4. Results

We have implemented our VTB with the Unreal Engine 4, mostly in C++. In this section we will present you some of the highlights of our actual implementation for the validation and verification of the VaMEx swarm algorithms. Our VTB offers interfaces for all VaMEx sub-projects: VaMEx-LAOLa, VaMEx-VIPE, and VaMEx-CoSMiC. Moreover, our VTB, especially the VR port, serves as common platform for the discussion of future developments.

In our VTB, a user specified arbitrary number of vehicles, i.e. UGVs, drones, UAVs and Charlie robots can be spawned. Paths for all swarm members can be easily set via the *Exploration Manager* provided by the DLR IKN. We currently use the build-in path-finding algorithm from Unreal, however, this can be easily improved in the future, thanks to our modular design.

We have created a large exploration area of 40^2 km. The basic surface was derived directly from

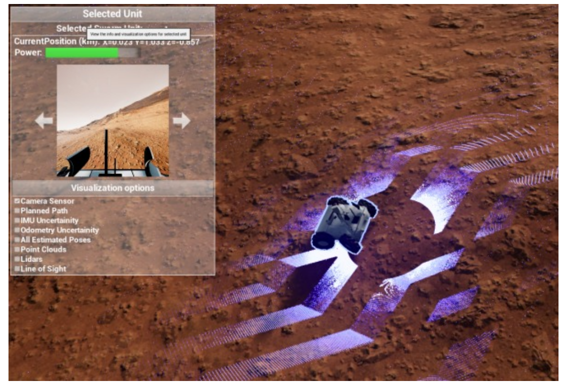

Fig. 11: Different camera views on the same screen: a Visualization of the Lidar point cloud and the RGB image of the rover's camera.

Fig. 12: Highly detailed model of the terrain of Valles Marineris.

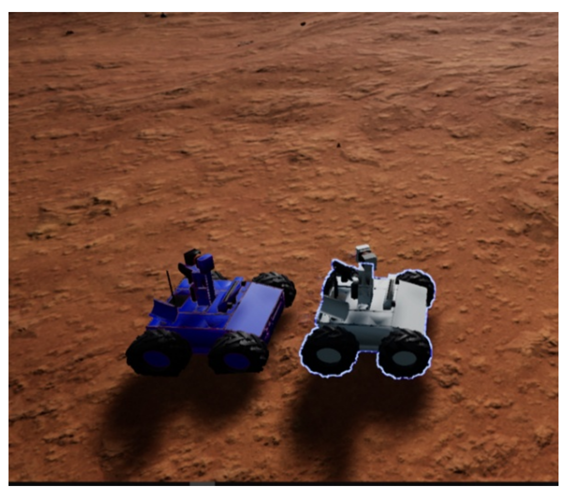

Fig. 13: A blue ghost model shows the expected position of the real rover.

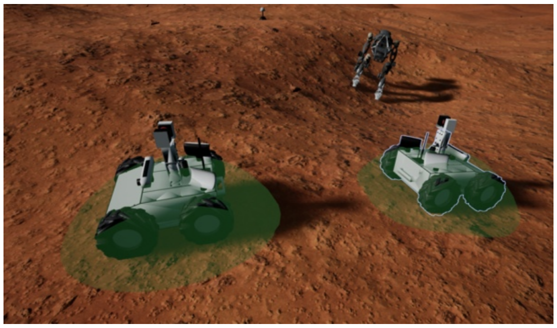

Fig. 14: Ellipsoidal error visualizations help to identify uncertainties of the rover's pose.

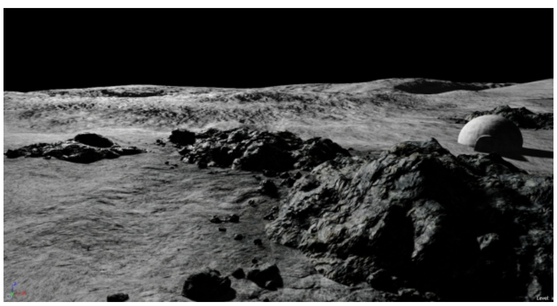

Fig. 15: As a second use case for our VTB we have implemented a very preliminary mission to explore the moon.

Fig. 16: The user can physically interact in VR with our VTB e.g. knock over a rover to simulate an accident.

the HiRISE camera that offers an accuracy of only 1m/pixel. We further populated the surface with high resolution textures and different physical behavior. The textures were created manually, however, we offer a simple 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 12) and a semi-realistic sky-box based on different pictures taken by existing Mars rovers. The VTB simulates a day-night-cycle in Mars time in real-time, however, the simulation can be also accelerated by a factor of up to 4096. Additionally, we have created a first prototype pf the Moon surface that is required for the EFRE innovation project*3D4Space* (see Figure 15).

The UGVs as well as the Charlie robots are both fully physically simulated (e.g. they can turn over) and directly react to the surface model Moreover, our VTB supports the synthesis of Lidar and IMU data as well as RGB(D)-images. The synthesis is implemented accurately: for instance, for the Lidar used by the CoSMiC rovers 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.

The simulated sensor data is obviously send to the individual modules via ROS, however, it can be additionally overlayed to the terrain. Also, additional camera views are supported (see Figure 11) to display this kind of data separately. Furthermore, it is possible to show the line-of-sights of the VaMEx-LAOLa beacons (see Figure 9) and to display environment color-coded environmental information gathered by the VaMEx-swarm (see Figure 10). 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 VaMEx-LAOLa of the vehicles to the ground truth (in the simulation)(see Figure 13). Incoming poses can be shown with color coded ellipsoids (see Figure 14). Further uncertainty visualizations are implemented for LAOLa, orb-slam and IMUs, to name but a few. Obviously, these additional visualizations (paths, ghosts, etc.) are invisible to the ROS-camera synthesis.

The simulation is accessible to the users in several ways. For instance, on a traditional 2D PC system all swarm units can be selected by a mouse cursor and we have realized a sophisticated menu system for interaction and visualization of swarm units and other data. Visualizations can be toggled on or off for the selected swarm unit or all units of the selected type or globally. Moreover, we have integrated a VR mode for different VR devices, including all head mounted displays (HMDs) and interaction devices that are compatible with the Unreal engine but we have also ported it to our Powerwall that allows a larger audience to view and interact simultaneously with the VTB. In VR we support e.g. teleporting for the movement and basic physical interaction with the swarm units with the controllers (see Figure 16).

5. Conclusions and Future Works

We have presented a brief overview on the VaMEx initiative for the swarm-based exploration of the Valles Marineris on Mars and in particular, we have introduced VaMEx-VTB, a virtual testbed for the verification and validation of complex planetary surveying missions. The main idea is to combine a modular software design with the flexibility and high-end graphics of a modern game engine and the connection to ROS via a ROS-bridge adapter. Our VTB supports the simulation of several sensor types in real-time, enhanced visualization modes including different camera views, point clouds and uncertainty measures and an immersive and interactive exploration of the scenario in VR. Finally, we have created a realistic and highly detailed model of Realistic Mars 40^2 km of the Valles Marineris that can be easily adjusted by simply painting terrain types on the surface.

We are confident that 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. A 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 the Mars to enable dynamic processes such as dust devils or sand storms. Moreover, we plan to enhance the basic physics simulation to support also effects such as battery drain, weather effects or wheel tracks. Finally, the navigation and communication technologies researched within the VaMEx initiative are of interest for future missions.

Acknowledgements

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

References

[1] Achint Aggarwal and Frank Kirchner. Object recognition and localization: The role of tactile sensors. *Sensors*, 14(2):3227–3266, 2014.

[2] Jan Albiez, Sylvain Joyeux, Christopher Gaudig, J Hilljegerdes, S Kroffke, C Schoo, Sascha Arnold, G Mimoso, P Alcantara, Rafael Meireles Saback, João Britto Neto, Diego Cesar, Gustavo Neves, Thomio Watanabe, P Merz Paranhos, Marco Reis, and Frank Kirchner. Flatfish-a compact subsea-resident inspection auv. *Proceedings of the MTSIIEEE OCEANS 2015*, pages 1–8, 01 2015.

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

[4] S. Chick, P.J. Sanchez, D. Ferrin, and D.J. Morrice. A simulation test bed for production and supply chain modeling. In *Proceedings of the 2003 Winter Simulation Conference, 2003.*, volume 2, pages 1174–1182 vol.2, Dec 2003.

[5] Christopher Crick, Graylin Jay, Sarah Osentoski, Benjamin Pitzer, and Odest Chadwicke Jenkins. Rosbridge: Ros for non-ros users. In *ISRR*, 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. Dion, J. Oh, and R. Robinson. Virtual testbed for assessing probe vehicle data in intellidrive systems. *IEEE Transactions on Intelligent Transportation Systems*, 12(3):635–644, Sep. 2011.

[8] M. Eich and M. Goldhoorn. Semantic labeling: Classification of 3d entities based on spatial feature descriptors. In *Best Practice Algorithms in 3D Perception and Modeling for Mobile Manipualtion*, 2010.

[9] M. Hildebrandt, J. Albiez, and F. Kirchner. Computer-based control of deep-sea manipulators. In *OCEANS 2008 - MTS/IEEE Kobe Techno-Ocean*, pages 1–6. IEEE, 2008.

[10] Daniel Kuehn, Moritz Schilling, Tobias Stark, Martin Zenzes, and Frank Kirchner. System design and field testing of the hominid robot charlie. *Journal of Field Robotics*, 34(Issue 4):666–703, 4 2016.

[11] Patrick Lange, René Weller, and Gabriel Zachmann. Multi agent system optimization in virtual vehicle testbeds. In *EAI SIMUtools*, Athens, Greece, August 2015. EAI.

[12] Johannes Lemburg, Jose de Gea Fernandez, Markus Eich, Dennis Mronga, Peter Kampmann, Andreas Vogt, Achint Aggarwal, Yuping Shi, and Frank Kirchner. Aila - design of an autonomous mobile dual-arm robot. In *ICRA*, pages 5147–5153. IEEE, 2011.

[13] Alfred S McEwen, Eric M Eliason, James W Bergstrom, Nathan T Bridges, Candice J Hansen, W Alan Delamere, John A Grant, Virginia C Gulick, Kenneth E Herkenhoff, Laszlo Keszthelyi, et al. Mars reconnaissance orbiter's high resolution imaging science experiment (hirise). *Journal of Geophysical Research: Planets*, 112(E5), 2007.

[14] Morgan Quigley, Brian Gerkey, Ken Conley, Josh Faust, Tully Foote, Jeremy Leibs, Eric Berger, Rob Wheeler, and Andrew 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.

[15] 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.

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

[17] Dirk Spenneberg, M. Albrecht, Till Backhaus, Jens Hilljegerdes, Frank Kirchner, A. Strack, and H. Zschenker. Aramies: A four-legged climbing and walking robot. In *Proceedings of 8th International Symposium iSAIRAS. International Symposium on Artificial Intelligence, Robotics and Automation in Space (iSAIRAS), Munich*, Munich, 2005.