

VERVE: A 3D Operations Tool for Space Robots

DW Wheeler^{a,*}, Mark Allan^b, Julius Gyorfi^c

^{a,b,c} KBR, Inc, NASA Ames Research Center, Moffett Field, CA, USA, dw.wheeler@nasa.gov, mark.b.allan@nasa.gov, julius.s.gyorfi@nasa.gov

* Corresponding Author

Abstract

Maintaining a sustainable presence in space requires operations tools that can monitor and control heterogeneous assets in a uniform and collaborative manner. The Visual Environment for Remote Virtual Exploration (VERVE) is a tool that can display telemetry from and control multiple robots simultaneously. Because of its modular design, VERVE has been easily adapted to many kinds of robots and situations. VERVE is currently in use as the rover driving interface for the Volatiles Investigating Polar Exploration Rover (VIPER) lunar prospecting mission and as the supervisory control interface for the Astrobees free-flying robots on the International Space Station (ISS). Previous uses of VERVE include analog field testing of planetary robots, control of a planetary rover from the ISS, and fleet management of self-driving cars.

Central to the VERVE interface is a 3D virtual world that displays robot state, location, and attitude in its environment, joint configuration, raw and derived sensor data, proposed and executing commands, maps, and map annotations in an intuitive manner. Each VERVE instance synchronizes proposed commands and map annotations with other VERVE instances, facilitating collaboration between operators at separate consoles. Time-dependent map layers keep operators up to date with dynamic terrain-specific data, such as sun shadows and direct-to-earth communication coverage. VERVE's detailed model of the world enables many additional features, such as automatic sun-angle and antenna-unwind decision support, collision-checking and duration estimation for proposed paths, virtual measurement of environmental features, and masking the rover body out of images for better stereo results.

This paper describes the VERVE features that have been developed and refined for varied robotic systems over more than ten years. We also discuss the technologies that enable VERVE's many capabilities, and the reasons for selecting those technologies. We conclude by exploring the way VERVE has been adapted for a lunar rover mission, and the ease of adapting VERVE for future missions.

1. Introduction

For over two decades, the Intelligent Robotics Group (IRG) at NASA Ames Research Center has been studying the remote operation of wheeled rovers [1,2], legged robots [3], underwater robots [4,5], zero-gravity free-flyers [6,7,8], and autonomous vehicles [9,10,11]. The form, tools, and operational environments of these robots differ widely, but their operators all need the same types of information for remote situational awareness: direct telemetry such as the robot's state, location, and sensor data; derived data such as detected hazards; and context such as maps, topography, and a traverse plan. IRG has experimented with immersive telepresence for situational awareness [5,12], but prolonged use of stereo or head-mounted displays can induce visual fatigue, nausea, and increased mental load [13,14,15,16]. Virtual 3D interfaces, depicted on a 2D screen, are commonly used to gain situational awareness for robotics [17,18]. In the field of space robotics, the Jet Propulsion Laboratory has used the Rover Sequencing and Visualization Program to create, view, and validate sequences of commands for robots on Mars, to be uplinked once per Martian day [19,20,21]. Because IRG's robots operate no farther away than the moon, their command latency from Earth is only a few seconds. The short latency means actions can be decided and commanded in real-time, rather than as pre-planned sequences, so new tools are required to provide real-time situational awareness to the operations and science teams. For this purpose, IRG has developed a desktop application called VERVE that uses a 3D virtual scene to display robot state and allow motion commands to be sent.

VERVE is built on the Eclipse Rich Client Platform and Open Services Gateway Initiative plug-in architecture, which lends itself to highly modular code. Separate teams have written plug-ins implementing different visualizations and then combined the plug-ins into one application; later projects reused some plug-ins while updating others. In this manner, VERVE (previously called Viz) has been continuously improved and upgraded to support a wide variety of robotic systems since 2008, developing new features without needing to rewrite existing ones. [7,22,23]



Fig 1. Screenshot of VERVE for VIPER.

2. Core Capabilities

All VERVE applications contain several smaller views which each serve a specific purpose; a typical VERVE application is shown in Figure 1. The primary VERVE view is always the 3D view, which displays an articulated robot avatar, graphical representations of sensor data, and a depiction of the robot's relevant environment, which may be terrain, road networks, or indoor structures. Other VERVE views display additional information such as sensor readings, the status of issued commands, or images from a camera. Views can be individually opened, closed and rearranged; views such as the Image Viewer can have multiple instances open, each connected to a different telemetry stream, allowing the user to monitor multiple cameras or even multiple robots as needed. Users can save a configuration of views suitable for a particular task as a reusable "Perspective".

2.1 Robot models

Every VERVE iteration includes robot avatars in the 3D view to indicate the physical states of those robots in their environment (e.g., location and kinematic state) and the robots' perceptions of their environment (through visualizations of raw and derived sensor data). Robot health and communication status are conveyed through icons in the Heads-Up Display (HUD), text callout boxes in the 3D scene, or by changing the color and appearance of the robot avatar.

The abstract data model that drives the robot avatar is defined in a plug-in that describes, at a minimum, the set of identifiers for that type of robot, a command bridge, a 3D model, the components that comprise the robot avatar, and a factory to construct those components. The identifier determines which plug-in should instantiate the robot avatar and how to specialize the instance for different hardware configurations. The command bridge adapts common VERVE commands to robot-specific commands so commanding can be done in a uniform manner, and any commands that are unique to a particular robot can be implemented in a sibling user interface plug-in.

2.1.1 High level architecture

VERVE loads robot models specified by the ROS Universal Robot Description Format (URDF) [24] or NASA Enigma [25] formats. These formats specify the kinematics of a robot and reference 3D mesh files that are associated with the robot's links. Supported mesh formats include .stl, .obj, Virtual Reality Modeling Language (VRML), and COLLaborative Design Activity (COLLADA). The robot description is translated into a scene graph subtree that is attached to the virtual world scene graph. After loading the robot description, the avatar builder walks the subtree and attaches visualization components to the graphical robot model.

The components that comprise the robot avatar are referred to as robot "parts." The RobotPart base class has facilities to connect and disconnect from telemetry streams, callbacks for updating internal state and graphics state, as well as common user interface actions such as toggling visibility. The state of each part can be changed by robot telemetry or user interaction; parts can provide 3D visualizations to the virtual scene, elements for the robot's HUD, in-scene

command interfaces, or data stores for other parts and user interface elements. The parts are designed to be reusable between robots wherever possible to provide a consistent look and feel but may be specialized to accommodate unique robot capabilities.

VERVE supports monitoring of multiple heterogeneous robots simultaneously, but interaction is encouraged to be with a single robot at a time to reduce the risk of commanding errors. To facilitate this, VERVE has the notion of an "agent of interest" that determines which robot the user interface is focused on. When the agent of interest changes, the 3D view centers on the selected robot, the HUD reconfigures, and ancillary views switch telemetry streams to display data from the newly selected robot. Telemetry continues to feed the data model of other robots although high-rate telemetry streams can optionally be dropped when a robot is no longer the agent of interest. If desired, the user can "pin" individual views to a specific robot if they do not want that view to track the current agent of interest.

Receiving telemetry from robots and state synchronization between VERVE instances depends on publish/subscribe networking middleware. Recent VERVE projects utilize the Robotic Operating System 2 (ROS2) middleware [26] which is layered on top of the Object Management Group's Data Distribution Service (DDS) [27]. Prior pub/sub middleware integrated into VERVE includes the DDS-based Robot Application Programming Interface Delegate (RAPID) [28] and Common Object Request Broker Architecture (CORBA) Notification Service [29]. Depending on the robot, commanding may be through pub/sub, remote procedure call, RESTful interface, or some combination thereof.

2.2 Planetary Terrain

A key feature of VERVE is the ability to incorporate georeferenced digital terrain models consisting of digital elevation models (DEMs) and orthorectified surface imagery (orthoimages) to produce a realistic view of a particular location or region in support of remote robotic activities. The DEMs are used to generate a topological mesh onto which surface imagery can be texture mapped.

VERVE utilizes the Ardor3D terrain engine to combine DEMs and orthoimages into a virtual landscape based on clipmaps [30]. The virtual landscape consists of high-resolution geometry near the center of the view with decreasing levels of resolution farther away. The default resolution in VERVE is a four cm grid falling off by a factor of two toward the horizon, though not all VERVE projects have had maps available at that level of detail. Ardor3D was chosen because it has several features imperative for planetary rovers, such as excellent support for large terrain data files and the ability to update terrain as new mapping data is gathered. VERVE loads additional terrain layers and markups using the Open Geospatial Consortium's Web Map Service (WMS) [31].

3. Applications

VERVE has been used to operate terrestrial rovers from Earth [22,32], ISS free-flyers from Earth [6,7,8], terrestrial rovers from ISS [23,33], and autonomous vehicles from a remote command center [9,10,11]. This section explores the most recent project to use VERVE: operating a lunar rover from Earth.

3.1 VIPER Mission

The Volatiles Investigating Polar Exploration Rover (VIPER) (Figure 2) is a NASA mission to explore at the lunar south pole to characterize subsurface volatiles [34]. The rover is powered by three solar panels located on the sides and the rear of the vehicle, and it relies on direct-to-Earth comms via a gimbaled high gain antenna (HGA). Prior to mission launch, a set of long-range strategic traverse plans are calculated from orbital imagery to bring the rover to areas of interest to science and to keep the solar-powered rover out of sun and comm shadows [35]. Shorter-term tactical plans, derived from the strategic plan, are continually updated in-mission to keep activities within the constraints of the strategic plan. VIPER takes advantage of the minimal lunar time delay by offloading compute-heavy sensor processing to software on the ground and by integrating human planning into each drive step. A four-person Drive Team plans each drive to avoid local obstacles while staying within constraints of the tactical plan, maintaining rover localization, and positioning the rover for science instrument activities. The Drive Team consists of a Driver, Co-Driver (Co-D), Rover Navigation (Nav), and Real Time Science (RTSci). All members of the Drive Team use VERVE for VIPER (Figure 1) for situational awareness, but only the Driver and Co-D have command authority.

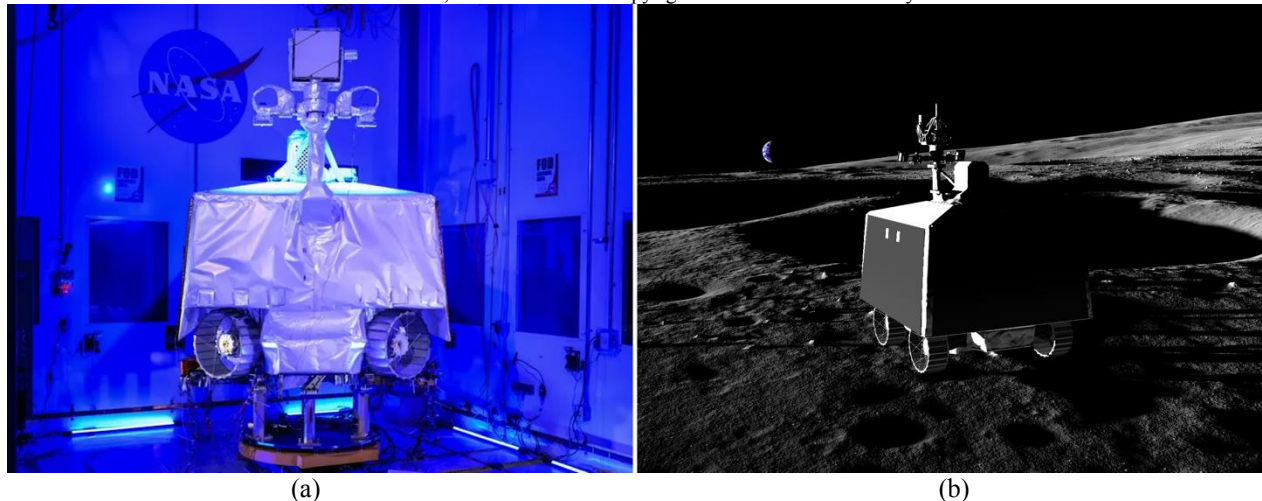


Fig 2. (a) VIPER prior to environmental testing. (b) Simulation of VIPER on the moon.

VIPER driving is done in a series of “drive steps” that each have the following substeps: (1) The Driver adjusts the NavCam gimbal and triggers a stereo image pair of the terrain. (2) VIPER captures and downlinks the images. (3) Ground software processes the images. (4) VERVE displays the images and derived products (Section 3.1.2) in the 3D view with the rover model. (5) The Driver creates a new waypoint (location and orientation) in the 3D view that is up to eight meters away and sends the drive command to VIPER. (6) VIPER turns to the commanded orientation and drives to the commanded location. Steps (6) and (1) typically overlap because a significant percentage of the time in a drive step is waiting for the drive to complete and images to downlink, so the Driver pipelines the sequence by pointing the NavCam and queuing an image command before the rover has reached the waypoint. Each complete drive step takes approximately five minutes, though a drive step may take longer if the Driver needs to take multiple image pairs to find a clear path.

The Nav position monitors the accuracy of rover localization. Nav uses VERVE's 3D view to assess the validity of point clouds and hazard maps, and to ensure that stereo image overlap is sufficient for visual odometry. Nav also looks ahead on the traverse at the DEM for features that would work well for terrain registration (aligning stereo point clouds with orbital imagery to refine the estimate of the rover's location). The RTSci position liaises between the Science Team and the Driver to communicate current science priorities; RTSci uses VERVE to evaluate hazards and local, meter-scale slopes against the backdrop of science activities and objectives, particularly when selecting and evaluating a drill site. RTSci also coordinates science image requests with the Drive Team through VERVE.

3.1.1 Rover pose

An estimate of VIPER's position is maintained by a factor graph which is fed by the rover's on-board extended Kalman filter, as well as by visual odometry from stereo imagery, terrain registration from stereo panoramas, and manual pose correction inputs from the ground. The factor graph tracks position uncertainty as a covariance matrix, and it sends the covariance matrix to VERVE to display as an error ellipse around the robot model.

VIPER's pose history is maintained by a frame store and can be displayed in the VERVE 3D view as a path of directional arrows that point the way the rover was facing (VIPER spends much of its time driving backward to maximize solar power generation). If the estimated pose diverges from ground truth, the landmarks in recent images will not align with the rover's location on the orbital base map in the VERVE 3D view. A manual pose adjustment can be submitted from VERVE by dragging an interactive widget to the user's estimate of the rover's location. VERVE sends the pose update to the frame store, which propagates the update through its previous poses.

3.1.2 Images and derived image products

VIPER has a gimballed stereo camera (NavCam) on a mast at the front, a fixed stereo camera (AftCam) in the solar panel facing aft, and fixed monocular cameras in each wheel well (HazCams). To illuminate shadowed terrain, VIPER has two lights on the NavCam gimbal (NavLights) and six lights fixed around the base of the chassis (HazLights). The Driver manually triggers images in VERVE by specifying cameras, lights, exposure times, and onboard processing settings in a camera command. After the images are downlinked, ground software processes the stereo image pairs into stereo point clouds and hazard maps (Section 3.1.2.1) and sends these derived image products to VERVE via ROS2. Derived image products share the timestamp of their progenitor image pair to simplify bookkeeping.

3.1.2.1 Hazard Maps

A hazard map is a visual indicator of the level of mobility risk associated with terrain in the immediate area in front of the cameras. A hazard map generally covers a wedge of terrain that ranges from about one to fifteen meters out from the camera, but that is highly dependent on the camera angle, terrain slope, and lighting conditions. Hazard maps are derived from a geometric analysis of stereo point clouds that reconstructs local terrain shape and estimates terrain slope and the step height of positive obstacles such as rocks. Ground software computes these attributes and combines them with vehicle operational limits to calculate hazard scores for 10 cm-sized grid locations of visible terrain. For each stereo pair, the hazard score and its component attributes are put as layers into a ROS2 GridMap message (the “hazard map”) and sent to VERVE for display.

The visualization of the hazard score layer in a hazard map in VERVE affords the Driver a rapid way to assess overall trafficability, and it is the primary layer viewed during nominal driving. Other layers in the hazard map can be visualized for specific terrain attributes, such as slope, gradient magnitude and direction, elevation isolines, and step magnitude. These layers are used to gain a detailed understanding of the local terrain for scenarios such as drill site selection or navigating through high-risk areas. Drivers can adjust the display parameters including color gradients to make identifying potential hazards easier.

3.1.2.2 Image Viewer

An image sample can be viewed in an Image Viewer window in VERVE (Figure 1). Multiple Image Viewers can be open at one time, each set to a different camera. Each Image Viewer displays the camera settings for the image and overlays a labeled grid on the image to facilitate conversation between operators. Image Viewers allow scrolling through past images, with an orange border around older images. The most recent image has freshness markers that indicate whether the NavCam orientation or the rover pose has moved since the image was taken.

3.1.2.3 Camera sensor samples in the 3D view

VERVE lists images and derived image products (point clouds and hazard maps) in a table in the Sensor Sample Manager view (Figure 1), with one row per timestamp. From this table, users can select image products to display in the VERVE 3D view. Because point clouds and hazard maps may overlap each other, the opacity of each image product is controlled via slider in the Sensor Sample Manager. Images can be placed in the 3D view as small billboards or “photo quads” directly in front of the relevant camera position. To understand the context of an image, the user can “zoom” to the photo quad; to zoom, the virtual camera flies to the position of the robot’s camera and views the photo quad from the robot’s perspective (Figure 3).

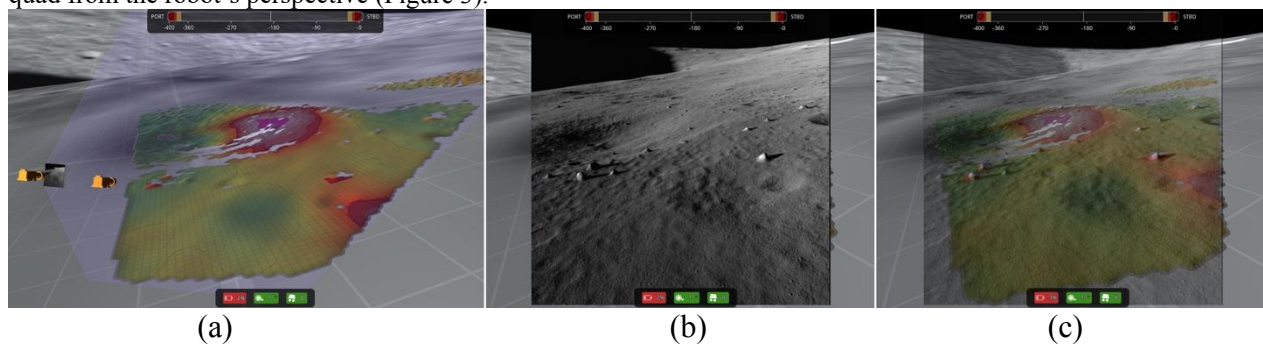


Fig. 3 – Viewing a photo quad in VERVE for VIPER. (a) shows VIPER's left and right NavCams, the left photo quad, and the hazard map from the stereo pair. (b) shows the 3D view “zoomed” to the left photo quad. Note that the photo quad image aligns with the terrain behind it. (c) is the same view as (b), with the photo quad slightly transparent to illustrate the alignment between image features and hazard scores.

3.1.3 Markup

Mission planning, navigation, and science operations require the capability to mark up or annotate terrain maps. These markups include naming specific locations, drawing outlines of science activity regions, planning traverse routes, and highlighting areas of interests. To do this, VERVE has plug-ins that support displaying GeoJson [36] and Keyhole Markup Language (KML) [37] data. VERVE for VIPER also introduced a collaborative Map Markup feature that allows users to draw points, lines, and polygons and share them among VERVE instances and other map display tools using a REST interface on a dedicated VIPER Map Server.

3.1.4 Features and Decision Aids Specific to the VIPER Rover

VERVE for VIPER includes several features specific to the VIPER hardware that are designed to reduce the mental load for the VIPER Driver. For instance, VERVE for VIPER lets the user place a virtual “CamPoint” on the terrain in

the 3D view and command the NavCam gimbal to point at the CamPoint, rather than typing in pan and tilt angles. VERVE computes for operators the rover orientation that maximizes power generation from the solar panels on the rover's aft and side faces, then displays the rover's degrees from this "ideal heading". VERVE can also automatically orient a prospective waypoint to the ideal heading.

Other VERVE for VIPER features were added to support VIPER's unique operational constraints. For instance, VIPER has a high gain antenna (HGA) on a gimbal that automatically points toward Earth. The pan gimbal's range of motion is 400 degrees, and it automatically unwinds itself when it reaches a hard stop. Unwinding the HGA gimbal breaks communication with Earth, leading to a loss of comm for up to several minutes. To prevent an unexpected loss of comm, VERVE automatically chooses a turn direction for the rover that will avoid antenna unwinding, which sometimes results in a turn of more than 180 degrees to reach a commanded heading. So that the Driver is not surprised when the rover turns "the long way," VERVE displays an arrow indicating the turn direction for each waypoint, as well as a prominent visualization of the current and prospective HGA angles in relation to the hard stops. This feature reduces the already considerable mental load for the Driver.

VERVE for VIPER also automatically calculates "Golden" approach points for potential drill sites. Scientists must take a well-lit image of the undisturbed drill site, from as close a distance as possible, before the rover drives over the site to position the drill that protrudes down from its belly. A "Golden Point" is the closest possible approach to the drill site from which VIPER can take an image without the shadow of the rover obscuring the site, with VIPER's solar panels at an ideal angle for power generation, and from which VIPER can place the drill over the site without trampling the site with its wheels. VERVE's VIPER-specific features improve command accuracy and efficiency and decrease the mental load of the Driver.

4. Conclusions

The modular design and loose coupling of VERVE have allowed it to be used in a wide variety of contexts, such as commercial and research robotics activities on Earth, on orbit, and planned on the Moon. Most robot interfaces benefit from 3D models of the robots of interest in their environment, parameterized by live telemetry, with visualizations of sensor readings and images from the robots. VERVE fills this role and is also a versatile commanding interface, having been adapted to use multiple forms of middleware to send commands, and allowing users to construct commands by placing targets in its interactive 3D environment. The greater situational awareness and intuitive commanding modes in VERVE enable users to efficiently and effectively conduct their operations.

Acknowledgements

Thank you to past VERVE team members David Williams, Josh Slack, Ryan Goetz, Lee Brownstone and interns Dean Ramos and Mei Wa Yu. Thanks also to those who gave feedback on VERVE and helped with the deployment and testing, including Charles Hacskeylo, Vickie Moore, Rusty Hunt, Jason Fugate, Charlie Yan, Andy Mishkin, Matt Machlis, Rick Elphic, Hao Thai, and Karen West. We'd also like to acknowledge all the VIPER Drivers: Terry Fong, Matt Deans, Antoine Tardy, Arno Rogg, Mark Allan, DW Wheeler, Parker Francis, Andrzej Jackowski, Emily McBryan, Lorenzo Fluckiger, and Hans Thomas. VIPER is funded by the NASA Science Mission Directorate. Astrobee development was funded by the Game Changing Development program in the NASA Space Technology Mission Directorate and by the Advanced Explorations Systems program in the NASA Exploration Systems Development Mission Directorate.

References

- [1] L. Piguet, B. Hine, P. Hontalas, T. Fong, E. and Nygren. The Virtual Environment Vehicle Interface: a Dynamic, distributed, and flexible virtual environment, IMAGINA '96: New Frontiers of CyberExistence, Monte Carlo, Monaco, 1996.
- [2] L. Nguyen, M. Bualat, L. Edwards, L. Flueckiger, C. Neveu, K. Schwehr, M. Wagner, and E. Zbinden, Virtual reality interfaces for visualization and control of remote vehicles. *Autonomous Robots* 11(1), 2001, pp. 9-18.
- [3] T. Fong, H. Pangels, D. Wettergreen, E. Nygren, B. Hine, P. Hontalas, and C. Fedor, Operator interfaces and network based participation for Dante II. SAE 951518, SAE 25th International Conference on Environmental Systems, San Diego, CA. 1995.
- [4] L. Piguet, T. Fong, B. Hine, P. Hontalas, and E. Nygren, VEVI: a virtual reality tool for robotic planetary explorations, *Virtual Reality World '95*, Stuttgart, Germany, 1995.
- [5] B. Hine, C. Stoker, M. Sims, D. Rasmussen, P. Hontalas, T. Fong, J. Steele, D. Barch, D. Andersen, E. Miles, and E. Nygren, The application of telepresence and virtual reality to subsea exploration, IARP 2nd Workshop on Mobile Robots for Subsea Environments, Monterey, CA, 1994.

- Copyright ©2025 by the Canadian Space Agency (CSA) on behalf of SpaceOps. All rights reserved. One or more authors of this work are employees of the government of the United States of America, which may preclude the work from being subject to copyright in the United States of America, in which event no copyright is asserted in that country.
- [6] T. Fong, M. Micire, T. Morse, E. Park, C. Provencher, V. To, and D. Wheeler, Smart SPHERES: A telerobotic free-flyer for intravehicular activities in space, AIAA Space 2013 Conference and Exposition, San Diego, CA, 2013, 10-12 September.
 - [7] M. Bualat, T. Smith, E. Smith, T. Fong, D. W. Wheeler, and the Astrobe Team, Astrobe: A New Tool for ISS Operations, Intl. Conf. on Space Operations (SpaceOps 2018), Marseille, France, 2018, 28 May – June 1.
 - [8] T. Smith, J. Barlow, M. Bualat, T. Fong, C. Provencher, H. Sanchez, E. Smith, et al., Astrobe: A New Platform for Free-Flying Robotics on the International Space Station, Intl. Symp. on AI, Robotics, and Automation in Space (i-SAIRAS), Beijing, China, 2016, 19-22 June.
 - [9] S. Thakur, A. Guerin, J. S. Györfi, K. Poulet, A. Laurent, M. B. Allan, O. Bentahar, System to recommend sensor view for quick situational awareness, US Patent 11,249,479.
 - [10] S. Thakur, A. Guerin, A. Kobashi, J. S. Györfi, M. B. Allan, Remote operation extending an existing route to a destination, US Patent 11,460,841.
 - [11] K. H. Wray, O. Bentahar, A. Vagadia, L. Cesafsky, A. Jamgochian, S. Witwicki, N. M. Baig, J. S. Györfi, S. Zilberstein, S. Sharma, Explainability of autonomous vehicle decision making, US Patent 11,577,746.
 - [12] C. Stoker, B. Hine, Telepresence control of mobile robots – Kilauea Marsokhod experiment, AIAA 34th Aerospace Sciences Meeting, 1996.
 - [13] A.D. Souchet, D. Lourdeaux, A. Pagani, et al., A narrative review of immersive virtual reality's ergonomics and risks at the workplace: cybersickness, visual fatigue, muscular fatigue, acute stress, and mental overload. *Virtual Reality* 27, 2023, 19-50.
 - [14] T. Porcino, D. Trevisan and E. Clua, Minimizing cybersickness in head-mounted display systems: causes and strategies review, 22nd Symposium on Virtual and Augmented Reality (SVR), Porto de Galinhas, Brazil, 2020, pp. 154-163.
 - [15] J. Y. Chen, E. C. Haas, and M. J. Barnes, Human performance issues and user interface design for teleoperated robots, *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 37, no. 6, pp. 1231-1245, 2007.
 - [16] J. Orlosky, K. Theofilis, K. Kiyokawa, Y. Nagai, Effects of Throughput Delay on Perception of Robot Teleoperation and Head Control Precision in Remote Monitoring Tasks, *Presence: Teleoperators and Virtual Environments* 2018; 27 (2): 226-241.
 - [17] H. Kam, S. Lee, T. Park, C. Kim, Rviz: a toolkit for real domain data visualization, *Telecommunications Systems*, Volume 60, Issue 2 Pages 337 – 345, 01 October 2015.
 - [18] Foxglove – Visualization and observability for robotics developers, <https://foxglove.dev> (accessed 4.4.25).
 - [19] V. Verma et al., "Results from the First Year and a Half of Mars 2020 Robotic Operations," 2023 IEEE Aerospace Conference, Big Sky, MT, USA, 2023, pp. 1-20.
 - [20] J. Wright, F. Hartman, B. Cooper, S. Maxwell, J. Yen and J. Morrison, Driving on Mars with RSVP, *IEEE Robotics & Automation Magazine*, vol. 13, no. 2, pp. 37-45, June 2006.
 - [21] J. Koch, G. Johnson, N. Wiltsie, F. Hartman, F. Alibay and J. Yen, Helicopter and Rover Operations on Mars using the Robot Sequencing and Visualization Program (RSVP), 2022 IEEE Aerospace Conference (AERO), Big Sky, MT, USA, 2022, pp. 1-9.
 - [22] Fong, Terrence, et al, Field testing of utility robots for lunar surface operations. AIAA SPACE 2008 Conference & Exposition. 2008.
 - [23] M. Bualat, D. Schreckenghost, E. Pacis, T. Fong, D. Kalar, and B. Beutter, Results from testing crew-controlled surface telerobotics on the International Space Station, International Symposium on AI, Robotics, and Automation in Space (i-SAIRAS), Montreal, Quebec, 2014, 17-19 June.
 - [24] ROS URDF, <https://wiki.ros.org/urdf> (accessed 18.2.25).
 - [25] D. Shores, S. Goza, C. McKeegan, R. Easley, J. Way, Enigma Version 12, 2013, September.
 - [26] S. Macenski, et al., Robot Operating System 2: Design, architecture, and uses in the wild. *Sci. Robot.* 7, eabm6074, 2022.
 - [27] Data Distribution Service, <https://www.omg.org/dds> (accessed 19.2.25).
 - [28] R. J. Torres, M. Allan, R. Hirsh and M. N. Wallick, RAPID: Collaboration results from three NASA centers in commanding/monitoring lunar assets, 2009 IEEE Aerospace conference, Big Sky, MT, USA, 2009, pp. 1-11.
 - [29] U. Lang, R. Schreiner, Developing Secure Distributed Systems with CORBA, Artech, 2002.
 - [30] C. C. Tanner, C. J. Migdal, M. T. Jones, The Clipmap: A Virtual Mipmap, 25th Annual Conf. on Computer Graphics and Interactive Techniques, 1998.
 - [31] Web Map Service, <https://www.ogc.org/publications/standard/wms> (accessed 21.1.25).
 - [32] M. Bualat, A. Abercromby, M. Allan, X. Bouyssounouse, M. C. Deans, T. Fong, et al., Robotic recon for human exploration: Method, assessment, and lessons learned. *Analogs for planetary exploration*. (2011) 117-135.

- [33] M. Bualat, T. Fong, M. Allan, X. Bouyssounouse, T. Cohen, L. Fluckiger et al., Surface telerobotics: development and testing of a crew controlled planetary rover system, AIAA Space 2013 Conference and Exposition, San Diego, CA, 2013, 10-12 September.
- [34] A. Colaprete, R.C. Elphic, M. Shirley, K. Ennico-Smith, D.S.S Lim, K. Zacny, The Volatiles Investigating Polar Exploration Rover (VIPER) Mission, In 52nd lunar and planetary science conference (No. 2548, p. 1523), 2021, March.
- [35] M. Shirley, E. Balaban, et. al., VIPER Traverse Planning, 2022 53rd Lunar and Planetary Science Conference, The Woodlands, TX, 2022, 7-11 March.
- [36] GeoJson, <https://geojson.org> (accessed 3.3.25).
- [37] KML, <https://www.ogc.org/publications/standard/kml>, (accessed 3.3.25).