Proceedings of the 1998 IEEE
International Conference on Robotics & Automation
Leuven, Belgium • May 1998

# Mars Pathfinder Mission Internet-Based Operations Using WITS

Paul G. Backes

Jet Propulsion Laboratory
California Institute of Technology
Pasadena, California

Kam S. Tso and Gregory K. Tharp

IA Tech, Inc.
Los Angeles, California

## Abstract

*The Web Interface for Telescience (WITS) is an Internet-based tool that the Mars Pathfinder mission used for both mission operations at JPL and public outreach. WITS enables the viewing of downlinked images and results in various ways, terrain feature measurement and annotation, and planning of daily mission activities. WITS is written in the Java language and is accessible by mission scientists and the general public via a web browser. The public can use WITS to plan and simulate their own rover missions. WITS will also be used in the 1998 lander and 2001, 2003, and 2005 rover missions to Mars.*

## 1 Introduction

The Web Interface for Telescience (WITS) enables scientists to participate collaboratively in planetary lander and rover missions from their home institutions. WITS was initially developed for use in the 2001 rover mission to Mars, but it achieved a high level of maturity early enough that it was used in the 1997 Mars Pathfinder mission [1, 2]. There were several motivations for its development. Future lander and rover missions to Mars (i.e., 1998, 2001, and 2003) will be long duration, on the order of a year, so it will not be feasible for the mission science team to reside at JPL for the duration of the mission. A mission planning and visualization tool was needed to enable the science team members to participate fully in the mission while being geographically distributed at their home institutions. WITS provides both the visualization of mission data and the collaborative planning tools needed for this scenario. A collaborative planning tool was needed to enable distributed mission scientists to generate the daily mission plan among themselves and with the rover engineering team. WITS provides this collaborative planning capability for all JPL and Internet-connected users.

A tool was needed that was not limited to one or a few operations computers or sites, but which was available to all users on any computer platform at any site. WITS provides this capability by being written in the Java language [3] and being accessed by a common web browser that runs on all major computer platforms and operating systems.

A tool was needed to speed up the daily mission planning process. WITS speeds up this process by enabling collaborative planning by the science team, collaborative planning between the science and engineering teams using a common planning tool, and elimination of paper-based communication of science requests between the science and engineering teams. WITS enables science and engineering planning to be done simultaneously to decide quickly on a feasible plan rather than sequentially with iterations between science and engineering planning.

A tool was needed that would provide complete mission data to the public as quickly as possible. WITS provides this capability by providing to the public the same tool the mission team uses, along with access to the same mission data. In addition, the public can use WITS, as the mission planning team does, to plan and simulate their own missions. The mission plan generated by a public user is resident only on their local computer. Other examples of World Wide Web-based robot operation can be found in [4, 5].

## 2 System Architecture

The WITS system is divided into two primary parts, the client part, which is downloaded and executed on the user's computer, and the server part, which resides and executes at JPL or another server location. The WITS system architecture is shown in Figure 1. There are two parallel WITS systems for a mission: the mission system, which the mission team

0-7803-4300-x-5/98 $10.00 © 1998 IEEE

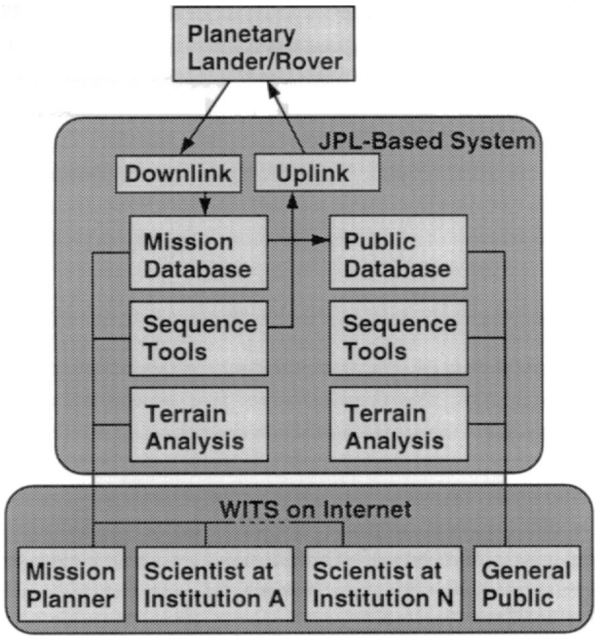


Figure 1: WITS System Architecture

uses to command the real spacecraft, and the public system, which the public uses to plan and simulate their own missions. The systems are identical but separate, with the database for the public system updated with the data from the mission system. No data from the public system is written to the mission system. Having a separate public system ensures that heavy usage by the public does not impact the mission system, as well as providing an element of security for the mission system. For the Pathfinder mission, the mission system server resided at JPL, and the public system server resided at a separate Sun Microsystems-sponsored site at Graham Technology Solutions, Inc.

The client part of WITS is written entirely in the Java programming language as a Java applet and is downloaded onto the user's computer. Users can download the complete WITS applet each time they want to use WITS, or utilize the Marimba, Inc. Castanet technology to download the WITS channel once onto their computer where it will remain and be automatically updated with new versions of WITS. The Castanet channel technology enables WITS to start quickly when a user's Internet connection is not fast, e.g., when using a 28.8K modem.

The server part of the system is not distributed over the Internet. The server communicates with the numerous Internet-based clients. The server has the common database, computationally intensive systems such as stereo processing and range map generation, and interface to the spacecraft uplink/downlink system. A server-located database and current mission plan permits collaboration by enabling all users to see the same data and plan. To modify the common database, e.g., to make an input to the current mission, a user must log in with a password. Users can see all inputs submitted by all other users.

## 3 Multiple Views

Various view windows in WITS permit looking at the images and data in different ways, as described below. Planning and measurement can be done in all views, with all inputs displayed in all views.

### 3.1 Descent View

The Descent View has an orbital image or an image taken by the spacecraft during descent to the surface. There may be various descent images covering different surface extents. These images could be generated by zooming in on one high altitude (e.g., orbital) image or by using descent images taken at different altitudes during the descent to the surface. No orbital or descent images were taken during the Pathfinder mission, so an orbital image of the Pathfinder landing site from the earlier Viking mission was used, as shown in Figure 2. The figure shows the landing site (small rover icon) and the Twin Peaks and Big Crater.

### 3.2 Panorama View

The Panorama View shows an overhead view of the area around the lander (or rover for future rover missions where the rover will take its own panorama image sets). The Panorama View can be shown with either a color-coded elevation map, as shown in Figure 3, or a texture mapped image. There are options for overlaying outlines showing the area on the ground covered by each image in the panorama (Figure 3), and/or a distance grid that is useful for mission planning.

### 3.3 Mosaic View

The Mosaic View (Figure 3) is a mosaic of the actual panorama images. For Pathfinder, the panoramic images were taken by the IMP camera on the lander. For the 2001 rover mission, the panoramic images will be taken from a mast on the rover. There are various options for viewing the mosaic. The mosaic center can

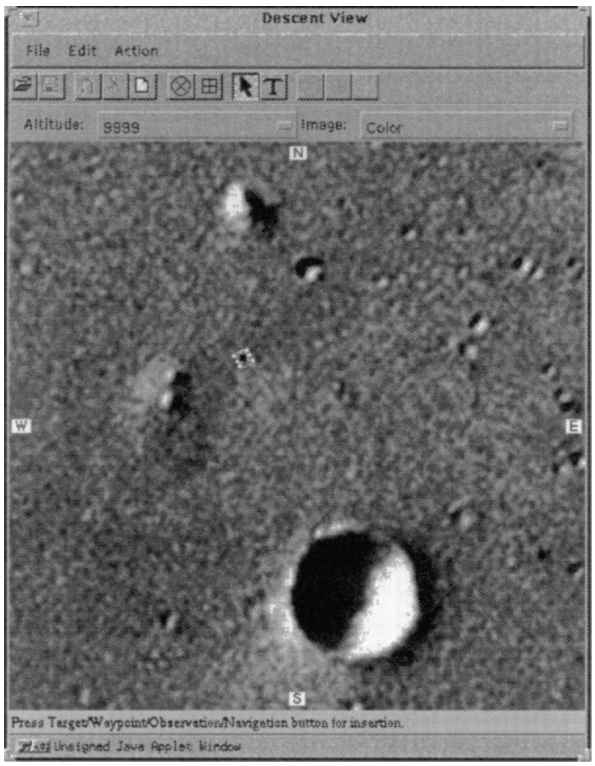


Figure 2: Descent View

be selected by either heading or centering a selected image. The mosaic images can be scaled to 1/8, 1/4, 1/2, or full size. A smaller scale mosaic allows viewing more of the full 360 degree mosaic. Vertical and horizontal slide bars are used to scroll through the mosaic. The images in the mosaic view are compressed at lower resolution than the original images so that the data can be downloaded to the user at a reasonable speed. If the user wants to see a specific image at full resolution, then a Wedge View can be used. Figure 3 shows the Panorama View, Mosaic View, a Wedge View, and a Target Task window all up at the same time. WITS allows any combination of views to be up at the same time. Each view window can be dragged to a desired location on the screen.

### 3.4 Wedge View

The Wedge View (Figure 3) is a view of a single image from the panorama. A Wedge View window is popped up when the user double clicks on the image in the Mosaic View or in its outline in the Panorama View. The Wedge View shows a full-size, full-resolution image. Multiple Wedge Views with different images can be popped up at the same time.

### 3.5 Close-up View

The Close-up View provides viewing of images taken by the navigation cameras on the rover. Planning for motions relative to the rover can be done in the Close-up View. Each image actually represents a stereo pair, so 3D target designation is also provided for planning in the Close-up View.

### 3.6 Anaglyph Stereo

The Mosaic and Wedge Views can be viewed in anaglyph stereo by selecting the anaglyph stereo option in the Panorama View view pull-down menu. With red-blue glasses, the user can see the scene in stereo.

## 4 User Types and Logins

The four user types can view data and plan and simulate missions: Viewer, Navigator, Scientist, and Mission Planner. A Viewer user cannot write anything to the common database; public users are Viewer user types. A Navigator user can save waypoints, waypoint tasks, navigation points, and hazards to the database. A Scientist user can save science targets, target tasks, and observation points to the database. A Mission Planner user can save all types of data to the database. All points (waypoint, target, hazard, observation, navigation) are owned by the user who created them, and only the owner can modify them.

A user can log into WITS with a password. A specific user login has a predefined user type. For the Pathfinder mission, user logins were associated with individuals and groups. There were one rover navigation and three science groups. Group logins are useful to enable anyone in the group to modify information associated with a point owned by the group.

## 5 3D Measurements

The terrain analysis system module of the system resides at the server site. The terrain analysis system processes downlink stereo images and generates the 3D range maps that map image pixel data to 3D coordinates in the scene. Each pair of images is processed by a fast stereo vision system developed at JPL [6] to produce a range map from image coordinates to a 3D world coordinate system. The range map is processed to determine both 3D positions and surface normal geometry. Surface normal geometry is useful for speci-

Figure 3: WITS Screen With Various View Windows

fying instrument placement. The WITS clients on the Internet query the server-based terrain analysis system to return the 3D locations on the terrain surface associated with specified image pixels. The full 3D terrain maps are too large to transfer over the Internet. Each of the 97 image pairs in the panorama used in the Pathfinder mission had a terrain map which was approximately 1.8 Megabytes in size.

When in pointer mode, a user can query WITS for two types of 3D measurements, single points and rulers. A user enters pointer mode, the default mode, by pressing the pointer icon, . If the user clicks on a point in any view, then the 3D coordinate for that point is displayed next to the point. If the user clicks on a point and drags the cursor to another point before releasing the mouse button, then a ruler is drawn between the two points and the magnitude of the distance is displayed next to the ruler. Additionally, the 3D start and end point values are printed in the status bar at the bottom of the Panorama View window.

## 6 Annotation

WITS provides three types of annotation points: observation, navigation, and hazard. These points are entered into a panorama and stay with the panorama, i.e., do not go away when the next plan is begun, as do science targets and waypoints. An annotation point is entered by first selecting its associated icon in the control bar and then the desired point to place it in an image. A text window for annotating the point is popped up by selecting the icon and then the annotation point. An annotation point, as well as waypoints and science targets, can be moved by clicking on it and dragging it to the new desired location. Observation points, , are used by scientists to annotate scientific information about the scene, e.g., results of investigations of a rock. Navigation points, , are used by the rover navigation team to annotate rover navigation information about the scene, e.g., traversability information. Hazards, , are a special type of navigation point that designates hazard areas which the rover should not traverse. Observation and navigation points are distinguished from targets and waypoints by color as well as not being connected by lines in the views. Annotation point icons are to the right of the icon in the icon control bar at the top of the views and target and waypoint icons are to the left.

## 7 Planning

The daily mission planning process includes science and engineering elements. The science team enters science targets, , designating where they want science tasks to be performed. Science targets are automatically associated with the closest waypoint (automatically visualized by a pink line connecting the target to the waypoint). The rover will perform the tasks associated with the target after arriving at the closest waypoint.

Scientists specify what tasks are to be performed at a specific science target location by selecting the icon and then the science target. A Target Task window pops up, as shown in Figure 3. On the right is the science target task to be performed at this target. On the left are subtasks and macros that can be inserted in the task. Subtasks are lists of commands and macros. When inserting a subtask into the target task, the list of subtask commands and macros is inserted. When inserting a macro into the target task, the macro name and its argument list are inserted. The target task of Figure 3 has only macros in it. A macro is expanded into a specific list of commands, dependent on the macro parameters, before sending the complete command sequence to the spacecraft. The use of macros enables specification of spacecraft activities without editing low-level spacecraft commands. A macro command window can be popped up by selecting the macro and then the Open Cmd button. Macro command windows enable setting macro command parameters, simulating macro commands, and displaying the sequence of commands into which this macro will be expanded. The Edit pull-down menu is used for editing the task.

The engineering team enters waypoints, , to specify the path for the rover to follow to reach all the science targets. The engineering team can specify waypoint tasks for the waypoints, using Waypoint Task windows, in the same way that the science team can specify target tasks for science targets. The task association with scene icons uses the Task Lines approach for visual representation and editing of robotic sequences [7].

After science targets and waypoints and their associated tasks are specified, the complete sequence to be sent to the rover is generated using the Sequence window. The Sequence window is popped up by selecting Generate/Execute in the Sequence pull-down menu of

the Panorama view. The sequence is automatically generated by following the waypoint path and concatenating the waypoint tasks and science tasks associated with the waypoints and science targets. The sequence can be simulated using the 3D rover model or sent to the rover for execution. In the simulation, the 3D rover follows the path between waypoints and performs the specified activities at the science targets, e.g., the user can see an APXS deployment. Currently only a kinematic simulation is supported.

## 8 Implementation

The WITS client which gets downloaded over the Internet is written entirely in the Java programming language as a Java applet. The server part of the system has two parts: the processing of rover downlink data and Common Gateway Interface (CGI) programs which respond to requests by the WITS applet clients. The server part of WITS is currently written in the C programming language although it could have been written in Java as well, and may be converted to Java in the future. CGI calls are used by the WITS client for access to the server-located database rather than socket communication to enable users who are behind firewalls to use WITS. Firewall security prevents socket communication between the client applet and the server. Java servlet or Remote Method Invocation (RMI) technology might be used in the future in place of the CGI calls to enable more efficient communication.

Security of the system is achieved in several ways. First, the public version is completely separate from the mission system. A mission user must log in to WITS with a password before being allowed to modify the common database. Another issue in security is protection of propriety mission data. The principal investigator for the Pathfinder IMP camera which took the panoramic images agreed to have reduced resolution copies of the panoramic images used in the WITS public system, but did not want technical information such as the range maps and camera models released. Therefore the database was split into a directory tree which was readable by the public for the images and a directory tree which was not readable by the public and which only the CGI programs could access. Thus the full functionality of WITS is provided to the public while maintaining the security of proprietary data.

Mapping of 2D pixel coordinates to 3D terrain locations and from 3D terrain locations to 2D pixel coordinates are integral parts of WITS. 2D pixel to 3D terrain mapping is done in WITS using a call to a CGI program which reads a terrain range map and returns the 3D coordinates associated with the 2D pixel for a specific image. As described in section 5, the range map files are computed using a fast stereo algorithm developed at JPL [6].

Mapping of 3D terrain locations to 2D image locations, for drawing on the WITS windows, is done using CAHV camera models [8]. WITS displays the left image of the stereo pair in its windows. The nominal camera models for the cameras were calibrated before the mission. The mapping from a 3D terrain location, $P$, to a 2D image coordinate, $(I_l, J_l)$, for the left image is given by the following equation for the left camera,

$$\begin{aligned} I_l &= \frac{(P - C_l, H_l)}{(P - C_l, A_l)} \\ J_l &= \frac{(P - C_l, V_l)}{(P - C_l, A_l)} \end{aligned} \quad (1)$$

where $P$ is the 3D terrain location, $C_l$ is the position of the focal center of the left camera, $A_l$ is a unit vector in the direction that the left camera is pointed, $H_l$ and $V_l$ are the horizontal and vertical vectors which are calibration values specific to the camera, and a comma represents a vector scalar product. Each stereo pair downlinked from Mars is processed to generate the corresponding range map file and camera model. The range map file gives the 2D to 3D mapping and equation 1 gives the 3D to 2D mapping.

When the ground that the rover is on is flat, then the range maps are not needed for the 2D to 3D mapping. In this case, the $Z$ component of the $P$ vector in equation 1 can be set to a constant and the $P$ vector can be computed from the $(I_l, J_l)$ pixel coordinates.

The $A$ vector of equation 1 is additionally used to determine where to place the image in the WITS Mosaic View. The equivalent pan and tilt values for the $A$ vector are computed. The pan angle indicates how far across the mosaic to place the image and the tilt angle indicates how far down in the mosaic to place the image. The Mosaic View images do not line-up exactly due to small pointing errors of the camera.

WITS provides simulation of the generated command sequence. There are two simulators: a 3D wireframe simulation in the Mosaic and Wedge Views, and a 3D solid model simulation in the VRML View. A

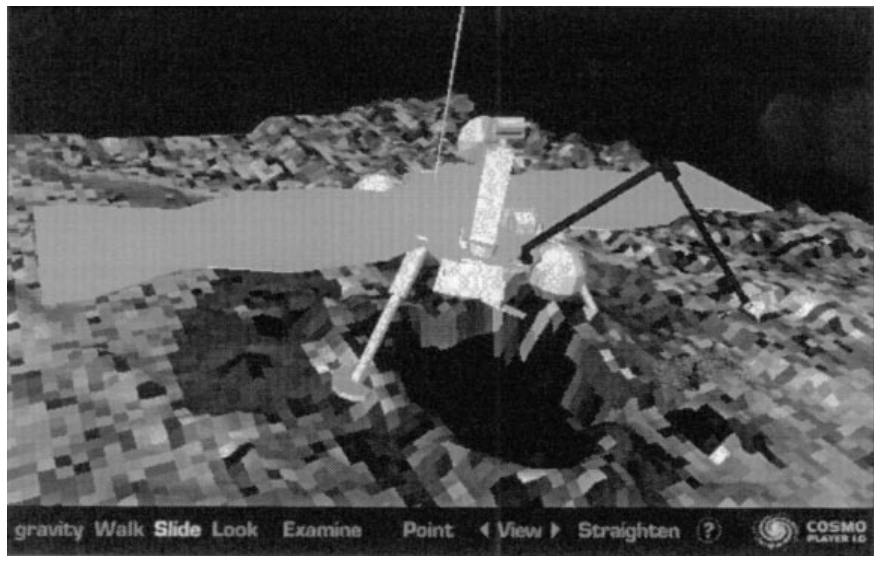


Figure 4: WITS VRML View Showing 1998 MVACS Mission Lander and Arm

wire frame model is used in the Mosaic and Wedge Views so that the model has minimal obscuration of the underlaying image. Simulation in the Mosaic and Wedge View is only viewed from the camera's viewpoint. Obscuration is less of an issue for the VRML View because the simulation can be viewed from any viewpoint. Since Java 1.0.2, which was used for the WITS implementation, does not have 3D graphics capability, a Virtual Reality Modeling Language (VRML) [9] parser was written in Java and integrated in WITS to provide the 3D wireframe simulation. Java 3D, which will be an integral part of future versions of Java, may replace this custom wireframe implementation in the future. The WITS VRML View runs in an embedded VRML browser, e.g., Cosmo Player from Silicon Graphics, Inc.

## 9 Use In Pathfinder Mission

WITS was used in the Pathfinder mission both at JPL for mission operations and as a public outreach tool. For JPL mission operations, WITS was used by scientists to annotate the scene, to determine 3D locations of points for inputs to science activity requests, and for measurements, e.g., sizes of rocks and distances between features. A duplicate of the WITS system which the scientists used at JPL, including a duplicate database, was available to the public at URL http://mars.graham.com/wits/ during the mission. After the mission, WITS will continue to be available to the public at http://robotics.jpl.nasa.gov/wits/. Over 330,000 public users visited the WITS site from July through December, 1997.

## 10 Future Enhancements

Various enhancements are under development for WITS. Two new view windows are being developed: the Color Stereo View and the VRML View. The Color Stereo View will enable viewing color images in stereo. The VRML View will enable 3D solid-model visualization and simulation, and viewing from different directions. The VRML View has already been developed for the next Mars mission, the 1998 Mars Volatiles and Climate Surveyor (MVACS) mission, where lander and arm models have been integrated with a Pathfinder terrain model, as shown in Figure 4. The MVACS lander and arm are visualized on the Pathfinder mission terrain. Most of the terrain is shown in low resolution and one patch of terrain under the arm is shown in high resolution. Multi-resolution terrain visualization is provided to accommodate Internet bandwidth and computer speed constraints.

Automated mission planning is also under development to enable automated waypoint generation based upon the specified science targets and science activities at the targets. The science team will then not have to wait for feedback from the engineering team. Instead, the current mission plan will always be generated and updated automatically. Also, the terrain

analysis system will generate terrain maps from descent imagery and orbiter imagery, and integrate the terrain maps from the various sources. The terrain analysis system will also provide feature maps and traversability analysis which will be used for the automated planning.

## 11 Future Missions

WITS will be used in the 1998 Mars Volatiles and Climate Surveyor lander mission and the 2001, 2003, and 2005 rover missions to Mars. A common core WITS system for visualization and command sequence generation will be used for all of the missions. Specific models and command sets will be used for specific missions.

## 12 Conclusions

WITS enables a new paradigm for mission operations where members of science and engineering teams who are geographically distributed can still participate fully in planetary lander and rover missions on a daily basis. The use of WITS in the Pathfinder mission demonstrates the feasibility of the system for use in the next Mars missions. Also, the success of WITS as a public outreach tool easily usable by the public around the world demonstrates that this is a valuable approach for planetary mission public outreach.

## Acknowledgements

WITS was developed as part of the Long Range Science Rover task [10] in the NASA Telerobotics program where it is used to command the Rocky7 research rover. The research described in this paper was carried out by the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

## References

[1] D. Shirley and J. Matijevic. Mars Pathfinder microrover. *Autonomous Robots*, 2(4):281–289, 1995.

[2] J. Matijevic. Mars Pathfinder microrover — implementing a low cost planetary mission experiment. In *Second IAA International Conference on Low-Cost Planetary Missions*, pages 16–19, John Hopkins Applied Physics Laboratory, Maryland, 1996. paper # IAA-L-0510.

[3] M. A. Hamilton. Java and the shift to net-centric computing. *IEEE Computer*, 29(8):31–39, August 1996.

[4] Eric Paulos and John Canny. Delivering real reality to the World Wide Web via telerobotics. In *Proceedings IEEE International Conference on Robotics and Automation*, pages 1694–1699, Minneapolis, Minnesota, April 22-28 1996.

[5] K. Goldberg, M. Mascha, S. Gentner, N. Rothenberg, C. Sutter, and J. Wiegley. Robot teleoperation via WWW. In *Proceedings IEEE International Conference on Robotics and Automation*, May 1995.

[6] L. Matthies, A. Kelly, T. Litwin, and G. Tharp. Obstacle detection for unmanned ground vehicles: A progress report. In Georges Giralt and Gerhard Hirzinger, editors, *Proceedings, Robotics Research 7th International Symposium*, October 1995. Published by Springer-Verlag, London 1996.

[7] Paul G. Backes, Stephen F. Peters, Linh Phan, and Kam S. Tso. Task lines and motion guides. In *Proceedings IEEE International Conference on Robotics and Automation*, pages 50–57, Minneapolis, Minnesota, April 22-28 1996. Volume 1.

[8] Y. Yakimovsky and R. Cunningham. A system for extracting three-dimensional measurements from a stereo pair of TV cameras. *Computer Graphics and Image Processing*, 7:195–210, 1978.

[9] ISO/IEC 14772-1:1997. *The Virtual Reality Modeling Language (VRML).* International Organization for Standardization and International Electrotechnical Commission, 1997 `http://www.vrml.org/Specifications/VRML97/`.

[10] S. Hayati, R. Volpe, P. Backes, J. Balaram, R. Welch, R. Ivlev, G. Tharp, S. Peters, T. Ohm, and R. Petras. The Rocky7 rover: A Mars sciencecraft prototype. In *Proceedings IEEE International Conference on Robotics and Automation*, pages 2458–2464, Albuquerque, New Mexico, April 1997.