Acta Astronautica 90 (2013) 268–288

Contents lists available at SciVerse ScienceDirect

Acta Astronautica

journal homepage: www.elsevier.com/locate/actaastro

Reusable science tools for analog exploration missions: xGDS Web Tools, VERVE, and Gigapan Voyage

Susan Y. Lee[a,*], David Lees[b], Tamar Cohen[a], Mark Allan[a], Matthew Deans[c,1], Theodore Morse[b], Eric Park[b], Trey Smith[b]

[a] *Stinger Ghaffarian Technologies, USA*
[b] *Carnegie Mellon University, USA*
[c] *NASA Ames Research Center, Moffett Field, CA 94035, USA*

A R T I C L E I N F O

Article history:
Received 25 June 2011
Received in revised form
22 November 2011
Accepted 7 January 2012
Available online 29 February 2012

Keywords:
NASA
Planetary analogs
Intelligent Robotics Group
Robotic Field Test
Desert Research and Technology Studies
Ground Data Systems
Gigapan Voyage
VERVE

A B S T R A C T

The Exploration Ground Data Systems (xGDS) project led by the Intelligent Robotics Group (IRG) at NASA Ames Research Center creates software tools to support multiple NASA-led planetary analog field experiments. The two primary tools that fall under the xGDS umbrella are the xGDS Web Tools (xGDS-WT) and Visual Environment for Remote Virtual Exploration (VERVE). IRG has also developed a hardware and software system that is closely integrated with our xGDS tools and is used in multiple field experiments called Gigapan Voyage. xGDS-WT, VERVE, and Gigapan Voyage are examples of IRG projects that improve the ratio of science return versus development effort by creating generic and reusable tools that leverage existing technologies in both hardware and software.

xGDS Web Tools provides software for gathering and organizing mission data for science and engineering operations, including tools for planning traverses, monitoring autonomous or piloted vehicles, visualization, documentation, analysis, and search. VERVE provides high performance three dimensional (3D) user interfaces used by scientists, robot operators, and mission planners to visualize robot data in real time. Gigapan Voyage is a gigapixel image capturing and processing tool that improves situational awareness and scientific exploration in human and robotic analog missions. All of these technologies emphasize software reuse and leverage open source and/or commercial-off-the-shelf tools to greatly improve the utility and reduce the development and operational cost of future similar technologies.

Over the past several years these technologies have been used in many NASA-led robotic field campaigns including the Desert Research and Technology Studies (DRATS), the Pavilion Lake Research Project (PLRP), the K10 Robotic Follow-Up tests, and most recently we have become involved in the NASA Extreme Environment Mission Operations (NEEMO) field experiments. A major objective of these joint robot and crew experiments is to improve NASAs understanding of how to most effectively execute and increase science return from exploration missions. This paper focuses on an integrated suite of xGDS software and compatible hardware tools: xGDS Web Tools, VERVE, and Gigapan Voyage, how they are used, and the design decisions that were made to allow

Abbreviations: IRG, Intelligent Robotics Group; GV, Gigapan Voyage; xGDS, Exploration Ground Data Systems; VERVE, Visual Environment for Remote Virtual Exploration; DRATS, Desert Research and Technology Studies; PLRP, Pavilion Lake Research Project; NEEMO, NASA Extreme Environment Mission Operations.

* Corresponding author. Tel.: +1 650 604 4725.

E-mail addresses: susan.y.lee@nasa.gov (S.Y. Lee), david.s.lees@nasa.gov (D. Lees), tamar.e.cohen@nasa.gov (T. Cohen), mark.b.allan@nasa.gov (M. Allan), matthew.deans@nasa.gov (M. Deans), theodore.f.morse@nasa.gov (T. Morse), eric.park@nasa.gov (E. Park), trey.smith@nasa.gov (T. Smith).

[1] Work done at NASA Ames Research Center.

doi:10.1016/j.actaastro.2012.01.002

them to be easily developed, integrated, tested, and reused by multiple NASA field experiments and robotic platforms.

1. Introduction

The Intelligent Robotics Group (IRG) [1] at NASA Ames Research Center is developing software and hardware tools for reuse in multiple analog planetary missions referred to as the Exploration Ground Data Systems (xGDS) suite for the purposes of this paper. Our xGDS work emphasizes the use of open data standards (e.g. KML and COLLADA) and open source software tools whenever possible to simplify integration and deployment of our software and hardware across multiple science missions and on multiple robotic platforms, including those developed by external collaborators. Specifically, Exploration Ground Data Systems Web Tools (xGDS-WT), Visual Environment for Remote Virtual Exploration (VERVE), and Gigapan Voyage are examples of xGDS-related technologies that continue to help improve the ratio of science return versus development effort for NASA's robotics program.

All the tools in the xGDS suite are designed to be flexible and therefore are capable of adapting to many different test scenarios. xGDS-WT provides software for collecting and presenting mission data for science and engineering operations, including tools for planning traverses, monitoring autonomous or piloted vehicles, visualization of data collected, documentation, analysis, and searching of collected data. VERVE provides high performance three dimensional (3D) user interfaces used by scientists, robot operators, and mission planners to visualize robot data in real time. Gigapan Voyage is another tool in the xGDS family that is a panorama capturing and processing technology to improve situational awareness for scientific exploration during human and robotic missions. It includes all the hardware and software necessary to choose panorama settings, create panoramas and to explore the stitched data products. The panoramas produced by Gigapan Voyage are compatible with the xGDS work flow. These technologies leverage open source and/or commercial-off-the-shelf tools to greatly improve the utility and reduce the development and operational cost of future similar technologies. They have already played a critical role in collecting and viewing data in multiple NASA led field campaigns over the past several years. These field campaigns include the Desert Research and Technology Studies (DRATS) [2], the Pavilion Lake Research Project (PLRP) [3], the K10 Robotic Follow-Up tests [4], and work has started to support the NASA Extreme Environment Mission Operations (NEEMO) [5] field experiment this year.

The following section will provide an overview of the various field experiments covered in this paper. Sections 3–5 will focus on each reusable technology within the xGDS suite in depth. We begin the technology sections by covering xGDS-WT and how it is used during various field tests. We then cover VERVE as another tool in the xGDS suite that provides 3D situational awareness. Finally, we cover Gigapan Voyage as both a tool built for reuse over multiple field experiments as well as an example data producer for the xGDS system. Each of these technology sections are broken up into four major subsections: "Overview", "Use cases", "Implementation", and "Field test use and results." In Section 6 we present our conclusion.

2. Recent field experiment overviews

This section provides a description of each of the field experiments mentioned in this paper focusing on how the xGDS technologies were used.

2.1. Desert Research and Technology Studies (DRATS)

The NASA Desert Research and Technology Studies (DRATS) project coordinates yearly field campaigns to analog planetary sites. Robotic assets that have participated in recent tests include the Space Exploration Vehicles, Centaur2, Robonaut2, K10, and the All-Terrain Hex-Limbed Extra-Terrestrial Explorer (ATHLETE). During these tests xGDS-WT is used to organize, prioritize, and present data to scientists, VERVE is used to help remote operators drive robotic vehicles, and Gigapan Voyage is used as one of the primary science instruments to provide situational awareness [2]. The field campaigns offer engineers, astronauts, and scientists from across the country an opportunity to come together to conduct technology development research in the Arizona desert (Table 1).

2.2. Pavilion Lake Research Project (PLRP)

The Pavilion Lake Research Project (PLRP) is an international, multidisciplinary, science and exploration effort to explain the origin of life in Pavilion Lake, British Columbia, Canada. For this test xGDS-WT is used to create submarine flight plans and, similarly to DRATS, organize data collected from the missions. The microbialites of these modern lakes are relevant to our understanding of ancient microbialites that were once common and diverse on early Earth.

2.3. Robotic Follow-Up

The Robotic Follow-Up field experiments took place in 2009 and 2010. In July of 2009, two geologists performed simulated lunar traverses to explore areas in, and around, Haughton Crater (Devon Island, Canada). One traverse focused on creating a map of the different geology features and structures. The other traverse focused on mapping subsurface structure using ground penetrating radar. To follow-up on these traverses, in 2010 IRG deployed the K10 planetary rover to the same area. K10 was equipped with a variety of science instruments, including ground penetrating radar, an XRF spectrometer, and a GigaPan panoramic imager. A science team and a flight control team sent xGDS-WT plans and remotely monitored K10 from the

Table 1
Recent NASA Analog Robotic Field Experiments and xGDS Technologies Involved. The specific goals of each analog field experiment vary, but the overarching purpose of these tests is to help NASA better understand how to safely, and cost effectively, conduct science experiments in space. For brevity, we have left out field tests that occurred prior to 2009, and have only included the technologies discussed in this paper in the "technologies used" column.

Field test	Led by	When	xGDS suite technologies used	Where
Desert Research and Technology Studies (DRATS)	Johnson Space Center	2009, 2010, 2011	xGDS-WT, VERVE, Gigapan Voyage	Black Point Lava Flow (Arizona)
Pavilion Lake Research Project (PLRP)	Ames Research Center and University of British Columbia	2009, 2010, 2011	xGDS-WT	Pavilion Lake (Canada)⁎ underwater
K10 Robotic Follow-Up Tests [6]	Ames Research Center	2009, 2010	xGDS-WT, VERVE, Gigapan Voyage	Black Point Lava Flow (Arizona), Haughton Crater (Canadian Arctic)
NASA Extreme Environment Mission Operations (NEEMO)	Johnson Space Center	2011	xGDS-WT, Gigapan Voyage	Key Largo (Florida)⁎ underwater

NASA Ames Research Center at Moffett Field, California using VERVE and the predecessor to xGDS-WT. The Robotic Follow-Up experiment was designed to test scenarios that could significantly increase the productivity and performance of future planetary exploration missions [6].

2.4. NASA Extreme Environment Mission Operations (NEEMO15)

Since 2001, the NASA Extreme Environment Mission Operations project has sent groups of NASA researchers to live in the Aquarius undersea research center for up to three weeks at a time. Aquarius operates 5.6 km (3.5 miles) off Key Largo in the Florida Keys National Marine Sanctuary. NEEMO 15 is the fifteenth such mission and is scheduled for October 17–26, 2011. This mission will test equipment and operations required for exploration of Near-Earth Asteroids (NEAs). xGDS-WT will be used for route planning, scheduling, and to help organize and view the data products and Gigapan Voyage is slated to be used underwater as a NEEMO science instrument for the first time [5]. For NASA, Aquarius provides a convincing analog to space exploration, and NEEMO crew members experience some of the same tasks and challenges underwater as they would in space.

Although it is clear that these four robotic analog tests differ greatly in their resources, goals, and challenges, they all have the common requirement of needing a way to provide situational awareness to the crews, remote science teams, and the engineering support teams. We will now go into detail about how our xGDS design and development team supported these tests by building core technologies that could be used across multiple missions.

3. Exploration Ground Data Systems Web Tools (xGDS-WT)

3.1. Introduction

Our Exploration Ground Data System Web Tools (xGDS-WT) provide a software interface for collecting and managing mission data for science operations, including tools for planning, monitoring, visualization, documentation, analysis, and searching. Our qualitative observations of xGDS-WT usage showed improvements in planning efficiency, crew and operator situation awareness, and ease of data browsing and searching at each field test we supported, and produced many operational lessons that we will apply to the next generation of our tools.

3.2. Use cases

The xGDS-WT use cases presented below are based on the steps that a science team typically encounters during the field experiments we have supported.

1. Planning: Planning of a mission starts with a priori map information including remote sensing data, known operational hazards or constraints, and targets of interest. xGDS-WT enables teams to create and share map content and export for tools such as Google Earth.
2. Monitoring execution: Execution monitoring is done via map-based tools to visualize the locations of assets (e.g. vehicle, robots, astronauts) in real-time during mission operations. Telemetry panels show the current status of systems and data. Real-time and post hoc documentation and annotation are also supported.
3. Archiving: xGDS-WT archiving tools ingest telemetry in real-time. After the data is in the system the data is reduced to more meaningful or more efficient representations and is organized into searchable databases.
4. Exploring: To explore the data after it is collected, the ability to quickly find out what data was collected, where and when it was collected, and search for particular kinds of data is required. Real-time semantic labeling of data greatly facilitates this, and we provide users with xGDS tools to add geolocated notes to data products and timelines during and after operations.

3.3. Implementation

All of our xGDS work, including xGDS-WT and VERVE, make significant use of open standards and open source software (see Table 2). This enables us to rapidly build

Table 2
Key xGDS-WT building blocks.

Component	Notes
Django	A flexible, open source Python-based web framework. All xGDS web applications are built on the Django framework
Python	A dynamic object-oriented programming language that supports rapid development and testing, as well as the creation of modular maintainable code
MySQL	Open source database
Apache	Open source web server software
Google Earth	A free 3D virtual globe program that provides detailed geographical information. xGDS uses both the plug-in and desktop client of this program. GE and web browsers are familiar but powerful platforms that help us provide xGDS users with rich data search and visualization tools without a lot of upfront training

scalable systems that are effective, easy to use, and easy to share with our collaborators. Some examples of open standards used in xGDS include HTML and KML. HTML content is ubiquitous and easily viewed. The KML [7] open standard simplifies creation, distribution, and visualization of map-based content. xGDS also leverages Google Earth, a tool that our science and operations teams already use. This minimizes development and training costs.

We use open source tools rather than implementing functionality ourselves whenever possible [8]. We build our web applications using the Apache web server, the MySQL database, the Django web framework for Python, and the jQuery JavaScript library. Our Java based client applications (e.g. VERVE) build on the modular Eclipse Rich Client Platform (RCP) and NASA's Ensemble framework [9]. This allows us to focus on writing software that is specific to our applications and the glue code between standard open source modules and tools rather than the modules themselves. We release our own tools as open source when possible, to more easily share code with colleagues and students. Potential users can evaluate our tools with minimal overhead.

xGDS-WT was designed to be easily customized to the requirements of different NASA analog field tests. Most notably, nearly all field applications involve common concepts (e.g. tracking the position of various assets such as people and vehicles, geocoding the locations of data products and collecting notes and observations from the operations and science teams). However, in most cases, the details of how this information is gathered can be quite different. For example, the interface to the tracking system for the submersibles used in PLRP or NEEMO is entirely different from the tracking system for the SEV at DRATS. To support these differences xGDS is designed in three key stages (see Fig. 1): (1) A data import stage which converts data from the native format delivered by the various field assets into a common format for storage and archiving. (2) A data storage/management stage that takes the output from stage 1 and archives data products and metadata in a database for later use by the science and operations teams. (3) A web-based data searching and exploration application, which gives the science and operations teams access to the data archive created by xGDS to support scientific analysis and to plan future operations (Figs. 2–4).

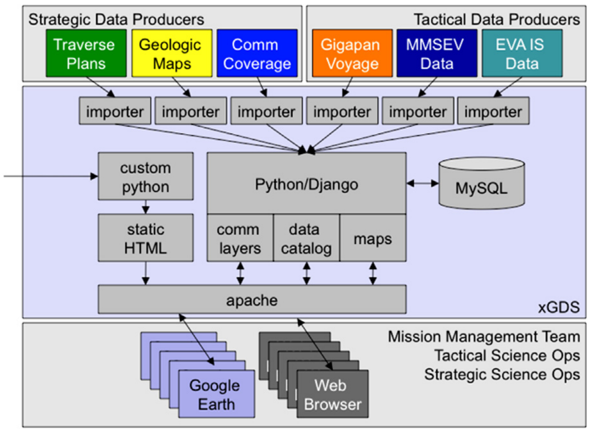
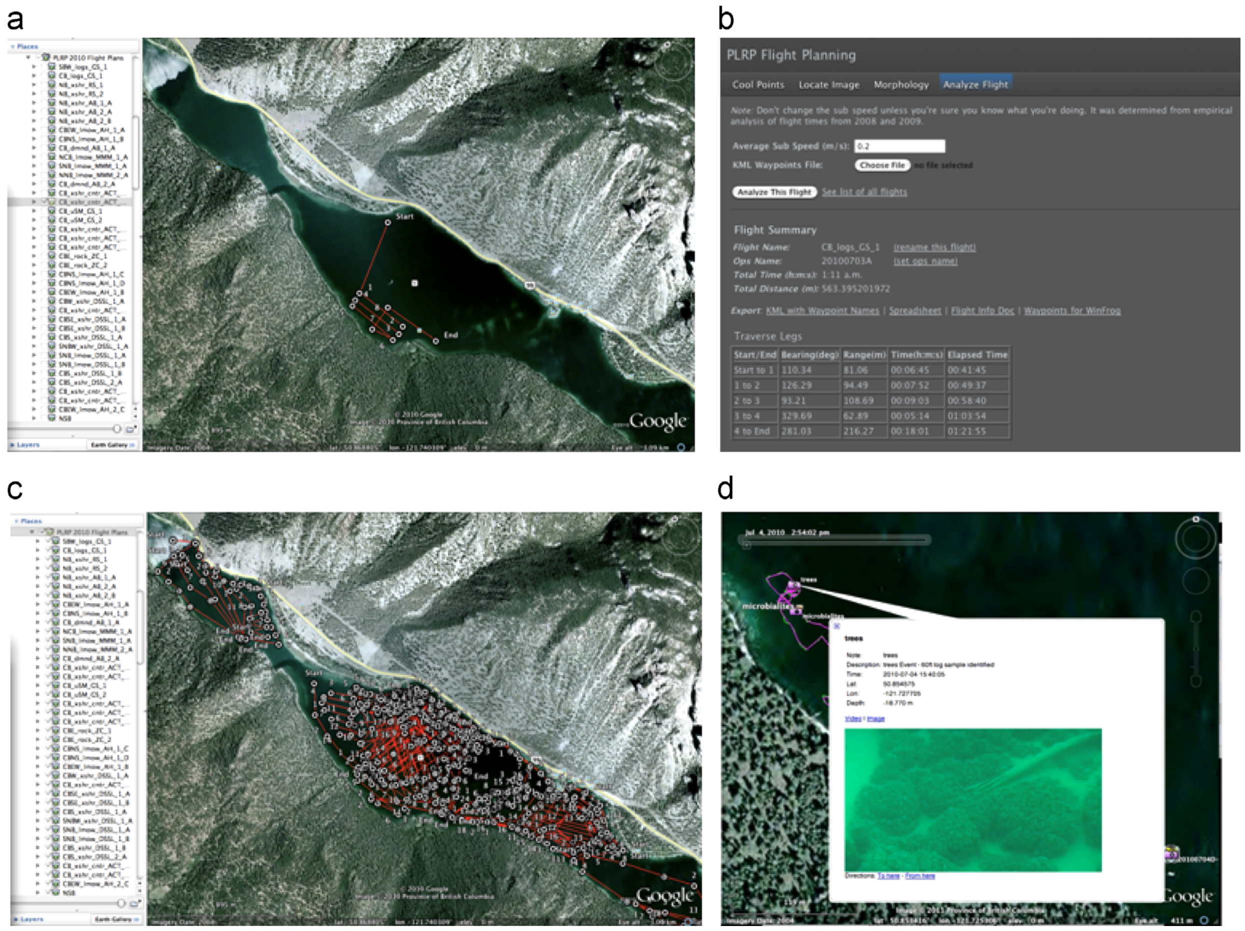


Fig. 1. The xGDS-WT architecture leverages open standards and open source software libraries to create a modular, flexible, easy to build and use Ground Data System.

3.4. Field test use and results

A primary objective of our field test deployments was to evaluate the effectiveness of our xGDS-WT tools for planning, monitoring, archiving, and searching in real operational environments.

At DRATS 2010 [2] there were two science teams, Tactical and Strategic, working in day/night shifts. xGDS-WT provided tools for Tactical science operations, including live tracking of the rover and crew locations and automatically cataloging data products. xGDS-WT also supported the Strategic team's overnight science operations, including analyzing the day's data and planning operations for the next day. The data catalog and search tools in xGDS-WT allowed the science team to explore rover telemetry, suit telemetry, videos and Gigapan Voyage panoramas collected by the rover and EVA crew. For PLRP [3] (see Fig. 2), we supported traverse planning, execution monitoring, data archiving, and data search. Traverse plans for 2010 were developed using Google Earth and uploaded to xGDS-WT server for analysis, cataloging, and sharing. PLRP 2011 used xGDS-WT directly for iterative browser-based planning. Real time science support was done on chase boats during suboperations. Post-mission analysis was done daily, with flight data archived and reviewed the same day it was collected. The data cataloged for the PLRP science team consisted of submarine telemetry and underwater video and audio. NEEMO 15 will be supported very similarly to PLRP 2011. The submersibles used and their telemetry system will be identical to PLRP. The xGDS-WT planning tool was designed to support map-based planning for any location and will be reused with no changes for NEEMO. In addition to planning support we will provide mission data cataloging and post-flight video processing using the same tools deployed at PLRP 2011 (Fig. 5).



PLRP Flight Planning

Cool Points | Locate Image | Morphology | Analyze Flight

Note: Don't change the sub speed unless you're sure you know what you're doing. It was determined from empirical analysis of flight times from 2008 and 2009.

Average Sub Speed (m/s): 0.2

KML Waypoints File: Choose File no file selected

Analyze This Flight See list of all flights

Flight Summary

Flight Name: CB_logs_GS_1 (rename this flight)

Ops Name: 20100703A (set ops name)

Total Time (h:m:s): 1:11 a.m.

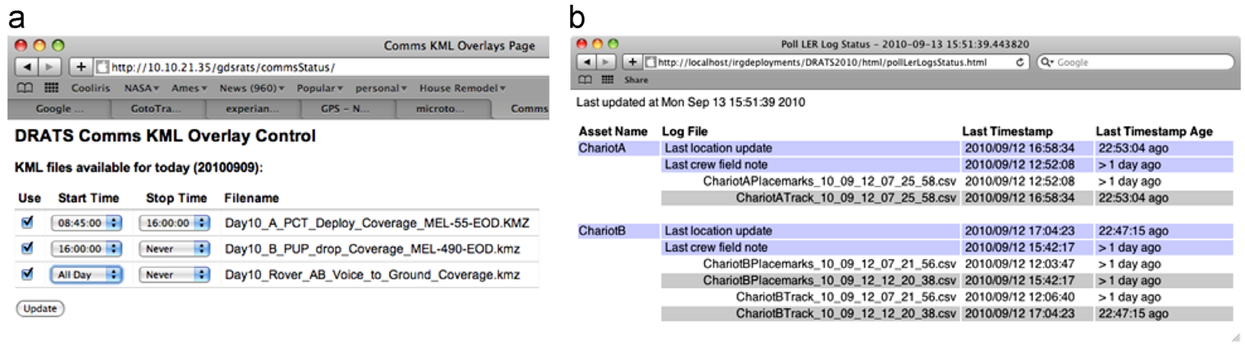
Total Distance (m): 563.395201972

Export: KML with Waypoint Names | Spreadsheet | Flight Info Doc | Waypoints for WinFrog

Traverse Legs

Start/End	Bearing(deg)	Range(m)	Time(h:m:s)	Elapsed Time
Start to 1	110.34	81.06	00:06:45	00:41:45
1 to 2	126.29	94.49	00:07:52	00:49:37
2 to 3	93.21	108.69	00:09:03	00:58:40
3 to 4	329.69	62.89	00:05:14	01:03:54
4 to End	281.03	216.27	00:18:01	01:21:55

Fig. 2. PLRP planning. PLRP used a web-based planning system where scientists drew submarine traverse plans ("flight plans") using KML in Google Earth and submitted the flight plan KML to a web-based analysis tool to calculate the total traverse distance and duration of their flight. The flight planner tool could also export annotated KML of a flight and waypoints to be used by the chase boat navigation system during operations. The bottom panel shows all 2010 PLRP flight plans in a single map.


a
b
Comms KML Overlays Page
Cooliris NASA Ames News (960) Popular personal House Remodel
Google ... GotoTra... experian... GPS – N... microto... Comms
Update
Poll LER Log Status – 2010-09-13 15:51:39.443820
Share

http://10.10.21.35/gdsrats/commsStatus/

DRATS Comms KML Overlay Control

KML files available for today (20100909):

Use	Start Time	Stop Time	Filename
✓	08:45:00	16:00:00	Day10_A_PCT_Deploy_Coverage_MEL-55-EOD.KMZ
✓	16:00:00	Never	Day10_B_PUP_drop_Coverage_MEL-490-EOD.kmz
✓	All Day	Never	Day10_Rover_AB_Voice_to_Ground_Coverage.kmz

http://localhost/irgdeployments/DRATS2010/html/pollLerLogsStatus.html

Last updated at Mon Sep 13 15:51:39 2010

Asset Name	Log File	Last Timestamp	Last Timestamp Age
ChariotA	Last location update	2010/09/12 16:58:34	22:53:04 ago
	Last crew field note	2010/09/12 12:52:08	> 1 day ago
	ChariotAPlacemarks_10_09_12_07_25_58.csv	2010/09/12 12:52:08	> 1 day ago
	ChariotATrack_10_09_12_07_25_58.csv	2010/09/12 16:58:34	22:53:04 ago
ChariotB	Last location update	2010/09/12 17:04:23	22:47:15 ago
	Last crew field note	2010/09/12 15:42:17	> 1 day ago
	ChariotBPlacemarks_10_09_12_07_21_56.csv	2010/09/12 12:03:47	> 1 day ago
	ChariotBPlacemarks_10_09_12_12_20_38.csv	2010/09/12 15:42:17	> 1 day ago
	ChariotBTrack_10_09_12_07_21_56.csv	2010/09/12 12:06:40	> 1 day ago
	ChariotBTrack_10_09_12_12_20_38.csv	2010/09/12 17:04:23	22:47:15 ago

Fig. 3. DRATS operations monitoring. Because the DRATS rovers covered large distances, it was important to monitor their communications coverage. Coverage maps were automatically updated on the operations teams map view during the day. A status page was generated so both the science and operations teams would be aware of the communications status of the rovers (b).

For our K10 rover deployments [10], we had a real time science backroom and flight control room at NASA Ames, connected to the robot via satellite. Strategic planning was done in advance with support from Google Earth and xGDS-WT. xGDS-WT provided tools to develop many tactical robot plans per day, monitor the robot in real time, and archive and browse science data in real time and after the mission. Data collected by K10 and displayed and cataloged by xGDS-WT included robot telemetry and data from the panoramic camera (a version

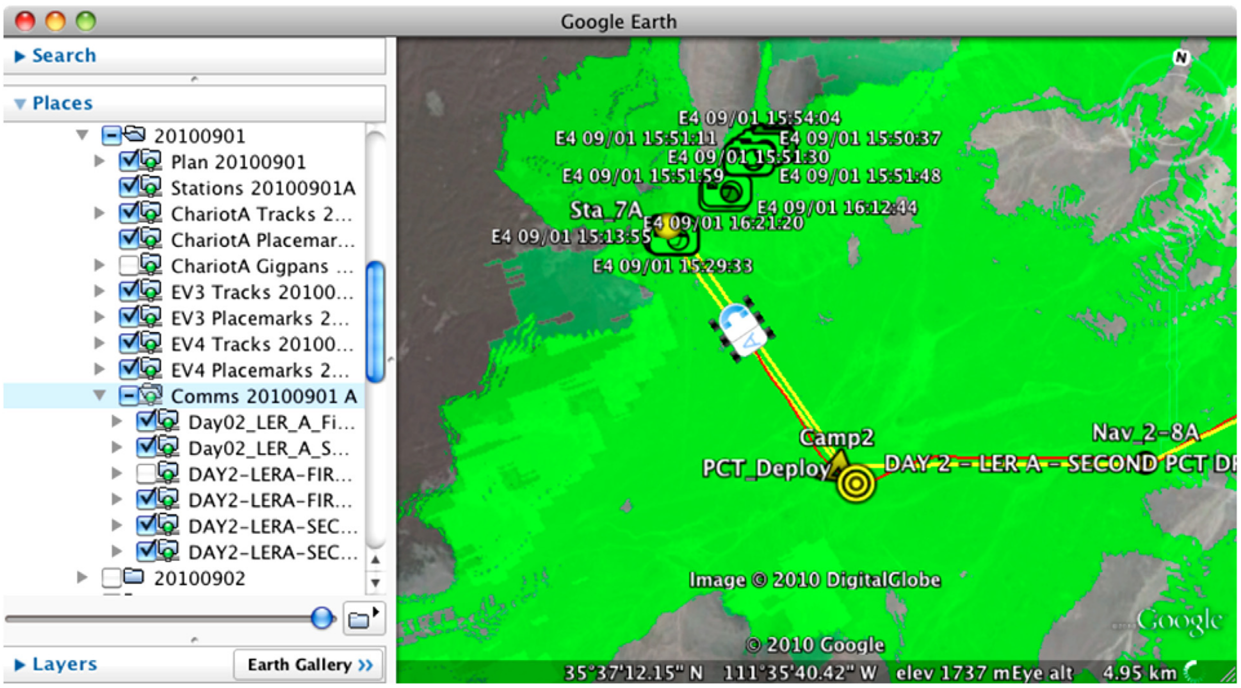


Fig. 4. DRATS operations monitoring. The rovers location, communication coverage and the location of data products collected during operations were monitored in real-time.

of Gigapan Voyage deployed on the rover), a 3D Lidar, ground penetrating radar, an X-ray Fluorescence (XRF) spectrometer, and a downward-facing microscopic imager.

As described previously in this paper, the route planning process for PLRP 2010 was to develop paths for routes in Google Earth, and upload these as KML files to the xGDS-WT server for analysis and file conversion. This process would take the scientists at least 2–3 h per plan. We also found that for the K10 Robotic Follow-Up experiments it would require a dedicated member of the xGDS team to assist scientists with their use of the Google Earth and Python-based planning tool. Often the scientists would want to do tasks such as adjust existing plans, reverse plans, overlay plans with each other and merge plans.

We used feedback from these 2010 field experiments to create the newer xGDS-WT planning tool. This is integrated directly in the xGDS-WT website, and utilizes the Google Earth plug-in along with JavaScript code to create route plans. xGDS users can easily collaborate with others on these plans, and turn on layers which show scientific/mapped data, tracks from routes taken and other plans.

Not only do we support editing plans, reversing and merging plans; we dynamically calculate the timing for the entire plan plus each segment and waypoint on the plan. Users can enter notes throughout the plan plus input detailed scientific documents describing the plans. Route planning duration went from 2 to 3 h per plan to 5–10 min per plan, and was so effective that during PLRP 2011 scientists would replan right up to when a flight was taking off.

One of the changes for PLRP 2011 was the introduction of a tether from one of the subs, providing video and audio feed all the way back to the Mobile Mission Control Center (MMCC) trailer. Instead of limiting note-taking to one science stenographer on the chase boat, xGDS-WT provided web-based note-taking tools. Working directly off the same database where the science stenographer's notes and the submarine's track would be uploaded at the end of the flight, backroom scientists used the web interface to create and edit notes for flights with the video feed. We went from about 20 to 40 science steno notes per subflight to upwards of 250.

Sometimes there would be an artificial 50 s delay introduced to the video and audio feed to simulate delays that would occur during lunar surface operations. The Capsule Communicator, the only person who can communicate directly with crew, would set the xGDS-WT note taking tool into delay mode and all of the times logged in the xGDS-WT tool would be automatically adjusted to compensate for the delay. This was a smooth, seamless workflow for the science team (Figs. 6 and 7).

Scientists in the backroom could integrate information gleaned from watching the live video feed with information from other subflights to make recommendations for new route plans and dive sampling sites.

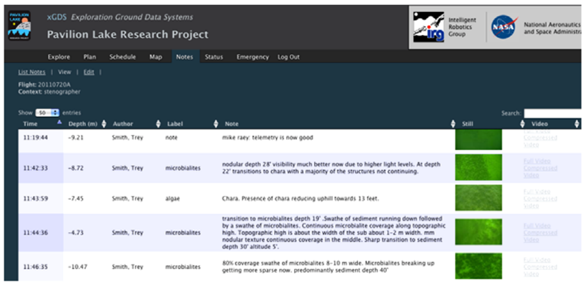
Since we had the tether in one of the submarines in 2011 we could provide a map feed so that the pilot would have the same contextual information that the CapCom had about their location and heading relative to the route they were on. We utilized the laptop that was in the sub and set it up to run Google Earth, to show them their current position and information about where they should go. This was an example of reusing technology we had (route and track generation) to provide more situational awareness to participants of the field experiment (Figs. 8 and 9).

In conclusion, despite seeming like a useful tool, this turned out to confuse the pilots because of several factors. The pilots of the subs have a very high task load. There is a large margin of error in terms of where the sonar system reports the submersible to be at a given time, so often the accuracy of the map is not fine enough for the pilot. It is also very crowded in the submersible, and the laptop would have to be extended over the pilot's lap. Future ideas include simplifying the display and mounting a small Android phone to the dashboard in the sub for a better and more compact viewing position.

Overall our experience with field testing xGDS-WT showed:

- Improved situational awareness: the centralized nature of the web-based tools allowed the entire science and operations team to see the latest operations plan and vehicle status for K10 tests and at PLRP 2011. In contrast, in at DRATS and at PLRP prior to 2011, operations plans were e-mailed or delivered on USB sticks and a fair amount of time was spent coordinating amongst team member to ensure that the proper plan was used.
- Improved collaboration: again, the centralized architecture of the planning tool allowed a geographically distributed team to create operations plans for PLRP even though they were working in different time zones and were only occasionally available for in-person meetings or teleconferences.
- Rapid turnaround data processing and analysis for operations and science: one of the goals of our xGDS tools is to place data in context quickly enough to

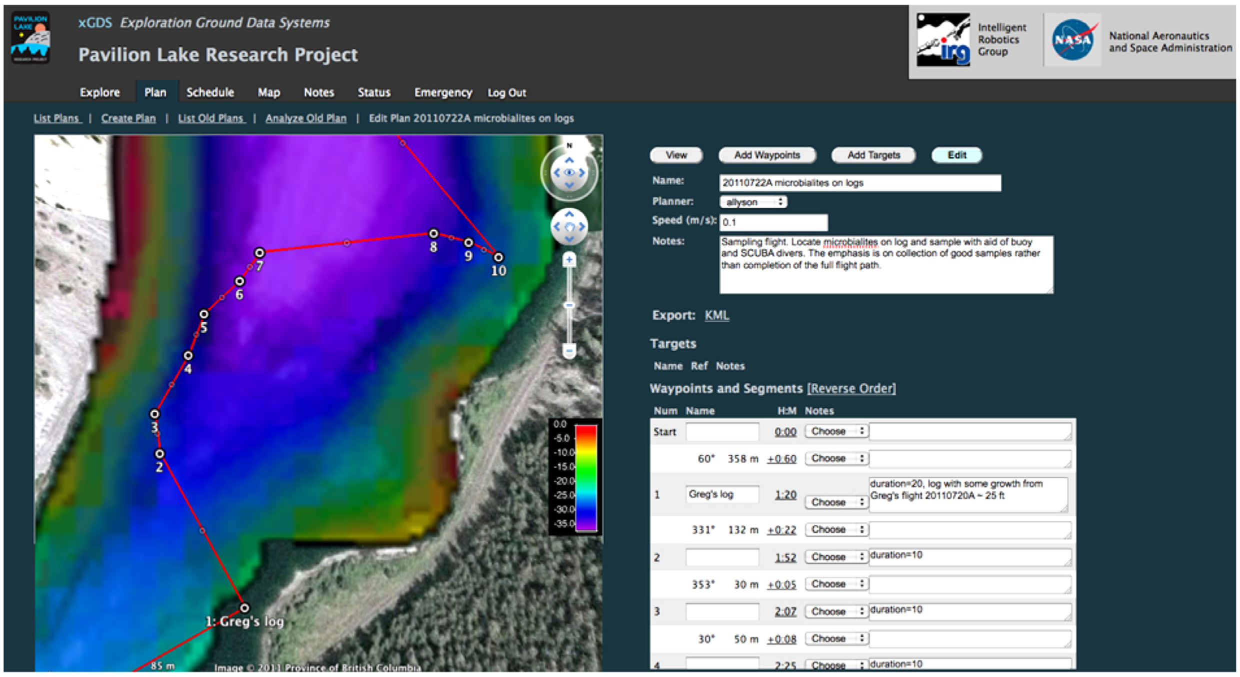
Fig. 6. Scientists in the backroom (MMCC) watching video feed and using xGDS-WT to take notes.


xGDS Exploration Ground Data Systems
Pavilion Lake Research Project
Intelligent Robotics Group
National Aeronautics and Space Administra
Explore
Plan
Schedule
Map
Notes
Status
Emergency
Log Out
List Notes | View | Edit |
Show 50 entries
Search:

Flight: 20110720A
Context: stenographer

Time	Depth (m)	Author	Label	Note	Still	Video
11:19:44	-9.21	Smith, Trey	note	mike raey: telemetry is now good	🟩	Full Video Compressed Video
11:42:33	-8.72	Smith, Trey	microbialites	nodular depth 28' visibility much better now due to higher light levels. At depth 22' transitions to chara with a majority of the structures not continuing.	🟩	Full Video Compressed Video
11:43:59	-7.45	Smith, Trey	algae	Chara. Presence of chara reducing uphill towards 13 feet.	🟩	Full Video Compressed Video
11:44:36	-4.73	Smith, Trey	microbialites	transition to microbialites depth 19' .Swathe of sediment running down followed by a swathe of microbialites. Continuous microbialite coverage along topographic high. Topographic high is about the width of the sub about 1-2 m width. mm nodular texture continuous coverage in the middle. Sharp transition to sediment depth 30' altitude 5'.	🟩	Full Video Compressed Video
11:46:35	-10.47	Smith, Trey	microbialites	80% coverage swathe of microbialites 8-10 m wide. Microbialites breaking up getting more sparse now. predominantly sediment depth 40'	🟩	Full Video Compressed Video

Fig. 7. Screenshot of note-taking interface after video and tracking data have been uploaded and processed. This same interface is used to edit notes.


xGDS Exploration Ground Data Systems
Pavilion Lake Research Project
Intelligent Robotics Group
National Aeronautics and Space Administration
Explore
Plan
Schedule
Map
Notes
Status
Emergency
Log Out
List Plans | Create Plan | List Old Plans | Analyze Old Plan | Edit Plan 20110722A microbialites on logs
N
1: Greg's log
2
3
4
5
6
7
8
9
10
0.0
-5.0
-10.0
-15.0
-20.0
-25.0
-30.0
-35.0
85 m
Image © 2011 Province of British Columbia
View
Add Waypoints
Add Targets
Edit
Export: KML

Name: 20110722A microbialites on logs
Planner: allyson
Speed (m/s): 0.1
Notes: Sampling flight. Locate microbialites on log and sample with aid of buoy and SCUBA divers. The emphasis is on collection of good samples rather than completion of the full flight path.

Targets

Name	Ref	Notes

Waypoints and Segments [Reverse Order]

Num	Name	H:M		Notes
Start		0:00	Choose	
	60° 358 m	+0:60	Choose	
1	Greg's log	1:20	Choose	duration=20, log with some growth from Greg's flight 20110720A ~ 25 ft
	331° 132 m	+0:22	Choose	
2		1:52	Choose	duration=10
	353° 30 m	+0:05	Choose	
3		2:07	Choose	duration=10
	30° 50 m	+0:08	Choose	
4		2:25	Choose	duration=10

Fig. 5. xGDS-WT planning tool for PLRP 2011. This shows integrated web-based plan creation and editing, with bathymetry data overlaid in the map, waypoint and segment timing and detailed information on the right hand side.

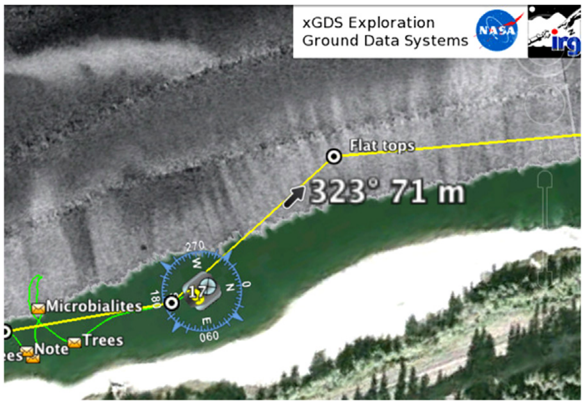


Fig. 8. Google Earth display within the sub, showing route plan (yellow), flown track (green), stenographer notes (envelopes), current sublocation and heading, and arrow to show heading and distance to next waypoint. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

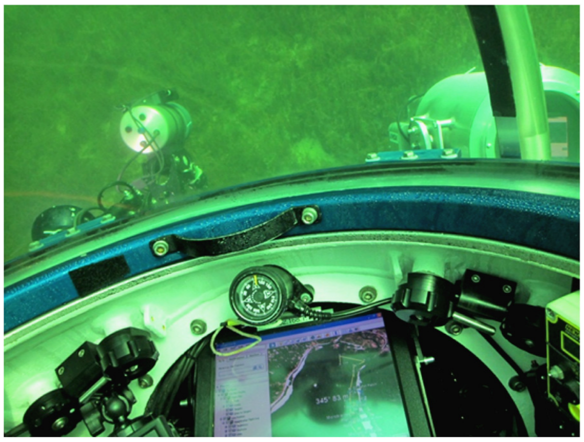


Fig. 9. Shot from inside the sub, you can see Google Earth on the laptop towards the bottom of the shot. Note how crowded it is.

allow a science team to adapt to new results and replan during operations. For K10 and PLRP tests, xGDS was used to allow the science team to monitor operations live, view data in a map and plan new data gathering operations on the same day. For DRATS 2010, data was processed and delivered to the overnight strategic science team (Fig. 10) within 2–3 h of the end of a day's operations and used to generate new operations plans by the following morning. In contrast, prior to the use of xGDS tools, the video data collected at PLRP field tests could not be reviewed in depth until after the field season was complete, and the results could not be used to guide operations and data collection until the following field season, almost a year later.

4. VERVE

4.1. Overview

The next technology in the xGDS suite we will cover is VERVE. VERVE provides high performance 3D user interfaces used by scientists, robot operators, and mission planners to visualize robot data in real time. It addresses situations (e.g. high bandwidth robot control and monitoring or full 3D visualization) where web-based tools like xGDS-WT cannot meet our operational requirements. We employ a flexible, component-based software architecture to support a wide variety of targeted, end-user applications and promote open development collaborations.

4.2. Use cases

One of the primary ways the IRG team uses VERVE is to support development of rovers. As we develop new algorithms that we use for rover navigation (Rover Software) it is critical to be able to visualize the executing logic on the rover. Rover Software developers can refine and tune software algorithms and verify that the rover is navigating as they expect.

During the Robotic Follow-Up experiments at Haughton Crater, VERVE was used simultaneously in three places. First, the robot engineers and operators brought VERVE to the field. They used VERVE to monitor the K10 rovers during all check out and operations. Local monitoring was critical as there was severely limited connectivity between the field site and the control rooms at NASA Ames. With VERVE, robot operators could easily see the location and state of the robot and quickly diagnose any issues.

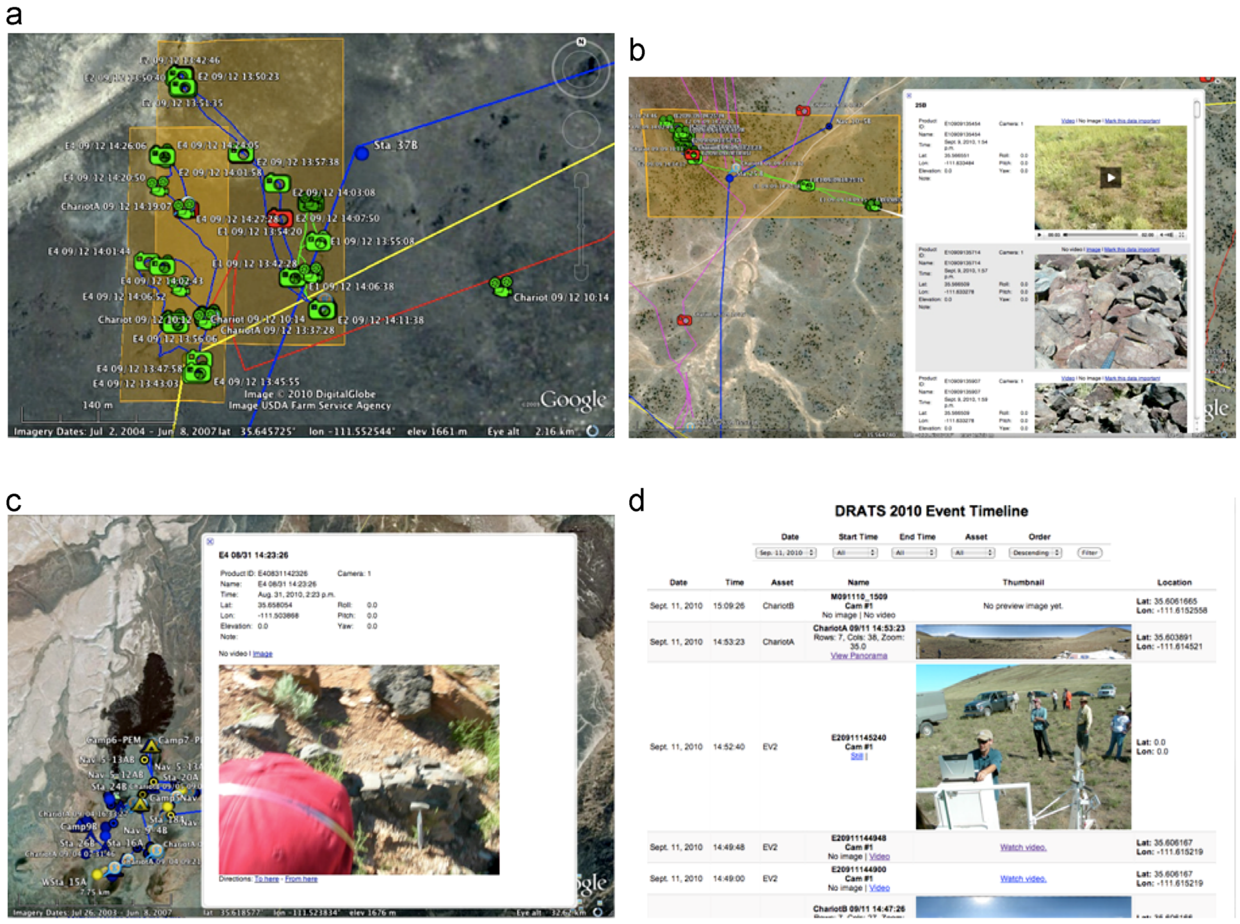
VERVE was also used in the Flight Room and in the Science Back Room. In the Flight Room, specific staff was assigned to monitor VERVE to ensure the rover was behaving as expected, and so they could communicate with field staff with full awareness of the rover's health and location. Since VERVE also supports loading and viewing KML files, information from the xGDS-WT tools was displayed embedded within the VERVE 3D View.

Lastly, VERVE was deployed in the Science Back Room. In this area, scientists were analyzing data retrieved with the xGDS-WT tools, and using the xGDS tools to create surveys for the rover. VERVE was important for the scientists to understand the health of the rover and to learn which instruments were available at a given time. With this information they could effectively use the xGDS-WT tooling to create new surveys.

4.3. Implementation

VERVE is implemented in Java and built on top of the Eclipse Rich Client Platform (RCP) [11] and Ardor3D [12] graphics engine. Since VERVE is a Java application running locally on end user computers, we have more control over user interface components and can achieve better performance than would be possible via a web application. Additionally, because VERVE subscribes to robot telemetry directly, end users are not dependent upon a centralized web server.

The Eclipse RCP is an open source framework, implemented as a set of deployed plug-ins that are OSGi [10] compliant. Plug-in components can be combined in various ways to create customized applications that share functionality and



E4 08/31 14:23:26

Product ID:	E40831142326	Camera:	1
Name:	E4 08/31 14:23:26		
Time:	Aug. 31, 2010, 2:23 p.m.		
Lat:	35.658054	Roll:	0.0
Lon:	-111.503868	Pitch:	0.0
Elevation:	0.0	Yaw:	0.0
Note:			

No video | Image

Directions: To here - From here

DRATS 2010 Event Timeline

Date: Sep. 11, 2010 · Start Time: All · End Time: All · Asset: All · Order: Descending · Filter

Date	Time	Asset	Name	Thumbnail	Location
Sept. 11, 2010	15:09:26	ChariotB	**M091110_1509** **Cam #1** No image | No video	No preview image yet.	**Lat:** 35.6061665 **Lon:** -111.6152558
Sept. 11, 2010	14:53:23	ChariotA	**ChariotA 09/11 14:53:23** Rows: 7, Cols: 38, Zoom: 35.0 View Panorama		**Lat:** 35.603891 **Lon:** -111.614521
Sept. 11, 2010	14:52:40	EV2	**E20911145240** **Cam #1** Still |		**Lat:** 0.0 **Lon:** 0.0
Sept. 11, 2010	14:49:48	EV2	**E20911144948** **Cam #1** No image | Video	Watch video.	**Lat:** 35.606167 **Lon:** -111.615219
Sept. 11, 2010	14:49:00	EV2	**E20911144900** **Cam #1** No image | Video	Watch video.	**Lat:** 35.606167 **Lon:** -111.615219
			ChariotB 09/11 14:47:26		

Fig. 10. DRATS data exploration. Clicking in an orange box (which represented an EVA site) opens a placemark bubble with a table of all the data products at that site. This feature is useful wherever dense map data must be analyzed. Because DRATS had two separate science teams in 2010, a timeline view of the data was valuable for the overnight team, so they could review the data collected during the day in the order in which it was gathered. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

reuse code. We leverage this feature to create versions of VERVE tailored to meet the needs of myriad deployments. Although some deployments consist entirely of VERVE components, it is common for VERVE to be integrated with components from other projects, such as in the RAPID Workbench [13] and ATHLETE FootFall [14].

Eclipse RCP applications make use of the local operating system's native user interface controls which provides a seamless, familiar, and efficient experience for our users. Further, Eclipse RCP applications support customization by the end-user of the views that are shown. We can therefore provide many independent views which display information for different domains, and users can decide what views they are interested in at a given time. VERVE supports multiple instances of almost all views, so users can easily monitor multiple robots simultaneously. VERVE views can also be extended to support newly developed robots and systems with minimal effort. These views provide critical information for operating a remote robot when it is important to know as much information as possible about where the robot is, what it is doing, and what it is about to do.

One the most useful views in VERVE are the 3D View. This view builds upon the open source Ardor3D graphics engine to provide robot operators and scientists a 3D representation of the scene they are exploring. A typical VERVE 3D View scene includes an articulated robot avatar, graphical representations of sensor data and robot

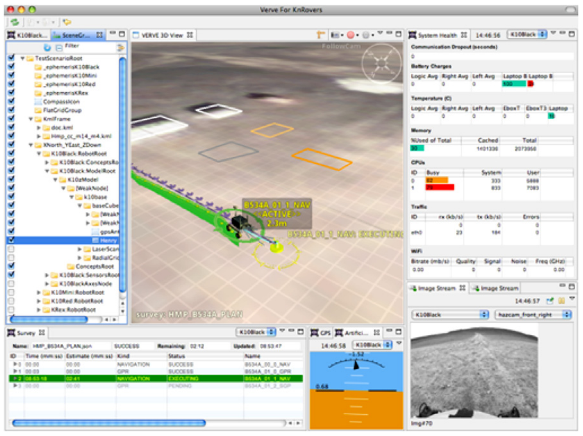


Fig. 11. VERVE monitoring K10 Black, Haughton Crater.

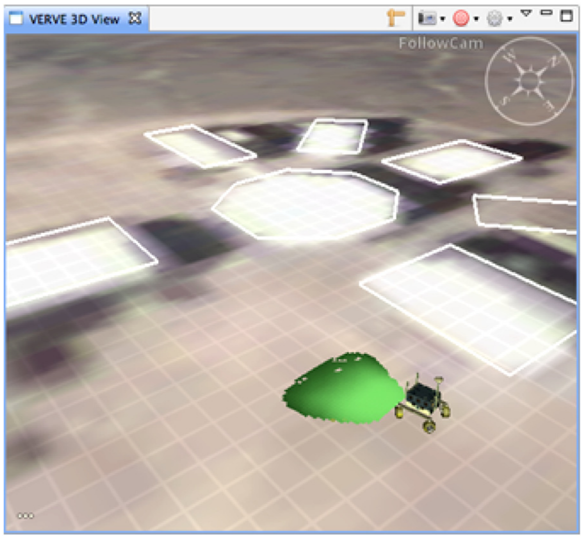


Fig. 12. VERVE 3D View showing KML terrain with markup and K10 Black Rover with traversability map (green). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

state, and terrain maps for context. Ardor3D is an Open Source retained mode scene graph for Java. The API offers full access to modern rendering features as well as compelling performance in comparison to other Java scene graphs. The project enjoys an active developer community that has grown steadily since the code was publicly released in 2009. OpenGL is used as the underlying rendering layer, and Ardor3D integrates well with Eclipse/SWT, Swing and native canvases.

The VERVE 3D View can support an arbitrary number of robots in the scene, and multiple views can be opened to monitor the scene from different angles or to focus on individual robots (Figs. 11 and 12). It also provides several camera controllers for navigating around the 3D scene using keyboard and mouse input. EarthCam offers control similar to Google Earth, the NadirCam offers top-down, plan view navigation, the FollowCam is optimized for rotating around and inspecting an object of interest, and the FirstPersonCam offers navigation control similar to many 3D video games. The FollowCam and NadirCam allow a user to select an object of interest (typically a robot) and the view will follow that object without any additional user input. Additionally, the avatars can define properties that give hints to the camera controllers to enable specialized behavior such as automatically orienting the camera so that the robot is in the foreground and the next waypoint is in the background at all times.

For the 3D View, robot avatars are either created at application startup, or upon discovery of robot telemetry. Each robot avatar consists of a collection of "Parts", where each Part binds a telemetry message to some action in the scene graph. In the simple case of a pose estimate message, the corresponding action would be to update the location and orientation of the robot model in the scene. More sophisticated actions could include modifying textures, creating new geometry, or altering shader parameters. Parts are reusable for all robots that share a common telemetry type, and actions are reusable for all robots. In order to support a new telemetry type, a set of Parts must be written to bind telemetry messages to actions.

As part of the situational awareness provided by the 3D View, VERVE also supports large scale terrain rendering through the use of geometry [15] and texture [16] clipmaps. Because geometry clipmaps generate geometry at runtime, we are able to load GeoTIFF DEMs directly and composite multiple DEMs without any preprocessing. For large terrain data sets that cannot fit into memory, we use tools from the GDAL [17] software suite to build tile pyramids. DEM tiles are loaded on demand, which significantly reduces memory requirements. Orthographic satellite imagery is handled in a similar manner for the texture clipmaps. As with the DEMs, data sets are arranged in layers and images of differing spatial resolution can be composited at runtime.

The 3D View additionally supports loading several common 3D file formats, such as COLLADA, VRML, OBJ, and others. VERVE also supports a subset of the OpenGIS KML [7] file format commonly used by Google Earth and other GIS software (Figs. 13 and 14).

OpenGIS KML is an XML based file format defined by a schema that provides an extension mechanism though the use of vendor namespaces. Because the KML specification is so large and continually evolving, we make use of the Eclipse Modeling Framework (EMF) to streamline the software development process. We have created an EMF Model based on the OpenGIS KML and Google KML Extensions schema. From the EMF Model, we auto-generate Java code to read and write KML files. When the OpenGIS KML or Google Extensions schemas are updated, we load the schema changes into the EMF model and regenerate our Java code to reflect those changes. We have built our own Java classes to construct Ardor3D scene graph elements based on the EMF models loaded from the KML files.

The KML definition includes a "Network Link," a reference to an external KML file that can be automatically refreshed at specified intervals. Network Links are commonly used in Google Earth to reference dynamic data and view animations. We provide support for Network Links in VERVE, enabling background reloading of KML files and selective updating of the scene graph in KML nodes that have changed.

KML is fundamentally important to VERVE as a data interchange format. Most of the scientists that we work with are familiar with Google Earth and use it as a research tool. KML is also used heavily by the xGDS-WT tool chain, as described earlier in this paper. Although there is some overlap in functionality between Google Earth and VERVE, it should be noted that it is not our intent to reimplement all of the features of Google Earth. Our goal is to provide a high fidelity common operating picture of robotic field activities, and KML offers a flexible and powerful means to exchange relevant data between our tools.

VERVE also includes tools for displaying robot properties, viewing incoming camera imagery, and route planning. VERVE can subscribe to telemetry from many different

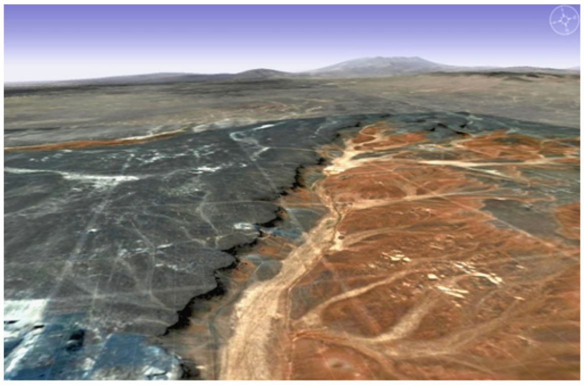


Fig. 13. Black Point Lava Flow terrain clipmap, with composite DEMs, orthoimagery, and grid overlay.

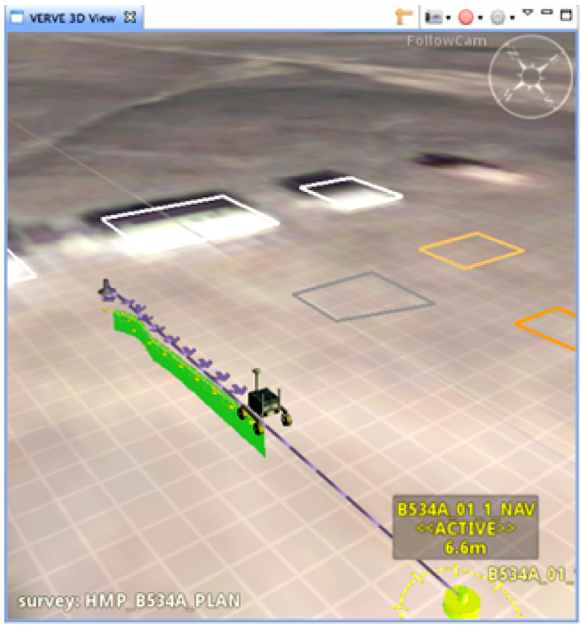
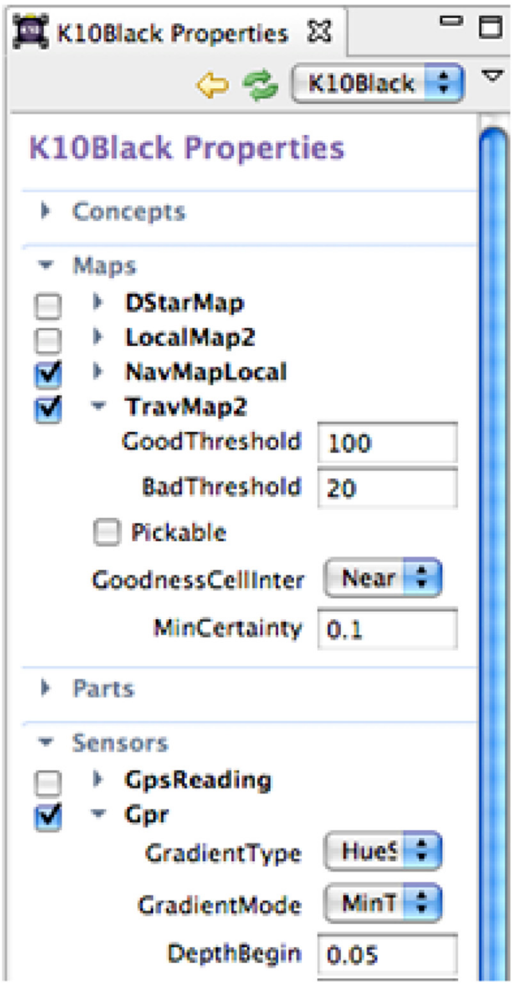


Fig. 14. VERVE 3D View showing Ground Penetrating Radar (GPR), survey path, waypoints and state.



K10Black Properties

K10Black

K10Black Properties

Concepts

Maps

☐	DStarMap
☐	LocalMap2
✓	NavMapLocal
✓	TravMap2
GoodThreshold	100
BadThreshold	20
☐	Pickable
GoodnessCellInter	Near
MinCertainty	0.1

Parts

Sensors

☐	GpsReading
✓	Gpr
GradientType	HueS
GradientMode	MinT
DepthBegin	0.05

Fig. 15. Properties View for K10 Black.

robots and instruments, and as a platform can be extended to support even more robots and instruments. We provide a dynamically generated set of controls to tune the rendering of the telemetry.

As VERVE is extended to support different robots, and as existing robots change over time, the dynamically generated Robot Properties View automatically gives the user control to display or hide information in the VERVE 3D View. Users can also tune how this information is drawn in the VERVE 3D View, setting thresholds, colors, and anything else that can be changed. Each robot that VERVE is listening to has its own Robot Properties View.

The content of the Robot Properties View (Fig. 15) is generated based on the definition in Java of the robot avatars. Java Reflection is used to explore the Java packages, member classes and public methods. Generated user interface components are grouped by package name (i.e. ‘Concepts’, ‘Maps’, etc.). We then employ Eclipse databinding to connect the user interface components directly to the instance of the avatar. This way, as soon as the user changes a value in the Robot Properties View, the avatar in the VERVE 3D View will update to reflect the changes. Conversely, if any changes happen to the avatar automatically, the Robot Properties View's values will update. As VERVE is extended to support new types of robots, the Robot Properties View will be automatically generated, populated and bound to the avatar. This greatly simplifies extending VERVE to support different robots.

One example of rover specific views can be demonstrated by IRG's Kn-Series rovers. The Kn-Series rovers use several cameras, and send imagery from the cameras over the telemetry stream. VERVE Image Stream View supports rendering image data. This can be very helpful for getting a clear idea of where the robot is while all the other information is being displayed in the other views. Note the local time of the last update is displayed in the view. We have also created a PanCam View that supports quickly changing pan, tilt, zoom and exposure of the Pan Tilt Camera used for taking GigaPans.

Other telemetry views show detailed tabular data, highlighting particular rows or cells when an important change has happened. For each type of telemetry, there are different thresholds and states that indicate a good status or a problem. For example, a low battery charge is a problem but a low CPU usage is good. The particular thresholds are customized per type of telemetry.

Another very important aspect of VERVE is its route planning capability. Currently scientists work with IRG staff to use xGDS route planning tools to specify where to send the rover(s) and what scientific instruments to use at various waypoints and segments of the survey. This will output a "survey" in a standard file format (json) that is then uploaded to one of IRG's Kn-series rovers. The survey and its current state are broadcast over the telemetry stream. The VERVE 3D View as well as the Survey View display information about the survey, so users can understand it in a geographical context and also learn about detailed survey status and settings. The rover will then automatically traverse along the survey, making the safest choices for its navigation path based on Rover Software.

The Survey View clearly shows the current step of the survey that the rover is executing. Steps can be expanded to see detailed science instrument settings, and the name of the step will correlate to the data product gathered and shown via xGDS tools. If there is an instrument or any kind of failure during execution of the survey, this will be reflected in both the Survey View and the VERVE 3D View.

Eclipse also provides a view that encapsulates the native, platform default web browser. This allows us to embed web-based tools directly within VERVE, and integrate them by intercepting hyperlinks. For example if a 3D file is clicked on within the browser, it is loaded into the VERVE 3D View. The integrated web browser allows us to open xGDS-WT displays inside of VERVE, enabling the full suite of xGDS tools to be viewed in one integrated application window. Having a single integrated, customizable application for users to monitor robot health, data gathering, and survey state provides a complete situational awareness platform for our end-users (Figs. 16–24).

4.4. Field test use and results

The VERVE 3D View was used during the 2010 DRATS Field Test to assist robot operators located at JSC to remotely drive two SEV rovers at the Black Point Lava Flow in Arizona. The drivers' primary interface was the RAPID Workbench, a collaborative development effort involving Ames, JSC and JPL. Components developed at each center have been integrated into a unified application for monitoring and commanding robotic assets. The interface was used to drive each robot independently, as well as a coordinated convoy driving demonstration.

The remote convoy driving scenario involved two SEV rovers driving in tandem, splitting up to follow independent paths, and then rejoining to continue tandem

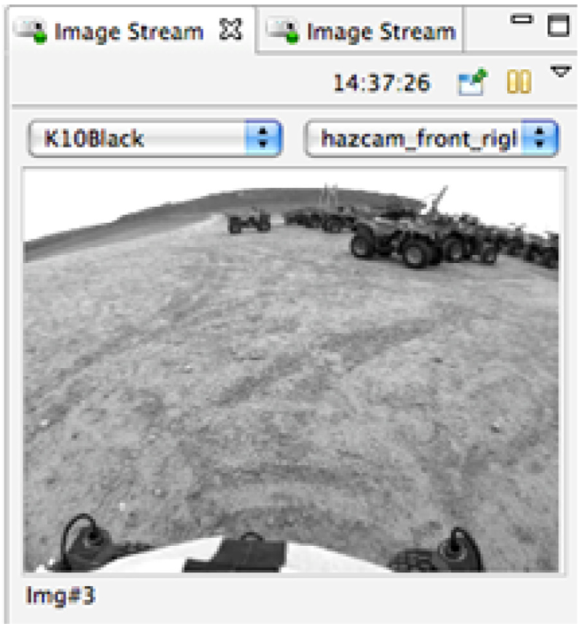
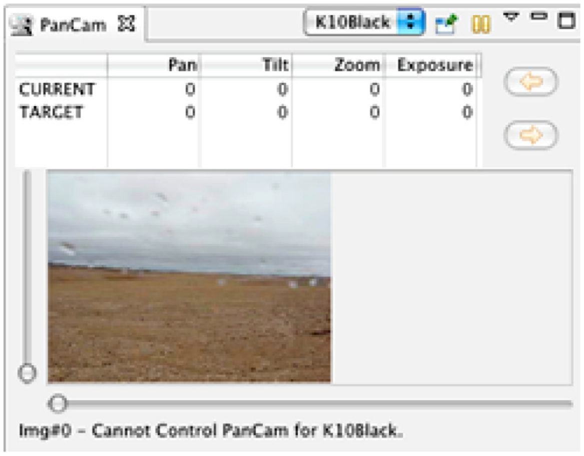


Fig. 16. Image Stream showing front right hazard camera.



	Pan	Tilt	Zoom	Exposure
CURRENT	0	0	0	0
TARGET	0	0	0	0

Img#0 – Cannot Control PanCam for K10Black.

Fig. 17. PanCam View to control the Pan Tilt Camera, i.e. sending commands to the robot.

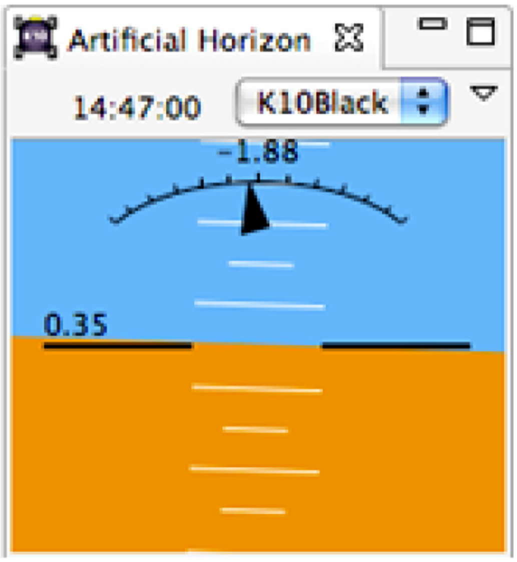
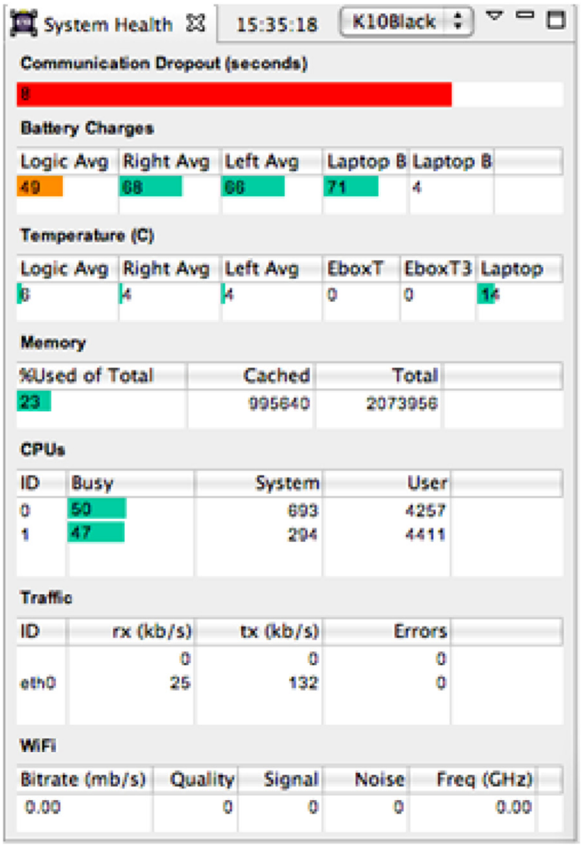


Fig. 18. Artificial horizon view quickly conveys pitch and roll for a rover.

driving. Each SEV was driven by an operator located at JSC, and each operator had a separate workstation running an instance of the RAPID Workbench. Each operator was able to monitor both vehicles, the planned actions of the other operator, and the predicted driving behavior of the vehicle they were responsible for.


System Health 15:35:18 K10Black

Communication Dropout (seconds)

8

Battery Charges

Logic Avg	Right Avg	Left Avg	Laptop B	Laptop B
49	68	68	71	4

Temperature (C)

Logic Avg	Right Avg	Left Avg	EboxT	EboxT3	Laptop
8	4	4	0	0	14

Memory

%Used of Total	Cached	Total
23	995640	2073956

CPUs

ID	Busy	System	User
0	50	693	4257
1	47	294	4411

Traffic

ID	rx (kb/s)	tx (kb/s)	Errors
	0	0	0
eth0	25	132	0

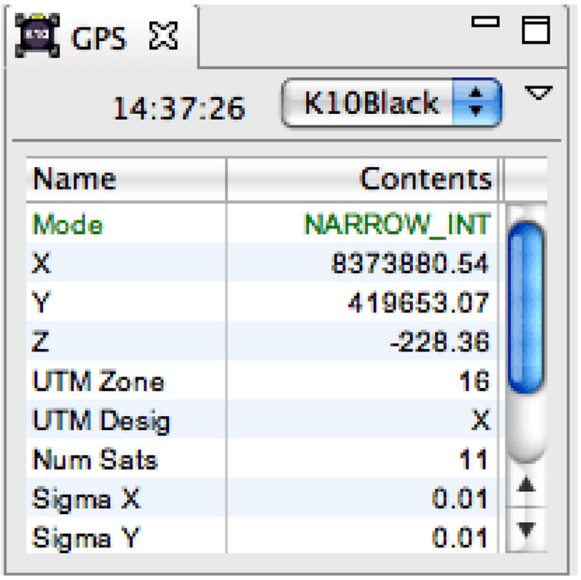
WiFi

Bitrate (mb/s)	Quality	Signal	Noise	Freq (GHz)
0.00	0	0	0	0.00

Fig. 19. The system health view gives summary information about important telemetry information. These values are color-coded for quick interpretation by robot operators. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

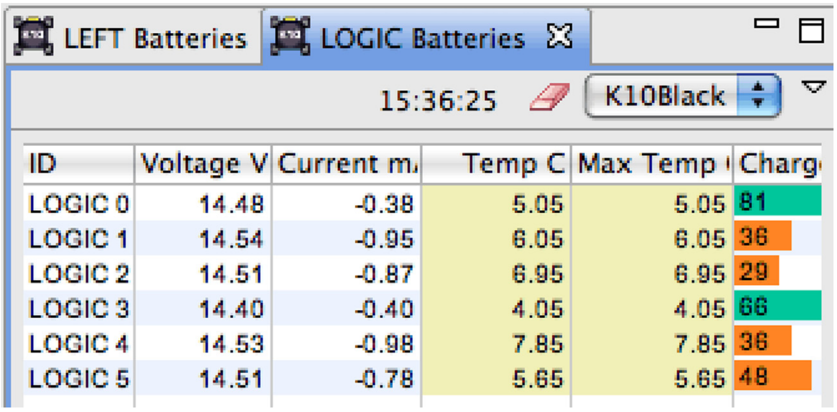
Drive commands were issued using the Predictive Interactive Graphical Interface (PIGI) developed at JSC. Using PIGI, an operator specifies a drive command, which is sent to a behavioral simulation of the robot. The operator reviews the expected outcome of the command, and commits the command to the vehicle if it is satisfactory [18]. VERVE provides the display component and communicates with other PIGI components via the same publish/subscribe (pub/sub) messaging system that is used for robot telemetry.

We chose to use pub/sub messaging between PIGI and VERVE rather than in-process method calls in order to enable a shared workspace between multiple Workbench instances. During the coordinated driving task, it was essential for an operator to be aware of the planned actions of other operator. An operator's planned waypoint is displayed in the VERVE 3D View as a "puck" that indicates position and orientation, and other operator's pucks are displayed in the same view in a different color


GPS 14:37:26 K10Black

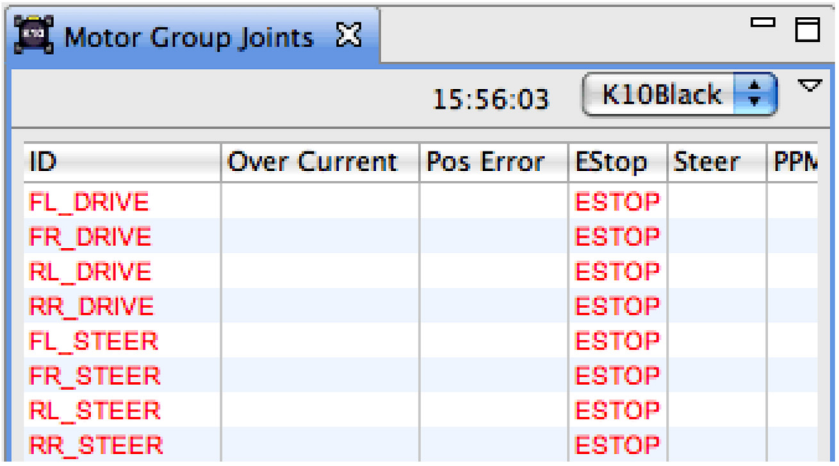
Name	Contents
Mode	NARROW_INT
X	8373880.54
Y	419653.07
Z	-228.36
UTM Zone	16
UTM Desig	X
Num Sats	11
Sigma X	0.01
Sigma Y	0.01

Fig. 21. Detailed GPS (global positioning system) view.


LEFT Batteries LOGIC Batteries 15:36:25 K10Black

ID	Voltage V	Current m.	Temp C	Max Temp	Charg
LOGIC 0	14.48	-0.38	5.05	5.05	81
LOGIC 1	14.54	-0.95	6.05	6.05	36
LOGIC 2	14.51	-0.87	6.95	6.95	29
LOGIC 3	14.40	-0.40	4.05	4.05	66
LOGIC 4	14.53	-0.98	7.85	7.85	36
LOGIC 5	14.51	-0.78	5.65	5.65	48

Fig. 20. Logic batteries telemetry view.



Motor Group Joints

15:56:03 K10Black

ID	Over Current	Pos Error	EStop	Steer	PP
FL_DRIVE			ESTOP		
FR_DRIVE			ESTOP		
RL_DRIVE			ESTOP		
RR_DRIVE			ESTOP		
FL_STEER			ESTOP		
FR_STEER			ESTOP		
RL_STEER			ESTOP		
RR_STEER			ESTOP		

Fig. 22. Detailed Motor Group Joints View, currently showing an emergency stop.

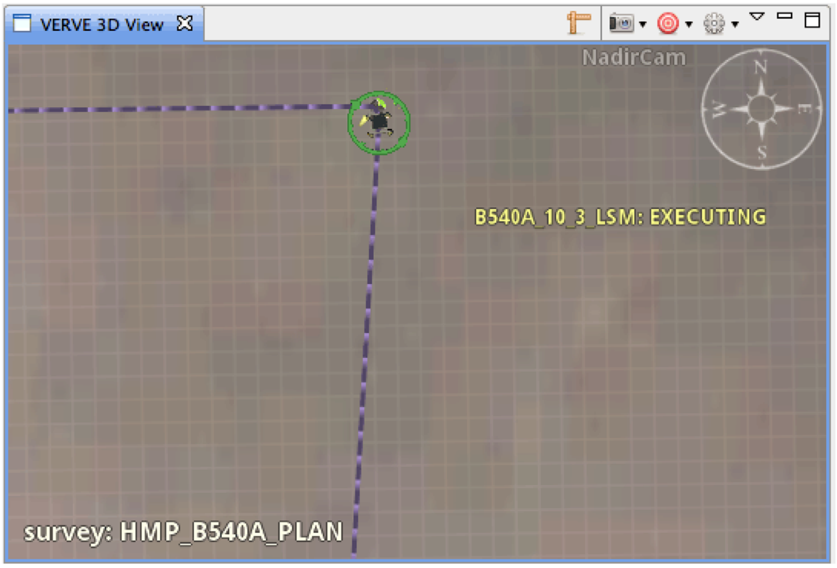
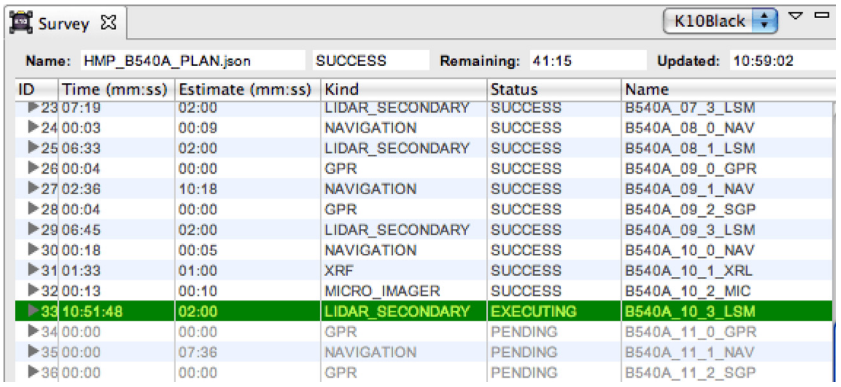


Fig. 23. VERVE 3D View showing survey information.



Survey

K10Black

Name: HMP_B540A_PLAN.json SUCCESS Remaining: 41:15 Updated: 10:59:02

ID	Time (mm:ss)	Estimate (mm:ss)	Kind	Status	Name
▶23	07:19	02:00	LIDAR_SECONDARY	SUCCESS	B540A_07_3_LSM
▶24	00:03	00:09	NAVIGATION	SUCCESS	B540A_08_0_NAV
▶25	06:33	02:00	LIDAR_SECONDARY	SUCCESS	B540A_08_1_LSM
▶26	00:04	00:00	GPR	SUCCESS	B540A_09_0_GPR
▶27	02:36	10:18	NAVIGATION	SUCCESS	B540A_09_1_NAV
▶28	00:04	00:00	GPR	SUCCESS	B540A_09_2_SGP
▶29	06:45	02:00	LIDAR_SECONDARY	SUCCESS	B540A_09_3_LSM
▶30	00:18	00:05	NAVIGATION	SUCCESS	B540A_10_0_NAV
▶31	01:33	01:00	XRF	SUCCESS	B540A_10_1_XRL
▶32	00:13	00:10	MICRO_IMAGER	SUCCESS	B540A_10_2_MIC
▶33	10:51:48	02:00	LIDAR_SECONDARY	EXECUTING	B540A_10_3_LSM
▶34	00:00	00:00	GPR	PENDING	B540A_11_0_GPR
▶35	00:00	07:36	NAVIGATION	PENDING	B540A_11_1_NAV
▶36	00:00	00:00	GPR	PENDING	B540A_11_2_SGP

Fig. 24. Survey View showing survey state.

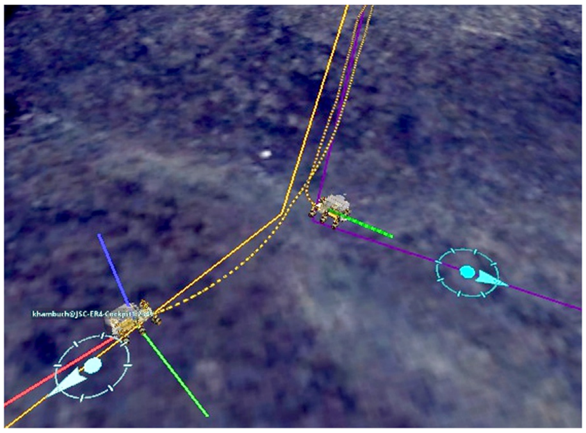


Fig. 25. The VERVE/PIGI display component used for DRAT.

and annotated with that operator's user name (Fig. 25). The remote operators were able to maintain consistent convoy spacing and coordination by monitoring the current location of both vehicles and the planned activities of the other operator. Both robots were successfully driven remotely for a total of 21.8 km, 0.65 km of which was coordinated convoy driving.

During the same year, VERVE was also used during the K10 Robotic Follow-Up test to provide both the engineering team and remote science backroom with current plan execution status and visualization of scientific data products. The K10 rover primarily drove in autonomous mode (following a plan created by the science team) so being able to view the current location and other sensor telemetry in real time was critical for the data collection and safety related aspects of the experiment. For example, if an instrument failed to start acquiring data, the lack of visualized data in the 3D VERVE view would alert the team of the failure. Additionally the health monitoring displays allowed the engineering team to monitor the battery levels, temperatures, and CPU usage, among other things, to quickly detect any anomalies.

5. Gigapan Voyage

5.1. Overview

Gigapan Voyage is a remotely operable high-resolution panorama capturing system designed for easy deployment on multiple robotic platforms and field tests (Figs. 26 and 27). It can display the captured panoramas via its own integrated interface or pass that data onto the xGDS system to be integrated with data collected from other platforms at a field experiment. Gigapan Voyage is based on a technology spun off from NASA Ames as a joint collaboration between NASA Ames, Carnegie Mellon University, and Google simply called "GigaPan" [19]. A "GigaPan" is an extremely large panorama that is usually captured via an automated system, and later stitched together. One of the key innovations of this technology is the ability to zoom into very small details in a panorama while still maintaining the much larger context. Gigapan Voyage encapsulates all the separate subcomponents of the commercial "GigaPan" system into a standalone package that can be controlled remotely using a simple web-interface [4]. We designed Gigapan Voyage to significantly minimize cross-center integration effort by creating a stand-alone system that can be used just as easily in a classroom or museum as it can on a rover. All the system requires from its "host" is power (24 V) and a wireless network connection.

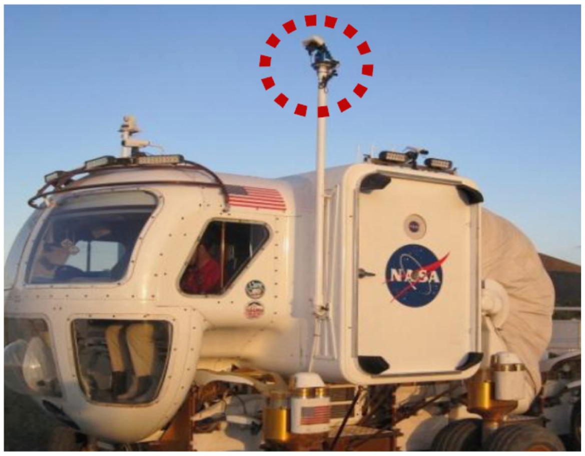


Fig. 26. Gigapan Voyage mounted on the Space Exploration Vehicle in 2010.

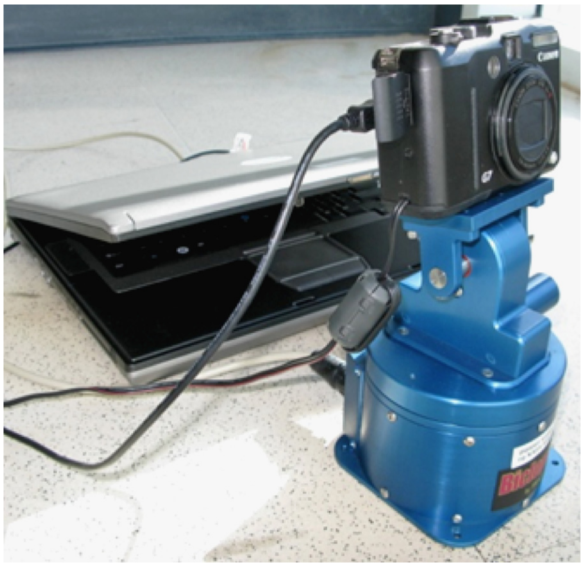


Fig. 27. Gigapan Voyage Hardware.

It is important to note that, although we do use the same image stitching software, we are currently not using any portion of the commercial GigaPan hardware due to operational constraints. Specifically, the Gigapan Voyage hardware has the following unique requirements that are currently not met by the commercial system:

1. Remote access to the system is required since the science team is not in close proximity to the hardware. This includes the ability to configure the camera and the pan/tilt head via a web interface.

2. Weatherized hardware is required to handle the harsh field conditions including significant vibrations during traverses, extreme temperatures, and light rain.

5.2. Use cases

Gigapan Voyage was initially designed specifically for use at the Desert RATS field tests by remote science teams, crew that might be living in a "local" habitat, and crew on board the rovers. Both the science team and the crews used the same web interface to acquire panoramas and had access to all the data products, although during DRATS 2011 the science backroom was subject to an artificial 50 s time delay as well as limited bandwidth for downloads to simulate Near Earth Asteroid operations that delayed their data by minutes to hours (Fig. 31).

After the panoramas were acquired and stitched the science backroom and the crew were able to use the Gigapan Voyage interface to take "snapshots" within the GigaPans and to annotate them.

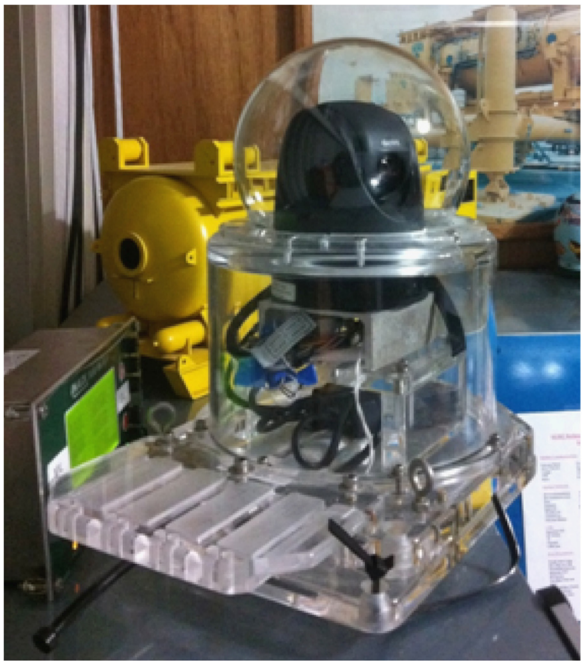


Fig. 28. Underwater Gigapan Voyage unit for NEEMO 15.

5.3. Implementation

5.3.1. Hardware

Virtually all of the hardware and software building blocks of the Gigapan Voyage system consist of technologies that are easily available and well supported. The Gigapan Voyage hardware currently consists of the following commercial, off-the-shelf components (see Table 3).

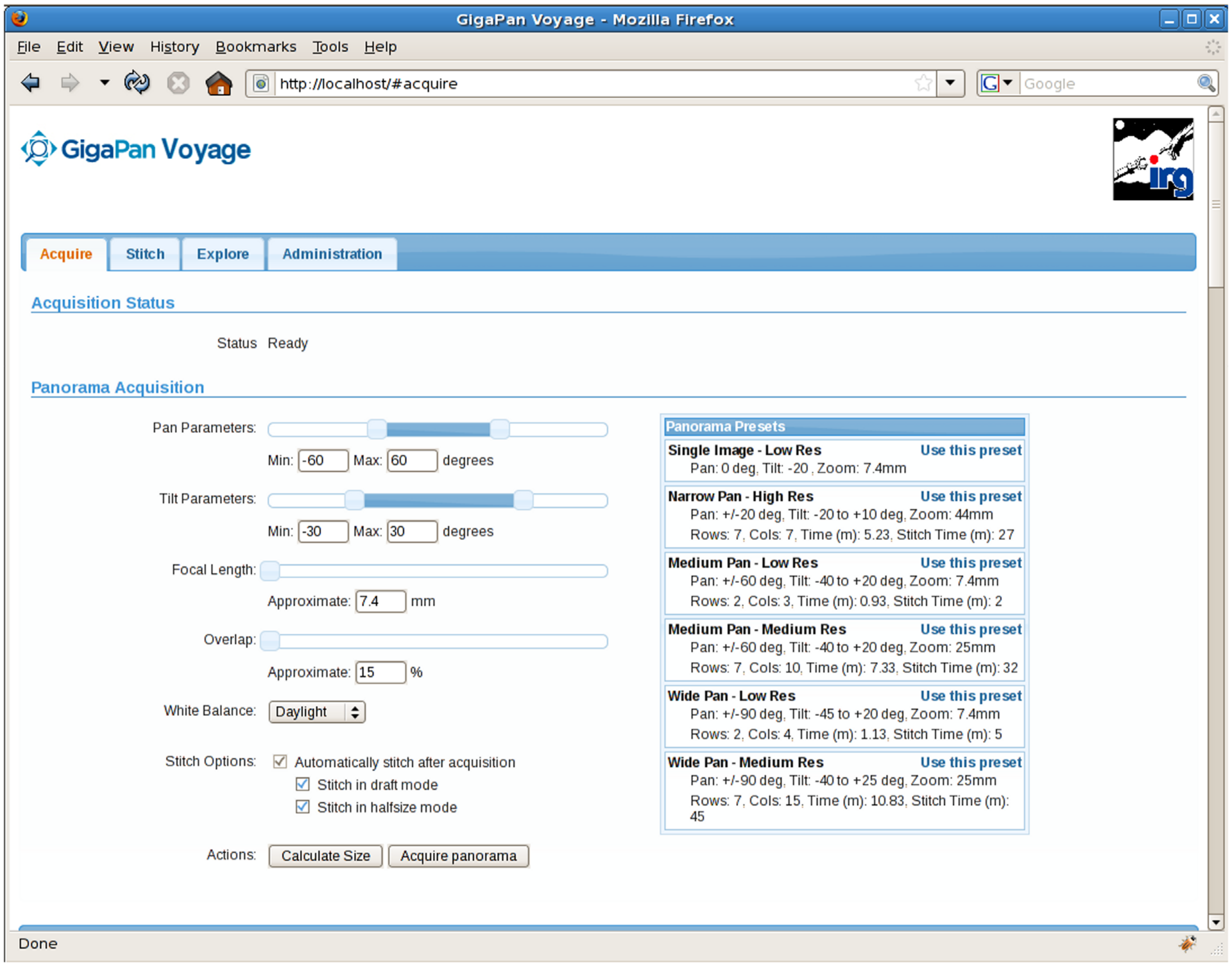
These off-the-shelf components were chosen because of their wide availability, our experience using them for other projects, and the software support freely available from IRG's internal software libraries, vendors, and/or open source communities. It is important to note that there are many other viable hardware manufacturers that could also be used. This year we are working to replace the existing hardware with waterproof hardware for the NEEMO field test. A commercial High Definition Pan Tilt Zoom (PTZ) camera and underwater dome have been identified as test candidates, and are currently being tested in Key Largo, Florida (Fig. 28). Although we had to write special software to control the PTZ, the majority of the Gigapan Voyage software will be reused. Additionally work is now underway to create a ruggedized time-lapse version of the system for use in the mountains of Chile for the Planetary Lake Lander Project [20].

5.3.2. Software

Gigapan Voyage interacts with users through a comprehensive web-based interface. The web interface consists of five possible tabs: "acquire", "crew", "stitch", "explore", and "admin" (see Fig. 29). Via the "acquire" tab, the user can set various capture parameters such as zoom, white balance, field of view, and amount of overlap between images. The "crew" tab provides the astronaut crew a simpler version of the full "acquire" tab. The astronauts only need to enter the desired azimuth of interest to command the system to automatically capture a low resolution 360° panorama and a high resolution panorama about their azimuth of interest. Once the desired parameters have been set, the panorama may then be captured. When complete, the panorama is added to the queue of panoramas waiting to be stitched. The user can then immediately start capturing a second panorama. Subsequent captured panoramas are also added to the end of the stitch queue. This queue of pending and executing stitch jobs can be viewed on the "stitch" tab. Stitch jobs can be monitored by the user for progress. The user can also view the raw images while waiting for the stitch to complete (see Fig. 30). Once a stitch job is complete, the resulting panorama is viewable

Table 3
Hardware components.

Hardware	Manufacturer	Notes
Pan Tilt Unit	TracLabs Biclops	Selected due to speed, payload rating, weatherization, and encoder feedback
Digital Camera	Canon G9	Theoretically any camera supported by the gPhoto open source library can be used
GPS unit	Global Sat BU-353	Small, inexpensive USB GPS unit
Laptop	Dell	Running RedHat Enterprise Linux 5


GigaPan Voyage - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://localhost/#acquire
Google
GigaPan Voyage
Acquire
Stitch
Explore
Administration

Acquisition Status

Status Ready

Panorama Acquisition

Pan Parameters:
Min: -60 Max: 60 degrees

Tilt Parameters:
Min: -30 Max: 30 degrees

Focal Length:
Approximate: 7.4 mm

Overlap:
Approximate: 15 %

White Balance: Daylight

Stitch Options: ✓ Automatically stitch after acquisition
✓ Stitch in draft mode
✓ Stitch in halfsize mode

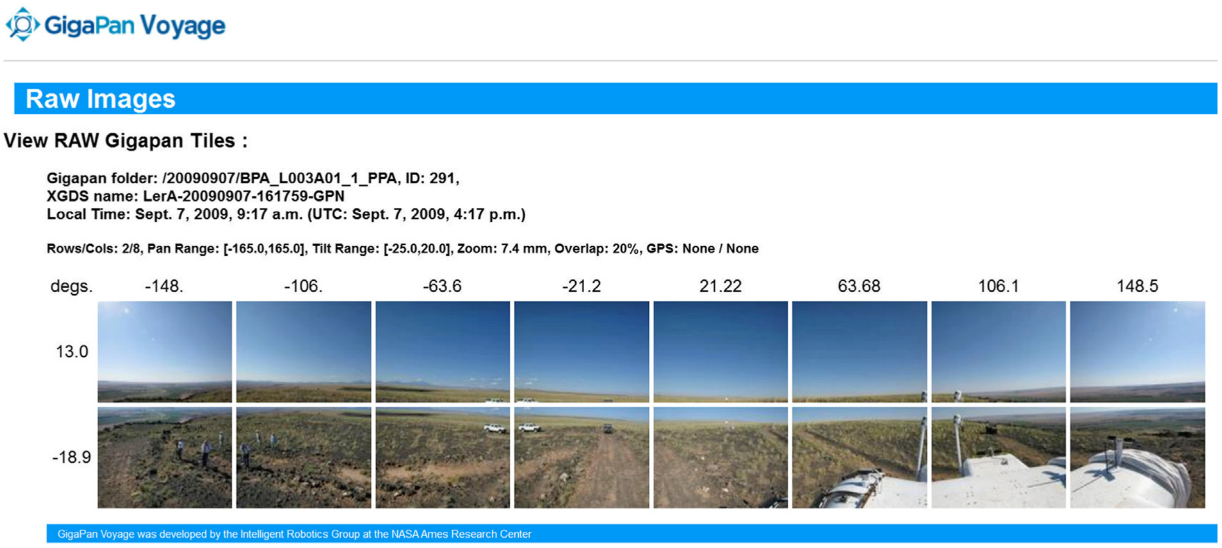
Actions: Calculate Size | Acquire panorama

Panorama Presets

Single Image - Low Res Pan: 0 deg, Tilt: -20 , Zoom: 7.4mm	Use this preset
Narrow Pan - High Res Pan: +/-20 deg, Tilt: -20 to +10 deg, Zoom: 44mm Rows: 7, Cols: 7, Time (m): 5.23, Stitch Time (m): 27	Use this preset
Medium Pan - Low Res Pan: +/-60 deg, Tilt: -40 to +20 deg, Zoom: 7.4mm Rows: 2, Cols: 3, Time (m): 0.93, Stitch Time (m): 2	Use this preset
Medium Pan - Medium Res Pan: +/-60 deg, Tilt: -40 to +20 deg, Zoom: 25mm Rows: 7, Cols: 10, Time (m): 7.33, Stitch Time (m): 32	Use this preset
Wide Pan - Low Res Pan: +/-90 deg, Tilt: -45 to +20 deg, Zoom: 7.4mm Rows: 2, Cols: 4, Time (m): 1.13, Stitch Time (m): 5	Use this preset
Wide Pan - Medium Res Pan: +/-90 deg, Tilt: -40 to +25 deg, Zoom: 25mm Rows: 7, Cols: 15, Time (m): 10.83, Stitch Time (m): 45	Use this preset

Done

Fig. 29. Gigapan Voyage web interface.



Raw Images

View RAW Gigapan Tiles :

Gigapan folder: /20090907/BPA_L003A01_1_PPA, ID: 291,
XGDS name: LerA-20090907-161759-GPN
Local Time: Sept. 7, 2009, 9:17 a.m. (UTC: Sept. 7, 2009, 4:17 p.m.)

Rows/Cols: 2/8, Pan Range: [-165.0,165.0], Tilt Range: [-25.0,20.0], Zoom: 7.4 mm, Overlap: 20%, GPS: None / None

GigaPan Voyage was developed by the Intelligent Robotics Group at the NASA Ames Research Center

Fig. 30. Raw tile view gives scientists access to the raw data as soon as the last image is acquired in a panorama.

via the "explore" tab. Finally, the "admin" tab is used to start and stop the software.

Coordination between the camera and pan/tilt unit is accomplished via open source software called Rover Software [21]. Rover Software allows us to take input from the web interface and pass it along to camera and pan tilt controllers. As its name suggests, Rover Software is primarily used to run rovers, but can conveniently also be used to control any instrument that is typically mounted on our robots. This software has been in development for the last decade by another project in our group. By utilizing software that is in active development by our group, we are both leveraging and contributing to work that will be reused by other projects.

The web-based software is the only component of Gigapan Voyage that we developed specifically for the purposes of the DRATS field experiments. All the software libraries used to build the web interface are freely available on the web and widely supported by various developer communities. Although merely open-sourcing software does not guarantee its quality, we feel that having a large, active user base around a piece of software is a fairly good indicator the software will be improved quickly and that it likely has intrinsic merit. By building on the work of a larger community we can focus our efforts on the nuances of the needs of the science team instead of the implementation details of the underlying software. As mentioned earlier, work has already begun to use slightly modified versions of our web interface for the NEEMO and Planetary Lake Lander projects.

The web interface is served using the Apache web server and built using a number of open source technologies including Django [22], mod_python [23], SQLite [24], and jQuery [25]. Furthermore, Asynchronous JavaScript and XML (AJAX) [26] are utilized to make the website more responsive and give semi-immediate feedback on errors that occur. The actual panorama stitching is performed using GigaPan Stitch [27]. A secondary benefit of reusing pieces of the software is that it allows the users to become very familiar with the tools and their capabilities, even allowing them to participate in the development of requirements for future tests.

5.4. Field test use and results

This section will focus on how the Gigapan Voyage system was utilized during the DRATS 2010 field experiment. Two Space Exploration Vehicles (SEV) [28] were used in various configurations and test conditions. Scientists gave the crew on each vehicle waypoints to drive to and areas to explore. Typically each time a rover would arrive at a designated Extra Vehicular Activity (EVA) location, a survey panorama would be initiated by the science team or by crew on-board the robot. A survey panorama is a wide area, low resolution panorama that is made up of approximately six to twenty images and stitches in about a minute. This low resolution panorama allowed the remote science team to quickly choose which sub-areas of the panorama were scientifically interesting. They then proceeded to take higher resolution panoramas of those regions. Since higher resolution panoramas can take significantly longer to acquire and stitch than simple survey panoramas, the raw images were made available to view immediately.

Over a period of 11 days, approximately 220 high resolution panoramas were captured using Gigapan Voyage. The size of each panorama varied greatly depending on the science goals for a particular waypoint, but averaged at about 250 MB and topped out at about 2 GB. The size of each panorama increased significantly starting at an average of about 150 MB per panorama on days one and two and increasing to over 500 MB per panorama by the end of the field experiment. All the data produced by Gigapan Voyage was imported into xGDS and presented to the scientists along with other telemetry collected from the robots throughout the day.

As the science team gained more experience and comfort with Gigapan Voyage, they used the system to gather panoramas with three to four times more detailed information even though taking larger panoramas came with the price of much longer acquisition times. For example, a standard "survey panorama" consisting of 2×10 image only took 2.5 min to acquire and less than 2 min to stitch, while a 47×11 image panorama took about 50 min to acquire and 3 h to stitch.

At the completion of the field experiment, science team members were given a survey regarding the scientific value of the system and its data products. Thirteen members of the field science team responded.

- 69% of the respondents rated the system a four or five. A rating of five indicates that the system was critical for their situational awareness. The remaining responders gave the system a rating of three—corresponding to the data giving them some amount of situational awareness.

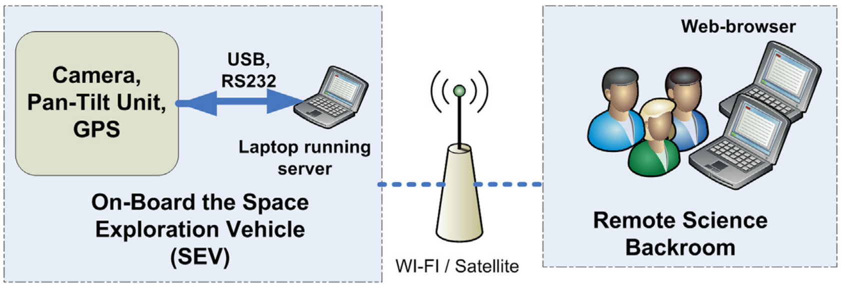


Fig. 31. Gigapan Voyage Communication Diagram. Inside SEV the Gigapan Voyage laptop communicates to the SEV computer via a wired Ethernet connection. Gigapan Voyage also gets all its power directly from the SEV.

- All respondents found the system intuitive to use.
- None of the respondents felt the panorama acquisition time caused delays to the astronaut mission.
- Most respondents felt that the stitching time could be improved, but was unavoidable. The ability to view raw tiles immediately helped mitigate the delay.
- 69% felt the final data product was of good or great quality. None of the survey responders felt the data was of poor quality.

Based on a survey taken immediately after the test, scientists and astronauts who participated in the 2010 DRATS field tests generally found the Gigapan Voyage system to be intuitive to use and valuable for situational awareness, documentation, and scientific analysis. Fig. 32 shows examples of the data products collected by the science teams. (The panoramas acquired during DRATS2011 have yet to be analyzed for statistical purposes, but the primary rover used during the 2011 tests captured over 170 high resolution panoramas over about a ten day period. All of the panoramas captured by Gigapan Voyage are now available for public viewing on www.gigapan.orgsearch word: drats2011.)

Due to the utilization of off-the-shelf technologies used in the development of Gigapan Voyage, the first version of the system launched in 2009 went from conception to deployment in less than six person months. Each subsequent version of the system that included usability improvements, a few bug fixes, and feature additions took about one to two person months per year plus the cost of hardware upgrades or repair. We expect

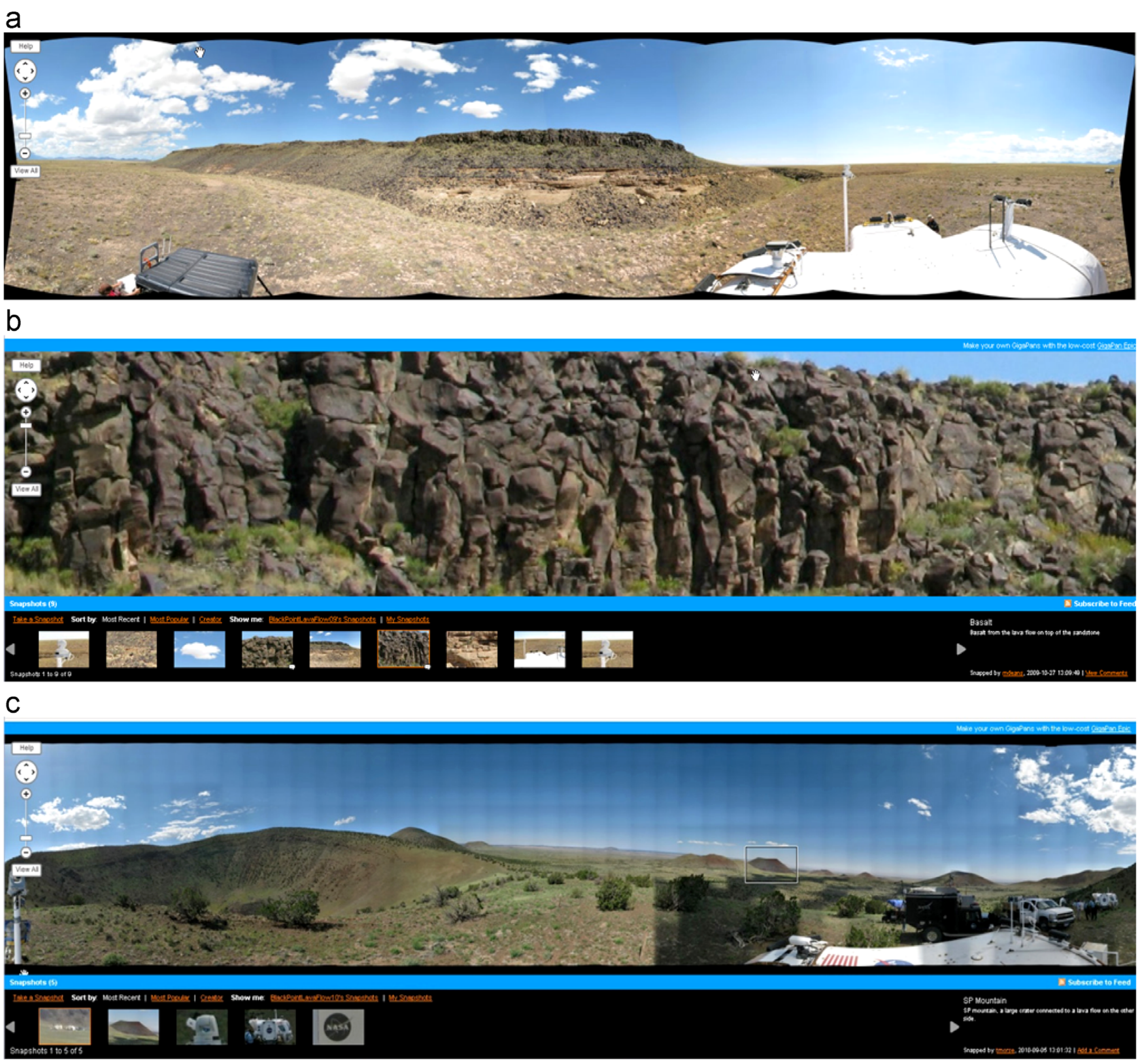

a
Help
View All
b
Make your own GigaPans with the low-cost GigaPan Epic
Help
View All
Snapshots (9)
Subscribe to Feed
Take a Snapshot Sort by: Most Recent | Most Popular | Creator Show me: BlackPointLavaFlow09's Snapshots | My Snapshots
Snapshots 1 to 9 of 9
Basalt
Basalt from the lava flow on top of the sandstone
Snapped by mdeans, 2009-10-27 13:09:49 | View Comments
c
Make your own GigaPans with the low-cost GigaPan Epic
Help
View All
Snapshots (5)
Subscribe to Feed
Take a Snapshot Sort by: Most Recent | Most Popular | Creator Show me: BlackPointLavaFlow10's Snapshots | My Snapshots
Snapshots 1 to 5 of 5
SP Mountain
SP mountain, a large crater connected to a lava flow on the other side.
Snapped by tmorse, 2010-09-05 13:01:32 | Add a Comment

Fig. 32. Gigapans taken during DRATS 2010. Image (a) zoomed out GigaPan of Basalt wall [29]. Image (b) same panorama as top, but zoomed into expose more detail. Image (c) is the Colton Crater at Black Point Lava Flow 2010 [30].

the development effort involved to adapt the Gigapan Voyage system to future projects, such as NEEMO15 and Planetary Lake Lander experiments, to be about the same.

6. Conclusions

Whenever possible, we deploy our xGDS tools as web-based applications. This is a contrast to the way Ground Data System tools have traditionally been designed and deployed at NASA, and it offers us the same advantages that web applications provide in other fields. In particular, we frequently work with geographically distributed teams who are co-located only during field operations (usually one to two weeks per year). All pre-test planning and post-test data analysis take place at the team members. home institutions. Using familiar open source and web tools to reduce the need for user training and having a central location for the data repository and application updates is a substantial advantage. This design approach was also successful for our Gigapan Voyage deployment at DRATS where the science team controlling the camera was located remotely from the rover, but was able to control it easily via the web interface.

However, there are situations (notably 3D visualization and real-time control and monitoring, especially for engineering and systems diagnostics) where web-based tools are not adequate to meet our operations requirements. Even in these cases, we endeavor to build on open source frameworks so that our work can more easily be shared with others. This is why VERVE was built as an Eclipse RCP application using as many open source and freely available tools as possible. VERVE provided our science and rover operations teams with real-time situational awareness and systems status during autonomous operations.

We deploy our xGDS tools and instruments like Gigapan Voyage at multiple field deployments to study different use cases; this helps us create a more complete and flexible software toolkit. No single analog is exactly like a "real" planetary mission either scientifically or technically, but together they provide a much more comprehensive testing scenario. DRATS is primarily an engineering focused exercise, which allowed us to test the coordination and tracking of multiple assets (human and robotic) and work with a large science and engineering team, similar to a real planetary mission. PLRP is the most science driven of our analogs, and tests our ability to process and present data quickly and efficiently to scientists who need it for rapid turnaround planning during the field deployment and for more detailed analysis after the test is complete. Finally, our K10 deployment tested an autonomous robot in a novel environment where both the engineering and science teams gained situational awareness from the rover's telemetry.

Analog exploration experiments have become increasingly important to NASA's future manned and robotic missions. In the past, each of these experiments was conducted relatively independently of each other and was designed to focus on answering certain domain specific questions. For example earlier DRATS tests have focused on studying how to utilize a robotic vehicle and habitats most effectively with two to four manned crews, while NEEMO has focused on testing tasks underwater to simulate some of the physics based constraints experienced in a real space environment. In 2011, there has been a push to start integrating technologies across NASA-led field experiments to create a suite of tools that can be tested in all the different field experiments.

IRG has been able to support all these missions and their objectives over the years by designing our systems around a flexible architecture and by leveraging existing technologies. This allows us to be able to reuse the majority of our xGDS-WT, VERVE, and Gigapan Voyage tools from year to year and test to test.

Acknowledgments

We would like to thank the following field teams for the opportunity to help support NASA in their analog tests:

- JSC Space Exploration Vehicle team.
- NASA Scientists, and crew involved in the DRATS experiment.
- The K10 team.
- The PLRP team.
- The NEEMO15 team.
- This work was supported in part by NASA Ames Research Center and Johnson Space Center under the Human Robotic Systems Project and sponsored by the NASA Exploration Technology Development Program.

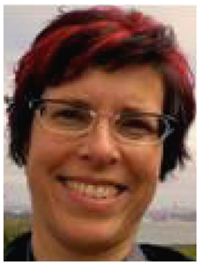
References

[1] Intelligent robotics group @ NASA Ames. URL 〈http://ti.arc.nasa.gov/tech/asr/intelligent-robotics/〉, 2011.
[2] Desert rats overview. URL 〈http://www.nasa.gov/exploration/analogs/desertrats/〉, 2011.
[3] Pavilion lake research project. URL 〈http://www.pavilionlake.com〉, 2011.
[4] S. Lee, T. Morse, E. Park, Gigapan voyage for robotic reconnaissance, in: Proceedings of the Fine International Conference on Gigapixel Imaging for Science, CMU University, Pittsburgh, PA, 2010.
[5] NASA extreme environment mission operations. URL 〈http://www.nasa.gov/mission_pages/NEEMO/index.html〉, 2011.
[6] Robotic recon. URL 〈http://lunarscience.nasa.gov/roboticrecon〉, 2011.
[7] Open geographical consortium, kml specifications. URL 〈http://www.opengeospatial.org/standards/kml/〉, 2011.
[8] J. Feller, B. Fitzgerald, Understanding Open Source Software Development, Addison-Wesley, London, 2002.
[9] Ensemble planning tool used by phoenix science team. URL 〈http://ti.arc.nasa.gov/news/ensemble-planning-tool-used-by-phoenix-science-team/〉, 2011.
[10] K10 rover. URL 〈http://wn.com/NASA_K_10_robots〉, 2011.
[11] Eclipse rcp. URL 〈http://www.eclipse.org/home/categories/rcp.php〉, 2011.
[12] Ardor 3d homepage. URL 〈http://ardor3d.com/〉, 2011.
[13] R.J. Torres, Rapid: collaboration results from three NASA centers in commanding/monitoring lunar assets, in: Aerospace Conference, IEEE, 2009, pp. 1–11, 7–14.
[14] Athlete footfall planning. URL 〈http://ti.arc.nasa.gov/tech/asr/intelligent-robotics/footfall/〉, 2011.
[15] H. Hoppes, F. Losasso, Geometry clipmaps: terrain rendering using nested regular grids, ACM Transactions on Graphics (Proceedings of SIGGRAPH 2004), vol. 23(3), ACM, 2004, p. 769776.
[16] C.C. Tanner, C.J. Migdal, M.T. Jones, The clipmap: a virtual mipmap, in: Proceedings of the 25th Annual Conference on Computer

Graphics and Interactive Techniques (SIGGRAPH '98), ACM Press, 1998, pp. 151–158.
[17] Gdal homepage. URL 〈http://www.gdal.org/〉, 2011.
[18] R. Burridge, K. Hambuchen, Using prediction to enhance remote robot supervision across time delay, in: IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE/RSJ International Conference on Intelligent Robots and Systems, St. Louis, MO, 2009, pp. 5628–5634.
[19] Gigapan. URL 〈http://gigapan.org〉, 2011.
[20] Planetary lake lander kicks off with first team workshop. URL 〈http://astrobiology.nasa.gov/articles/planetary-lake-lander-kicks-off-with-first-team-workshop/〉, 2011.
[21] L. Fluckiger, V. To, H. Utz, Service-oriented robotic architecture supporting a lunar analog test, in: International Symposium on Artificial Intelligence, Robotics, and Automation in Space (iSAIRAS), 2008.
[22] Django. URL 〈https://www.djangoproject.com〉, 2011.
[23] mod_python. URL 〈http://www.modpython.org〉, 2011.
[24] Sqlite. URL 〈http://www.sqlite.org〉, 2011.
[25] Jquery. URL 〈http://jquery.com〉, 2011.
[26] Ajax. URL 〈http://www.w3schools.com/ajax/default.asp〉, 2011.
[27] Gigapan stitcher. URL 〈http://gigapansystems.com/gigapan-products/gigapan-software/gigapan-stitcher-software-information.html〉, 2011.
[28] NASA's space exploration vehicle. URL 〈http://www.nasa.gov/exploration/home/SEV.html〉, 2011.
[29] Gigapan of black point lava flow. URL 〈http://www.gigapan.org/gigapans/32832/〉, 2011.
[30] Gigapan of black point lava flow: colten crater. URL 〈http://www.gigapan.org/gigapans/58434/〉, 2011.

Susan Young Lee is the Lead Hardware Engineer for the Intelligent Robotics Group (IRG) at NASA Ames and has over 9 years of experience integrating, operating, and maintaining robotic test-beds. Susan joined the IRG in 2002 shortly after receiving her Bachelors and Masters in Mechanical Engineering/Mechatronics from Stanford University. Her responsibilities include project management, instrument integration, mechanical design, and embedded system design. Susan has also been a field team member in many field tests including the "Haughton Crater Lunar Site Survey" and the "Robotic Recon." Projects on Devon Island (2007 and 2010), and the DRATS robotic experiments in Arizona (2008–2011).

David Lees is a senior researcher with the Intelligent Robotics Group at NASA Ames. David's research interests include human-robot interaction, data visualization, user interfaces and the automated analysis and organization of large volumes of geo-spatial data. In addition to the work described here, he has also helped to adapt earlier versions of IRG's GDS tools into the flight data systems of the MER and Phoenix missions to support 3D visualization and data management for science analysis. David holds MS and Ph.D. degrees in Mechanical Engineering from Stanford University.



Tamar Cohen works with NASA Ames' Intelligent Robotics group to bridge the gap between robots and humans with a background in software engineering and fine art. She designs and implements software to provide situational awareness for remote robotic operation and for ground data systems. Tamar has worked in industry for 20 years, including developing interactive 3D computer games, internet collaboration software, software development kits for game development, and Eclipse plug-ins and applications. Tamar holds a BS in Computer Science from Cornell University, and an MA in Art, Visual Information Technology, from George Mason University.

Mark B. Allan is a Senior Software Engineer with the Intelligent Robotics Group at NASA Ames Research Center. Mark specializes in data visualization and has worked in the areas of ground control systems for remote exploration, novel human/computer interfaces, massively parallel data flow architectures, and flight simulators. Current topics of interest include the use of virtual worlds to effectively explore remote worlds, application of technology to enhance individual and team effectiveness, and architectures that enable efficient human-robotic coordination. Mark holds a Bachelor in Biological Sciences from UC Santa Barbara and a Masters in Information Systems from Santa Clara University.

Matthew Deans is the Deputy Group Lead of the Intelligent Robotics Group (IRG), in the Intelligent Systems Division (Code TI), at NASA Ames Research Center. Matthew's research focuses on applications of machine vision for robot navigation and control as well as automating image processing for remote science. He has significant experience in Field Robotics, participating in field experiments in Antarctica, the Arctic, Atacama Desert, Canadian Rockies, and several sites in the continental US. He was the field team lead for the K10 test at Haughton Crater, Nunavut Canada, in July 2007 and again in July 2010.

Theodore Morse graduated from the University of Evansville in 2007 with a Bachelor in Computer Science and Computer Engineering. Throughout his three years with IRG he has worked in a variety of different areas including stitching of gigapixel imagery, web-based application development, deployment and support for acquiring, stitching and viewing gigapixel imagery (Gigapan Voyage), mobile application development to aid disaster response; as well as back-end development and support for Ground Data Systems (xGDS).

Trey Smith studies techniques for sharing geospatial data with mobile devices, supporting field operations for applications ranging from disaster response to planetary analog robotics to field science in extreme environments. He received his Ph.D. from the Carnegie Mellon University Robotics Institute.