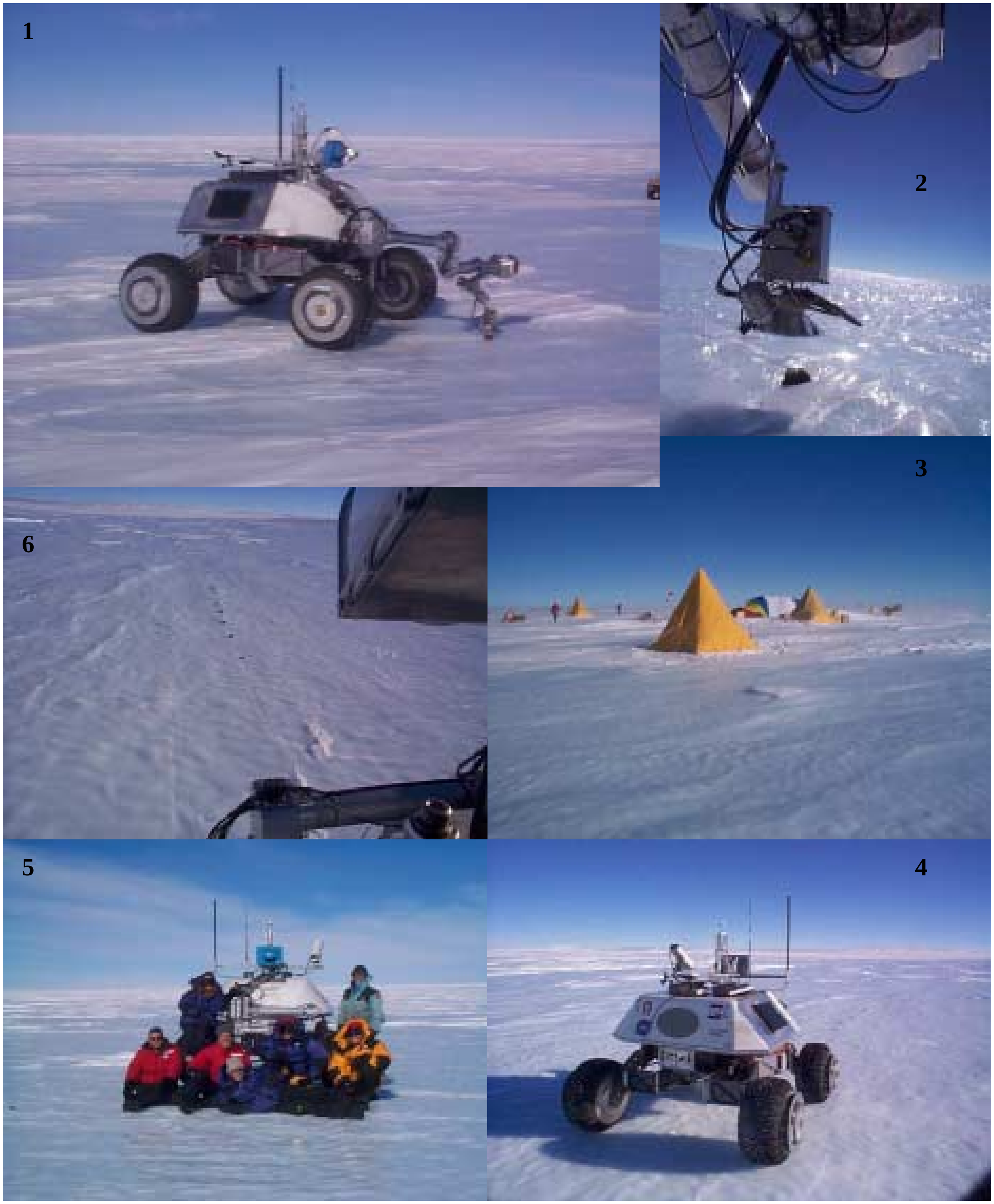
Experimenter's Notebook: Robotic Search for Antarctic Meteorites 2000 Expedition

CMU-RI-TR-00-13

Michael D. Wagner
Email: mwagner@ri.cmu.edu

Field Robotics Center, The Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213
Phone: (412) 268-8158; Fax: (412) 268-5895

June 2000



Clockwise from upper left: Nomad’s manipulator deployed to a meteorite; a close view of Nomad’s wrist examining a meteorite; the Elephant Moraine January 2000 base camp; Nomad in autonomous search mode; the expedition team; the view from Nomad’s high-resolution camera

Introduction

Purpose

This document serves as an archive of expedition notes taken by Michael Wagner in January 2000 during the Robotic Search for Antarctic Meteorites expedition to Elephant Moraine, Antarctica. The culmination of a three year effort developing Nomad as a robotic explorer occurred when Nomad made the first meteorite discoveries by a robot. Through the use of autonomous science target recognition, Bayesian classification techniques and mission planning, Nomad classified meteorites while performing patterned searches. Three meteorites were autonomously found and classified and, in total, five meteorites were classified correctly throughout the expedition. A total of 42 samples were analyzed after 2500 m^2 of blue ice and snow search. Crucial winterized electromechanical components that comprise Nomad such as its steering system and manipulator arm proved very capable in this extreme environment. In addition to Nomad, the expedition included six researchers from CMU (Stewart Moorehead, Liam Pedersen, Ben Shamah, Kim Shillcutt, Mark Sibenac and Michael Wagner) and John Schutt from the ANSMET program.

Details of this accomplishment are given here to provide a complete description of the expedition and how Nomad's science autonomy system (SAS) functioned. The document is not intended to summarize the activities performed or draw conclusions. As such, it should be seen as a companion to future publications.

Expedition Timeline

The notebook starts on January 9, 2000. The field team was deployed from McMurdo to the Elephant Moraine camp on January 10. Camp and robot setup occurred until January 14. To evaluate their performance in the extreme Antarctic environment, Nomad's components were tested until January 20, when the first of ten demonstration took place. The notebook continues until January 30, the last day of Nomad's operation. The team returned to McMurdo a few days afterward.

References

There are references in this document to other sources of information: spectra taken, spreadsheets of results and so on. These references, specified with a number in square brackets, often include a very large amount of data. For the sake of brevity and to allow these results to be augmented with new facts in the future, these documents are not included here. Instead, Appendix A includes URLs the reader may visit to access this additional and useful information.

Background Information

Computing Structure

The Nomad robot itself carries three computers. These include:

- `meteorite`: a Red Hat Linux 5.2 Pentium Pro computer running the science autonomy software and interfacing with the science sensors.
- `nav1`: another Red Hat Linux 5.2 Pentium computer running the autonomous navigation and obstacle avoidance software and interfacing with the laser range finder.

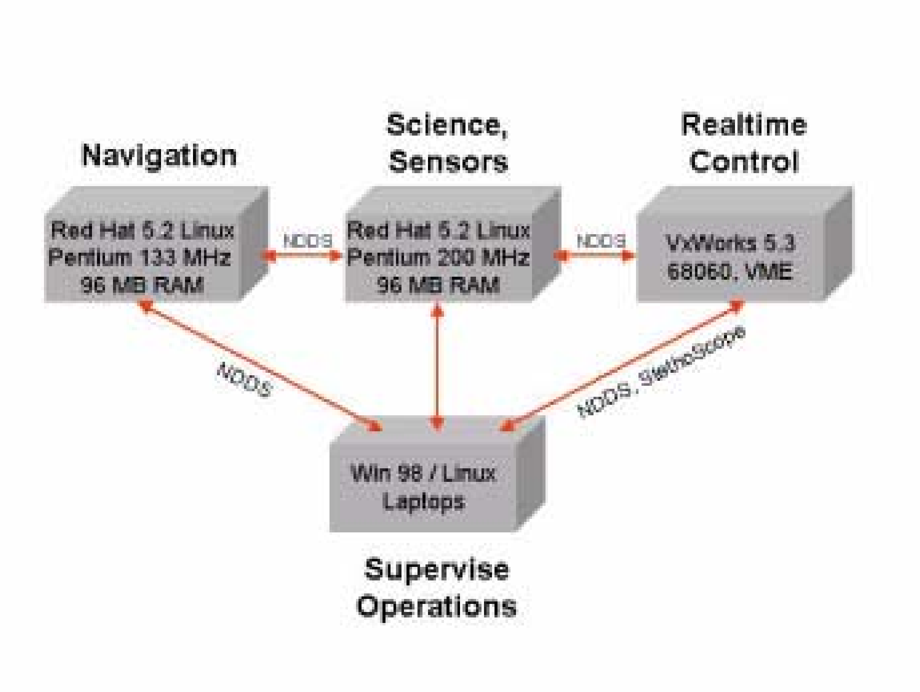


Figure 1. Nomad's computing structure

- `vw2`: the realtime control computer, running VxWorks 5.3 on a Motorola 68060 CPU in a VME cage.

Several laptops were also used to supervise Nomad's autonomous operations during the expedition. These are:

- `needsfixed`: a Dell Inspiron 7000 laptop running Red Hat Linux version 5.2. This laptop included software to connect to and compile software for Nomad's realtime computer.
- `sweeper`: a Dell Inspiron 3200 laptop also running Red Hat Linux version 5.2. This laptop included software to display video from Nomad's cameras for verification of rock finds and robot location.
- `author`: an IBM ThinkPad running Windows 98. This laptop ran a prototype Java application that graphically displayed Nomad's telemetry.

When mentioned in this document, host names are listed in `courier`. Please see Figure 1 for a high-level description of how these computers interface with each other.

Science Autonomy System

Most of the information in this document pertains to the SAS onboard Nomad. The SAS is a control architecture for exploration and *in situ* science that integrates sensing, navigation, classification and mission planning. Its development is an ongoing effort to enable autonomous robotic scientists. Specifically, it enables Nomad to search Antarctic ice fields and discriminate potential meteorites from terrestrial rocks. Several software modules are required to achieve this: sensors must be deployed, samples must be classified and search patterns must be executed. The names of these modules are also shown in `courier` in this document. These programs are described in Appendix B.

The science autonomy system is configured by the values of several dozen variables. When referenced in the text, these variables are shown in *italics*. The values of these variables are set by the `SAS_Config` program. Changing a variable's value can often alter a science autonomy module's behavior at runtime. See Appendix C for a description of each variable.

Aspects of this expedition such as autonomous navigation, real-time computing and other experiments may not be suitably detailed here.

Other Supplementary Information

Many rock samples that Nomad examined are listed in this document as "Rock A" or "Liam's rock #1". For a description of each of these rocks, see Appendix D.

Experimenter's Notebook - January 9, 2000

The team met today outside our temporary offices in McMurdo Station to list the most important action items to be finished upon arrival to Elephant Moraine camp.

- **Set up camp.** Obviously, setting up tents and HF radio contact with McMurdo Station is of primary concern.
    - Set up five scott tents. These include four living tents and a kitchen tent.
    - Set up endurance tent. This is the work tent where operations will take place. It is large enough to comfortably seat five or six people at a time.
    - Set up expedition (bathroom) tent.
    - Set up and turn on generators to provide camp with power.
    - Verify HF radio functionality.
    - Test Iridium phone.
    - Set up and test INMARSAT satellite phone equipment.
- **Set up robot.** Once the camp is complete, Nomad can be assembled, both physically and from a software standpoint.
    - Remove the robot from palette that was used during airlift from McMurdo Station.
    - Turn on Nomad’s on-board generator.
    - Drive the robot off the palette using its joystick and test steering.
    - Attach Nomad’s sensor mast and sensors (high-res camera, pan / tilt unit, laser rangefinder, panoramic camera).
    - Install ARLAN, radio modem and DGPS antennas.
    - Attach Nomad’s manipulator arm.
    - Check Java and Tcl/Tk user interfaces.
- **Calibrate robot.** Before autonomous operation is possible, the robot’s subsystems must to be calibrated. These subsystems include Nomad’s high-resolution camera, laser rangefinder, magnetic compass and image segmentation software.
    - Calibrate high-res camera with its calibration sheet. This white sheet with red dots is used to calibrate camera parameters such as pan, tilt, roll, focal length, aspect ratios, CCD width and translation from the DGPS sensor to the camera. These parameters are used to make estimates of the location of new science targets found.
    - Calibrate laser rangefinder. This involves accurately measuring the angle at which the laser is mounted on the sensor mast so accurate distances to potential obstacles can be calculated.
    - Check compass magnetic north offsets. These are very different in Antarctica than in Pittsburgh, so they must be verified and entered into the system, or else the compass will fail.
    - Tune high-res and arm camera segmentor. The high-res and arm cameras both use one image segmentation algorithm to pick out rocks. This algorithm thresholds

pixels based on color ratios and calculates centroids of connected "rock pixel" regions. There are several parameters that needed to be hand-tuned (see Appendix C). To accomplish this, many images from each camera must be taken and segmentation results examined.

- Train classifier segmentor. Nomad's target classifier has yet another segmentation routine used on zoomed in images taken of rocks. This algorithm requires than an initial target centroid be given, which comes from the high-res segementation algorithm. The classifier segmentor uses statistics to pick out a rock from the background. It is trained with 10 to 30 zoomed in images of rocks.

- **Test subsystems.** Once calibrated, specific functions of the robot can be tested.
    - Test video from robot's cameras (using `pxc_ui` interface). Make sure the bandwidth of the wireless ethernet connection is high enough to support this feature.
    - Test the touch probe to see if it trips properly, and only when it should.
    - Verify automatic classification from spectral data. Make sure that a spectrum taken by the SAS will automatically be properly processed by the target classifier.
    - Observe the auto-calibration of spectrometer. This involves placing the manipulator arm over a white reference target, taking a spectrum and checking that the results match white reference spectra taken in the lab.
    - Record wind direction sensor readings to test their validity. This sensor's reliability has been brought into question recently.
    - Perform search patterns with laser-based obstacle avoidance. One unanswered question is: how will search patterns be executed when Nomad's obstacle avoidance module is running? The laser rangefinder data is a single stripe three or four meters in front of the robot. Therefore at times the robot has no information about areas closer than that single stripe. However, tight (3 m radius) turns required by some search patterns may be force the robot into these closer regions. How will the system handle this? Will obstacle avoidance still be effective in these situations?
    - Record images from the panoramic camera and see if the mirror and lens are aligned properly. Additional problems could include fogging of the glass enclosure.
- **Experiments.** These integrated tests validate multiple subsystems at once.
    - Rock density experiment. Rocks will be placed in a line on the ice at locations surveyed with John Schutt's DGPS. The robot will drive along this line and acquire as many of these rock targets as possible. This will test the image and target acquisition systems and provide data about the limits of rock distribution density acceptable to the software.
    - Arm visual servoing and spectrometer placement will be tested to provide data on quality of autonomously taken sample spectra and repeatability of arm visual servoing.

Other general information:

- Goal: hold initial demonstrations during week 2
- Download data from the robot at the end of each day
- Record latitude and longitude of DGPS base station as soon as it is set up
- Operator must always be cognizant of robot's status (still or moving, etc.)
- Hold at least 2 email links / day: one in the morning to receive email, one at night to respond
- General work strategy: morning meeting, perform demonstration, meet to analyze results, change code, debug changes

Experimenter's Notebook - January 12, 2000

Today was a bad weather day. Tried to set up robot components despite a strong wind.

8:00pm: Mounted sensor mast with laser. Briefly started generator. No problems.

Experimenter's Notebook - January 13, 2000

4:00pm: Aligned laser, mounted PTU and high-res camera, ARLAN antenna and wireless modem antenna. Removed front shell without difficulty even with wind speeds around 30 mph. Hooked up all cables to units on the mast, including back shell of high-res housing. Started generator and main ebox power. Drove Nomad off the palette with the joystick to behind a scott tent for minimal wind protection. Realtime computer works well, ARLANs connect fine to robot inside camp. Have not yet turned power on to the `meteorite` or `nav1` computers, or the PTU.

Note: Altered high-res software to save images during all steps for debugging.

6:00pm: Turned on all systems on Nomad (minus arm, which is not connected). PTU started up and went through its calibration procedure. It did not require human assistance to move wires, etc. Inside the tent users could telnet to `meteorite` and `nav1`. There were problems mounting `meteorite's /usr` volume via NFS - it mounted but the access rights on `sweeper:/mnt/meteorite/mwagner` are not set correctly. Nothing can be read in the directory. However, files can be changed in subdirectories like `mwagner/nomad`. Files can be edited on `sweeper` but must be compiled on `meteorite.` NDDS started up fine - no problems with licenses or Y2K. Using `pxc_ui` on `meteorite`, images were displayed from the high-res camera. The `videoServer` / `pxc_ui` combination did not work, however. Using `grabImages` another image was received and the PTU was commanded to a pan of 0 degrees and a tilt of -10 degrees. When this was finished, it returned to its home position properly. Then `transportStow` was used, and it worked. Then `hiResSensor` was started and it servoed the PTU to the target acquisition position. Then the following SAS programs were started up on `meteorite`:

```
dbServer -v
dbClient
hiResSensor hiResSensor
sensManMain (after the arm code was #defined out in smManipulatorDriver and
recompiled)
targetAcqMain -test
```

On `sweeper`, `sensManClient` was started. On `meteorite` a target with a DGPS position of (0, -5, -1.84) was inserted into the database. With `sensManClient` the high-res camera was deployed to this target. It of course detected many small targets, as it's looking at a scott tent. It auto-focused and zoomed without any noticeable problems. The saved images looked reasonable. Then on `sweeper` we ran `targetAcqMsgs_client` to start `targetAcqMain` up. It detected 20 targets in one image and inserted them all into the database normally.

The database automatically saved itself, providing an archive of today's activities. Code was updated on `meteorite` for the software changes made since the robot left Pittsburgh, but first CVS needs to be pointed at the repository on `needsfixed` so we can tag the current version.

9:00pm: Attempted to point CVS'd files on `meteorite` to the repository on `needsfixed`. It seemed to work, but when an attempt was made to tag all files as `Jan13_2000`, it seemed as if some file somewhere is still looking at an AFS directory.

Experimenter's Notebook - January 14, 2000

Problem: fiber optic spectrometer cable shielding cracked in the cold - it was reinforced with tape and strips of plastic.

Note: Found compass accuracy to be better than 2 degrees.

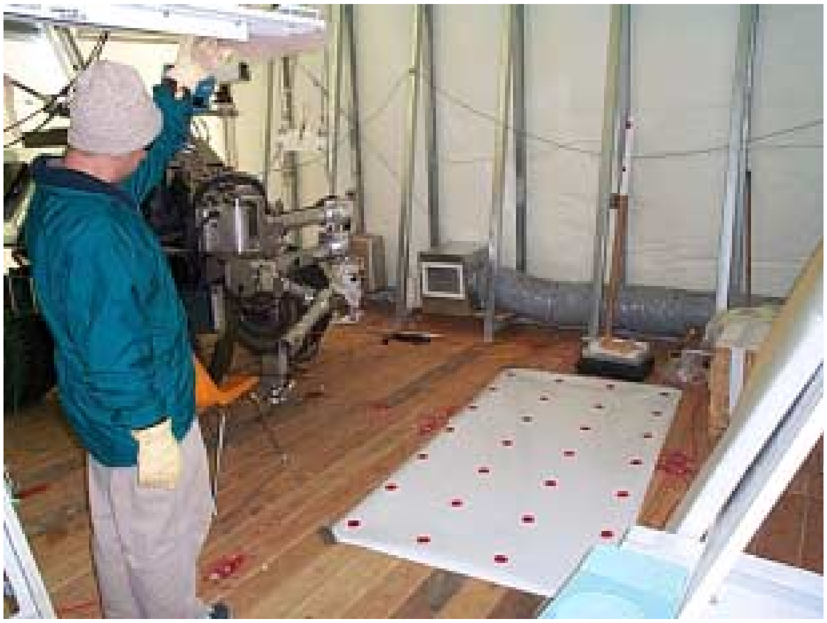


Figure 2. Nomad presented with calibration sheet (in McMurdo Station, Antarctica)

High-Resolution Camera Calibration

The calibration sheet was aligned in front of the robot (see Figure 2 for an example of this setup). The following convention will be used to number the dots on the sheet (note there's no column numbered 1 - 4 because of a mistake):

5	9	13	17	21	25	29	33
6	10	14	18	22	26	30	34
7	11	15	19	23	27	31	35
8	12	16	20	24	28	32	36

The dots are 30.5 cm apart in x and y directions. Dot 5 is at (3.71 m, -1.25 m) in robot DGPS sensor-based coordinates, with the y-axis coming out the front of the robot.

Pan (deg)	Tilt (deg)	Dots Segmented	# Segmented
-10.5	-37.5	25, 17, 13, 22, 18, 14, 15	7
-12.5	-37.5	25, 17, 13, 22, 18, 14, 15	7
-14.5	-33.5	25, 17, 22, 14, 15	5
-14.5	-35.5	25, 17, 22, 18, 14, 15	6
-14.5	-37.5	25, 17, 22, 18, 14, 15	6
-14.5	-39.5	25, 17, 30, 22, 18, 14, 15	7
-14.5	-41.5	30, 22, 18, 14, 15	5
-16.5	-37.5	29, 25, 17, 30, 22, 18	6
-18.5	-37.5	33, 29, 25, 17, 30, 22, 18, 31	8

Table 1. Images taken for high-res camera calibration.

After three tries (first two failures from mismatching values in `U.txt and X.txt`), the system seemed to work. See the MATLAB diary during calibration [1]. During calibration, the targets from all images above are combined into one data set. The calibration procedure involves iteratively minimizing the error between measured and calculated positions of the red dots. During the process, two iterations looked significant: steps 4 and 9. Step 9 appears to be the best fit overall, but displays no inherent slant, which in the past has indicated an angle of the sheet off the perpendicular. It is doubtful that no angle existed, but perhaps the calibration procedure accounted for it? Another problem is that dot #33 (in the -18.5, -37.5 image) is off in a manner unlike the others. However, the other points appear well clustered around actual dots. In step 4, we see a trend that may indicate an overall sheet yaw angle, but the points are much less clustered around real dots (up to 4 cm off). But dot #33 fits the yaw trend, unlike in step 9. The calibration parameters of step 9 will be used.

Summary and Conclusions

- Mounted arm, spectrometer and panoramic camera.
- Grabbed images from high-res for:
    - High-res calibration (9 images).
    - Tuning SAS segmentor parameters (10 images).
- Tuning classified segmentor parameters (about 25 images). To get these images, I used target acquisition to find the rocks, then the sensor manager to zoom in.
- Overall, had good success with autonomous zooms and image capture. Failed 3 times on one rock that was on shadowed blue ice. Could try to zoom in a bit more if it would help the

classifier.

- Still need to check centroid calculation code.
- Is the color balance too far off? Unfortunately, there's no good way to calibrate here.
- Improved performance by playing with new parameters for the segmentor based on lighting and terrain: sun, shadow, ice, snow. Keep track of these trends.
- Some fogging on high-res camera.
- Grabbed panoramic and arm camera images.
- Panoramic camera seriously fogged up.
- Haven't yet checked alignment of arm camera rotation angle.
- Performed most of high-res calibration procedure.
- Set spectrometer attenuator and verified that signal strength looked good even after cable shielding cracked.

Experimenter's Notebook - January 15, 2000

Segmentation Findings from January 14

- *targetAcqMaxStddevRatio* in `SAS_Config` is now set at 50 (see Appendix C). This is good for all conditions.
- *targetAcqPercentage* changes depending on weather? Yesterday (partly sunny) it worked at around 0.70. On a bright day at Willy Field, it worked at 0.55, so we believe that when the sun is bright it should be set < 0.70. If it is set higher, more false positives seem to appear.
- Only on sunny days did *targetAcqPercentage* not equal *armCamPercentage*.

Arm Testing

- Arm stowed properly.
- Positioned arm at (-1 m, 1 m) arm-relative position. Position looked good.
- Stowed arm again when an oil leak was detected from Nomad's generator.
- Spectrometer head intrudes well into arm camera image - need to move back or visual servoing will be adversely affected. Spectrometer will be removed for the rest of these tests.
- Visually servoed to (-1 m, 1 m). Found 3 targets, failed in IK (bad segmented blob chosen as real target?). The false targets were found along the sides of the image. Need to update code to the version written in McMurdo so the arm camera's segmentation can be tweaked without changing those of target acquisition. Should wait for new CVS repository to be completed before changing code.

CVS Tagging

On `meteorite:/usr/mwagner/nomad`, the following directories were CVS committed to the `needsfixed` repository, and tagged as `Jan13_2000`:

- Science autonomy code
- Video teleoperation code
- NDDS messages

Now changes made on `sweeper` can be copied to `meteorite`. This will change the following components:

- Target acquisition
- SAS configuration code
- Database
- Sensor manager
- Arm controller code
- Camera controller code

Panoramic Camera

Dewarped a panoramic image: it is very pixelated and low resolution, especially at the bottom. See Figure 3. One reason for this could be that the mirror and lens are misaligned.

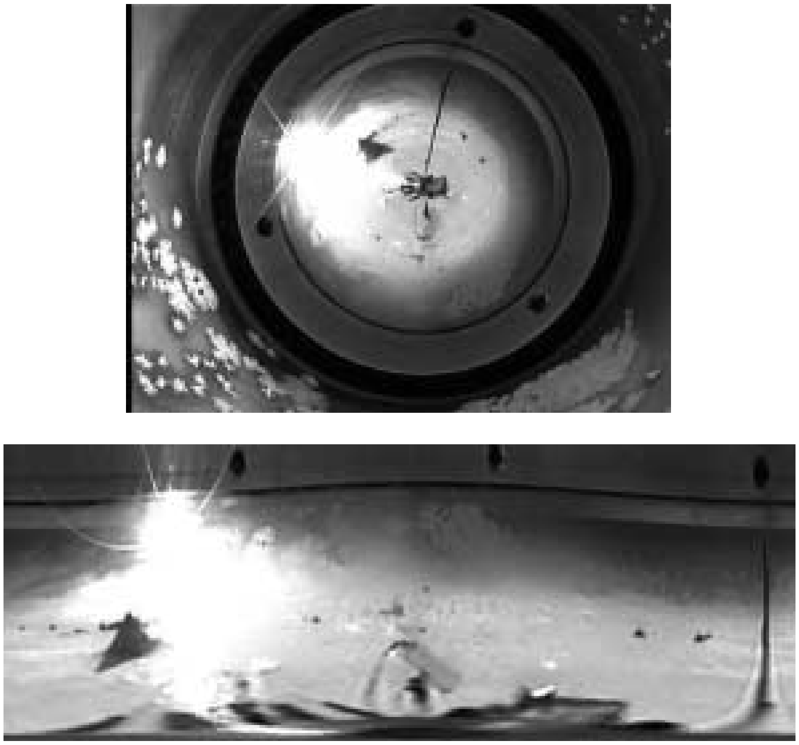


Figure 3. Warped and depwarped panoramic images taken in camp.

Continued Arm Testing

The visual servoing algorithm moves the wrist horizontally for multiple perspectives from the camera to the target, a process similar to stereo vision. It draws multiple lines between the camera and target in each position, finds the lines' bisection and uses it as the new target location. If multiple targets are found and different ones close to the center of the image, this bisection may be way underground and way off in x and y. Maybe don't let it be too far down underground - use this as an error detecting condition?

Test deployed to (-1 m, 1 m). Visually servoed to rock after 2 failed attempts. Manually commanded arm twice. Failed because the segmentor was still using red segmentation (used in Pittsburgh for development). See Figure 4.

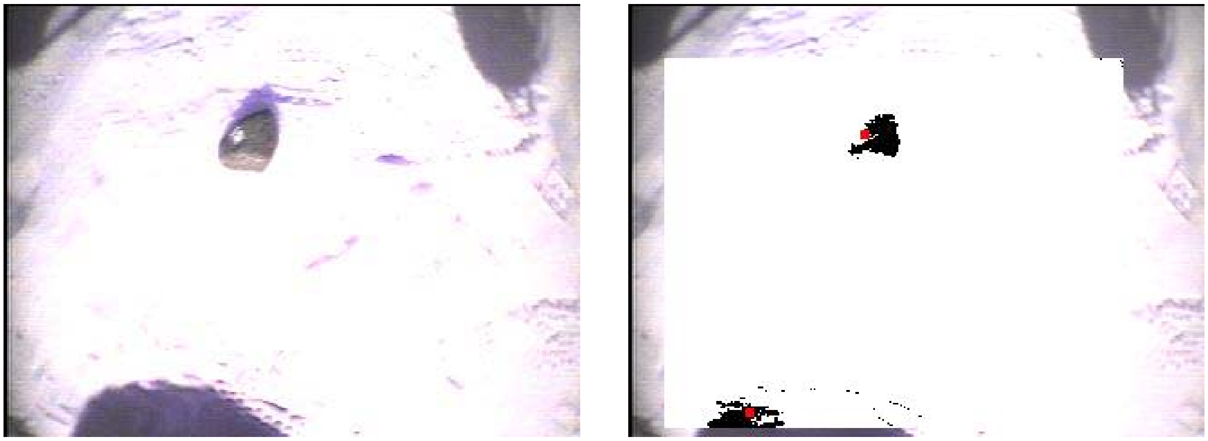


Figure 4. Raw and segmented images from arm camera when deployed to (-1 m, 1 m)

Test deployed arm to an initial estimate of (0 m, 1.2 m) on the same rock as above. Failed to find it well, as it found a few other targets (see Figure 5). Changed *armCamMinCol* to 20 and *armCamMinRow* to 30 (see Appendix C).

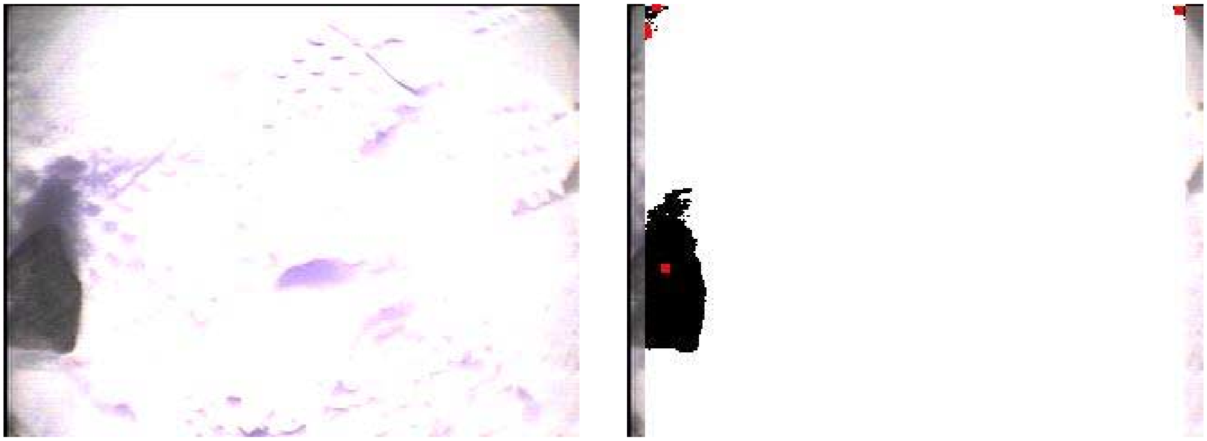


Figure 5. Raw and segmented images from arm camera when deployed to (0 m, 1.2 m)

Tried again. Still lost the target. Changed *armCamMaxCol* to 310.

Tried again. Found target, but didn't align wrist well (this was probably not a segmentation problem).

Moved robot onto blue ice without shadows to verify the procedure in the simplest case (segmentation has always worked better on blue ice than snow). Deployed to the same rock, again with an initial estimate of (0 m, 1.2 m). Found the rock centered in the 2nd step, so triggered "already centered" code and immediately moved the wrist down on it.

Note that the rock was centered but was segmented strangely, seeing it as 4 targets because of the rock's texture (See Figure 6).

Now tried going to the same rock, but with an initial estimate of (0 m, 1.1 m). Same rock, same position, same result.

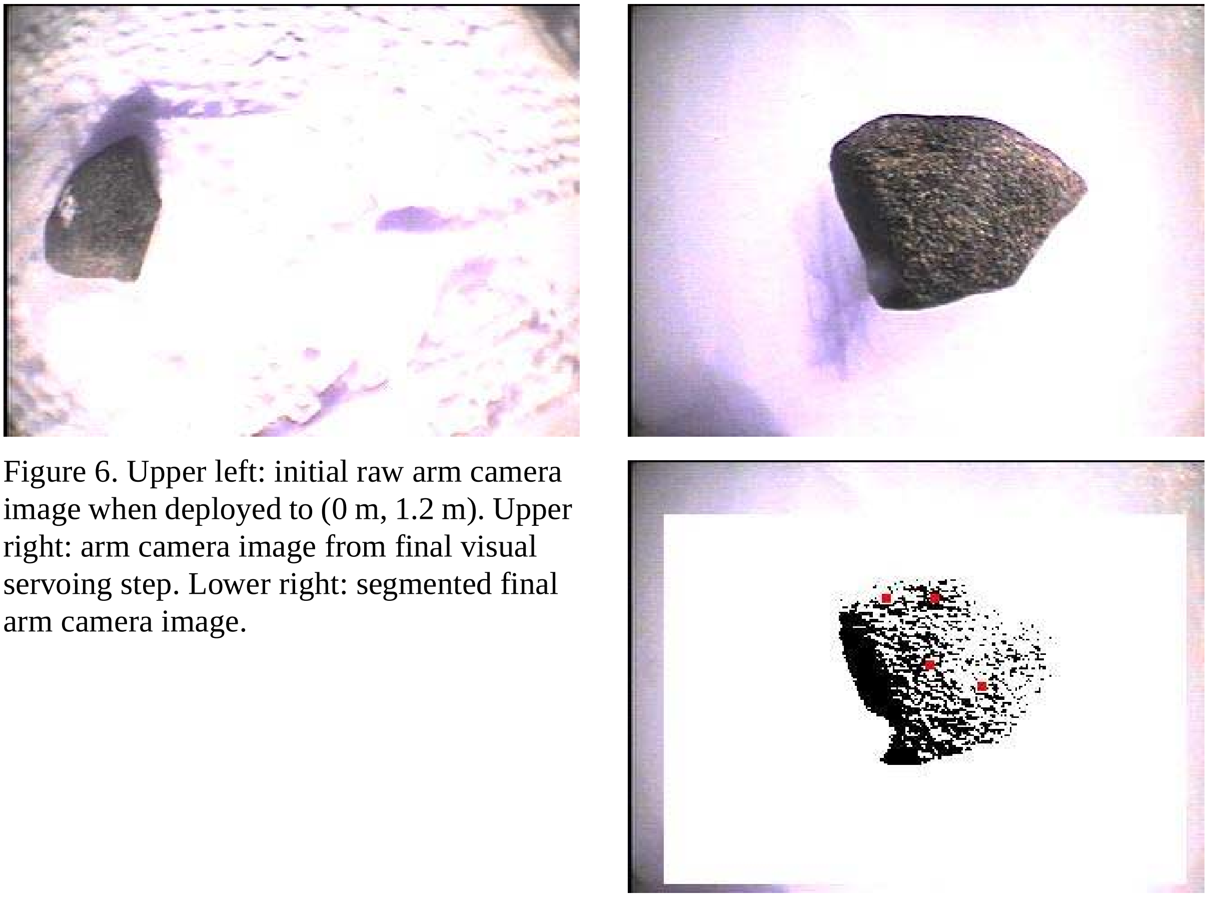


Figure 6. Upper left: initial raw arm camera image when deployed to (0 m, 1.2 m). Upper right: arm camera image from final visual servoing step. Lower right: segmented final arm camera image.

Next, moved the robot into shadowed blue ice (the robot was casting the shadow in front of itself). Deployed to a rock with an initial estimate of (-0.6 m, 1 m). Rock was on shadow boundary. The visual servoing steps were executed as follows:

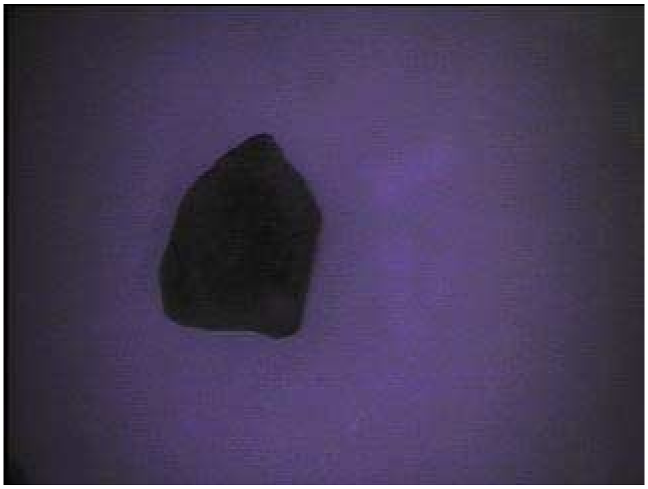


Figure 7. Final arm camera image when deployed to (-0.8 m, 1.3 m)

1. Target @ row = 58, col = 100. Appears to have picked the real rock target out of a total of 5 segmented (false) targets.
2. Target @ row = 80, col = 140. Looks good. Found a total of 3 targets.
3. Found 1 target, which was the correct one. A little bit of shadow around the corners.
4. Found 3 targets. Target chosen is at row = 166, col = 251. Looks good.
5. Good segmentation.

The final result was good.

Moved the same rock deeper into the shadowed ice. Deployed to an estimate of (-0.8 m, 1.3 m). The visual servoing steps were executed as follows:

1. Found 7 targets total. 2 were the rock. Target is @ row = 56, col = 137. Looks like it picked a rock target.
2. Found 3 targets, chose the rock row = 78, col = 167.
3. Segmentation good. Found the rock.
4. Segmentation combined the rock and a surrounding shadow into one target.
5. Segmentation split up rock into 5 targets, like an above case.

Result was good (see Figure 7).

Classifier / Spectrometer Tests

Issues: bellows could not be used because the cut out too much of the arm camera images.

Note: spectrometer automatically turns lamps on at startup.

Note: software could not read if lamps were on or off (digital I/O in didn't have proper information). Therefore, just assume that the digital output operation has worked properly.

Problem: scaling is done incorrectly in high-res images. This may be due to a software bug, because the process seems correct.

Problem: MATLAB spectrum file reader in classifier won't read filenames starting with a number.

Idea: perhaps the quality of spectrometer calibration can be quantified by comparing it to a hand sampled "good calibration". For a well-taken reference spectrum, the absolute difference between the two should be small. Therefore, we can describe the reference spectrum's quality by:

$$\text{Quality Rating} = \int \left| (\text{known good ref spec }(\lambda) - \text{ robot sampled ref spec}(\lambda)) \right| \delta\lambda$$

Increased quality of spectrum results in a lower quality rating.

Experimenter's Notebook - January 16, 2000

To do

- Automate spectrometer calibration
- Test arm height accuracy
- Quantify the quality spectrometer calibration
- Run classifier on different samples (dolerite, meteorite, with different spectrometer placements)
- Fix database save problem
- En/disable pauses in visual servoing at runtime

Arm Testing

Set arm height to 20 cm. Measured 21.5 cm -- height differences probably caused by ground height changes.

Spectrometer calibration target is measured at a height of 37.3 cm from the ground.

Note: disabled the reset arm bit on DAQ card. The default state of this DAQ card output bit is cleared. However, the arm reset line is active low. This would cause the arm to be reset by default, which often caused the arm to crash into the ground.

Classifier / Arm Tests

These types of tests involve presenting the robot with a rock sample; the robot does not find these targets during a patterned search. The robot acquires this new target placed in front of it and autonomously deploys its arm. Therefore these tests provide results about the quality of DGPS estimates, visual servoing, sensor readings and classification results. Each sample has at least one target ID (multiple IDs result from segmenting the rock into multiple small targets).

IDs 0 to 2 are Bill's meteorite.

IDs 0 to 1 had the meteorite placed too far away to servo the arm to. Could only deploy the camera. Classifier had problems on ID 0, and on ID 1 the `hiResSensor` didn't provide a good rock centroid coordinate.

We moved the meteorite closer and commanded Nomad to reacquire it for ID 2. Only deployed arm. P(Extraterrestrial) = 0.46.

Moved the meteorite out a bit, acquired as target ID 3. Had to restart sensor manager because `nddsStartDaemon` was accidentally stopped. Took high-res image. Classifier was able to segment, even though the centroid was still off. We had to calibrate spectrometer again. Took a spectrum, but the lights were off just before it turned them on to take the sample. Classifier gave P(Extraterrestrial) = 0.22. The resulting spectrum was bad -- perhaps initial calibration was bad.

ID 4 was still meteorite, in the same spot. Took no high-res image. Redid calibration after moving the spectrometer calibration target a bit more into the spec lamps. The spectrometer is 1.5 - 2 cm from the calibration target. Same weirdness in the resulting reflection spectrum -- either the sample spectrum is brighter than expected or the reference is darker than expected.

ID 5: Liam's rock #1. ID 5 is this rock. ID 6 is bogus. Deployed high-res camera on rock. Successfully segmented rock in image. P(Igneous) = 0.55, P(Extraterrestrial) = 0.15 after image classification. Arm couldn't get to ID 5.

Acquired IDs 7, 8, and 9. They are all on the same rock, rock #1. The rock was in the robot's shadow. Deployed arm only on ID 7. Visual servoing seemed to be a bit off, but at least the rock is on the side of the spectrometer. Spectrum seemed to be good, results seemed promising although it could've been a bit corrupted by ice.

Acquired IDs 10, 11 and 12. The rock is Liam's rock #2, and is very white -- much whiter than any meteorite. Deployed arm to ID 10. Visual servoing seemed to work well even though the rock was so white. Spectrum seemed bad to our visual inspection, but somehow the classification result seemed reasonable.

See [2] for more results and a description of each sample.

An ideal white reference is depicted in [3].

Problem: the `hiResSensor` scaling fixes made yesterday had a bug. Fixed today, and recompiled.

Problem: target centroid in `hiResSensor` not always reliable. The classifier displays the image segmented and marks the given centroid. By visual examination these can be far from the actual centroid, near the rock’s outer edge. Need to change this.

Problem: spectrometer lamps are not fully illuminating calibration target. Moved target a bit - seems better.

Note: segmentation works well in visual servoing on overcast days like today.

Note: need to add code to automatically calibrate spectrometer in arm software.

Error: database still breaks when saving? Don't know why.

Problem: spectrometer lamps not turned on BEFORE taking sample spectrum.

Improvement: zoom in high-res camera more, if possible.

Conclusions:

- Move spectrometer sensor head closer to target - this means it will occlude more of the arm camera image, but it's necessary to get good readings.
- Verify that the calibration target is being illuminated well.
- Get lamps working right.
- Check out database saving ASAP.
- Implement calibration quality measurement.
- Zoom in high-res more?

Pattern Test

A pattern test involves Nomad executing a search pattern and examining targets found along the way. The entire procedure is autonomous, although the robot can be paused during the test to evaluate results, verify findings and possibly even change software.

Test started test at 13:05 local time.

Rocks from Liam's collection (representative of the moraine) placed on blue ice.

Using target acquisition (in test logging mode), sensor manager, `hiResSensor`, database and classifier. Both camera and spectrometer will be deployed.

Transcript from the test:

ID 0: a real rock

Deployed high-res, classifier segmented resulting image properly. Performed spectrometer calibration. Deployed arm. Couldn't see a rock in first arm cam image, so the ID was thrown out and the arm stowed. DGPS estimates were probably off.

Restarted mission planner and classifier. The mission planner was not responding to commands, so the system should be started fresh.

Observer with the robot said the robot passed two rocks without detecting them. Cause unknown: bad segmentation? Rocks outside target acquisition area, which is very small?

IDs 1 and 2: real rocks. One might be the same as ID 0? Deployed high-res to ID 1. Classifier segmented properly. Arm was deployed to completely the wrong place.

Restarted the row, with rocks spaced further apart so the robot observer could tell what target the arm is trying to go to, to get an idea about what type of error is being encountered. Navigation planning software reported a 3 deg heading error, which could result in a error of 33 cm.

Found target ID 3, failed to get a good image. Failed to deploy arm. Position in arm coordinates (0.25 m, 0.50 m).

Found target ID 4. Target acqusition is running slower than it will during real demonstrations because every image captured is saved to disk and logged. Several targets required backing up for high-res images because by the time the mission planner knew about a new target it had traveled too close to it. High-res failed to see anything.

Found target ID 5. High-res segmented a shadow that was closer than the rock was, and found it instead.

Restarted `targetAcqMain` without saving images. It will therefore run faster.

Acquired ID 6, which appears to be the same rock as ID 5. This time it found the rock instead of the shadow, and zoomed in properly. Robot still had to back up however. Classifier segmented the image properly. Arm was deployed poorly again. Position in arm coordinates was (-0.40 m, 0.60 m).

Replaced the calibration parameters found on January 14 with those calculated on December 30 at Willy Field. This calibration had worked well in McMurdo. Tried again.

Found target ID 7. Didn't have to back up, but failed to get anything in the high-res image.

All errors measured are around 1 meter. This is more than can be expected from the apparent compass error.

Error: targets are NOT being re-acquired before deploying arm. This probably requires recompilation.

Error: arm deployment failed to pick up rock on the initial visual servoing image. Arm servoing worked when the robot is not moving, so the initial DGPS estimates are very wrong. The error could be caused by the robot's pose (roll, pitch or yaw) being nonzero during calibration. In Pittsburgh this was not a problem, since we calibrated on the floor, but on the ice it certainly could be different.

Stopped test because DGPS estimates were too far off to continue. The high-res camera should be recalibrated.

See [4] for more results.

Recalibrating the Camera

The camera was recalibrated, this time sweeping the tilt angles from -41 to -31 degrees in steps of 5 degrees with a nominal tilt of -37.5 degrees, and pan angles from -5 to -20 degrees in steps of 5 degrees with a nominal pan of -14.5 degrees. Here are how the dots on the calibration sheet were numbered. Note that this time, it starts with 1 rather than 5. Also, note that it is quite difficult to lay the sheet down flat on the rippled ice when any type of wind is blowing. Next time a better way of doing this should be used - perhaps something attached to the robot, or perhaps a rigid sheet.

The sheet’s dots are numbered:

1	5	9	13	17	21	25	29
2	6	10	14	18	22	26	30
3	7	11	15	19	23	27	31
4	8	12	16	20	24	28	32

The dots are 30.5 cm apart in x and y directions. Dot 1 is at (3.95, -0.84) in robot DGPS sensor-based coordinates, with the y-axis coming out the front of the robot.

Pan (deg)	Tilt (deg)	Dots Segmented	# Segmented
-10.5	-37.5	2,5,6,7,8,9,10,11,12,13,14,15,16,17,18	15
-14.5	-31	5,6,7,9,10,11,13,14,15,17,18,19,20,21	14
-14.5	-36	5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21	17
-14.5	-41	6,7,8,10,11,12,15,16,19,20	10
-15	-37.5	5,6,7,8,9,10,11,12,13,14,15,16,18,19,20	15
-20	-37.5	9,10,11,12,13,14,15,16,18,19,20,22,23	13
-5	-37.5	2,5,6,7,8,9,10,11,12,13,14	11

Table 2. Images taken for high-res camera calibration.

See Figure 8 for an image of the calibration sheet. Calibration parameter results can be seen in [5]. There were two outlier dots left at the last step - these could've been caused by a typo.

We also re-measured the distance in x from the PTU to the robot's DGPS sensor in the robot's frame. We found it to be 0.65 m, which was very close to the existing measurement.

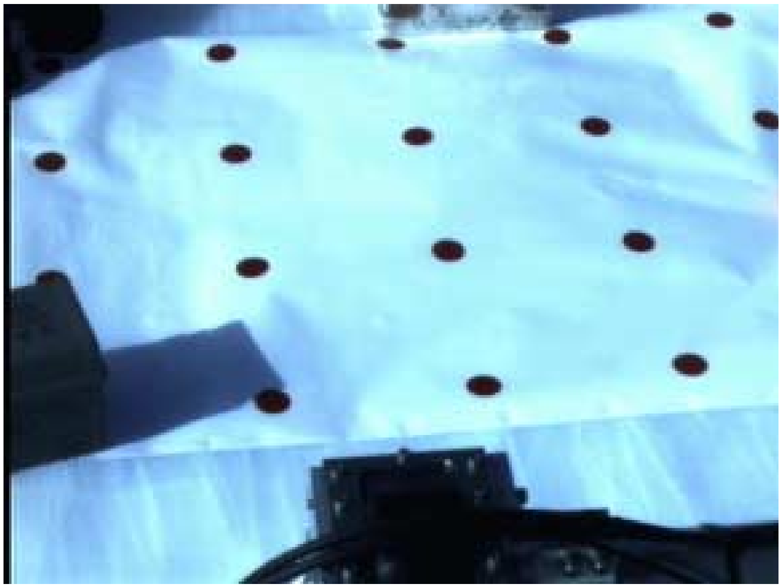


Figure 8. Calibration image taken at pan = -10 deg, tilt = -37.5 deg

Second Pattern Test

With the new calibration, we attempted another test of the system, similar to the first pattern test performed today. The only difference was that target acquisition was run in regular mode, so images were not saved and the system ran faster. See [6] for all of the results.

Conclusions:

- Jog switch software is not reliable.
- Having problems with high-res camera finding the arm instead of a rock.

Experimenter's Notebook - January 17, 2000

Code Development

- Spectrometer lamps now turn on immediately after each dark reference spectrum.
- Centroids of targets in high-res images are now calculated more correctly, although there are cases where the new method would fail. Now we simply return the point halfway between the first (upper left) and final (lower right) corners as the centroid.

  Centroid row = (firstPixel.row + lastPixel.row) / 2

  Centroid col = (firstPixel.col + lastPixel.col) / 2

- Added *hiResTargetZoomVal* to `SAS_Config`. It can be set to 140 (it's current setting) or any other 8-bit number and the software will step down to this lowest (i.e. most zoomed) value.
- Updated `dbServer` to timestamp archive file names.

Hard Drive Failure

Some directories on `meteorite` started having “I/O Errors” when a user would attempt to `cd` into them. Then the next time it was booted up, it never appeared on the network. The robot was opened and a portable keyboard / monitor was connected. It was asking for the root password to drop to a shell. The `/dev/hda6` (7.5 GB) partition was fsck’d manually. However, each time it rebooted afterwards, the partition failed. So the computer was removed from Nomad and taken to the tent to have its hard drive replaced. The spare hard drive was unpacked and placed in `meteorite`. This drive was identical to meteorite's hard drive when it left Pittsburgh in December. It booted fine, so all of the source code was updated and recompiled.

Arm Testing

This test was designed to take data for arm segmentation tuning, and to test visual servoing with the new spectrometer placement. When the wrist is moved into calibration position, the arm camera sees something like the image in Figure 9. Notice the spectrometer head jutting out of the lower left corner of the image.

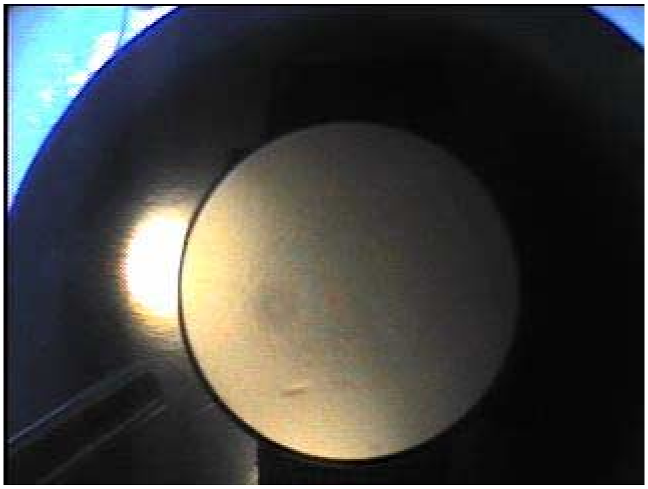


Figure 9. Spectrometer calibration target seen from arm camera

Nomad attempted to visually servo to a rock. See the images in Figure 10 for the results. In the first step, shown segmented above, there are two targets: the rock itself and the spectrometer head. The first step worked, however, because the rock's centroid is closer to the center of the image. The visual servoing algorithm picks the target closest to the center of the image. However, in the second image, several problems occurred. First, the spectrometer head is being segmented, as in the initial

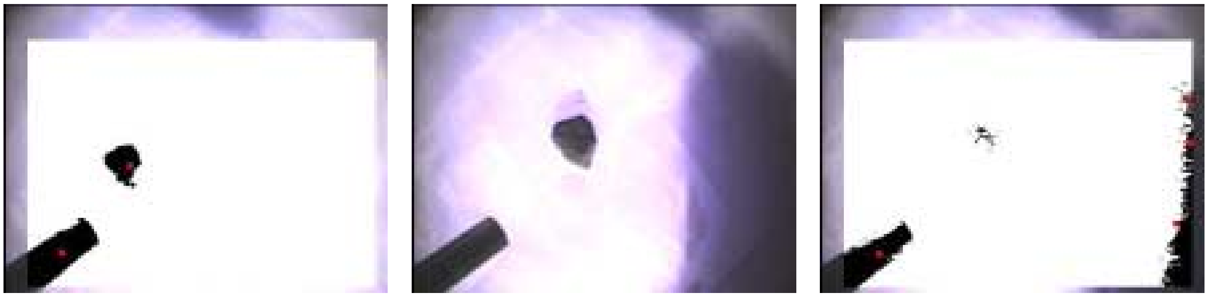


Figure 10. From left to right: segmented image from first visual servoing step, raw image from second step (notice the robot’s shadows on the right) and segmented image from second step

step. However, in this case, the rock is not segmented - instead, shadows on the right side of the screen are dark enough to themselves be segmented. These two things cause visual servoing to fail. The rock was most likely not segmented because shadow compensation probably altered the intensity both the shadows on the right and the rock itself.

Another error condition would be if the rock were further from the center of the image than the spectrometer head.

The spectrometer head being segmented causes one other problem - if the segmentation fails to see the rock, it will never pause for user intervention because there's always at least one target.

Two initial solutions were proposed:

- Put bright red sleeping bag patch-up tape around sensor head. Unfortunately this doesn't work - the (R G B) of the red tape in the shadow is (23, 21, 20). This is more black than red.
- Crop out a region of the screen. If it's a rectangular region, this will hurt visual servoing by removing a decent percentage of the image. We could mask out the spectrometer, but we'd have to pad it a bit to allow for changes in lighting that will change segmentation results.

Finally, we decided on another idea. We added *armCamMaxSpecRow*, *armCamMaxSpecCol*, *armCamMinSpecRow* and *armCamMinSpecCol* to `SAS_Config`. This defines a region of an image in which the spectrometer's centroid will be located. The software can then simply ignore any targets whose centroids are in this box. Note that this is different from cropping out that region -- a rock can be partially occluded by the spectrometer and still be found because it would move the centroid. Right now however, we're ignoring so much of the screen that this probably won't happen. The relevant variables are currently set to:

- *armCamMinSpecRow* = 172
- *armCamMaxSpecRow* = 240
- *armCamMinSpecCol* = 0
- *armCamMaxSpecCol* = 83

Tried visual servoing again. The arm performed calibration, which looked all right. The spectrometer was 1.5 to 2 cm above rock, but the spectrum didn't look good. The rock and spectrometer were, however, well aligned from the arm camera's point of view. Still had problems with finding the spectrometer - perhaps a bug in the code?

Put *specCalibTargetHeight* in `SAS_Config` so the spectrometer calibration target height can be changed at runtime. Set it to 0.378 m.

Tried calibrating and servoing again. It still found the spectrometer, but visual servoing worked. The spectrum looked good.

Line of Rocks Test

The robot did a search along a DGPS-surveyed line of rocks (see Figure 11). In addition to simply acquiring the targets, it also ran the mission planner and sensor manager, so sensors would be deployed.

See [7] for more results.

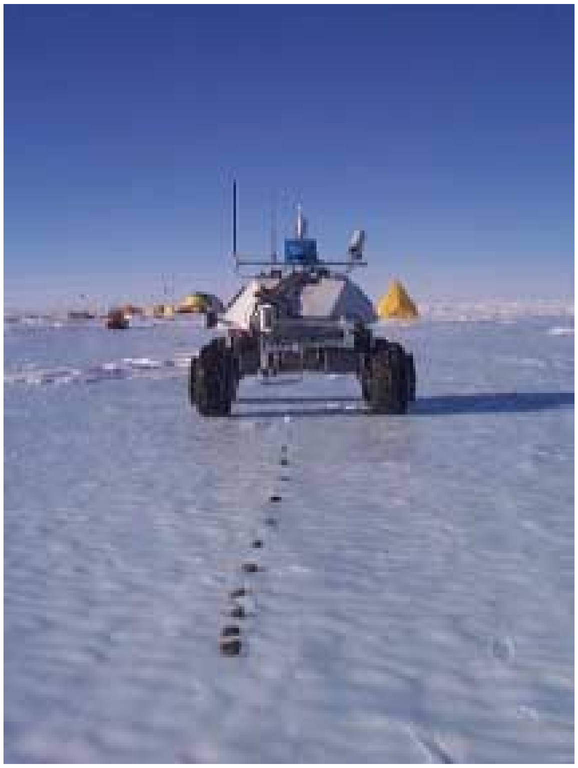


Figure 11. Nomad driving along surveyed line of rocks

Experimenter's Notebook - January 18, 2000

Segmented Spectrometer Test

This test was designed to get more experience visual servoing with the spectrometer head in the image. The coordinates of the spectrometer head in the image will be recorded each time.

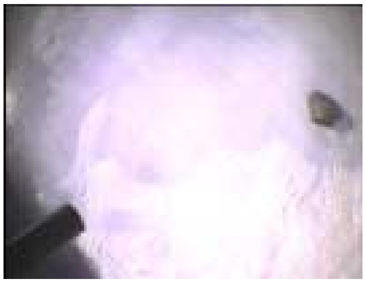
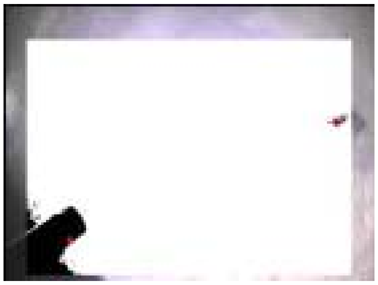
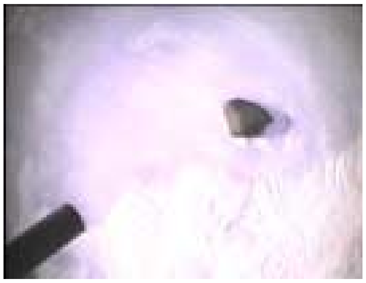
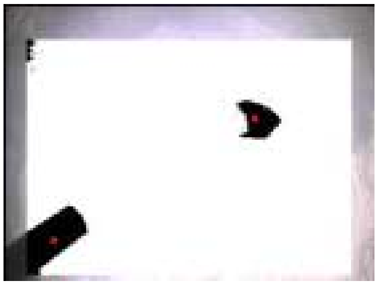
There were six visual servoing attempts: four to gather data on how large an area needs to be searched to find the spectrometer head’s centroid and another two to test the new settings.

Attempt 1

A rock was placed in front of the robot, with estimated arm relative coordinates of (0 m, 1 m). Later discovered the rock was discovered to be closer to (0 m, 1.2 m). The visual servoing saw too many shadows and failed.

Attempt 2

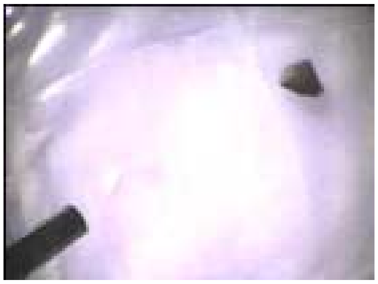
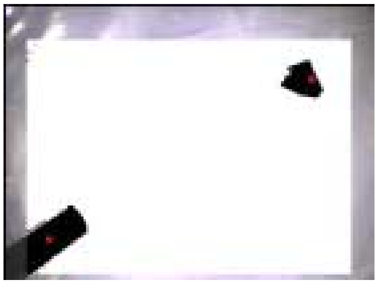
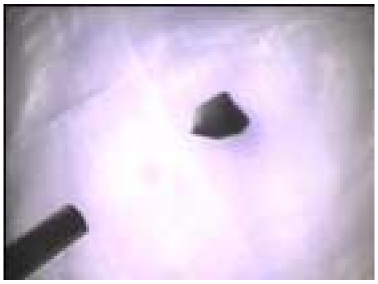
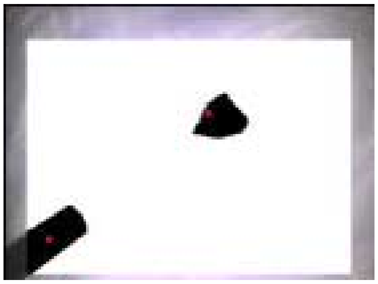
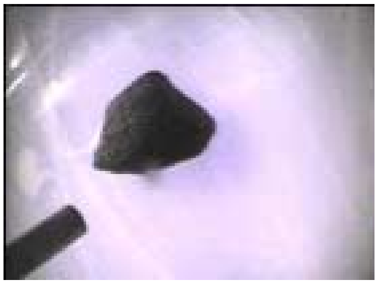
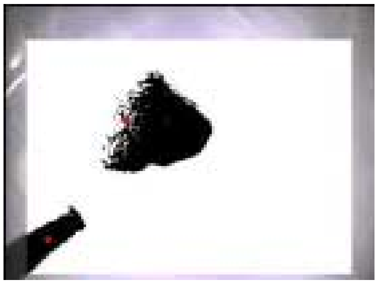
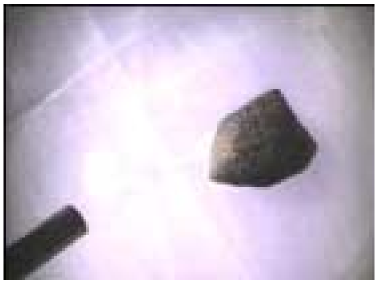
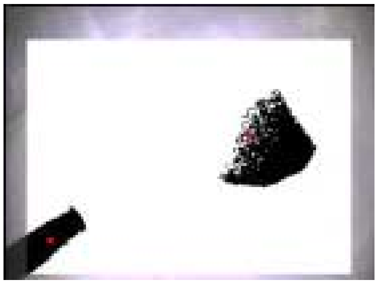
Changed *armCamMaxCol* to 300 in `SAS_Config`. However this attempt failed after the second step when it tried to move down. At the time we were not sure of what caused the failure:

	Raw Image	**Segmented Image**
Step 1		
Step 2		
Figure 12. Images from visual servoing attempt 2		

From the images in Figure 12, it can be seen that the rock was not segmented well on the first step. Additionally, shadows caused the spectrometer's centroid to shift over. These two things resulted in the spectrometer's centroid being closer to the center of the image than the rock, so the spectrometer was chosen as the target for step 1. This will always disrupt visual servoing.

Attempt 3

This time the rock was moved to approximately (-1 m, 1 m). This time visual servoing succeeded without any problems. See Figure 13 for images from the steps.

	Raw Image	**Segmented Image**
Step 1		
Step 2		
Step 3		
Step 4		

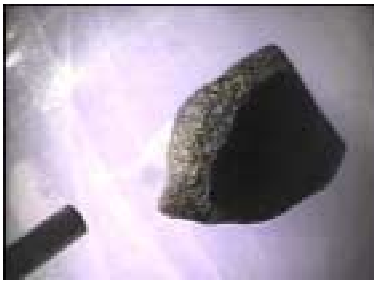
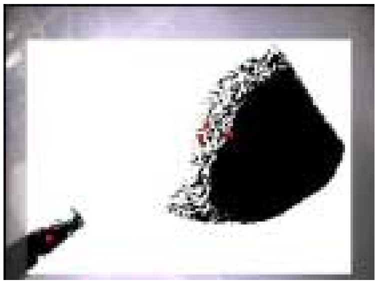
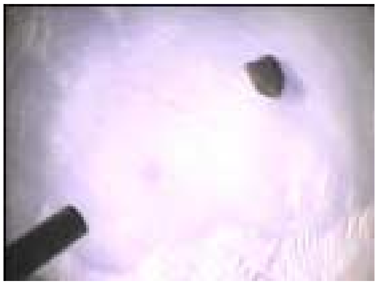
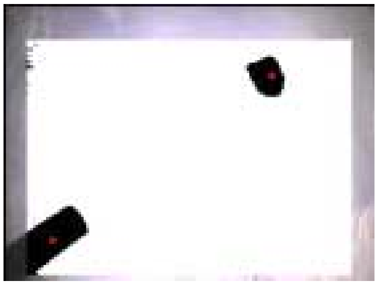
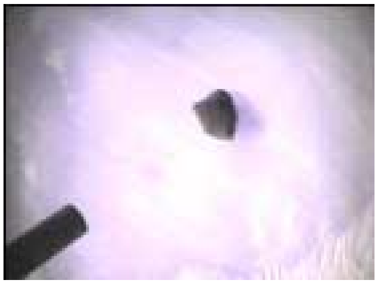
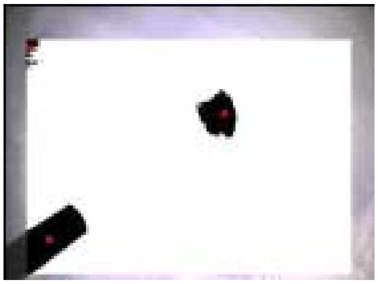
	Raw Image	Segmented Image
Step 5		

Figure 13. Images from visual servoing attempt 3

It seems a trend that as the wrist moves down and the rock gets bigger in the image, shadow compensation is probably performed, which causes the spectrometer to be segmented less properly. However, the results seem good for the spectrometer in this location.

Attempt 4

The time the rock was moved back to (0 m, 1.2 m). But again, visual servoing failed on the downward step. See Figure 14.

	Raw Image	Segmented Image
Step 1		
Step 2		

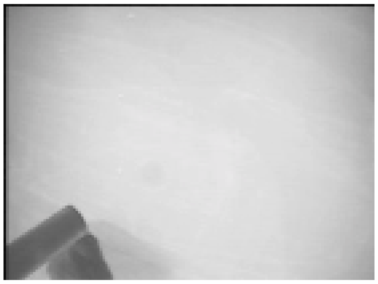
	Raw Image	**Segmented Image**
Final Image		Not available

Figure 14. Images from visual servoing attempt 4

It is unclear why the rock ended up in this position. However, it is clear that with the rock adjacent to the spectrometer, its centroid will be within the *armCamSpec** boundaries implemented on January 17, and therefore ignored.

At this point, the spectrometer centroid locations from the first 4 attempts were examined. The results are in Table 3.

Attempt / Step	**Row**	**Column**
Attempt 4 / Step 1	204	43
Attempt 4 / Step 2	205	40
Attempt 3 / Step 1	205	39
Attempt 3 / Step 2	205	40
Attempt 3 / Step 3	204	38
Attempt 3 / Step 4	206	41
Attempt 3 / Step 5	203	40
Attempt 2 / Step 1	204	57
Attempt 2 / Step 2	205	44
Mean	204.6	40.6
Stddev	0.9	2

Table 3. Calculated spectrometer centroids

These results prompted the following parameters to be changed to define an area 3 σ around the mean spectrometer centroid location:

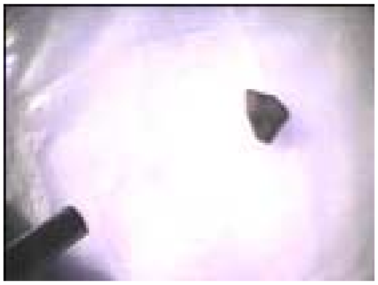
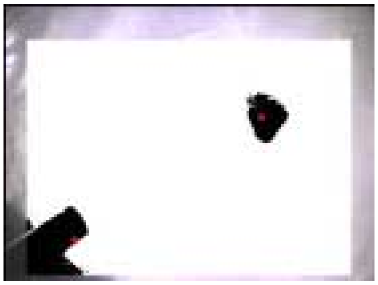
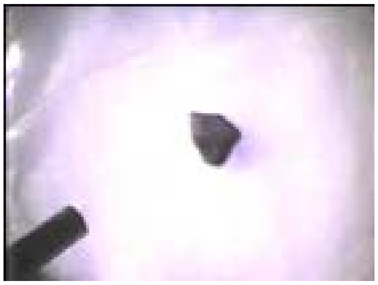
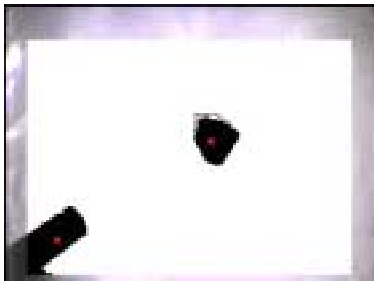
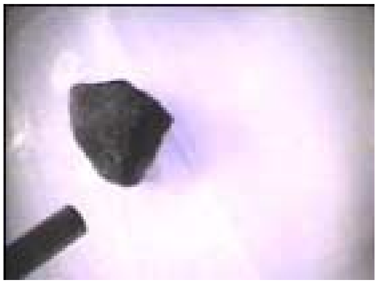
- *armCamSpecMinRow* = 205 - 3 σ_{row} = 202
- *armCamSpecMaxRow* = 205 + 3 σ_{row} = 208
- *armCamSpecMinCol* = 41 - 3 σ_{col} = 35

- *armCamSpecMaxCol* = 41 + 3 σ_{col} = 47

Once these changes were performed, two more attempts were made to see how well this implementation would work.

Attempt 5

Again, the same rock was used, placed approximately at (-1 m, 1 m).

	Raw Image	Segmented Image
Step 1		
Step 2		
Step 3		Not available

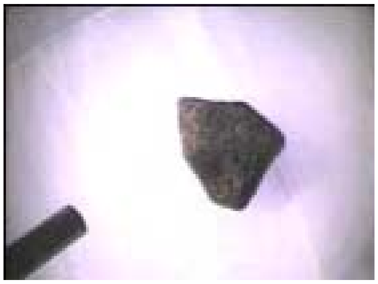
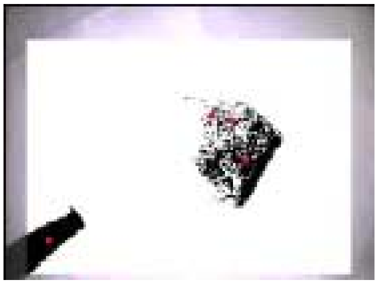
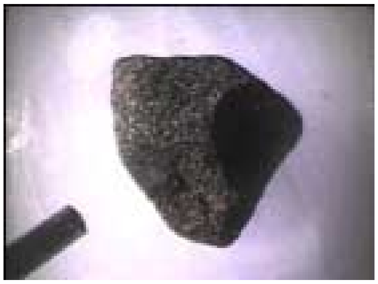
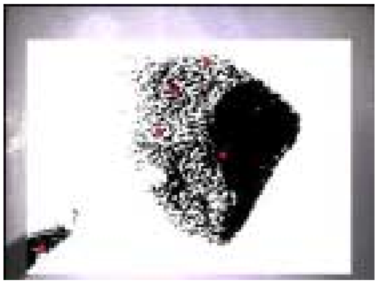
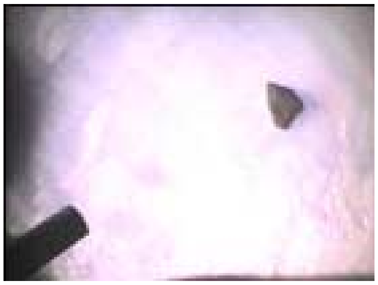
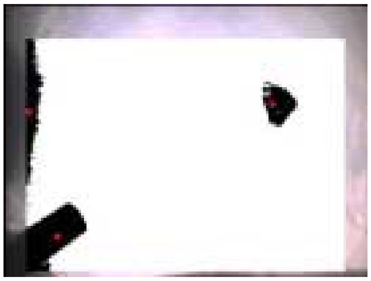
	Raw Image	Segmented Image
Step 4		
Step 5		

Figure 15. Images from visual servoing attempt 5

This was a successful visual servoing attempt, as seen in Figure 15. Again note how the rock and spectrometer are broken up during segmentation. This may be due to shadow compensation.

Attempt 6

We placed the rock at (0 m, 1 m), the same location as attempt 1. Attempt 1 failed, but attempt 6 did not. The images are shown in Figure 16.

	Raw Image	Segmented Image
Step 1		

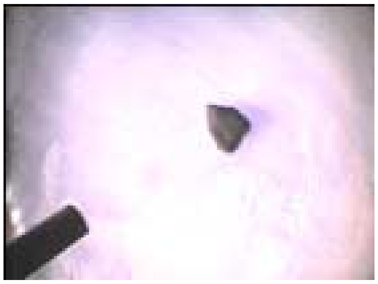
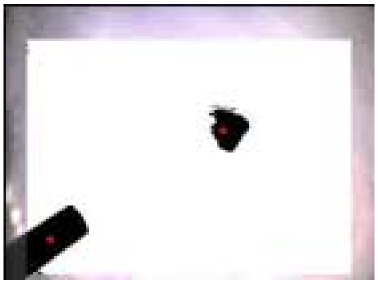
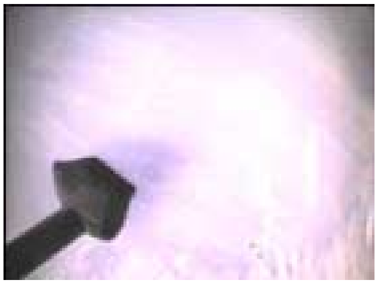
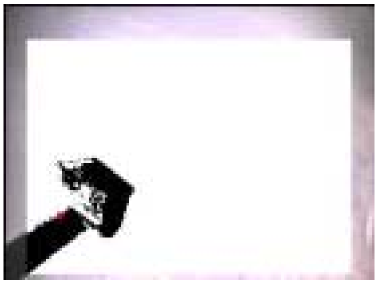
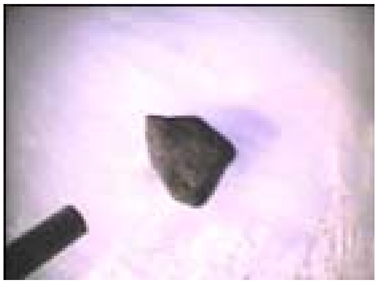
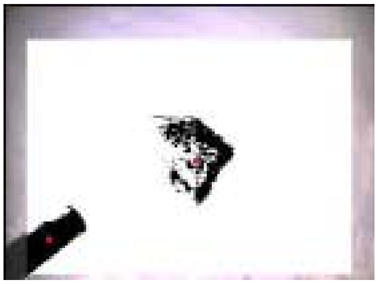
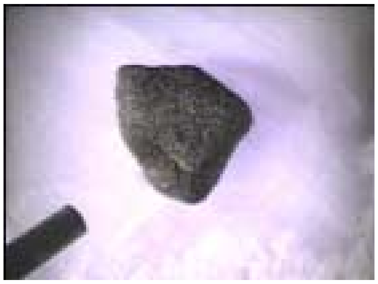
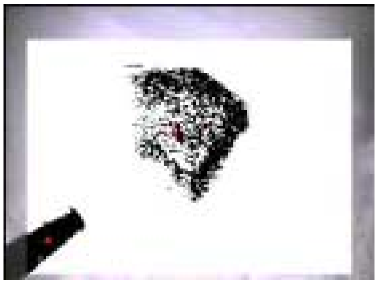
Step 2		
Step 3		
Step 4		
Step 5		

Figure 16. Images from visual servoing attempt 6

Notice that in step 3 the rock is adjacent to the spectrometer. However, this caused the centroid of to shift enough not to ignore it. This supported our *armCamSpec** parameter changes before attempt 5 -- they ignored enough of the image to generally remove the spectrometer, but not too much to find a rock adjacent to the spectrometer. Overall the results supported the code changes made to eliminate the spectrometer from a list of segmented targets.

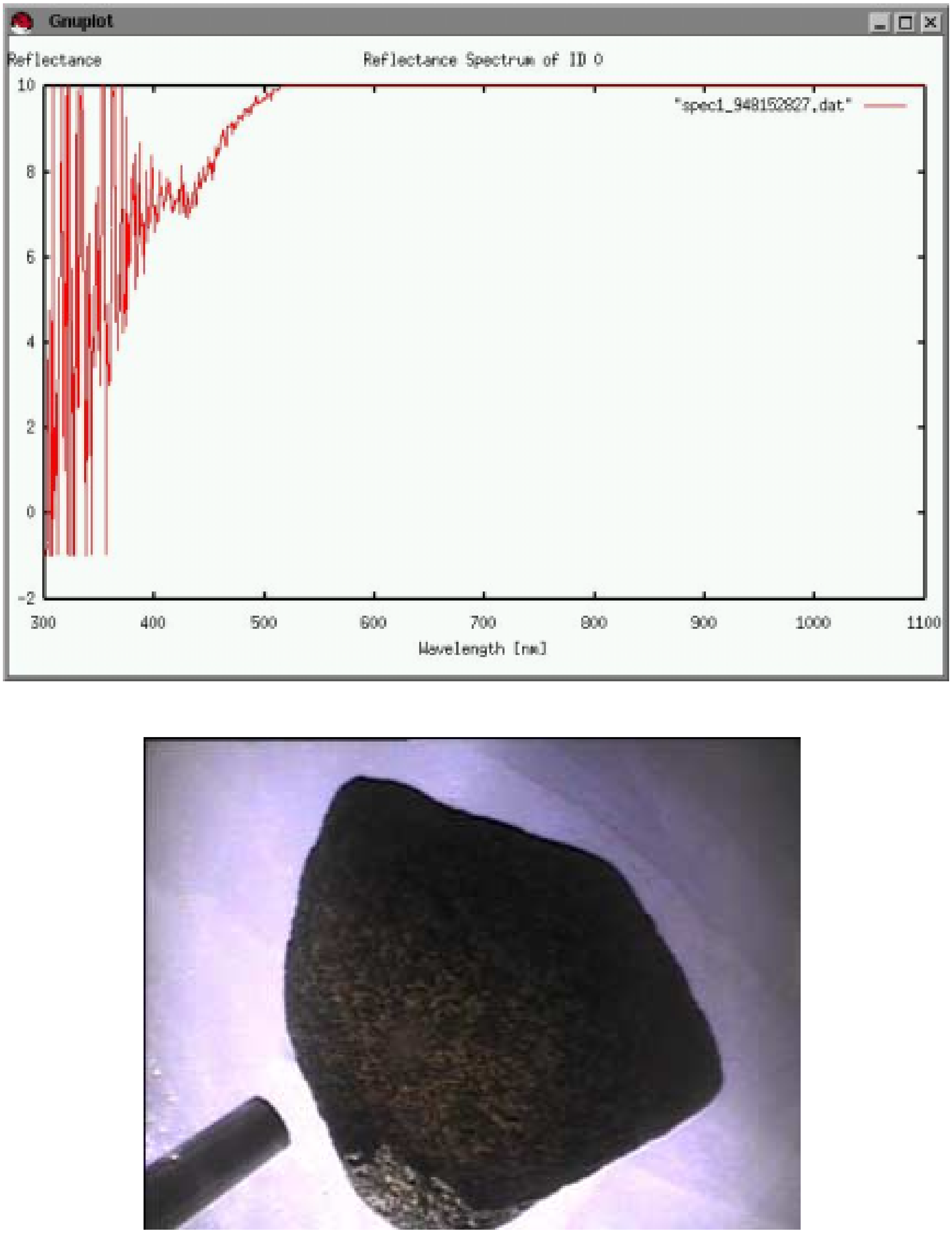


Figure 17. Top: reflectance spectrum of ID #0 (spectrometer head > 2 cm from rock), bottom: arm camera image as spectrum was being taken

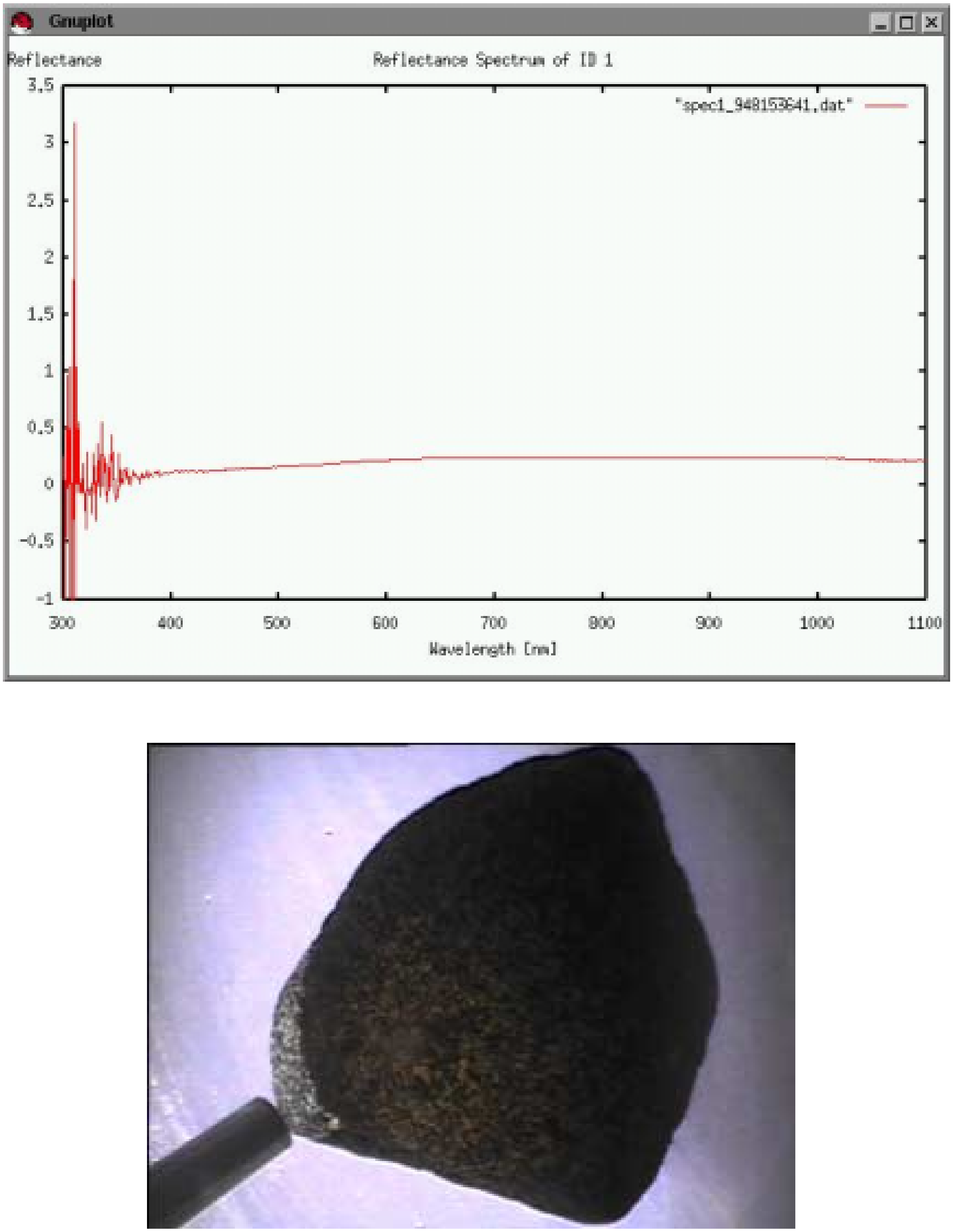


Figure 18. Top: reflectance spectrum from ID #1 (spectrometer head 1 to 1.5 cm from rock), bottom: arm camera image as spectrum was being taken

Wrist Lowering Tests

Because the touch probe is nonfunctional, manually moving Nomad's wrist down onto the rock after its last visual servoing step is required. This involves safely lowering it from a height of about 15 cm. The scheme will be to use arm camera imagery to lower the wrist through teleoperation. Someone out with the robot will be verifying that it's safe to lower each step.

Modified arm code to ask the user to jog the wrist down manually or remotely step down.

An observer went out next to the arm. An operator viewing the arm camera images from `pxc_ui`. The robot is running the new deploy down fully code. How this teleoperation setup works has yet to be determined.

Using the SAS, the arm was deployed to (0 m, 1 m). The rock was lost for an unknown reason. An observer noticed that the wrist moved down more in `ArmTestBench` than using the SAS.

Tried a second arm deployment. This attempt is ID 0. Observer still said it looked lower. However, spectrometer calibration worked. Visual servoing worked well, and the wrist was lowered, but when the spectrum was acquired, the sensor head was more than 2 cm from the target.

Tried a third arm deployment. This attempt is ID 1. Again, visual servoing worked. This time the sensor head was 1 to 1.5 cm from the rock.

Code Development

Removed some automatic report generation capabilities from the database, it was causing too many problems and isn't important enough to deal with now.

Pattern Test

Performed a pattern search near camp. Changed *targetAcqStopRow* back to 240 in `SAS_Config`, as it was in Willy Field.

Pattern had four 20 m rows towards the left.

Target acquisition was taking around 3.2 s / image. The spectrometer calibration target was moved to a new height of 38.2 cm. Changed in `SAS_Config`.

First pattern had to be aborted because human error caused SAS software to get caught in a loop.

See [8] for more results.

Code Changes

Additionally, the following variables were added to `SAS_Config`, so that all segmentation parameters would be tunable:

- *lineRatio*
- *maxStddevRatio*
- *CBlue*
- *CGreen*
- *targetAdjBoundary*
- *targetLineRatio*
- *minPixels*

Arm Camera Segmentation Tests

Tried to take more data to understand arm camera segmentation. Noticed a lot of shadows, so the *maxStddevRatio* parameter was re-tuned to get rid of them. This parameter determines when shadow compensation is performed. If it is higher, then shadow compensation will occur less often. So, we set it to 50. The resulting image is in Figure 19:

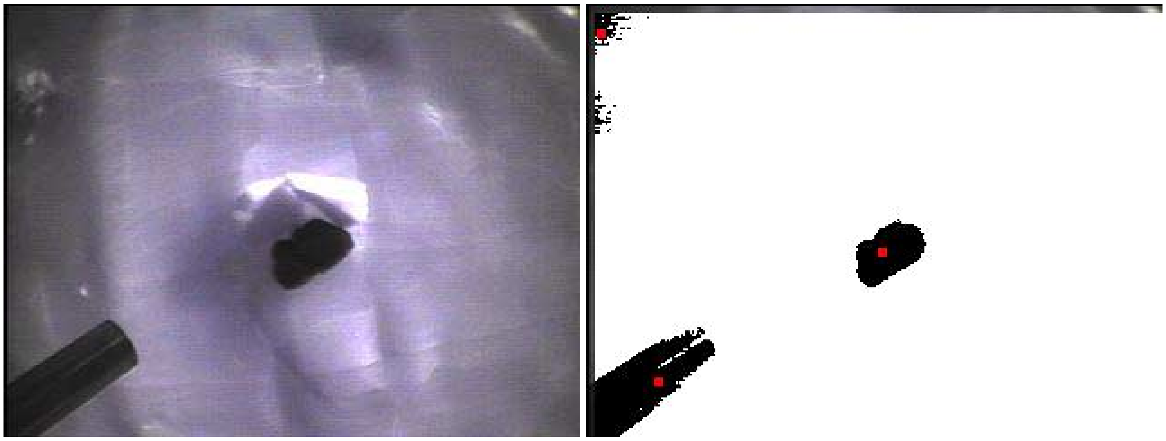


Figure 19. Raw and segmented arm camera images from segmentation test

Other Observations

As the wrist moves down, shadows around the edges actually decrease. This is probably because of the camera's auto-iris.

Rocks seemed to segment well up close due to less shadows. Less shadows means less shadow compensation which means that segmentation is not so strict.

Experimenter's Notebook - January 19, 2000

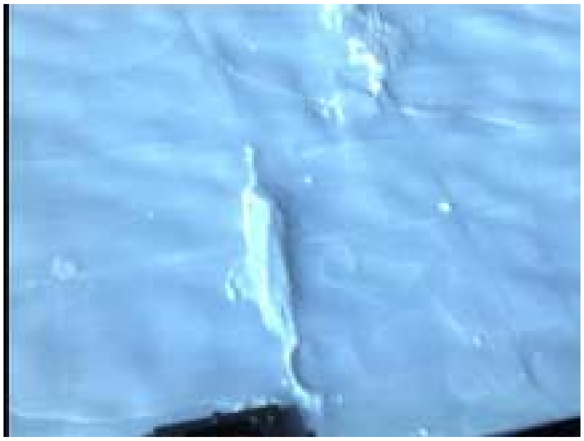
Surveyed Line Test

Changed some code to save all necessary data for this test:

- Changed arm camera code to save all initial and segmented images with time stamped names.
- Changed `targetAcqMain` to save raw target acquisition images even if no targets are found.
- Changed `acqHiResDriver` to save unsegmented image file names in telemetry.

Again the robot drove along the DGPS-surveyed line of rocks. Figure 20 shows the sequence of images that Nomad took as it was running target acquisition. When a target was found, the upper half of the image is saved as a binary image with a red dot designating where the robot believes a rock to be. The duration of each target acquisition cycle is listed under each picture (note that the target acquisition times are a bit longer than usual because the images are being saved).

4 s



4 s

4 s

4 s

4 s

4 s

4 s

4 s

4 s

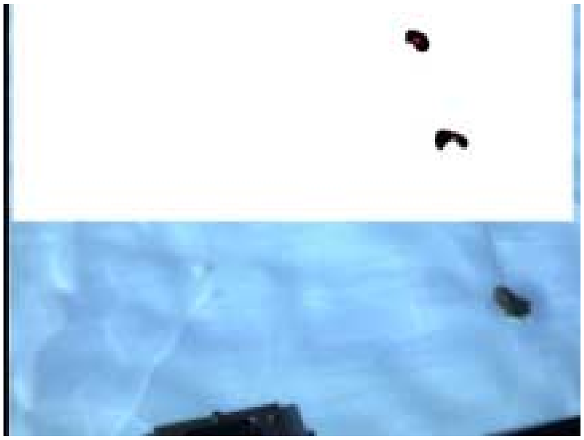
5 s

4 s

4 s

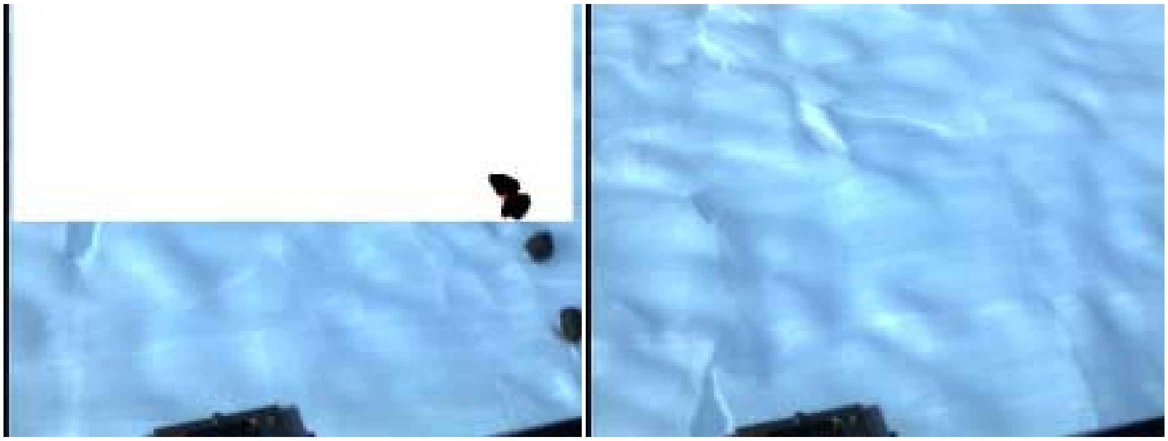
4 s

4 s



5 s

4 s



5 s

4 s

Figure 20. Sequence of target acquisition images taken by Nomad’s high-resolution camera during surveyed line test

See [9] for all targets found along with their location. In the second sheet all of the target rows and columns as well as camera state are listed for future reconstruction efforts.

Arm Camera Image Segmentation Test

Took some more arm camera images to test segmentation. This time we added the foot onto the spectrometer sensor head. The foot is used to shield the spectrometer from scraping against a rock. The aim of this test was to see if the foot could be a teleoperation aid when lowering the wrist remotely. Figure 21 shows an image of the spectrometer (with foot) performing calibration.

Note that the foot actually casts a shadow on the target. This could cause problems with calibration. A rock was placed in front of the robot, with an initial arm-relative location estimate of (-1 m, 1 m). Results are shown in Figure 22.

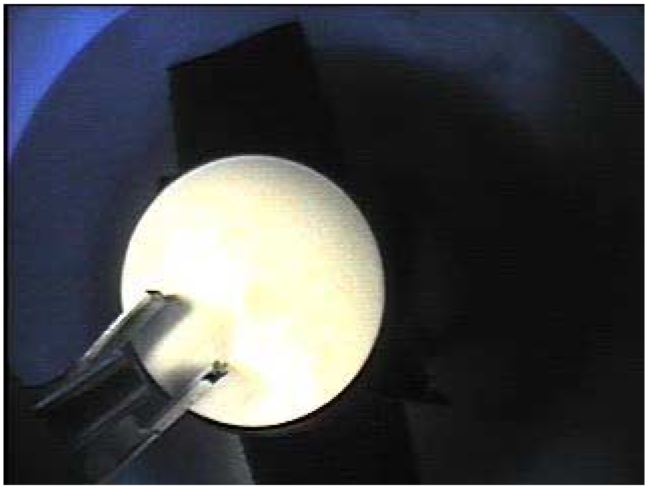
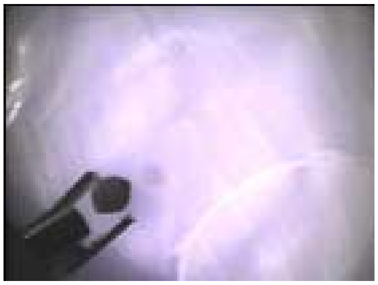
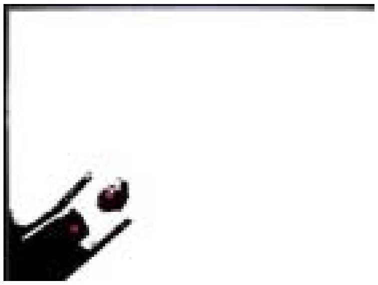
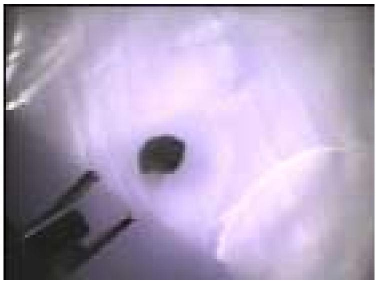
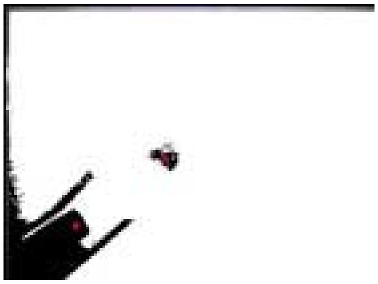
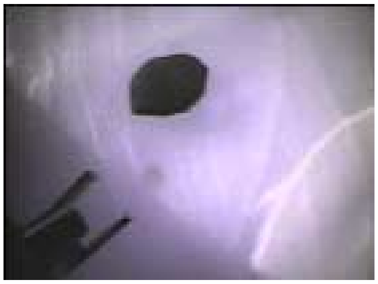
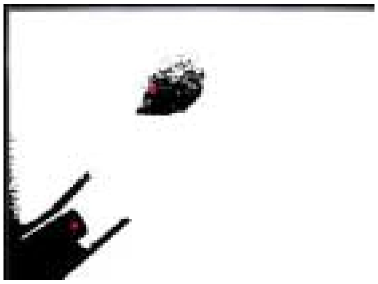
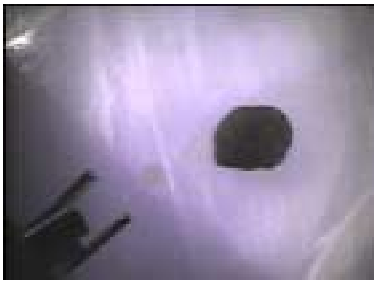
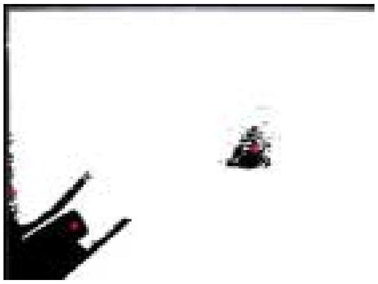


Figure 21. Arm camera image of spectrometer calibration target with foot

Notice that shadow compensation was more likely used in each image. Also, the foot was not found to be useful enough to make up for the extra image space it occupied.

	Raw Image	**Segmented Image**
Step 1		
Step 2		
Step 3		
Step 4		

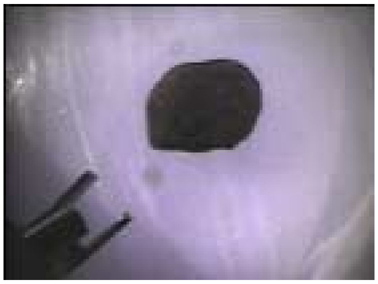
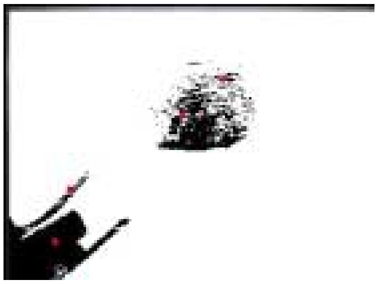
	Raw Image	**Segmented Image**
Step 5		

Figure 22. Images of visual servoing steps with spectrometer foot

From these tests the spectrometer calibration target height was also changed to 0.388 m.

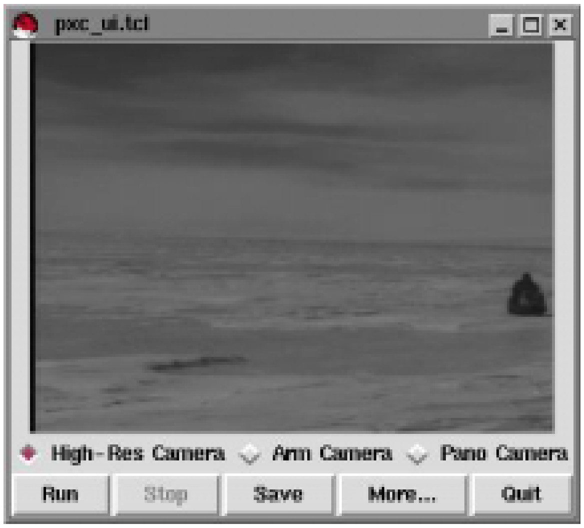
Traverse to Outer Moraine Area

The robot was sent towards the outer edges of the moraine for testing. One person had to walk with the robot, as blatantly incorrect range readings from the laser were noticed (NOTE: these readings were later verified and the laser was found to be faulty).

The following images were taken while the robot was traversing:

12:09 pm: Image from high-res camera while Nomad traversed. Pan and tilt angles are at 0 degrees, and the lens is fully zoomed out.

12:22 pm: Warped panoramic image taken during traverse.



12:48 pm: Screen shot of Nomad's teleoperation interface. It displayed live images from any of Nomad's three cameras at rates from 0.3 to 0.5 Hz, even when the robot was 1.2 km away. Images are in grayscale to reduce bandwidth. In this screen shot, a high-resolution image shows a skidoo following along side Nomad.

Figure 23. Some images taken during Nomad’s traverse into the moraine

At 1 pm, the tent's generator ran out of gas, and contact was temporarily lost with the robot.

Pattern Test

This test is similar to the pattern tests performed earlier, except it is around 1 km from camp and is searching through the outer edges of a natural moraine area.

When the arm was deployed to the first target found (ID 0), the arm camera segmentation looked quite good. Here are the centroid pixel coordinates of the spectrometer head in these image:

Row	Column
210	44
210	45
212	39

Table 4. Calculated centroid pixel coordinates of spectrometer head in arm camera images

These results led to change *armCamSpecMinRow* to 205, *armCamSpecMaxRow* to 215, *armCamSpecMinCol* to 35 and *armCamSpecMaxCol* to 50. These error margins are a bit larger to accommodate some cases previously seen.

Also implemented the capability to step the wrist down by variable increments.

At this point it was noticed that the rocks encountered by the robot varied in height quite a bit. Could they just be skipped over by ignoring large targets? If they're too big, the final visual servoing step will run the wrist into the rock. Can the final step height be raised to 0.25 m? This final height was tried on ID 0 again, but it was very high and required a lot of time for teleoperation to lower it. The final step height was changed back to 0.15 m. If a big rock is encountered, a manual intervention will tell the robot to skip over it.

For ID 3, the robot found no targets when it deployed the high-res camera. There were, in fact, no targets there. Therefore, although target acquisition found something, the high-res sensor deployment didn't. Perhaps the robot could ignore the target if nothing is found in the high-res image?

See [10] for more information.

Conclusions:

- If high-res camera deploys on a target and sees nothing, there's probably no rock, so just ignore the target from then onward.
- Set a maximum number of pixels that an acquired target can have. This way, large rocks will still be segmented, but they won't show up as target IDs in the database.
- It would be good to have an option to ignore many targets at once in the mission planner.
- Decrease *maxTargetsInData* from 20 to about 5. Also check to see if target acquisition finds more than this number of targets - if so, it's probably bogus, so ignore all of these targets. (Actually, the number was set to 10 rather than 5).

Code Changes

- Defined a specific sensor manager status to send to the mission planner if no targets are found in high-res image data. This status will inform the mission planner that it should ignore the current target ID.
- Decreased *maxTargetsInData* from 20 to 10.
- Added *targetAcqBottomMaxPix* (set to 40000), *targetAcqTopMaxPix* (set to 12500) and *armCamMaxPix* (set to 76800) to `SAS_Config`. Used these to ignore big rocks in target acquisition code.

Experimenter's Notebook - January 20, 2000

Wind chill today was about -40 C.

Code Development

Took an image of two larger rocks on blue ice with the high-res camera in target acquisition position (see Figure 24).

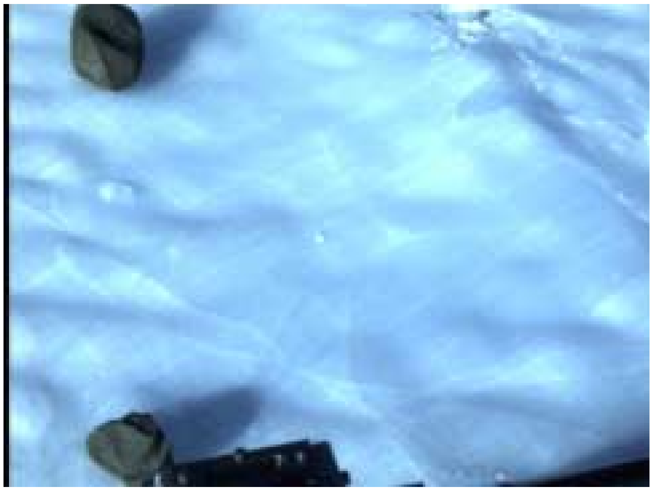


Figure 24. Two rocks on blue ice

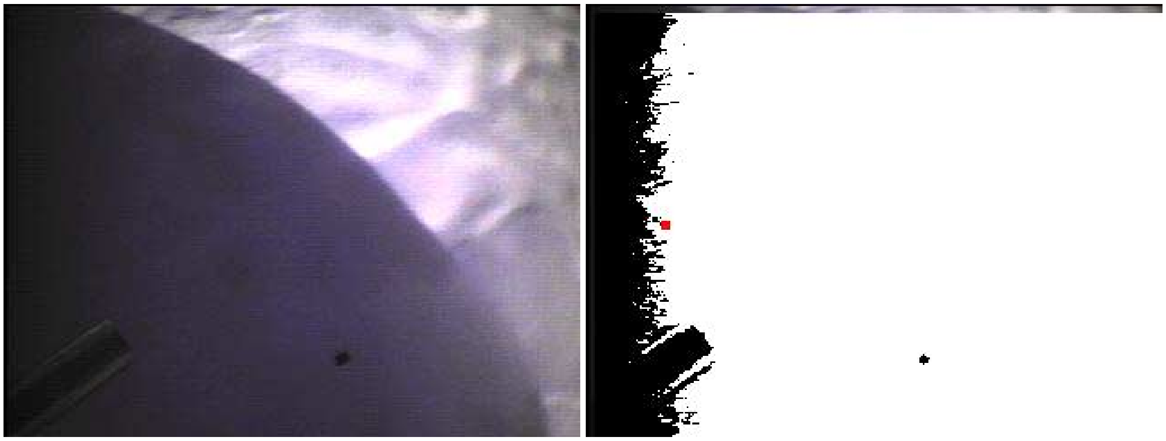
The top rock fits in a box of 83 x 87 pixels. This is a total size of 7221 pixels. The bottom rock fits in a box of 97 x 76 pixels. This is a total size of 7372 pixels. The variable *targetAcqTopMaxPix* was therefore set to 7200 pixels while *targetAcqBottomMaxPix* was set to 7300 pixels.

Altered arm control module to take 4 spectra of each rock. It pauses before taking each spectrum and puts them all into the database as separate readings of a single ID.

Pattern Test

Performed another search test in the same location as yesterday. The robot travelled in 200 m long rows. The pattern was started at approximately 10:20 am local time.

Visual servoing failed for the fifth target found (ID 4) failed because the rock was very small and shadowed. Figure 25 shows the initial step, where it failed.



Raw image

Segmented image

Figure 25. Images from visual servoing to ID #4

Note that visual servoing did not stop to ask for manual intervention because the shadows were segmented as a target. However, this shows how small a rock could be found.

After many false targets and 8 targets in one image, the pattern was paused and segmentation tested. A few images were grabbed in target acquisition test mode, and it was determined that the pattern should resume but in a new direction -- too many shadows were causing problems with the sun behind the robot. Also, the following parameters were changed:

- *targetAcqStddev* is now 10
- *targetAcqPercentage* is now 0.65
- *targetAcqStartRow* is now 5

After this was done, more false positives were found. The test was stopped, and another begun. Although the database information for this test is not available, it had very similar results, with way too many false positives. It too was stopped, and targetAcqPercentage was lowered to 0.60 to try to remedy the situation.

See [11] for all other statistics.

Traverse to the Meteorite Area

Moved the robot to another area, where a meteorite had found during a skidoo traverse.

This meteorite area is about 0.8 km from base camp.

Meteorite Demo #0

Once in the meteorite demo area, this demonstration, the first of the expedition, was held. In these demonstrations, there is no pausing to recompile code or change parameters in `SAS_Config`. There is minimal communication over VHF radios between camp and the people in the robot. The only acceptable communications are during wrist lowering and to verify false or missed targets after the robot has processed them. If the person with the robot determines that it may run over a meteorite, he/she will hit it's e-stop button, and use the joystick to drive the robot around the meteorite. The robot will be re-enabled and the pattern then resumed.

The demonstration consisted of 30 m rows running to the right. See [12] for all the statistics.

Conclusions:

- Segmentation seemed OK, there were no false targets.
- Maybe the robot shouldn't show out targets if high-res fails? (this was never changed)
- DGPS estimates look bad.
- The robot runs over 50% of rocks because tire tracks cover 50% of search area.

Debugging DGPS Estimates

From the last few tests' results, it seems that the target acquisition DGPS estimates are off. We checked the laser range data, and found that the ground appeared to be shifted up about 1 degree. This could mean that sensors and/or the sensor mast have tilted down (in the negative tilt direction). This could be causing the problems we've been seeing. So, the following variables were added to `SAS_Config`: *dgpsEstRelXFudge*, *dgpsRelYFudge* and *dgpsRelZFudge*. These fudge factors are added to the robot-relative frame when calculating estimated DGPS locations.

Experimenter's Notebook - January 21, 2000

Debugging DGPS Estimates

After thinking more about the DGPS fudge factors, they were changed from relative x, y, z distances to relative pan/tilt angles. So now the fudge factors used are *dgpsEstRelPanFudge* and *dgpsEstRelTiltFudge*. The pan fudge should be 0, and the tilt fudge is set to -2.0 degrees. These should be used instead because the errors in the system come from the sensors slipping.

Pattern Test

The goal of this task is to test the latest changes in a natural pattern. This pattern was performed near demo 0. If the robot approaches the meteorite in this region, we will also test the ability to pause the pattern, have the robot joysticked around the robot by someone in the field and the to resume the pattern.

It seemed that the database lost the file name of the last spectrum saved.

The spectrometer had to be calibrated on each arm deployment for this test because each deployment took 4 spectra.

See [13] for more information.

Conclusions:

- Target acquisition segmentation looked good.
- Target acquisition DGPS estimates looked good.
- Taking 4 spectra each time is really slow -- from now on just take 2.
- Too many targets in when re-acquiring targets before deploying the arm should not result in an error. This can occur because the arm itself could be in the image, but since re-acquisition selects the new target with a DGPS estimate closest to the initial estimate, it should never be selected.

Meteorite Demo #1

Demonstration conducted in the same area as demo #0. It was started at approximately 15:30 local time, with a 60 m x 40 m pattern, rows going to the right.

See [12] for all the statistics.

Conclusions:

- Arm camera segmentation shouldn't fail when too many targets are found. Most of the time, too many targets means shadows are present. If it were considered very conservatively, the robot could prompt for manual assistance in this case, but an operating assumption will be that most of the time the real rock will be found in all of those targets.
- A bug caused the second spectrum file name not to be placed in the database. Therefore the classifier had problems with it. Only 1 spectrum should be taken of each rock in a demonstration.
- Perhaps *armCamPixelSize* should be increased from 40 to keep smaller rocks?

Experimenter's Notebook - January 22, 2000

Pattern Test

Started this test at about 10:00 local time. It took place around the same location as meteorite demos #0 and #1. The sky was very clear.

Immediately after starting, 9 false positives were found. PTU position mismatch occurred, and `hiResSensor` had to be restarted. Next several real rock targets were acquired, only to have the camera miss them when deployed. Finally 8 more targets were found at once, with only 1 real rock there. The pattern was stopped, and the science autonomy system was taken down as the team evaluated what went wrong.

Two test runs were performed in which the robot is driven along a straight line with target acquisition running. When the robot drove in the same direction in which the pattern had been executed, shadows were common (the sun was behind the robot) and often caused segmentation to fail.

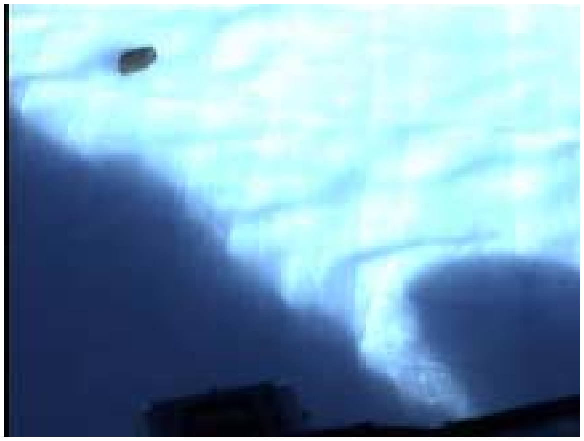


Figure 26. Rocks on blue ice with sun behind Nomad

In Figure 26, the robot's shadow is cast into the area of target acquisition. This causes the shadow compensation routines to run, which causes darker pixels to be ignored. In this case, the rock in the upper left corner of the image was ignored. Furthermore, so many shadows cause the auto-iris to open up, making ice look even brighter that usual.

When the robot was turned so the sun was shining on the side of the it, segmentation worked more effectively. However, even in this case, once three targets were all assigned to one rock.

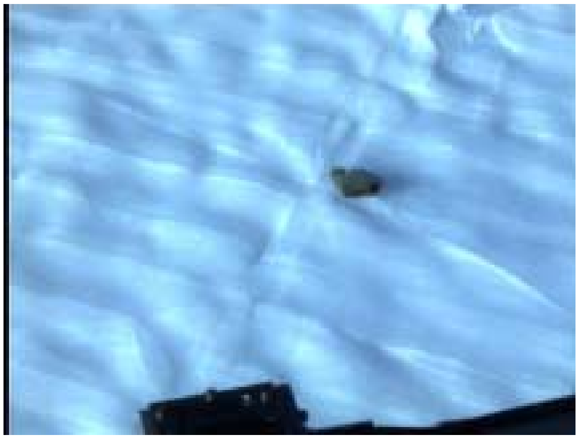


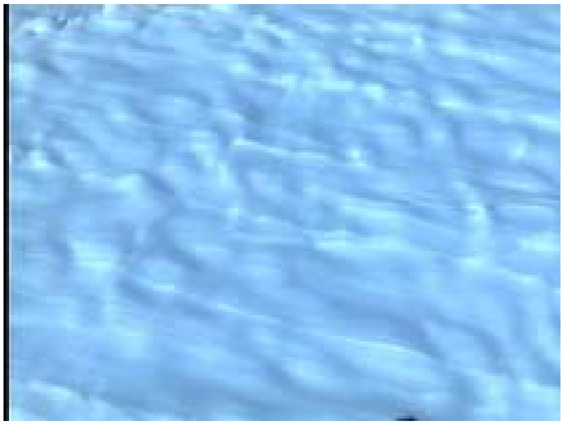
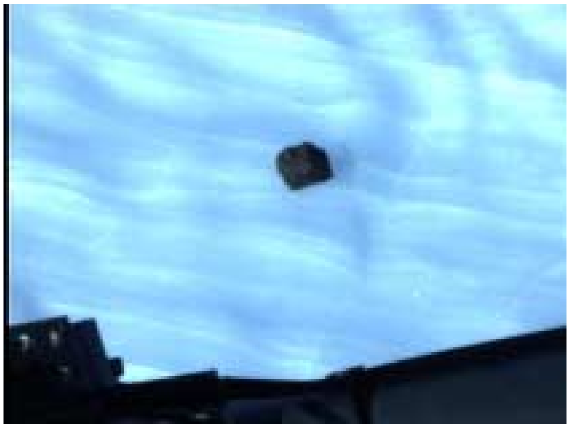
Figure 27. Rock on blue ice without shadows

In Figure 27, notice that now the shadows are not seen, and the ice looks a bit less overexposed. However, the rock in this picture was segmented into three distinct targets.

The conclusions from this test were that the pattern should be performed in this new direction perpendicular to the sun's rays, and that it would be good to add code that would combine multiple targets whose DGPS estimates are very close together.

After this, a new pattern was begun. Before the first spectrometer calibration, the arm was not properly stowed, so when it moved into calibration position, it was misaligned. In another case

(ID 3), the high-resolution camera servoing seemed to be working well, but in the last step, the arm ended up in the image. The visual servoing steps are shown in Figure 28. Notice that in the first image, the rock is just barely visible at the bottom -- this is probably due to a very bad DGPS estimate.

Step	**Image**
Step 1	 The rock is barely visible at the bottom
Step 2	 Now it is well aligned

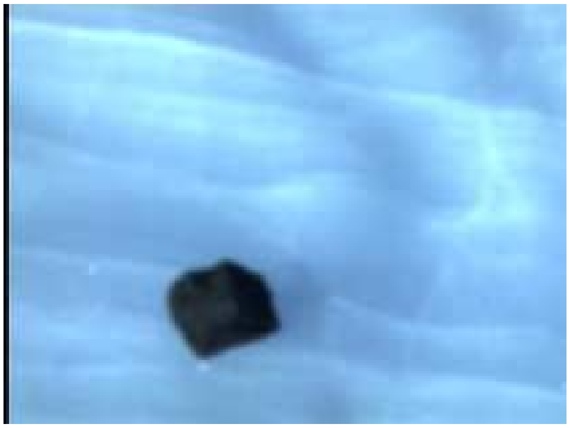
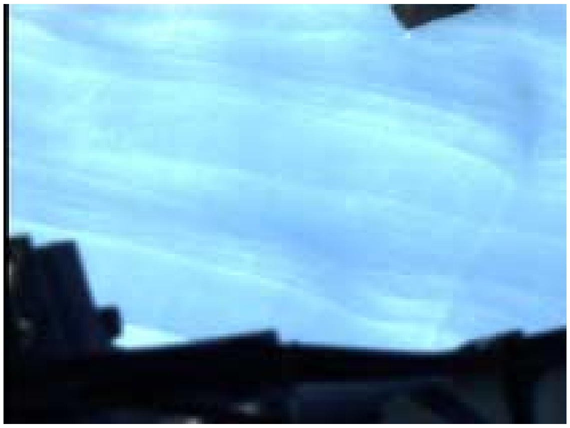
Step	Image
Step 3	 Here it is still reasonably well aligned, considering there is a visually noticeable amount of play in the PTU
Step 4	 Although it wasn't even in the last image, the arm gets chosen as the final target

Figure 28. Images from servoing the high-resolution camera to ID #3

See [14] for the statistics.

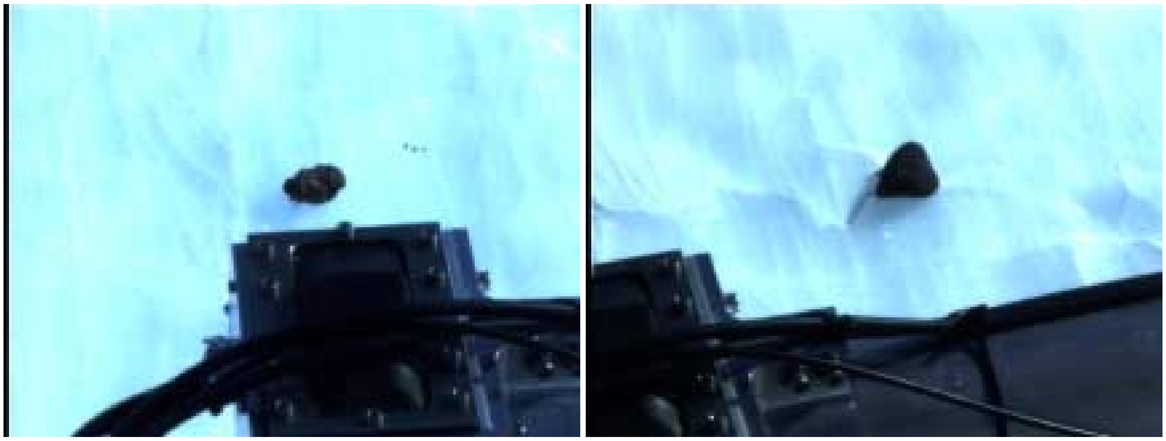
Conclusions:

- The camera lens still not working reliably for some unknown reason
- The arm still showing up in final high-resolution images
- Two rocks were completely missed by target acquisition
- Repeatedly analyzed a one rock 3 times
- Segmentation seemed good in all respects in the new direction - even arm camera worked except for a white rock
- Should definitely merge targets with close DGPS estimates into a single target

Meteorite Demo #2

This demo was again in the same location. Rows were in a 30 m x 20 m rectangle going to the right. The demo was started approximately at 13:45 local time.

For IDs 4 and 6, too many targets were found when deploying the high-res camera, so these targets were subsequently ignored. However, there is clearly only one rock in each image. The images are shown in Figure 29.



ID #4 high-res deployment image

ID #6 high-res deployment image

Figure 29. High-resolution camera images of target IDs #4 and #6

Most likely, the arm being so prominent in these photos caused many targets to be segmented. This could be due to the arm being segmented as many smaller targets due to its coloring, or because the arm is so prominent that it causes shadow compensation to occur.

ID 14 was the first meteorite autonomously found by Nomad. It's 22% probability of being extraterrestrial is significantly higher than the other samples classified. Table 5 shows the probabilities returned.

Target ID	Rock Description	P(Extraterrestrial) [%]
0	Basalt Fragment	4.68
3	Basalt Fragment (repeat)	8.57
5	Basalt	9.33
8	Basalt (repeat)	4.85
9	Tan Silkstone	0.19
12	Dolerite	1.51
13	Gray to Black Shale ("meteorwrong")	9.26

Table 5. Classifier results for meteorite demo #2

Target ID	Rock Description	P(Extraterrestrial) [%]
14	Meteorite 12067	21.89

Table 5. Classifier results for meteorite demo #2

The meteorite was given the field number 12067 by John Schutt. Including the meteorite itself, the mean extraterrestrial probability is 8.61%, σ = 6.75%. Therefore the meteorite's probability was 1.97 σ from the mean.

See [12] for all the statistics.

Experimenter's Notebook - January 23, 2000

Collecting Meteorite Spectra

The robot was left in front of meteorite 12067, and was commanded to take many spectra of it. This was not a search, simply visual servoing, storing the spectra and classification based on these spectra.

During these tests very frequent “SCA1” stop code errors were observed. This error in the arm motion control hardware means “independent move finished”. This may not be bad, but the software thought it was. The test had to be cut short and spectra taken manually afterwards.

See [15] for statistics.

Conclusions:

- SCA1 may not be bad -- added code to prompt user to a success status even after an 'illegal' stop code.
- Not sure where SCA1's are coming from, but the arm joint encoders should be tested just in case.
- Classifier classified the same meteorite with probabilities of 9.7% and then 87% P(extraterrestrial). Not sure why this is happening.

After this the robot drove back to camp.

Arm Encoder Tests

Added the ability to query encoder position in `ArmTestBench`. Table 6 gives the results of an investigation of the arm joint encoders.

Joint / Absolute Angle [deg]	Potentiometer Voltage [V]	Potentiometer Angle [deg]	Encoder Angle [deg]
Joint 1			
180	6.06	179.66	-0.57
160	4.82	160.11	-20.19
140	3.68	140.26	-40.1
120	2.57	120.1	-60.23
100	1.43	100.47	-80.05
90	0.85	90.11	-90.3
Joint 2			
80	0.28	80.17	-100.45
60	-0.86	60.21	-120.77

Table 6. Results of arm joint encoder tests

Joint / Absolute Angle [deg]	**Potentiometer Voltage [V]**	**Potentiometer Angle [deg]**	**Encoder Angle [deg]**
40	-1.99	39.7	-140.92

Table 6. Results of arm joint encoder tests

Note that the sense of joint 2 is opposite that of joint 1, therefore putting joint 2 at 80 degrees in the real world results in the joint 2 encoder reading 180 - 80 degrees = 100 degrees.

Finally tested visual servoing was tested by going rocks to the following estimated locations to see if we could get the SCA1 problem. See Table 7 for explanation.

Arm Relative X Location [m[	**Arm Relative Y Location [m]**	**Results**
0	1	Success
-0.1	1.3	Success
-0.3	1.5	Failed - target out of reach

Table 7. Results of an arm servoing test

No SCA1 errors were seen. Therefore, there seems to be no systematic error, the encoders just could be flaky.

Experimenter's Notebook - January 24, 2000

The windchill was -42 F today. The sky was overcast, and snow was blowing along the ground. To record these lighting conditions, several target acquisition images were taken. Figure 30 shows a few of them. Notice how different they look from images taken on a bright day. The light is very flat and the rocks are more distinct. Incidentally, this type of lighting condition is considered best for human meteorite searches.

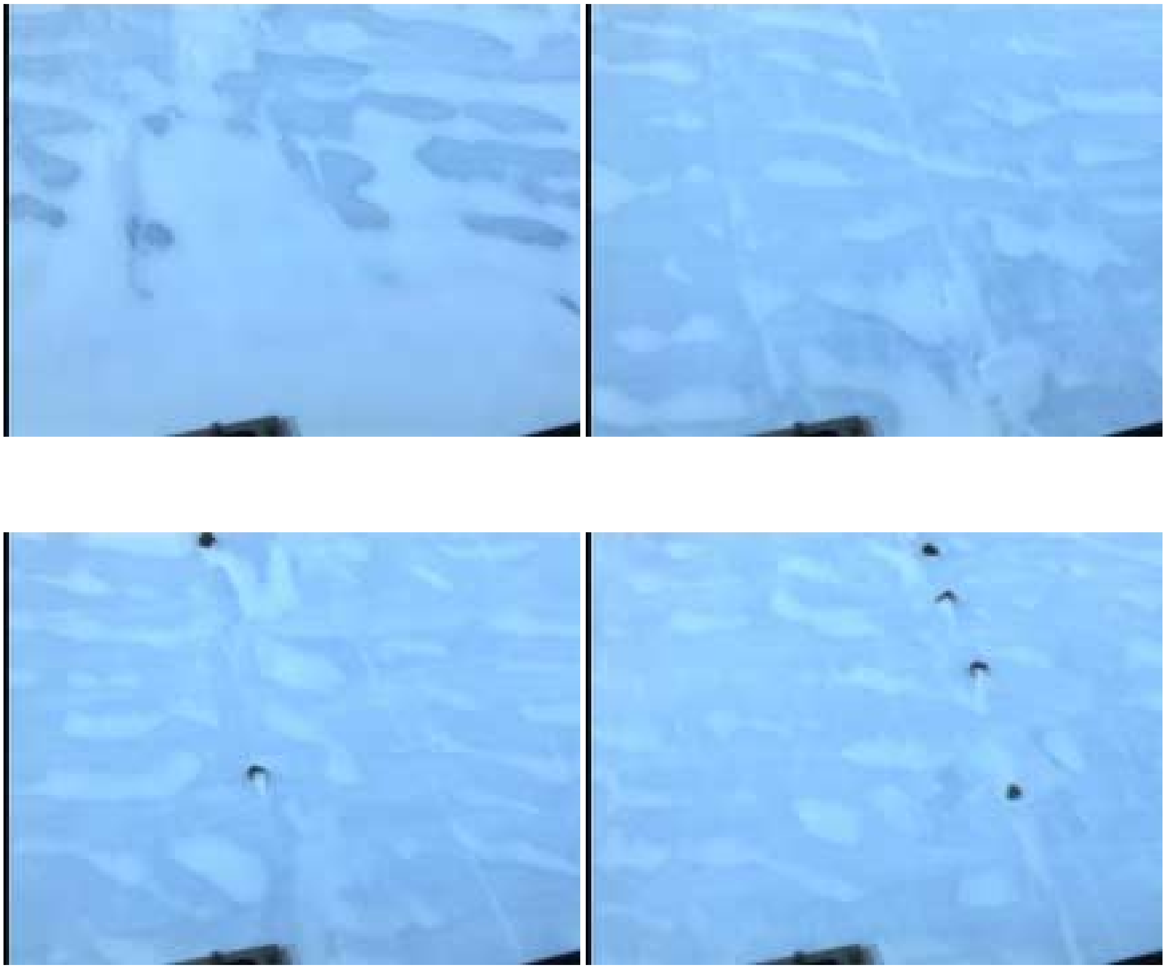


Figure 30. Target acquisition images of rocks taken during overcast conditions

Notice that the ice looks a deeper blue than during normal lighting conditions. Also, there are no shadows because the light is extremely diffuse. The blowing snow also creates snow trails on the leeward side of the rocks, and fills ripples of blue ice with snow patches.

Collecting Meteorite Spectra

Meteorite 12067 had been taken back to camp, so in the bad weather it was decided to keep the robot nearby and take more data both of it and other samples we had. These other samples included a kulnine meteorite from the collection of Bill Cassidy and several meteorwrongs. See

[16] for more information.

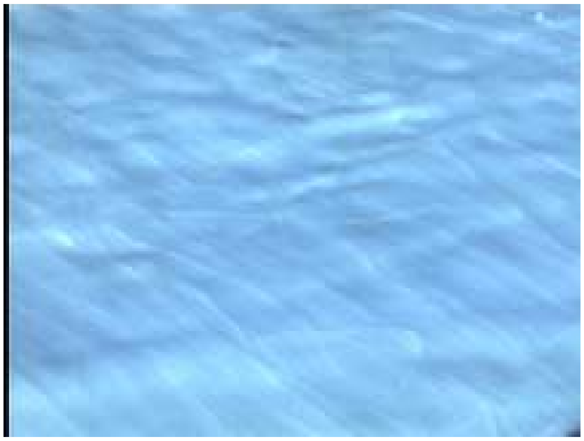
Afterwards, the robot drove to the second meteorite site, which was approximately 0.7 km to the north northwest of the base camp. At this site, 3 meteorites had been found previously. Along with one other rock, these were the only samples in the area. Such an extremely low rock density is typical for blue ice.

Experimenter's Notebook - January 25, 2000

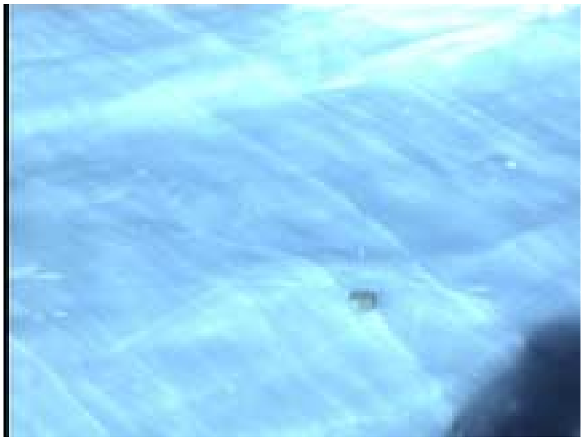
Pattern Test

A pattern with 20 m long rows was conducted near the second meteorite area. This time, a high-res camera zoom encoder target value of 100 was used. The zoom encoder value is represented by a one byte number, with around 235 being zoomed out fully, and near 30 being zoomed in fully. Previously in this expedition it had been 140, which meant that the system was now attempting to zoom in further. Doing this in Willy Field had caused the targets to get lost for an unknown reason. The robot would bring the zoom value down in steps of 60. However, zoom values near 30 had worked in Pittsburgh on the slag heaps.

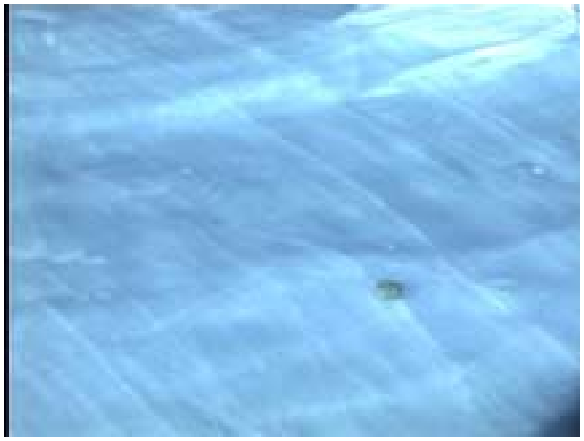
There were some problems with throwing out real rock targets. In particular, IDs 16 through 19 were troublesome. Figure 30 shows the images that caused the high-res camera software to fail for each of these. IDs 17 and 18 were known as “Rock E” which was a very white rock that was easily missed by the target acquisition segmentation method.



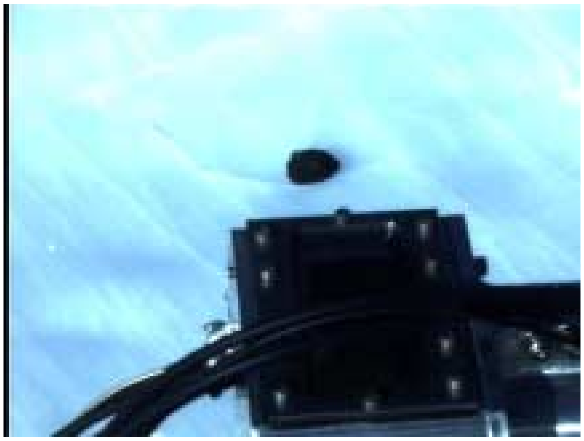
ID #16 (rock D) -- found no targets



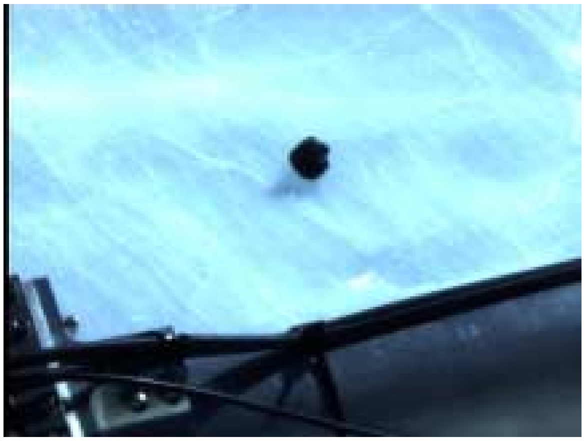
ID #17 (rock E) -- found no targets



ID #18 (rock E) -- found no targets



ID #19 -- found too many targets



ID #20 -- found too many targets

Figure 30. High-resolution camera images that resulted in failures

See [17] for statistics.

Meteorite Demo #3

This demo was held at the second meteorite location. It consisted of 55 m long rows going to the right. It was started at approximately 18:50 local time.

This blue ice field area had an extremely low rock density. There were 3 meteorites in the area, given field numbers 12331, 12097 and 12372. There were also one or two terrestrial rocks that were brought in from the moraine by John Schutt. This type of area is characteristic of many blue ice fields -- when humans search these areas, they examine just about every rock they see closely. There are so few of them, and chances are if you find one, it's most likely a meteorite.

Only meteorite 12331 was seen in this demonstration, and it was classified twice. The first time it was found, it was given a P(extraterrestrial) of 19%, and on the second time, 35%.

At approximately 20:00, the first pattern was completed, so a new pattern was started at that spot. It consisted of 55 m rows to the right.

Please see [12] for all the statistics.

Experimenter's Notebook - January 26, 2000

This day consisted of four demos in the second meteorite area: demos #4 - #7. The samples in this area were very sparse.

See [12] for the statistics on these demonstrations.

Meteorite Demo #4

This consisted of a pattern with 40 m rows going to the left. It started approximately at 11:00 local time. The magnetic north offset was set to 2.73 radians in Nomad's real-time software. Finally, the spectrometer calibration target was a bit off, and was realigned. A new "ideal" reference spectrum was taken in its new location.

Halfway through the demo much of the science autonomy system had to be restarted (target acquisition, mission planner), so a new pattern was started just before target ID 6 was found. This pattern had 20 m rows going to the left.

The first demonstration of the day was rather unsuccessful. Here were the conclusions:

- Visual servoing failed 3 times out of 3 attempts.
- 2 failures came about because of an IK error. These were probably caused by an inconsistency between the arm's physical workspace and the mission planner's workspace model. This is an unusual error though, and it is unlikely that it would have occurred twice in one demonstration, but it did with IDs 3 and 7. This was the same rock (meteorite 12097), but it was approached from different directions.
- The other failure, which occurred on the second attempt on ID 3, occurred because the steps of the visual servoing process brought the rock underneath the spectrometer in the arm camera image. This occlusion caused the servoing software to lose the rock on the second visual servoing step. Apparently, it did not cause the centroid of the spectrometer to shift out of the "ignored" zone.

Conclusions:

- Change arm controller software to return an OUT_OF_WORKSPACE error on an IK failure so the mission planner will automatically re-maneuver and re-deploy. Currently it just redeploys without necessarily maneuvering, and if the rock is really out of reach, it will still be out of reach on the second attempt.
- Change the mission planner's arm workspace right angle from 6.05 to 6.2 radians. This will make the mission planner's concept of the arm workspace smaller so less out of workspace errors should occur. However, will this smaller area make maneuvers too difficult? This has happened on the slag heaps before.

Meteorite Demo #5

This pattern consisted of 20 m rows going to the right. It started at approximately 13:45 local time.

The maneuver planner had problems maneuvering to within the arm's new, smaller workspace of a target. Therefore, the only conclusion was to take the right angle back to 6.05 radians.

Meteorite Demo #6

This pattern consisted of 20 m rows to the left, and started at approximately 14:45 local time. Meteorite 12097 was misclassified with a P(extraterrestrial) of 1.4%, so the demo was aborted and several additional spectra were taken of it (IDs 1 to 8). See [18] for the statistics.

Meteorite Demo #7

This pattern consisted of 30 m rows going to the right, and was started at approximately 15:45 local time.

Meteorite 12372 was found in this demonstration. It was given a 27% probability of being a meteorite.

After Nomad passed meteorite 12372, it studied a dolerite. This was the last sample in the area, so the pattern was finished as a success. After this, three more spectra were taken of meteorite 12372.

Experimenter's Notebook - January 27, 2000

Traverse

Before the day's operations, Nomad drove into the moraine for its final demonstrations. The destination area of the moraine is over 1.1 km away, almost due west of camp. The moraine is mostly sastrugi and packed snow rather than blue ice. The rocks are much denser than the blue ice as well. Finally, unlike the blue ice, the rocks here are much more likely to be terrestrial. Figure 31 shows a panoramic image taken along the trip.

Figure 31. Panoramic image taken during traverse into the moraine

Rock and Meteorite Spectra Collection

When this test began, the robot failed to servo the arm to 5 targets. After about an hour of debugging, the problem (a simple bug in the code) was found and corrected. Once this was completed, the test was restarted. 19 spectra were collected of the following four numbered samples, shown in Table 8.

Sample Field Number	Description from John Schutt
12387	Ordinary chondrite meteorite
12322	Probably an achondrite
12346	Probably just a strange looking rock
12341	Probably just a strange looking rock

Table 8. Description of collected spectra sources

Visual servoing in general did quite well, occasionally failing only because of questionable segmentation. There were no cases of failure after 2 attempts. Three attempts required manual intervention when no rocks were found. Figure 32 shows one case where segmentation failed, on ID 3. Notice how washed out the image is.

Figure 33 shows some arm camera images of the meteorites studied in this activity.

See [19] for statistics.

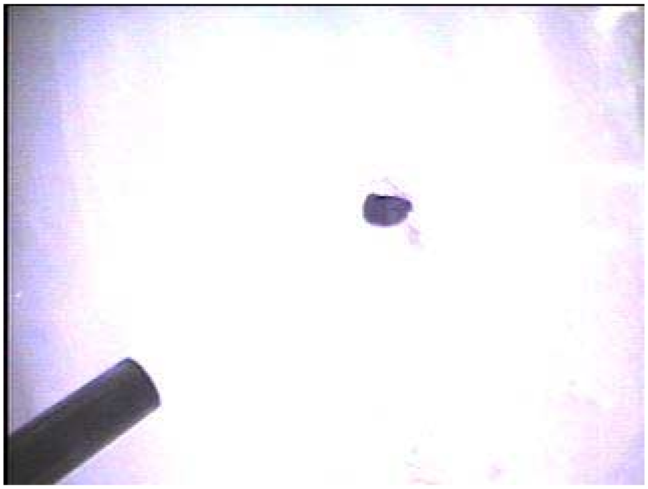


Figure 32. Arm camera image from failed visual servoing attempt

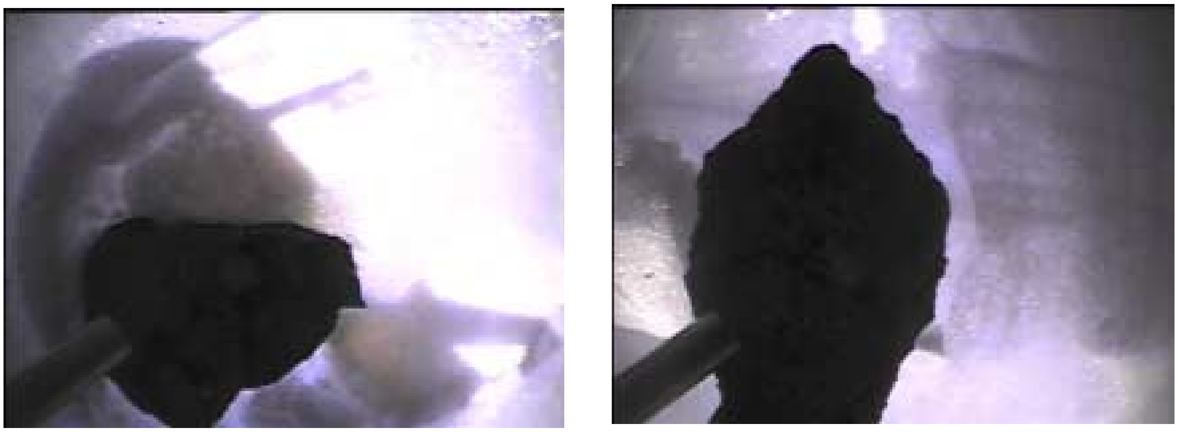


Figure 33. Arm camera images from left: meteorite 12387, meteorite 12322

Experimenter's Notebook - January 28, 2000

It snowed overnight. The fresh snow was about 1 inch deep. The sky went from overcast to partially sunny as the day went on.

Meteorite Demo #8

The demo took place in the moraine. The pattern's rows were 200 m long and proceeded to the left. Many rocks were partially covered with freshly fallen snow. Additionally, in the moraine the size of rocks varies dramatically, from 1 to 50 cm. Some larger rocks that were snow covered appeared as a few unconnected rock regions, so Nomad reasonably considered them multiple targets (see Figure 34).

While there were no meteorites found (John Schutt believed there were probably none in the area), this was the most successful demonstration. All the robot's systems: search, servoing, etc., worked very well and the robot studied 15 unique samples. These samples were later collected to from the "Moraine Rock Set" (see Appendix D). This demonstration provided a glimpse of some terrain unlike what Nomad had previously seen. The density of rocks was much higher (around 1 / m^2), and possibly the distribution of rock types is different. Figure 35 shows the different types of rocks seen by the high-res camera. These rocks range from light gray to black, and in size from dozens to a few centimeters in diameter. Figure 36 shows four of the samples that were seen by the arm camera. Note how much detail this view provides, especially with respect to rock color (notice the red basalt color of the upper right image) and texture.

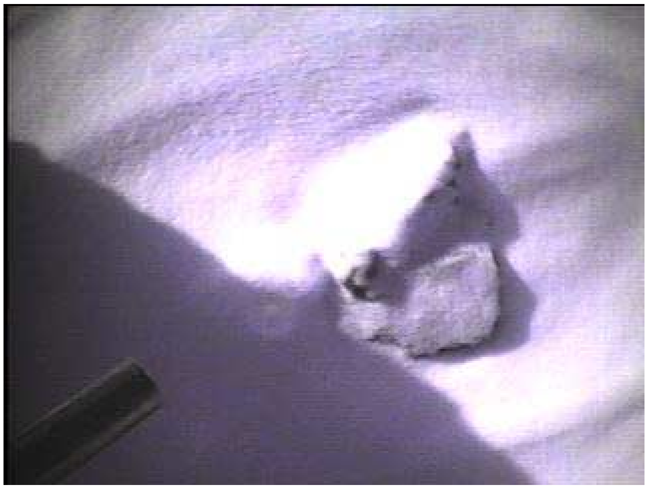


Figure 34. Snow covered rock in the moraine, as seen by Nomad's arm camera

See [12] or all the statistics of this demonstration.

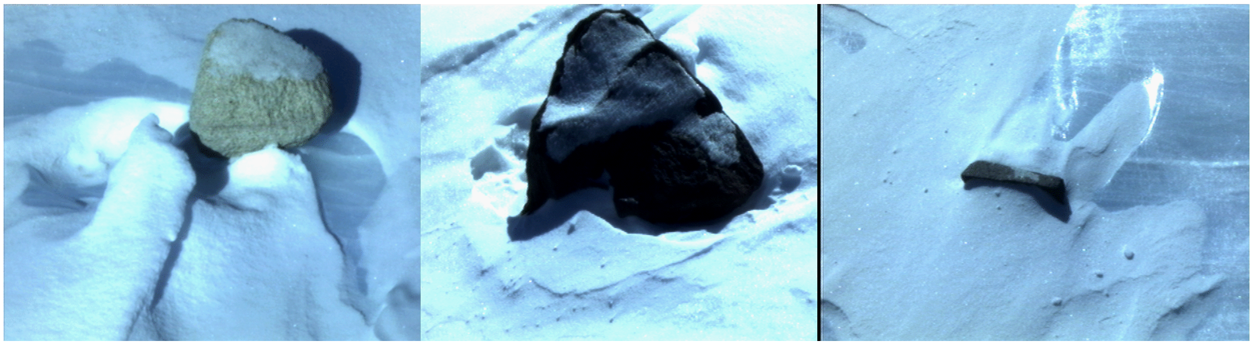


Figure 35. High-res images of various rocks in the moraine. Note the differing color, size and shape.

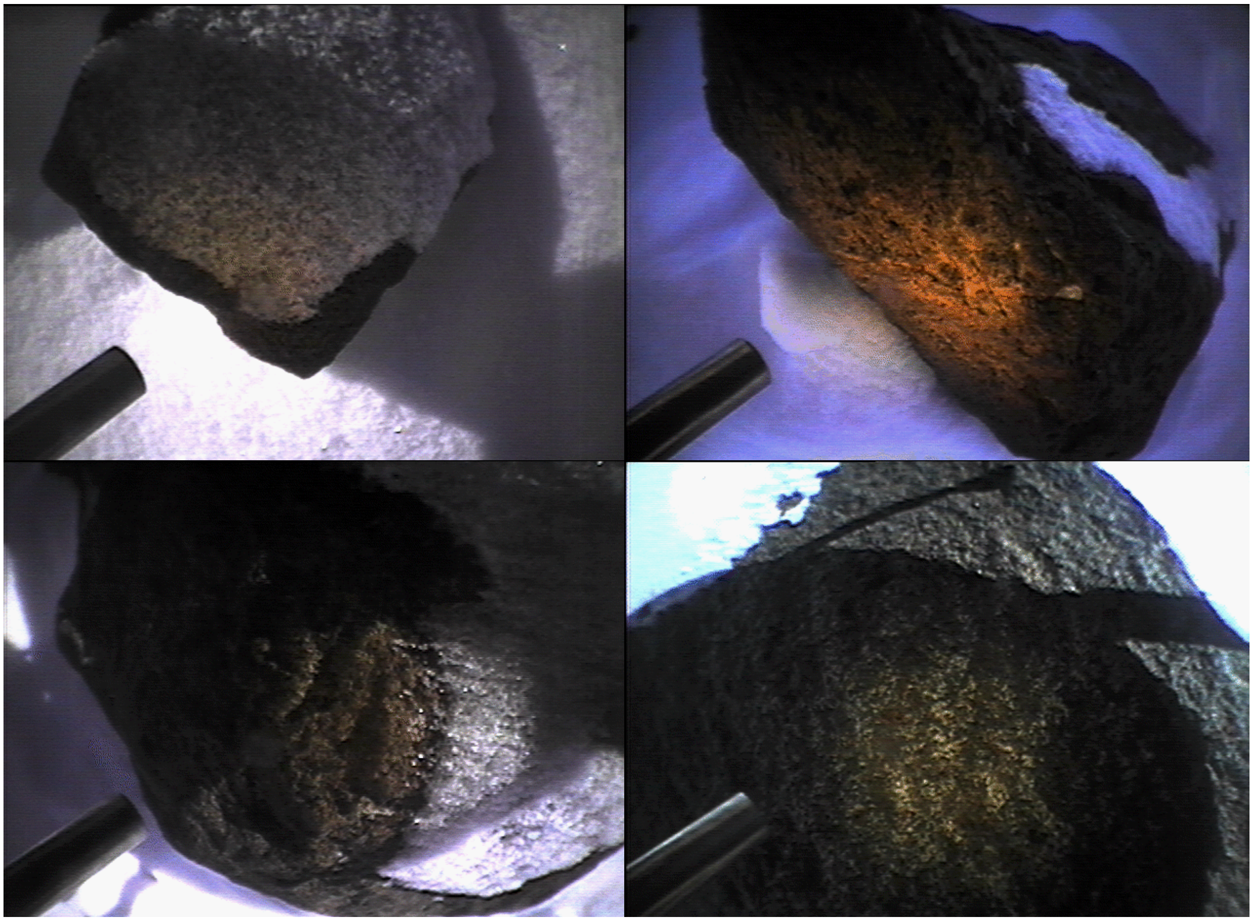


Figure 36. Arm camera images of various rocks in the moraine. Note how well color and texture are shown.

Experimenter's Notebook - January 29, 2000

Meteorite Demo #9

Today Nomad performed another demonstration in the moraine, adjacent to where meteorite demo 8 was held. The rocks were similar to those in demo #8 as well, both in size and because they were covered with snow. Overall, the system did well even though the first three rocks were thrown out by the high-res camera -- probably due to the snow. 10 unique targets were found, although only 6 samples were studied with the spectrometer due to problems visual servoing. In fact, only 7 out of 15 visual servoing attempts were successful. One important conclusion from this demonstration is that visual servoing doesn't work well when there are a few rocks clustered near each other.

Figures 37 and 38 show high-res and arm camera images of the rocks seen in this demonstration. The conditions were similar to those seen in demonstration #8.

See [12] for all the statistics.

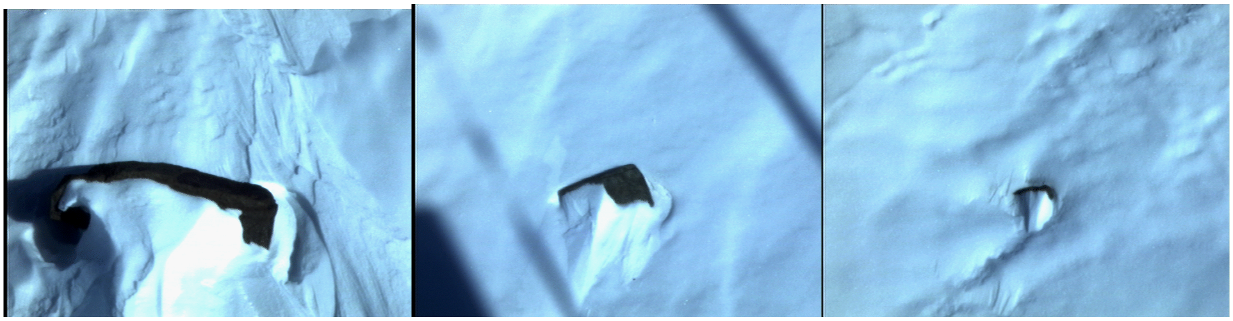


Figure 37. High-res camera images of three rocks seen in demonstration #9. Note the wide range of sizes: from almost 30 cm wide (left image) to around 5 cm wide (far right image).

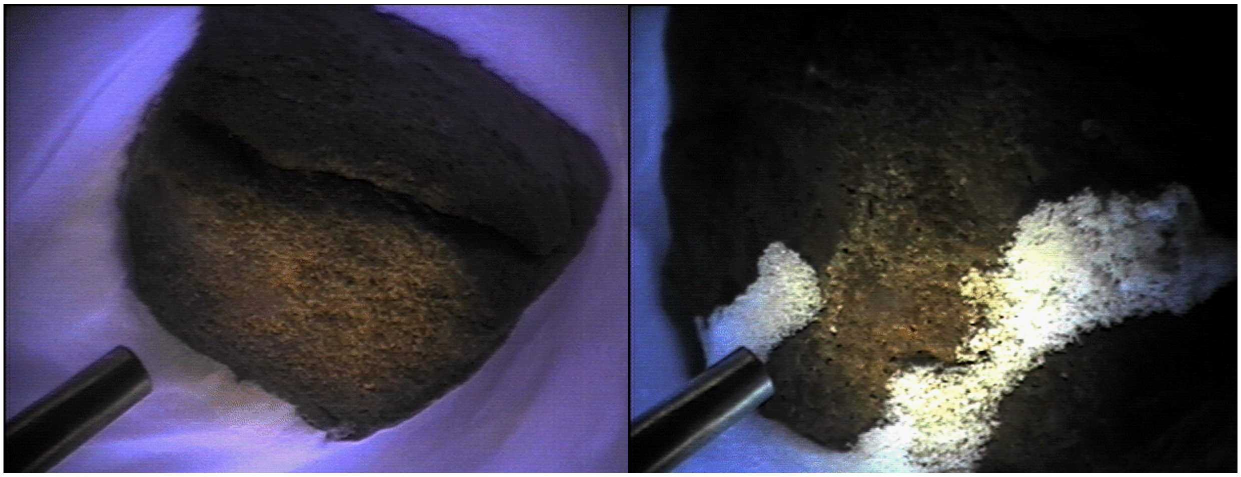


Figure 38. Arm camera images of two rocks seen in demonstration #9. The image on the right is partially covered in freshly fallen snow.

Experimenter's Notebook - January 30, 2000

Meteorite Demo #10

The last meteorite demonstration was attempted today. It consisted of a 400 m long row, and was started at approximately 13:00 local time. It had to be aborted within 30 minutes, however, because the hard drive failure on `meteorite` precluded it from properly storing any sensor data. Being that this was the last day of field operations, the problem was not debugged.

After this failure the robot drove back to the camp to finish operations.

Collecting Rock Spectra

To take advantage of the rest of the day, the team brought back some moraine rocks and took spectra of them manually with the robot's sensor wrist.

See [20] for the file names.

Appendix A

The list below contains links to documents referenced in this text.

[1] MATLAB diary during high-resolution calibration procedure: http://www.ri.cmu.edu/~meteorobot2000/results/01_16/calib_diary.html

[2] January 16 classification test results: http://www.ri.cmu.edu/~meteorobot2000/results/01_16/Classification_Test_Results.html

[3] Ideal white reference spectrum: http://www.ri.cmu.edu/~meteorobot2000/results/01_16/Ideal_White_Spectrum.html

[4] January 16 first pattern test results: http://www.ri.cmu.edu/~meteorobot2000/results/01_16/Pattern_Test_Results.html

[5] January 16 calibration parameters: http://www.ri.cmu.edu/~meteorobot200/results/01_16/calib_jan16.html

[6] January 16 second pattern test results: http://www.ri.cmu.edu/~meteorobot2000/results/01_16/Pattern_Test_Results_2.html

[7] January 17 surveyed line test results: http://www.ri.cmu.edu/~meteorobot2000/results/01_17/Line_Test_Results.html

[8] January 18 pattern test results: http://www.ri.cmu.edu/~meteorobot2000/results/01_18/Pattern_Test_Results.html

[9] January 19 surveyed line test results: http://www.ri.cmu.edu/~meteorobot2000/results/01_19/Line_Test_Results.html

[10] January 19 pattern test results: http://www.ri.cmu.edu/~meteorobot2000/results/01_19/Pattern_Test_Results.html

[11] January 20 pattern test results: http://www.ri.cmu.edu/~meteorobot2000/results/01_20/Pattern_Test_Results.html

[12] Expedition demonstration results: http://www.ri.cmu.edu/~meteorobot2000/results/Demo_Results.html

[13] January 21 pattern test results: http://www.ri.cmu.edu/~meteorobot2000/results/01_21/Pattern_Test_Results.html

[14] January 22 pattern test results: http://www.ri.cmu.edu/~meteorobot2000/results/01_22/Pattern_Test_Results.html

[15] January 23 pattern test results: http://www.ri.cmu.edu/~meteorobot2000/results/01_23/Pattern_Test_Results.html

[16] January 24 spectrum collection: http://www.ri.cmu.edu/~meteorobot2000/results/01_24/Spectrum_Collection.html

[17] January 25 pattern test results: http://www.ri.cmu.edu/~meteorobot2000/results/01_25/Pattern_Test_Results.html

[18] Demo #6 additional results: http://www.ri.cmu.edu/~meteorobot2000/results/01_26/

Demo_6_Additional_Results.html

[19] January 27 spectrum collection: http://www.ri.cmu.edu/~meteorobot2000/results/01_27/Spectrum_Collection.html

[20] January 30 spectrum collection: http://www.ri.cmu.edu/~meteorobot2000/results/01_30/Spectrum_Collection.html

Appendix B

The list describes all science autonomy programs referenced in the text.

Program Name	Usage	Description
ArmTestBench	`ArmTestBench`	Arm teleoperation and debugging command interface.
SAS_Config	`SAS_Config`	Global SAS parameter editor.
armCamImageScanner	`armCamImageScanner [-h] [boxSize pixSize]`	Grab an image from the arm camera and segment it. The original image will be saved to `/tmp/initial.ppm` and the segmented image will be in `/tmp/segmented.ppm`. Specify particular box sizes and pixel sizes to affect how segmentation works.
dbClient	`dbClient`	Database client that communicates with dbServer via NDDS.
dbServer	`dbServer [-v]`	Database server that stores info about science targets. Use the -v option to run in verbose mode.
getCalibImages	`getCalibImages`	Grab a sequence of images across various pan and tilt angles (for the sake of calibration). File names will specify each image's pan and tilt values.

Table 9. Science autonomy programs.

Program Name	Usage	Description
grabImage	`grabImage [-h] [source pan tilt zoomEnc focusEnc outFile]`	Grab a single image a specified pan and tilt angle as well as zoom and focus encoder values. Parameters can either be entered interactively (if no arguments are specified), or via the command line as follows: • source: 1 for high-res camera, 2 for arm camera, 3 for panoramic camera. • pan: high-res camera pan (deg), must be specified even if source is not high-res camera. • tilt: high-res camera tilt (deg), must be specified even if source is not high-res camera. • zoomEnc: high-res lens zoom encoder value, must be specified even if source is not high-res camera. 255 means zoom out fully, 0 means zoom in fully. • focusEnc: high-res lens focus encoder value, must be specified even if source is not high-res camera. The range of values is from 0 to 255. • outFile: the file that will hold the resulting image.
hiResSensor	`hiResSensor driverName`	The high-res camera sensor driver.

Table 9. Science autonomy programs.

Program Name	Usage	Description
javaToNdds	`javaToNdds nddsDomain ipAddress port`	A program to convert commands from a Java program to NDDS. ipAddress and port must be those of the machines running the program.
nddsToJava	`nddsToJava nddsDomain ipAddress port`	Program to convert NDDS messages into Java objects. ipAddress and port must be those of the machines running the program.
segCalibImages	`segCalibImages`	Program that segments a range of images gathered by getCalibImages, using the red thresholding method.
segment	`segment in.ppm out.ppm`	Segments an image using the default segmentation method
sensManClient	`sensManClient`	Client to the sensor manager, displays a text interface allowing the user to send any type of message to the sensor manager, and listens to the sensManReply and sensManComplete messages.
sensManMain	`sensManMain`	The sensor manager.
spectrometer	`spectrometer`	The spectrometer Tcl/Tk teleoperation interface (not recommended for use over low-bandwidth networks!)
targetAcqMain	`targetAcqMain [-test]`	The target acquisition manager. Specify -test to save images while doing target acquisition (NOTE: this will considerably slow down target acquisition, so do not run it in test mode when doing an autonomous search!)

Table 9. Science autonomy programs.

Program Name	Usage	Description
targetAcqMsgs_client	`targetAcqMsgs_client domain cmd`	A client for the target acquisition manager. Specify domain for the NDDS domain used, and cmd for the command to send to the target acquisition manager (0 to start, 1 to stop).
telemetryLogger	`telemetryLogger`	The SAS telemetry logger.
transportStow	`transportStow`	Servos the PTU to a position (90 deg clockwise) that will fit inside the trailer.
pxc_ui	`pxc_ui serverIP serverPort [imageSavePath]`	A video client that received images from videoServer. Run it, you must specify the IP address and port of the machine running videoServer. If you'd like to save images, you can specify a directory in which they'll be saved with imageSavePath. Otherwise, a default will be used.
videoServer	`videoServer port`	A video server that sends images from a PXC/200 framegrabber across a UNIX socket.

Table 9. Science autonomy programs.

Appendix C

This appendix lists the configuration variables used throughout the science autonomy system. Many of them are referenced in *italics* throughout this document. These variables are altered either before or during runtime with the program `SAS_Config`. They are saved in a text file on the robot. The following table lists each variable's name, description and value at the end of the expedition. As shown in the text, often these values were found after days of fine tuning. Ideally they all should be calibrated in a principled fashion, however no such calibration procedure yet exists. A special note should be given to the variables defining sensor deployment costs (such as *manipulatorTimeCost*). These were defined simply as constant unitless values. There was no attempt to take variable sensor costs into account - they were chosen rather arbitrarily to simply determine sensor deployment order (i.e. the arm's costs were ten times higher than the camera's, so the camera was always deployed first). A great deal more work should be done in this area. Other variables did exist in the system, but for the sake of clarity only those implemented are listed.

Variable Name	**Description**	**Final Value**
ptuSerialPort	The Linux serial port number on `meteorite` that the pan / tilt unit controller used.	0
ptuMinTilt	A software-based minimum tilt limit for the pan / tilt unit. The hardware limit should be lower than this. The units are encoder positions.	-3500
lensControllerSerialPort	The Linux serial port number on `meteorite` that the lens focus / zoom controller used.	1
NDDS_Domain	The NDDS domain number used by the science autonomy, navigation and real-time software systems.	0
useAntarcticSegmentation	This value is generally set to 1. If 0, the segmentation software is in debug mode, whereby it segments red rather than dark pixels.	1
telemetryLogPeriod	The period (in seconds) for which the `telemetryLogger` program saves telemetry to disk.	2.0

Table 10. Science autonomy system configuration variables

Variable Name	Description	Final Value
targetAcqCamPan	The nominal high-res camera pan angle (in degrees) in target acquisition mode.	-14.5
targetAcqCamTilt	The nominal high-res camera tilt angle (in degrees) in target acquisition mode.	-37.5
targetAcqStartRow	The image pixel row at which the high-res target acquisition driver begins segmentation. Any pixels in row numbers less than this (i.e. higher in the image) will be ignored. This helps define the image area in which new targets are found.	5
targetAcqStopRow	The image pixel row at which the high-res target acquisition driver ends segmentation. Any pixels in row numbers greater than this (i.e. lower in the image) will be ignored. This helps define the image area in which new targets are found.	240
targetAcqMinCol	The image pixel column at which the high-res target acquisition driver begins segmentation. Any pixels in column numbers less than this (i.e. further left in the image) will be ignored. This is needed because the first several columns in an image are often black.	10
targetAcqMaxCol	The image pixel column at which the high-res target acquisition driver ends segmentation. Any pixels in column numbers greather than this (i.e. further right in the image) will be ignored. This is needed because the last several columns in an image are often black.	630

Table 10. Science autonomy system configuration variables

Variable Name	Description	Final Value
targetAcqTopBoxSize	Segmentation parameter that defines the maximum distance between "adjacent" pixels at the top of a high-res image. The "Box" settings allow a cluster of rock pixels to be lumped together as one region. The box size at the top of a high-res image is smaller because pixels at the top represent patches of ground more distant than those at the bottom.	20
targetAcqBottomBoxSize	Segmentation parameter that defines the maximum distance between "adjacent" pixels at the bottom of a high-res image. The "Box" settings allow a cluster of rock pixels to be lumped together as one region. The box size at the bottom of a high-res image is greater because pixels at the bottom represent patches of ground less distant than those at the bottom.	40
targetAcqTopPixelSize	Segmentation parameter that defines the area of the patch of ground represented by one pixel at the top of a high-res image. This is not an absolute area measurement. Instead, the ratio between the pixel sizes at different rows in the image is important.	3
targetAcqBottomPixelSize	Segmentation parameter that defines the area of the patch of ground represented by one pixel at the bottom of a high-res image. This is not an absolute area measurement. Instead, the ratio between the pixel sizes at different rows in the image is important.	8

Table 10. Science autonomy system configuration variables

Variable Name	**Description**	**Final Value**
targetAcqTopMaxPix	Segmentation parameter defining the maximum number of pixels in a segmented rock region at the top of a high-res image. Any regions with a greater number of pixels will be ignored as a rock too large to be examined with Nomad's arm.	270
targetAcqBottomMaxPix	Segmentation parameter defining the maximum number of pixels in a segmented rock region at the bottom of a high-res image. Any regions with a greater number of pixels will be ignored as a rock too large to be examined with Nomad's arm.	350
targetAcqLineRatio	Segmentation parameter setting the length / width ratio used to determine if a thresholded region is too much like a line to keep in a high-res image.	1.5
targetAcqMaxStddevRatio	Segmentation parameter controlling shadow compensation in high-res images. If the standard deviation of the thresholding color ratio is greater than this value, shadow compensation will be used. Set this value low to force shadow more often.	10
targetAcqCBlue	Segmentation parameter used to threshold a high-res image based on pixel color.	2.5
targetAcqCGreen	Segmentation parameter used to threshold a high-res image based on pixel color.	3.1
targetAcqPercentage	Segmentation parameter defining the pixel thresholding limit in high-res images.	0.60
targetAcqRedThreshRG_Coeff	Coefficient used in segmentation debug (red segmentation) mode.	1.5

Table 10. Science autonomy system configuration variables

Variable Name	**Description**	**Final Value**
targetAcqRedThreshRB_Coeff	Coefficient used in segmentation debug (red segmentation) mode.	1.5
armCamBoxSize	Segmentation parameter that defines the maximum distance between “adjacent” pixels at the bottom of a high-res image. The “box” settings allow a cluster of rock pixels to be lumped together as one region.	3
armCamPixelSize	Segmentation parameter that defines the area of the patch of ground represented by one pixel in an arm camera image.	40
armCamMaxPix	Segmentation parameter defining the maximum number of pixels in a segmented rock region in an arm camera image. Any regions with a greater number of pixels will be ignored as a rock too large to be examined with Nomad’s arm.	76800
armCamLineRatio	Segmentation parameter setting the length / width ratio used to determine if a thresholded region is too much like a line to keep in an arm camera image.	1.5
armCamMaxStddevRatio	Segmentation parameter controlling shadow compensation in arm camera images. If the standard deviation of the thresholding color ratio is greater than this value, shadow compensation will be used. Set this value low to force shadow more often.	35
armCamCBlue	Segmentation parameter used to threshold arm camera images based on pixel color.	2.5
armCamCGreen	Segmentation parameter used to threshold arm camera images based on pixel color.	3.1

Table 10. Science autonomy system configuration variables

Variable Name	Description	Final Value
armCamPercentage	Segmentation parameter defining the pixel thresholding limit in arm camera images.	0.75
armCamMinRow	The image pixel row at which arm camere image segmentation begins. Any pixels in row numbers less than this (i.e. higher in the image) will be ignored. This is needed because the first few rows of an arm camera image are often black.	5
armCamMaxRow	The image pixel row at which arm camere image segmentation ends. Any pixels in row numbers greater than this (i.e. lower in the image) will be ignored. This is needed because the last few rows of an arm camera image are often black.	240
armCamMinCol	The image pixel column at which arm camere image segmentation begins. Any pixels in column numbers less than this (i.e. further left in the image) will be ignored. This is needed because the first few columns of an arm camera image are often black.	20
armCamMaxCol	The image pixel column at which arm camere image segmentation ends. Any pixels in column numbers greater than this (i.e. further right in the image) will be ignored. This is needed because the last few columns of an arm camera image are often black.	300
armCamSpecMinRow	The minimum image pixel row in which a segmented region representing the spectrometer in an arm camera image can lie and be ignored as a target.	195

Table 10. Science autonomy system configuration variables

Variable Name	Description	Final Value
armCamSpecMaxRow	The maximum image pixel row in which a segmented region representing the spectrometer in an arm camera image can lie and be ignored as a target.	225
armCamSpecMinCol	The minimum image pixel column in which a segmented region representing the spectrometer in an arm camera image can lie and be ignored as a target.	15
armCamSpecMaxCol	The maximum image pixel column in which a segmented region representing the spectrometer in an arm camera image can lie and be ignored as a target.	70
specCalibTargetHeight	The defined height of the spectrometer calibration target. This height, given in meters, is defined in a frame where the wheels of the robot are on a ground plane at 0 m.	0.388
hiResFullZoomAutofocusSearchRange	When the high-res camera software operates the lens controller to focus the image automatically, it initially steps through the entire range of focus values. These values are given in a one-byte potentiometer value, where 255 is fully zoomed out and 0 is fully zoomed in. This variable defines the step value of this initial search.	20
hiResTargetZoomVal	The maximum zoom value of the high-res camera lens when deploying to a target. The value is given in a one-byte potentiometer value, where 255 is fully zoomed out and 0 is fully zoomed in.	140

Table 10. Science autonomy system configuration variables

Variable Name	Description	Final Value
hiResZoomStep	The high-res camera is zoomed into a target in a series of steps to avoid losing the target. This variable defines the step value, given in a one-byte potentiometer value, where 255 is fully zoomed out and 0 is fully zoomed in.	40
manipulatorTimeCost	The constant, unitless time cost for deploying Nomad’s manipulator arm.	10.0
manipulatorEnergyCost	The constant, unitless energy cost for deploying Nomad’s manipulator arm.	10.0
manipulatorWorkspaceMinDistance	The minimum radial distance defining the workspace of Nomad’s manipulator arm, given in meters. The distance is defined from Nomad’s DGPS sensor.	2.1
manipulatorWorkspaceMaxDistance	The maximum radial distance defining the workspace of Nomad’s manipulator arm, given in meters. The distance is defined from Nomad’s DGPS sensor.	3.3
manipulatorWorkspaceLeftAngle	The left angle defining the workspace of Nomad’s manipulator arm, given in radians. The angle is defined from Nomad’s DGPS sensor with 0 rad pointing straight out the front of the robot. This angle is defined in the CCW direction.	0.52
manipulatorWorkspaceRightAngle	The right angle defining the workspace of Nomad’s manipulator arm, given in radians. The angle is defined from Nomad’s DGPS sensor with 0 rad pointing straight out the front of the robot. This angle is defined in the CW direction.	6.05
hiResTimeCost	The constant, unitless time cost for deploying the high-res camera.	1.0

Table 10. Science autonomy system configuration variables

Variable Name	**Description**	**Final Value**
hiResEnergyCost	The constant, unitless energy cost for deploying the high-res camera.	1.0
hiResWorkspaceMinDistance	The minimum radial distance defining the workspace of the high-res camera, given in meters. The distance is defined from Nomad’s DGPS sensor.	3.75
hiResWorkspaceMaxDistance	The maximum radial distance defining the workspace of the high-res camera, given in meters. The distance is defined from Nomad’s DGPS sensor.	7
hiResWorkspaceLeftAngle	The left angle defining the workspace of the high-res camera, given in radians. The angle is defined from Nomad’s DGPS sensor with 0 rad pointing straight out the front of the robot. This angle is defined in the CCW direction.	0.7854
hiResWorkspaceRightAngle	The right angle defining the workspace of high-res camera, given in radians. The angle is defined from Nomad’s DGPS sensor with 0 rad pointing straight out the front of the robot. This angle is defined in the CW direction.	5.4978
dgpsEstPan_Fudge	The pan angle fudge factor (in degrees) used to correct the estimated DGPS position of rock targets. This angle is added to the measured high-res camera pan angle.	0.0
dgpsEstTilt_Fudge	The tilt angle fudge factor (in degrees) used to correct the estimated DGPS position of rock targets. This angle is added to the measured high-res camera tilt angle.	-2.0

Table 10. Science autonomy system configuration variables

Appendix D

Many different rocks and meteorites were examined by Nomad each day of the expedition. Some of these are listed in the text as “Rock A”, “Liam’s rock #1” or “Meteorite 12332”. This appendix describes these samples in detail.

For a good online library of rock types, see the Big Signal rock library at www.bigsignal.net.

Rock Set A

These rocks were acquired from the outer edges of Elephant Moraine January 15, 2000 by Liam Pedersen. The rocks were initially identified by John Schutt.

Field Identification	Description
1	Dolerite
2	Quartz sandstone
3	Red basalt (hydrothermally altered)
4	Fine grained sandstone (small pieces of coal?)
5	Basalt
6	Travertine (black calcite)
7	Quartzite?
8	Breccia / highly latered basalt, with quartz or calcite intrusions
9	Dolerite
10	Dolerite
11	Quartz sandstone
12	Dolerite

Table 11. Rock set A listing

Rock Set B

This sample set was used for the meteorite search demonstrations between January 21 and January 23, south of camp, in the vicinity of Nomad’s first meteorite find. The rocks were collected and initially identified by John Schutt.

Field Identification	Description
1	Black shale (often considered a meteorwrong)
2	Dolerite
3	Tan siltstone
4	Basalt fragment
5	Dolerite
6	?

Table 12. Rock set B listing

Rock Set C

This rock and meteorite test set was used on the blue ice west of camp -- the location of meteorite search demonstrations #3 - #7 (January 25 and 26). The samples were initially identified by John Schutt.

Field Identification	Description
A	Dolerite
B	Dolerite
C	Red basalt (hydrothermally altered)
D	Dolerite
E	Quartz sandstone
F	Subglacial travertine (black calcite, often considered a meteorwrong)
59650	Kulnine (meteorite provided by William Cassidy)

Table 13. Rock set C listing

Rock Set D

These rocks and meteorites were found and initially identified by John Schutt at the periphery of the Elephant Moraine, northeast of camp. These samples were examined by Nomad on January 27 and later collected and sent to Johnson Space Center by John Schutt.

Field Identification	ANSMET Field Number	Description
A	12387	Ordinary chondrite meteorite
B	12322	Unusual achondrite
C	12346	Unusual black rock, considered a meteowrong
D	12341	Unusual black rock, considered a meteorwrong

Table 14. Rock set D listing

Samples C and D are strange black rocks, similar to black sandstones in appearance but slightly fused. John Schutt cannot say with certainty that they are not meteorites. Furthermore, they are distinctly different and smaller from all other rocks in the area, which are mostly sandstones, red basalts and dolerites. They were located in an area with very few terrestrial rocks and known to contain meteorites.

Moraine Rock Set

These rocks were in the path of Nomad's meteorite search demonstrations on January 28 and 29 (demonstrations #8 and #9). These demonstrations were performed in Elephant Moraine proper, mostly on sastrugi. The samples were identified by John Schutt and Liam Pedersen.

Rock Identification	Description
M0	Red basalt
M1	Dolerite
M2	Dolerite
M3	Sandstone
M4	Dolerite (red iron stain)
M5	Dolerite (slight red iron stain)
M6	Dolerite / basalt
M7	Dolerite (red iron stain)
M8	Dolerite (slight red iron stain)
M9	Dolerite (slight red iron stain)
M10	?

Table 15. Moraine rock set listing

Rock Identification	Description
M11	Dolerite
M12	Dolerite
M13	Dolerite
M14	Basalt (black)
M15	Dolerite (red iron stain)
M16	Dolerite (very red iron staining)
M17	Dolerite (very red iron staining)
M18	Dolerite
M19	Basalt
M20	Dolerite
M21	Basalt
M22	Medium-grained sandstone
M23	Dolerite (very red iron staining) with quartz band (possibly flow basalt)
M24	Dolerite
M25	Sandstone (fine, white grained)
M26	Chert or thermally metamorphosed lake sediment, shale
M27	Basalt
M28	Basalt, with large vesicles so it probably solidified near the surface
M29	Basalt
M30	Quartzite metasediment, derived from clast within conglomerate
M31	Large-grained dolerite (red iron stain)

Table 15. Moraine rock set listing

Meteorwrong Set

This set includes various meteorwrongs collected by John Schutt and Liam Pedersen.

Rock Identification	Description
MW1	Brown shale with desert varnish
MW2	Black shale

Table 16. Meteorwrong set listing