Deployment Methods For Testing and Verification of a Planetary Landing Sensor Package

Josh Caputo
Mechanical Engineering
Carnegie Mellon University
Pittsburgh, PA 15213
jmcaputo@andrew.cmu.edu

Salem Cherenet
Mechanical Engineering
Carnegie Mellon University
Pittsburgh, PA 15213
salemch@gmail.com

Raul H. Figueroa
Engineering & Public Policy
Carnegie Mellon University
Pittsburgh, PA 15213
raulf@cmu.edu

Fraser Kitchell
Energy Science, Technology and Policy
Carnegie Mellon University
Pittsburgh, PA 15213
fraser.kitchell@gmail.com

Abstract

Autonomous lunar landers estimate position and orientation to land and navigate safely on the moon by registering sensor data to pre-compiled terrain models. The sensor package contains an Inertial Measurement Unit (IMU), stereo cameras, a nodding Laser Imaging and Ranging unit (LIDAR) unit, and a micro-computer, to exhibit this principle. Testing and verification of the sensor package requires collecting sensor data in a variety of platforms and terrains so that software algorithms are subjected to alternative data scenarios. This paper presents a set of methods to generate the sensor data that will be used to characterize sensor performance for navigation and landing. Sensor data is then compared to GPS data to develop qualitative metrics for data quality and algorithm effectiveness. The implementation of these methods will yield a repository of IMU and camera data to aid in the gradual refinement of the estimation algorithms.

1 Introduction

Terrestrial autonomous vehicles commonly use the Global Positioning System (GPS) to determine their position, direction and speed. (Sukkarieh, Nebot, & Durrant-Whyte, 1999) Planetary landers do not have this luxury. Spacecraft can navigate using radio triangulation and Doppler shift with Deep Space Network (DSN) satellites and ground stations. (Cesarone, Deutsch, & Abraham, 2007) Moreover, these capabilities are expensive and not fully available for non-Federal missions. Therfore, planetary landings will require alternate navigational technologies and capabilities. (Li & Liu, 2009) Robotic missions must increasingly sense and register their landing destination, and with this goal in mind NASA has sought the capability of pin-point planetary landing since the early 1990's. (Li & Liu, 2009)

Astrobotic Technology, Inc. has developed a collection of sensors (referred to as a "sensor package") for the Lunar Lander that can estimate position and attitude with algorithms that combine stereo image data with IMU data. This is done in a similar fashion as the Applanix Corp. POS/AV model suggested by Ip, et al. in their 2001 paper. (Ip, Mostafa, & El-Sheimy, 2001)

The research discussed here elaborates a sensor deployment methodology for the incremental test and verification of the sensor package. A suite of deployment methods are discussed. This paper also includes discussion of data gathered from test runs with the sensor package installed on a stationary frame, on a movable cart, on an off-road vehicle, while descending a custom built zip-line made of a cart suspended by pulleys running down a tight cable, and during flights on an autonomous helicopter.

Further work was done with a nodding LIDAR unit—also contained in the sensor package (Figure 1)—to determine the capability to detect hazard obstacles that might be encountered during landing. Hazard detection is a mission critical capability, and flight readiness requires the ability to detect and avoid obstacles. The capability of the LIDAR unit and the hazard detection software was tested using data gathered from a stationary frame.

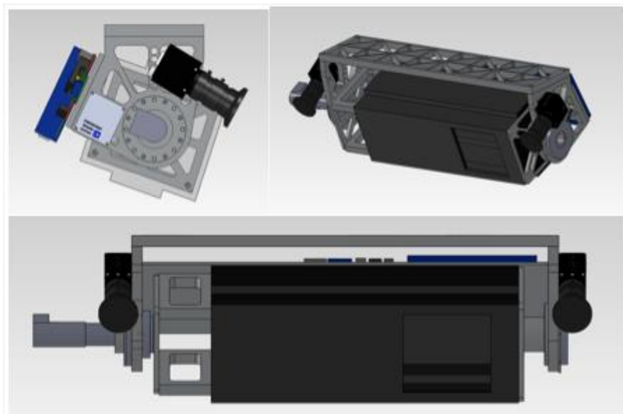

Figure 1: Plans showing the details of the sensor package. Clearly visible is the LIDAR unit (center, black) and the Stereo Cameras (sides, black). The IMU is housed inside. (Source: Astrobotic Technology, Inc.)

1.1 Collection, pre-processing and post processing.

Figure 2 shows the process flow of how data logging and pre-processing is accomplished. The IMU, LIDAR, stereo cameras, and motor encoder package information and pass it through filters and onboard computers to the flight planner. Variable values are recorded into Robotic Operating System (ROS) libraries. ROS provides several libraries for data recording and interpretation. Raw data is recorded and saved as a "bag" file (a file format for storing ROS messages) for subsequent processing.

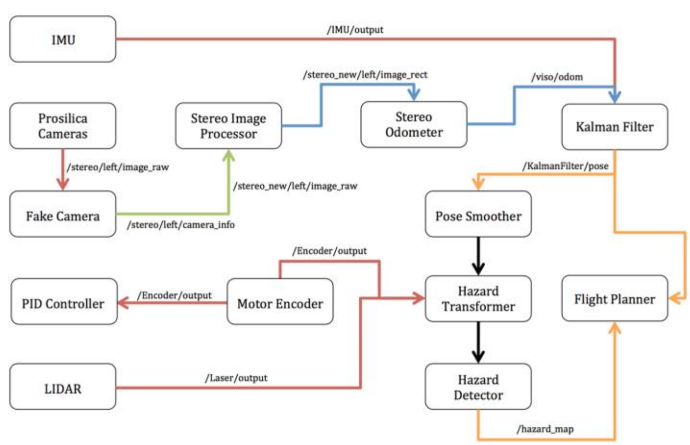

Figure 2: Flow diagram of sensor data pathways, hazard detection, and flight planning.

Once raw data is recorded it is passed through filters to reduce noise, and finally the post-processed data is fed into the navigation, hazard detection, and flight planning algorithms. Data storage is discussed further in the appendix.

To house the large amounts of data collected during the test and verification process it is convenient to have a database to organize and maintain the repository of data collected over time. This database must be easily accessible to future researchers to refine sensor and software capabilities. An example of such a database is proposed in the appendix.

1.2 Motivation

Designing, testing and verifying a sensor and software package that can navigate and detect hazards is a crucial step in demonstrating the ability to land safely on the moon. Lunar landing could not feasibly be attempted without a robust system aboard that assures it can land safely. (Epp & Smith, 2006)

For the Astrobotic lander, a successful landing implies that the lander touches down, intact, with four landing foot-pads firmly planted on flat and solid ground, and with enough clearance for ramp deployment and for the lunar rover to descend to the surface. Simply landing on the surface is not enough to guarantee a successful mission: landing on too steep a slope, on a boulder, or in a crater would doom a mission. Shown in Figure 4 is the approximate landing trajectory that a lunar approach will follow.

Thus testing the sensor package and the technologies that will be used on a potential lunar mission is critical. The technical validity of individual systems must be established, and the limits of their capabilities must be pushed and documented. The goal of testing is to establish quantitative metrics of performance and reliability.

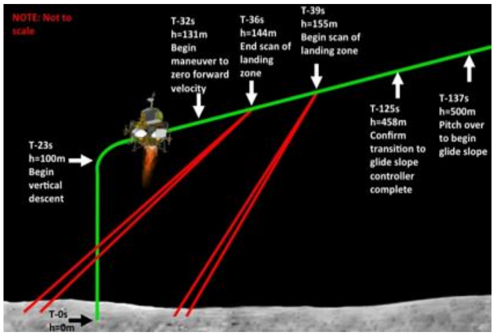

Figure 4: These flights approximate the lunar-relevant trajectory depicted above. The sensor package will be relied upon to manage this final descent.
(Source: Kevin Peterson, Astrobotic Technology, Inc.)

2. Previous work

There has been much previous work on navigating using stereo cameras combined with IMU. The Autonomous Precision Landing Hazard Avoidance Technology (ALHAT) project, a NASA thrust to develop and deploy technologies that could allow a spacecraft to land on an extra-terrestrial surface without human intervention, was initiated in 2005. (Epp & Smith, 2006). The goals laid out in ALHAT demand that technology demonstrates the ability to detect and avoid obstacles larger than 30 cm, along with demonstrable precision and slope avoidance. (Ibid) However, while ALHAT documentation suggests types of sensors that might be used for navigation, the specific use of one sensor over another is not suggested. Capability trumps the choice of any given technology. Cameras and IMU offer independent redundancy that might allow precise, robust navigation in the absence of GPS or other external aids. (Mirzaei & Roumeliotis, 2008) Both systems are robust, compact, highly developed, and energy efficient; ideal qualities for instrumentation on extended space missions. (Mourikis & Trawny, 2009)

LIDAR units have been used for field ranging nearly from the outset of their invention, and the techniques allowing lasers to be used for mapping become increasingly advanced over the years. (Wehr & Lohr, 1999) Astrobotic's sensor package uses a nodding LIDAR to scan across the ground in two axial planes. LIDAR-based hazard detection, a capability demanded of the sensor package, has only been used more recently. A team at Carnegie Mellon University has developed and tested laser based hazard detection, showing the potential capabilities of the technique. (Henriksen & Krotkov, 1997) Methods for testing and verification of similar sensor packages have more often been tailored to specific application on terrestrials scenarios. However, published work depicting methods to test systems like this for planetary landing applications in a systematic way, has been scarce. This paper fills part of that gap.

3. Sensor Package Testing

A variety of deployment methods were developed to collect sensor data. Bench tests, performed indoors with the sensor and attached to a stationary frame, were used initially to confirm that software and instrument communication was calibrated and active. Vehicle-based tests also offered quick accessibility to testing grounds and introduced uncontrolled variables of vibration, lighting and constant-elevation movement, under soil conditions that closely mimic lunar conditions are preferable. Zip-lines were used to deploy the sensor package in a 30 degree descending trajectory similar to the one Astrobotic's lander will take during final descent stages on the moon. Additionally, an autonomous helicopter, guided by pre-programmed GPS waypoints, provided the capability to test the sensor package from greater heights and in a variety of trajectories. The setup of each test method is described in more detail below.

3.1 Frame tests

Frame tests, performed with the sensor indoors and attached to a rigid, moveable frame (Figure 5), were used in the beginning of the testing program. Frame tests provide for quick and easy confirmation that the software and instruments are calibrated and communicating adequately. In a highly controlled environment, these tests are simple, repeatable, and independent of weather. Data collection using the frame proved crucial in initial tests of a hazard detection algorithm.

Figure 5: The sensor package, containing an IMU, stereo camera, LIDAR and microcomputer. The image shows the configuration of the sensor package mounted on the stationary frame.

3.2 Driving Test

An off-road vehicle was used to carry the sensor package on a trajectory tangential to the ground. These trials were a valuable in determining sensor and software performance in motion in a dynamic environment, at higher speeds, and under heavy vibration. The vehicle used contained an Applanix Corp. POS/AV which provided a precise estimate of location and speed. The trajectory calculated by the sensor package was compared to that calculated by the Applanix.

The test site was a brownfield in Pittsburgh, PA, with dirt roads and large abandoned fields. The field was proximal to campus, and facilitated testing the sensor package at low speeds, safely, and conducting tests without disturbing traffic. The sensor package was tested in conditions of low light and cold temperatures (Evening; 2° C)

The data gathered from the cameras, LIDAR and IMU was logged on the onboard computer.

Figure 6: The Hummer off-road vehicle with the sensor package attached

3.3. Zipline Tests

Ziplines allow for a controllable, gravity driven, repeatable, and constant trajectory descent. A zipline was built to test the sensor package at higher elevations and at a trajectory that would mimic an actual lunar landing. The zipline began 10 meters above the ground descended to a point approximately 1 meter above the ground. The sensor package was attached to pulleys and a computer logged the data during the tests. Eight trial descents generated LIDAR, IMU and camera data.

Figure 7. The zipline setup shows the sensor package hanging on cables below the computer. The cables were attached to the staircase seen in the background, approximately 10 meters off the ground.

3.4 Autonomous Helicopter Flights

The most complete data sets were compiled using an autonomous helicopter. The helicopter was operated by a team from Virginia Polytechnic Institute and State University (Virginia Tech), with mechanical and software interfaces accomplished by the Astrobotic Technology and Carnegie Mellon University team. The entire sensor package was mounted between the skids of the helicopter, and data was collected from the cameras, IMU and LIDAR. Flight time totaled 1.5 hours, spread over two days, and provided the collection of a large data set. This test also introduced vibration and field depth variables unavailable on prior test setups.

Figure 8: The helicopter with the sensor package mounted below the body of the craft.

3.5 Random hazard field generation and detection

As part of the process of testing the sensor package is ensuring that the LIDAR is capable of detecting obstacles, work was done to develop methodologies to generate and deploy a hazard field. To ensure that the hazards fields were as random as possible, a "random hazards field generator" was developed. This code randomized both size and position of the hazards within a pre-determined grid. Figure 9 shows a typical output from the hazards field generator. For physical test, boxes were distributed as per the output of the algorithm.

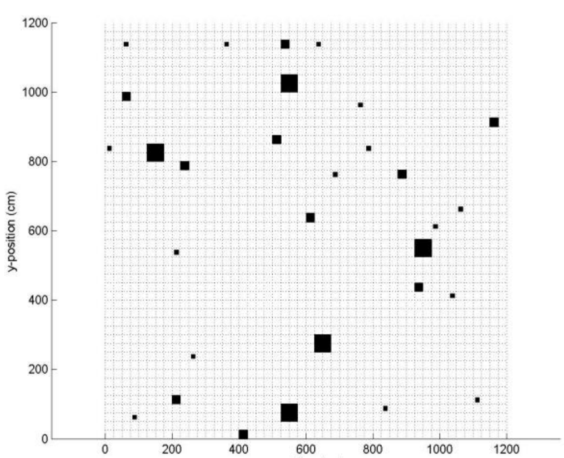

Figure 9. Typical output from the random hazards field generator.

4. Results

A total of 15 data sets comprising 8.5 hours of test time were collected from bench, cart, off-road vehicle, zipline and helicopter trials. Two hazard detection tests proved the LIDAR capable of detecting objects at the smallest length scale tested, 12.5 cm. This is enough to comply with ALHAT standards, which state that the landing sensors must be capable of detecting obstacles larger than 30cm. Navigation data generally proved incompatible between major revisions of the algorithms in its earlier stages, due to necessary variation in the software's front end, but hazard detection algorithms proved reliable.

4.1 Stationary and mobile cart tests results

Ground testing was used for both camera and IMU testing and LIDAR hazard detection. The data from these tests was used for subsequent tests in more rigorous testing environments.

4.2 Off-road vehicle test results

Off-road vehicle testing was performed on two dates, and furthered the integration of the instrumentation. During each day, data with the sensor package in motion were logged from the IMU, the cameras and the LIDAR. The IMU and camera data were processed with a Kalman Filter. The position estimated by the sensor package was compared with position estimates from the Applanix. Results indicated that the instrumentation of the filter was unable to correctly navigate through the field. Turns that were approximately 90° on the GPS registered as a greater angle through the filter, indicating that the computer believed the vehicle was turning at a greater angle than it actually was. Two complete loops were driven, and while the GPS showed an exact return to position, the Kalman data placed the vehicle approximately 100 meters away from it's actual position. This finding spurred the further refinement of the navigation algorithms.

4.3 Zipline test results

Eight zipline descents provided IMU, camera and LIDAR data. LIDAR data proved able to discern objects. The zipline data had the added dimension of descent. Again, this data helped evaluate intermediate incarnations of the navigation software.

4.4 Helicopter flight results

Helicopter flight tests collected 1.5 hours of data from camera, IMU and LIDAR. Flight patterns are shown in Figure 11. The purple line shows the path of the helicopter as recorded by the autonomous helicopter's onboard GPS. Altitude data is not shown in this image. Data gathered confirmed that navigation algorithms still need further refinement to match the GPS data. The flight patterns programed into the autonomous helicopter, using GPS waypoints, followed a straight line path in 30 degree descents from different angles over the same hazards field. Wind, as expected, caused deviations from the desired straight-line path. The non-linearity of the paths depicted in Figure 11 is due to wind gusts and corrections during flight.

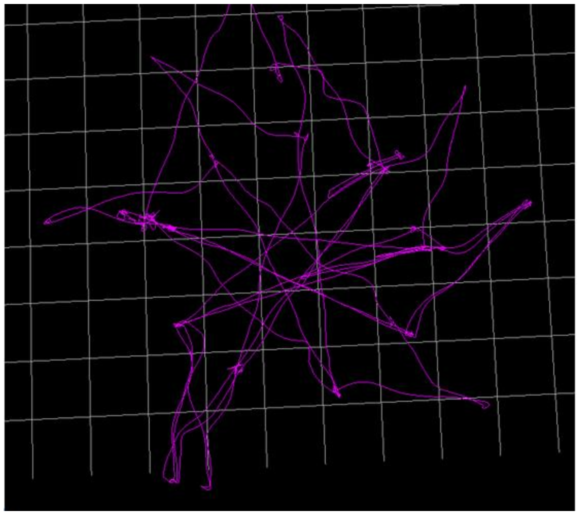

Figure 11: Overhead map of flight patterns during the helicopter test.

4.5 Hazard detection results

The initial hazard detection test was performed using the cart setup described above. The hazards field was set as per the random generator previously described. Figure 11 shows the point cloud imagery detected by the nodding LIDAR. Raw point cloud data was processed and the algorithm correctly determined adequate landing positions. The results of the algorithm are shown in figure 12. Green areas indicate areas that meet the qualities required for a safe landing, as calculated by the algorithm. Close agreement between the two images can be observed.

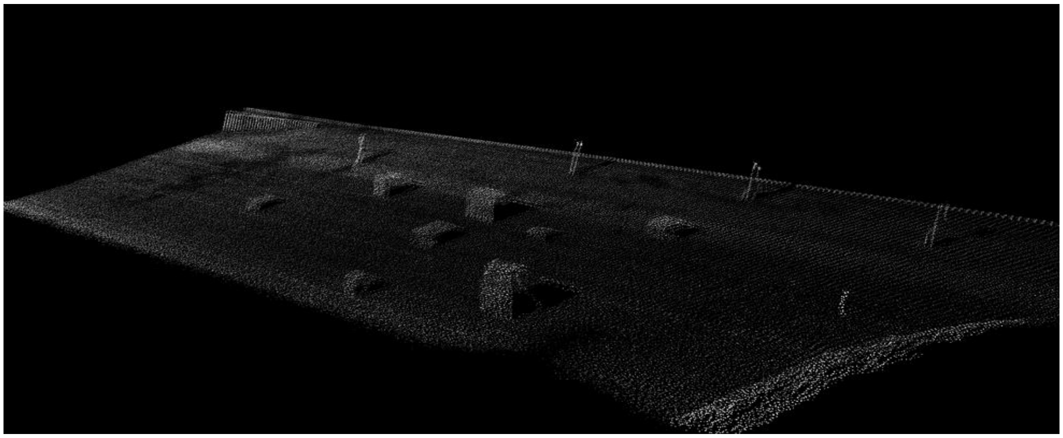

Figure 11: Hazard field LIDAR scan, showing obstacles and terrain topography.

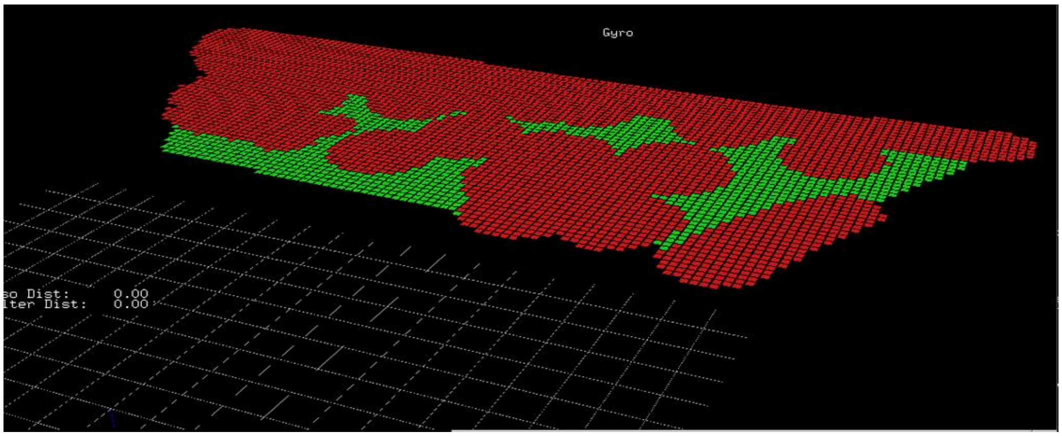

Figure 12: Hazard detection algorithmic output, showing landing areas (green) and hazard zones (red). Landings would target only green zones.

5. Conclusions

The data sets collected have edged instrument and software development closer towards the goal of a robust, autonomous navigation. However, due to changes in the design and integration of the sensor package and software, it was discovered that data sets were not useful for testing subsequent, more refined versions of navigation algorithms. Thus, continuous field testing will be required as the software evolves towards its final space launch version.

Tests proved that more work is needed before IMU and camera integration can be considered reliable. Hazard detection proved more qualitatively capable, demonstrating a no false positives and no false negatives.

The collaboration with Virginia Tech proved invaluable to Carnegie Mellon's and Astrobotic's earlier test of the sensor package and its algorithms, but travel time, system integration, and team coordination proved challenging. Further test of the navigation and landing sensor package and software will require the frequent use of a reliable aerial platform. Investing in a mini helicopter, such as the Pulse Aero Scout used at Virginia Tech, might be worthwhile, and perhaps unavoidable, unless other affordable and readily deployable alternatives are identified in nearby.

6.0 Bibliography

Wehr, A., & Lohr, U. (1999, March). Airborne laser scanning—an introduction and overview. 68-82.

Cesarone, R. J., Deutsch, L. J., & Abraham, D. S. (2007). Prospects for a Next Generation Space Network. Proceedings of the IEEE , 1902 - 1915.

Epp, C. D., & Smith, T. B. (2006). Autonomous Precision Landing and Hazard Detection and Avoidance Technology (ALHAT). IEEEAC , 1-7.

Ip, A. W., Mostafa, M. R., & El-Sheimy, N. (2001). System Performance Analysis of ING/DGPS Intergrated System for Mobile Mapping Systems (MMS).

Henriksen, L., & Krotkov, E. (1997, April). Natural Terrain Hazard Detection with a Laser Range Finder. IEEE , 968-973.

Li, S., & Liu, Z. (2009). Autonomous navigation and guidance scheme for precise and safe planetary landing . Aircraft Engineering and Aerospace Technology , 516-521.

Mirzaei, F. M., & Roumeliotis, S. I. (2008). A Kalman Filter-Based Algorithm for IMU-Camera Calibration: Observability Analysis and Performance Evaluation. IEEE Transactions on Robots , 1143-1156.

Mourikis, A., & Trawny, N. (2009, April). Vision-Aided Inertial Navigation for Spacecraft Entry, Descent, and Landing. IEEE Transactions on Robots .

Sukkarieh, S., Nebot, E., & Durrant-Whyte, H. (1999). A high integrity IMU/GPS navigation loop for autonomous land vehicle applications. Robotics and Automation , pp.572,578.

Appendix

7. Future Work

7.1 Hazard Detection

The two essential software components for landing site selection are an estimate of where the lander is in space and how it is orientated relative to the celestial body's surface, and the textural details of that surface. In the case of the Moon, rough details of the surface are given by a lunar-orbiting satellite (the Lunar Reconnaissance Orbiter, LRO), but fine details that can significantly influence landing are too small to be detected by these high-altitude methods. Thus, the lander will scan the terrain for obstacles in real-time using its LIDAR scanner. A successful landing depends in great part in the lander's ability to detect the obstacles above a certain size

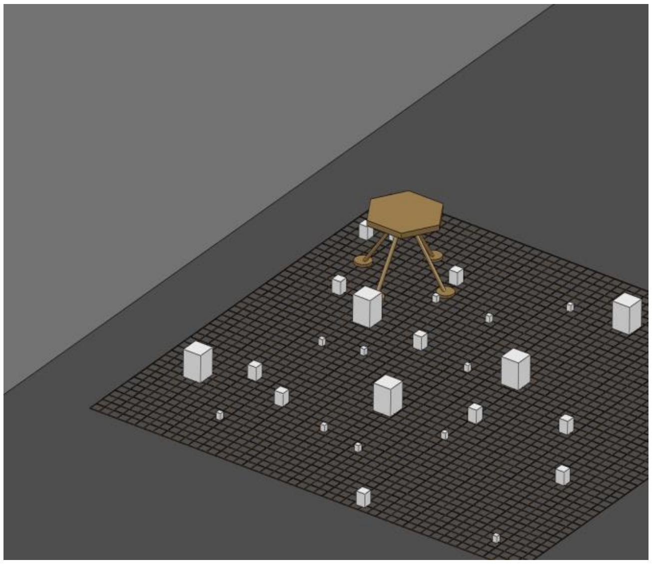

Figure 10: Scale drawing of the hazard detection algorithm testing scene. Number and size of grid squares and boxes appear as they will in the test. Boxes are placed randomly throughout the scene. Illustration of the lander included for scale.

In testing, the surveyed terrain—the hazard field—should be gridded into 25cm sections and an algorithm determines, grid square by grid square, whether a hazard occupies the grid square or not. Performance metrics are percent false positives and percent false negatives in the detection (Equations 1, 2).

$$\%FP = \frac{number\ of\ obstacles\ not\ present\ but\ detected}{total\ number\ of\ obstacles} * 100$$

$$\%FN = \frac{number\ of\ obstacles\ present\ but\ not\ detected}{total\ number\ of\ obstacles} * 100$$

Equations 1, 2: False positive and false negative performance metrics for evaluating the hazard detection software.

In trials, thirty boxes are to be randomly distributed on a 12m x 12m grid. 12.5cm, 25cm, and 50cm cube boxed are used based both on the obstacle sizes that are anticipated on the lunar surface and the minimum resolution of the scanning sensor. Subsequent tests should include larger obstacles and uneven ground. Many variables could affect the performance of the hazard field analysis, such as position of the sun in the sky and position and orientation of the sensor package relative to the scene. Thus, tests should be conducted in the morning, mid- afternoon, and late afternoon. Practically, the sensor package can only view the scene from one edge of the grid, so tests should be conducted viewing the scene straight-on and from each extreme viewpoint.

Systematic evaluation of the lander's sensor suite must be performed in order to determine readiness for a lunar mission. Evaluation is to be conducted at all levels of the system, from the specific low-level hardware components, to flight control and planning algorithms. Our efforts have focused on testing the pose-estimation and hazard detection algorithms. The following details our plans for systematic testing of these aspects of the sensor's functionality. As the sensor package and its software are refined, the researches will need to decide on which test to carry out. There are trade-off with these choices. Tables 1 and 2 provide a basic look recommended testing methods and the trade-offs involved.

Table 1 Recommended testing methods, sensor package

Order Evaluation	Trajectory Description	Vehicle	Ground Truth
1	Standing still	Cart	-
2	Rotating in place	Cart	Laser tracker
3	Forward and back	Cart	Laser tracker
4	Travel in a circle	Cart	Laser tracker
5	Travel in a closed loop	Hummer	GPS + IMU
6	Controlled descent	Zipline	Laser tracker
7	Landing trajectory	Horizontal landing simulator	GPS+IMU
8	Landing trajectory	Helicopter	GPS + IMU
9	Landing trajectory	Rocket Propulsive Vehicle	Laser tracker

Table 2. Qualitative ranks of testing methods

	Ease	Repeatability	Capacity	Realism	Expense
Frame	10	10	9	2	9
Vehicle	9	8	7	6	7
Zipline	7	8	6	7	8
Horizontal Landing Simulator	6	8	9	8	7
Helicopter	4	7	5	8	4
Propulsive Lander	2	9	7	9	2

7.2 Pose Estimation

Pose estimation is a necessary component of many higher-level lander tasks, such as flight path control and landing site selection. Therefore, rigorous testing of the pose estimation algorithms is critical. The pose estimate is a combination of an estimate of the lander's position and its orientation, each of these estimates will be computed and reported separately. The error in the estimate is defined by the RMS error (Equation 3). Through revision of the sensor's software, the objective is to reduce this error below a to-be-determined % of the total length of the trajectory.

$$RMS = \sqrt{\frac{1}{n}((x_1 - x_1^*)^2 + (x_2 - x_2^*)^2 + \cdots + (x_n - x_n^*)^2)}$$

Equation 3: RMS is the measure of error (position or orientation error), n is the number of samples in the data set being evaluated, x is the magnitude of the pose estimate (square-root of the sum of the squares of position or orientation) at a particular time step, and x* is the ground truth measurement at that time step.

Evaluation of the sensor is conducted in this way on trajectories of progressively increasing complexity, starting with immobile pose estimation up to pose estimation during a flight path that mimics that of the lunar landing.

Basic tests are to be conducted using a wheeled-cart with ground truth provided by commercially-available laser tracker hardware. The laser tracker tracks the position of a reflective prisms in Cartesian space. Using two prisms and laser trackers, orientation of the sensor on a plane can also be determined. Three prisms are required to track orientation in full 3D space. To test that the sensor performs satisfactorily well in all 6 dimensions (3 translation, 3 rotational), the full suite of cart tests will be conducted with the sensor package oriented in three, preferably orthogonal, orientations. This simplifies testing by reducing the number of prisms to 2 and keeping the trajectories on a 2D plane.

The ability of the position and orientation estimates to maintain accuracy over long traverses will be tested using the DARPA Grand Challenge Topographer Hummer. Ground truth can be provided by this vehicle's on board commercially-available pose estimate unit, since the range of the test is too great for standard laser-tracking systems. This will simultaneously test all of the trajectories performed using the cart, as the Hummer will drive over 3D terrain. Since the vehicle travels in a closed loop, the position and orientation of the sensor is the same at the beginning and ending of the trajectory. Thus, an interesting metric is the RMS error of the distance between the starting and ending pose.

The trajectory taken during the hummer test is qualitatively quite different than that which will be taken by the lunar lander. To work towards a more moon-realistic trajectory, zipline tests will be conducted where the line is strung at the angle used during the lunar descent. Speed will be controlled to approximate that of the lunar lander. Beyond the standard RMS error, error between the known speed and measured speed and the known line angle and estimated angle would be additional interesting metrics. This test will be ground-truthed using the laser tracker hardware as it is conducted over a relatively short traverse.

Finally, with the sensor unit attached to the Virginia Tech remote control helicopter, performance of the algorithm can be tested in highly-realistic manner. Ultimately, the helicopter would be replaced by a propulsive lander for increased realism. This evaluation is the most relevant to the mission's trajectory, but also the most difficult to conduct. While the cart tests could be conducted hundreds of times, this is impractical using the helicopter. Thus, early in the development of the landing sensor, focus is to be given to the simple testing methods, in preparation for further testing on the helicopter. Using a horizontal landing simulator, mentioned before, and described in section 6.3, would be a practical and economical way to scale up in preparation for flight tests.

7.3 Batch Processing

Batch processing must be robust and repeatable. Therefore, one of the focuses of this project was to develop a simple, interactive interface to create a database for raw and processed data. This research team approached this problem by first identifying what type of information the name of each data should contain. Table 3 shows the different classifications of required information for creating a database that is easy to navigate through.

Type	**Location**	**Test Date**	**SVN#**
Static	Robot City	Month_year_date_hour_minute_second	Software version number
Zipline	CMU Highbay		
Arial	CMU Gates Bridge		
Hummer	Virginia Tech		

Table 3: Information classification for structured database

The first column describes what kind of test was done. The lists in the second column of Table 3 are few of different places where data has been collected. It is important to establish a standardized system of naming the different values recorded from the sensor package. Future researchers will need to take pains to keep variable names consistent wherever possible.

Figure 14 shows a snapshot of a suggested GUI. At the beginning, the user saves the raw data inside a folder "data/inidata". Then he/she needs to run the python gui file "db_maker.py" by giving the command "./db_maker.py" or "python db_maker.py" inside a unix terminal in order to launch the GUI.

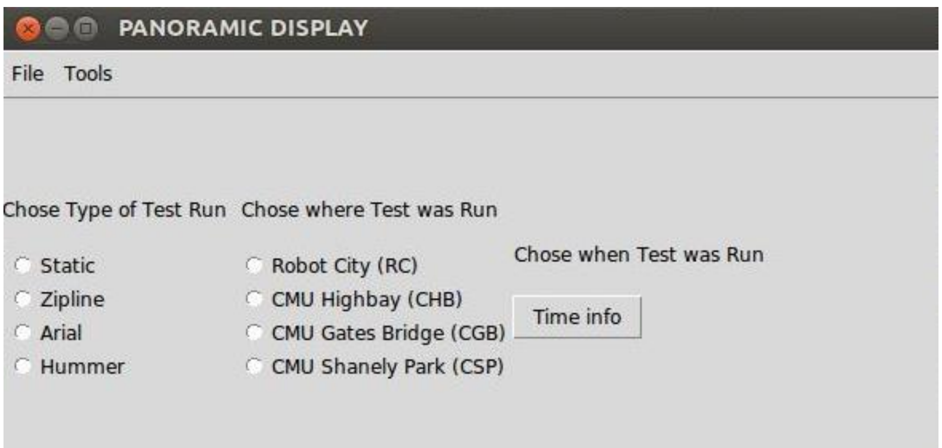

Figure 14: GUI for creating structured database

Once the GUI launches, the user may choose:

A. Type (CMU Highbay, CMU Schenley Park, VaTech, Robot city)
B. Location (year, month, date, hour, min, sec)
C. Software Version (SVN number)

Then click "save" under the file menu

The save command looks through the "data" folder if a "type of run_svn#' folder exists. If it does not exist, this command will create a new folder. Next it looks for a subdirectory of where the test was taken. If the directory doesn't exist it will create that as well. Finally it copies the raw Rosbag to this folder and renames it as “date.rosbag”

For example, a user might enter the following data:

"Static "	-For type of test
"Robot City"	-For the test location
"2013 3 31 11 57 46 1"	-For Y, M, D, H, M, S, Svn num.

Clicking "save" would create a new directory called "data/static_1". Then the program creates a subdirectory "data/static_1/RC" then a new dialog would open asking the user to choose the rosbag file they want to save. Once the user chooses the right rosbag, the program saves the file as "data/static_1/RC/2013_3_31_11_57_46.rosbag"

7.4 Horizontal Landing Simulator

Landing trajectories can be simulated by moving the sensor package towards a vertical surface. A vehicle, such as the off-road Hummer used before, provides enough stability and size to install a tower on top of which the sensor package can ride. This methods could provide many advantages including lower cost per test, and controlled speeds and trajectories less prone to wind disturbance as compared to test performed on autonomous helicopters. Further research could develop this testing method. Testing surfaces could range from mine walls, to barns, to academic buildings. The simulated obstacles could be boxes hung from the top in patterns suggested by the random hazards field generator.