UNIVERSITY OF PENNSYLVANIA

GENERAL ROBOTICS, AUTOMATION, SENSING & PERCEPTION

(GRASP) ROBOTICS LABORATORY

RASC-AL ROBO-OPS 2012 FINAL REPORT

ENVOY2:
Exploration and Navigation Vehicle for geolOgY 2

Authors / Team Members:

Jonas CLEVELAND, Mabel Mengzi ZHANG, Sanchit ARORA, Neel SHAH, Vivian CHU, Nicu ŞTIURCĂ, Rajeev KUMAR, Yedong NIU, Varsha SHANKAR, Praveen PITCHAI, Tyrone Christopher THOMAS, Bennett LEE, Michael HURST, Keunho KIM, Siddharth RAMANATHAN, Kevin ALCEDO, Julieta GARCIA, Shafag IDRIS, Agata CIESIELSKI

Faculty advisor:
Dr. Mark YIM
yim@grasp.upenn.edu

Abstract

In this paper we describe USMART's entry in the NASA RASC-AL Robo-Ops Competition. The Upenn Space Mobile and Autonomous Robotics Team (USMART) is comprised mainly of several graduate students in the GRASP Laboratory. We have also garnered participation from undergraduates on Penn and Drexel campuses; they have assisted assembly, in-house manufacture, and our outreach activities. We pride ourselves first in the impact of our outreach programming, which reached a wide audience across demographics. On the robot platform, USMART has developed a four-wheel drive rover, capable of traversing varying terrains and scooping objects of multiple shapes. We employ autonomous and semi-autonomous systems in order to achieve high control and reliability in a changing environment. Our robot utilizes a modular design philosophy for rapid development cycles and maintenance time and the ability to swap sub-systems.

1. System Overview

ENVOY2 is designed with careful consideration for the constraints and conditions robots face on lunar and Martian environments. In a simple yet reliable mechanical framework, all electronic hardware is encased in a sealed aluminum structure for cooling and protection. ENVOY2 communicates with a modem over the 4G network and powers a CPU with flash memory. Moreover, all hardware components have been in use by industry for over 7 years. Each wheel is independently controlled for fail-safe redundancy. We use a light-weight, quick-charge lithium polymer battery to provide over 2 hours of battery life during continuous operation. Our frame and mechanical components have been tested in various positions in simulation to insure their integrity. We use a diverse array of sensors for reliable estimation in unexplored environments. Additionally, ENVOY2 has inherited some components from ENVOY1, which was tested at the Johnson Space Center Rockyard, including the majority of the mechanical design for the arm.

Selecting a professional framework for convenient and reliable development proved to be crucial in implementation and testing in a pressing schedule. We practice a modular design philosophy in both hardware and software by using the popular Robot Operatying System (ROS) from Willow Garage as the foundation for our software framework [4]. ROS is conducive to parallel development, which enabled us to develop communications, localization, and tele-operation nodes independently. Carefully planning the information published and subscribed, or the ROS message passing terminology between nodes, integrating the components in our initial design was smooth.

Inheriting from ENVOY1 to ENVOY2, we kept many positive design aspects from last year, including a similar drive-train. For the design of other mechnical components, we borrowed ideas from components that had been tested on platforms previously designed in our lab. Our platform has now been tested on a variety of terrains including a rock yard in a construction site and rolling hills at a field on our campus.

1.1. Mechanical Design

In Figure 1 and 2, we present the originally proposed mechanical structure and the current platform, for comparison. The ENVOY2's design evolution was inspired by testing, which focused on efficiency and robustness. Initially, ENVOY2 was designed to have six-wheel drive-train for an advantage when climbing over difficult terrains. The simple six-wheel drive-train only uses the two middle wheels if stuck on an obstacle. Through testing, we realized that decreasing the wheel base and increasing the diameter of ENVOY2 was sufficiently resistant to becoming stuck in bumpy terrains. The tradeoff in wheels also decreased weight and power drain.

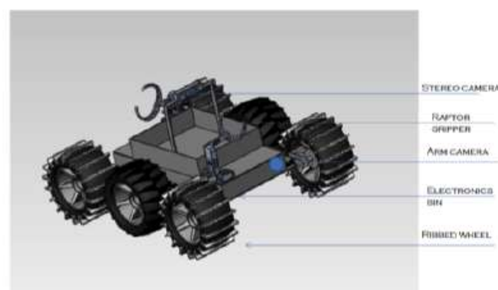

Figure 1. Originally proposed mechanical design.

1.2. Software Design and Sensors

All actuators on the robot are controlled by one software node. Figure 3 and 4 show more details of the framework and illustrate the modular approach to software and hardware integration using ROS.

Various sensors are installed on the platform. In a nutshell, driving and scooping objects are driven by seven servo motors and four DC brushless motors; multiple sensors filter this information on-board in an Extended Kalman filter (EKF) to determine an accurate pose estimation; a web-camera and stereo vision rig capture colored images of the field for autonomous computer vision algorithms in visual odometry and localization, and for streaming images back from the robot on the field to the GUI at the mission control center (MCC).

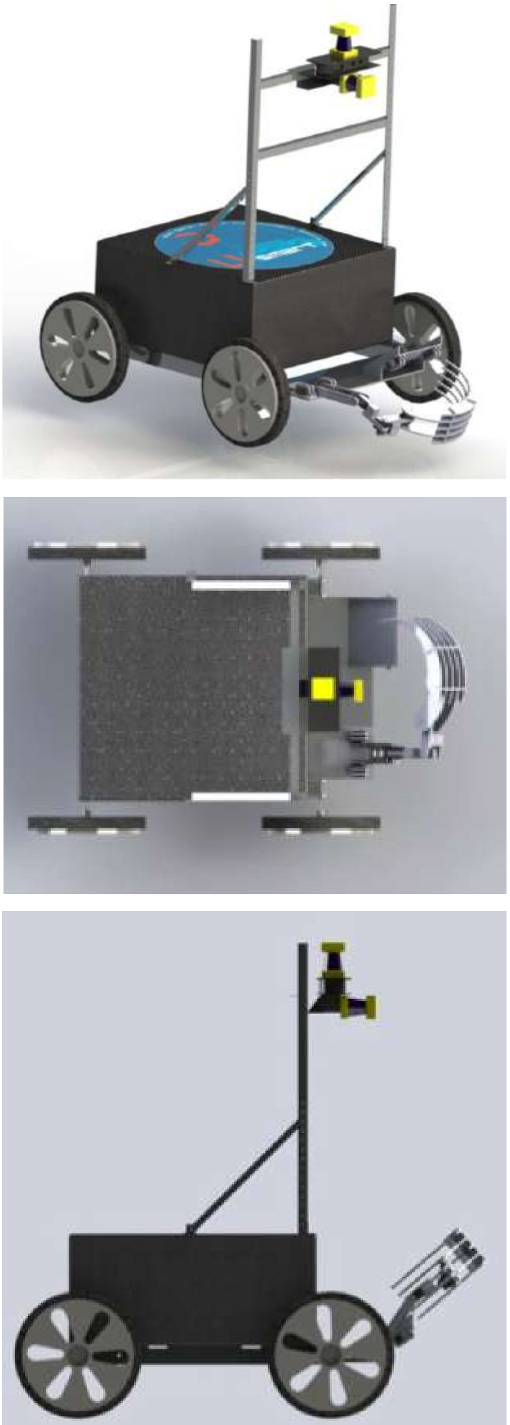

Figure 2. Perspective, top, and side views of robot.

1.3. Production and Testing Approach

In order to ramp up development while building our current platform, we have been using a platform previously developed in our lab to test our ROS framework, includ-

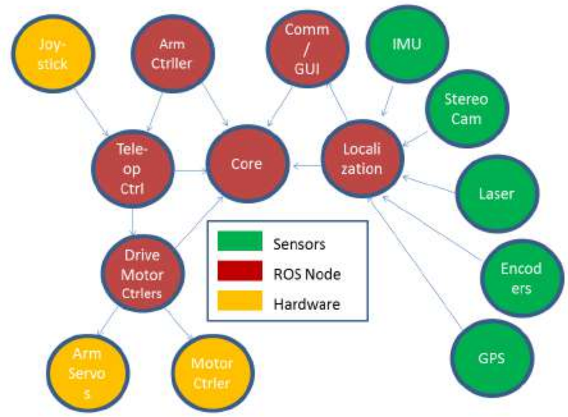

Figure 3. System framework diagram. Best viewed in color. Green: sensors; red: ROS nodes; yellow: hardware components.

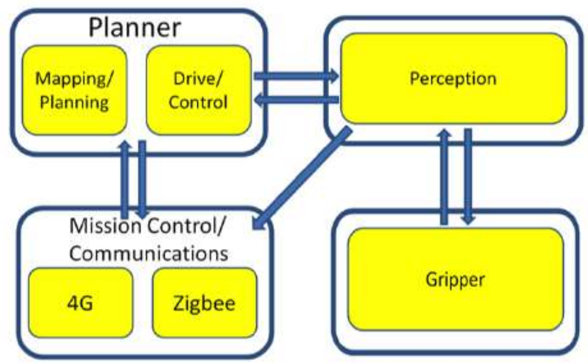

Figure 4. System modules

ing the navigation, mapping, object recognition packages, and the message passing communication among them. We have also tested our communications in ROS, by implementing a package that allows us to remotely control the video streamed from different cameras on the platform. In fact, our data transmission was tested at the Johnson Space Center before the competition.

2. Physical Components

2.1. Structure

Our drive-train is composed of 1/4 inch thick aluminum alloy. The drive chain, sprockets, and axle sit between two aluminum rails 22 inches long. Our DC motors are designed to mounted by cantilever.

2.2. Stress Analysis

As our design progressed, our original six-wheel design has evolved into four larger wheels. We performed stress analysis on the wheels, and the results are presented in Figure 5. As shown, the stress level is very low on the robot chassis, and the highest is around the motors, which is expected.

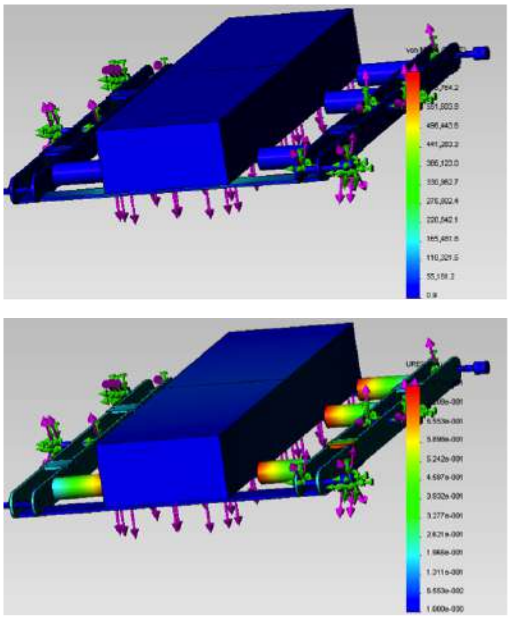

Figure 5. Top: static nodal stress analysis. Red: high stress, 606 764.2; dark blue: low stress, 0.9; interval 55 161.2. Bottom: static displacement analysis. Red: 0.766 4; dark blue: 0.001; interval $\sim$ 0.065. Best viewed in color.

2.3. Drive and Actuator Control

On our four-wheel drive, we utilize four brushless DC motors in a tank-style skid-drive system. Each wheel is driven by a 1:1 ratio chain. We chain the two idler wheel shafts on each side together. Although this reduces the efficiency of the motors, we now utilize the strength of all four wheels when climbing obstacles and guarantee that the wheels on either side are traveling at the same speed at all times.

A Pololu Maestro Servo Board controls all the actuators on the robot. The board receives serial commands from an implemented ROS driver to set the rate of acceleration change, acceleration, and target velocity. All twelve channels are configured to output R/C signals. The first 8 channels control high-end hobby-style servos. Two of the channels control both wheels on either side of the robot. Each drive-channel connects to a y-cable with two outputs to a SyRen 10A Regenerative Motor Control board. Each of the boards is connected to one motor and allows the recovery of charge when the wheel is braking. This is useful for circumstances where power is limited.

We use 4 high-torque Brush 24 V DC with gearbox. Each of these motors has an optical encoder directly attached to the shaft tail.

2.4. Manipulation and Robot Arm

The remaining eight channels on the Servo Controller command one servo each. The servo features an aluminum heat-sink case, four Titanium Gears, 180 degree rotation, a wide operating voltage (3.3-7.4 V), a Stall torque 333.29-416.61 oz-in, Current drain (3mA), Dual Ball Bearings MR106, 0.17s/60 degrees max speed, and 2.18oz weight.

ENVOY2's initial arm was borrowed from ENVOY1. This was a reliable unit with low power consumption. However, the manipulator evolved several times since its first rendition. Despite its advantages, the initial design required a skilled tele-operator to reliably pick up an object. One arm required a "picking motion"- the end effector must move downwards upon the object before grasping it. However, testing showed that a "scooping motion", in which the end effector brushes an object into a container, offers several advantages. Scooping allows for objects to sit in a wider area in front of the robot and still be picked up, rather than requiring the robot to drive to an exact location with multiple attempts to command the arm into the correct position. Furthermore, the scooping action is one single autonomous motion as opposed to a carefully human operated coordination of seven motors.

2.4.1 Design Timeline

Individual Arms Our first arm designs focused on a claw structure that would position above the object and pick it up, much as a human hand would (Figure 6). This design required six servos to power the manipulator and a human to operate the arm (Figure 7).

Single Arm and Scooper In the next design, we completely changed the method of retrieving an object. The arm link was modified to swing horizontally and scoop the object into a ramp. The ramp would then lift the object into the robot's storage. However, through testing, we realized that additional modifications were necessary.

Single Arm and Rake Platform Finally, the current design removes a redundant link in the arm. Moreover, the lip of the ramp is changed into a rake structure (Figure 8). This allows for objects to slide into the ramp rather than risking being caught on the edge, as in the previous design. The added steel increases robustness to cope with uneven terrains.

2.4.2 Manipulation Algorithm

Our code base is written entirely in ROS. The design of our manipulator emphasizes robust and innovative mechanical design in order to simplify its controlling software. There are a fixed number of preset motions that the arm and the ramp can cycle through. In manual mode, the tele-operator commands an action such as "scoop object," with a joystick

Figure 6. Old arm from ENVOY1.

Figure 7. New arm servo.

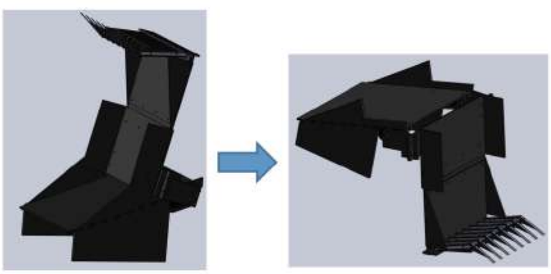

Figure 8. New scooping ramp.

via the communication GUI, which is translated into servo motion, i.e. the presets of the manipulator and ramp. Figure 9 summarizes the arm subsystem algorithm.

The design of our scooping mechanism relies on the fact that the object recognition on board moves the robot to a coordinate in front of the object, close enough for the object to be within reach of the scooper manipulator.

Once in position, the manipulator is commanded to go through a series of preset motions, the first being lowering of the scooper manipulator and ramp onto the ground. Each motion might involve one or more frames or discrete sets of servo motor positions. These frames are run through sequentially until the manipulator is in the desired position for that preset. The servo frequencies for each of the eight motors are encoded in a YAML file to allow quick and clean modifications throughout the development phase.

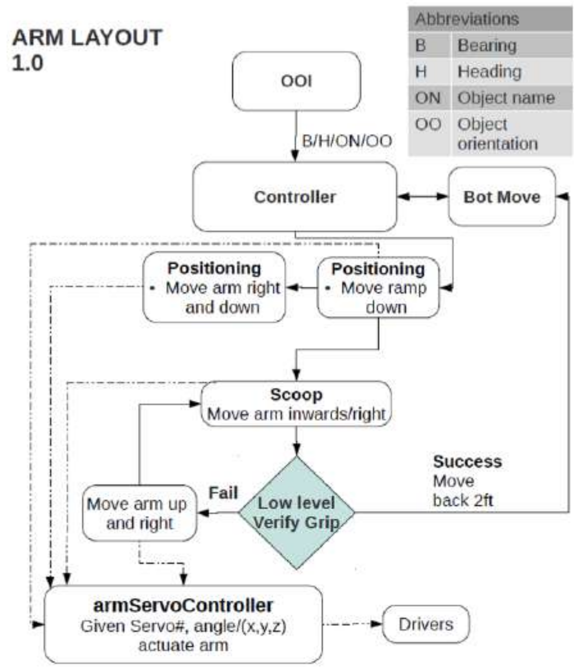

Figure 9. Arm software diagram.

Two ROS nodes control the motion of the manipulator; A tele-operated joystick ROS node publishes the motion that the manipulator needs to be taken through, for example, “prepare to scoop”. The controller node listens to this topic, picks the preset motion command that needs to be sent to the manipulator, and publishes it.

A configuration reader node listens for the preset motion command and reads the appropriate set of the frames from the aforementioned YAML file. Each frame has with it an associated delay which is approximately the time that it takes the hardware to complete the movement. This is incorporated into software as a delay between frame commands sent to the servo drivers.

2.5. Electronics

Our electronics board is mounted on a 8-inch by 18-inch Acrylonitrile butadiene styrene (ABS) sheet that can be easily pulled from the bin for repair or alteration (Figure 10). This board also handles voltage regulation to satisfy the power supply presented in Table 1. Two microcontrollers manage data from the encoders and Inertial measurement unit (IMU). These also manage the lighting on the robot. For processing, we use a 2012 Mac Mini at 2.5GHz and 2GB DDR3 RAM.

3. Software Components

We provide two modes to operate the robot; one using direct remote control, the other using vision and odometry

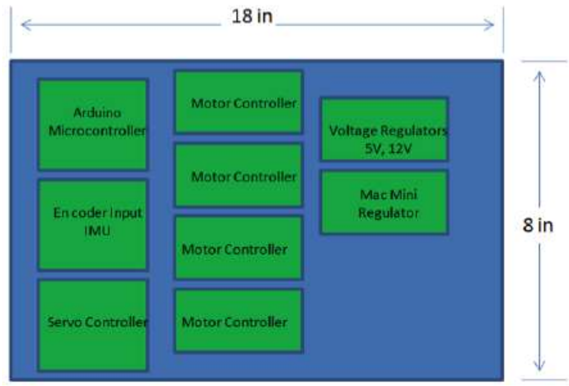

Figure 10. Electronics board.

Device	Power Supply
Hokuyo UTM 30Lx	12 V
Mac Mini	13.5 V
Encoders	5 V
Pololu Microncontroller	12 V
Logitech Onboard Cameras (2)	10 V (USB powered)
Ublox 6 GPS Engine	3.6 V (USB powered)
Verizon Broadband Card	5 V (USB powered)
IMU	2.5 V (microcontroller powered)

Table 1. Electronics power draw breakdown.

for autonomous navigation. Here we discuss the communication channel we chose to transmit data from the rock field back to the MCC, and the mapping algorithms using machine perception techniques.

3.1. Communications and Tele-Operation GUI

The Clear 4G Data Plan supports 150 KBps. Utilizing ROS compression algorithms, ENVOY2 is able to stream full color video at 640x480@30fps in real time using about 70-80KBps. The mission controller has the ability to dynamically reduce video stream quality down to 10KBps. This is extremely useful for tele-operation and areas where the data signal is low. Our GUI (Figure 12), built using Qt, allows dynamically managing message subscriptions to any of the video cameras and other sensors on the rover in order to conserve bandwidth. Figure 11 shows the previous interface, upon which our new interface will improve.

Through this 4G communication module, camera views are streamed from the robot over back to the GUI interface operated from the MCC. A human operator is then able to observe the camera images and issue input to tele-operate the robot, point to a destination for the robot to drive to using vision algorithms, or control the robot directly via a joystick. The user command is then fed back to the robot through the 4G network.

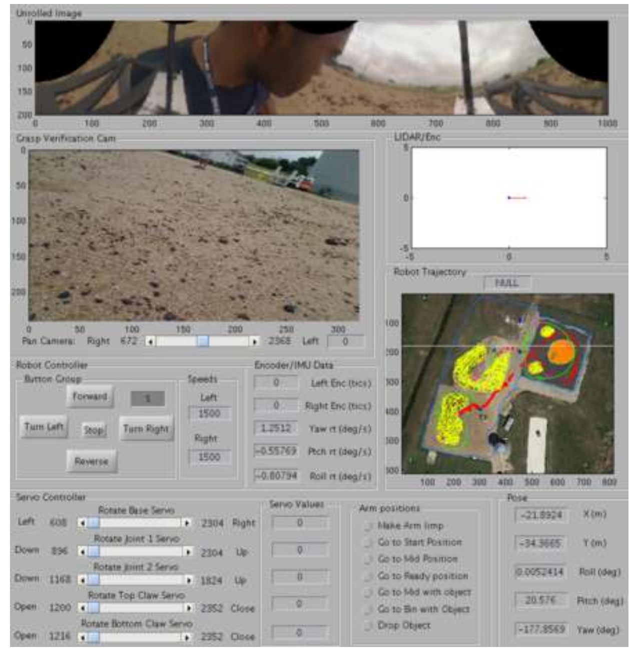

Figure 11. Our previous GUI in MATLAB.

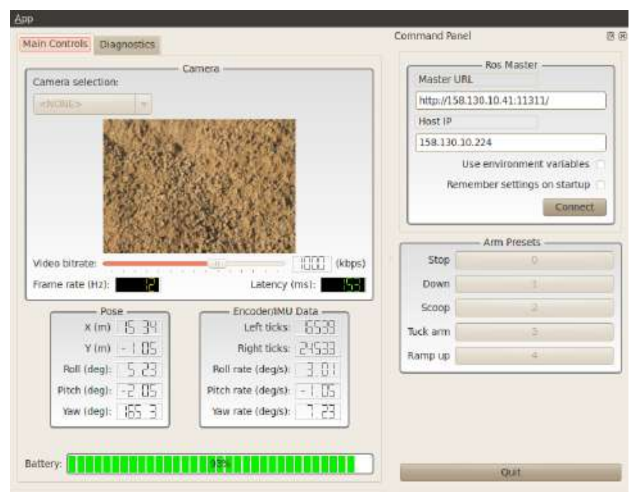

Figure 12. Our MCC GUI provides ability to dynamically control what data the robot streams to the MCC over the 4G network.

3.2. Object Recognition and Odometry

Using a range finder, we tested the vision algorithm for 3D reconstruction at a field on campus, we were able to produce 3D maps of the entire field area that the robot navigated, much like the elevation map in [1]. This map is helpful in odometry and planning the autonomous routes of the robot.

Along with the imagery used for planning, we also have stereo cameras on board the robot to observe objects nearby in fine-grained detail, so that the robot can recognize objects using vision algorithms to decide whether to execute arm motion for picking up an object.

3.2.1 Localization

Compared to ENVOY1, ENVOY2 consists of improved components, which makes the localization task more robust, enabling a tele-operator to navigate around the field much more efficiently.

Since the cameras only give us what the robot sees in its heading direction, it can be difficult for the tele-operator to know where the robot is in the field with respect to where it started. To solve this, we have 2 lasers, a stationary horizontal one and a tilting vertical one, which are used to create a 2D as well as a sparse 3D map of the environment as the robot moves around. This allows the tele-operator to localize the robot in the field of play and to plan how to move around and explore.

Some results from the tests can be seen in Figures 13, 14, and 15. In Figure 13, we see a sample 2D map of the environment created by the robot. In Figure 14 we see an instance when the robot is in a certain area of the created map. Evidently, when the robot is looking at a corner, the camera stream would not be very useful for the tele-operator. The map gives a much better idea of how to plan going into particular, possibly unexplored, areas of the environment.

Another helpful feature is the sparse 3D mapping. Information about obstacles not captured by the horizontal laser can be observed from the 3D map. This can be computationally expensive, so it is not the primary source of information about the environment, but it can be useful in case the tele-operator is checking for pits or uneven terrain and wishes to move around them. This information would not be available from the 2D map. A sample result from outdoor tests can be seen in Figure 15.

3.2.2 Object Detection and Recognition

The object detection module is another subsystem to aid the human in the loop (HIL) at the MCC. The two classifier methods, described below, employed give us some possible object locations overlaid on the camera stream and help focus attention to limited areas of interest.

The first method uses a fast image segmentation based on hamming codes. Since our target objects are multi-colored rocks, these are easily segmented and separated from the background. This is then highlighted for the tele-operator to make a possible move towards the object. Results from segmentation of an outdoor scene can be seen in Figure 16.

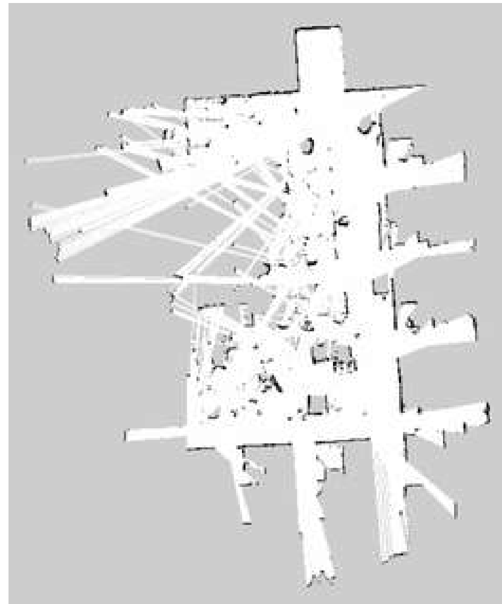

Figure 13. 2D map of the environment created by the robot.

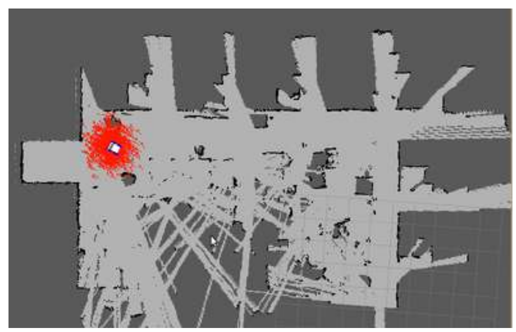

Figure 14. Indication of the location of the robot, calculated by odometry and localization. Robot's positions is in red. Best viewed in color.

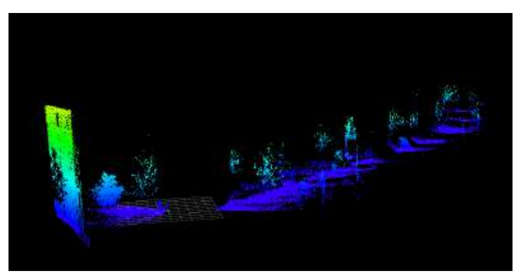

Figure 15. Model of the environment captured by sparse 3D mapping, which provides information not present in the horizontal scanner that models the 2D environment.

Our second method fits a color or shape model onto the edge map of the image frame. This provides object hypothesis and the HIL can then easily filter a few matches rather than scanning the entire image. To save on bandwidth and computation, these hypothesis locations are transmitted independently from images and overlaid by GUI at the MCC.

Robust object detection techniques like template matching have also been attempted. These would allow the robot to work autonomously to pick up objects because the object tracking would be substantially more accurate. However, these algorithms are still being developed.

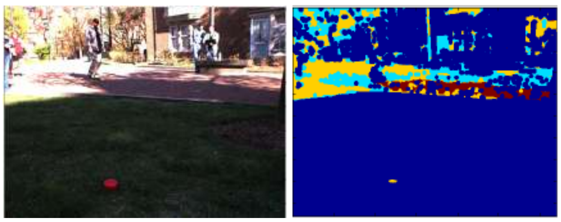

Figure 16. Left: a photograph of the robot's view. Right: Segmented version, where the target object (a hockey puck) is clearly marked in the lower part of the image. This indicates to the teleoperator of where to move.

3.3. Marker Tracking and Mapping

In our original design, we planned for an off-board dispatch station or in-air observatory component that would assist in the mapping and localization of the robot on the terrain. Due to the height constraint and the vertical actuation of the sensor mount, we were not able to include the module on board the robot for recognition. Even though the contribution of this module is limited in this task, we continued the development for possible future tasks, where the robot could drop off a small camera rig on the field at departure, and it would observe the robot from the departing point.

To localize the robot on the field, we trained images of special black and white markers, which may be attached onto the sides or top of the robot for ground or aerial detection. We built a stereo rig which consists of two GoPro 2 cameras, each with a Eye-Fi SD card for storage and wi-fi transmission of the photos to the robot. The two cameras are connected to a microcontroller that enables shooting control from a ROS node. The delay between taking a photo and transmitting on board is approximately 5 seconds. The GoPro cameras are calibrated using the OcamCalib Toolbox [6, 7, 5].

After we obtain one image from each camera, forming a stereo pair, we rectify each image using the camera intrinsic matrix from the calibration, then detect a marker using the ARToolkit [3], project the camera images into 3D, triangulate the points, and calculate the 3D coordinates of the marker. The coordinates are then sent to the mapping and planning module to determine the next motions of the robot. Most of the vision algorithms were implemented using OpenCV [2].

4. Educational Outreach

Our goal for community outreach is to promote the STEM (Science, Technology, Engineering, and Mathematics) fields, as the number of engineers in the country will be in severely short supply in a few decades. Robotics and space technology are great areas to promote STEM, as they incorporate various engineering disciplines, mathematics, and natural sciences. In alignment with this goal, we organized and joined activities that would help more sectors of the population become aware and excited about the possibilities in the areas of space and robotics, especially for younger students to start early and to consider it as a future pursuit.

This year, our outreach audience has a large diversity in demographics, ranging from undergrads; high school, middle school, and elementary school students; to Philadelphia families of all ages. Expanding to include more demographics is important for all in the community to become more supportive of future generations' involvement in space and robotics.

USMART wants to be clear that while the GRASP Laboratory has an office for outreach coordination, our events are separate or on top of these events. We seek to leverage the connections and experiences of our office, while branching out and having a USMART specific impact. We list some of our exemplar events below to summarize our efforts.

4.1. National Robotics Week GRASP Open House

In this annual event, where students from local schools are selected through an application process to come to Penn GRASP lab for a day of exciting lab demos, activities, and presentations, we offered two out of ten lab demonstrations and activities, and hosted two presentations for high school and middle school students.

A total of 200 students attended the day. During our two lab activities, we demonstrated robot drive at one and hosted a design activity at the other, where we asked the middle school students, if an asteroid is colliding with the earth soon, what robot would they design to avoid the asteroid collision. Students were asked to work in a group of around 5 to come up with a design and draw it on paper in 20 minutes. In total, we hosted around 125 students, rotating in 5 groups of 25. The groups came up with various prevention or avoidance techniques and presented innovative drawings. We explained the gravitational asteroid deflection solution as a possible countermeasure at the end of the sessions. Figure 17 shows three of our team members hosting the robot demo, helping students to drive one of the existing robot platforms that we used to test our vision and mapping algorithms.

During the two presentations, which all 200 students attended, we presented about the team, the RASC-AL Robo-Ops competition, recent rovers launched by NASA, and space and robotics technologies in general. We received very positive feedbacks at the end of the day, with numerous students reportedly wanted to become astronauts and roboticists, some even both.

Figure 17. Three USMART team members showing our vision and mapping test platform to middle school students at the National Robotics Week GRASP Open House. Each of our two lab activities hosted about 125 students, rotating at 5 groups of 25.

4.2. Philadelphia Science Festival

At the second annual Philadelphia Science Festival Carnival Day, we reserved our own "GRASP USMART: Robots in Outer Space" booth and exhibited a full day among more than 100 booths at the Franklin Parkway for families of all backgrounds and ages to enjoy. Our booth was received with welcoming interest, as we let children and adults drive around the robot in front of our booth and explained the arm and sensors on board the robot.

In addition to robot remote controlled driving, we also provided robot puzzles, coloring activities, distributed USMART promotional items, and explained the competition to the visitors. Our booth was almost always full with children coloring robot puzzles, and the area in front of our booth was always crowded with families lining up to drive the robot and watching it roaming around. Figures 18, 19, 20 are a few of the many moments from the day, where children and adult alike were excited about space and robotics.

4.3. NASA and Space Undergraduate Space and Robotics Day

We organized a new event this year, entirely run by our team, to reach out to undergraduate college students in the Philadelphia region and to expose them to opportunities in space or robotics as a career. The day started with an introduction of the team and the RASC-AL Robo-Ops competition, followed by a presentation by our faculty advisor

Figure 18. At Philadelphia Science Festival, one of our team members instructed the children on how to drive the robots, "Have you played video games with joysticks before?"

on his past projects in space-related robotics. Then a graduate panel comprised of students with experience working for NASA and JPL, where they discussed unique experiences and took audience questions. After this, we hosted a GRASP Robotics Lab tour, where students were shown various robots and cutting-edge developments in the lab. Finally, the day ended with a LEGO paint ball catapult robot competition, where several undergraduate teams competed and tested their catapult robots outdoors. Feedback surveys showed attendees almost unanimously reported that the day was "awesome", out of several multiple choices ranging from positive to negative, and that they would definitely like to see something similar again. Figure 21 shows a shot from the LEGO paint ball catapult competition, Figure 22 shows the panelists discussing exciting experiences at the graduate student panel.

4.4. Upper Bound Math and Science SeaPerch Building

We set up SeaPerch build sessions with the Upper Bound Math and Science, a college preparation program that three most dangerous challenged high schools in Philadelphia participate in and come to Penn for programs several times each week. Our team provided two Saturdays of mentoring and assistance with the high school students to build a SeaPerch underwater Remotely Operated Vehicle (ROV).

Even though initially, the students were not sure what SeaPerch was or how they would like to participate, at the end of the day, they became experts in soldering, stripping wires, and working better in teams. Almost all the students who participated the first day chose to come again the second Saturday to finish the robot. Figure 23 shows one of our team members teaching the students to solder the control board.

Figure 19. At Philadelphia Science Festival, our booth "GRASP USMART: Robots in Outer Space" stole everyone's attention from the festival.

4.5. FIRST LEGO League Regionals

During the annual FIRST LEGO League Regionals held at Penn, USMART team members served as robot judges, Core Values judges, and referees for around 50 competing schools and groups selected from the qualifiers. Figures 24, 25 show one of the all-girl FLL teams at work, and some of our team members as judges.

Figure 20. At Philadelphia Science Festival, the USMART booth was always full of children coloring robot puzzles and gathering around the robot.

Figure 21. At the Undergraduate Space and Robotics Day entirely run by our team, two undergraduate teams were preparing to launch the LEGO robot they built with two hours of hard work for catapulting paintballs into a bullseye.

Figure 22. At the Undergraduate Space and Robotics Day entirely run by our team, we hosted a graduate students panel where students spoke about their past experiences working for NASA and JPL.

5. Acknowledgements

We would like to thank GRASP Lab, NASA RASC-AL, Qualcomm, Google, Nextfab, and Weiss Tech House

Figure 23. On a Saturday with high school students from the Upper Bound Math and Science program, our team members mentored the students in each of the three core components in building the SeaPerch ROV - electrical, mechanical, and waterproofing the motor. Students became experts after the first day.

Figure 24. At FIRST LEGO League regionals, USMART team members served as robot judges, Core Values judges, and referees for around 50 teams, among which several Girl Scouts teams, one of which won the regionals to head to the championship competition.

for their support of USMART and the development of ENVOY2. Our sincere gratitude goes to Dr. Kostas Daniilidis, who has been very supportive throughout the build process, and Alex Kushuylev, for his advice and hardware assistance. In addition, we are grateful for Peter Szczesniak, who helped us with hardware fabrication. Other individuals whose support and assistance have been crucial to the suc-

Figure 25. At FIRST LEGO League regionals, all the judges with the winning team. Some of our team members were among the judges in red.

cess of this project and our outreach activities include Dr. Daniel Lee and Rebecca Stein.

References

[1] J. Bares, M. Hebert, T. Kanade, E. Krotkov, T. Mitchell, R. Simmons, and W. Whittaker. Ambler: An autonomous rover for planetary exploration. *IEEE Computer*, 22(6):18–26, June 1989. 5

[2] G. Bradski. The opencv library http://opencv.willowgarage.com/wiki/. *Dr. Dobb's Journal of Software Tools*, 25(11):120, 122–125, Nov. 2000. 7

[3] H. Kato, M. Billinghurst, I. Poupyrev, K. Imamoto, and K. Tachibana. Virtual object manipulation on a table-top ar environment. In *International Symposium on Augmented Reality (ISAR)*, pages 111–119, Munich, Germany, 2000. 7

[4] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. B. Foote, J. Leibs, R. Wheeler, and A. Y. Ng. Ros: an open-source robot operating system. In *International Conference on Robotics and Automation*, Open-Source Software workshop, 2009. 1

[5] M. Rufli, D. Scaramuzza, and R. Siegwart. Automatic detection of checkerboards on blurred and distorted images. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2008)*, Nice, France, Sept. 2008. 7

[6] D. Scaramuzza, A. Martinelli, and R. Siegwart. A flexible technique for accurate omnidirectional camera calibration and structure from motion. In *IEEE International Conference of Vision Systems (ICVS)*, New York, Jan. 5-7 2006. 7

[7] D. Scaramuzza, A. Martinelli, and R. Siegwart. A toolbox for easy calibrating omnidirectional cameras. In *IEEE International Conference on Intelligent Robots and Systems (IROS 2006)*, Beijing, China, Oct 7-15 2006. 7