Faculty Advisor Statement

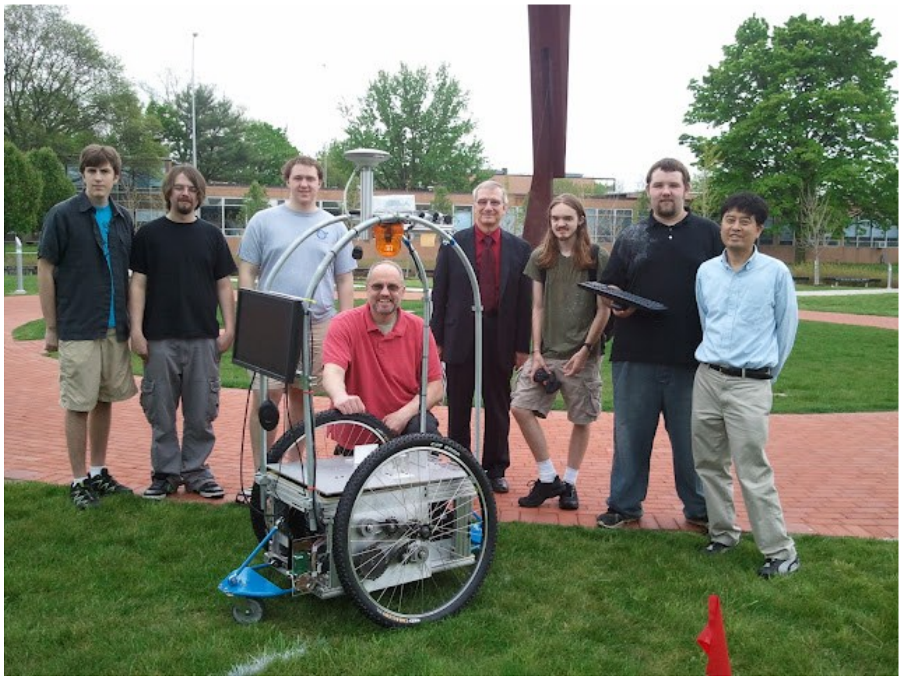
I, Dr. CJ Chung of the Department of Math and Computer Science at Lawrence Technological University, certify that the design and development on the vuLTUre 2 platform by the individuals on the design team is significant and is either for-credit or equivalent to what might be awarded credit in a senior design course.

Signature Date

The vuLTUre 2 platform is an improved design evolution of previous robots. Subsystems that are significantly different from those used in previous years include:

- Mechanical design, including
  - New design motor plates with 7^0 camber angle
  - Increased ground clearance
  - Revised suspension
  - Improved electronics packaging.
- Electrical design, including
  - Computer systems
  - Camera sensors
  - Energy efficient monitor
- Software design, including
  - Stereo vision system
  - Vision processing
  - Path planning
  - User interface
  - Programming language Java to C#
  - Integration process

vuLTUre 2



Design Report

for the

2012 Intelligent Ground Vehicle Competition

prepared by

Lawrence Technological University

vuLTUre 2 Design Team

www.robofest.net/igvc

1. Introduction

This report describes the vuLTURe 2 robotic platform, designed and built for the 2012 Intelligent Ground Vehicle Competition (IGVC). This report is organized into sections describing the design team, the design processes, and various aspects of the design, followed by performance and cost information.

2. Team Organization

The vuLTURe 2 design team is comprised of the following members:

Name	Course of Study	Role(s)
Paul Wright	BSTM	Team Captain, Mech. designer
Ryan Matthews	BSCS	Software Team Leader
Jonathan Ruszala	MSCS	Software and Electrical
Daniel Anderegg	BSCS	Software
Dan McGee	BSEE	Software
Jonathan Nabozny	BSRE	Software
Taiga Sato	MSCS	Software
Christopher Kawatsu	MSCS	Software, Stereo Vision
Jiaxing Li	MSCS	Software, Stereo Vision
Stephen Osterhoff	BSBME	Mechanical
Anthony Knapp	BSRE	Mechanical
Philip Bigos	BSRE	Mechanical
Dr. CJ Chung	PhD, CS	Faculty Advisor

3. Concept Development

A post-competition review of the 2011 IGVC results identified the areas in which our vuLTURe entry distinguished itself, as well as those areas in which improvement was needed. When the vuLTURe 2 team was formed for the 2012 competition, a number of 'brainstorming' sessions were held in which each of the problem areas was considered and possible solutions discussed. Our strength areas were also examined for opportunities for improvement. The results of these discussions and analysis are the basis for the innovative features found in the vuLTURe 2 platform.

3.1 Failure Analysis

The following areas were identified as needing improvement during failure analysis. These received special attention in the design of the platform:

- **Suspension** – Large forces caused deformations of components of the stabilization system. While no catastrophic system failures were attributed to this, a more robust design was needed.

- **Vision Processing Software** – Our vision processing software was unable to clearly distinguish the white markings on the grass of the field, particularly in direct sunlight.
- **Electrical Power** – We ran low on power for the non-propulsion electrical subsystems on a number of occasions, despite continuous charging of batteries during our downtime.
- **Clearance** – Our platform experienced difficulties in maneuvering over "speed bumps". This was traced back to an inadequate ground clearance.
- **Tire Slip** – The considerable forces involved in the acceleration of the platform caused the tires to slip in low *mu* conditions.

3.2 Identified Strengths

The following areas were considered strengths of our previous entries such as Culture Shock and vuLTUre with two large wheels.

- **Symmetric, Low-Moment Design –** The stabilized two-wheel design has proven to be superior in many aspects, including stability and maneuverability.
- **Large Drive Wheels** – The large drive wheels provide platform shock isolation, and provide a larger contact patch than similar entries with smaller wheels.
- **Low Center of Gravity** – A low center of gravity provides stability during maneuvering and stopping. Furthermore, having the center of gravity below the wheel axles makes our platform inherently stable, as the low-energy state is the normal upright position.
- **Vision-Based Obstacle Detection** – The stereo cameras used for obstacle detection is deemed superior when evaluated with other technologies in the areas of size, weight, power, and cost (SwaP-C). Due to their solid-state construction, they also have superior performance with regards to mean time between failures.
- **DC-DC Electrical Conversion** – Using direct DC-DC conversion for the electrical systems decreases complexity, parts count, and power consumption (as opposed to the use of a DC-AC converter coupled with AC-DC power 'adapters').

3.3 vuLTUre 2 Design Goals

The vuLTUre 2 platform represents an evolution of the vuLTUre platform, with improvements and refinements to nearly every subsystem, as well as new features to accommodate 2012 IGVC rule changes. The following summarizes the design goals for the platform, and highlights the innovations and improvements that these goals represent over previous years' entries.

- **Compact Size –** By minimizing the footprint of the platform, there is more room to maneuver within the competition area. A hard limit of 34" inches is imposed on the width, to ensure that the platform can be maneuvered easily and without modification through standard-sized doorways.
- **Large-Diameter Wheels –** The use of large-diameter wheels allows the center of gravity to be positioned below the wheel axles, which greatly enhances stability. The relatively large volume of air

contained within the tires provides some measure of shock isolation. Finally, the large-diameter wheels provide a larger contact patch than similar wheels of smaller diameter.

- **Wide Stance** – The stance of the platform is maximized within the width constraints by angling the wheels. This provides greater stability on turns and inclines. This represents an innovative change to the design over previous years' entries.
- **Low Center of Gravity** – The center of gravity of the platform is placed as low as possible while maintaining minimum ground clearance. This lends to platform stability during maneuvers and while on inclines.
- **Low Moment of Inertia** – Placing dense components close to the vertical axis enhances maneuverability, as less force is required to change platform direction.
- **Backlash Management** – Backlash is minimized through the introduction of chain tensioners in the drive train. These maintain the tension in the chain and absorb sudden changes in force, reducing wear and tear on chassis and drive train components while aiding in accurate dead reckoning. Software improvements have minimized excess torque on startup by slowly ramping up the application of power.
- **Swappable Batteries** – Incorporate swappable battery banks into the design, allowing banks to be charged while platform is in use. All battery banks are identical.
- **Stereo Vision-Based Obstacle Detection** – Develop our own stereo vision system to sense the immediate environment. Develop and implement vision-processing software to identify and categorize obstacles.
- **Ease of Maintenance** – Minimize the use of custom-designed and custom-built components to aid in ease of maintenance. Where custom components are necessary, make and keep spares on hand.
- **Prefer Solid-State Components** – As the platform is mobile and can potentially be subject to shock, solid-state components will provide greater reliability. For example, choose cameras over a laser measurement device, and use solid state hard drives in the computational components.
- **Detect Battery Voltage Levels** – Incorporate voltage sensors to detect low-battery conditions. This represents an innovative change to the design over previous years' entries.
- **Improved E-stop** – Simplify the E-Stop electronics, and improve E-Stop reliability.
- **Increased Top Speed** – 2012 IGVC rules call for a top speed of 10 mph. The platform should be able to quickly reach and maintain this speed over level ground, as well as accommodate 15% gradients at a minimum of 8 mph.
- **JAUS Conformance** – Implement the JAUS features described in the 2012 IGVC rules document.

4. vuLTUre 2 Design

Platform design was divided into three main categories, or groups: mechanical, electrical, and software. Individuals were assigned to one or more of these groups based on their experience and areas of expertise. Beginning in December 2011, twice-weekly meetings were held to track status, as well as aid in collaboration and information exchange between groups. Each individual was responsible for tracking hours worked. The following sections detail the development of the various design aspects.

4.1 Mechanical Design & Fabrication

4.1.1 CAD & Modeling

SolidWorks® was used for capturing the mechanical design of the platform. This selection was based on the team members' experience with the tool, and its availability to students. An Internet "drop box" was used to share and update files among the group members during development. During fabrication, 2D prints and dimensions exported from SolidWorks® were used for shaping components.

4.1.2 Drive Train

Major drive train components such as motors and gears were reused from last year. However, extensive redesign for improved performance was implemented requiring new fabrication of related components.

Figure 1 – Revised drivetrain and suspension

4.1.3 Wheel Mounting

Field testing showed the previous 10 degrees of wheel camber reduced tire contact patch enough to reduce traction under some conditions. A test comparison of 10°, 7°, and 5° camber angles found that the angle of 7° was the ideal compromise between stability and traction. Other design parameters were optimized to increase ground clearance to 6", reduce track width to 32" while maintaining a low center of gravity. The previous 36 spoke wheel, custom hub and 18-tooth sprocket assemblies were carried over from last year.

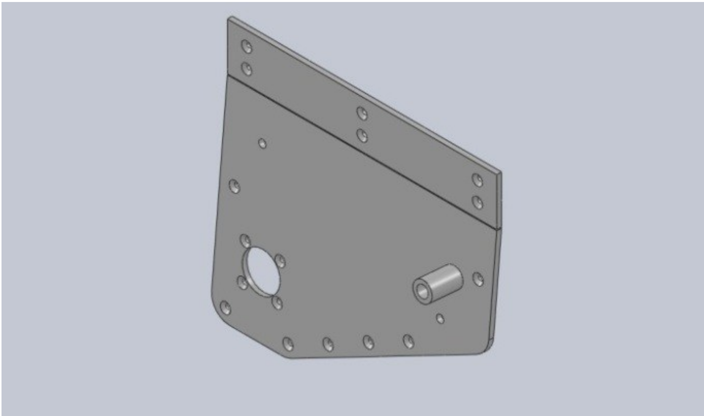


Figure 2 A new ¼” 6061-T6 aluminum mounting plate was designed to provide the revised 7° wheel camber angle, and increase ground clearance to 6”. Plates are asymmetrical and mount either left or right side.

4.1.4 Chassis

Previous battery configuration was retained. The configuration selected supports a swappable battery module concept while keeping the moment about the vertical axis to a minimum. Battery modules are identical; allowing one or more “hot” spares to be charged while the platform is being used. Battery modules can be easily removed without tools for transport – the platform without battery modules weighs a manageable 100 pounds.

Structural “t-slot” aluminum is used for carriage construction. A reinforced box configuration uses aluminum plate to provide rigidity and protection from the elements. Standard ¼-20 socket-head fasteners are the primary means of assembly; with regular maintenance components secured using thumb knobs. Plates are mounted with flush fasteners to maximize clearance (ground clearance has been improved to 6” when traveling over soft ground). Casters are attached to the carriage using a robust rocker-arm system incorporating adjustable coil-over shock absorbers. These casters keep the platform upright during acceleration and while negotiating inclines. Outrigger suspension design was extensively revised and improved for 2012. New king-pin-less casters were custom made using modified 5” x 3” mower deck wheels for improved performance on the grassy IGVC course.

Critical electronics are housed inside the chassis structure, protecting them from the elements. Releasing two spring-loaded latches allows the top to swing up and out of the way for easy access to electronics, motors and other components. A tubular super-structure provides a flexible framework for mounting sensors, controls, and splash guards (required in wet conditions, as the cambered wheels cast off water toward the chassis). During challenge runs, the payload is secured to the top of the chassis structure with tabs designed for the purpose.

4.2 Electrical Subsystems Design

The electrical design is comprised of two largely independent subsystems. The 24V subsystem powers the drive motors and motor controller. The 12V subsystem powers the processors, sensors, and controls.

Each battery module has two connectors, one for each battery. These connectors are mated to both the charging system and the chassis. For the 24V subsystem, circuitry on the chassis connects the batteries in series. For the 12V subsystem, circuitry on the chassis connects the batteries in parallel. The two subsystems share a common electrical ground reference.

4.2.1 Motors & Motor Controller

The motors are a pair of Midwest Motion Products model D33-455H-24V planetary gear motors, with a rated peak torque of 1062 in-lbs, and a continuous torque of 90 in-lbs at 5.1 amps. A 100 pulse-per-revolution encoder is affixed to the armature shaft. When combined with the 19.2:1 planetary gearing, this provides a detection capability of approximately 2mm over ground. The motor controller is a RoboteQ model AX3500, capable of providing 60 amps of current to each of the motors. The controller is configured for analog input control on reset. Since there is nothing attached to the analog control inputs, a voltage divider internal to the controller keeps the voltage centered, which in turn causes the motors to be actively held steady. This capability is used to satisfy the E-stop requirements, as simply resetting the motor controller will cause the platform to come to a safe, controlled stop. A specific sequence of commands over the serial (RS-232) interface is required to switch the controller into serial operation mode. The controller has been pre-configured to use a closed-loop control while in the serial mode of operation, with a built-in top speed, configured for 10mph in this competition.

4.2.2 E-Stop

The wireless receiver has a range of over 150 ft, and can be paired with multiple wireless transmitter fobs. It incorporates a relay. The "normally closed" circuit of the relay is opened momentarily when the receiver is activated via wireless transmitter. During normal operation, both the Manual E-stop switch and the wireless transmitter circuits are closed, causing the (active-low) reset input on the motor controller to be held high. If either the manual E-stop switch or the wireless receiver relay is opened, the reset input on the motor controller is pulled low by the pull-down resistor, and the motor controller is reset to the analog input control state. The circuit includes sense connectors that allow the state of the E-stop system to be monitored by the processors. A circuit board was designed and constructed to implement this circuit. While the manual E-stop switch is open, the motor controller is held in the reset state, and the motors are de-energized. The platform is easily maneuvered by hand while in this state. The wireless receiver has a second channel that is used to produce a one-button 'start' signal when initiating autonomous operation.

4.2.3 Power Conversion & Conditioning

The 12V subsystem incorporates three DC-DC 160W converters, designed for use as motherboard power supplies. Two of these are used for powering the computer systems; the third provides conditioned –12V, +12V, and +5V power to sensors and controls that require regulated power.

4.2.4 Processing Resources

The platform is configured with two computer systems. Each system is comprised of a microATX format motherboard, Intel Core i7-2700K processor, 4GB of RAM, and 64GB solid-state drive. This particular microprocessor incorporates on-board graphics processing. When paired with a compatible motherboard the resulting system requires no additional graphics hardware, with the result that less power and space are required for the complete system. Each processor is housed in a Cooler Master slim case measuring 12.5” x 10.3” x 2.7”, a substantial reduction in space claim over previous years’ entries. The integrated power supply was removed from the case (since power is supplied via DC/DC conversion) which allows the CPU fan to displace enough air to keep the system cool without needing additional cooling fans.

4.2.5 General Purpose I/O

The Platform incorporates an ACCES I/O model USBP-II8IDO4A general-purpose I/O board for general purpose sensing and control. This board has 8 isolated digital inputs, 2 16-bit A/D inputs, and 4 solid-state (high-side FET) relays. The A/D inputs are used to monitor the 12V and 24V battery voltage levels for the purposes of measuring remaining available power. The solid-state relays are used to switch power for the safety light and horn. Digital inputs are used to sense E-stop circuit status as well as the reception of the start signal. Communications with the I/O board is over USB 2.0. Power is supplied via the 12V conditioned power source.

4.2.6 Safety Light

We modified an inexpensive ‘emergency beacon’ to incorporate a standard socket for an automotive turn-signal lamp. The turn signal lamp has two circuits: one controls a low-intensity ‘parking lamp’ indicator, and the other a high-intensity ‘turn lamp’ indicator. The ‘parking’ indicator circuit is hard-wired into the platform power circuit so that it is illuminated at all times that power is available on the platform. The ‘turn’ indicator flashes under software control during autonomous operation. This is accomplished via a solid-state relay on the general-purpose I/O board. The lamp selected is LED-based and exhibits high-reliability, superior brightness, 360-degree visibility, and low power consumption attributes.

4.2.7 GPS

Platform position is obtained via a NovaTel ProPak-LB GPS receiver that incorporates differential corrections obtained via the OmniStar service for sub-meter accuracy in positioning. Communications with the GPS receiver is via RS-232. Power is from the 12V conditioned power source.

4.2.8 Electronic Compass

Platform heading is obtained via a PNI model TCM 3 electronic compass, selected for its configurability, sensitivity, and update rate. Communications with the electronic compass is via RS-232. Power is from the 5V conditioned power source.

4.2.9 Web Cameras for Stereo Vision System

The platform incorporates two Microsoft LifeCam Studio cameras. Communication with the cameras is managed through Emgu CV, which is a C# wrapper for the OpenCV libraries which talks to the camera through USB 2.0.

4.2.10 WiFi Network Adapter

When required for the JAUS portion of the competition, the platform uses a Wireless USB Network Adapter. Power and communications are via USB 2.0.

4.3 Platform Software

Platform processing components use Windows 7 Professional 64 bit as the operating system. Other than device drivers and device libraries provided by device vendors, platform software was developed using the C# language. The C# language was selected for its combination of ease of development, and excellent peripheral support specifically for the off the shelf web cameras which we use for our stereo vision.

Within the software, a number of sub-modules were created, corresponding to major areas of processing. Interfaces were established between groups where appropriate, and development preceded largely in parallel using agile software development techniques. Configuration management was via a secure Subversion server. Using the Visual Studio development environment, along with a subversion client, coordination of code between individuals and sub-groups was a trivial exercise. The sub-groups are: Common Classes, Device Interfaces, Vision Processing & Obstacle Detection, JAUS, Global Path Planning, Local Path Planning, and Simulation.

4.3.1 Common Classes

A number of common classes were developed for use by all developers where appropriate. This includes common-use patterns, data structures, dimensional, and geometric classes.

Two coordinate systems are used during processing. The ‘local’ coordinate system moves with the platform. Its origin is where the platform’s natural axis of rotation intersects the ground plane. The positive Z axis points forward, the positive X axis points to the right, and the positive Y axis points up from the ground. The ‘global’ coordinate system is fixed, with its origin located at some convenient point (a fixed latitude and longitude). In the global coordinate system, the positive X axis points north, the positive Y axis points east, and the positive Z axis points into the ground. Classes were developed that allow straightforward mapping between latitude/longitude and the global coordinate system, as well as between global and local coordinate systems.

4.3.2 Device Interfaces

Device-specific classes were developed. These classes interact with the devices over various communications interfaces, and convert between device-specific values and engineering units. All of our device data is managed through a basic wrapper class which provided the appropriate methods to interact with each device.

4.3.3 Vision Processing & Obstacle Detection

Vision processing relies on a stereo vision system which was developed by the Vision sub-team using Emgu, a C# wrapper for the OpenCV library. The stereo vision system requires many stages which are accomplished using Emgu functions. First, the cameras must be calibrated using a series of chessboard images. The location of the chessboard corners for the left and right camera images is found to the nearest pixel and further refined using a sub pixel search function. The locations of the chessboard corners are then used to calculate camera parameters for the left and right cameras in addition to the transformation which relates geometric location of the two cameras. Once these calibration parameters are known the next step is to rectify the images. Rectifying the images guarantees that corresponding pixels in the left and right images will be in the same vertical pixel row. An example rectified image can be seen in Fig. 4. Once the images are rectified, a semi global block matching algorithm is used to find the location difference between corresponding pixels in the left and right rectified images. An example disparity image is shown in Fig. 3. The three dimensional coordinates can be determined using the disparity image and Emgu's ReprojectImageTo3D function. This function relies on the following matrix

$$Q = \begin{bmatrix} 1 & 0 & 0 & -C_x \\ 0 & 1 & 0 & -C_y \\ 0 & 0 & 0 & f \\ 0 & 0 & -1/T_x & (C_x - C_x')/T_x \end{bmatrix}$$

where (*Cx*, Cy) is the principle pixel in the left rectified image, *f* is the focal length of the left camera in pixels, *C'x* is the *x* coordinate of the principle pixel in the right camera and *Tx* is the spacing between the two cameras. The value of the *Q* matrix is obtained during the calibration process; however, some of the values must be modified in order for the function to work. The disparity image actually returns the difference in pixels multiplied by 16 so the -1/*Tx* term must be divided by 16. Additionally, the 1/*Tx* term has the wrong sign which results in large nonlinear errors in the resulting 3D coordinates. Therefore the -1/*Tx* term is multiplied by -1/16 to obtain the correct results from the ReprojectImageTo3D function. Using these settings the depth measurement provided by this function is accurate to roughly 1/10 inch from 2 to 10 feet away from the stereo cameras.

The vision system distinguishes between three types of obstacles. During obstacle detection the 3D camera coordinates are rotated so that the xz plane is parallel to the ground. This is done by measuring the height of the cameras and the distance to the principle point on the left rectified image. For each obstacle type the vision system provides a list of rotated (x, z) points where obstacles are present.

Obstacle Detection

Obstructions are detected by finding all points where the rotated y distance from the plane is greater than 10 inches. For each pixel in the disparity image, 3D coordinates are found and rotated. The (x, z) coordinates are then passed to the local and global grids.

Surface Feature Detection

Line detection is done using the color version of the left rectified image. The brightness of the blue values is adjusted by subtracting 60 from all blue values. The contrast of the blue values is adjusted using

$$B' = B * contrast + 255 * (1 - contrast) * 0.5$$

where contrast = 200, B' is the contrast adjusted blue value, and B is the original value. These adjustments cause white lines painted on green grass to appear blue. The pixels with B' > 150 are considered to be candidates for lines. If more than half of the pixels in a 4 by 4 pixel block have B' > 150 the block is considered to be a line. The 3D coordinates for each pixel are found using the depth image from SGBM (Semi Global Block Matching). The coordinates are rotated so xz is parallel to the ground and the (x, z) coordinates are passed to the local and global grids. Pixels in the left rectified image identified as lines can be seen in Fig. 5. Note that part of the obstacle is detected as a line. These points are later discarded because the y value for the pixel is too large.

Flags are detected by finding points between 6 and 18 inches above the xz plane. Points with red > 150 are red flags. Similarly points with blue > 150 are blue flags. No contrast or brightness adjustments are done for flag detection.

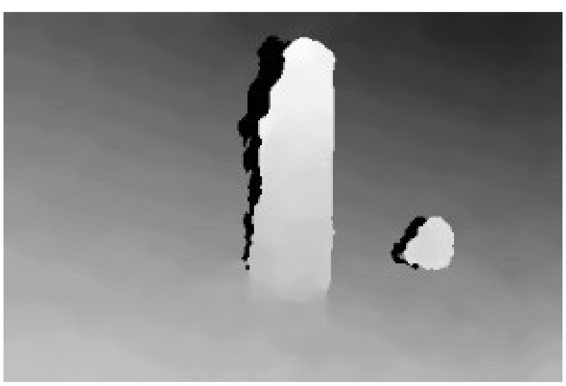


Figure 3 Disparity Image

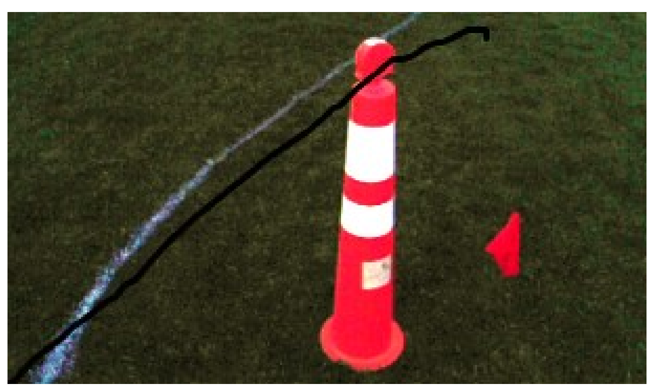


Figure 4 Left Rectified Image

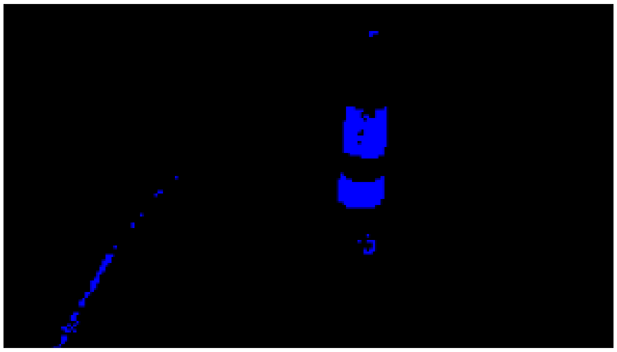


Figure 5 Line Detection

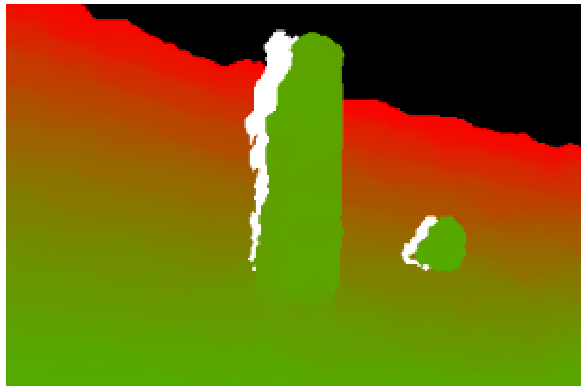


Figure 6 z Distance Image

4.3.4 Global Obstacle Grid

The global obstacle grid acts as a buffer between image processing and path planning. Labeled vision coordinates, compass heading, and GPS location are passed from the vision module to the grid module. The vision coordinates contain horizontal offset, depth, as well as a label which is the type of obstacle the point represents. The compass heading and GPS location are passed at the same time as the points to ensure that the placement is synchronized with vuLTUre's orientation and position. The grid has 2 sets of user parameters that determine its size and resolution: bounding lines and space size. The bounding lines are the northernmost, easternmost, southernmost, and westernmost points around the course. These lines can be determined by as little as 2 GPS coordinates outside the course. Secondly, the space size is simply the size of one grid space. When combined, these 2 parameters can be used to find the grid resolution. Simple formulas can be used to convert from meters to degrees in both latitude and longitude. The user parameters are placed in the program before compilation. The grid can be thought of as a top down view of the course divided into numerous spaces. When the vision coordinates are sent to the grid, they are relative to the cameras' coordinates and as such need to be transformed. This transformation can be broken into 2 main parts: rotation based on compass heading and shifting based on the GPS coordinate of vuLTUre relative to the grid. The rotation is determined by the standard clockwise rotation matrix transformation. Once the rotated x (longitude) and z (latitude) have been determined, they are then shifted. The grid works on an occupancy system. If a grid space is occupied, then that space is put into a hash table that is hashed with key = x + (z * grid_width), where x and z are integers. If a grid space is empty, or is found later to have been populated with noise from the cameras rather than valid vision coordinates, it is removed from the hash table. The coordinates of a space can be easily found given the key. Noise from the cameras images can be very detrimental to the path planning. To correct noise, simple polygon testing is used. Only points in the polygon which the vision system can see are updated.

4.3.5 Global Path Planning

We determined that navigating features such as switchbacks with the potential for dead ends would require global path-planning using a persistent map of the environment. Our path-planner uses an occupancy grid obtained by transforming the output of the vision processor into a global coordinate system, and it outputs a path to the local path-planner, which is responsible for maneuvering the robot to each waypoint on the global path.

We selected a path-planning algorithm called Theta*, a variant on the popular A* path-planner, which was developed at the University of Southern California. Theta* operates nearly identically to A* but produces smoother, shorter paths that are easier to follow than A* paths in real-world applications.

An unmodified A* planner works by exploring outward on a grid from the starting position. It assigns a cost to each new explored cell based on the distance between the cell and the start and an estimated distance to the goal, and then based on cell costs it explores along the most optimal known path. Each time a cell explores its neighbors, it checks whether or not the neighbor could be more efficiently reached by traveling through it, and if so, it assigns itself as the neighbor's "parent" cell. When A* finds the goal, it follows the trail of parent cells backwards from the goal to the start to build the final path.

Like A*, Theta* explores from start to goal on a grid using estimated path costs to optimize which parts of the grid are explored first. However, when it explores around a node, in addition to checking the path from the node to its neighbors, it attempts to make straight-line paths from the node's parent to each neighbor. If the direct path is unblocked and shorter than driving through the node, Theta* uses the direct path. To facilitate the process of checking for unblocked straight-line paths, Theta* builds paths between cells' vertices instead of between cells themselves.

4.3.6 Local Path Planning

After a global path has been planned with a global path-planning algorithm a smaller section of the global grid centered on the robot is processed with a local path-planning algorithm that determines the objective trajectory (velocity and angular velocity) of the platform.

The local path-planning algorithm uses vector field techniques to arrive at this objective trajectory. The obstacles, flags and white lines populate the vector field with forces pushing away from themselves; additionally the white lines create flow forces parallel to them and the flags push left and right perpendicular to the flow depending on their color. Points are sampled along the vector field and projected paths and averaged according to weight to the vector of the path and the vector two the next waypoint to obtain the desired direction of travel. The desired velocities forward and rotational are computed from the desired direction of travel via math with reduced acceleration optimizations and maximum forward velocity optimizations.

As the platform is constantly moving, the objective heading and velocity are constantly changing, and so too is the objective trajectory changing. The result is a dynamic, self-stabilizing approach to local path-planning.

4.3.7 Simulation

During the early stages of the project, we needed a way to test our path planning and driving algorithms before we had a physical robot ready. We selected an open-source software package called Player/Stage to enable us to simulate our robot's behavior in an obstacle-rich environment.

Player/Stage

Player/Stage is a real-time simulator for robots that models their drive motors, range-finding sensors, and GPS. We chose Player/Stage because it encompasses all of the elements we needed to simulate, we could interface with it using the same programming language in which we wrote our navigation program, and because its widespread adoption by well-known robotics firms and government research agencies led us to believe that it would be stable and practical.

Simulation Process

We interfaced Player/Stage with our preliminary driving program using an adapted version of the program that drives our robot and processes input from the sensors. Instead of sending commands to motors and getting data from our image processing code, it communicates with Player/Stage to command and monitor the virtual robot. From our program's perspective, there is no distinction between controlling the simulator as opposed to an actual robot.

To test our navigation algorithm, we recreated the navigation challenge's waypoints and obstacles on the simulator's map and ran our virtual robot through the course. This simple, swift testing process allowed us to tweak, fix, and optimize our program, minimizing the amount of work we needed to do when we completed the mechanical robot.

Challenges

While Player/Stage gave us a distinct advantage during the development process, it also presented us with a number of challenges. Despite its popularity, there is little documentation on performing even simple tasks. As a result, programming the simulator interface took significantly longer than it otherwise would have. Additionally, we could not simulate our robot's characteristics with complete accuracy.

Due to the two-dimensional nature of the program, we had to simulate vision with a laser rangefinder instead of a camera. The rangefinder gave us accurate readings and provided us with a good way to simulate our path finding algorithm, but we had no means of analyzing the effects of any inaccuracies or artifacts that our camera vision system might produce. We also had to use approximations for a number of physical constants (such as the robot's maximum acceleration) that have a significant impact on the behavior of our driving algorithm.

4.3.8 JAUS

For this year's competition the JAUS code had been re-factored from the previous year's entry. The JAUS software module we are using was originally written in Java. Using this code a network wrapper was created which allowed the JAUS Java module and the C# main program to communicate. We used a rudimentary Common Operating Picture (COP) which was developed to test the proper sending and receiving of data packets. For debugging purposes, the Wireshark application was used in conjunction with the COP.

UDP packets are used to send JAUS messages between the platform and the COP. Once the JAUS module receives a message it parses the message into header and the payload, which are each then checked for validity. The details of valid messages are passed to the proper JAUS module using a loopback network connection back to the proper C# module to be processed and the response returned if applicable.

5. Predicted Performance

The following points describe predicted performance and the methods used to determine these numbers.

- **Speed –** Propulsion system design and component selection were undertaken with a goal of achieving the maximum allowed speed of 10 miles per hour under anticipated conditions. This top speed is enforced by setting the motor controller into a PID feedback loop configuration, and by adjusting the encoder scaling parameters appropriately.
- **Ramp climbing –** Propulsion system design and component selection were undertaken with a goal of performing at top speed on a 15% gradient, the specified maximum under IGVC rules. Performance to this goal has been verified in trials.
- **Reaction times –** The vision system achieves a sustained throughput of 15 frames per second for each camera. Based on an analysis of latency in acquisition, processing, and communications paths, it is estimated that an obstacle presented within the field of effect will affect motor speed in 75 +/- 25 milliseconds.
- **Battery Life –** Battery life is highly dependent upon the operational environment. Under continuous load and with a full charge, the 24V battery module life is estimated at 2 hours. The 12V battery life is estimated at 5 hours under full processing and sensor load.
- **Obstacle Detection Distance –** This is configurable via parameters to the stereo vision processing software. Detection is presently limited to 6 meters.
- **Complex Obstacle Negotiation –** Switchbacks and traps are handled as a natural consequence of the path planning algorithm, described in the software design section.

- **Navigational Accuracy –** The geolocation equipment used is capable of sub-meter accuracy when used with satellite- or earth-based augmentation. The GPS sensor is capable of employing satellite-based augmentation and is presently configured to use the OmniSTAR service for differential corrections, which after initial settling will generally achieves a standard deviation of 0.3 meters or less from actual.

6. Cost Data (in dollars)

Mechanical / Propulsion		Sensors		Processing / Electrical	
Wheels, hubs, and tires	500	Web Cameras	120	Computer Systems	1500
Chain & Sprockets	150	GPS System	2,700	Power Supplies	150
Aluminum/Steel Stock	950	Compass	775	E-Stop System	40
Miscellaneous Hardware	250	Motor Controller	410	Safety Light	40
Batteries	340			Touch Screen	315
Motors	1,790			Misc. Electrical	200
Shocks & Casters	160				
				TOTAL	$10,390

7. Labor Data (in man hours)

Mechanical		Electrical		Software	
Design	80	Design	40	Device Interfaces	90
Fabrication	300	Component Selection	20	JAUS-Specific	20
Assembly	160	Integration	40	Algorithm Development	520
				Administrative	180
Sub Total	540	Sub Total	100	Sub Total	810
				TOTAL	1,450