AUTOMATON

University of Central Florida

Team Members:

Michael Scherer, David Adams, Jake Carr,
Jonathan Mohlenhoff, Robin Adams

Faculty Advisor:
Daniel Barber, Institute of Simulation and Training

Faculty Statement: I certify that the vehicle has been significantly modified for this year's competition.

____

Daniel Barber

1 INTRODUCTION

The Robotics Club at the University of Central Florida is proud to present AUTOMATON, a brand new rugged and robust platform that builds on the collective experience of many minds. Construction began in fall 2010 by a small team of dedicated individuals of various ages and expertise who collaborated to produce a competitive platform. Team members were organized based on their respective fields: software, mechanical, and electrical engineering. The goal of this effort is to produce an extensible vehicle which can be built upon by future Robotics Club members in the years to come, by standardizing documentation and code, and by promoting open standards and open-source software.

2 DESIGN PROCESS

Figure 1. Agile Development Cycle

The Agile Development methodology is used for the ongoing creation and deployment of AUTOMATON. It was chosen for its iterative design process which permits the direction taken to be changed dynamically as new problems and design requirements arise. This paradigm encourages frequent releases of working software, shared between developers and customers using an open-source Subversion repository called Zebulon. The Agile design process focuses on direct communication, face to face when possible, to ensure clarity of design and system requirements. Specific attention is given towards the quality of work, where working iterations are used as a measure of progress.

Figure 1 demonstrates the multiple loops involved in the Agile Development strategy. Closer to the center are continuous iterations of development, analogous to the daily software releases of AUTOMATON's code. Further from the center are the more abstract goals. This multi-loop system enables the team to develop in a very dynamic way, reducing risks associated with challenges faced throughout the design process.

Team Member	David Adams	Jake Carr	Jonathan Mohlenhoff	Michael Scherer	Robin Adams	Total Man Hours
Mechanical Hours		350				1250
Electrical Hours			120	100		
Software Hours	350			250	80	

Table 1. Team members and hours dedicated

3 MECHANICAL

AUTOMATON employs four direct-drive wheels attached to an aluminum frame. The four independently driven wheels provide ample torque for traversing off-road terrain. The aluminum frame, chosen for its weight and non-ferrous properties, do not interfere with onboard electronics. All electronics are housed inside the frame except the camera, GPS antenna, and LIDAR. The camera and GPS antenna rise above the vehicle on the mast, while the LIDAR sticks out the front leading the vehicle to maximize scan range. This central body design gives Automaton a clean look with ample room for electronics and future expansions, while maintaining high functionality and weather resistance.

3.1 FRAME

The frame of AUTOMATON is a rectangular prism with dimensions 1x2x3 ft. It is made mostly out of 1 in. x 1 in. aluminum tubing for strength and light weight. Four pieces of 1 x 2 in. tubing on the bottom provide a wide, flat surface for motor mounts and extra structural support. Also on the bottom is an 'X' pattern to provide extra stability for two 12V sealed lead-acid batteries. Cross-braces incorporated into the top of the frame provide support for vehicle payload, with spacing chosen to allow easy access to batteries and other electronics during maintenance. Finally, a sleeve is welded onto the back of the frame for the GPS antenna and camera mast.

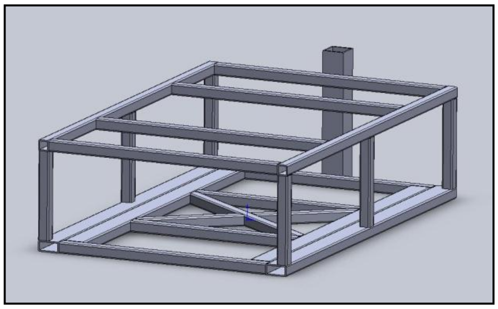

Figure 2: The vehicle's frame

3.2 DRIVE SYSTEM

Calculations performed indicated that four power chair motors provide enough torque and RPM for the specifications of the course. An additional fixture plate added to each motor shaft acts as a mount for quadrature encoders used for closed loop control of each motor. To mount the motors to the frame, a fixture plate was fabricated that provided a flat surface for the motors to mount over frame welds. The motors are located on the frame in an exact square pattern measured from the apex of the center of the tires for differential steering. Wheel hubs made for the motor spindles are held in place by a keyway. Pneumatic tires were chosen so that the vehicle would absorb some vibrations without adding a suspension system. No two piece wheels were available that fit the hubs, so a custom interface plate was designed and manufactured. This made them compatible while allowing extra clearance room between the tires and the frame.

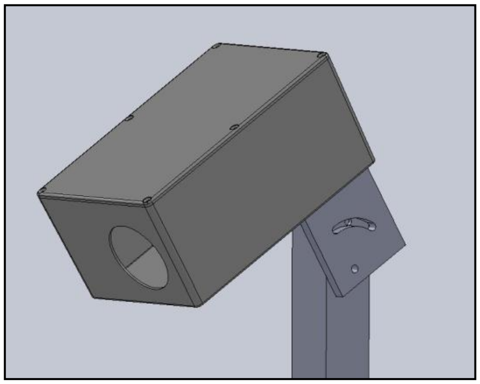

Figure 3: The camera box and mount

3.3 FABRICATION

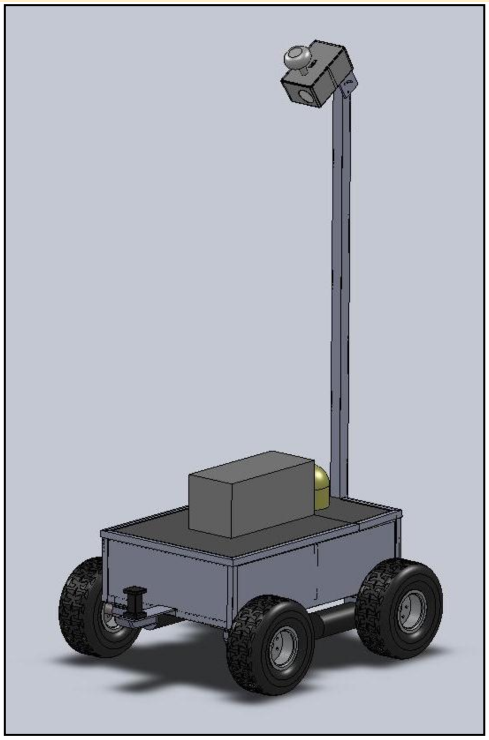

Figure 4: The fully assembled vehicle

Multiple manufacturing processes were used in the building of AUTOMATON. The lengths of tubing used for the frame were saw cut on a horizontal band saw with active cooling and automatic feed. A stop block was used to ensure pieces were precisely cut to length. The frame pieces were then professionally TIG welded to form the rectangular prism shape. Many of the more intricate parts on the vehicle were manufactured on a Milltronics MB-11 three axis CNC machine with active cooling system. This decision allowed the parts to be precision manufactured in a timely and costly manner. Using the part's SolidWorks model, CAM Works was utilized to generate the G Code required by the machine. All parts were made out of easy to machine aluminum, a half inch diameter, two flute, high speed steel (HSS) end mill was chosen for most of the milling work. Due to size limitations, a quarter inch two flute HSS end mill was used on the arc shaped cut out on the camera box mount. All holes less than half an inch in diameter were drilled to their respective drill size. The motor mount plates, the interface plates between the hubs and wheels, and the 'L' shaped piece to mount the camera box were all manufactured on the CNC machine. The camera box is made of cast aluminum and was purchased as a whole. Then the mounting holes and cable run holes, designed to align with the ones on the mounting piece, and lens hole were precision machined. The LIDAR mount was designed to give the widest range of view without being unstable, and was machined to match the hole patter of the LIDAR mounting holes. Using the motor mounting plates as a guide, the holes in the frame for the motors were marked and drilled on a drill press. The sheet metal exterior and trim pieces were cut to size on a vertical band saw and riveted to the frame, giving the vehicle a shiny, sleek look. The polycarbonate lid was was purchased to width and cut to length on a vertical bandsaw and accented with the trim to seal the edges. The tinted finish of the polycarbonate gives a peek inside the electronics.

3.4 INNOVATIVE FEATURES

3.4.1 CAMERA BOX

The design of the camera box aligns the center of vision with the center of the vehicle, as well as allowing adjustability of the vantage angle of the camera. Being centered on the vehicle, the cone of vision produced can easily be implemented into vision algorithms. The adjustability of the camera allows the user to get exactly the range of vision desired. Inside the box, the camera is held in place with two plastic screws to eliminate any interference between the camera and the box. Cables are run up the mast and quietly sneak into the box through the cut-out positioned right above the center of the mast. This increases weather resistance while maintaining a clean look. Cushioning the camera from vibrations is a piece of Sorbothane rubber sandwiched between the box and the camera. The highest component on the vehicle is the GPS antenna mounted on the top of the camera box for best signal quality. The whole mast and camera assembly is removable from the sleeve for ease of transportation.

3.4.2 MAIN BODY COVER

Another innovative feature is the lid system on the main frame. Separated into two pieces, the lid enables easy access to the on-board electronics. It also contains the chords of the vehicle status light and provides weather resistance. The front section of the lid is completely removable and allows direct access to electronic components and easy removal of batteries. The rear section is hinged from the frame and opens to reveal the on-board computer and motor controllers. The light on the back portion of the lid is conveniently flipped up while the wires stay out of the way. The lid is held on by six black latches that prevent it from coming off while navigating the course. The tinted look of the polycarbonate gives AUTOMATON a professional look.

3.4.3 AIR COOLING

An air cooling system was designed utilizing two fans pushing 12 CFM through the vehicle. The two fans are placed on opposite corners of the vehicle, actively cooling the electronics.

4 ELECTRICAL

The electrical system in this vehicle is designed to be simple, easy to maintain, and made of Commercial Off-The-Shelf (COTS) parts to minimize costs and to reduce risks associated with faulty wiring. In the end, results show this philosophy to be very effective in making a platform that performs consistently and effectively.

4.1 POWER SYSTEM

The entire platform runs on two marine batteries which are easily interchangeable with any other marine or car battery. These batteries are hooked up individually in parallel with an on-board charger with an accompanying built-in standard AC outlet, making it very easy to charge the vehicle, hot-swap power, or otherwise work on it while it is not moving without draining the batteries for onboard electronics. A 24V and a 12V line are accessed from the batteries for all components of the robot. The motors run off of the 24V line, while the sensors run off of the 12V line. The computer and camera are powered using a 24V DC to AC inverter so that a standard computer power supply could be used.

4.1.1 SAFETY

Safety is a primary concern of the team, and thus precautions are in place in the event of system failure. There are two main methods of disabling the vehicle: wireless E-Stop and hard E-Stop. The wireless E-Stop is activated by the RC controller, which bypasses the motor controllers to force the robot to stop all actuation. The hard E-Stop is activated by a button about 4 feet from the base of the vehicle. This function completely disables the robot's motion, through power loss, and requiring both a reset of the E-Stop button and for the motor system to be power cycled. Additionally, the system is tied actively to the E-Stop button in such a way that if the line connecting the button were disabled, the system would go into E-Stop mode requiring a power cycle and reconnection of the button.

4.2 ELECTRONICS

4.2.1 COMPUTER

AUTOMATON is equipped with a quad-core Hyper-threaded Intel Sandybridge Core i7 processor with 8 GB of DDR3 RAM running Ubuntu Linux version 10.10. It also has support for connecting to devices which use 1394 FireWire, USB 2.0 and 3.0. RS232 communication is achieved using USB to Serial connectors.

4.2.2 MOTOR CONTROLLERS & REMOTE CONTROL

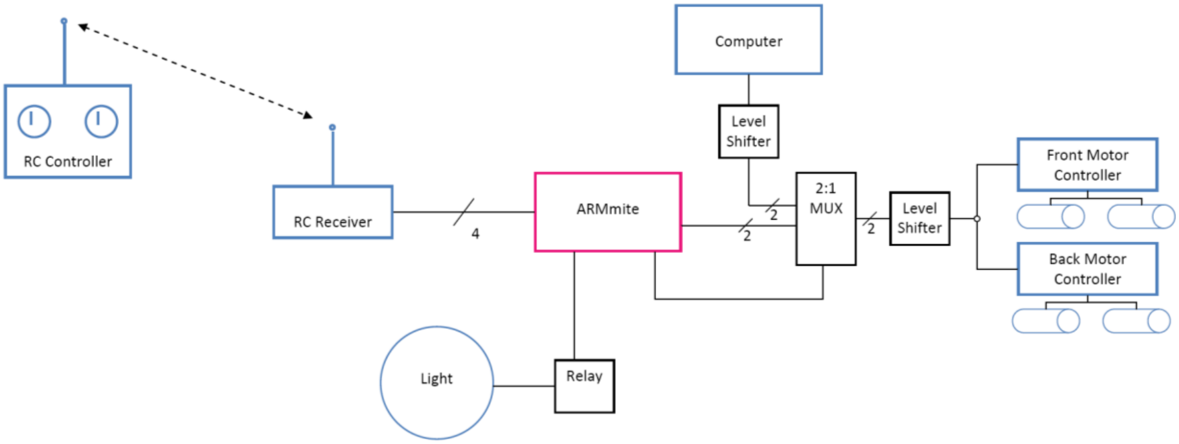

Figure 5. Motor control configuration

Two Roboteq AX3500 motor controllers are employed to actuate the robot's motors. Each controller can output 60A per channel, with one motor connected to each channel. This allows the robot to have the capability to actuate each motor independently of the others. Commands to the controller are sent over RS232 serial. RC and PC control are toggled using an ARMmite processor on a customer circuit board, enabling AUTOMOTON to be driven without the onboard PC being turned on.

4.3 SENSORS

AUTOMATON takes advantage of several different sensors to understand its own kinematics and surroundings. Kinematics are resolved using a 3-axis digital compass, the Coral AHRS, along with 4 quadrature encoders (one for each wheel), and a differential GPS to interpret its motion. The Coral AHRS updates at 100 Hz with onboard filtering using gyroscopes and accelerometers reducing errors while navigating on rough terrain. The encoders, which run at 180 ticks per revolution before the gear box, deliver sub-millimeter accuracy in the rotation of each of the individual wheels, enabling high precision closed-loop operation. The Differential GPS on board provides sub-meter accuracy, with tests indicating 2 to 8 decimeters.

Other sensors have been employed for detecting surroundings: a Sony DCRHC1000 3 CCD Digital Camera and a Hokuyo Laser Rangefinder. The camera outputs full 720x480 resolution over FireWire. The Hokuyo can scan objects up to 30 meters away, sweeping 270° at 60Hz, revealing a large visible range for obstacle detection.

4.4 INNOVATIVE FEATURES

The outside of the vehicle has a standard AC plug which can be used to charge the batteries while still running. This way the transition from charging to deployment is seamless for the vehicle. The RC receiver system is novel because it does not require computer power in order to function, and also allows a clean transition between it and the AI system. Thus, it is easy for a user to take control of AUTOMATON at any given time, reorient it, and then return control to the vehicle. To minimize the need for several USB to RS232 cables, USB to serial chips were placed on the board. Cable management was a key concern for our team, and thus cable ties and mounts were placed around the outer border of the frame which allowed all wires to be routed cleanly throughout the system.

5 SOFTWARE

Over time, an enormous base of software for all robotic vehicles made at the Robotics Club at UCF has been built-up within a completely open-source online repository called Zebulon. This repository contains code for everything from visualizations, sensor input, and embedded system interoperability. It has been extended with new code to implement a brand-new architecture on AUTOMATON that takes full advantage of SAE-JAUS in a manner which is clean and consistent with the standard. All code is commented using Doxygen formatting guidelines for automatic generation of code documentation, making the software clear and understandable for the development team and 3rd party developers.

5.1 STRUCTURE

Joint Architecture for Unmanned Systems (JAUS) compliance is a requirement for AUTOMATON to support the JAUS Challenge and customer demands for open-standards. Pronounced "jaws", and maintained by the Society of Automotive Engineers (SAE), SAE-JAUS is a service-based architecture, defining re-usable capabilities broken down to the lowest level possible. This architecture heavily influenced AUTOMATON's software design and implementation because of its well defined interfaces, facilitating independent development of system modules. Use of SAE-JAUS builds upon previous teams experience and lessons learned for organizing software, and improved past solutions.

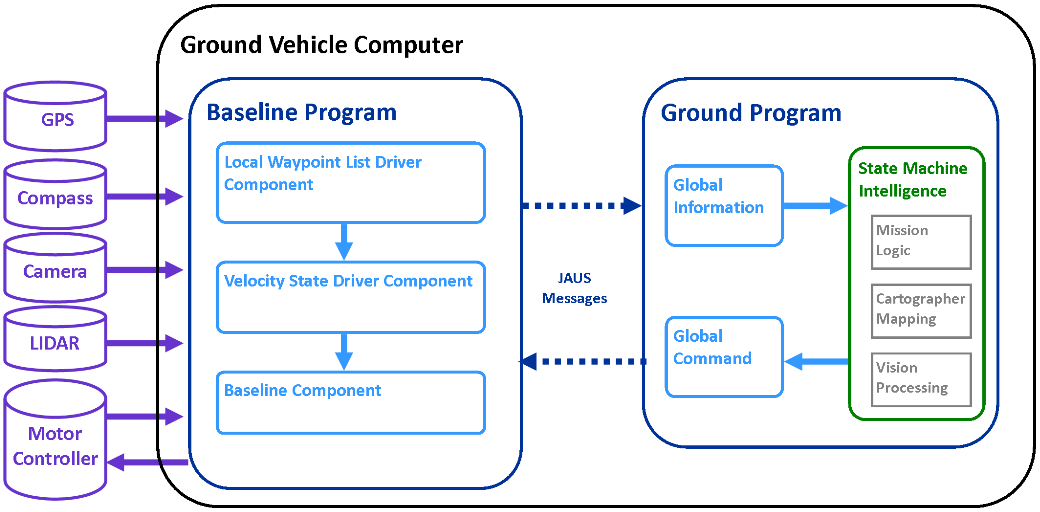

Figure 6. AUTOMATON Software Structure

AUTOMATON is composed of two main programs, "Baseline" and "Ground." The Baseline Program shown in Figure 6, integrates all hardware and sensors, abstracting them via JAUS services. This program is standalone, and allows for any JAUS compliant program to subscribe to data streams or request control, which is how the Ground Program performs autonomous navigation. This Ground Program contains all of AUTOMATON's intelligence, including vision processing, mapping and obstacle avoidance. These two programs work together in a master/slave relationship, the Ground Program analyzes data from Baseline, makes decisions, and produces commands that are sent back to appropriate Baseline services to execute.

The programs are currently using open source JAUS++ library available on sourceforge.net to for SAE-JAUS compliance. JAUS++ was selected for this project because of its strong documentation and support, friendly license, and proven compliance with SAE-JAUS at other AUVSI sponsored JAUS Challenges.

5.1.1 BASELINE PROGRAM

In addition to abstracting sensors and other hardware using JAUS services, the Baseline program provides support for the JAUS Challenge via the following services implemented within the Ground program: SAE-JAUS Core Services, Velocity State Sensor, Local Pose Sensor, List Manager, and Local Waypoint List Driver. Moreover, Baseline also implements other JAUS services to support open and closed-loop drive control. The Primitive Driver service lets other JAUS components acquire low level open-loop drive control of AUTOMATON. The Velocity State Driver is one such service that uses the Primitive Driver, with subscriptions to sensor services like Local Pose Sensor and Velocity State Sensor to provide close-loop velocity and rotation rates. Therefore, Baseline provides a standalone process that developers can leverage to construct autonomous applications that can control AUTOMATON in a multitude of ways depending on customer needs.

5.1.2 GROUND PROGRAM

Having a robust and stable software platform is part of the challenge in creating a competitive autonomous vehicle. AUTOMOTON accomplishes this with its Baseline Program, but of course, full autonomy is the final goal. The Ground Program builds upon Baseline, adding intelligence by way of path planning, vision processing, and state machine logic to accomplish all tasks provided by IGVC.

5.2 MAPPING AND OBSTACLE AVOIDANCE

All of the maps are constructed in a framework called Cartographer. This framework is very flexible, allowing the system to build vector or raster based maps depending on what is deemed best for a given situation. Abstracting the type of underlying map, developers can implement different types of path planning algorithms for obstacle avoidance and navigation. This specific feature supports prototyping many different algorithms to determine what works best for the challenges presented. The map used is a 2 or 3 dimensional, top down view of the world. Map information is stored in a Quadtree structure to take advantage of spatial locality of objects in order to minimize search space for path searching algorithm.

5.2.1 AUTONOMOUS CHALLENGE

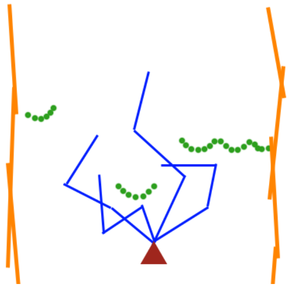

Figure 8. Conceptual example of beam search algorithm

For the autonomous challenge, there are two types of objects added to the map. The first are points, generated from the LIDAR, represent elements of a point cloud in the environment. These points come together to form barrels and other obstacles. The other type is the lines, which are treated like walls in the map's representation. The computer vision system projects the lane lines it finds into the ground coordinate space, and then added to the map as a line segment (or wall segment). All obstacle avoidance is then carried out to navigate around these "dots and lines". Lane lines are given a unique ID within the map, for determination of lane direction and center.

For path generation, a beam search algorithm was chosen. This type of algorithm attempts to find a set of connected beams which will reach the desired goal without hitting any obstacles. A depth of 3 beams slightly wider than the vehicle are chosen to give tolerance to the size of the vehicle and also because it gave a sufficient search space for the robot to find a path in most instances. A single path is chosen using a fitness function which weighs paths on their length and proximity to the goal. Using the results of the algorithm, a velocity and heading are generated for closed-loop drive control.

Higher and lower speeds can be tolerated dynamically by adjusting the beam length and depth. For instance, if the robot travels at a faster speed, the length of the beams can be extended and the maximum angle which the beams are allowed to bend is reduced so that the robot can anticipate motion in advance and does not need to make as drastic of turns. Vehicle speeds are selected based on how cluttered the surrounding environment is, with high speeds chosen in low density areas, and low speeds when object density is high.

5.2.2 NAVIGATION CHALLENGE

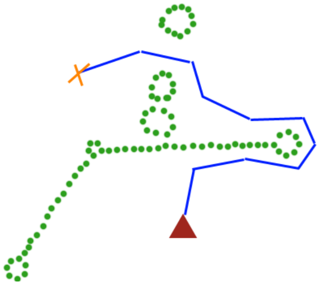

Figure 9. Conceptual example of best-first search result

For the navigation challenge a similar system to the autonomous challenge is employed, with the exception being that there are no lines added to the map. Additionally, maps maintain a longer persistence in this challenge, due to the higher tendency of traps which could be encountered repeatedly. Although this creates some problems with map data conflicting when something has been viewed at different times, showing up in different locations, these anomalies are accounted for through combining objects discovered objects with similar parameters, applying a time-to-live for objects within the map, and continuous reassessment of the environment. Navigation is done using a best-first traversal algorithm, where goal points are set in a list given before run time. This algorithm was chosen because it is one of the fastest approaches to search. However, unlike Dijkstra's algorithm and A*, which take O(n) amount of memory, best-first uses memory O(n^2). The system can tolerate this memory usage because it has 8 GB of on-board RAM, and therefore memory was traded in favor of speed.

5.3 VISION PROCESSING

Figure 10. Sample Image of IGVC course showing an orange and white obstacle and white lane boundaries

Vision processing, a simple every day process for a human, remains an immensely difficult problem for computers and AI. While difficult, the onboard vision processing is a necessary and critical part of creating a vehicle capable of navigating autonomously through the challenges presented by IGVC. Vision would be best applied to the problem of identifying and locating boundary lines of the course and providing supplemental cues to the AI about potential obstacles or objects of interest.

Our goal for AUTOMATON's vision system is to find, classify, and estimate distances and angles of lane boundary lines, as well as to recognize and filter out potential obstacles and objects in its path. This is accomplished with help from the OpenCV library, used for its powerful and well documented computer vision tools. The problem is broken up into two logical steps, obstacle filtering and lane line detection.

5.3.1 OBSTACLE FILTERING

The final IGVC course contains several types of objects that could be physical barriers or technical barriers to recognizing and staying within boundary lines. Using images captured from past years IGVC events, as well as this year's Washington D.C. event, the software team was able to perform offline testing of actual conditions and create a system robust enough to recognize and filter out unwanted objects.

5.3.1.1 COLOR-SPACE CONVERSION

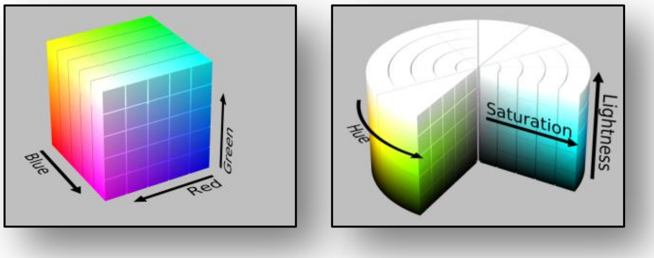

Figure 11. RGB vs HLS 3D color-space representations (1)

Images that come straight from the camera are in a red green blue format (RGB), but several other formats exist, including those that separate hue, saturation and lightness, which can be more reliable in computer vision applications. This allows easier color classification, since lightness and saturation channels can be ignored in favor of just hue (1).

5.3.1.2 COLOR CLASSIFICATION

Rather than explicitly classifying certain features in the images as obstacles, AUTOMOTON simply removes them from the image as to reduce interference with lane line finding. This is achieved through use of color classification, a method that, while primitive, has proven itself in past competitions for its speed and simplicity. Many objects in the IGVC challenge are made of distinct bright solid colors. The algorithm searches for these specific colors which must be sampled from example images; pixels falling within a certain threshold are activated and together with blob finding, a binary mask is created of areas to ignore. Some objects can be completely detected this way; others remain a bit more challenging.

5.3.1.3 BLOB FINDING

Once colors of interest have been identified, "blobs" of colored areas can be compared; blobs of a certain size will be removed from the image, while others will be left alone. More complex objects can contain several small areas of color and white, and blob finding may detect a single object as several small objects.

Figure 12. Conceptual example of blob detection and blob merging of complex

By taking advantage of regularities in the patterns of obstacles, AUTOMATON can confidently merge blobs after considerations to proximity, size, shape and angle. While this method is not a robust solution for objects the real world, it is adequate for the controlled environment of the IGVC course.

5.3.2 LANE LINE DETECTION

The most important function of AUTOMATON's vision processing is the ability to estimate position relative to discovered lane lines. Through a combination of filters, perspective correction, edge detection and line fitting AUTOMATON accomplishes this goal. The software team researched existing approaches to solving this problem, including solutions from previous IGVC events; ultimately a combination of methods proposed from several sources in years past were referenced: Princton's PAVE algorithm (2) and UCF's own Calculon.

5.3.2.1 FILTERING

Figure 16. Original Image

Figure 14. Converted color space, smoothened and warped for perspective.

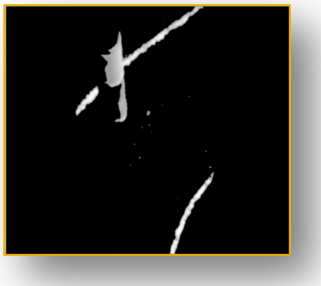

Figure 15. Threshold applied to highlight bright areas. Obstacle remains.

Similar to color classification in object removal, a white filter is applied to the image after obstacles have been removed in order to identify bright areas in the image. Usually there are many areas in an image that can appear white or bright, particularly on sunny days, so more filtering steps must be taken. The image is passed through a Gaussian blur to smooth and a threshold to set dark areas to zero.

5.3.2.2 EDGE DETECTION

Edge detection is used to outline any contrasting areas of the image. If run on an image of a complex scene, thousands of edges could be found, if run on a fairly sparse image though, a much more reasonable picture emerges.

Images being processed at this stage are usually fairly free of any complex shapes and hard edges due to object removal and filtering stages. This makes edge detection ideal for identifying the last remaining features in the image: the lane lines. A Canny edge detector is used to process images at this stage which will primarily outline edges of white lines on green grass.

Figure 18. Edge detection with barrel obstacle removed, some noise remains.

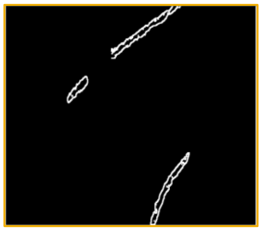

Figure 17. Removal of noise through small blob finding.

However, the process is not perfect, brown patches in grass or other unanticipated objects can be picked up. So once edge detection is run blob detection is used to remove any edges smaller than some predefined level; this removes much of the random noise while maintaining line edges.

5.3.2.3 LINE FITTING

The image at this point is essentially black with white pixels outlining various objects that stand out against the background. They must now be fitted to a computable line. There are several common solutions to this problem, the software team chose to use the RANdom SAmple Consensus (RANSAC) algorithm for line fitting. Tests against popular Hough line transforms showed better consistency in fitting lines, and had the added benefit of returning a single best fit line rather than many lines which needed additional processing (2). RANSAC also deals well with gaps in lines, which can happen often through either course design or data loss from overzealous filters.

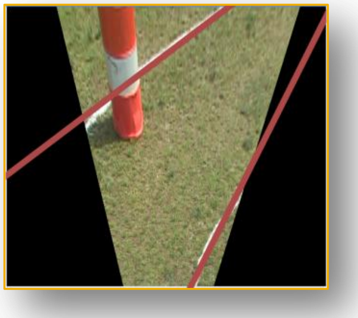

Figure 19. Final Lines, superimposed on original image.

Since RANSAC only returns a single best fit line, the algorithm has to be run twice to detect both left and right lane lines. If on the first run a line is found, it is simply masked with black and RANSAC is run again to find a second line, if one exists.

5.3.2.4 PERSPECTIVE CORRECTION

AUTOMATON's single camera is mounted about 6 feet above ground and aimed at a downward angle to see the ground directly in front of the vehicle. In a 2D image captured from the camera, objects and lines close to the vehicle appear large or far apart, while objects and lines further from the vehicle appear small or close together. This is the illusion of perspective, and because of this illusion an accurate mapping of line angles, lengths and locations cannot be done without some transformation of the 2D image.

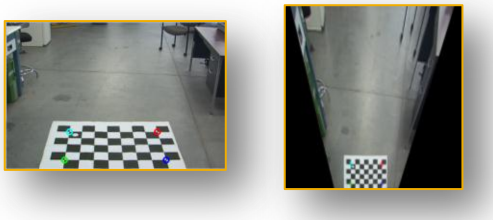

Figure 20. Calibration and results. Correction gives image a top down appearance.

Correcting perspective requires an offline calibration phase in which a rectangular object of known size is placed on the ground plane in front of the vehicle. The corners are picked out, and a matrix is generated that transform relative pixel coordinates of the corners to the actual known relative corner positions. This matrix can then be applied to the entire image to correct each pixel for perspective distortion. This has the primary benefit of correcting angles that would be incorrect due to perspective, such as the illusion of parallel lines converging in the distance (2).

The secondary benefit of this correction is that pixels can also be mapped to an arbitrary unit of distance; in AUTOMATON's case each pixel is mapped to 1 centimeter. This mapping only applies to features on the ground (such as lane lines), and can be fairly accurate depending on the quality of image and calibration.

5.4 INNOVATIVE FEATURES

Writing software is an incremental process, each stage hopefully builds on previous work and research. The goal with AUTOMATON was not to reinvent the wheel, but to leverage the experience and expertise gained from previous competitions and add improvements where necessary.

5.4.1 REAL-TIME CONFIGURATION

Most important variables that tweak AUTOMATON's behavior have been added to XML files which are easily human readable and modifiable. These files contain information about the robot's JAUS configuration, hardware devices, vehicle dimensions, mission, and so on. For situations that require input or training, such as vision processing or Proportional Integral Derivative (PID) controller tweaking, AUTOMATON can load changes in real-time so that effects can be observed immediately.

5.4.2 LOGGING AND PLAYBACK

Recording data for analysis is an important part of improving performance of this vehicle. Through XML, logging of all sensors, including video and LIDAR can be turned on at any specified rate. Recording runs through courses will be valuable data for not only the current team but for future UCF teams that wish to attend this competition.

Because of the flexible and modular nature of AUTOMATON's software systems, logged data can be replayed through AUTOMATON's AI, this allows for a powerful way to understand the dynamics of AUTOMATON's decision making, as the system knows no difference between being fed logged data or real-time data.

5.4.3 REAL-TIME MAPPING

The Cartographer mapping library has been used before in previous competitions, but not to the same extent as AUTOMATON. This year sees the addition of more efficient map storage and retrieval, improved path planning and navigation and for the first time cooperation with the vision processing subsystem. Lane lines found through vision are merged with LIDAR data to create a detailed and cohesive map of the environment around AUTOMATON.

6 VEHICLE PERFORMANCE & ANALYSIS

6.1 PERFORMANCE

AUTOMATON has been designed to meet or exceed the requirements of this competition now and years to come. It is rugged and robust, built to be weather resistant and terrain dominating.

Metric	Analysis
Speed	7.6 miles per hour
Battery Life	2-3 hours (approximately)
Max Tested Incline	50°
Reaction Time	.07 seconds (approximately)
Object Visible Range	30 meters
Trap, Dead End Compensation	Global map stored so that alternate routes can be found
Waypoint Accuracy	20-80 centimeter

6.2 BUDGET

Although several of these parts were inherited from previous projects at the lab, their cost has been included in order to better describe the price to duplicate the work.

Item	Unit Cost	Quantity	Total Cost
Computer	$800.00	1	$800.00
Aluminum for Frame	$350.00	-	$350.00
Welding Work	$100.00	-	$100.00
Wheelchair Motors with Gearbox	$299.99	4	$1,199.96
Mini-MAX DGPS	$2,000.00	1	$2,000.00
180 PPR Quadrature Optical Encoders	$50.00	4	$200.00
Wheels, Inner tubes, Hubs	$78.04	4	$312.16
Hokuyo LIDAR	$5,000.00	1	$5,000.00
Coral AHRS Digital Compass and IMU	$1,245.00	1	$1,245.00
Sony DCRHC1000 3 CCD Digital Camera	$1,183.00	1	$1,183.00
Roboteq AX3500	$400.00	2	$800.00
Miscellaneous Electronics	$400.00	-	$400.00
Miscellaneous Mechanical	$200.00	-	$200.00
Total:			**$13,790.12**

7 CONCLUSION

The designers of AUTOMATON play to be competitive for the customer, and that is why so much time and effort is placed into making a clean and polished product. It is the work that defines this organization who hopes what has been built exceeds customer expectations, setting a new bar for quality. This platform is designed for the long haul, with an open design for continuous improvement and advances by the developers and customers.

7.1 SPECIAL THANKS

We would like to give a special thanks to Dr. Gary Stein for his continued support of the club, providing us with advice and help when we needed it. We would also like to thank Daniel Barber, our faculty advisor for his support and continued efforts. We would also like to send a big thanks to Dr. Cyrus Hillsman, who let us spend countless hours in the machine shop to work on a lot of the parts for our robot. Finally we would like to send enormous thanks to the Institute of Simulation and Training and the United States Army Simulation and Training Technology Center for providing our incredible lab space and continued financial support.

7.2 REFERENCES

1. *Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography.* **Fischler, Martin A. and Bolles, Robert C.** 6, s.l. : Communications of the ACM, 1981, Vol. 24.

2. **Yaniv, Ziv.** Ransac Implementation. [Online] http://isiswiki.georgetown.edu/zivy/.

3. **Wikipedia.** HSL and HSV. *wikipedia.com.* [Online] May 3, 2011. [Cited: May 8, 2011.] http://en.wikipedia.org/wiki/HSL_and_HSV.

4. **Bradski, Gary and Kaehler, Adrian.** *Learning OpenCV.* s.l. : O'Reilly Media, 2008.

5. *Kratos: Princeton University's Entry in the 2008 Intelligent Ground Vehicle Competition.* **Chris A. Baldassano, Franken, Gordon H., et al., et al.** s.l. : Proceedings of the IS&T/SPIE Electronic Imaging Conference, 2009.