SCHILDKRÖTE INTELLIGENT ROBOTIC PLATFORM

Oakland University
Brian Neumeyer, Lucas Fuchs, Nick Anger, Michael Grinols, Christopher Hideg, Mackenzie Hill, Patrick Hines, Rani Karana, Michael Lohrer, Dakota Perna, Vincent Van, Daniel Vega, Rene Zorzi
Ka C Cheok Ph.D. (cheok@oakland.edu)

ABSTRACT

This paper presents the Schildkröte robotic platform developed to compete in IGVC for 2016. An octogonal aluminum frame houses a powerful powertrain capable of near zero-point turns and allows for off-road operation. Inside the frame is an aluminum electronics box with a custom-made control board, motor controllers, power supply circuitry and LIDAR receiver. LIDAR is attached to the top of the base along with a sensor tower which includes the stereo camera and GPS antennas. To process the sensor input, two laptops are housed in the frame to protect them from outdoor elements. The Robot Operating System (ROS) is used as a programming environment to take advantage of pre-existing, open-source packages such as path planning and image preprocessing. Innovations in the software system include multi-rate Kalman pose estimation, effective interpretation of stereo vision data, fusion of LIDAR with monocular cameras, line fitting with the RANSAC [2] algorithm, 3D map-based path planning, and the ability to create and load reusable maps of the environment.

INTRODUCTION

Oakland University is proud to enter Schildkröte into the 24th annual Intelligent Ground Vehicle Competition (IGVC). Schildkröte is a two-wheel drive platform, employing differential drive steering. A custom-made embedded microcontroller board was designed to meet the specific requirements of the IGVC vehicle. All software systems, including vision processing and map-based path planning, were simulated and integrated in the powerful Robot Operating System (ROS) environment.

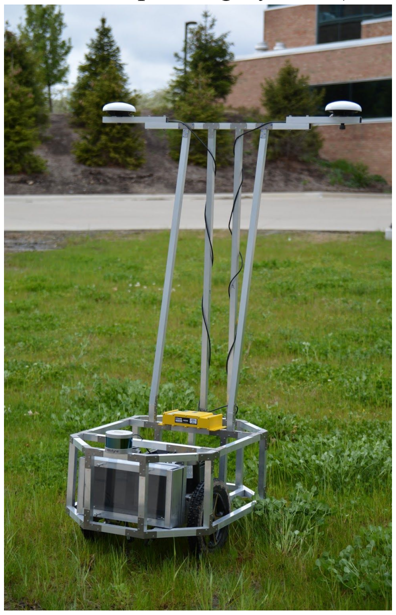


Figure 1 - The Robotic Platform Schildkröte

DESIGN PROCESS

A classic 'V-Model' design process was followed to develop Schildkröte, shown in Figure 2. After defining the requirements of Schildkröte, a design was formed using CAD and a detailed simulation environment was created to develop the navigation system; see Figure 3. After implementing the design and integrating the various components, a rigorous test cycle began where consistent failure points were identified and rectified through both minor adjustments and larger design changes as necessary.

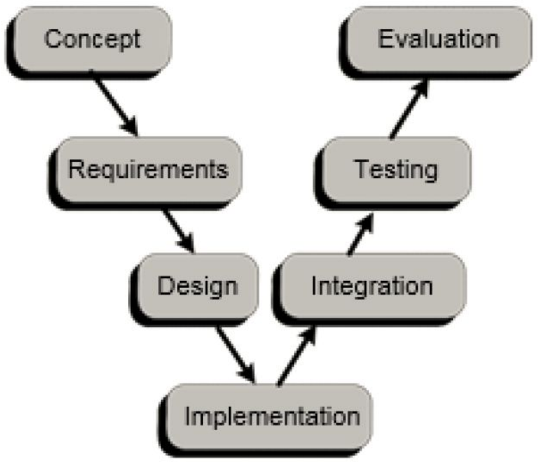


Figure 2 - The design process of the build

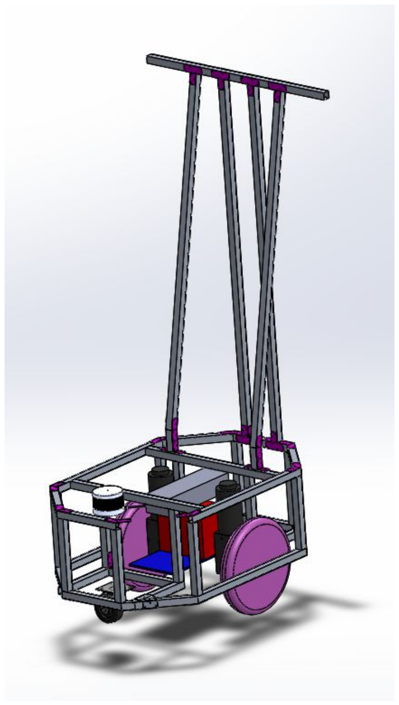


Figure 3 - CAD Rendering of Schildkröte

MECHANICAL DESIGN

At the beginning of the design phase, all of the requirements were listed and accounted for; including the sensor positions, drivetrain restrictions, payload carrying capability, size and speed constraints. Based on past IGVC experiences, it was determined that the robot should be able to perform close to a zero-point turn. This capability greatly simplifies the path planning and control. The simplest and most widely used drive method that enables zero-point turns is differential steer. Additionally, differential steer simplifies the mechanical aspects of the drive train, since it operates solely on wheel rotation, and does not require any additional moving parts.

Once the constraints were set, a CAD design was made in Solidworks. An octogonal frame was chosen due to its simplicity yet durability and all of the pre-determined components were added to the model. The CAD model allows for an easy visualization which assists in fitting all components within the frame.

The aluminum octogonal base is 36" x 24" while the overall height including the sensor tower is 70". The base is comprised of square extruded aluminum with 1/16" wall thickness and held together using steel brackets and rivets. This combination makes the base sturdy, yet light. In addition, it allows for powerful electrical motors to be mounted near the geometrical center so that the platform is able to make near zero-point turns. The base has a single caster wheel to assist in support. Using careful weight distribution the robot is able to remain balanced using only the three contact points created by the wheels.

Aside from drivetrain, the base contains compartments as to allow other components to be placed in a compact manner. These components include the electronics box, batteries, battery charger and weather-resisting materials. A large piece of polycarbonate composite has been cut to size to shelter the sensitive internal components and thereby increase the weather-resisting capabilities of the robot. This polycarbonate sheet is also used to mount the LIDAR near the front of the robot and the payload near the center, to shift the center of mass forward.

The top of the base provides support to a tall aluminum structure which houses sensors including the stereo camera and GPS antennas. The structure is made as tall as possible, without passing the IGVC-specified maximum height, in order to created the largest field of vision for the stereoscopic camera and prevent potential interference of the metallic base on the pair of GPS antennas.

Creating Schildkröte's base from scratch has proven beneficial since the design of certain components could be tweaked to optimization as it was being built - such as its center of mass. By using lightweight materials and a minimalist design as displayed in Figure 4, the total weight came to around 95 pounds.

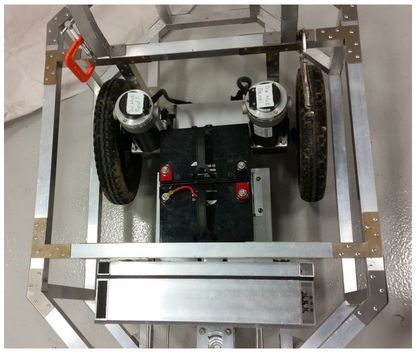


Figure 4 - The Base for Schildkröte

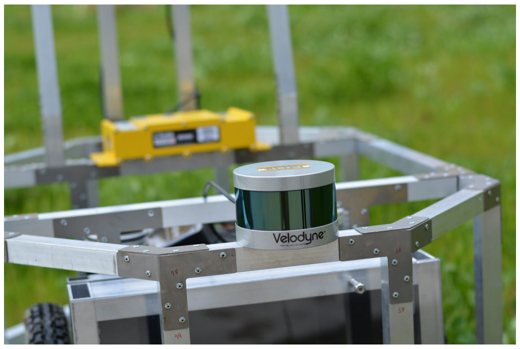


Figure 5 - Velodyne LIDAR

ELECTRONIC COMPONENTS

Schildkröte is designed with a fully modular electronics bay as shown in Figure 6. This efficient design allows for the main power systems to be separated from the chassis via series a of easy disconnects. The main electrical systems of the robot include 12V, 5V, and 3.3V regulators, a circuit breaker, a motor power relay, a Velodyne power supply, a dual-channel H-bridge and drive controls.

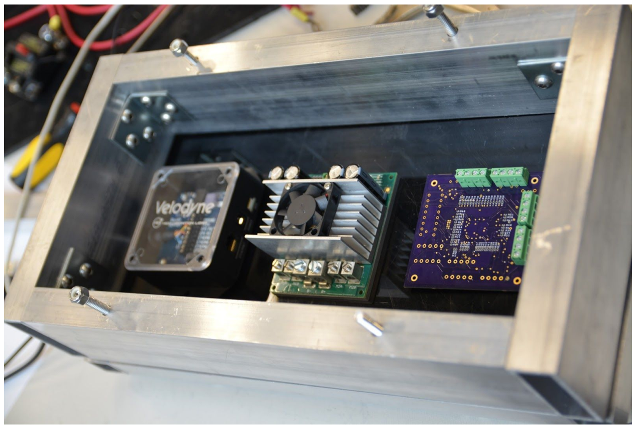


Figure 6 - Electronics Box

Figure 7 - Sabertooth 2x60 Dual Motor Driver

The removable electronics also allows for easy off-chassis integration, diagnostics and testing. The connectors were set at both sides of the electronic bay and they were identified per connector as well as based on I/O numbers for universality between the power and load distributions.

A Sabertooth 2x60 dual-channel H-bridge, shown in Figure 7, was used to power the motors at 24V. Due to the potential of high current draw by the motors, a robust H-bridge was required. The Sabretooth fits that constraint, with the following features:

- Capable of 60 A continuous draw per channel
- Absolute peak load of 120 A per channel
- 6-30 V input range
- Small footprint (3” x 3.5’ x 1.8”)
- 24 kHz switching frequency
- Automatic shutoff for overcurrent and overheating

Not only that, but according to the manufacturer's specifications the Sabertooth motor control is suitable for robots up to 1000 lbs; which is more than enough for this purpose. Another unique and useful feature taken into consideration was that this motor driver is one of the first synchronous regenerative motor drivers in its class - allowing the batteries to be slightly recharged whenever the robot is decelerating. This helped with safety since the back-EMF will go back into the batteries.

Drive Control Board

Schildkröte needed a main controller unit in order to control and communicate between most of the electrical devices and motors and the PC. Since the team has been successful with custom drive control electronics in the past, this motivated the design and construction of an entirely new board. Shown in Figure 8, this custom board is built around the 5LP Programmable System-on-Chip (PSoC) from Cypress Semiconductor. The chip contains 24 Universal Digital Blocks (UDBs) as well as an ARM processor, which allows it to make use of a synthesis of hardware and firmware. The embedded control board is designed to satisfy all hardware needs and provide the flexibility of a commercially available development board without the extra bulk and space of unused functionality. The board was designed to serve several functions:

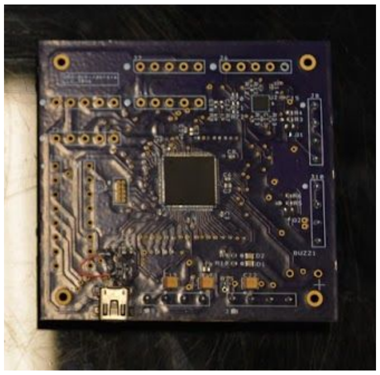


Figure 8 - Custom Controller Board

- Interface between motors and laptops using USB communication
- Allow Schildkröte to be controlled with an RC radio transmitter
- Read encoder signals from the motors
- Control the emergency shutoff relay for the motors
- Read motion data from an IMU
- Allow quick embedded development

This board is able to perform these function through the following methods:

- The PSoC reads the encoders using hardware, ensuring accurate readings
- Serial or PWM is used to control the Sabretooth H-bridge
- A mosfet is used to switch a 24V relay
- RC radio receiver is read using PSoC hardware, giving accurate readings

Figure 9 shows a block diagram of Schildkröte's power distribution system. Schildkröte's power comes from two 12V automotive batteries wired in series to give 24V. Due to the possibility of high amperage draw from the motors, both 8 gauge wire [3] and a 35A inline fuse are necessary. Using this size wire gauge helps to avoid heat build up in the connections around it during times of continuous high current draw. A circuit breaker is used to switch the power on and off to the power distribution system. The operator has the ability to power-up the embedded control board and sensors separate from the H-bridges for testing and safety purposes. The batteries can be conveniently charged on-board, or quickly replaced by another set to achieve optimal runtime. Power and ground were distributed using two terminal rails; one for power and other was for ground. This allows an even distribution for both common ground and power among the elements of the electronic bay. The electronics bay houses a 12V regulator which is then regulated to 5V and 3.3V through separate buck converters. 5V and 3.3V are needed in order to power the main controller board as well as the GPS receiver.

For Schildkröte, a Trimble BX982 GNSS GPS Antenna and receiver pair was used, shown in Figures 10 and 11. Trimble has dual-antenna input which makes it provide better heading information. In the previous competition, accurate heading information was a problem, which influenced the decision of using dual antenna-input GPS. Through the use of this GPS, the true heading accuracy increased from 1.5m to centimeter accurate positions and less than a tenth of a degree based on on the manufacturer datasheets and with a recommended 2 meter baseline. The USB, CAN, and Ethernet ports allow one to easily incorporate the receiver with the rest of the hardware.

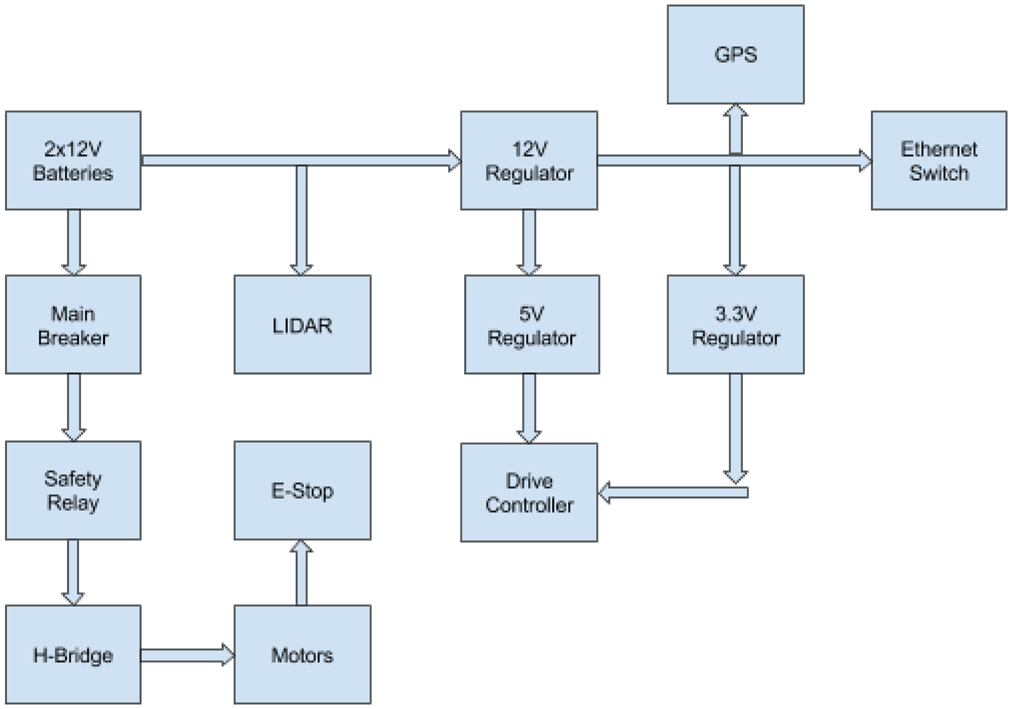


Figure 9 – Power Distribution System

Schildkröte is equipped with an array of sensors that allow it to detect obstacles, compute its location, heading, speed and be operated in a safe and reliable manner. All the sensors make it possible for Schildkröte to locate itself and have a high precision when maneuvering.

Figure 10 - Trimble GPS Antennas

Figure 11 - Trimble GPS Receiver

The sensor array consists of:

- Trimble BX982 GPS receiver ○ 20 Hz, with centimeter accuracy, shown in Figure 11.
- Oriense-Tech OrSens stereo camera ○ 7 meter range, 30 Hz at 640x480, shown in Figure 13.
- Velodyne VLP-16 LIDAR sensor ○ 100 meter range, 360° FOV and ±15° Vertical FOV, 20 Hz, shown in Figure 5.
- InvenSense MPU-6000 6-axis gyro and accelerometer, integrated shown in Figure 8.
- US Digital E3 Wheel 2500 CPR encoders
- DX6i wireless R/C aircraft joystick
- Embedded controller-based manual control and wireless E-stop

Safety Considerations

Many precautions were taken into account when designing Schildkröte's emergency stop system. In addition to two conventional turn-to-release E-stop switches, a DX6i joystick is used for disabling the motor output wirelessly. The DX6i has a range of several hundred feet. To protect against a variety of failure conditions, the drive control system automatically turns off the motors if it fails to receive commands from the computer or joystick after 200 ms.

COMPUTING HARDWARE

Embedded Controller

The microcontroller sends inertial measurements and wheel speeds at a rate of 200 Hz. Closed loop velocity control is critical to accurately follow a plan generated by higher level software and, therefore, closed loop velocity PI controllers were implemented on the embedded microcontroller for each wheel. The velocity feedback is reported to higher level software for localization. Velocity commands come from higher level software or the DX6i wireless joystick when in manual control.

Laptop Computers

All high-level processing is performed on two Lenovo Thinkpad W530 laptops. Their processor is a quad core, 3.4 GHz Intel i7, and has 16 GB of RAM. To make the laptops robust to the vibration encountered on a ground vehicle and to enable saving the data in real-time, a solid state drive is used instead of a conventional hard disk drive. The operating system is Ubuntu 14.04, and runs ROS Indigo. An ethernet switch manages the communications between all the devices.

ROS SOFTWARE PLATFORM

Schildkröte's software systems are implemented on the Robot Operating System (ROS) platform. ROS is an open-source development environment that runs in Ubuntu Linux. There are a multitude of built-in software packages that implement common robotic functionality. Along with the many drivers for common sensors like LIDAR, cameras and GPS units, there are also general-purpose mapping and path planning software modules that allow for much faster implementation of sophisticated navigation algorithms.

Efficient Node Communication

A ROS system consists of a network of individual software modules called "nodes". Each node is developed in either C++ or Python and runs independently of other nodes. Communication messages between nodes are visible to the entire system similar to an automotive CAN bus. Inter-node communication is made seamless by a behind-the-scenes message transport layer. A node can simply "subscribe" to a message that another node is "publishing" through a very simple class-based interface in C++. This allows for the development of easily modular and reusable code, and shortens implementation time of new code.

Debugging Capabilities

One of the most powerful features of ROS is the debugging capability. Any message passing between two nodes can be recorded in a "bag" file. Bag files timestamp every message so that during playback, the message is recreated as if it were being produced in real time. This way, software can be written, tested and initially verified without having to set up and run the robot. Bag playback is especially helpful when testing the mapping and vision algorithms to visualize and reproduce failure cases.

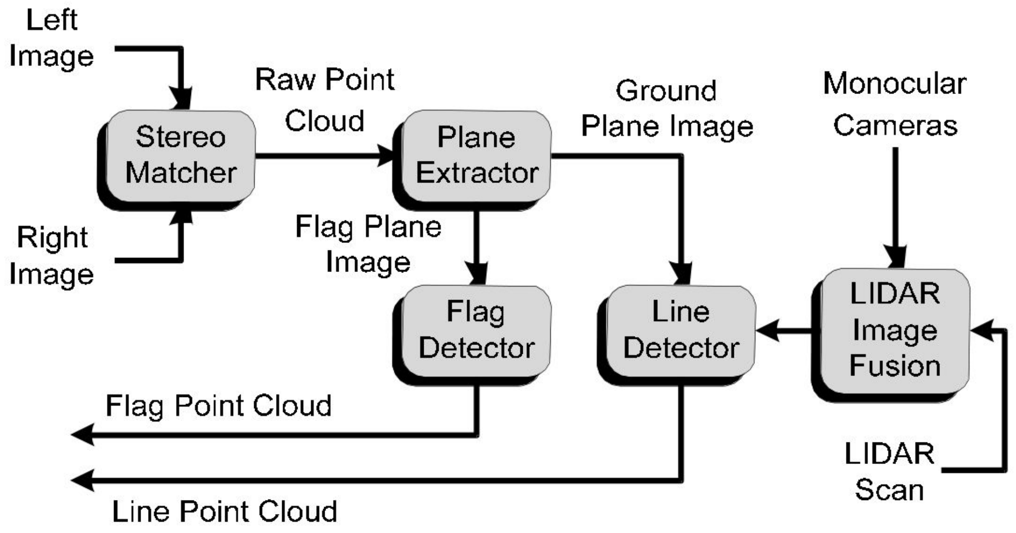


Figure 12 - Block Diagram of the Vision System Modules

Another convenient debugging feature is the reconfigure GUI. This is an ROS node that allows users to change program parameters on the fly using graphical slider bars. This tool is invaluable since most robotic vehicle controllers require precise adjustment of several parameters; therefore, being able to change these parameters while the program is running proves to be very beneficial.

Simulation

Gazebo is an open source simulation environment with a convenient interface to ROS. To rigorously test Schildkröte's artificial intelligence, simulated IGVC courses were constructed. These courses contain models of commonly encountered objects: grass, lines, barrels, and sawhorses. The configurations are designed to emulate the real IGVC course as accurately as possible. The simulation environment has proved to be extremely useful to the development process, since, unlike recorded data, the simulation responds to robot decisions and generates appropriate simulated sensor data.

Figure 12 provides a block diagram overview of the vision pipeline. A front facing stereo camera pair in the form of a OrSens camera and a LIDAR scanner comprise the sensors utilized by the vision system. From the stereo pair input, the stereo matcher feeds the plane extractor with 3D point cloud data that is filtered by the plane extractor into domain specific height data used by the flag and line detection units. The flag and line point clouds outputs are inputs to the navigation system.

Stereo Vision

Most LIDAR sensors can only detect objects on one plane; however, the Velodyne LIDAR on Schildkröte can detect objects in 16 planes from -15 to +15 degrees - yet this is not precise enough to obtain small obstacles such as flags. At past competitions, this limitation caused problems, especially in the case of the sawhorse-style obstacles, seen in the foreground of the image in Figure 13. The horizontal bar of the obstacle would not be in the scan plane, thereby going undetected, and the vehicle would frequently try to fit between the two legs of the obstacle.

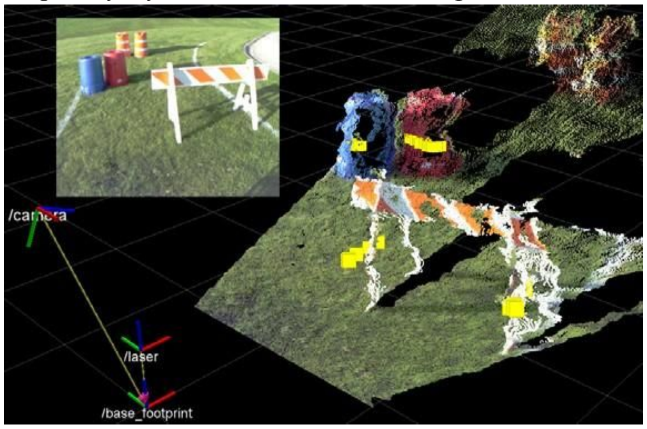


Figure 13 – Example Stereo Point Cloud

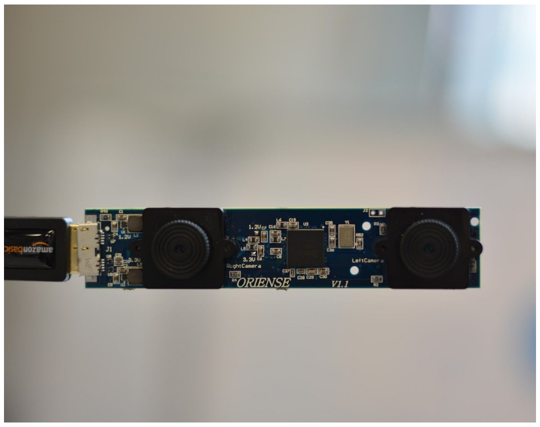


Figure 14 - Oriense-Tech OrSens camera

Schildkröte addresses this severe sensor limitation by using stereo vision. Applying open-source functions for stereo image processing, the image and depth from the stereo camera pair are processed to generate a 3D cloud of points corresponding to everything in the frame. An Oriense-Tech OrSense stereo camera is mounted to the top of the robot. Some obstacles of uniform color do not have enough texture to find confident matches between images and, due to this, data near the center of the uniform obstacles tends to be absent. Edges, grass and everything else is reliably detected and mapped to a 3D point. Figure 13 shows an example image and the corresponding point cloud. Each point in the cloud is marked with the color of the image pixel is corresponds to and the LIDAR data is then transformed on top of the point cloud as indicated by the yellow squares.

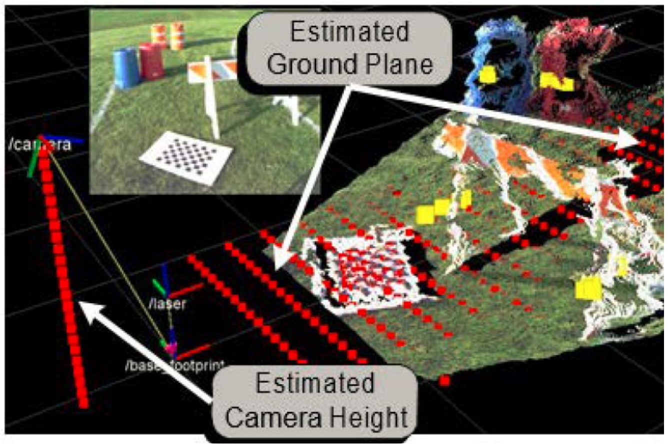
Automatic Camera Transform Calibration

To calibrate the transform from the cameras to the ground, an automatic algorithm was developed using a checkerboard. Assuming the checkerboard is flat on the ground, the 48 vertices in an 8x6 grid are optimally fitted to a plane equation to detect the camera position and orientation. This transform is used to place stereo and LIDAR data on the map from different coordinate frames. Figure 15 shows an example of how the transform calibration is performed on a real image. The calibration algorithm has proven to be very reliable and yields very accurate point clouds.

Plane Extraction

The plane extractor is responsible for analyzing the raw point cloud output from the stereo matching algorithm. The goal is to generate images that contain only pixels within a certain height window and black out the rest. Specifically, the two planes of interest are the ground plane, where the lines will be detected, and the flag plane, which is used for the flag detection algorithm. The height window to detect points on the ground plane is adjusted according to experimental results and the height window for the flags is set according to the expected height of the flags on the course.

Additionally, the height information of obstacles is used to eliminate obstacle pixels from the generated ground plane image in order to make the line detection algorithm more robust. Figure 17 shows examples of ground plane image generation, where pixels corresponding to points above the ground are blacked out, as well as regions of the ground pixels that correspond to where objects meet the ground. Notice how the resulting image primarily contains just grass and line pixels.



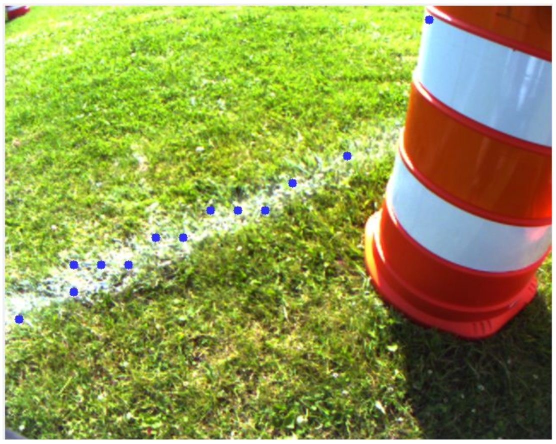
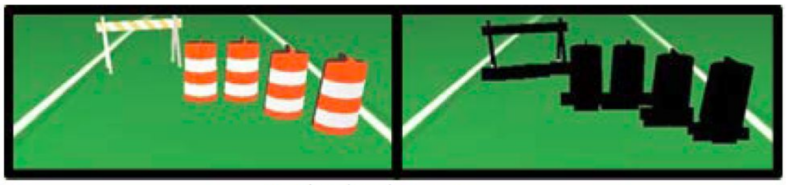
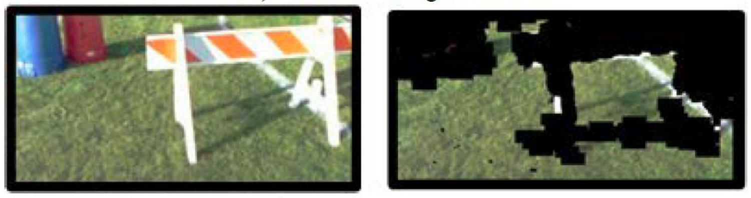


Figure 15 - Automatic Transform Calibration Procedure **Figure 16 - Line detection**



a) Simulated Images



b) Real Images

Figure 17 a & b - Example ground plane images in simulated and real scenarios.

Line Vision

Based on our previous experiences designing the lane detection algorithm for the IGVC path, noisy images were consistently found to be a problem. The noise in the images, dead grass for example, makes it almost impossible to design a robust algorithm to detect the white lane without being affected by the noise. For the example image given in Figure 16, detecting white lines with traditional image processing techniques is very complicated. For this reason, it was concluded that an intelligent and self-learning algorithm was needed to detect white lines with less sensitivity to the noise. The Artificial Neural Networks (ANNs) have been used to train Schildkröte to extract while lines in the image and therefore discard most of the noise in the image.

Every neural network has a learning algorithm that modifies and tunes the weights of the network during training. The self-tuning of the ANN is related to the learning function that measures the prediction error of the network. The learning algorithm in this case is the least mean square algorithm, in which the network tries to achieve the minimum target error based on the test features passed to the network. This technique of learning is called a supervised learning occurring on every epoch (cycle) through a forward activation flow of outputs as seen in Figure 19.

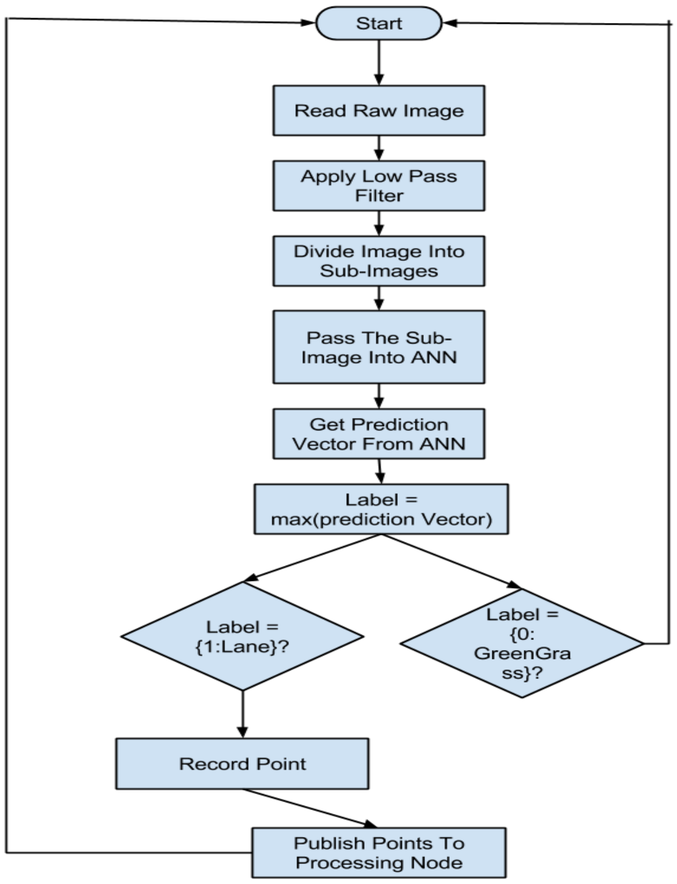

Start
Read Raw Image
Apply Low Pass Filter
Divide Image Into Sub-Images
Pass The Sub-Image Into ANN
Get Prediction Vector From ANN
Label = max(prediction Vector)
Label = {1:Lane}?
Label = {0: GreenGrass}?
Record Point
Publish Points To Processing Node

Figure 18 - Training process flow

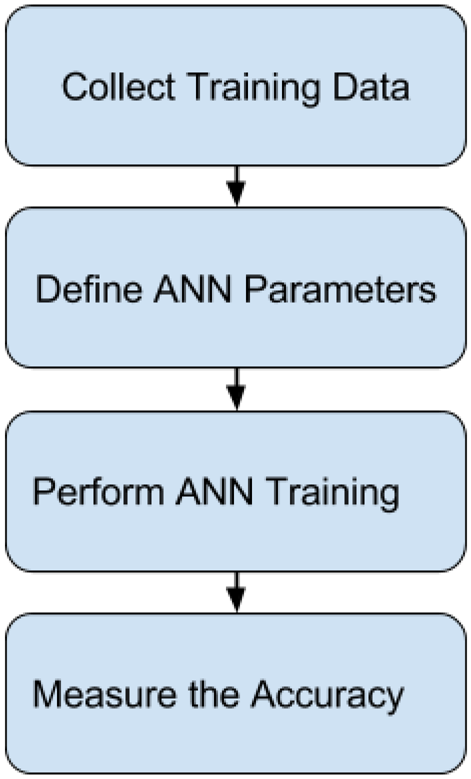


Figure 19 - Vision logical flow

Prediction

Every received raw image is divided into a set of sub-images. Every sub-image is passed to the ANN model to predict the label it belongs to. The returned parameter from the prediction algorithm is a vector, whose length equals the number of the labels. Every index of the label has a value that indicates the likelihood that image belongs to that label. The label that has the maximum value is the winner label. If the winner label is a white line, a point is recorded into a vector of points to be published to the navigation algorithm.

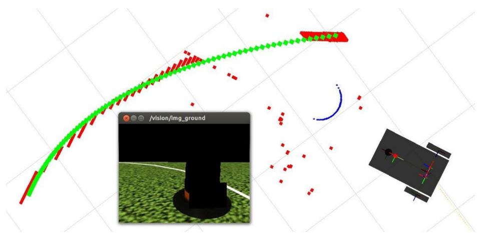


Figure 20 - Example Lane Detection

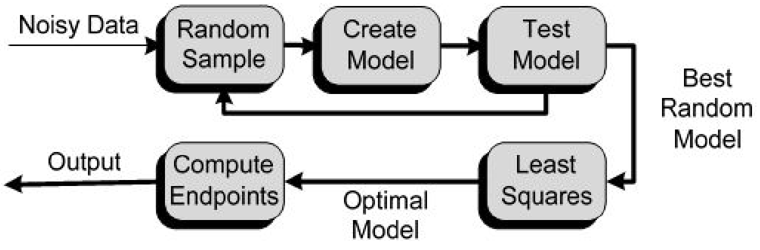


Figure 21 - RANSAC Block Diagram

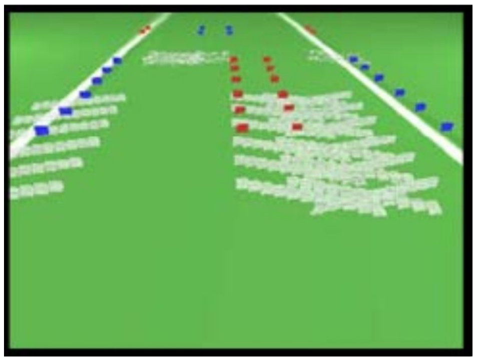


Figure 22 - Example Flag Negotiation Logic.

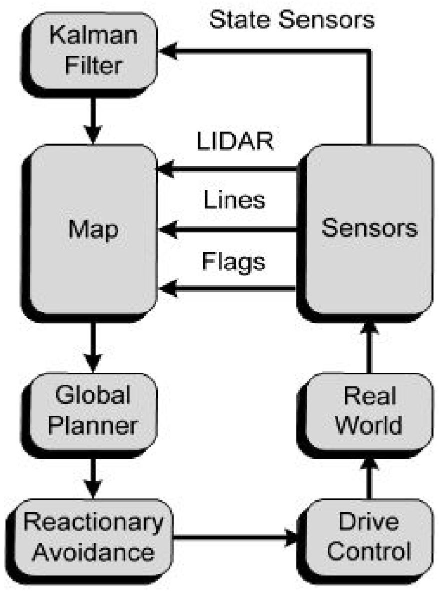


Figure 23 - Diagram of the Navigation System.

Model fitting is especially effective with dashed lines, occlusions and gaps. Even though parts of the line may be missing, the parts all fit a single 2nd order polynomial model. Other approaches such as clustering would separate each piece and leave gaps. Figure 20 shows an example scenario where a barrel blocks part of the line from view, however RANSAC was able to fit a 2nd order polynomial to the line and fill in the gap. In Figure 20, blue is LIDAR data, red is the projection of all pixels determined to be close to white and green is the optimal model fit output.

Flag Detection

The flag plane image from the input is fed to the flag detector. In this image, red and blue flags are separated by thresholding hue. To direct the robot towards the correct path, artificial lines are drawn on the map. Red flags draw to the right, and blue flags draw to the left. This blocks invalid paths and funnels the robot into the correct path. Figure 22 shows a simulated flag scenario illustrating this approach, where artificial lines are shown in white.

NAVIGATION SYSTEM

Kalman Filtering

The Kalman Filter fuses data from many sensors to accurately estimate position and orientation. Each sensor updates at a different rate, and the filter updates with the fastest sensor, 200 Hz. This results in accurate dead-reckoning between slow GPS updates. To avoid the discontinuity of traditional Euler angles, the orientation is represented using a quaternion. In the two dimensional case, the yaw angle can be represented by a 2D vector. Figure 23 shows information about the sensors being fused together, and which state variable each is measuring.

Mapping

The Kalman Filter was found to be accurate enough to build a map without Simultaneous Localization and Mapping (SLAM). SLAM requires a cluttered environment to match incremental data, but obstacles on the IGVC course are relatively sparse.

Schildkröte's mapping algorithm places time-stamped information on the map using the Kalman estimated position and orientation. The map is represented in 3D as 5 planar layers. Object information from three sources is placed on the map: 3D stereo data, LIDAR data, and detected lines. Line data is on the ground plane, LIDAR data is parallel to the ground plane and elevated, and 3D stereo data is present in all heights.

Each sensor can mark and clear space on the map. Clearing is done by tracing a ray through 3D space from the sensor source to the obstacle, and clearing every cell in that path. Two instances of the mapping algorithm, global and local, run in parallel. The local map is a 15 meter square with 5 cm resolution. The global map is a 100 meter square with 10 cm resolution. The global map can be initialized a priori with a map generated from a previous run.

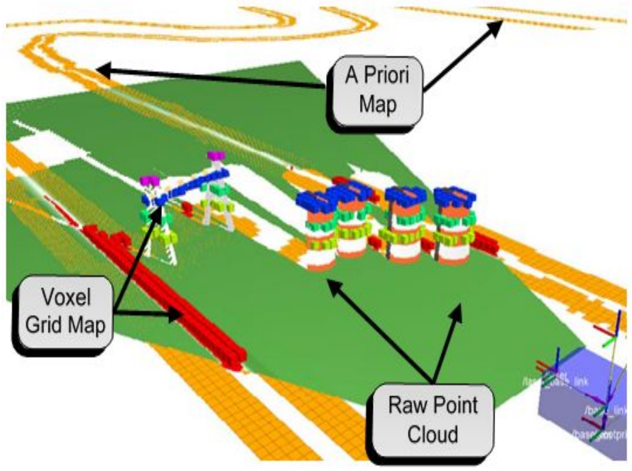


Figure 24 - Example of Mapping Procedure

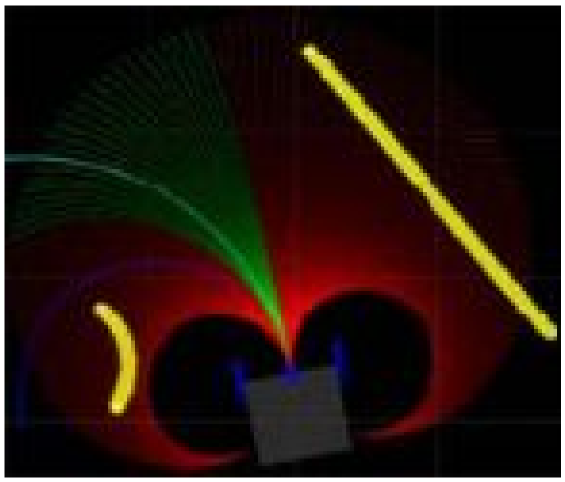


Figure 25 - Reactionary Avoidance

Global Path Planning

The global planner uses the global map to select the path to a goal point with minimum cost. The occupied squares on the map are inflated using the robot's width and an exponential cost function. The global planner is not allowed to plan a path passing through any inflated square. In Figure 24 the differently colored cubes represent the different layers of the 3D voxel grid map. The orange squares represent the inflated 2D projection of the 3D voxels onto the ground plane.

Furthermore, the cost function allows the path planner to choose the optimal path, even in both cluttered and open environments. The global planner implements Dijkstra's algorithm, and is an open-source package built in to ROS.

Reactionary Avoidance

Reactionary avoidance uses the local map to avoid collisions. This is the last stage in the path planning, and is responsible for overriding commands that would cause a collision. Future collisions are detected by simulating trajectories along evenly spaced turning radii. The best trajectory is the one closest to the requested command that doesn't result in a collision. A safety factor is also applied to avoid driving unnecessarily close to objects. The speed of the trajectory is scaled inversely by the angular velocity to prevent quick turns which could smear data on the map. Figure 25 shows an example of the local reactionary avoidance paths.

PERFORMANCE ANALYSIS

Maximum Speed

Schildkröte's motors spin at 144 RPM at nominal load, so combined with 12.5 inch diameter wheels, the resulting maximum speed is 5.3 mph. This estimate correlates with the observed performance. Since this is slightly over the course's maximum speed limit of 5 mph, the embedded controller limits the max speed to 5 mph.

Ramp Climbing Ability

At nominal load, the drive motors provide 200 in-lbs of torque. Assuming a realistic vehicle weight of 100 lbs, this corresponds to a max slope of 18 degrees. However, experiments have shown that Schildkröte can handle much steeper slopes, up to approximately 30 degrees, although the motors will perform outside of the nominal operating envelope.

Reaction Time

The artificial intelligence systems were designed to handle data from the sensors at the sensor's maximum frequency, thereby allowing the robot to make new decisions at the slowest sensor sampling rate of 20 Hz or 50 ms. Sensor data rates are shown in Table 1 below.

Battery Life

The high capacity AGM batteries on Schildkröte provide a total of 39 AH. The sensor suite, controller board, and peripherals consume a total of approximately 1.5 amps. Testing has shown that the drive motors consume a total of 25 amps maximum in grass, the environment typically encountered at IGVC. Based on these observations, total battery life is approximately 1.5 hrs of continuous run time. The large battery capacity coupled with efficient electronics lends itself to extended testing and runtime.

Obstacle Detection

The Velodyne LIDAR has a range of about 100 meters, and has 16 360-degree scanlines at angles between -15 and +15 degrees. The stereo cameras are oriented to see 4 meters away from the vehicle.

GPS Accuracy

The Trimble BX982 GPS receiver is accurate to within 1 centimeter using the RTK technique. It also utilizes a second antenna to provide heading information. Position updates are sent at 20 Hz, but the Kalman filter algorithm fuses the GPS readings with the rest of the sensors to provide faster position updates.

VEHICLE EQUIPMENT COST

A breakdown of the cost of the components on Schildkröte is shown in Table 2.

CONCLUSION

Schildkröte has proven to be very rugged, efficient and reliable, performing well while driving on any kind of terrain. The new robot design shows promising results, and the Oakland University team has great confidence going into this year's competition.

Table 1. Sensor Data Rates

Sensor	Data Type	Frequency
Kalman Filter	Position and Orientation	200 Hz
Velodyne LIDAR	Obstacles	20 Hz
uEye Cameras	Lane Obstacles	20 Hz
OrSens Stereo Camera	Obstacles (flags)	30 Hz

Table 2. Cost Breakdown of Vehicle

Item	Cost
Trimble BX982	$5000
Two Lenovo Laptops	$3000
Velodyne VLP-16 Lidar	$8000
uEye Camera x2	$917
Camera Lenses x2	$150
Oriense-Tech OrSens Camera	$200
12V Battery x2	$320
Motors and wheels	$500
Frame Material	$650
Sabertooth 2x60 Motor Controller	$190
Wheel Encoders	$480
Total	$19407

ACKNOWLEDGMENTS

The Oakland Robotics Association would like to thank several people and organizations. The team first of all thanks their advisor, Dr. Ka C. Cheok, for his invaluable advice and guidance. Thanks also to the School of Engineering and Computer Science (SECS) and its dean, Dr. Louay Chamra, for generous funding and lab space to work in. Immense thanks to Peter Taylor, machinist in the SECS for his help and guidance in the fabrication of Schildkröte. Additionally, without guidance from our past teammates we would not have the solid foundation to necessary to build this complex system. They are many, but we would like to especially thank Kevin Hallenbeck, Micho Radovnikovich Ph.D., Steve Grzebyk, and Lincoln Lorenz. Finally, thanks to our external sponsor Denso Corporation whose donations of certain components of the vehicle were critical to its development.

REFERENCES

1. IGVC Rules Committee, "IGVC Rules 2015," http://www.igvc.org/rules.htm (Accesssed April 5, 2015).
2. Non-linear RANSAC method and its utilization. Radek Beneš, Martin Hasmanda, Kamil Říha. December 2011, Electrotechnics magazine ISSN 1213-1539, Vol. 2. 4.
3. Dimension Engineering, "Sabertooth 2x60 User's Guide," https://www.dimensionengineering.com/-datasheets/Sabertooth2x60.pdf (September 2011)