



## Intelligent Ground Vehicle Competition 2008

### TEAM MEMBERS

Allen Akroush, Gregory Czerniak, Edward Klacza, Lisa Linna,  
Anthony Lucente, Vimlesh Shukla, Jason Smith





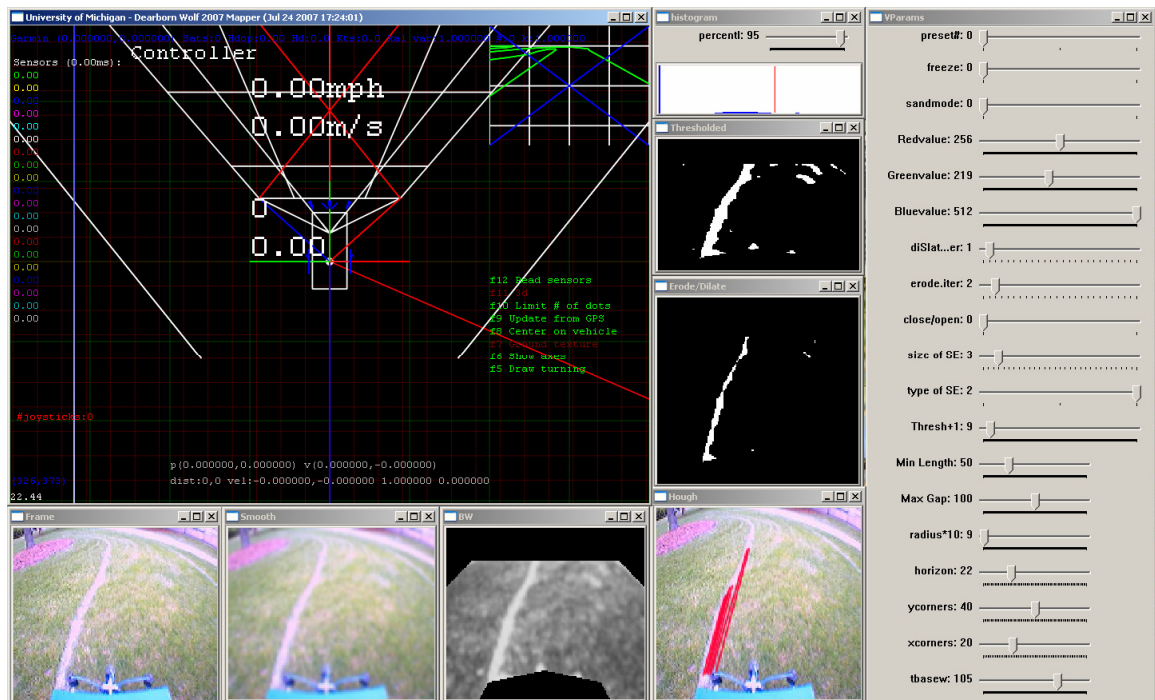
## 1. Introduction

UM-D Wolf is a multi-purpose autonomous vehicle that can function within a variety of ground-based operations. It was designed and built by the Intelligent Systems Club, a student organization at the University of Michigan-Dearborn. UM-D Wolf was originally entered in the 15<sup>th</sup> Annual IGVC competition, which took place in 2007. The current version of UM-D Wolf inherits the same chassis and hardware, but has an improved image processing system.

UM-D Wolf is highly maneuverable, with a zero-turning radius and a low center of gravity. At the front of the vehicle is a robust array of ultra-sonic and infrared proximity sensors, which are used to detect obstacles and prevent collisions. A high-speed firewire webcam is used for lane detection, which is mounted above the robot. The chassis can be adapted for a variety of applications as new technologies become available. Additionally, multiple strategies were taken on all fronts throughout the design process to ensure safety.

## 2. Design Innovations

UM-D Wolf's vision system has been re-structured to run more efficiently and to facilitate the calibration of important image-processing parameters in the testing phase. This is accomplished through a collection of GUI-controls, which can be easily removed through compiler directives to enhance speed. The following image shows how GUI controls were used to calibrate the image processing pipeline.



The new image-processing pipeline has a unique approach to how morphological operations are used in feature extraction. Due to perspective, lane markings at the bottom of the image appear thicker than lane markings at the top. It was found that lines of uniform thickness are more desirable, so the top and bottom portions of the frame are now handled differently in the morphological stage of processing.

UM-D Wolf still uses the Hough transform for line detection, but the new software does post processing to extend the length of detected lines towards the vehicle. The purpose of this is to make broken lines appear more continuous to the control module.



## **3.1 Team Organization**

The design and implementation of Wolf was done by both undergraduate and graduate students of the Intelligent Systems Club. Each member was assigned a primary responsibility; however, the group frequently worked together when troubleshooting and for problem solving. Although decisions were made by individuals for their respective tasks, the group as a whole often made critical design decisions.

## **3.2 General System Design**

The Wolf team used an eight-step process to design the vehicle:

1. Identify the problem: This step required the team to know the specifications that had to be met and the constraints placed on the team. The main constraints that the Wolf team encountered were time, funding and manpower.
2. Brainstorm: At this stage of the design the constraints were not a factor. The designer was free to think about any solution. The key was to bring as much creativity to the problem as possible.
3. Choose the best solution: The third stage, choosing the best solution, was meant to bring the engineer back to earth. The constraints, which had been ignored earlier, were taken into consideration.
4. Prototype: Fourth, a prototype was built and tested. This meant building test circuits, developing computer programs, or creating computer simulations.
5. Test and Evaluate: The fifth stage was testing and evaluation. During this stage the engineer had to determine if the proposed solution would solve the problem. Data was





gathered and compared to the specifications in order to determine if further refinement was needed or if an entirely new solution was required.

6. Fix design flaws or choose a different solution: Any changes in the design occurred during the sixth stage. The sixth stage was also the point where a new design was selected if the current design was inadequate and could not be modified to solve the problem.
7. Build the final design: When the solution was adequate for the problem at hand, the final design was built. In this stage circuits were packaged and code was properly commented and organized.
8. Integrate: Finally, in the eighth stage the subsystems were integrated into the robot. Integration not only involved putting the system together with other systems, but also testing to ensure trouble free operation.

### 3.3 Performance

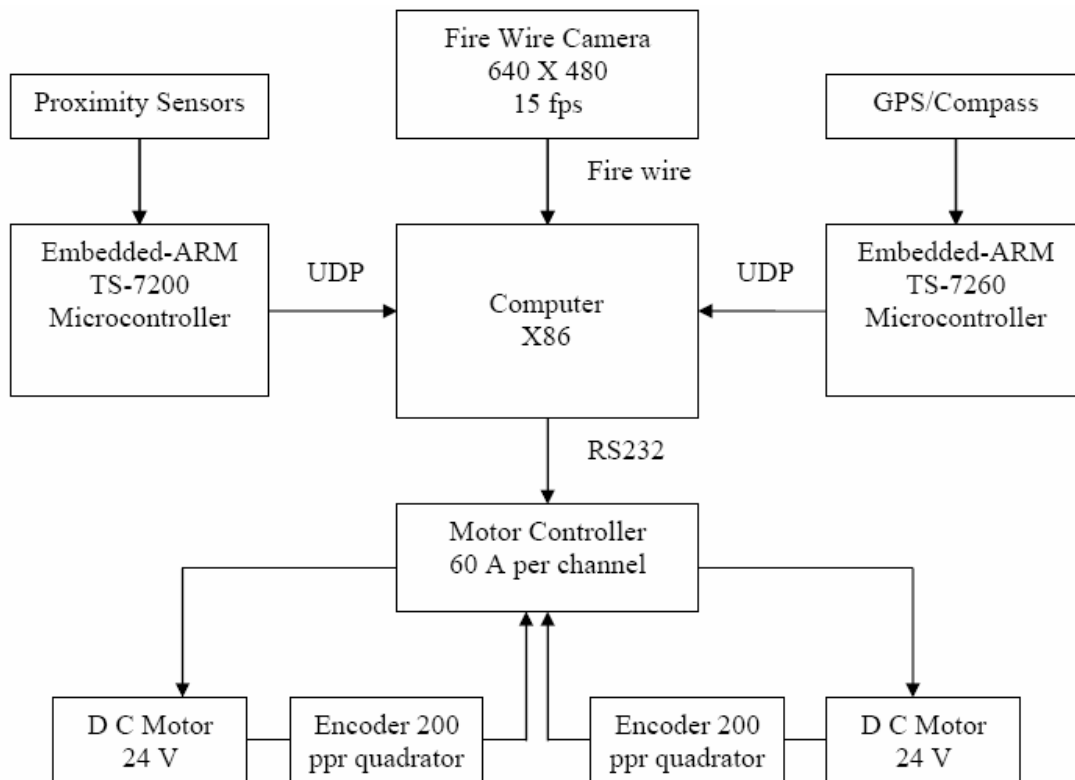
The following table shows design predictions that were decided before any design was implemented. The design did not deviate from these design constraints:

Attribute	Design Prediction
Maximum Speed	5.0 mph
Climbing Ability	30 Degree Ramp
Nominal Power Consumption (Watts = Amps x Volts)	240 watts
Battery Operating Time (24v 55AH Battery System)	6 hours
Distances at which objects can be detected	20 Feet
Waypoint accuracy (DGPS)	<3 Meters WAAS
Reaction Times	100ms
How vehicle deals with complex obstacles	Reactive Fuzzy Logic



## 4. Hardware Design

The hardware of the robot consists of three sections; the drive train, the control and the perception. The robot has a custom chassis, which was tailored to meet the IGVC requirements. The chassis is made of welded metal bars and sheet metal. The platform is strong enough to hold over 150 lbs and sturdy enough to make sharp turns at 5 miles an hour. The whole system is compact and easily maneuverable. Two electric wheel chair motors are used to drive the robot. The control part consists of a Roboteq AX2580 motor controller. It also has optical encoders to give precise information about the speed. The motor controller is connected to a laptop with an Intel dual core processor. Embedded-ARM microcontrollers parse data from proximity sensors and the GPS unit and communicate with the laptop via UDP packets. Frames for vision processing are captured from a Uni-Brain Fire-I webcam, and image processing is performed directly on the laptop. The following diagram illustrates the system architecture:



System Architecture

## 4.1 Motor Controller

UMD-Wolf uses a two channel Roboteq Ax2850 Motor Controller, with each channel powering one side of the robot. Each channel is capable of supplying a maximum of 120 Amps for up to 30 seconds. Two 60 amp fuses were placed on each channel to prevent over current from destroying the motor controller. Two industrial waterproof optical encoders are used for speed and odometry. All information is then sent back to the robot's main computer through a serial cable.





Starting speed is controlled to eliminate high inrush currents; if the motor speed is not ramped, these currents can reach up to 100A. By changing the speed, the maximum startup current is limited to 25A and only requires approximately 8A to stay running. Power distribution to the motors is also controlled. If the motor on channel 1 runs slower than the motor on channel 2, the motor controller can be programmed to send a more power to the first motor so that the motors on both channels will spin synchronously.

## 4.2 Motors

Extensive research indicated that a widely available wheelchair motor was the most appropriate choice for our vehicle. These motors run at approximately 140RPM and have a 12.5” wheel connected to them, allowing us to achieve the 5 MPH speed limit. Two styles of these motors are created, so that they can be mounted on both the left & right of the robot. The motors are rated as follows:



Permanent magnet, reversible, rated for operation on 24 VDC.
The no-load speed is approx 140 rpm with a no-load current of approx 3.6 amps.
The rpm is about 130 rpm with a load of 60 in-lb @ 7.9 amps.
Max rated torque is 120 in-lb with an output 94 rpm @ 13.2 amps.
Gear ratio of 32:1.

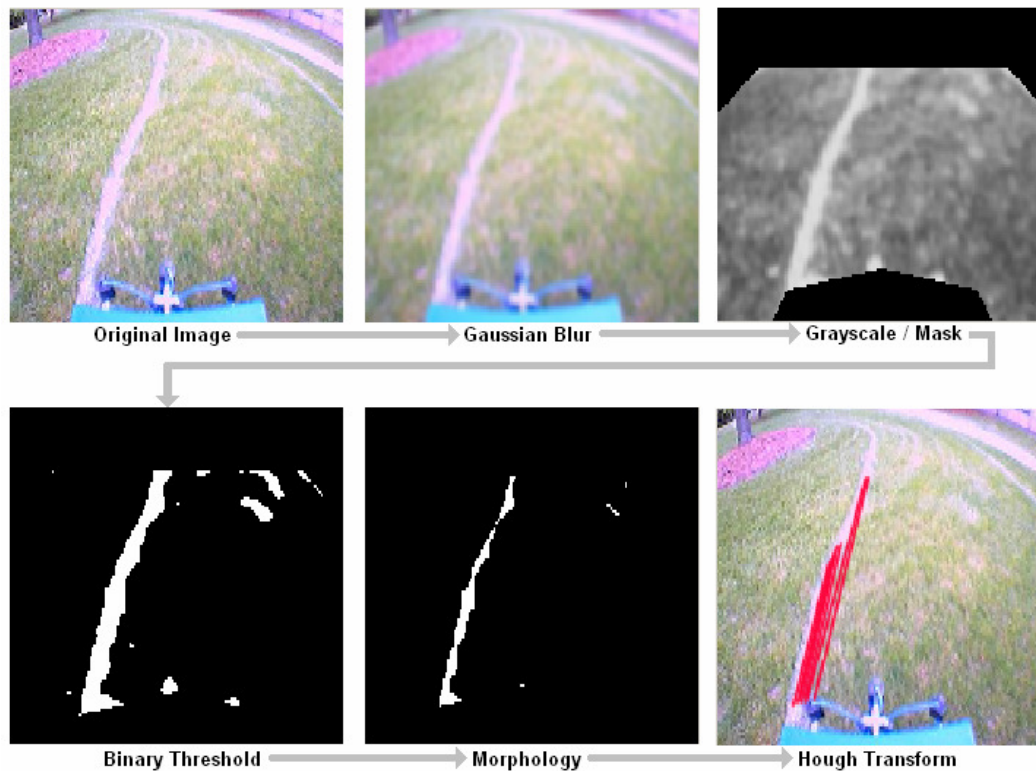




## 5. Image Processing and Lane Detection

Image processing is performed on the laptop, using a Uni-Brain Fire-I webcam and the Intel OpenCV library.

Several operations are necessary to extract lines from the captured image. First, the image is convolved with a 5x5 Gaussian kernel to remove noise. Next the image is converted to grayscale via a weighted mixture of RGB to make white lines more prominent. Following that, undesirable features such as the horizon and the sensor array at the front of the robot are masked out. Next, a histogram is computed and the 90<sup>th</sup> percentile is retrieved. The grayscale image is then thresholded at the 90<sup>th</sup> percentile, resulting in a binary image. Morphological operations are then performed on the binary image to erode any remaining noise. Finally, lines are extracted with the Hough transform and sent via UDP to the control module. The following diagram illustrates this chain of operations. At the final stage detected lines are shown in red.



## 5.1 Grayscale Conversion

The image from the camera is in 24 bit RGB (red, green and blue) format. Each pixel of the RGB image is composed of three 8 bit values, with each value corresponding to the intensity of red, green and blue. To reduce the amount of data that has to be processed, the image is converted to gray scale. The conversion process uses information from the green and blue channels in a way that helps isolate lane markings from the grass. It was found that by subtracting 1/3 of the green channel from the blue channel, white lines could be isolated and brightness from dead and brown grass could be reduced.



## 5.2 Binary Threshold

After masking out the horizon and other extraneous items, a histogram is computed and a global binary threshold is applied at the 90<sup>th</sup> percentile. This removes most of the grass by setting lower intensity pixels to zero in the binary image. Higher intensity pixels are set to one in the binary image. The resultant image consists mostly of the white lane marking, however dead grass and other features can occasionally be of sufficient brightness to be above the threshold.

## 5.3 Binary Morphology

Because dead/brown grass is more sparse and less connected than the white lane-markings, morphological erosion is applied to reduce or eliminate them. It is also beneficial to make the thresholded lane markings thinner, so that when the Hough transform is computed there are fewer lines for the robot to sort through and process. Because of the camera perspective, lane markings appear thicker at the bottom of a frame. To make the lines have a more uniform thickness, the frame is split into two sections, and the bottom third is eroded more than the top.

## 5.4 Hough Transform

The Hough transform is the final stage in the image processing pipeline and is used to extract straight lines. Since it is expected that the image will have only a few long linear segments, the probabilistic Hough transform is more computationally efficient. Extracted lines are sent to the control module via UDP for lane following.

## 5.5 Lane Following



The control module counts the number of lines that are located on the left half of the field of view and compares them to the number on the right. If there are more lines on a particular side, the robot will steer in the opposite direction. Although the system is purely reactive, testing shows it to be an effective approach.

## **6. Obstacle Detection**

A fully functional platform for onboard distance measuring sensors which will help control Wolf was implemented. Due to budget constraints, the team was not able to go with the obvious choice, which would have been a laser range finder which cost upwards of \$5000. We needed a cheap, efficient and reliable system to prevent the robot from hitting any of the obstacles at the competition. Ultra-sonic and infrared sensors were chosen to address our needs and provide us with the ability to compete with teams that had the much more advanced and costly laser range finders.

In previous years, the Intelligent Systems Club at the University of Michigan-Dearborn limited itself to using only a webcam for both obstacle avoidance and line detection. Due to an increase in the difficulty of the competition race each year, this has proven unreliable. We needed a better solution for determining the obstacles located in front of the robot. Our competitors over the years had already decided on an alternative to a solely camera controlled robot and purchased a laser range finder. Although these sensors provide prodigious data on everything located in front of the robot, they are extremely expensive and were not an option for the team to obtain at the time of the competition. The team decided to create an impenetrable barrier with ultrasonic and infrared sensors in front of the robot. These two types of sensors are



inexpensive and are more reliable than just using a camera. Furthermore, they will be able to provide us the ability to get through the course without hitting anything.

The obstacle avoidance system uses five Maxbotix Maxsonar EV1's and three Sharp GP2Y3A002K0F infrared sensors. While code and calibration values were created for the GP2Y3A003K0F infrared sensor, the decision was made to go with three of the GP2Y3A002K0F IR sensors. All these sensors interface with an embedded-arm TS-7200 microcontroller using the analog to digital converter and the digital I/O pins. The Maxsonar Ev1 has a measured range of 0- 254 inches, the Sharp GP2Y3A002K0F has a range of 20-50cm and the GP2Y3A003K0F has a measured range of 40-300cm. These eight sensors are strategically placed around the robot to create a virtual barrier so that the robot can avoid obstacles that may be in front of it. The distance measurements are sent to the robot main computer through UDP to be processed by the mapping program.

## 7. Navigation

To travel to selected waypoints, a least cost path planning algorithm was used to calculate optimum changes in heading. Position and heading were calculated by combining successive GPS readings, dead reckoning from wheel encoders, and applying a kalman filter to reduce errors. If an obstacle is encountered, the system records its position on the map and plans a path around it and any other obstacles it may have previously encountered. GPS data was retrieved from a Garmin 16a. Since the GPS uses a WAAS based correction service, GPS readings can have a margin of error that exceeds 1 meter. To compensate for this, the robot circles back through the area it traversed before heading toward the next waypoint.





## **8. Safety, Reliability and Durability**

Our team wanted to ensure the safety, durability and reliability of the vehicle and as a result, the following designs were implemented:

### **8.1 Manual emergency stop**

The manual emergency stop consists of red push button to stop the vehicle immediately. Pushing the stop button cuts off the power to the motor and locks the wheels by turning on the electronic brakes, bringing the vehicle to an immediate stop.

### **8.2 Wireless emergency stop**

A wireless remote keyless entry unit with a range of 100ft (30 meters) was modified to stop the vehicle remotely. When the remote button is pressed, power is cut to the motor, stopping the motor immediately.

### **8.3 Software Protections**

If a critical system should fail, the robot will enter safe mode. For example, if the robot's E-stop is hit and the motor controller turned off, the code will know that the motor controller is not present. The robot will not operate again until the motor controller is online. Also, if a ½ second goes by and the robot has not received a UDP packet, the robot will go into safe mode and stop. This prevents the robot from running away in an autonomous and remote control situation. If the robot becomes out of range, or the transmitter loses power, then the robot will simply halt until it gets a command instead of carrying out its last command.



## 9. Detailed Cost Analysis

Description	Cost	Quantity	Total
Chassis	\$250.00	1	\$250.00
Wheels	\$40.00	2	\$80.00
Casters	\$20.00	2	\$40.00
Laptop Computer	\$1,000.00	1	\$1,000.00
Embedded ARM Microcontrollers	\$555.00	2	\$1,110.00
MaxSonar-EZ1-Ultra-Sonic Sensor	\$125.75	5	\$125.75
5-Beam IR Sensor	\$52.00	3	\$156.00
9" IR Proximity Sensor	\$12.50	8	\$100.00
Motor Controller	\$700.00	1	\$700.00
Fire Wire Camera	\$173.45	1	\$173.45
Mounting Supplies	\$49.47	1	\$49.47
Batteries	\$159.95	2	\$319.90
Painting/ supplies	\$28.90	1	\$28.90
		<b>Total</b>	<b>\$4,133.47</b>

## Acknowledgments

The team would like to thank Larry Sieh for his help and support for the project.

## Faculty Advisor Certification

I hereby certify that the engineering design in the vehicle by the current student team has been significant and equivalent to what might be awarded credit in a senior design course.

---

N. Natarajan  
Associate Professor,  
Department of Electrical and Computer Engineering  
University of Michigan-Dearborn