Vision System for Wunderbot IV Autonomous Robot
EGR494: Senior Project in Computer Engineering

James G. Painter
Department of Physics and Engineering
Elizabethtown College
Elizabethtown, Pennsylvania
Email: painterj@etown.edu

***Abstract*— The Robotics and Machine Intelligence Club of Elizabethtown College has maintained an ongoing autonomous robot project for nearly seven years, but the robot has yet to feature an effective vision system for unmanned navigation. Here we describe the steps taken toward the development of such a system, including physical mounting of a camera, image processing, and motor control. The resulting vision system establishes a platform for competing at the 2008 Intelligent Ground Vehicle Competition.**

I. Introduction

Autonomous robots serve mankind in areas ranging from search and rescue to space exploration. In attempt to elicit new, creative designs from college students [1], the Association for Unmanned Vehicle Systems International (AUVSI) holds the annual Intelligent Ground Vehicle Competition (IGVC), an engagement of roughly 30 unique robots designed by colleges and universities from various countries, including the United States, Japan, Canada, and India. Their objective is to excel at complex autonomous tasks used to measure the robots' navigability and strength of design.

At Elizabethtown College in Pennsylvania, the Wunderbot autonomous robot project has been progressing for over five years. From the financial assistance and donations of numerous corporate sponsors, new equipment and software is added to the robot between competitions. Its primary objective is to compete in the IGVC, with the potential of administering future automated tours on the college campus. Wunderbot IV is led by the team of James Painter, David Coleman, and Jeremy Crouse, with support from Chris Yorgey and Daniel Fenton, and advised by Dr. Joseph Wunderlich. The project has become a staple of the school's Physics and Engineering Department, attracting prospective students and drawing the attention of local industry and media.

II. IGVC

The IGVC provides an excellent opportunity for students to explore the possibilities of unmanned vehicles. 2008 will mark the 16th anniversary of the competition and the third entry for Elizabethtown's Wunderbot. IGVC consists of four challenges for the autonomous robot, each outlined below [2].

A. Autonomous challenge

The autonomous challenge pits the robot against an outdoor obstacle course, traversed by remaining on a path of grass approximately three inches tall, bounded by spraypainted solid or dashed lines. The robot must avoid obstacles, including fences, construction barrels, trees, and shrubs. Potholes, inclines, and sand pits may also be strewn about the course. Scores are calculated based on the distance traveled through the course and the elapsed time.

The autonomous challenge will rely on the robot's vision system moreso than any other challenge, due to the imperative condition that the robot remain within two-dimensional white lines, which go undetected by the laser range-finder.

B. Navigation challenge

In the navigation challenge, a field of approximately one acre is marked with a number of GPS waypoints (approximately ten). Each team is provided with the coordinates in latitude and longitude of each waypoint. Obstacles similar to those on the autonomous challenge course may also be placed randomly on the navigation challenge course. Scores for the navigation challenge are determined by the number of waypoints traversed by the robot and the time taken to do so. This challenge will, for the most part, neglect the capabilities of the vision system. All obstacles can be detected by the laser range-finder, and the GPS/compass sensing will be responsible for finding the targets.

C. JAUS challenge

The challenge for Joint Architecture for Unmanned Systems, although not mandatory, demonstrates the robot's ability to communicate using a standardized wireless messaging system that is growing in popularity in engineering fields [3]. A section must be included in the written report that describes the robot's JAUS capabilities, and the robot must demonstrate a pre-defined working ability to communicate using the JAUS message type. The JAUS challenge will ignore the vision system entirely.

D. Design challenge

The design competition exists as a separate entity of competition, in that the robot's performance has no influence on the design score. The design competition measures the team's procedures, workmanship, and innovation to determine product quality. Each team must submit a typed report prior to the main competition date, detailing the conceptual design of the vehicle and its components, and emphasizing design changes from the team's previous contest entry as well as technological innovations that distinguish it from the rest of the field. Teams must also prepare a ten-minute oral presentation. The third component of the design challenge involves judges' hands-on examination of the robot, for such aspects as neatness, safety, originality, and style.

III. Related Work

The Wunderbot vision system's most closely-related work is in the IGVC competition itself. In any given year, roughly 30 other robots, all having identical objectives, are available for comparison. Following each competition, the organizers make all teams' design reports publicly available online, encouraging the spread of successful ideas. As such, we find trends in particular subsystems among the contest entrants. The laser range-finder, for example, has become standard for obstacle avoidance, having been built into nearly all of 2007's competing vehicles.

While many teams may share similar components, each year brings new innovation in overall design. Part of the initiative in this matter is the scoring of the design challenge. The scoring is partially dependent on the vehicle's display of a significant subsystem or software upgrade over that which represented the team previously.

The competition has seen many different vision configurations, as well as an assortment of corresponding software packages for image processing. Some teams opt for camcorders, some use industrial cameras, and others choose to mount webcams [4]. Stereo vision has given robots the advantage of a line of sight extending to the sides or rear of the vehicle [4]. Once images have been grabbed, teams have performed successful filtering through the use of Intel's OpenCV library [5], MATLAB [6], and LabVIEW [4]. Alas, the IGVC has no cookie-cutter winning formula. In fact, oft-champion Virginia Tech's traditional three entries per year are all structurally distinct from one another [7], a further testament to the competition's flexible path to success.

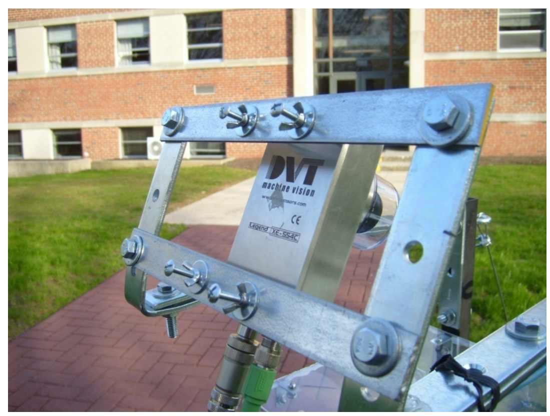
IV. Preliminary Work

Prior to the start of the project, a DVT Legend 554C XE high-resolution video camera was acquired and configured. In addition, a LabVIEW sub-VI was written to acquire the camera's TCP/IP communication string, in which is contained the manually-formatted results of any image processing performed. The camera is hard-coded to capture color images at 1280×1024 resolution. The image quality is far better than needed, but Section V describes how this can be used to our advantage by zooming out while still maintaining sharp objects in the distance.

V. Camera Mount

A steel camera mount, shown in Figures 1(a) and 1(b), was build atop the Wunderbot's utility pole, which also houses the GPS and digital compass. Two 30.5cm×2.5cm flat steel bars were fastened to the utility pole supports using L-brackets. At the far end, two more L-brackets were attached, making the entire camera mount extend about 40.5cm back from the rear bumper of the vehicle. The L-brackets were bent to form roughly 45-degree angles. Two 25cm×2.5cm steel bars were secured across the L-brackets in order to provide a stable mounting surface for the camera. Through the 25cm bars were inserted 10cm bolts that screw directly into the four threaded holes in the back corners of the camera. Wing nuts allow very precise fine-tuning of the angle at which the camera is directed. The data and power cables for the camera were concealed with the 2.5cm plastic conduit that runs along the utility pole.

(a)



(b)

Fig. 1. Wunderbot camera mounted on steel brackets.

The angle at which the camera was mounted played a crucial role in the eventual image processing step of the vision system. Mounting the camera farther from the front of the vehicle would widen and deepen the field of view. A more downward mounting angle would enable the camera to see directly in front of the bumper, while a more upward angle would extend the depth of view. This situation is illustrated in Figure 2. Another consideration was the image processing time on the software end of the system, which could be accelerated by trimming the edges of the rendered image. A larger field of view yields more unnecessary regions of the image, which can be eliminated to reduce the processing time.

Various configurations were tested, measuring the range of view and corresponding image processing times. For instance, with the camera positioned 1.2m directly above the rear

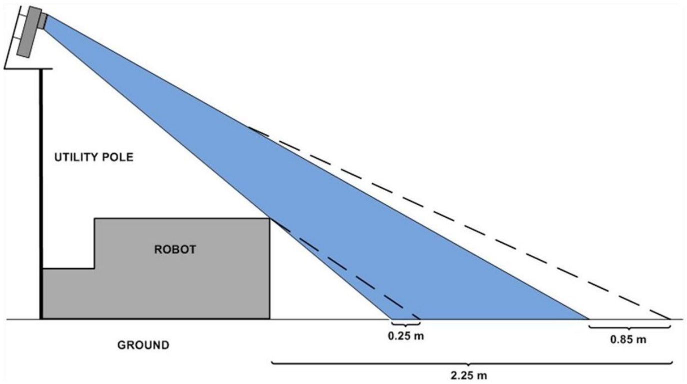
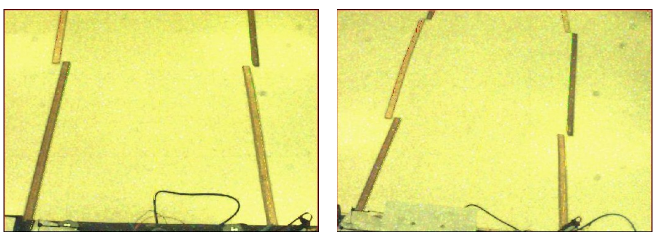


Fig. 2. Camera viewable region, with camera mounted directly above rear bumper (blue fill) and with camera shifted back 40.5cm from rear bumper (dashed lines).

bumper, the camera was able to see approximately 1.4m ahead of the front bumper, as depicted in Figure 3(a). With the camera at the same height, but 40.5cm back, the viewing distance was extended about 85cm to roughly 2.25m, at the expense of about 25cm lost directly in front of the bumper, as shown in Figure 3(b). This sacrifice was acceptable, since the tradeoff is either seeing farther ahead or trimming the top edge of the image to reduce processing time.



(a) (b)

Fig. 3. Viewable region of camera when mounted (a) directly above rear bumper, and (b) when mounted 40.5cm behind rear bumper.

Processing time reductions when trimming the image were significant enough to implement the feature. The percentage speedup was a nearly-linear relationship to the percentage of the image that was trimmed, and the final implementation incorporated the cropping of the top 200 lines, for a reduction of about 110ms in processing time.

Top Edge Cropped	Processing Time Speedup
15% (153 lines)	16% (90ms)
24% (246 lines)	25% (140ms)

VI. Image Processing

The vision system's image processing is performed from within the camera's proprietary software, DVT Intellect v2.2. First, an erosion filter is applied to the image, using a 3×3 kernel. This closes many holes of noise, such as small dirt patches that appear through the grass, while still maintaining the shape of the desired white lines, since the lines (including dashed lines) will always be wider than three pixels. Larger kernels could produce an even more accurate image, but processing times increase sharply as the kernel grows larger.

Once noise has been filtered, an Intellect "line thickness" sensor is applied. This measurement sensor first uses a variable 60% intensity threshold to deduce a binary image. The sensor then scans every row in the image to find the two edges closest either side. Optionally, all edges can be found and more accurately be used as input for the line fitting algorithm to follow; however, the extra computations lengthen the processing time roughly three-fold. To help eliminate noise, all edges less than 50 pixels wide are discarded. Next, a Hough Transform with resolution of four is performed on the detected edges in order to fit two lines, one closest to the left side of the image and one closest to the right. These final lines are measured for separation width, and the average of the two is measured for straightness, contrast, and angle.

Final line pass/fail conditions are used to filter shadows and other undesirable objects in the field of view. A maximum width condition of 300 pixels is combined with a "straightness" condition that will fail the test if the sum of the distances between the data point that is farthest away in one direction, and the one farthest away in the other direction, of the resulting average line.

A formatted string is sent via TCP/IP to the on-board PC. This string contains (in units of pixels), the dimensions of the viewing window, the x- and y-coordinates of the point on the left line with the lowest y-value (nearest to the robot), and the corresponding points on the right line. Once these are received by the PC, logic is used to determine the direction in which to turn.

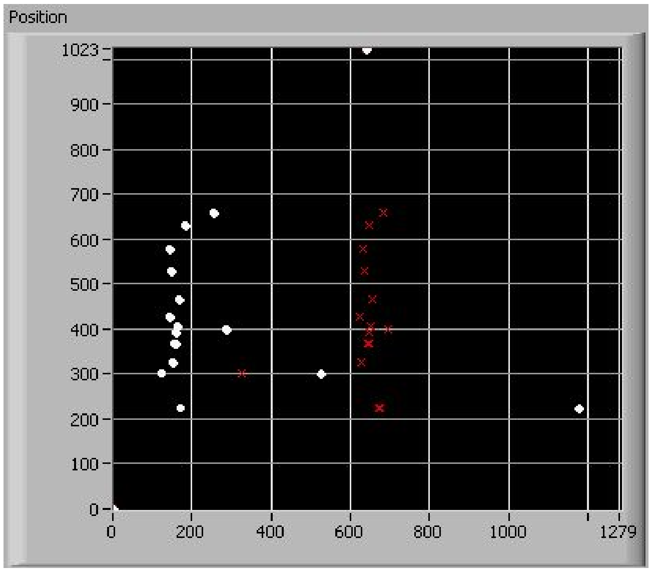
VII. Motor Control

LabVIEW 7.1 was used to develop the all cognition of Wunderbot IV. This section explains the method for turning the vehicle and for achieving accurate motion response.

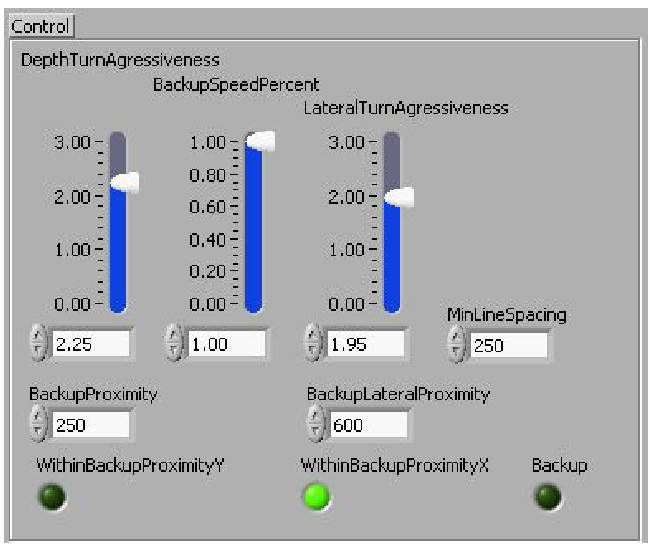
A. Turning

For responding to the white line positions parsed and sent by the camera, both the x- and y-coordinates are taken into consideration. Rough scaling factors for both the x and y direction were used to convert pixels to centimeters. Because the actual width of the view widens when extending outward, the scale is only an approximation. The scaling factors were then used to estimate the depth and width of the camera view. Once these measurements were obtained, they could be used to plot the detected lines on a local map with target locations, as seen in Figure 4(a). The physical locations of these points are critical in giving the Wunderbot a global sense of position, which is used to determine how sharply to turn away from white lines and how to coordinate with other sensing subsystems, such as the GPS, digital compass, and LIDAR.

In general, when two lines are found, the target location is the average of their x-coordinates and the actual value of their y. When only one line is found, the target becomes the point directly centered between that line and either the left or right edge of the viewable region. If the line is on the left, the target is placed on the right, and vice versa. If no lines are found, the target is placed in the center on the horizon, such that the robot will move directly forward at full-speed.



(a)



(b)

Fig. 4. (a) Detected white lines and calculated target points, both plotted on local map using pixel scale. (b) LabVIEW control panel with adjustments for vehicle movement.

Controls, shown in Figure 4(b), were designed in the main LabVIEW sub-VI to adjust the weight of both depth and lateral position of the lines as they affect the vehicle's degree of turning. Additional controls enable the user to adjust the proximity (both depth and lateral - both must met) within which a detected line will force the robot to move in reverse, and another control sets the percentage of the forward speed to use when backing up.

B. PID Controller

One of the most costly problems encountered during early testing was inaccurate vehicle motion. When operated on smooth indoor surfaces, Wunderbot was able to move roughly in the intended direction, but once the vehicle was tested outdoors on grass, motion response had a large degree of error. The largest cause for error is the front casters, which require a disproportional amount of force in order to change direction. This problem is an typical case for a PID controller to amend.

The PID closed-loop control was developed in LabVIEW and is very straightforward. The P, I, and D are all user-adjustable via the front panel, and feedback comes from the U.S. Digital optical encoders. Unfortunately, the robot itself is extremely difficult, if not impossible, to model via differential equations, hence classic methods of control theory could not be instituted to determine the value of the PID's constants. Instead, it was a trial-and-error procedure, which led to P=0.500, I=20.00, and D=0.001. Very subtle variations in the derivative constant led the robot to accelerate out of control. A PID controller's derivative constant in general is highly susceptible to noise, and therefore an adjustable low-pass filter was designed for the D [8]. This kept the D from fluctuating too rapidly, while still allowing it to quicken the output's rise time. The resulting transient response can be seen in the figure below.

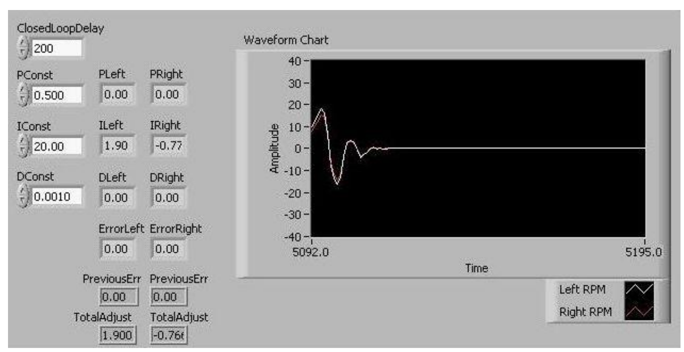


Fig. 5. LabVIEW control panel for PID controller with robot's resulting transient response.

VIII. Results

The effectiveness of the vision system, combined with a well-developed motion control system, was seen in several live demonstrations. Wunderbot IV was able to follow a white-lined, one-turn path in grass, albeit slower than desired competition speed. Future improvements include the creation of a global map, on which the position of the robot will be tracked by the optical encoders. In time, the global map will also incorporate the GPS navigation system and LIDAR in order to visually display all facets of the surrounding environment - target GPS points, white lines, and obstacles.

REFERENCES

[1] B. Theisen, “The 15th annual intelligent ground vehicle competition: intelligent ground robots created by intelligent students,” in *Proc. of SPIE*, vol. 6764, Sep 2007.

[2] B. Theisen, et. al., *The 16th annual intelligent ground vehicle competition (IGVC): official competition details, rules and format*, ”september 2008” ed., 2007.

[3] D. Carroll, K. Mikell, and T. Denewiler, “Unmanned ground vehicles for integrated force protection,” in *Proc. of SPIE*, vol. 5422, 2004, pp. 357–377.

[4] M. Bovard, et. al., “Design evolution of the Trinity College IGVC robot ALVIN,” *Journal of Robotic Systems*, vol. 21, no. 9, pp. 461–469, Sep 2004.

[5] M. Tedder, et. al., “An affordable modular mobile robotic platform with fuzzy logic control and evolutionary artificial neural networks,” *Journal of Robotic Systems*, vol. 21, no. 8, pp. 419–428, 2004.

[6] M. Tedder and C. Chung, “Autonomous robot vision software design using MATLAB toolboxes,” in *Proc. of SPIE*, vol. 5608, Oct 2004, pp. 99–106.

[7] AUVSI, “IGVC Design Reports,” 2007. [Online]. Available: http://www.igvc.org/reports.htm

[8] J. H. Lumkes, *Control strategies for dynamic systems: design and implementation*. Marcel Dekker, Inc., 2002.