**Vision**

For line detection, we use the LabView vision module. The first step is to collect the images. LabView has a VI which can collect images from the camera. Even though the device is capable of higher resolutions, we collect the images at 640x480 pixels to decrease the time needed to process the images.

We would like to determine the exact location of lines from the camera’s image. This requires a perspective transform. Using a grid with known distances, we were able to generate the transformation using a tool built into LabView.



When this transformation is applied to an image on the course, the image appears to be viewed from straight above, and distances on the ground can be determined. A color threshold is placed on the image to find the white lines. These areas are then dilated and eroded to remove specks while also filling up gaps in the line. A line detection algorithm then finds the lines. These lines are returned in real world coordinates, due to the perspective transformation.

**Sensor VIs**

One of the first things to occur in the code is to set up and start the sensors and motor controller. The majority of the sensors work similarly. All sensors call an initialization VI. These VIs not only initialize and configure the sensors, but they confirm that the proper COM ports were selected. The only sensor to not run off of a COM port is the camera; however its data is collected in a similar method. The GPS also includes the extra initialization steps of enabling WAAS and setting an approximate initial coordinate.

The sensors Vis start collecting data. All the collection loops are timed loops. This regulates the time spent by each loop and prevents one loop from hogging the processor. All the data is saved into global variables. These variables can be accessed from other VIs and processed as data is received. This allows the processing loops to run separately from the data collection loops. Finally, when the program is stopped, all the sensor’s serial ports are closed.

The data acquired is integrated together into the plan loop. Data is returned from a LIDAR in the form of arrays of angle and distance to the obstacle. The distances to the lines found in the vision system is then integrated into these arrays. The arrays (along with a goal direction) are then passed to a histogram VI, and returns the best angle to drive towards to avoid the goal.