|  |
| --- |
| **Rose-Hulman Institute of Technology** |
| **Status Report Week 8** |
| **CSSE463: IGVC Vision** |
|  |
| Ander Solorzano;Ruffin White;Kurtis Zimmerman;Donnie Quamme |
|  |

Rose Hulman Robotics Team CM 5000

5500 Wabash Avenue

Terre Haute, IN 47803

**Grass Filter**

The work done this week on grass filtering was done by applying a SVM to individual pixels in RGB space. After extracting thousands of pixels from representative images a script was developed to train a grass pixel classifier. Due to the large number of pixels used in the training the classification was exceedingly slow. Additionally the classifier was not able to perform exceptionally after being trained on only a single pixel. Work is currently being done to perform classification on a 3x3 neighborhood around the target pixel in hopes that it will provide enough additional information in the classification process.

|  |  |
| --- | --- |
|  |  |
| Original image and output image filtered image of the grass. | |

**White line detection**

Lane line detection is being handled using relative brightness to detect whiter, brighter pixels in an area of grass pixels, as presented in the GOLD Report. Like an edge detector, the algorithm identifies lines of sharp increase in brightness to mark off potential lane lines. Then Hough transforms are used to detect lines among the identified lane line pixels. The primary consideration thus far has been how to compute the brightness of a pixel. Given a pixel’s RGB values, we experimented with two formulas: the value in HSV space of the pixel and the perceived brightness calculated as the weighted distance in 3D RGB space () as given on the following page: <http://zoltanb.co.uk/how-to-calculate-the-perceived-brightness-of-a-color/> . The HSV method had a much higher accuracy, and since the algorithm used was developed with top-level images in mind, accuracy should continue to improve after we have the ability to remap the perspective on our images.

|  |  |
| --- | --- |
|  |  |
| Prominent solid white lines are recognized consistently, though not always as one solid line. Similarly, prominent dashed lines are identified as solid continuous lines due to the nature of the Hough transform. | |
|  |  |
| Even arcs can be identified fairly reliably as a set of solid lines, which is something we did not necessarily anticipate. One problem still is areas of faded lines or light dirt which interferes with the expected white pixel pattern. Several images in our test set with uniformly faded lines failed to be identified correctly. | |

**White line detection: Construction Barrels**

For this type of the project, the task was divided into 3 categories: detecting barrels via color classification, detecting barrels via pattern matching, and detecting barrels via stereoscopic vision and acquire spatial distance as well.

Using color classification techniques to find a barrel in a RGB image can be quite difficult and tricky due to the brightness of the image, varying sizes of the object of interest, different viewing angles, and background noise. The approach was examined using LabVIEW’s built-in tools for color matching. For now, the use of primitive VIs that would allow us to parameterize the input signals for better handling of the results was not used. Instead the VI Vision Assistant blocks (image processing and image detection built-in tools) were used to obtain quick results we would expect.

|  |  |
| --- | --- |
| C:\Users\solorzaa\Desktop\Barrels\barrel9.jpg | C:\Users\solorzaa\Desktop\Barrels\SR Results\figure1.JPG |
| C:\Users\solorzaa\Desktop\Barrels\barrel4.jpg | C:\Users\solorzaa\Desktop\Barrels\SR Results\figure2.JPG |
| The top row shows how the color classifier did manage to find the correct barrel locations in the RGB images. This is an ideal image since there is minimal noise from the background and the barrels are in the same plane (same size). The bottom row however, has more noise involved and the barrels in the background cannot be detected since the scores for those barrels are not high enough to be detected. You can also see how some False Positive readings. These positive readings can be manipulated using primitive VIs to acquire better results. | |

|  |  |
| --- | --- |
| C:\Users\solorzaa\Desktop\Barrels\barrel10.jpg |  |
| C:\Users\solorzaa\Desktop\Barrels\barrel9.jpg |  |
| C:\Users\solorzaa\Desktop\Barrels\barrel4.jpg |  |
| In this image, the obstacle detection via pattern matching is shown. Before pattern matching is performed, the image has the green color plane extracted to create the gray image. A template is then selected and the algorithm search everywhere in the picture to find matches and assigns scores to the possible matches. Matches with a high enough score will be detected. The top two rows show the classifier working appropriately and as expected but the bottom row still has some small issues. However, the bottom row did a better job dealing with the noise and it also detected small angle shifts from the sample image. | |

The next step would be to use a dynamic classifier that can handle and reject areas of the image in real-time (horizon line, bright spots, etc.) and to incorporate the stereoscopic vision for spatial locations.

**Stereoscopic Perspective Correction**

|  |
| --- |
|  |

This week Ruffin started implementing the perspectives and nonlinear distortion correction routines for converting raw captured images from the camera to a 2-D planar perspective correction image. This would allow the ground plane of the robot as seen through the birds-eye view of the mounted cameras above to be seen from an undistorted perspective directly above. This will allow the parallel lines that construct the path through the IGVC field to appear parallel as seen from a 2D plan. From the image above, we can see two different viewpoints. The video feed on the left is indeed the feed from the left stereo camera, while the right feed is from the right camera. The left feed is left untouched and is simply the raw viewpoint from the left camera. The right feed has however been corrected using a calibration file generated prior. This corrected image now renders the dot matrix as right-angled rectangular shape with the corresponding coulombs now appearing truly parallel. Using a correctional calibration file for each camera, we can alter both video feed to later be searched for lines in parallel.

|  |
| --- |
|  |

GitHub Repository Location: <https://github.com/rhrt/IGVC-2013-Imaging>