CMOS imaging sensor

For the ARLISS project we are using a combination of a CMOS imaging sensor and at least one ultrasonic sensor in order to detect and avoid obstacles. The path of the satellite, on a macro level, is determined by the GPS sensor, but modifications to the path will need to be set locally using this system.

The idea behind the imaging sensor is that, when there is an abrupt change in color in an image, especially when referring to nature, it often shows that there is an abrupt change in the depth being viewed. For our purposes we want to use this observation to detect objects in the satellite’s path, as well as, changes in terrain like a steep incline or a steep decline.

Manipulating raw camera frames can require a relatively large amount of space, as well as consume a large amount of energy. Energy is the most valuable resource for our project because we have a strict limit to the size and weight of the satellite so we have a very limited space for a battery to fit. There are several methods which we must use to minimize this cost, and there is one critical library in particular which we will use to help conserve this limited battery power.

As far as libraries go, we will be utilizing the cv.h library provided by OpenCV. This open-source library provides the capability of image and video I/O, image processing for individual video frames, as well as, built-in object recognition functionality. All foreseeable video manipulating should be able to be done with this single library.

The plan for reducing energy consumption is simple; first we will take a frame of raw footage as input from the CMOS device, then we will convert it to grayscale, and from grayscale we can convert it to binary data. Manipulating this binary data will consume far less energy than performing complex algorithms on the original, raw video footage.

Using the OpenCv library we will be able to capture video and directly load each video frame from the sensor to an IplImage object. OpenCv also has a built-in function called cvCvtColor to convert each image to grayscale. There will then need to be an algorithm which will create a digital map of the image using a two-dimensional array, and which will assign values of 1 or 0 depending on the difference in color change between pixels. When objects or abrupt changes in elevation are detected we can combine data from the ultrasonic sensor to determine the distance of the object, and modify the satellite’s route to avoid the obstacle.

This functionality of importing the live video feed, converting it to frames, and applying functions to the data to convert it to binary will need to exist in its own class. This class will be responsible for the flow of the input video data from the sensor, and outputting the data as an array of binary values to a separate class. The output from this class will combined with the output from the class responsible for the ultrasonic sensor data, and this combination of data will be used elsewhere to help determine what sequence of events needs to occur in order to appropriately navigate the terrain.