Water Classifier

Jordan Campbell

Shankara Narayanan Sethuraman

1. Pre-processing

The hardest problem with a classification problem is generating data that is separable. Using color spaces to separate the classes of water and non-water did not yield reasonable results. Figure 1 shows a histogram of the grayscale intensities for the water training images and non-water training data.

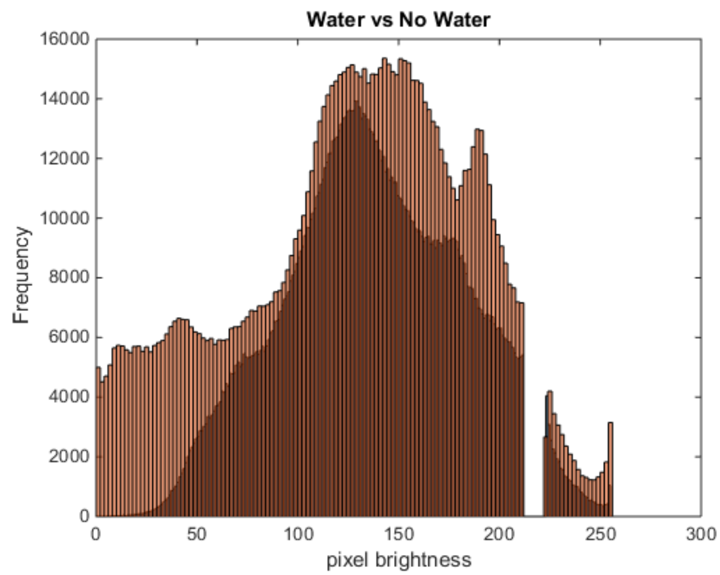

Figure 1. Histogram for water and non-water images

As you can see, there is way too much overlap in the data to distinguish between water and non-water. The gap in the data are the pixels that were in the background that I manually cropped with yellow in Microsoft Paint. Color spaces other than RGB (Red, Green, Blue) and gray scale such as HSV (Hue, Saturation, Value), CIE 1976 L*a*b, NTSC, CIE 1931 XYZ, and YCbCr yields better results but still have large overlap in their respective histograms. I chose to use range to characterize the classes. Range yields an image where each pixel is the difference between the maximum value and the minimum value in a 3-by-3 neighborhood in the corresponding input image. Thus, the range can affectively reveal edges or areas in an image where there is a large difference in brightness between neighboring pixels.

I first split the image into three channels, red, green, and blue so that I can take the range of a 2-dimensional image. I have chosen to use the blue channel although all 3 yield similar results. Figures 2 and 3 show the histograms the output of the range filter for the blue channel.

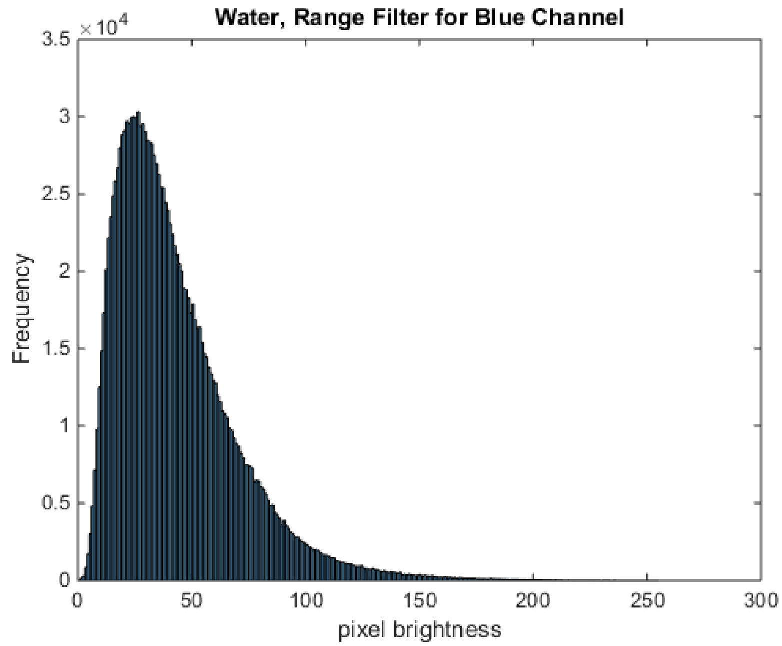

Figure 2. Histogram for water range image

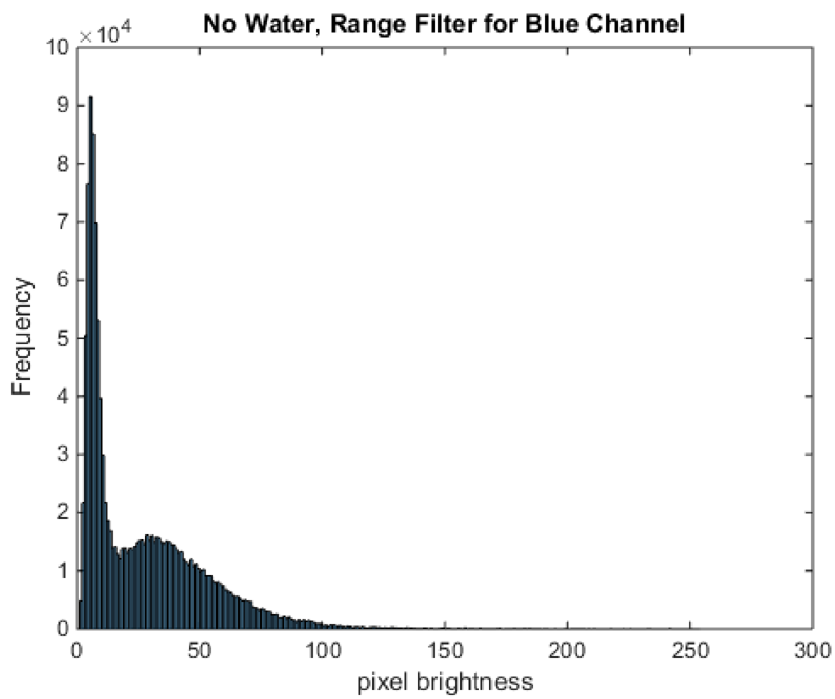

Figure 3. Histogram for non-water range image

The mean of the water data is 44.2458, and the mean of the non-water data is 26.72. Both histograms are then normalized and modeled with gamma distributions with different scaling and shape parameters for the classifier.

2. Flood Detection

The Bayesian classifier generates a histogram of data representing the range image for the blue channel of the input image. It then calculates the conditional probabilities, $p(x|w_1)$ and $p(x|w_2)$, where x is the input image and w_i is class i. $P(w_i)$ is 0.5 for each class. For training the classifier, I used 20 water images and 10 non-water images. For testing, there were 10 water images and 10 non-water images. The classifier correctly classified 17 of 20 images. It mislabeled 2 non-water images and 1 water image.

3. Segmentation

The technique I used to segment the image was to recursively cut image into parts, record the statistics of each part, and label the corresponding part as water or non-water based on its statistics. The general process is shown in figures 4 and 5.

Figure 4. Input Image

Figure 5. Segmented, 1st iteration

In the 1st iteration, the image is segmented into 4 parts. Each individual part is then classified as water or non-water. In the second iteration, each part is then again segmented into 4 parts, yielding a total of 16 parts. Again, each part is classified as water or non-water. More iterations yield better results, but of course also causes more computational memory time. Afterwards, I ran an algorithm called "connected components." I treat the image now as a binary image with the parts labeled water as 1 and parts labeled non-water as 0. Connected components is an efficient algorithm for distinguishing foreground and background. For this application, I treat the water as foreground and non-water as background. Figures 6 and 7 show an example of an input image and the segmented image.

Figure 6. Input Image

Figure 7. Segmented Image

To further distinguish a river bed from a large flooded area, I built a classifier that was not based on the water region's data, but rather on the background's statistics. I again used the range filter and designed a Bayesian classifier similar to the preprocessing section.