Keerthi Raj Nagaraja (nagaraj1@purdue.edu)

10-25-2014

1 Image Segmentation

In Image segmentation, given an image, we segment out the foreground from the background and extract the foreground contour. It involves two main steps. In the first step, we use otsu's algorithm to find an optimal pixel threshold value to separate foreground and background pixels and binarize the image. In the second step, we use a contour extraction algorithm to extract the border pixels of the foreground which provides the segmentation boundary.

We implement both RGB-based segmentation and texture-based segmentation.

1.1 Otsu's Algorithm

The otsu's algorithm finds an optimal threshold on the pixel intensity which separates foreground pixels from the background ones. Following are the steps involved in Otsu'algorithm:

(1) Split the given color image into B (Blue),G(Green) and R(Red) channels for RGB based segmentation method. For texture based method, find the variance of pixels within a 3x3, 5x5 and 7x7 window and consider each variance as a channel value.

(2) For each of the 3 images, do the following:

(i) Calculate the histogram of the image

(ii) Divide the histogram into two classes using a threshold k where, k is a candidate threshold value from 0 to L (255 for first iteration of otsu). For each value of k, find the between-class-variance ($\sigma_b{}^2$) as shown below: Probability of an intensity value i in the histogram is,

$$p_i = n_i/N$$

where, n_i is the number of pixels at intensity i and N is the total number of pixels in the image (or histogram). The weights of class 0 ($< k$) and 1($>= k$) are given by,

$$w_0 = \sum_{i=0}^{k} p_i$$

$$w_1 = \sum_{i=k+1}^{L} p_i$$

These are calculated with less computations using the following two iterative equation:

$$w_0^k = w_0^{k-1} + p_k$$

$$w_1^k = 1 - w_0^k$$

with $w_0^0 = p_0$
Mean of each class is given by,

$$\mu_0 = \frac{\sum_{i=0}^{k} ip_i}{w_0}$$

$$\mu_1 = \frac{\sum_{i=k+1}^{L} ip_i}{w_1}$$

These are calculated with less computations using the following iterative equations:

$$s^k = s^{k-1} + kp_k$$

with $s^0 = 0$ and $s^L = \sum_{i=0}^{L} ip_i$

$$\mu_0^k = \frac{s^k}{w_0^k}$$

$$\mu_1^k = \frac{s^L - s^k}{1 - w_0^k}$$

Now, the between-class-variance is given by,

$$\sigma_b{}^2 = w_0 w_1 (\mu_0 - \mu_1)^2$$

(iii) The optimal threshold is the one which gives the maximum $\sigma_b{}^2$. Let K be such optimal threshold.

(iv) Discard all the pixels (make them 0 intensity) that are below the threshold K (or above the threshold depending on where your foreground pixels are concentrated in the histogram)

(v) If needed, repeat from step (i) with the retained pixels and the new histogram range until satisfactory segmentation results are found manually.

(3) Once the indvidual channel images are thresholded, merge all three channel images by logical AND operation. This will produce the image where all the background is eliminated (Ideally).

(4) Convert this to binary image by replacing non-zero pixels with the maximum value of 255. The image produced will be the binary foreground image.

1.2 Contour Extraction

The contours or boundaries of the segmented region are extracted using the following algorithm:

(1) For each non-zero pixel in the binary foreground image, if any 4-connected pixels has zero-intensity, then the pixel is on the boundary and hence retain them.

(2) If the non-zero pixel is not a boundary, then make the pixel value zero.

This gives us the contours of boundaries of the foreground regions of the image.

2 Observations

- Otsu's algorithm has to be applied multiple times to get satisfactory results.
- I observed that while extracting the segmentation boundaries and applying the binary mask to the original color image, if boundary pixels are made 0, we can eliminate small regions of noisy areas. However, it comes at a cost of having more error in localizing the true boundary since the boundaries get shifted by 1 pixel in both x and y directions.
- Texture-based method provided better results compared to RGB-based method for "Lake" image since lake had a smooth texture compared to the background. However, for the "Tiger" image the texture-based method didn't do well since the texture of the tiger's skin was almost same as the texture of the neighboring ground and trees. The result was comparitively better with 5,7 and 9 window variances than 3,5 and 7 window variances.
- As one might expect, depending on the image, not all 3 channels are helpful in segmenting the foreground. If most of foreground is concentrated in one channel, then that particular channel will be the deciding one during logical AND operation. Lake image is a typical example where only blue channel eliminates most of the background from the foreground.

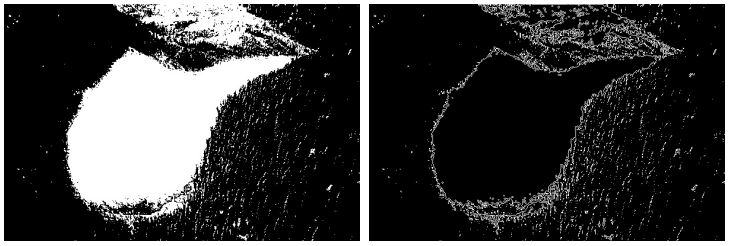
3 Results

3.1 Lake Image with RGB-based segmentation

Figure 1: Input Image

(a) Blue (b) Green (c) Red

Figure 2: Individual channels segmented



(a) Binary Image (b) Contour Image

Figure 3: Binary and contour images of foreground

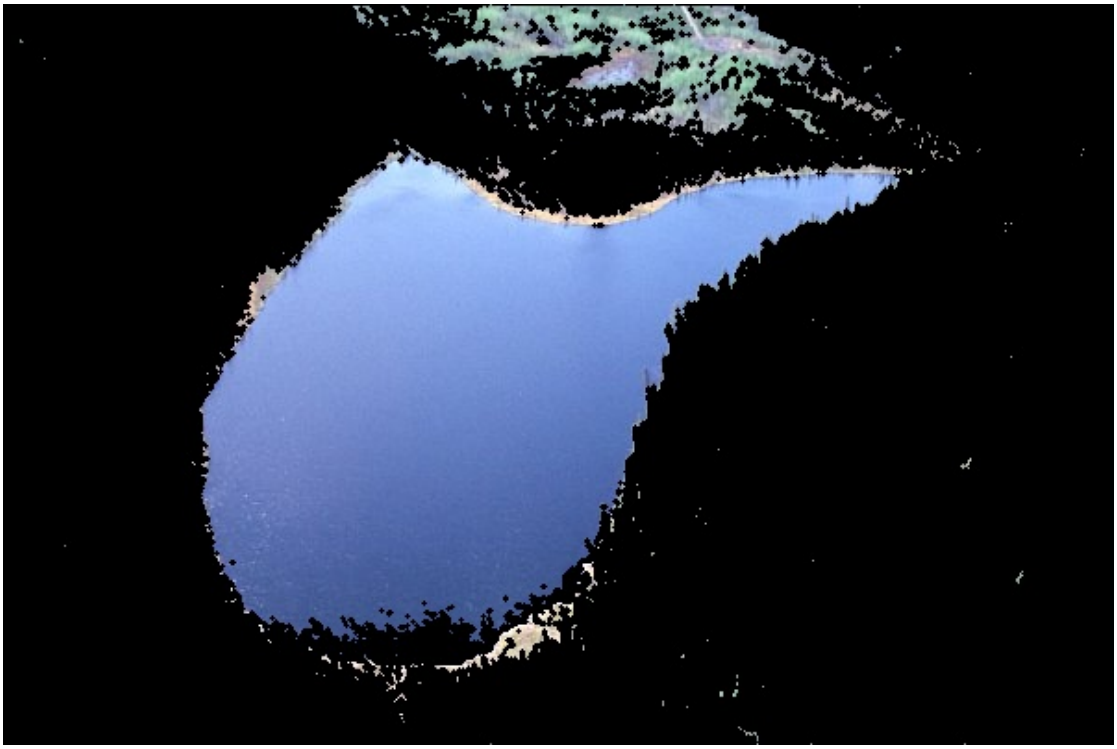


Figure 4: Segmented lake from original image after noise suppression

3.2 Lake Image with Texture-based segmentation

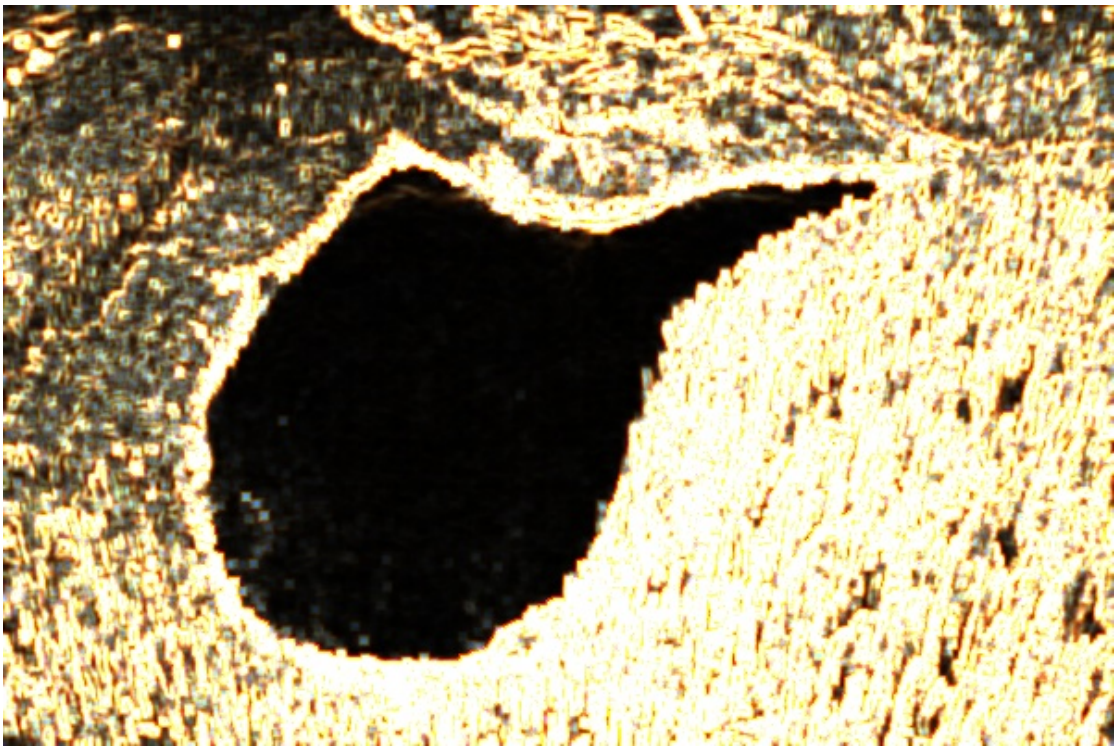
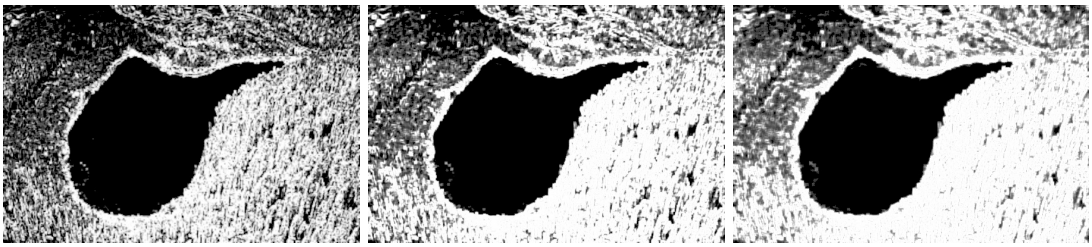
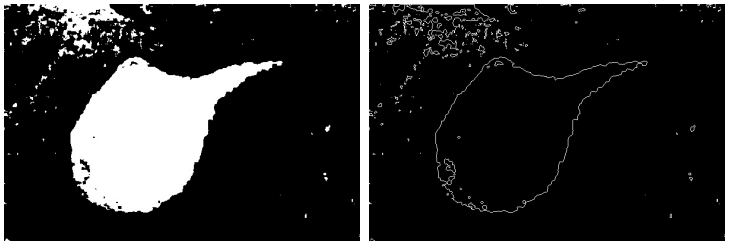


Figure 5: Input Texture Image



(a) 3x3 Variance (b) 5x5 Variance (c) 7x7 Variance

Figure 6: Individual channels segmented



(a) Binary Image (b) Contour Image

Figure 7: Binary and contour images of foreground

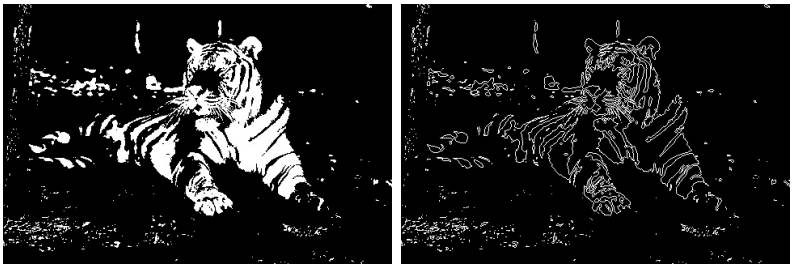
Figure 8: Segmented lake from original image after noise suppression

3.3 Tiger Image with RGB-based segmentation

Figure 9: Input Image

(a) Blue (b) Green (c) Red

Figure 10: Individual channels segmented



(a) Binary Image (b) Contour Image

Figure 11: Binary and contour images of foreground

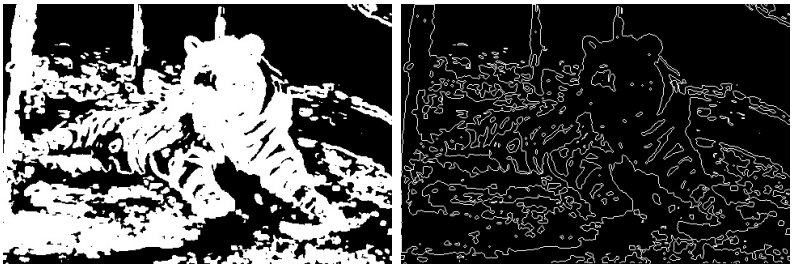
Figure 12: Segmented tiger from original image after noise suppression

3.4 Tiger Image with Texture-based segmentation

Figure 13: Input Texture Image

(a) 5x5 Variance (b) 7x7 Variance (c) 9x9 Variance

Figure 14: Individual channels segmented



(a) Binary Image (b) Contour Image

Figure 15: Binary and contour images of foreground

Figure 16: Segmented tiger from original image after noise suppression