Namrata Vivek Raghavan
nrgahav@purdue.edu

Image Segmentation

In this homework we are required to separate the foreground and the background from three different images and then to extract the contour of the segmented outputs. We have done this using two different set of methods. The first method involves applying the Otsu's algorithm to the three RGB color channels of the image. The second method involves applying the Otsu's algorithm after determining the texture-based features in the three images. Once the image segmentation is done using the two different methods the contour extraction is done to get the boundary of the foreground

1. Otsu's algorithm

The Otsu algorithm is used to calculate the optimal grey level to separate the foreground from the background. The following is the method to apply to Otsu's algorithm to calculate the optimal threshold to separate the foreground from the background.

1. Convert the image to its gray scale image.
2. Let L be the number of gray scale levels in an image. Let N be the total number of pixels in the gray scale image.

$$N = n_1 + n_2 + \cdots + n_L$$

The gray level histogram is normalized, and calculate the probability distribution at a given gray level i given by

$$p_i = \frac{n_i}{N}$$

3. Given a threshold k, Let C_0 be a class of all the pixels that belong to the background (gray level lesser than k) and C_1 be a class of all the pixels that belong to the foreground (gray level greater than or equal to k). Calculate the probability of class occurrences and the class mean levels given by

$$\omega_0 = \Sigma_{i=1}^{k} p_i$$

$$\omega_1 = 1 - \Sigma_{i=1}^{k} p_i$$

$$\mu_0 = \frac{\Sigma_{i=1}^{k} i p_i}{\Sigma_{i=1}^{k} p_i}$$

$$\mu_1 = \frac{\Sigma_{i=1}^{L} i p_i - \Sigma_{i=1}^{k} i p_i}{1 - \Sigma_{i=1}^{k} p_i}$$

4. Calculate the between class variances given by

$$\sigma_b^2 = \omega_0 \omega_1 (\mu_1 - \mu_0)^2$$

Namrata Vivek Raghavan
nrgahav@purdue.edu

5. Repeat this procedure for all possible values for threshold ($k = 1, 2, \ldots . L$)
6. Choose the value the value for k that maximizes the between class variances

The segmentation using the Otsu's algorithm can be improved by repeating the procedure above for a number of iterations. In this experiment I have carried out the iteration 3 times for optimization.

2. RGB Image Segmentation

In the following method we first separate into the original image into its three separate R G B channels and then apply the Otsu's algorithm to the three separate channels the following are the steps to conduct RGB image segmentation:

1. Separate the original image to its three separate RGB channels and convert the channels to its gray scale level values.
2. Apply the Otsu's algorithm to the three channels separately and obtain the masks for each channel.
3. Combine the three masks using logical AND and OR operators. The combination of AND and OR operators used depends on the original image.
    a. In case of the image of the light house, since red was the most prominent color of the light house we used the following combination
    $$mask_{overall} = mask_{red} \& \sim mask_{blue} \& \sim mask_{green}$$
    b. In case of the image of the baby, since both the foreground and the background had an overall red the following combination of masks were used.
    $$mask_{overall} = \sim mask_{blue} \& \sim mask_{green}$$
    c. In case of the image of the ski man , since there are no dominant colors all three masks are used.
    $$mask_{overall} = mask_{red} \& \sim mask_{blue} || \sim mask_{green}$$

3. Texture based Image Segmentation

In the following method we first separate into the original image into three separate texture feature channels and then apply the Otsu's algorithm to the three separate channels the following are the steps to conduct texture-based image segmentation:

1. Convert the original image to its gray scale image
2. Generate a new gray scale image where the pixels are the variance of the pixels in a N x N neighborhood in the original gray scale image. Repeat this process for three different values of N (N = 3, 5, 7).
3. Apply the Otsu's algorithm to the three channels separately and obtain the masks for each channel.
4. Combine the three masks using logical AND operators. The combination of AND operators used depends on the original image.

Namrata Vivek Raghavan
nrgahav@purdue.edu

a. In case of the image of the light house, since the foreground was relatively smooth the following combination was used.

$$mask_{overall} = mask_5 \& mask_7$$

b. In case of the image of the baby, since there is a lot of different textures all three masks are used.

$$mask_{overall} = mask_3 \& mask_5 \& mask_7$$

c. In case of the image of the ski man , since there is a lot of different textures all three masks are used.

$$mask_{overall} = mask_3 \& mask_5 \& mask_7$$

4. Contour Extraction

Once the foreground has been separated from the background and the image segmentation is done, the contour is extracted to get the exact boundary of the foreground. The following is the procedure for contour extraction from a segmented image:

1. Loop through every pixel in the segmented image.
2. If the pixel is 0 (background), then it is not a contour point.
3. If the pixel is 1 (foreground), then check if any of its four nearest neighbors are 0. If any of the neighbors are 0 then it is a contour point. If none of the four nearest neighbors are 0 then it is not a contour point.

5. Observation

From the results it is seen that for both the RGB image segmentation and the texture-based image segmentation, the segmentation method depends on the original image. Overall the RGB image segmentation method is seen to work the best for the light house image while the texture-based segmentation is seen to work the best for the image of the baby and the man skiing. This is because in case of the picture of the light house, the foreground has one dominant colour red. Thus, image segmentation of the lighthouse with the RGB segmentation method is easy. In case of the baby both the foreground and the background have an overall red hue thus image segmentation using the RBG segmentation method does not work well. Similarly, in case of the man skiing the foreground does not have one dominant color, thus image segmentation with the RGB method does not work well. In case of the texture-based image segmentation method, it works best for the image of the baby and the man skiing. This is because the texture-based image segmentation works the best when the foreground had more textures than the background. In case of the image of the baby and the man skiing the foreground has more textures than the background and thus the texture-based method works well for these images. The image of the lighthouse has a very smooth foreground compared to that of the background, thus the texture-based method does not work well for the lighthouse image.

Namrata Vivek Raghavan
nrgahav@purdue.edu

6. Results

1. Input Image 1 : Light house

Figure 1: Input image of the lighthouse

a. RGB segmentation

Figure 2: Red channel mask created using the Otsu's algorithm

Namrata Vivek Raghavan
nrgahav@purdue.edu

Figure 3: Blue channel mask created using the Otsu's algorithm

Figure 4: Green channel mask created using the Otsu's algorithm

Namrata Vivek Raghavan
nrgahav@purdue.edu

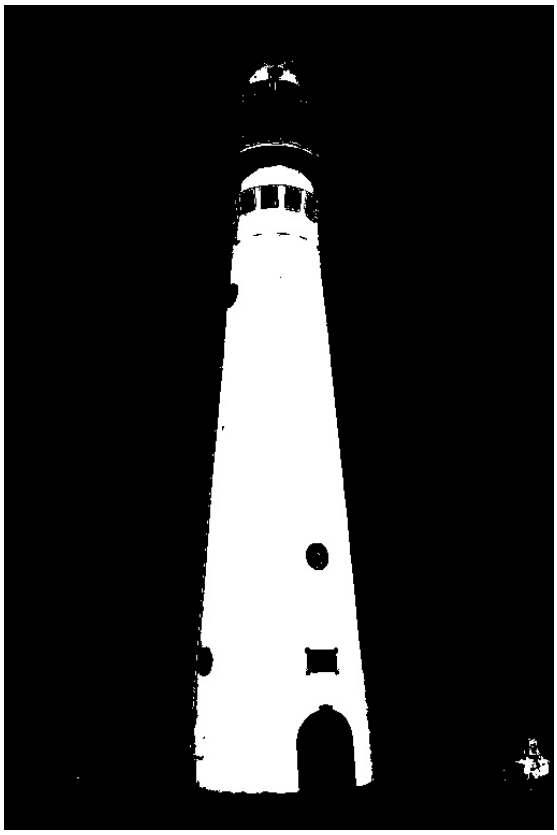

Figure 5: Image segmentation using the RGB segmentation method

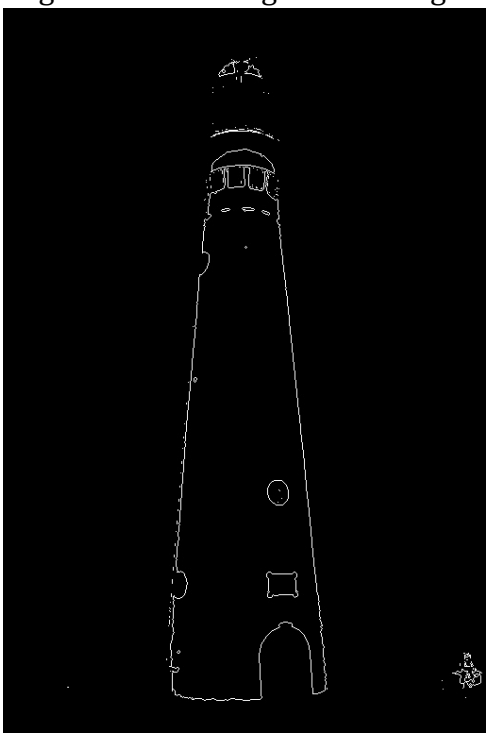

Figure 6: Contour Extraction using the RGB segmentation method

Namrata Vivek Raghavan
nrgahav@purdue.edu

b. Texture-based image segmentation

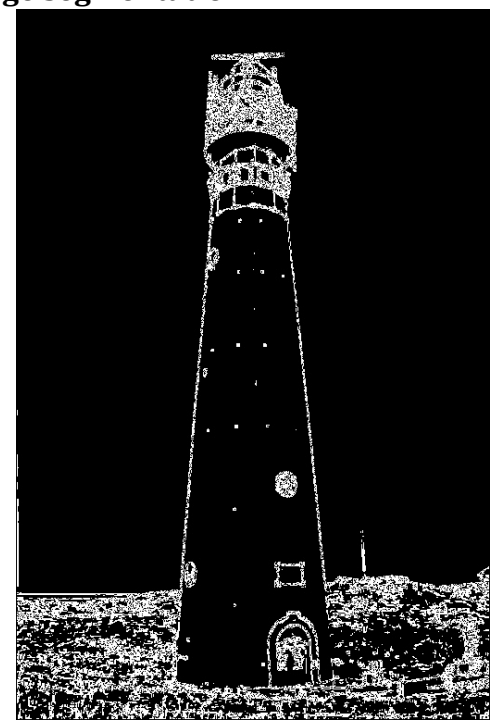

Figure 7: N = 3 channel mask created using the Otsu's algorithm

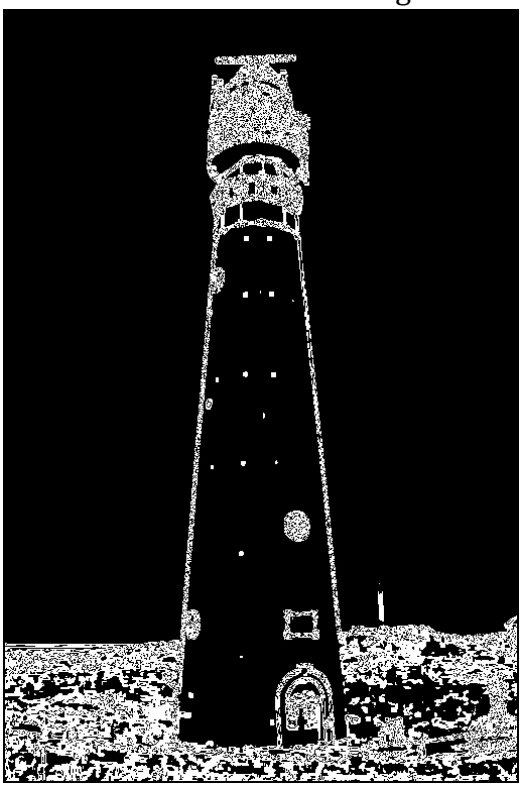

Figure 8: N = 5 channel mask created using the Otsu's algorithm

Namrata Vivek Raghavan
nrgahav@purdue.edu

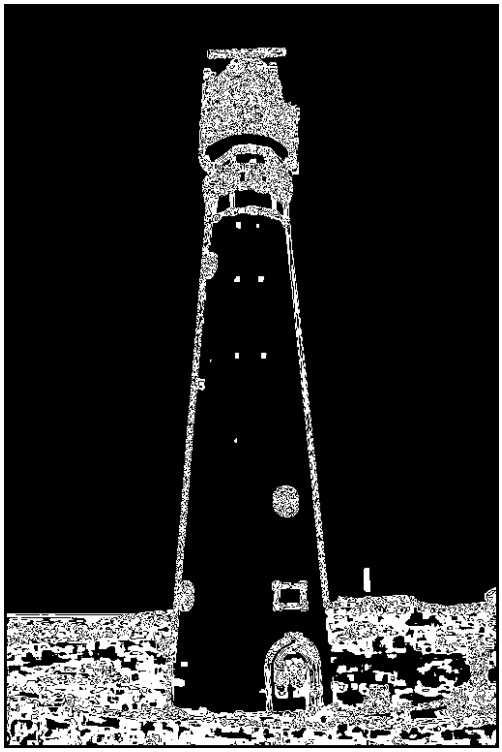

Figure 9: N = 7 channel mask created using the Otsu's algorithm

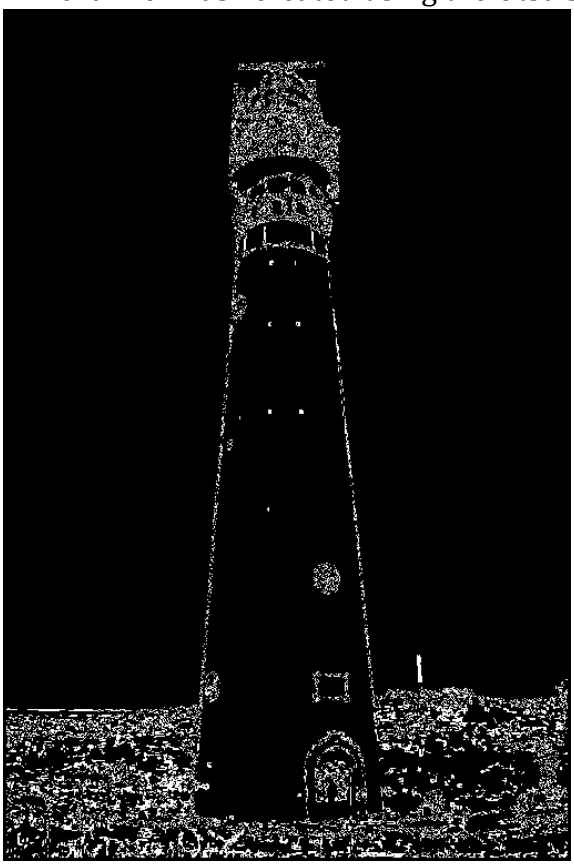

Figure 10: Image segmentation using the texture-based method

Namrata Vivek Raghavan
nrgahav@purdue.edu

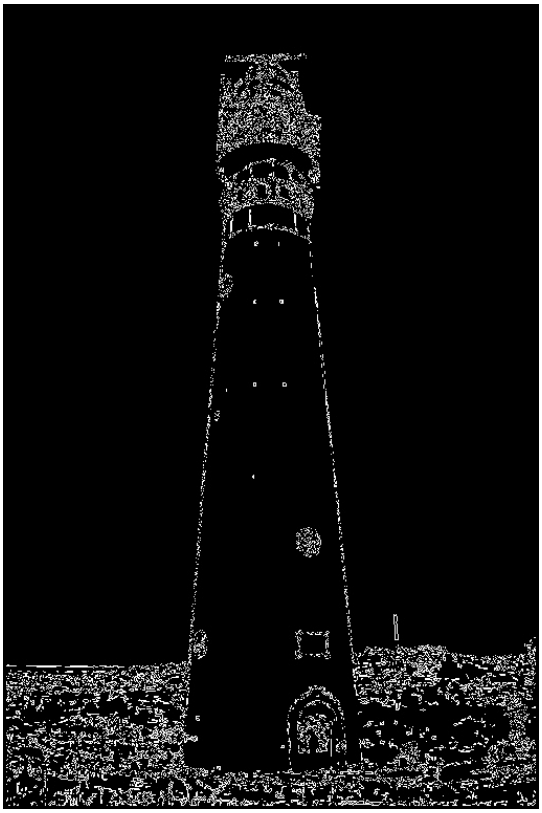

Figure 11: Contour Extraction using the texture-based method

2. Input Image 2 : Baby

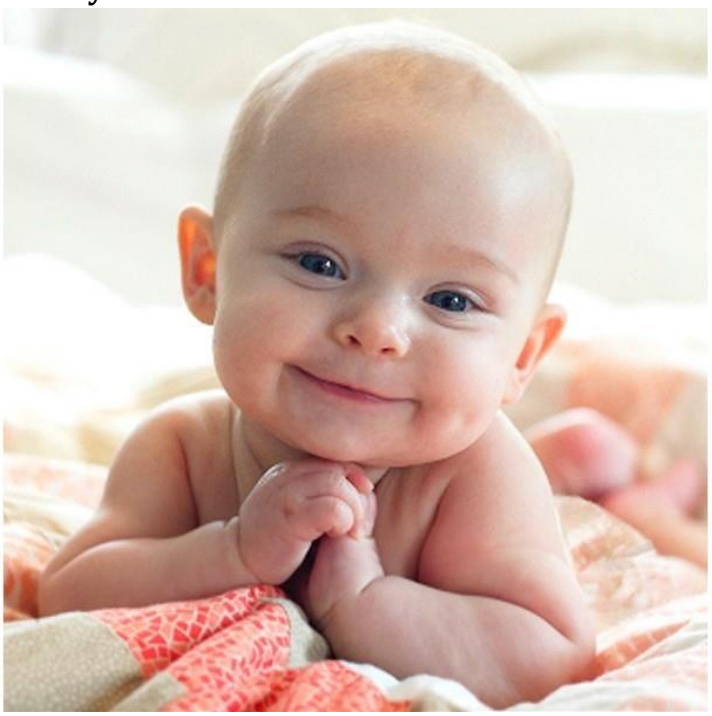

Figure 12: Input image of the baby

Namrata Vivek Raghavan
nrgahav@purdue.edu

a. RGB segmentation

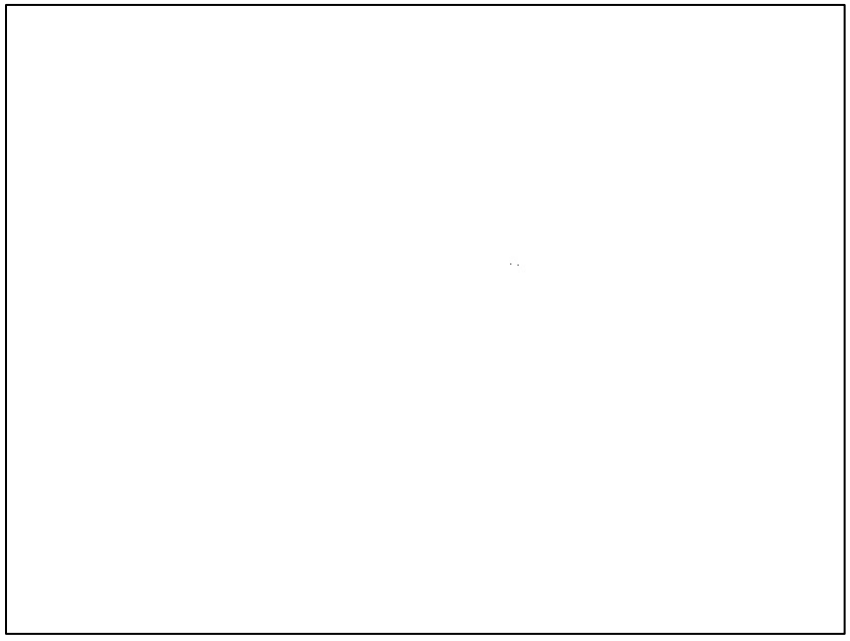

Figure 13: Red channel mask created using the Otsu's algorithm

Figure 13: Blue channel mask created using the Otsu's algorithm

Namrata Vivek Raghavan
nrgahav@purdue.edu

Figure 14: Green channel mask created using the Otsu's algorithm

Figure 15: Image segmentation using the RGB segmentation method

Namrata Vivek Raghavan
nrgahav@purdue.edu

Figure 16: Contour Extraction using the RGB segmentation method

b. **Texture-based image segmentation**

Figure 17: N = 3 channel mask created using the Otsu's algorithm

Namrata Vivek Raghavan
nrgahav@purdue.edu

Figure 18: N = 5 channel mask created using the Otsu's algorithm

Figure 19: N = 7 channel mask created using the Otsu's algorithm

Namrata Vivek Raghavan
nrgahav@purdue.edu

Figure 20: Image segmentation using the texture-based method

Figure 21: Contour Extraction using the texture-based method

Namrata Vivek Raghavan
nrgahav@purdue.edu

3. Input Image 2 : Man skiing

Figure 22: Input image of the man skiing

a. RGB segmentation

Figure 23: Red channel mask created using the Otsu's algorithm

Namrata Vivek Raghavan
nrgahav@purdue.edu

Figure 24: Blue channel mask created using the Otsu's algorithm

Figure 25: Green channel mask created using the Otsu's algorithm

Namrata Vivek Raghavan
nrgahav@purdue.edu

Figure 25: Image segmentation using the RGB segmentation method

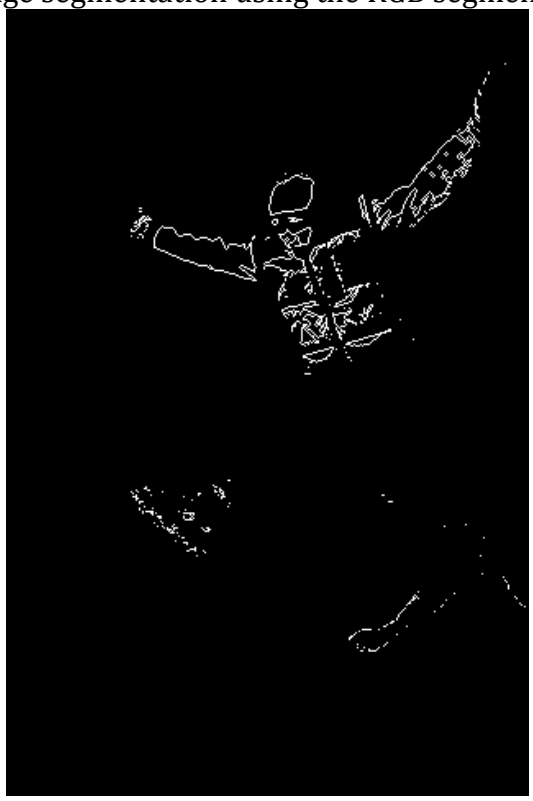

Figure 26: Contour Extraction using the RGB segmentation method

Namrata Vivek Raghavan
nrgahav@purdue.edu

b. Texture-based image segmentation

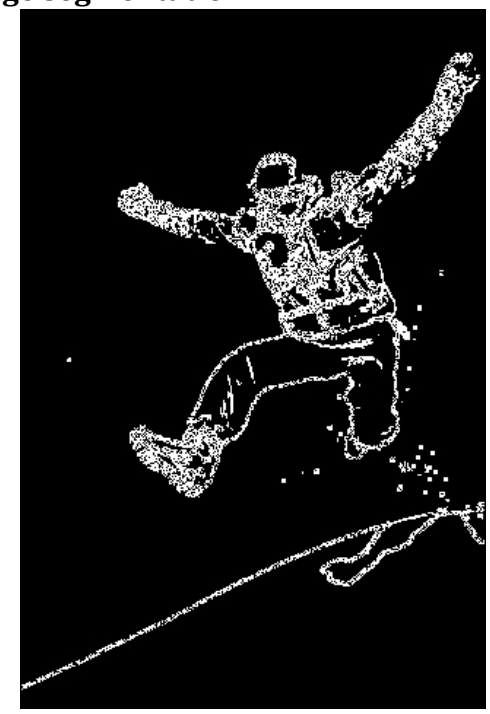

Figure 27: N = 3 channel mask created using the Otsu's algorithm

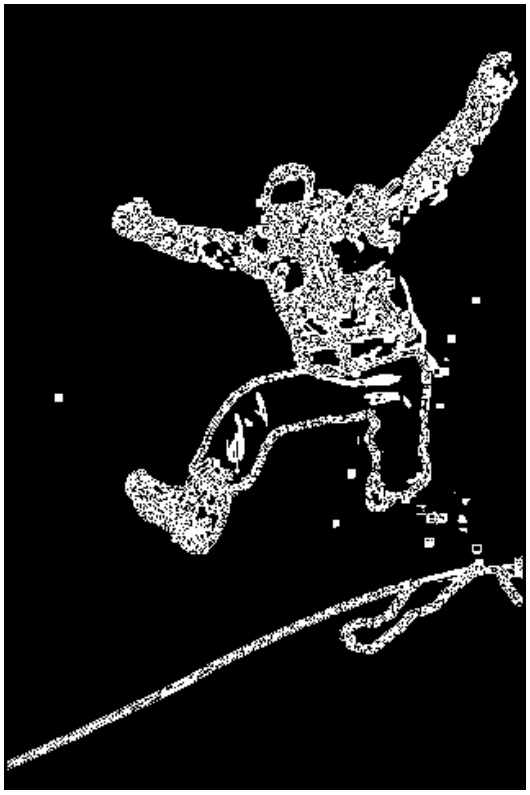

Figure 28: N = 5 channel mask created using the Otsu's algorithm

Namrata Vivek Raghavan
nrgahav@purdue.edu

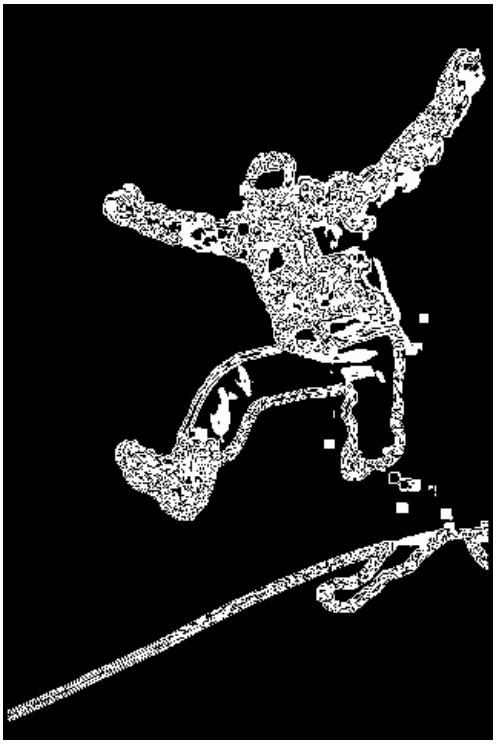

Figure 29: N = 7 channel mask created using the Otsu's algorithm

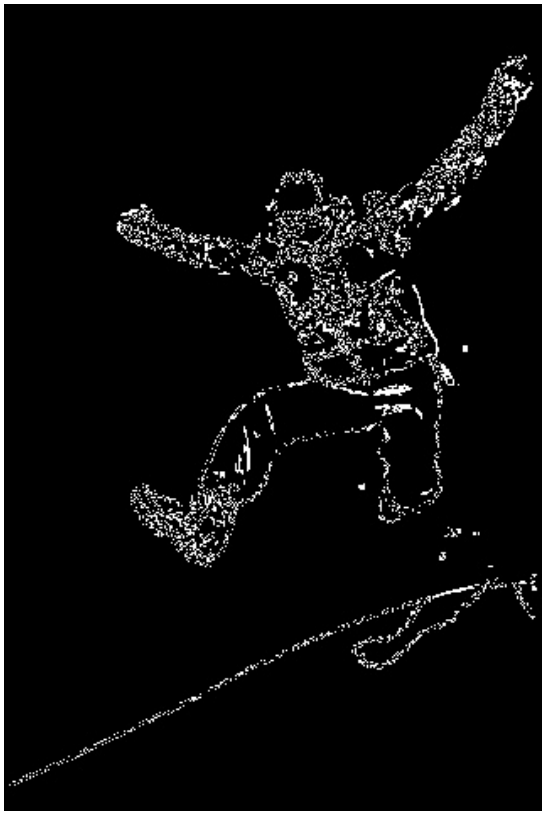

Figure 30: Image segmentation using the texture-based method

Namrata Vivek Raghavan
nrgahav@purdue.edu

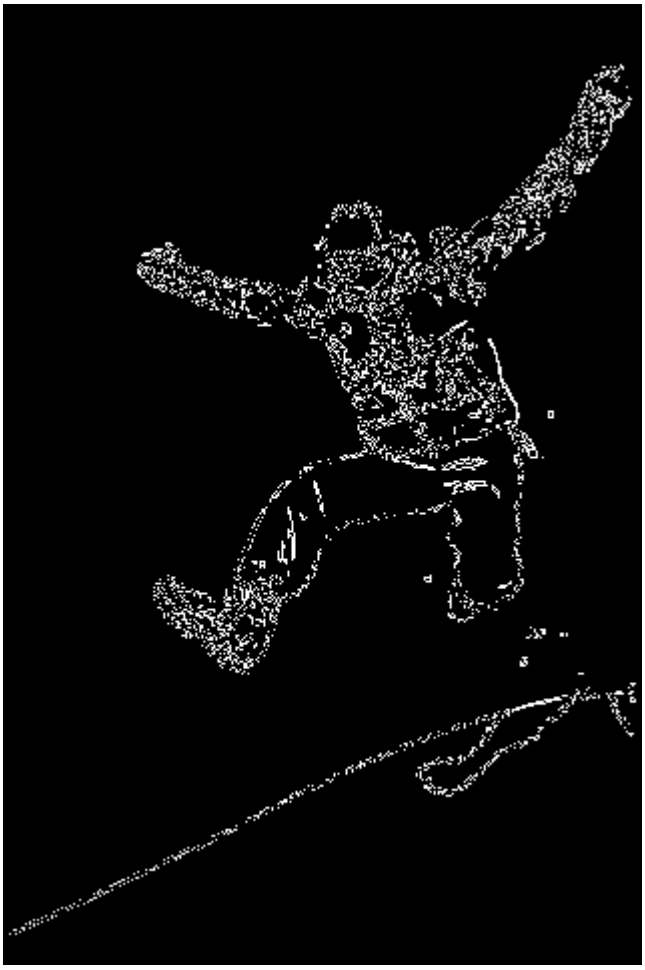

Figure 21: Contour Extraction using the texture-based method