Activity 5:
IMAGE SEGMENTATION

Genesis Vertudez – 202003099

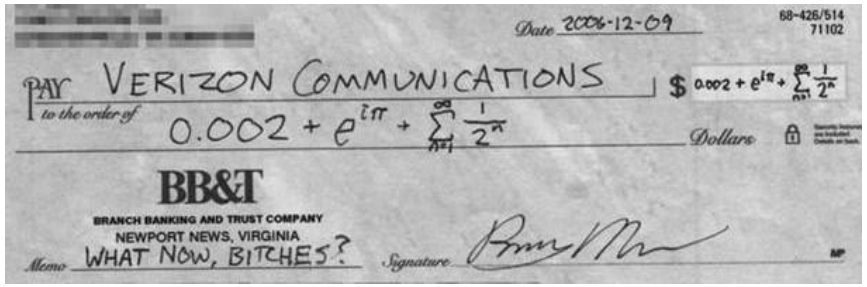
App Physics 157 - Computational Analysis and Modeling in Physics

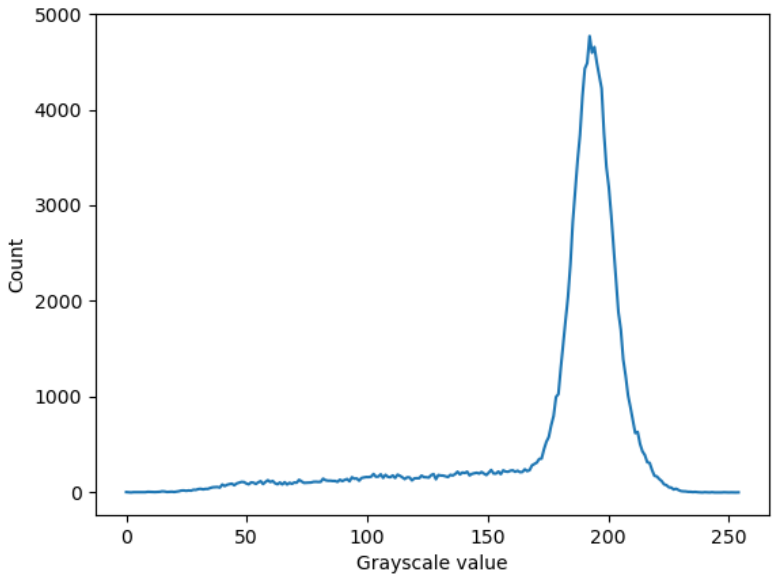
Submitted to Dr. Maricor Soriano; Mx. Rene Principe Jr.

OBJECTIVES

- Apply threshold to grayscale histogram to separate selections from the background
- Normalize an image
- Obtain the normalized chromaticity coordinates of an image
- Implement parametric segmentation to select chosen colors in an image
- Implement nonparametric segmentation to select chosen colors in an image

GRAYSCALE HISTOGRAM THRESHOLDING





One technique to select a region of interest (ROI) in an image is by **thresholding its grayscale histogram**. The image on the left shows an already grayscale image of a check. Selecting the texts from the background should be easy because the color distinction is clear; text is black, background is gray.

Looking at the histogram of the grayscale image, we see that there is a peak. This corresponds to the background pixels since it is much more than the text pixels. Hence, if we want to select the text, we only turn on values below this peak. If however, we want the background, then we should turn on only those values in the peak.

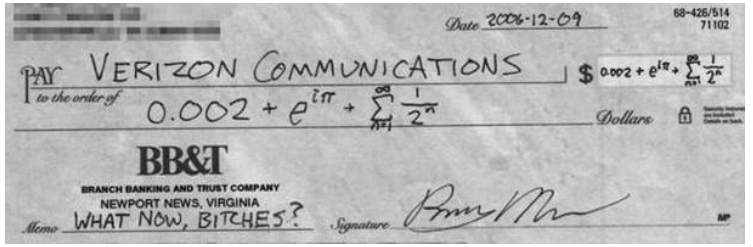
Codes and files: https://github.com/genvert/AP_157_FX-2_Vertudez/tree/main/Activity%205

GRAYSCALE HISTOGRAM THRESHOLDING

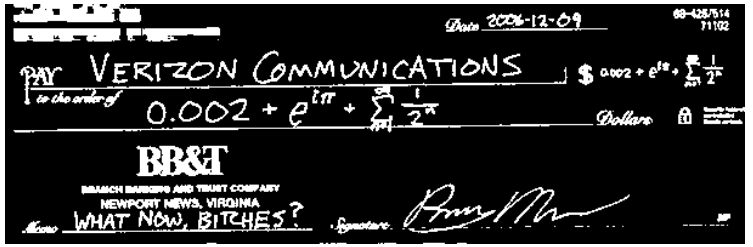
Here we see the result of the thresholding. The first image is the original image for comparison.

In the second image, we selected only the grayscale values the is below the peak, particularly < 150. Again, 1's corresponds to white and 0's corresponds to black. Hence, only the texts are "turned on" in this binary image.

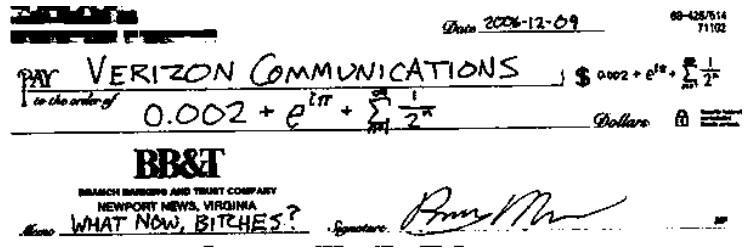
On the other hand, the third image shows the resulting binary image when the selected grayscale values are those that are > 150. In this case, the background is the one that is selected or "turned on".



Original image



Text selected

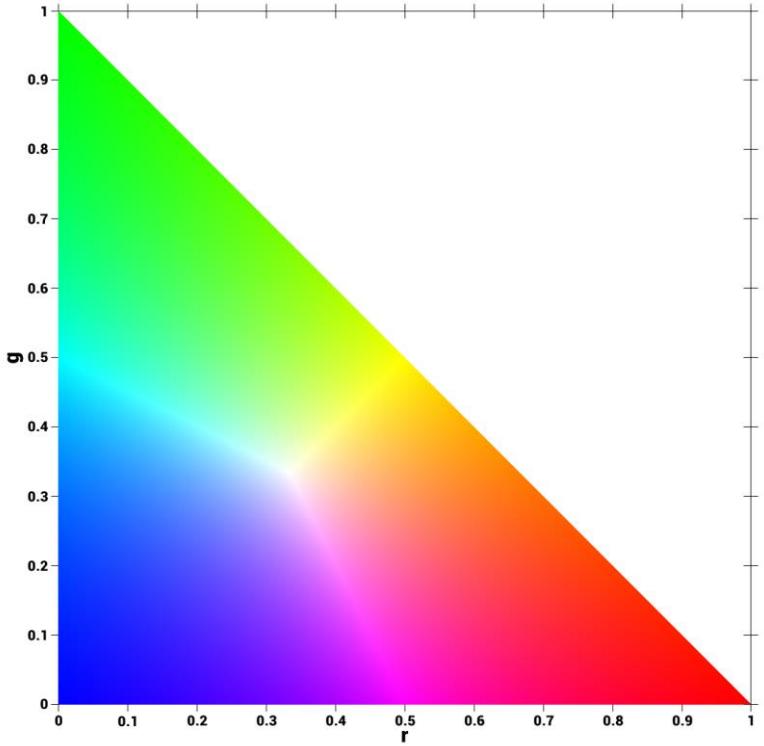


Background selected

Codes and files: https://github.com/genvert/AP_157_FX-2_Vertudez/tree/main/Activity%205

NORMALIZED CHROMATICITY COORDINATES

In this next section, we will look at how to select ROIs from images if grayscale histogram thresholding is not applicable (maybe in the grayscale world the colors look similar). Before, diving into the techniques, we first discuss the **Normalized Chromaticity Coordinates (NCC)**.



NCC

The NCC is a way to determine if two colors are in the same family. Say you have an image with different lightings on the same color. Applying grayscale thresholding will be hard because they will appear different colors (one is darker, other is lighter) even though they are, say, both red.

In NCC, the intensity, which determines the brightness of the object, is divided into the RGB channels. This way, the new values rgb is normalized such that

$$r+g+b=1 \qquad (1)$$

for every pixel. By doing this, we remove the dependency of the color on intensity so that the computer sees the same colors as the same color, regardless of the intensity.

Moreover, because of equation 1, it is sufficient to know r and g value only, since b = 1-r-g. The image on the left shows NCC, where the x axis is the r value, and the y axis is the g value. This is actually a 2D histogram that shows which colors are present on an image.

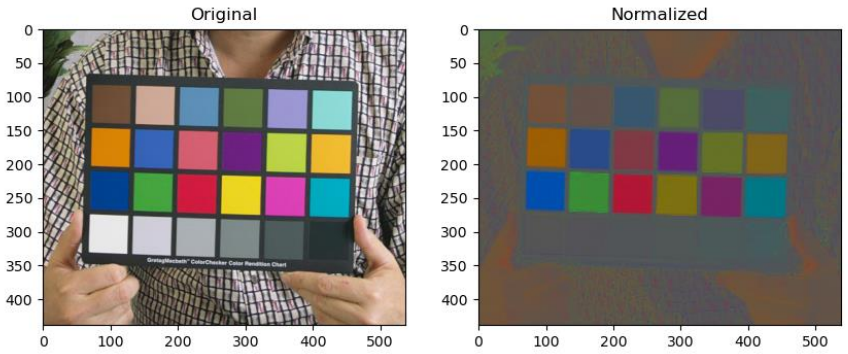
Codes and files: https://github.com/genvert/AP_157_FX-2_Vertudez/tree/main/Activity%205

NORMALIZED CHROMATICITY COORDINATES

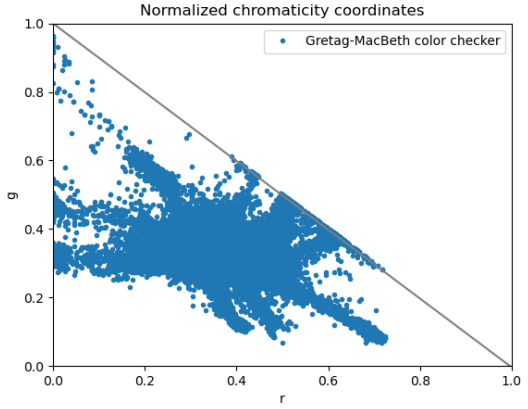
Here is an example of a normalized image. The Gretag-Macbeth color checker is used commonly in televisions in order to check colors as it appears in the TV. The first image shows the comparison of the original image and the normalized image.

We see that in a normalized image, grayscale values appear the same; they are tagged as the same color. This is because if we want to select ROIs using grayscale, then we might as well just use histogram thresholding. Hence, tagging them as the same color in a normalized image is useful as every grayscale part is automatically excluded when we want to select a certain non-gray color.

We see a scatter plot in the NCC of the image. Again, this is actually a 2D histogram of the colors present in the image. The more red colors there are, the more points in the lower right. The more green in the image, the more points in the upper left. Lastly, the more blue in an image, the more points in the origin where r and g are 0.



Gretag-Macbeth color checker



Gretag-Macbeth NCC

Codes and files: https://github.com/genvert/AP_157_FX-2_Vertudez/tree/main/Activity%205

PARAMETRIC COLOR SEGMENTATION

We now proceed to the first technique to select a colored ROI in an image which is called **parametric segmentation**. In parametric segmentation, we take an ROI, then we calculate the probability that a pixel belongs to that color, then threshold the probability which we decide "belongs" to the color.

The probability of r or g to belong in the r and g of the ROI is just given by a Gaussian distribution

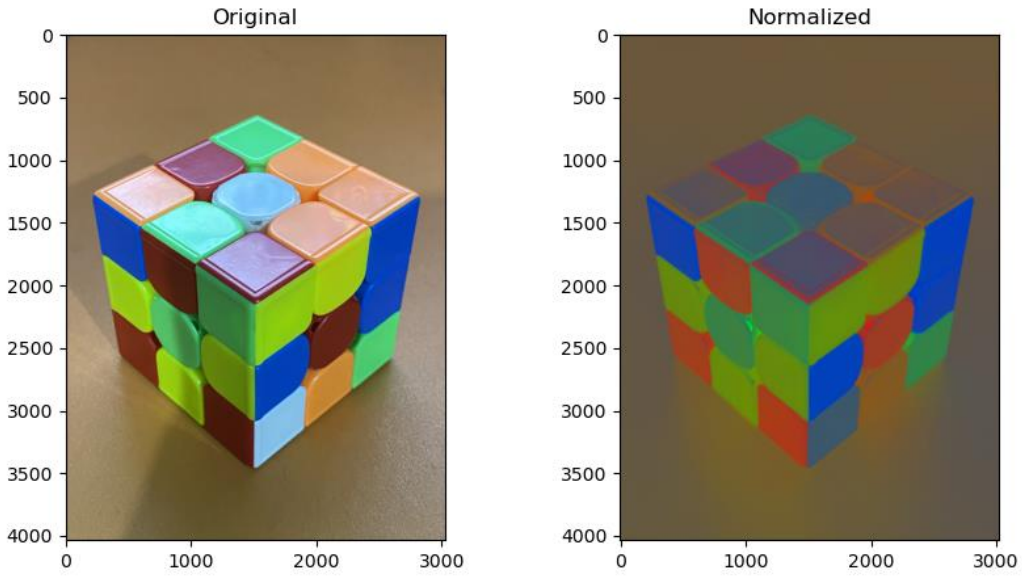
$$p(r) = \frac{1}{\sqrt{2\pi\sigma_r}} e^{\frac{-(r-\mu_r)^2}{2\sigma_r^2}} \qquad p(g) = \frac{1}{\sqrt{2\pi\sigma_g}} e^{\frac{-(g-\mu_g)^2}{2\sigma_g^2}}$$

where $\sigma_r, \mu_r, \sigma_g,$ and μ_g are the mean and standard deviations of the ROI's r and g values, respectively. The probability that a pixel then belongs to the ROI is just the product of the two, $p(r)p(g)$.

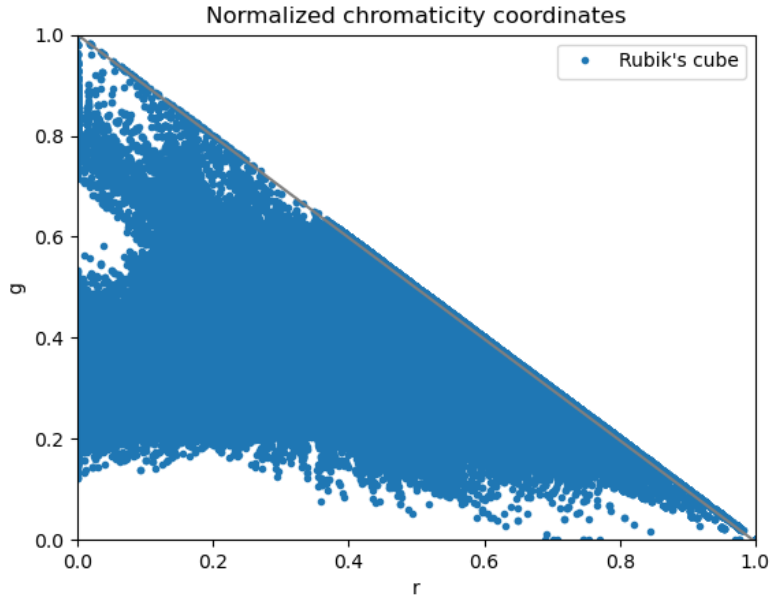
Codes and files: https://github.com/genvert/AP_157_FX-2_Vertudez/tree/main/Activity%205

PARAMETRIC COLOR SEGMENTATION

For our first example, take a look at the Rubik's cube below and its normalized version. The image on the right shows its NCC. Since the Rubik's cube contains six common colors, the NCC is essentially filled out. What we will do is segment each color of the cube, namely blue, white, green, red, yellow, orange.



Cube

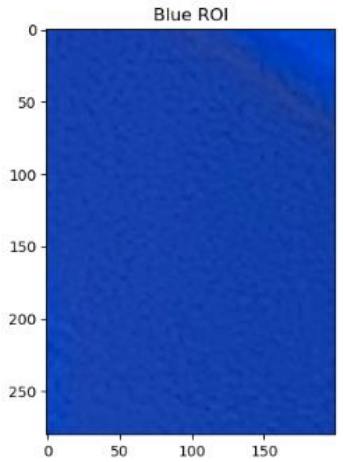


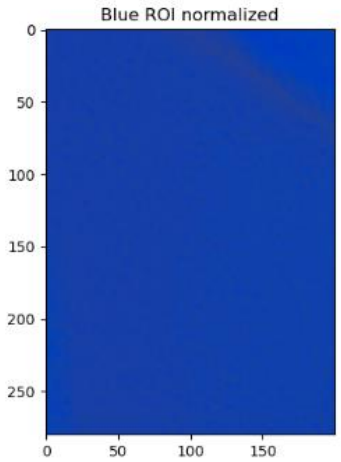
Cube NCC

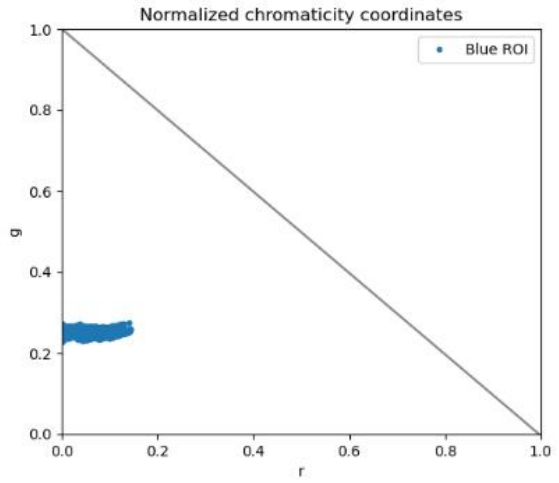
Codes and files: https://github.com/genvert/AP_157_FX-2_Vertudez/tree/main/Activity%205

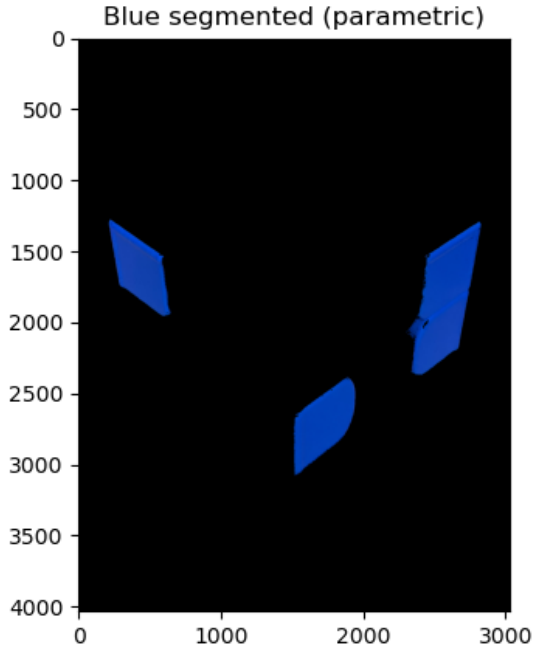
PARAMETRIC COLOR SEGMENTATION

The first color we look at is **blue**. We take a sample ROI on the upper left piece of the cube. I set the threshold to a very small value in order to admit all pixels with a nonzero probability of belonging in the blue color. The first image shows the Blue ROI, the second its normalized version, and the third its NCC. As expected, the pixels present are near the origin since that corresponds to blue. The last image on the right shows the segmented blue colors
in the cube. Amazing!







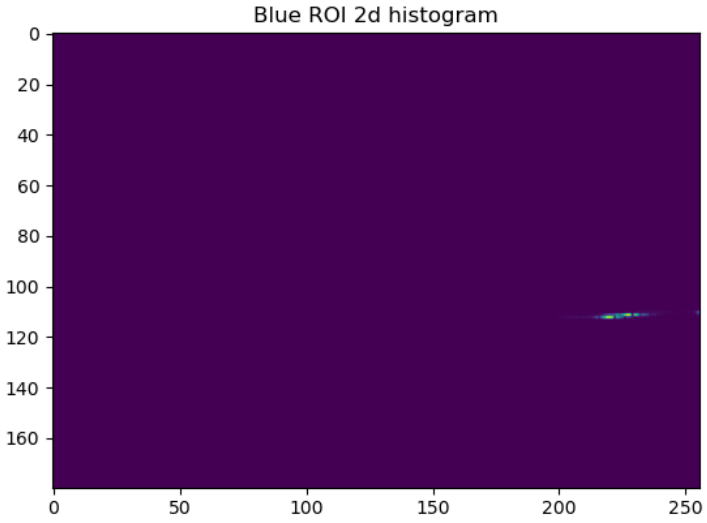


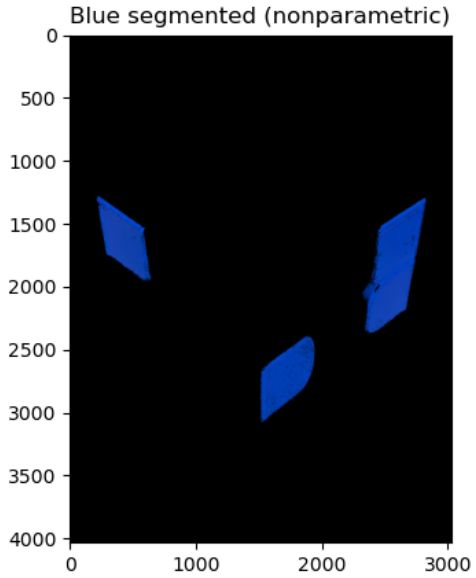
Codes and files: https://github.com/genvert/AP_157_FX-2_Vertudez/tree/main/Activity%205

NONPARAMETRIC COLOR SEGMENTATION

The other technique to segment the colored image is the **nonparametric segmentation**. The idea in this technique is by using backprojection, just like we did in the very first activity. The only difference now is that our histogram is 2D, and we work on HSV channels instead of RGB. We backproject the HSV of the image to the 2D histogram of the HSV of the ROI.

The image on the left shows the 2D histogram of the blue ROI. Backprojecting the original Rubik's cube image to this, we get the result shown on the right image.

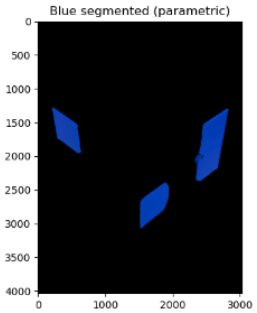


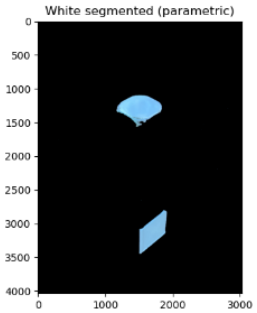


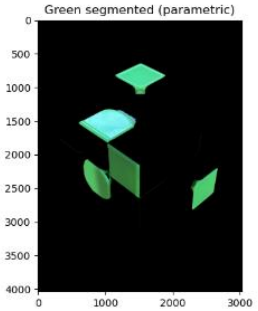
Codes and files: https://github.com/genvert/AP_157_FX-2_Vertudez/tree/main/Activity%205

PARAMETRIC VS NONPARAMETRIC

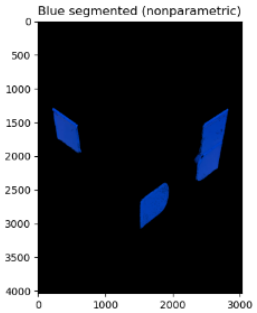
Parametric

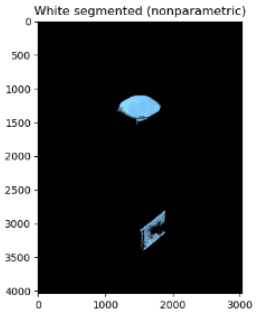


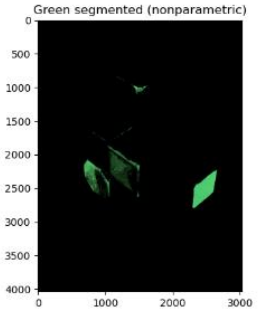




Nonparametric





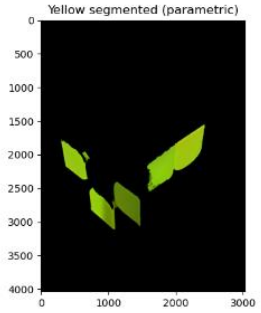


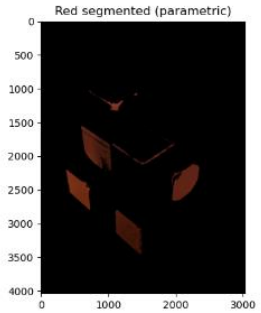
We see some particular problem on the red and orange colors.

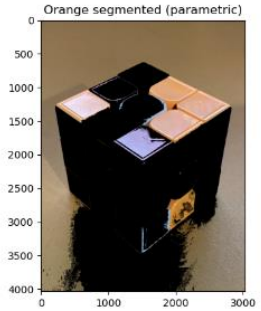
In parametric segmented orange, the table on the background was selected as well because it is also orange It also picked up the red corner which reflected light.

However, nonparametric segmentation did not pick this up which is good, but at the cost of not picking the lower orange color.

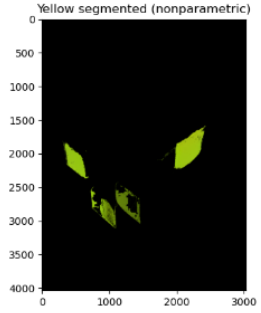
Parametric

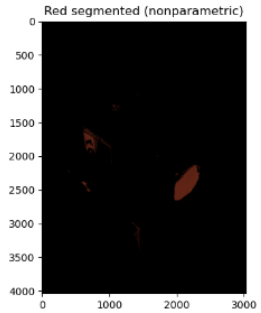


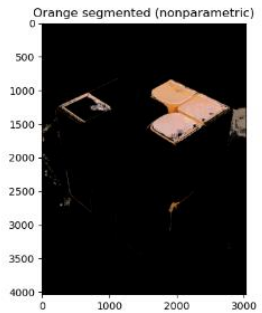




Nonparametric





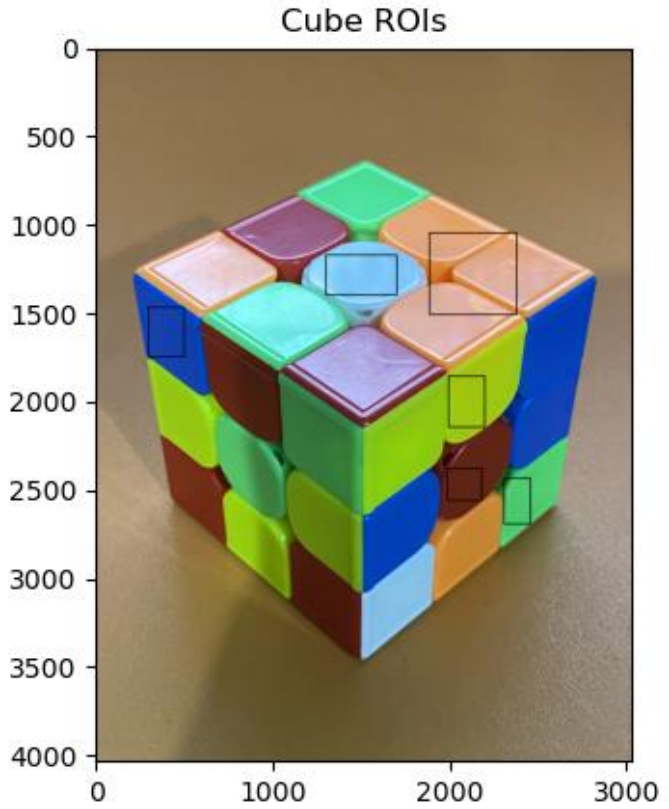


Codes and files: https://github.com/genvert/AP_157_FX-2_Vertudez/tree/main/Activity%205

PARAMETRIC VS NONPARAMETRIC

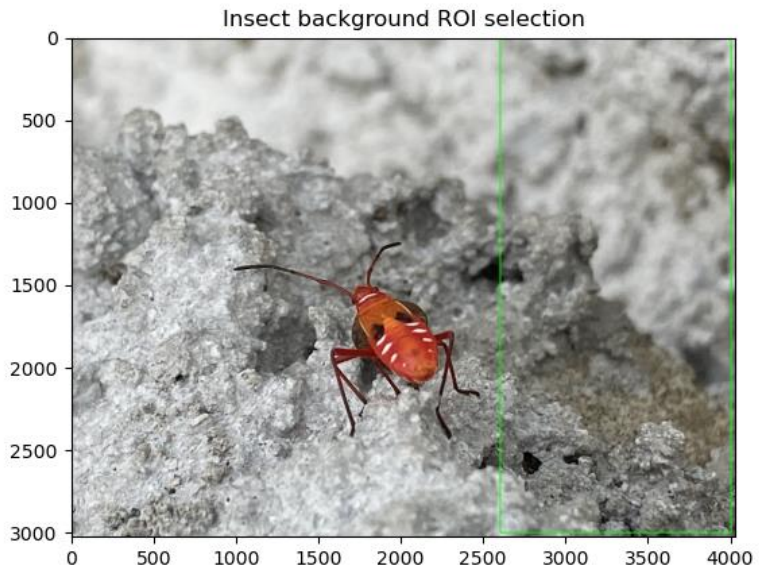
Comparing parametric and nonparametric color segmentation, it is clear the parametric segmentation does a better job. This is because the parametric segmentation relies on a continuous Gaussian probability, while the nonparametric segmentation relies on a discrete histogram backprojection.

For reference, the images on the right shows the ROI selected for each color on the cube



Codes and files: https://github.com/genvert/AP_157_FX-2_Vertudez/tree/main/Activity%205

PARAMETRIC VS NONPARAMETRIC



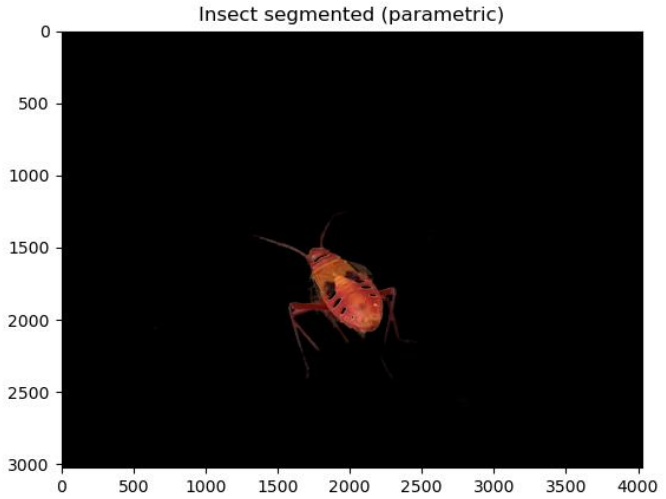
Here is another example where instead of selecting objects with the same color as the ROI, we select those not in the ROI.

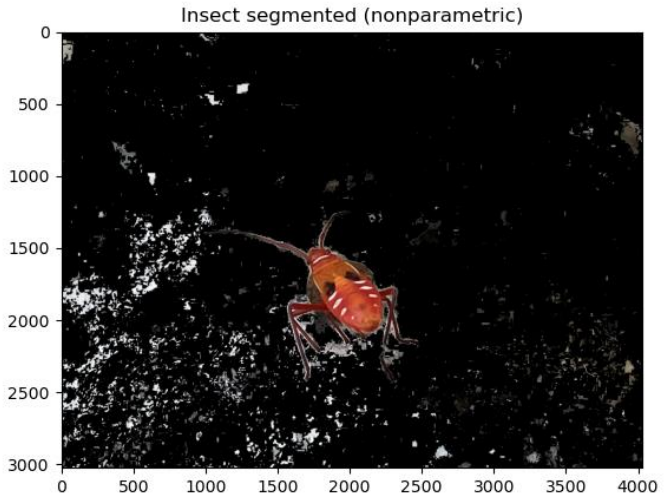
The image on the left shows a cotton strainer bug which I took a photo of. My goal is to select the insect, but not selecting the insect as my ROI. I noticed that the background is more uniform, so I instead used the background as my ROI, and reversed my threshold to show the insect instead of the background.

Codes and files: https://github.com/genvert/AP_157_FX-2_Vertudez/tree/main/Activity%205

PARAMETRIC VS NONPARAMETRIC

Again, the parametric segmentation did a better job at isolating the insect. However, remember that parametric segmentation uses NCC where grayscale values appear the same. Because of this, the white spots on the insect are accounted to belong in the background since they are grayscale colors. This is the where the nonparametric wins in terms of the remaining parts of the insect; however, it did not only leave out the white spots on the insect, but also white spots in the background! So parametric supremacy still.

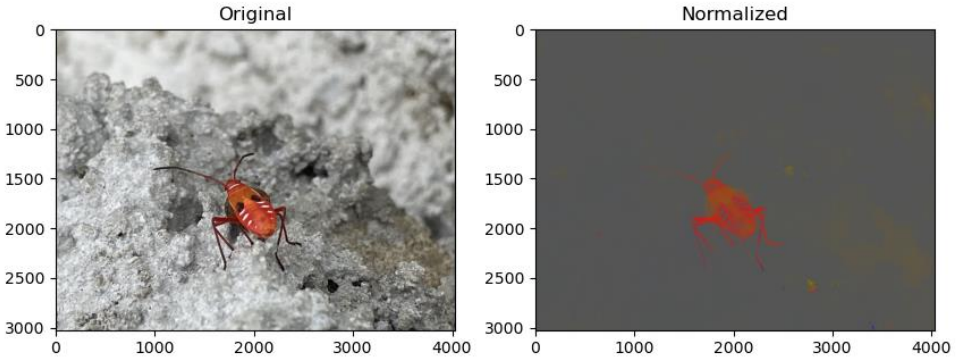


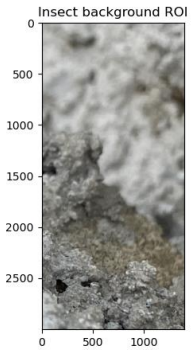


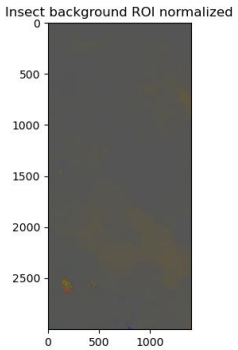
Codes and files: https://github.com/genvert/AP_157_FX-2_Vertudez/tree/main/Activity%205

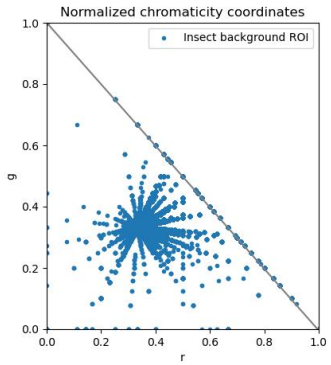
PARAMETRIC VS NONPARAMETRIC

Here shown on the right is the original and normalized image of the bug.









And here below shows the ROI chosen in the background. Look at the cool pattern made in the NCC of the ROI! It looks like a spider web LOL.

Codes and files: https://github.com/genvert/AP_157_FX-2_Vertudez/tree/main/Activity%205

REFLECTION

This activity was very very fun. I struggled at first to understand how to tag the membership of the pixels, but once I got it, it was easy from there.

I am really glad I got to use the random picture of the cotton strainer bug that I took just outside NIP on the way to UPTC. I interchangeably used the terms “insect” and “bug” here since I did not want to be technical.

It was satisfying to see the colors getting segmented, especially the blue, white, green, and yellow which were almost perfect!

I can say that this has been my favorite activity of them all.

SELF-GRADE

- Technical correctness: 35/35
  - I am confident that I understood the difference between parametric and nonparametric segmentation.
- Quality of presentation: 35/35
  - I have explained the theory and the steps on manually segmenting colors in images using Python. Distinction of the parts of the activity was clear.
- Self-reflection: 30/30
  - This activity was fun. I struggled at first understanding how to obtain the Gaussian probability of each pixel and how to use it to tag the membership of pixels, and how to obtain the 2D histogram of the ROI and how to use it to tag membership as well. Eventually, I got them and segmenting images based on colors was fun.
- Initiative: 10/10
  - I did every color in the Rubik's cube. Plus, I took a nice photo of the *Dysdercus koenigii* and was able to isolate it in the image.

REFERENCES

[1] Soriano, M. (2023). AP 157 module. Activity 5 - Feature Extraction Image Segmentation (Part 1 of 3).

[2] Lyon, R. a Gretag–Macbeth ColorChecker color chart being held in a photogrpahic portrait setting. Retrieved from https://en.wikipedia.org/wiki/ColorChecker#/media/File:Gretag-Macbeth_ColorChecker.jpg