HOMEWORK 2 U1
ADAPTIVE THRESHOLD

Héctor Mauricio Mendoza Xicoténcatl

Abstract—In this report we documented our coding process of basic segmentation operations. From the selected images we chose the most suitable operation.

Index Terms—Thresholds, Color Segmentation, RGB, HSV, OpenCv2, Bitwise Operation, Adaptive Threshold, Adaptive Gaussian.

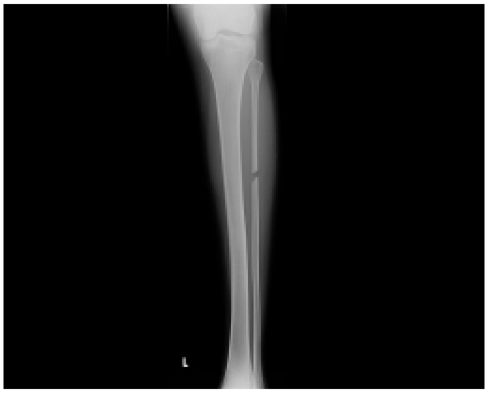
1 ADAPTIVE THRESHOLD

ADaptive thresholding is the method where the threshold value is calculated for smaller regions and therefore, there will be different threshold values for different regions. Basic threshold has problems with images that may have different illumination. Unlike fixed threshold, the threshold value at each pixel location depends on the neighboring pixel intensities. There are two types of adaptive thresholding:

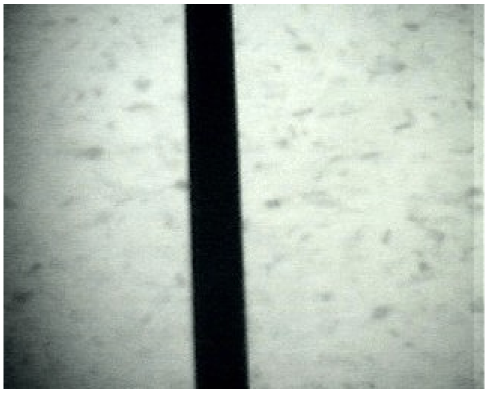
1) Adaptive Mean.
2) Gaussian.

2 PROBLEM TO SOLVE

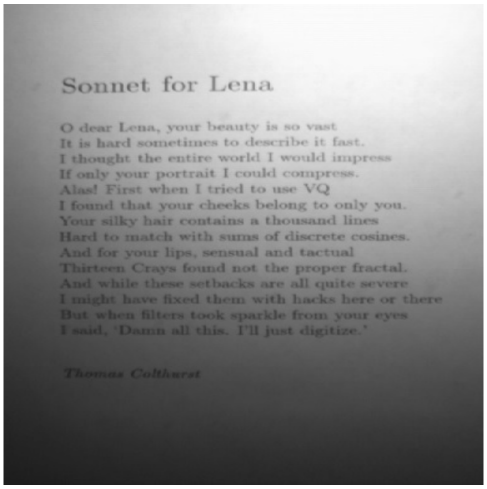
From the attached images perform the following:
Apply the most suitable Threshold Operation to segment a new image showing up only the bone.



Apply the most suitable Threshold Operation to segment a new image showing up only the stick.



Apply the most suitable Threshold Operation to segment a new image showing up only the text.



Sonnet for Lena

O dear Lena, your beauty is so vast
It is hard sometimes to describe it fast.
I thought the entire world I would impress
If only your portrait I could compress.
Alas! First when I tried to use VQ
I found that your cheeks belong to only you.
Your silky hair contains a thousand lines
Hard to match with sums of discrete cosines.
And for your lips, sensual and tactual
Thirteen Crays found not the proper fractal.
And while these setbacks are all quite severe
I might have fixed them with hacks here or there
But when filters took sparkle from your eyes
I said, 'Damn all this. I'll just digitize.'

Thomas Colthurst

3 CODING PROCESS

3.1 FIRST IMAGE

In order to get the output image the most similar to the one requested, we decided to use the **"Binary Threshold"** for the first image.
First, we first imported the libraries needed to run our code. Then, we used the command **cv2.imread(")** to read our source image which will be considered as our main input image.
With the **cv2.cvtColor(")** command, we convert our input image from BGR to gray-scale in order to apply the threshold. Then, we used the command: **cv2.threshold()** to put the corresponding data to apply the five thresholds.
Finally, we use the command **cv2.imshow(")** to show the final results of the output image.

3.2 SECOND IMAGE

For the second image, we decided to use the same type of threshold as for the exercise number one. This is the **"Binary Threshold"**. Besides being the same threshold used, we changed different value parameters that helped us to get an accurate, general result. Similar to the one requested. The difference between the use of both thresholds was the line:
ret, th2 = cv2.threshold(img2, 20, 255, cv2.THRESH_BINARY)
Within this line we changed the second parameter of the code, having a value of twenty. This value changed the white value to be considered by the program.

3.3 THIRD IMAGE

For this last image we decided to us two different type of unique adaptive thresholds **"Adaptive Median"** and **"Adaptive Gaussian**".
This types of thresholds is useful if an image has different lighting conditions in different areas. In that case, adaptive thresholding can help. Here, the algorithm determines the threshold for a pixel-based on a small region around it. So we get different thresholds for different regions of the same image which gives better results for images with varying illumination.

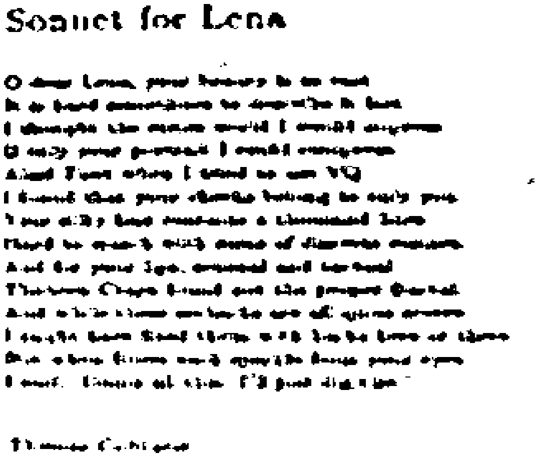
4 CONCLUSIONS

These exercises were the result of the first topics seen in unit one of Artificial Vision. This was helpful because we decided to make something practical with the knowledge learned. There were not many problems regarding the exercise, besides having some syntax coding mistakes.

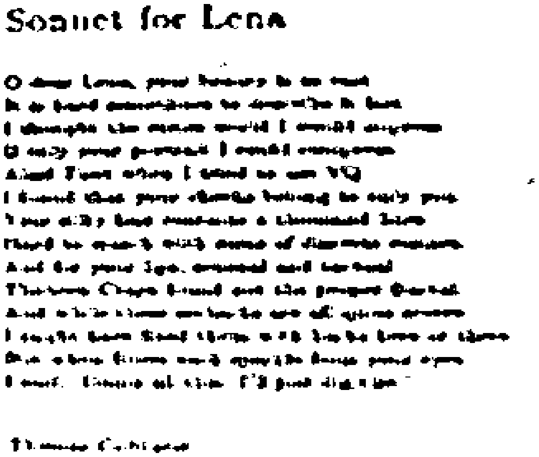
5 RESULTS

Bone Result - Binary Threshold

Stick Result - Binary Threshold



Stick Result - Mean Threshold



Stick Result - Gaussian Threshold

References

[1] ”OpenCV”, Es.wikipedia.org, 2020. [Online]. Available: https://es.wikipedia.org/wiki/OpenCV. [Accessed: 12-Sep- 2020]

[2] ”OpenCV: Image Thresholding”, Docs.opencv.org, 2020. [Online]. Available: https://docs.opencv.org/master/d7/d4d/tutorial_py_thresholding.html. [Accessed: 13- Sep- 2020]

Héctor Mendoza Héctor Mauricio Mendoza Xicoténcatl is a Computational Robotics Junior student, currently stu-diying Universidad Politécnica de Yucatán.