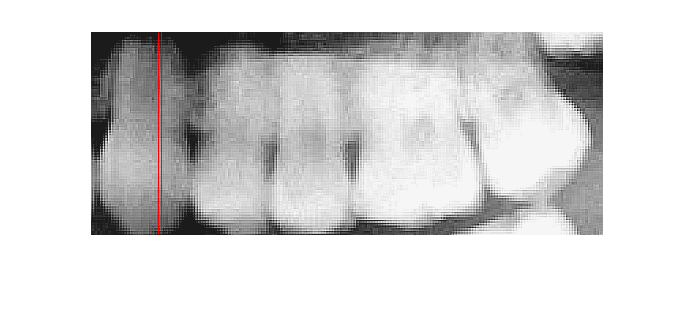
Creating the layout for the teeth requires first creating an improved contrast for the image, then segmenting the image into upper and lower halves, using probability to determine the dividing position. In order to separate the image effectively, I used the probability function to determine the most likely row, or “Y” position to place the dividing line. In the image below, the first image is the original, and the second is the contrasted image with the dividing line.

Horizontal Line Maximum Probability:

Y Line Position: 210



Since the image now can be divided in two, the upper half of the image will go through the same process used to divide the original image in half, though instead of using 1 line, four lines will be used to create separations for each of the teeth. To create these lines, the same probability function is used, though is ran for each line. However, when creating this (as can be seen in the image below) the probabilities (which rely on given points from the user) are skewed and merge to the same probability location. This may be due to data from the original points being reused; however, since data is written over, the found probabilities should be more accurate in correlation to the given points. In addition, the method to obtain points may be using the entire window, instead of the image coordinates, which could also affect the probabilities.

Upper Teeth



Bottom Teeth

The following data shows the probabilities of the lines shown above, to provide more detail on the lines.

Upper Teeth Maximum Probability:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| X Line Number | 1 | 2 | 3 | 4 |
| X Line Position | 36 | 72 | 73 | 73 |

Lower Teeth Maximum Probability:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| X Line Number | 1 | 2 | 3 | 4 |
| X Line Position | 1 | 39 | 93 | 93 |

Because of these problems, I was unable to create proper teeth bounding boxes. Since these boxes could not be created properly, I was unable to move forward to implement the methods to find the borders of each tooth. While the concept of finding the borders of the teeth is strait-forward, the implementation is far more challenging. This is because, while the concept of starting from a CenterPoint and using gradients to create vectors to determine the edge of a tooth is straightforward, attempting to get these gradients is not easily feasible due to the positions being skewed by the inclusion of the window in the obtained image.

In summary, teeth detection uses far more automation, which requires planning and preparing for problems with accessing data, as well as ensuring that the correct data is used with the method. Attempting to do the segmentation proved trying, as accessing the data was difficult due to the input method for getting the position of “Y-Hat” and each tooth. Moving forward to using the two methods to discover the individual teeth borders would also prove difficult, as getting the starting center point of each tooth would be skewed by the inclusion of the window, which would cause the algorithm to not center around the tooth, instead at a semi-random position. Both of these problems must be addressed to properly create mappings of the teeth of the image.