What has been done:

After the Office hour we outlined this method with which the handwritten text could be identified from the image. I tried to implement it using the steps below:

- Make the image 8-bit
- Split the image to top and bottom
- Take the histogram lists normalize it until a certain point (240 in this case)
- Use the line equation on the image to get rid of the background and change the bottom half of the image as well.
- Subtract the new image from the original one to keep only the “changed” parts of the image which will be the handwritten part.

I tried this method a couple of times, but it was not working for general cases when the printed text was mainly above for like one line, and the handwritten text was most of the page.

I searched for Threshold algorithms and found this one called Otsu's method that computes the threshold and applies it to create a binary image. I wrote the code and applied it to the image, which isolated the printed text. I will include the code in the github as well.

After isolating printed text I tried to manipulate the image a bit. I made it binary, sharpened and then dilated. Then I used the Hough algorithm to understand the angles of the lines and you can see the pictures below of what I got. Tried to understand what these lines mean.

OOP.MT.170317.H007.p1.jpg (75%)

219x1757 pixels; 8-bit; 2MB

AMERICAN UNIVERSITY OF ARMENIA

College of Science and Engineering

CS 120 Introduction to Object-Oriented Programming

MIDTERM EXAM

Date, Time: Friday, March [illegible] 2017 at [illegible]:50

Duration: 2 hours

Attention: **ANY TYPE OF COMMUNICATION IS STRICTLY PROHIBITED!!**

Write down your section number and ID at the top of all pages!

Participation

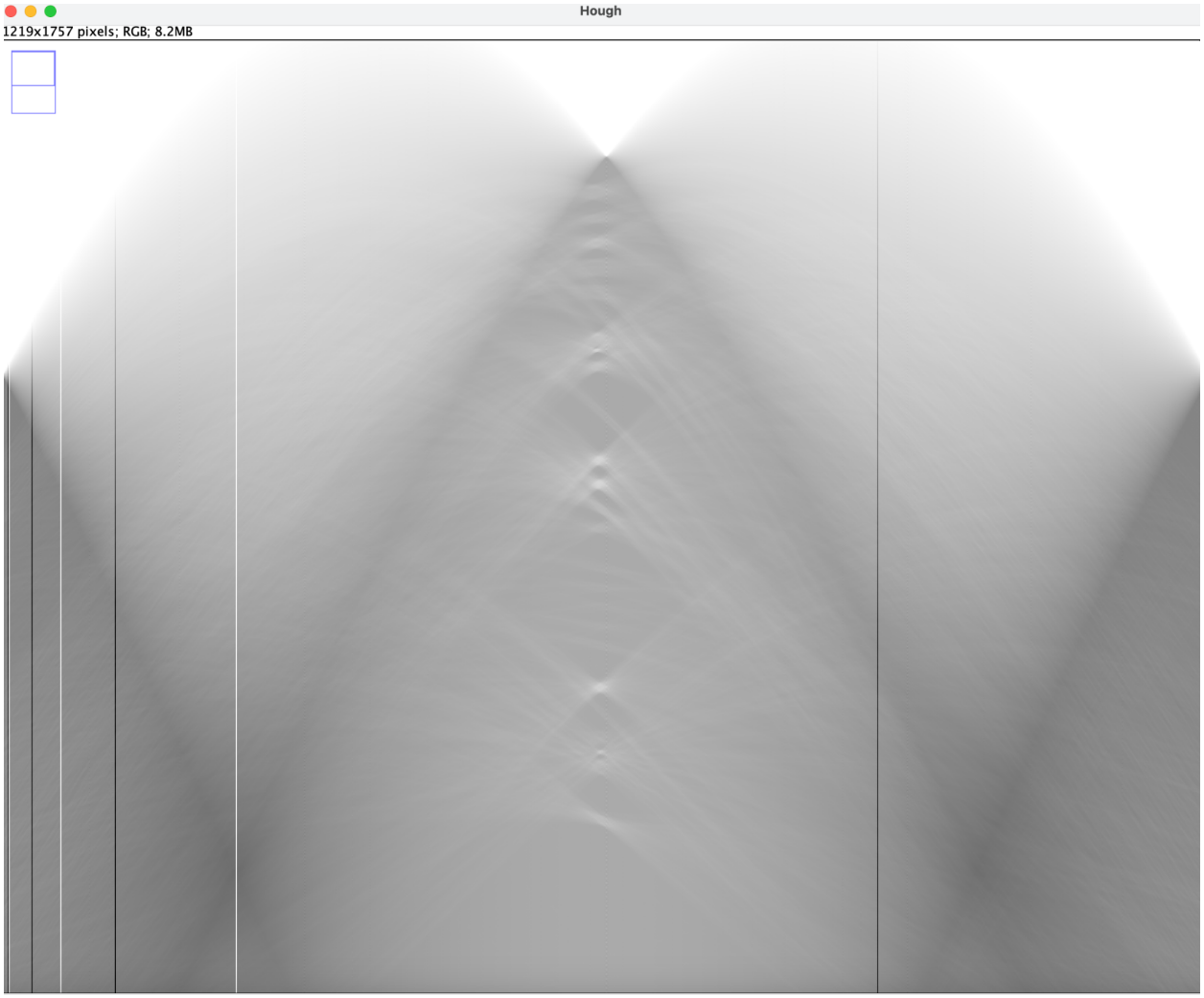
Problem 1: Consider below a C++ function *float kahan(float num1, float num2, float& compensation)* that implements the *Kahan Summation Algorithm* for high-precision compensated summation of its arguments *float num1* and *float num2*:

```
float kahan(float num1, float num2, float& compensation)
{
    float ...
    ...
    ...
    ...
    return ...;
}
```

Using this function, write a C++ function *float pi(int n)* that computes the value π by the following formula

$$\pi = 16\sum_{k=0}^{n}\frac{(-1)^k}{(2k+1)5^{2k+1}} - 4\sum_{k=0}^{n}\frac{(-1)^k}{(2k+1)239^{2k+1}} = \frac{16}{1\cdot 5} - \frac{4}{1\cdot 239} - \frac{16}{3\cdot 5^3} + \frac{4}{3\cdot 239^3} + \cdots$$

The initial value of *float compensation* is *0.0*.


Hough
1219x1757 pixels; RGB; 8.2MB