

**Name:** Alif Rahi

**Section:** CSCI 381 - Computer Vision / Tues-Thurs 1:40-2:55pm

**Project:** 1

**Due:** Feb 9, 2023

## **Main algorithm steps:**

**Step 0:** inFile1, inFile2, outFile1, debugFile open via argv []

**Step 1:** numRows, numCols, minVal, maxVal read from inFile1. x1, y1, x2, y2 read from inFile2. histAry dynamically allocate (size of maxVal + 1) and initialized to zero. maxHeight loadHist (histAry, inFile) // loadHist ( ) returns the largest value of histogram. dynamically allocate all other arrays and initialized to zero.

**Step 2:** dispHist (...)

**Step 3:** deepestThrVal deepestConcavity (x1, y1, x2, y2, histAry, debugFile) outFile1 output DeepestThrVal to outFile with caption.

**Step 4:** BiGaussThrVal biGaussian (histAry, GaussAry, maxHeight, minVal, maxVal, debugFile, outFile1) output BiGaussThrVal with caption

**Step 5:** close all files

## Source code

```
#include <iostream>
#include <math.h>
#include <fstream>
using namespace std;

int maxHeight;
int r, c, minN, maxX;
int *histArr;
int *GausArr;

void setZero(int *arr)
{
    int i = 0;
    while (arr[i] != '\0')
    {
        arr[i] = 0;
    }
}

int calculateThreshold(ofstream &myfile, char inFileName[], int *arr, int max)
{
    ifstream take(inFileName);
    double x1, y1, x2, y2;
    take >> x1 >> y1 >> x2 >> y2;
    double slope = (double)(y2 - y1) / (double)(x2 - x1);
    double b = (double)y1 - (slope * (double)x1);
    int maxDiff = INT_MIN;
    int idx;
    int *diffArr = new int[max + 1];
    myfile << "Two Peak points: (" << x1 << ", " << y1 << "), "
        << "(" << x2 << ", " << y2 << ")" << endl;
    setZero(diffArr);
    for (int i = x1; i <= x2; i++)
    {
        int yValue = (int)slope * i + b;
        diffArr[i] = abs(arr[i] - yValue);

        if (diffArr[i] > maxDiff)
        {

```

```

        maxDiff = diffArr[i];
        idx = i;
    }
}

take.close();

return idx;
}

int displayHist(ofstream &myfile, char **argv)
{
    ifstream take(argv[1]);
    take >> r;
    take >> c;
    take >> minN;
    take >> maxX;
    histArr = new int[maxX + 1];
    setZero(histArr);
    for (int i = 0; i < r; i++)
    {
        int idx;
        int val;
        take >> idx;
        take >> val;
        histArr[idx] = val;
    }

    for (int i = 0; i <= maxX; i++)
    {
        myfile << i << " "
            << "(" << histArr[i] << "):";
        int c = 0;
        while (c < histArr[i])
        {
            myfile << "+";
            c++;
        }
        myfile << "\n";
    }

    int thrsHold = calculateThreshold(myfile, argv[2], histArr, maxX);

```

```

myfile << "DeepestThrVal: " << thrshold << endl;

take.close();

return thrshold;
}

double computeMean(ofstream &debugFile, char **argv, int l1, int l2, int &max,
int *histArr)
{
    debugFile << "Entering computeMean method" << endl;
    double top = 0;
    double bottom = 0;
    max = 0;
    for (int i = l1; i < l2; i++)
    {
        top += (histArr[i] * i);
        bottom += histArr[i];
        if (max < histArr[i])
        {
            max = histArr[i];
        }
    }

    double result = ((double)top / (double)bottom);

    debugFile
        << "Leaving computeMean method. MaxHeight = " << maxHeight << ", Result
= " << result << endl;

    return ((double)top / (double)bottom);
}

double computeVariance(ofstream &debugFile, char **argv, int l1, int l2, int
mean, int *histArr)
{
    debugFile << "Entering computeVariance method" << endl;
    double top = 0;
    double bottom = 0;
    for (int i = l1; i < l2; i++)
    {
        top += (double)histArr[i] * ((double)i - mean) * ((double)i - mean);
    }
}

```

```

        bottom += histArr[i];
    }

    debugFile << "Leaving computeVariance method" << endl;
    return ((double)top / (double)bottom);
}

double modifiedGauss(int idx, double mean, double var, int max)
{
    double Gval = 0;
    Gval = max * exp(-(((idx - mean) * (idx - mean)) / (2 * var)));
    return Gval;
}

double fitGaussian(ofstream &debugFile, char **argv, int *GausArr, int l1, int
l2, int max, int *histArr)
{
    int maxt;
    debugFile << "Entering fitGauss method" << endl;
    double mean = computeMean(debugFile, argv, l1, l2, maxHeight, histArr);
    double var = computeVariance(debugFile, argv, l1, l2, mean, histArr);
    double Gval = 0.0;
    double sum = 0.0;
    for (int i = l1; i <= l2; i++)
    {
        Gval = modifiedGauss(i, mean, var, maxHeight);
        sum += abs(Gval - (double)histArr[i]);
        GausArr[i] = (int)Gval;
    }
    debugFile << "Leaving fitGauss method, sum is: " << sum << endl;

    return sum;
}

int biGaussian(ofstream &debugFile, ofstream &myfile, char **argv)
{
    debugFile << "Entering fitGauss method" << endl;
    GausArr = new int[maxX + 1];
    int offSet = (int)(maxX - minN) / 10;

    int divPtr = offSet;

```

```

    int bestThr = divPtr;
    double minSumDiff = 999999.0;
    double total = 0.0;

    while (divPtr < maxX - offSet)
    {
        setZero(GausArr);
        double leftSum = (double)fitGaussian(debugFile, argv, GausArr, 0,
divPtr, maxX, histArr);
        cout << "Sum1: " << leftSum << endl;
        double rightSum = (double)fitGaussian(debugFile, argv, GausArr, divPtr,
maxX, maxX, histArr);
        cout << "Sum2: " << rightSum << endl;
        total = (leftSum + rightSum);
        cout << "Total: " << total << endl;
        if (total < minSumDiff)
        {
            minSumDiff = total;
            bestThr = divPtr;
            cout << "Total < MinDiff, so BestThr: " << bestThr << endl;
        }
        divPtr++;
    }

    debugFile << "Leaving biGaussian method, minSumDiff = " << minSumDiff << ",
bestThr = " << bestThr << endl;
    myfile << "BiGaussThrVal: " << bestThr << endl;

    return bestThr;
}

void loadHist(int &maxHeight, int *histArr)
{
    maxHeight = INT_MIN;
    int i = 0;
    while (histArr[i] != '\0')
    {
        if (histArr[i] > maxHeight)
        {
            maxHeight = histArr[i];
        }
        i++;
    }
}

```

```

        cout << maxHeight << endl;
    }

int main(int argc, char **argv)
{
    ofstream myfile(argv[3]);
    ofstream debugFile(argv[4]);
    displayHist(myfile, argv);
    loadHist(maxHeight, histArr);
    biGaussian(debugFile, myfile, argv);

    myfile.close();
    debugFile.close();

    return 0;
}

```

## Output

```

0 (10):*****
1 (14):*****
2 (17):*****
3 (20):*****
4 (22):*****
5 (31):*****
6 (28):*****
7 (33):*****
8 (48):*****
9 (86):*****
10 (70):*****
11 (80):*****
12 (120):*****
13 (150):*****
14 (182):*****
15 (210):*****
16 (182):*****
17 (172):*****
18 (132):*****
19 (100):*****
20 (89):*****
21 (78):*****
22 (62):*****
23 (50):*****
24 (48):*****
25 (18):*****
26 (8):*****
27 (6):*****
28 (6):*****
29 (7):*****
30 (6):*****
31 (5):*****
32 (4):*****
33 (4):*****
34 (5):*****
35 (8):*****
36 (10):*****
37 (12):*****
38 (22):*****
39 (25):*****
40 (40):*****
41 (45):*****
42 (72):*****
43 (80):*****
44 (80):*****
45 (100):*****
46 (120):*****
47 (150):*****
48 (180):*****
49 (190):*****
50 (170):*****
51 (140):*****
52 (120):*****
53 (110):*****
54 (80):*****
55 (80):*****
56 (70):*****
57 (60):*****
58 (30):*****
59 (20):*****
60 (12):*****
61 (8):*****
62 (6):*****
63 (5):*****

```

**Two Peak points: (15, 210), (49, 190)**

**DeepestThrVal: 32**

**BiGaussThrVal: 31**



## Debug (first page of debug file)

Entering fitGauss method  
Entering fitGauss method  
Entering computeMean method  
Leaving computeMean method. MaxHeight = 31, Result = 3.07895  
Entering computeVariance method  
Leaving computeVariance method  
Leaving fitGauss method, sum is: 65.0047  
Entering fitGauss method  
Entering computeMean method  
Leaving computeMean method. MaxHeight = 210, Result = 32.9012  
Entering computeVariance method  
Leaving computeVariance method  
Leaving fitGauss method, sum is: 5237.59  
Entering fitGauss method  
Entering computeMean method  
Leaving computeMean method. MaxHeight = 31, Result = 3.65493  
Entering computeVariance method  
Leaving computeVariance method  
Leaving fitGauss method, sum is: 72.7634  
Entering fitGauss method  
Entering computeMean method  
Leaving computeMean method. MaxHeight = 210, Result = 33.0919  
Entering computeVariance method  
Leaving computeVariance method  
Leaving fitGauss method, sum is: 5193.77  
Entering fitGauss method  
Entering computeMean method  
Leaving computeMean method. MaxHeight = 33, Result = 4.28571  
Etc.....  
Leaving fitGauss method, sum is: 121.199  
**Leaving biGaussian method, minSumDiff = 1206.17, bestThr = 31**