Computer Vision Programming Language: C++

Project #1 Histogram

Andres Quintero

Due Date:

Soft copy: 2/4/2020

Hard copy: 2/6/2020

Main():

Step 0: Open input file and output files, assign threshold argument to variable thresholdValue

Step 1: Read header values from input file and assign to varibles-numRows, numCols, minVal, maxVal

Step 2: Dynamically create 1-D array, histAry, of size maxVal+1 and fill with zeros

Step 3: Call ComputeHist() method

Step 4: Call PrintHist() method

Step 5: Close input file

Step 6: Re-open input file

Step 7: Call thresholdOp() method

Step 8: Close second output file

Step 9: Re-open second output file

Step 11: Call prettyPrint() method

Step 12: Close all files

**Source Code: main.cpp**

#include <iostream>

#include <fstream>

#include <string>

using namespace std;

// Prototypes

void ComputeHist(fstream& inFile, int\* histAry);

void PrintHist(int\* histAry, fstream& outFile1, int numRows, int numCols, int minVal, int maxVal);

void thresholdOp(fstream& inFile, int thresholdValue, fstream& outFile2);

void prettyPrint(fstream& outFile2, fstream& outFile3);

// MAIN

int main(int argc, char\* argv[]){

cout << argv[1] << " is the input image file" << endl;

cout << argv[2] << " is the histogram file" << endl;

cout << argv[3] << " is the threshold image file" << endl;

cout << argv[4] << " is the prettyPrint thresold file" << endl;

// Variables

int numRows, numCols, minVal, maxVal, thresholdValue;

fstream inFile(argv[1]); // Input File

fstream outFile1(argv[2]); // Histogram

fstream outFile2(argv[3]); // Threshold image

fstream outFile3(argv[4]); // Pretty Threshold image

if (!inFile || !outFile1 || !outFile2 || !outFile3) {

cout << "There is an error in opening the file" << endl;

cout << "Exiting program" << endl;

exit(1);

}

// Reading first 4 values AKA the 'header' AND threshold value

inFile >> numRows;

inFile >> numCols;

inFile >> minVal;

inFile >> maxVal;

thresholdValue = stoi(argv[5]);

// Printing values out to console

cout << "Rows: " << numRows << endl;

cout << "Cols: " << numCols << endl;

cout << "Min: " << minVal << endl;

cout << "Max: " << maxVal << endl;

cout << "Threshold: " << thresholdValue << endl;

// Dynamically creatinng a 1-D array THEN filling them with zer0s

int\* histAry = new int[maxVal + 1];

for(int i = 0; i < maxVal+1; i++){

histAry[i] = 0;

}

// Fills histAry from inFile

ComputeHist(inFile, histAry);

// Prints histogram

PrintHist(histAry, outFile1, numRows, numCols, minVal, maxVal);

//Re-opening inFile

inFile.close();

inFile.open(argv[1]);

// Prints thresold image w/ header

thresholdOp(inFile,thresholdValue, outFile2);

//Re-opening outFile2

outFile2.close();

outFile2.open(argv[3]);

// Pretty print thresold image w/o header

prettyPrint(outFile2, outFile3);

// Closing all files

inFile.close();

outFile1.close();

outFile2.close();

outFile3.close();

return 0;

}

void prettyPrint(fstream& outFile2, fstream& outFile3){

int numRows, numCols, minVal, maxVal;

outFile2 >> numRows;

outFile2 >> numCols;

outFile2 >> minVal;

outFile2 >> maxVal;

// Comparing only 1 and 0, no need for actual thresold value

int pixel\_val;

for(int iRow = 0; iRow < numRows; iRow++){

for(int iCol = 0; iCol < numCols; iCol++){

outFile2 >> pixel\_val;

if (pixel\_val > 0) {

outFile3 << 1 << " ";

} else {

outFile3 << " " << " ";

}

}

outFile3 << endl;

}

}

void thresholdOp(fstream& inFile, int thresholdValue, fstream& outFile2){

int numRows, numCols, minVal, maxVal;

inFile >> numRows;

inFile >> numCols;

inFile >> minVal;

inFile >> maxVal;

// New header, with min-max as 0-1

outFile2 << numRows << " " << numCols << " " << 0 << " "<< 1 << endl;

// Write 1 if pixel value is greater than threshold value (We pick thresold)

int pixel\_val;

for(int iRow = 0; iRow < numRows; iRow++){

for(int iCol = 0; iCol < numCols; iCol++){

inFile >> pixel\_val;

if (pixel\_val >= thresholdValue) {

outFile2 << 1 << " ";

} else {

outFile2 << 0 << " ";

}

}

outFile2 << endl;

}

}

void PrintHist(int\* histAry, fstream& outFile1, int numRows, int numCols, int minVal, int maxVal){

//Writing header

outFile1 << numRows << " " << numCols << " " << minVal << " "<< maxVal << endl;

// Writing histogram

for(int i = 0; i < maxVal+1; i++){

int totalDisplayed;

outFile1 << i << " (" << histAry[i] << "):";

if( histAry[i] > 0) {

if (histAry[i] > 60) { //Setting max + printed to be 60

totalDisplayed = 60;

} else {

totalDisplayed = histAry[i];

}

for(int j = 0; j < totalDisplayed; j++) {

outFile1 << "+";

}

}

outFile1 << endl;

}

outFile1.close();

}

void ComputeHist(fstream& inFile, int\* histAry) {

int pixel\_val;

while(inFile >> pixel\_val){

histAry[pixel\_val]++;

}

}

**Input file: HistogramThreshold\_Data.txt**

31 40 0 9

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

2 3 1 3 2 3 1 1 2 3 1 1 2 1 1 1 2 1 0 1 1 0 0 0 0 0 0 0 0 0 2 3 1 3 2 3 1 1 2 3

0 1 1 9 2 1 2 1 2 9 2 2 2 1 2 1 2 1 2 1 8 1 2 1 1 1 2 1 1 1 2 1 1 9 9 1 1 1 2 1

2 3 2 3 2 3 2 1 2 3 2 1 2 3 2 3 2 3 1 7 9 9 1 1 2 3 2 1 2 3 2 3 1 1 0 3 1 1 2 3

0 0 2 3 1 8 2 3 0 2 3 1 2 3 1 1 2 3 9 8 8 7 9 2 3 1 1 2 3 2 3 1 1 2 2 3 1 1 2 3

1 0 1 2 0 2 2 0 3 0 3 0 1 0 2 0 1 7 7 9 9 8 8 7 0 1 2 0 2 2 0 3 0 3 0 1 0 2 0 2

0 0 2 1 1 1 2 1 1 1 2 1 1 1 2 1 9 8 8 7 9 8 0 7 9 0 0 0 1 1 2 1 1 1 2 1 1 1 2 1

2 1 1 1 2 1 1 1 2 1 1 1 2 1 1 8 7 9 9 7 7 0 7 8 8 8 2 1 1 1 2 1 1 1 2 1 1 1 2 1

0 1 2 1 0 1 2 1 2 1 2 2 2 0 8 7 9 8 7 7 9 9 8 8 7 9 9 1 2 0 2 0 1 3 2 0 1 1 2 0

2 3 2 3 2 3 2 1 2 3 2 1 2 8 7 9 8 7 7 9 9 8 8 7 9 9 7 9 1 1 2 1 1 1 2 1 1 1 2 1

0 1 3 2 0 1 1 2 0 0 0 0 8 7 9 8 7 7 9 9 8 8 7 9 9 8 7 9 9 0 0 0 0 3 2 0 1 1 2 0

0 0 0 3 2 0 1 1 2 0 0 8 7 9 8 7 7 9 9 8 8 7 0 0 7 9 8 8 7 8 3 2 0 1 1 2 0 0 0 0

0 3 2 0 1 1 2 0 1 0 7 9 8 7 7 9 9 8 8 7 9 9 0 8 9 9 8 8 7 8 9 0 0 3 2 0 1 1 2 0

0 3 2 0 1 1 2 0 0 9 8 8 8 9 8 8 7 7 7 9 9 7 9 9 8 8 7 8 9 7 7 9 3 2 0 1 1 2 0 0

0 1 3 2 0 1 1 2 7 9 8 7 7 9 9 8 9 8 8 7 0 0 7 7 8 7 8 9 7 8 8 7 9 3 2 0 1 1 2 0

3 2 0 1 1 2 0 9 8 9 8 8 7 9 8 7 7 9 9 8 0 7 9 9 6 9 8 8 7 8 6 9 8 8 2 1 1 1 2 3

0 0 0 0 0 0 1 1 7 9 9 6 9 8 8 7 9 9 8 8 7 9 9 8 7 9 9 6 9 8 8 7 9 2 1 2 1 0 0 0

0 0 1 9 0 0 0 1 1 9 8 8 7 9 0 0 7 9 9 8 8 7 9 9 6 0 0 8 7 8 6 9 2 1 0 2 9 0 0 0

0 0 8 1 0 0 0 0 1 1 8 7 7 9 0 0 9 8 8 7 9 9 7 7 8 0 0 9 7 8 8 2 1 0 0 1 9 0 0 0

3 2 0 1 1 2 0 0 0 1 1 8 8 9 8 8 7 7 7 9 9 7 0 9 8 8 7 8 9 7 2 1 8 9 0 0 0 0 0 0

0 1 9 3 0 1 1 2 0 0 1 1 8 7 7 9 9 8 8 7 9 9 7 7 9 9 8 8 7 2 1 0 8 9 2 0 1 1 2 0

0 0 2 9 0 1 1 2 0 0 0 1 1 9 8 7 7 9 9 8 8 7 9 9 1 9 8 8 2 1 1 1 2 1 1 1 2 1 1 1

0 2 0 1 3 2 0 1 1 2 0 1 1 1 9 8 7 7 9 9 8 8 7 9 9 8 7 2 1 0 2 0 1 3 2 0 1 1 2 0

0 1 3 2 0 1 1 2 0 0 0 0 0 1 1 9 8 7 7 9 9 8 8 7 9 9 2 1 1 1 2 1 1 1 2 1 1 1 2 1

1 1 2 1 1 1 2 1 1 1 2 1 2 0 1 1 9 8 7 7 9 9 8 8 7 2 1 1 2 0 2 0 1 8 9 0 1 1 2 0

2 3 2 1 2 3 2 1 2 0 0 0 0 0 0 1 1 9 9 7 7 9 9 8 2 1 1 1 1 1 2 1 1 8 9 1 1 1 2 1

0 0 8 9 0 3 2 0 1 1 2 0 1 0 0 0 1 1 8 7 9 9 7 2 1 0 0 0 1 1 2 1 1 1 2 1 1 1 2 1

0 2 2 9 1 2 3 2 1 2 0 0 0 0 0 0 0 1 1 9 9 8 0 1 0 1 2 0 2 2 0 3 9 3 0 1 0 2 0 2

0 0 0 0 0 0 0 0 0 2 3 2 7 2 3 2 1 2 1 1 8 0 1 0 0 0 0 0 0 0 0 0 0 0 0 8 0 0 0 0

0 1 1 1 2 1 2 1 2 8 2 2 2 1 2 1 2 1 2 1 2 1 2 1 1 1 2 1 1 1 2 1 1 1 2 1 1 1 2 1

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

**Output File 1: Histogram**

31 40 0 9

0 (313):++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

1 (294):++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

2 (196):++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

3 (64):++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

4 (0):

5 (0):

6 (6):++++++

7 (102):++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

8 (124):++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

9 (141):++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

**Output File 2: Threshold image**

31 40 0 1

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 0 0 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0

0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0

0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0

0 0 0 1 0 0 0 0 0 1 1 1 1 1 0 0 1 1 1 1 1 1 1 1 1 0 0 1 1 1 1 1 0 0 0 0 1 0 0 0

0 0 1 0 0 0 0 0 0 0 1 1 1 1 0 0 1 1 1 1 1 1 1 1 1 0 0 1 1 1 1 0 0 0 0 0 1 0 0 0

0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 0 0 1 1 0 0 0 0 0 0

0 0 1 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 1 1 0 0 0 0 0 0

0 0 0 1 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0

0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0

0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

**Output File 3: Pretty Threshold Image**

1 1 1 1 1

1 1 1

1 1 1 1 1 1

1 1 1 1 1 1 1

1 1 1 1 1 1 1 1

1 1 1 1 1 1 1 1 1 1

1 1 1 1 1 1 1 1 1 1 1 1 1

1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

1 1 1 1 1 1 1 1 1 1 1 1 1

1 1 1 1 1 1 1 1 1 1 1

1 1 1 1 1 1 1 1 1 1 1

1 1 1 1 1 1 1 1 1

1 1 1 1 1 1 1

1 1 1 1 1

1 1 1

1