CSC 381 Computer Vision ( C++ )

Project 1.1: Histogram

Han Wen Loh

Soft copy: 02/01/2019

Hard copy: 02/05/2019

Algorithm Steps:

step 0: open input and output files

step 1: numRows, numCols, minVal, maxVal <-- get from input file

dynamically allocate the hist array and initialize to 0

step 2: // process the input file from left to right and top to bottom

p(i,j) <- read from input // you must read one integer at a time

hitogram[p(i,j)]++

step 3: repeat step 2 until the file is empty

step 4: output histogram array to output file // follow the format given

step 5: close input file and output file

/\*

find Histogram of an image

\*/

#include <fstream>

#include <iostream>

using namespace std;

class Histogram {

private:

int numRows;

int numCols;

int minVal;

int maxVal;

int \*hist;

~Histogram() {

delete[] hist;

}

public:

Histogram(ifstream &inFile) {

inFile >> numRows >> numCols >> minVal >> maxVal;

hist = new int[maxVal + 1]();

}

void computeHistogram(ifstream &inFile, ofstream &outFile) {

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

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

int p;

inFile >> p;

++hist[p];

}

}

outFile << numRows << " " << numCols << " " << minVal << " " << maxVal;

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

outFile << endl << i << " " << hist[i];

}

inFile.close();

outFile.close();

}

};

int main(int args, char \*\*argv) {

ifstream inFile;

ofstream outFile;

inFile.open(argv[1]);

if (inFile.fail()) {

cout << "ERROR: cannot find \"" << argv[1] << "\"\n";

exit(1);

}

outFile.open(argv[2]);

Histogram \*hist = new Histogram(inFile);

hist->computeHistogram(inFile, outFile);

return 0;

}

31 40 0 9

0 313

1 294

2 196

3 64

4 0

5 0

6 6

7 102

8 124

9 141