

Homework2

Basic Image Manipulation

Description

Write a program to generate:

(a) a binary image (threshold at 128)

Scan every pixel, if the grayscale intensity is smaller than 128, then set it to 0. Otherwise, set it to 255.

Source code

```
for (int i = 0; i < img.rows; i++) {  
    for (int j = 0; j < img.cols; j++) {  
        if (A.at<uchar>(i, j) < 128) { // 0~127  
            A.at<uchar>(i, j) = 0;  
        }  
        else {  
            A.at<uchar>(i, j) = 255;  
        }  
    }  
}
```

(b) a histogram

1. Count the numbers of pixels of each intensity and save them into an array.
2. Find the maximum number of data to adjust the ratio of the graph. (since the y-axis is too high)
3. Create an image for the histogram
4. Plot the graph.

Source code

```
//1b histogram
//data collection
int value[256] = { 0 };
int max = 0;

for (int i = 0; i < img.rows; i++) {
    for (int j = 0; j < img.cols; j++) {
        value[img.at<uchar>(i, j)] ++;
    }
}

//highest data
for (int i = 0; i < 256; i++) {
    if (value[i] >= max) {
        max = value[i];
    }
}

//graph resize
float ratio = max / 768.0;
//float ratio = max / 256.0;

//background
//Mat graph(256, 256, CV_8UC1, Scalar(0)); //y, x
Mat graph(768, 768, CV_8UC1, Scalar(0)); //size*3

//plot
for (int x = 0; x < 256; x++) { //0~255
    for (int y = 0; y < value[x]; y++) { //number
        graph.at<uchar>(767 - floor(y / ratio), x * 3) = 255;
        graph.at<uchar>(767 - floor(y / ratio), x * 3 + 1) = 255;
        graph.at<uchar>(767 - floor(y / ratio), x * 3 + 2) = 255;
    }
}
```

(c) connected components (regions with + at centroid, bounding box)

I use a **4-connected components algorithm** and **iterative algorithm** to obtain the result.

1. Create an array and label the pixel. (-1 for white pixel, 0 for black pixel)
2. Implement the first top-down passing and label the connected pixels with different tags.
3. Repeat top-down passing and bottom-up passing in the “loop” function until no change.
4. Count the numbers of pixels of each tag.
5. Filter the tags that pixel counts less than 500 and omit it.

6. Find four vertices of the region and select the maximum coordinate for the bounding box.
7. Draw the bounding box and centroid by using rectangle() and circle() function.
8. Repeat step 6~7 to find all the bounding boxes.

Source code

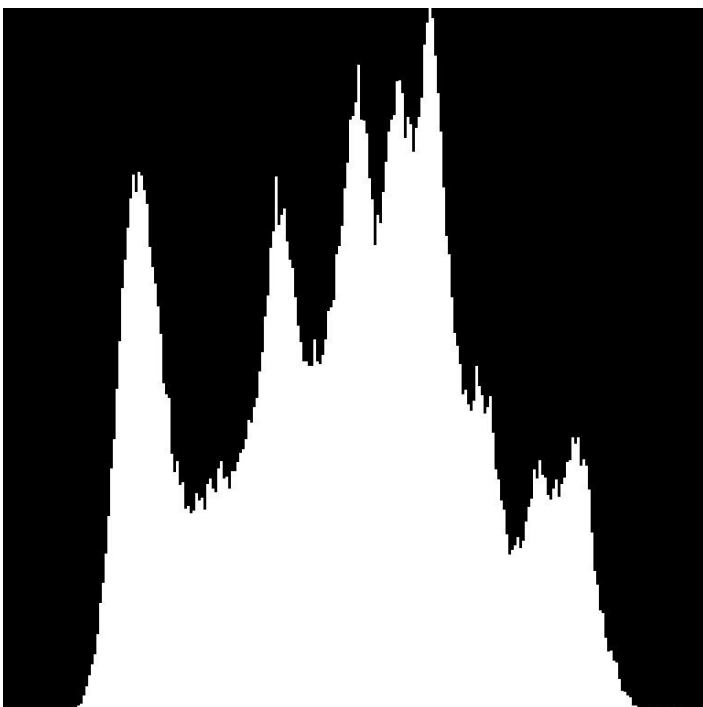
Please refer to the main.cpp. (Line 10~108, 164~364)

Result

(a) a binary image (threshold at 128)



(b) a histogram



(c) connected components(regions with + at centroid, bounding box)



Reference:

1. <https://docs.opencv.org/2.4/>
2. https://docs.opencv.org/3.4/d3/d63/classcv_1_1Mat.html
3. https://docs.opencv.org/3.4/d6/d6e/group_imgproc__draw.html