...b\11. Image Analysis with Microcomputer\exercises\Ex2-CV-main.cpp

```cpp
/*****************************************************************************************
*                                    EXERCISE 2                                          *
*                                   ------------                                         *
*                                  BINARY IMAGES                                        *
*****************************************************************************************/

#include <opencv2/opencv.hpp>
#include <iostream>
#include <string.h>
#include <stdlib.h>
#include <math.h>

#define LINES 'L'
#define RECTS 'R'
#define PI 3.14159265358979323846

using namespace std;
using namespace cv;

int main(int argc, char ** argv) {

    /*************************************************************************
    *                                 PART 1                                 *
    *                                --------                                *
    *                              Thresholding                              *
    *************************************************************************/
    /// Uploading and converting to grey scale
    IplImage * img = cvLoadImage(argv[1] , 1);
    IplImage * G   = cvCreateImage(cvGetSize(img) , IPL_DEPTH_8U , 1);
    cvCvtColor(img , G , CV_RGB2GRAY);

    /// Uploading directly in grey scale
    //IplImage * G = cvLoadImage(argv[1], CV_LOAD_IMAGE_GRAYSCALE);

    ///_____________________________Histogram_____________________________________
    ///                DRAW A PRELIMINARY SIMPLE HISTOGRAM

    IplImage * his = getHistogram(G , LINES);

    cvNamedWindow("Grey scale" , 0);
    cvShowImage("Grey scale"   , G);

    cvNamedWindow("histogram" , 0);
    cvShowImage("histogram"   , his);

    /// First threshold selection
    printf("\nTake a look at the grey scale image and histogram then press any key.\n");
    cvWaitKey(0);

    ///_____________________Binary Image Formation_____________________________


    // These are needed in PART 2 to calculate the coordinates of the centre of mass.
    // Declared now because they can be calculated in parallel with the thresholding.
    int tot = 0, xcm = 0, ycm = 0;

    IplImage * bin;
```

```
64      ///_____________INTERACTIVE THRESHOLDING - A little unstable...
65      // Grey scale histogram converted to 3-channel image first.
66      // It will be useful later to draw the threshold on the histogram.
67
68      IplImage * Chis = cvCreateImage(cvGetSize(his) , IPL_DEPTH_8U , 3);
69      cvCvtColor(his , Chis , COLOR_GRAY2RGB);
70      cvFlip(Chis , Chis , 0);
71
72      bin = interactiveThresholding(G , Chis , his , &tot , &xcm , &ycm);
73
74      cvSaveImage("histogram.png" , Chis);
75
76      ///_____________________SIMPLE THRESHOLDING_____________________________
77      /// Observe grey scale and histogram, select a threshold once, and hope!
78
79      //bin = simpleThresholding(G, his, &tot, &xcm, &ycm);
80
81
82      /*********************************************************************
83      *                                PART 2                              *
84      *                                --------                            *
85      *                            Centre of Mass                          *
86      *********************************************************************/
87      /// Coordinates
88      ycm /= tot;
89      xcm /= tot;
90
91      printf("\nCenter of mass: (%d , %d)\n", xcm, ycm);
92
93      /// Draw a cross in the picture
94      IplImage * I = cvCreateImage(cvGetSize(img), IPL_DEPTH_8U, 3);
95      cvCvtColor(bin, I, COLOR_GRAY2RGB);          // To draw in the binary image
96      //cvCvtColor(G, I, COLOR_GRAY2RGB);          // To draw in the grey scale image
97
98      /// Draw a cross in the center of mass
99      unsigned short A = 10;
100     cvLine(I, cvPoint((xcm - A),(ycm - A))  , cvPoint((xcm + A),(ycm + A)) ,
101            cvScalar(0,0,255) , 2, 4);
102     cvLine(I, cvPoint((xcm - A), (ycm + A)) , cvPoint((xcm + A),(ycm - A)) ,
103            cvScalar(0,0,255) , 2, 4);
104
105     /// Just a different style of cross
106     //cvLine(I, cvPoint(xcm,(ycm - A)), cvPoint(xcm,(ycm + A)),
107              cvScalar(255,0,255), 2, 4);
108     //cvLine(I, cvPoint((xcm - A), ycm), cvPoint((xcm + A), ycm),
109              cvScalar(255,0,255), 2, 4);
110
111
112
113     /*********************************************************************
114     *                                PART 3                              *
115     *                                --------                            *
116     *                            Image Moments                           *
117     *********************************************************************/
118     double M_20 = 0, M_02 = 0, M_11 = 0;
119     reducedCentralMoments(bin, xcm, ycm, &M_20, &M_02, &M_11);
120
121     /// Inclination of Principal Axis
122     double th = 0.5 * atan2((2 * M_11), (M_20 - M_02));
123     double th_deg = (th / PI) * 180;
124     printf("\nInclination of main axis (deg): %.2f\n", th_deg);
125     double m = tan(th);
126     printf("\nAngular Coefficient: %.2f\n", m);
```

```
127         /// Draw a line at an angle theta, through the centre of mass
128         cvLine(I, cvPoint((xcm - 30),(ycm - 30*m)), cvPoint((xcm + 30),(ycm + 30*m)),
129                cvScalar(255,0,0), 2, 4);
130
131         cvNamedWindow("Center of Mass", 0);
132         cvShowImage("Center of Mass", I);
133
134         cvSaveImage("Binary.png", bin);
135         cvSaveImage("Center_of_Mass.png", I);
136
137
138
139         /*****************************************************************
140         *                                PART 4                          *
141         *                               --------                         *
142         *                          Moment Invanriances                   *
143         *****************************************************************/
144
145         ///_____________________________________________________________IN CLASS
146
147
148         //*****************************************************************
149
150         ///_____________________________Clean-Up_____________________________
151
152         cvWaitKey(0);
153
154         cvReleaseImage(&img);
155         cvReleaseImage(&his);
156         cvReleaseImage(&G);
157         cvReleaseImage(&bin);
158         cvReleaseImage(&I);
159         cvReleaseImage(&Chis);  // Comment this out if you're doing simple thresholding
160
161         cvDestroyAllWindows();
162
163         return 0;
164     }
```