

**FACULTY OF INFORMATION TECHNOLOGIES**

**DEPARTMENT OF INFORMATION SYSTEMS MANAGEMENT**

**MACHINE VISION**

*Report*

**Laboratory work #3**

|  |  |
| --- | --- |
|  | **Done by: Sailaubekova Zhansaya** |
|  | **Checked by: Auezova Anel** |

**Almaty 2015**

**Source code: main.cpp**

//

// main.cpp

// opencv2

//

// Created by Zhansaya on 22.09.15.

// Copyright (c) 2015 Zhansaya. All rights reserved.

//

#include <stdio.h>

#include "opencv2/imgproc/types\_c.h"

#include "main.h"

int main(){

Mat src = imread("/Users/Zhansaya/Desktop/fruits.jpg");

if (src.empty())

return -1;

Mat roi(src, Rect(88,92,152,158));

Mat gray;

cvtColor(src, gray, CV\_BGR2GRAY);

imshow("src", src);

imshow("gray", gray);

imshow("roi", roi);

showHistogram(src);

showHistogram(gray);

showHistogram(roi);

waitKey(0);

return 0;

}

**Source code: main.h**

//

// main.h

// opencv2

//

// Created by Zhansaya on 22.09.15.

// Copyright (c) 2015 Zhansaya. All rights reserved.

//

#ifndef opencv2\_main\_h

#define opencv2\_main\_h

#include <opencv2/highgui/highgui.hpp>

#include <opencv2/imgproc/imgproc.hpp>

using namespace cv;

using namespace std;

void showHistogram(Mat& img)

{

int bins = 256; // number of bins

int nc = img.channels(); // number of channels

vector<Mat> hist(nc); // array for storing the histograms

vector<Mat> canvas(nc); // images for displaying the histogram

int hmax[3] = {0,0,0}; // peak value for each histogram

for (int i = 0; i < hist.size(); i++)

hist[i] = Mat::zeros(1, bins, CV\_32SC1);

for (int i = 0; i < img.rows; i++)

{

for (int j = 0; j < img.cols; j++)

{

for (int k = 0; k < nc; k++)

{

uchar val = nc == 1 ? img.at<uchar>(i,j) : img.at<Vec3b>(i,j)[k];

hist[k].at<int>(val) += 1;

}

}

}

for (int i = 0; i < nc; i++)

{

for (int j = 0; j < bins-1; j++)

hmax[i] = hist[i].at<int>(j) > hmax[i] ? hist[i].at<int>(j) : hmax[i];

}

const char\* wname[3] = { "blue", "green", "red" };

Scalar colors[3] = {

Scalar(255,0,0), Scalar(0,255,0), Scalar(0,0,255)

};

for (int i = 0; i < nc; i++)

{

canvas[i] = Mat::ones(125, bins, CV\_8UC3);

for (int j = 0, rows = canvas[i].rows; j < bins-1; j++)

{

line(

canvas[i],

Point(j, rows),

Point(j, rows - (hist[i].at<int>(j) \* rows/hmax[i])),

nc == 1 ? Scalar(200,200,200) : colors[i],

1, 8, 0

);

}

imshow(nc == 1 ? "value" : wname[i], canvas[i]);

}

}

#endif

**Eroding and dilating**

#include <opencv2/highgui/highgui.hpp>

#include <opencv2/imgproc/imgproc.hpp>

using namespace cv;

int main(){

Mat image= cv::imread("/Users/Zhansaya/Desktop/Unknown.jpg");

Mat erodedImage, dilatedImage;

erode(image,erodedImage,Mat());

dilate(image,dilatedImage,Mat());

namedWindow("Eroded Image");

namedWindow("Dilated Image");

imshow("Eroded Image",erodedImage);

imshow("Dilated Image",dilatedImage);

waitKey(0);

return 0;

}