视觉检测技术 课外项目报告
图像二值化处理（及二维傅里叶变换）

姓　名：史杰灵 ，　学　号：19121663

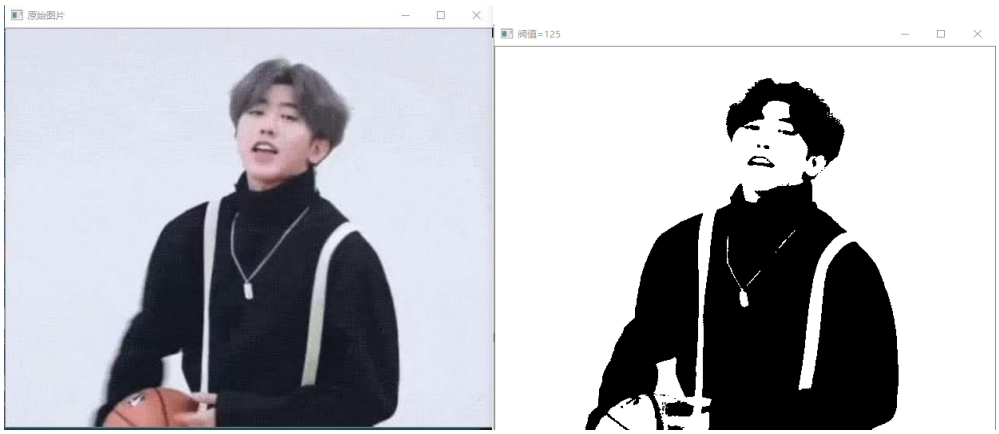
1 图像二值化处理

1.1 图像二值化算法

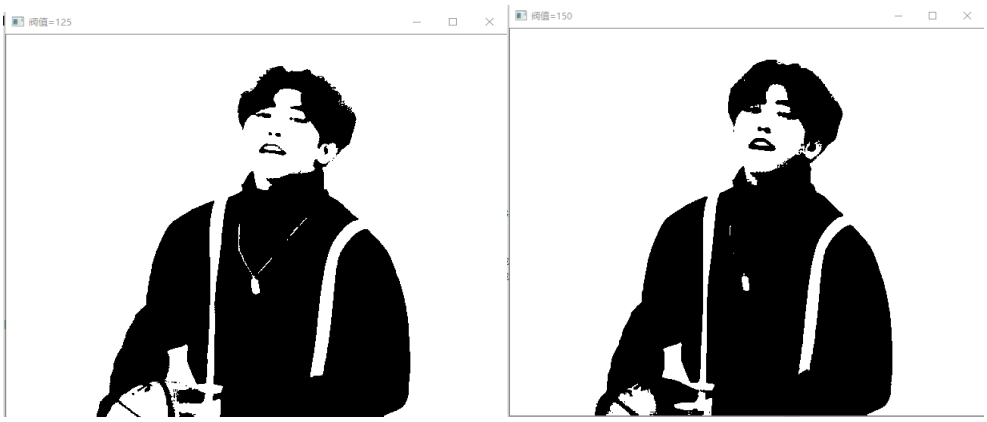
```
#include<opencv2/opencv.hpp>
#include<iostream>
#include<vector>
using namespace std;
using namespace cv;

int main()
{
    Mat img = imread("2.jpg");
    if (img.empty())
    {
        cout << "图片读取失败" << endl;
    }
    Mat gray;
    cvtColor(img, gray, COLOR_BGR2GRAY);  //将RGB图像img转为灰度图gray
    //灰度图BINARY二值化
    Mat binary_gray1, binary_gray2;
    threshold(gray, binary_gray1, 125, 255, THRESH_BINARY);
    threshold(gray, binary_gray2, 150, 255, THRESH_BINARY);
    imshow("原始图片", img);
    imshow("阀值=125", binary_gray1);
    imshow("阀值=150", binary_gray2);
    waitKey(0);
    return 0;
}
```

1.2 OPENCV环境下的图像二值化实现



1.3 比较不同阈值下二值化处理后图像的变化



有图片可以看出，阈值更大的图片黑色越多，白色越少，因为阈值更大，越多的部分位于阈值以下。

2 二维傅里叶变换

2.1二维傅里叶变换算法

```
  #include <iostream>
#include <opencv2\core\core.hpp>
#include <opencv2\highgui\highgui.hpp>
#include <opencv2\imgproc\imgproc.hpp>

using namespace std;
using namespace cv;

int main()
{
    //以灰度模式读取图像并进行显示
    Mat srcImage = imread("2.jpg", 0);
    if (!srcImage.data)
    {
        cout << "读入图像有误，请检查文件" << endl;
        return false;
    }
    imshow("原始图像", srcImage);

    int m = getOptimalDFTSize(srcImage.rows);
    int n = getOptimalDFTSize(srcImage.cols);
    Mat padded;
    copyMakeBorder(srcImage, padded, m - srcImage.rows, 0, n - srcImage.cols, 0,
BORDER_CONSTANT, Scalar::all(0));

    Mat planes[] = { Mat_<float>(padded), Mat::zeros(padded.size(), CV_32F) };
    Mat complexI;
    merge(planes, 2, complexI);

    dft(complexI, complexI);


    split(complexI, planes);  //将多通道数组分为两个单通道数组
                              //planes[0] = Re(DFT(I)),planes[1] = Im(DFT(I))
    magnitude(planes[0], planes[1], planes[0]);
    Mat MagnitudeImage = planes[0];

    MagnitudeImage += Scalar::all(1);
    log(MagnitudeImage, MagnitudeImage);    //求自然对数
```

```
    MagnitudeImage = MagnitudeImage(Rect(0, 0, MagnitudeImage.cols &-2,
MagnitudeImage.rows &-2));
    int cx = MagnitudeImage.cols / 2;
    int cy = MagnitudeImage.rows / 2;
    Mat q0(MagnitudeImage, Rect(0, 0, cx, cy));      //ROI区域的左上方
    Mat q1(MagnitudeImage, Rect(cx, 0, cx, cy));//ROI区域的右上方
    Mat q2(MagnitudeImage, Rect(0, cy, cx, cy));//ROI区域的左下方
    Mat q3(MagnitudeImage, Rect(cx, cy, cx, cy));    //ROI区域的右下方

    //交换象限，左上与右下交换
    Mat tmp;
    q0.copyTo(tmp);
    q3.copyTo(q0);
    tmp.copyTo(q3);
    //交换象限，右上与左下进行交换
    q1.copyTo(tmp);
    q2.copyTo(q1);
    tmp.copyTo(q2);

    //归一化，用0~1之间的浮点数将矩阵变换为可视图的格式
    normalize(MagnitudeImage, MagnitudeImage, 0, 1, CV_MINMAX);
    //显示效果图
    imshow("频谱幅值", MagnitudeImage);
    waitKey();
    return 0;
}
```

2.2 OPENCV环境下的图像二值化实现