**《图像信息处理》课程**

**实验报告**

**题目**

**专业**

**班级**

**学号**

**姓名**

**指导教师签名**

**华东理工大学信息学院计算机系**

**2019年03月27日**

1. **实验序号**

实验二

1. **实验题目**

图像的几何变换

1. **实验日期**

2019年3月27日

1. **实验内容（实验要求）**

1. 将实验图像分别实行以下操作： 1）等比例放大2倍； 2）缩小0.8倍，平移到点（88,88），绕图像中心逆时针旋转60°。

要求显示原图像和各种仿射变换后的图像。

2. 利用人机交互方式实现图像投影变换（参考教材第77页到80页的例子）。

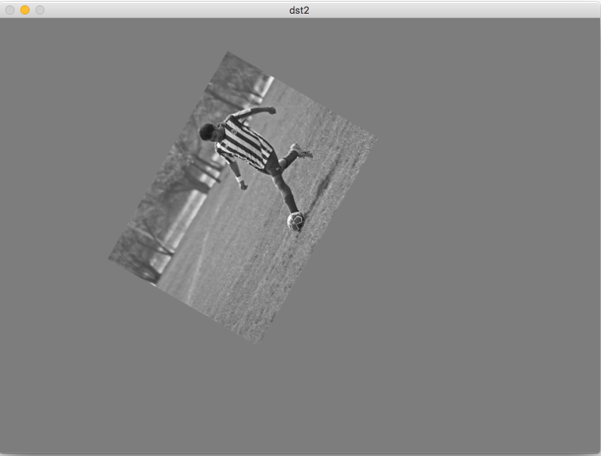
1. **实验结果（包含源程序、输出结果）**

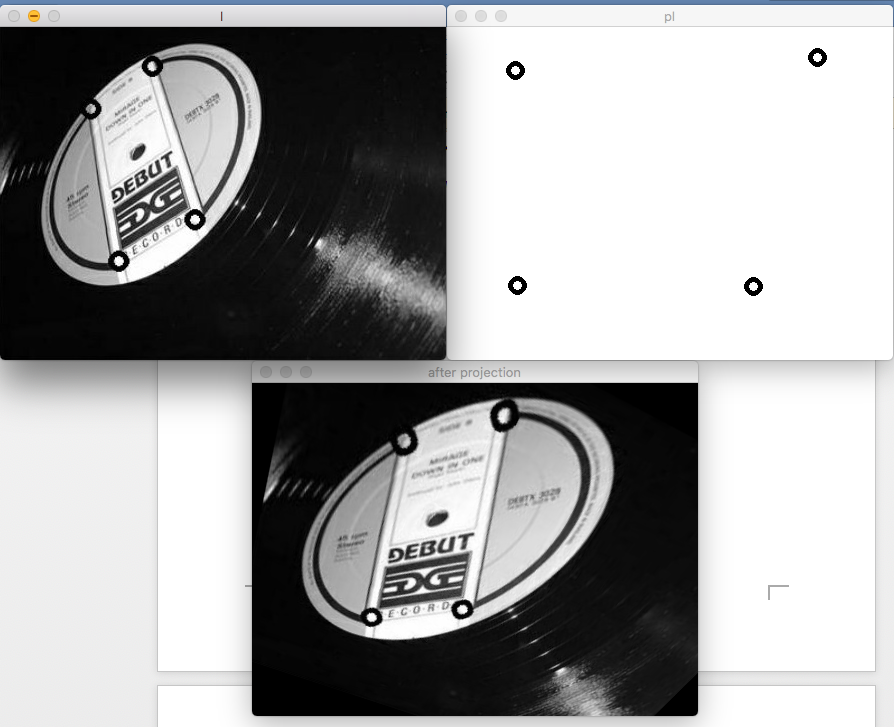
**main.cpp**  
**int** main(**int** argc, **const char** \* argv[]) {  
Mat I = imread(**"/Users/dddwj/ownCloud/mac/备份/学习/1笔记/201902学期/计算机图像处理/LearnOpenCV/exp2/img.jpg"**, ***CV\_LOAD\_IMAGE\_GRAYSCALE***); *// 按灰度图来读取，I.channels() = 1* **if**(!I.data){  
 cout << **"Can't open image!"** << endl;  
 **return** -1;  
 }  
 imshow(**"raw"**, I);  
  
 *// 仿射变换：[法一 warpAffine函数] 等比例放大2倍* Mat dst1\_1;  
 Mat dst1\_M = (Mat\_<**float**>(2,3) << 2,0,0, 0,2,0);  
 warpAffine(I, dst1\_1, dst1\_M, Size(I.cols\*2, I.rows\*2)); *// Size()内，先'宽' 后'高'  
  
 // 仿射变换：[法二 rezise函数] 等比例放大2倍* Mat dst1\_2;  
 resize(I, dst1\_2, Size(I.cols\*2, I.rows\*2), 2, 2);  
  
 *// 仿射变换：缩小0.8倍，平移到点(180,400)，绕图像中心逆时针旋转60度* Mat dst2;  
 **double** dst2\_M\_rotate\_angle = 60./180 \* **M\_PI**;  
 Mat dst2\_M\_scale = (Mat\_<**double**>(3,3) << 0.8,0,0,  
 0,0.8,0,  
 0,0,1);  
 Mat dst2\_M\_trans = (Mat\_<**double**>(3,3) << 1,0,180,  
 0,1,400,  
 0,0,1);  
 Mat dst2\_M\_rotate = (Mat\_<**double**>(3,3) << cos(dst2\_M\_rotate\_angle), sin(dst2\_M\_rotate\_angle), 0,  
 -sin(dst2\_M\_rotate\_angle), cos(dst2\_M\_rotate\_angle), 0,  
 0,0,1);  
 Mat dst2\_M = dst2\_M\_scale \* dst2\_M\_trans \* dst2\_M\_rotate; *// 使用OpenCV重载的\*运算符，实现点乘* dst2\_M = dst2\_M.colRange(0,3).rowRange(0,2); *// 取2\*3的仿射变换矩阵M* warpAffine(I, dst2, dst2\_M, Size(I.cols\*2, I.rows\*2), ***INTER\_LINEAR***, ***BORDER\_CONSTANT***, 125);  
imshow(**"dst2"**, dst2);  
 waitKey(0);  
 **return** 0;  
}

**Projection.cpp**

#include **<iostream>**#include **<opencv2/opencv.hpp>**#include **<opencv2/highgui.hpp>**#include **<opencv2/imgproc.hpp>  
  
using namespace** std;  
**using namespace** cv;  
  
Mat pI; *// 投影变换后的图*Mat I; *// 原图*Point2f IPoint, pIPoint;  
**int** i = 0, j = 0;  
Point2f src[4]; *// 存储原坐标*Point2f dst[4]; *// 存储对应变换的坐标  
  
// 通过以下鼠标事件，在原图中取四个坐标***void** mouse\_I(**int** event, **int** x, **int** y, **int** flags, **void** \*param){  
 **switch** (event){  
 **case *CV\_EVENT\_LBUTTONDOWN***:  
 IPoint = Point2f(x, y);  
 **break**;  
 **case *CV\_EVENT\_LBUTTONUP***:  
 src[i] = IPoint;  
 circle(I, src[i], 7, Scalar(0), 3); *// 标记* cout << **"i:"** << i ;  
 i += 1;  
 **break**;  
 **default**:  
 **break**;  
 }  
}  
  
*// 通过以下鼠标事件，在要输出的画布上取四个对应的坐标***void** mouse\_pI(**int** event, **int** x, **int** y, **int** flags, **void** \*param){  
 **switch** (event){  
 **case *CV\_EVENT\_LBUTTONDOWN***:  
 pIPoint = Point2f(x, y);  
 **break**;  
 **case *CV\_EVENT\_LBUTTONUP***:  
 dst[j] = pIPoint;  
 circle(pI, dst[j], 7, Scalar(0), 3); *// 标记* cout << **"j:"** << j;  
 j += 1;  
 **break**;  
 **default**:  
 **break**;  
 }  
}  
  
**int** main(**int** argc, **const char** \* argv[]) {  
 *// 输出图像* pI = 255 \* Mat::ones(I.size(), **CV\_8UC1**);  
 *// 在原图窗口上，定义鼠标事件* namedWindow(**"I"**, ***WINDOW\_AUTOSIZE***);  
 setMouseCallback(**"I"**, mouse\_I, **NULL**);  
  
 *// 在输出窗口上，定义鼠标事件* namedWindow(**"pI"**, ***WINDOW\_AUTOSIZE***);  
 setMouseCallback(**"pI"**, mouse\_pI, **NULL**);  
  
 *// 等待用户点击* imshow(**"I"**, I);  
 imshow(**"pI"**, pI);  
 **while**(!(i > 3 && j > 3)){  
 imshow(**"I"**, I);  
 imshow(**"pI"**, pI);  
 **if** (waitKey(50) == **'q'**)  
 **break**;  
 }  
 imshow(**"I"**, I);  
 imshow(**"pI"**, pI);  
  
 *// 移除鼠标事件* setMouseCallback(**"I"**, **NULL**, **NULL**);  
 setMouseCallback(**"pI"**, **NULL**, **NULL**);  
  
 *// 计算投影变换矩阵* Mat p = getPerspectiveTransform(src, dst);  
 *// 投影变换* Mat result;  
 warpPerspective(I, result, p, pI.size());  
 imshow(**"after projection"**, result);  
 waitKey(0);  
 **return** 0;  
}

**输出结果：**

****

****