

Ex5：A4纸矫正

14331098 黄建武 计应

- 实验环境
 - Linux Deepin 15.3
 - g++ 6.2.0
 - C++11
- 参考资料
 - A4纸边缘提取
 - 透视变换 Perspective Transformation
- 实验要求

输入图像是普通A4打印纸,上面可能有手写笔记或者打印内容,但是拍照时可能角度不正。输出图像是已经矫正好的标准普通A4纸,并裁掉无用的其他内容,只保留完整A4纸张。

- 实验方法
 - 对输入图像进行灰度图转换, 并进行高斯模糊。
 - 计算灰度图的梯度。
 - 进行Hough变换, 求出霍夫空间的峰值点。
 - 找出投票数最大的4个点即A4纸的4个角点。
 - 对A4纸进行Warping转为标准普通A4纸并裁剪。
- 实验实现

在Ex4的基础上求出A4纸4个角点的坐标后, 已经能够确定源图像中A4纸的位置, 接着需要进行矫正, 该实验中采用的是Projective Transform (透视变换), 透视变换是将图片投影到一个新的视平面, 透视变换涉及到一个 3×3 的变换矩阵, 具体推导过程可参见[透视变换 Perspective Transformation](#), 在进行透视变换的时候, 要注意将A4纸的4个角点进行顺时针标记序号, 由于前面获取到的角点只有坐标没有方位信息, 因此需要通过坐标求出四个角点分别对应的方位。

```
// 计算两点间的距离
inline float dis(float x, float y){return sqrt(x * x + y * y);}

// 按角点到源图像左上角的距离排序
bool Image::cornerCmp(const pair<double, double> a, const pair<double, double> b) {
    return dis(a.first, a.second) < dis(b.first, b.second);
}

// 对4个角点顺时针排序
void Image::orderCorners() {
    sort(corner.begin(), corner.end(), cornerCmp);
    double width = dis(corner[0].first - corner[1].first,
                        corner[0].second - corner[1].second);
    double length = dis(corner[0].first - corner[2].first,
                        corner[0].second - corner[2].second);

    // 源图像中的A4纸是横向的
```

```

if (corner[1].first < corner[2].first && width < length) {
    swap(corner[1], corner[2]);
    swap(corner[2], corner[3]);
    corner.push_back(corner[0]);
    corner.erase(corner.begin());
}
// 源图像中的A4纸是竖向的
else {
    swap(corner[2], corner[3]);
}
}

```

成功确定角点的方位后，按照推导出来的结果，计算出透视变换的变换矩阵，然后将变换矩阵分别映射到每个像素点，因为图像并不大，使用插值后效果并不明显，我选择直接将坐标转换为整数，即取临近的像素点替代，减少运算量，提高运行速度。

```

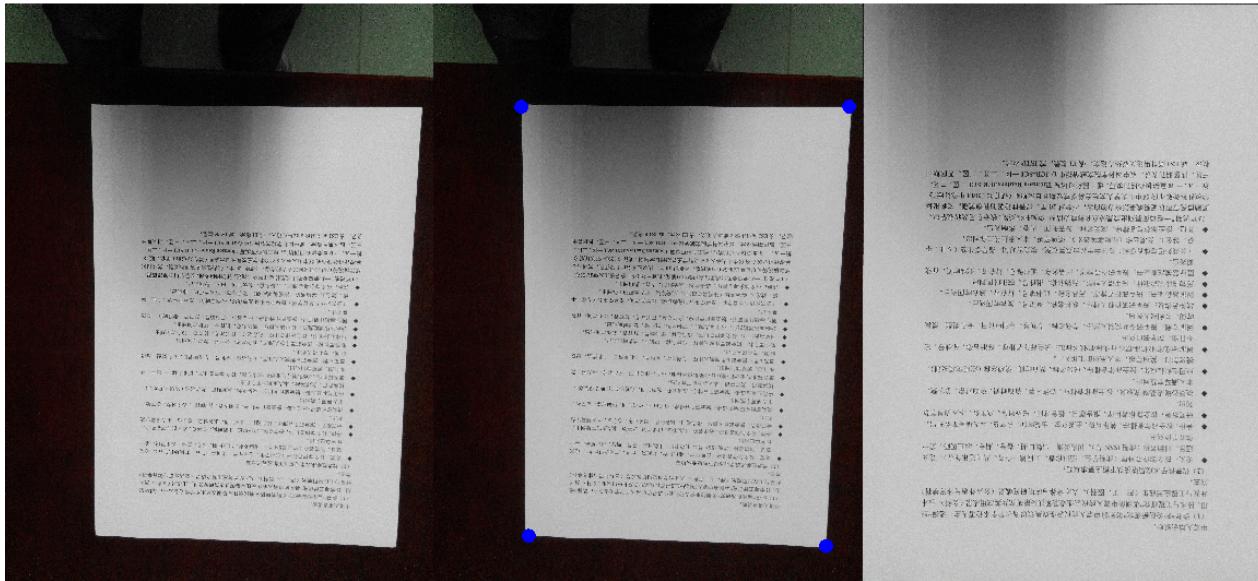
cimg_forXY(result, x, y) {
    double _x = x / width, _y = y / height;
    double denominator = M[2] * _x + M[5] * _y + M[8];
    double tx = (M[0] * _x + M[3] * _y + M[6]) / denominator;
    double ty = (M[1] * _x + M[4] * _y + M[7]) / denominator;
    cimg_forC(result, c) {
        result(x, y, c) = source((int)tx, (int)ty, c);
    }
}

```

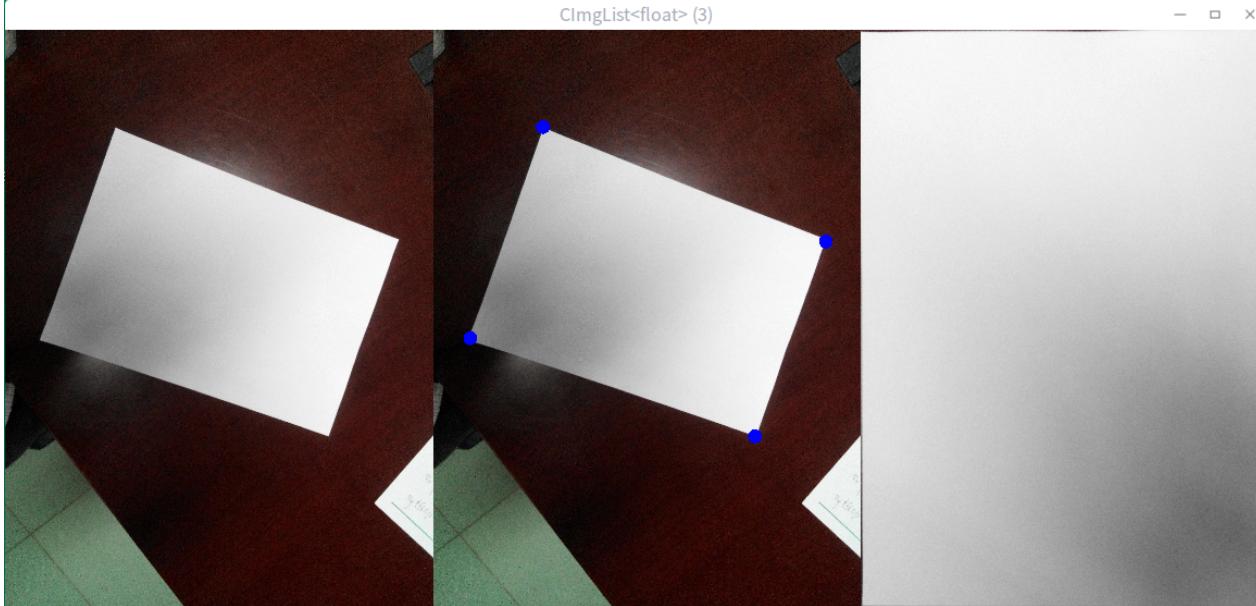
完整实现请看源代码。

- 实验结果

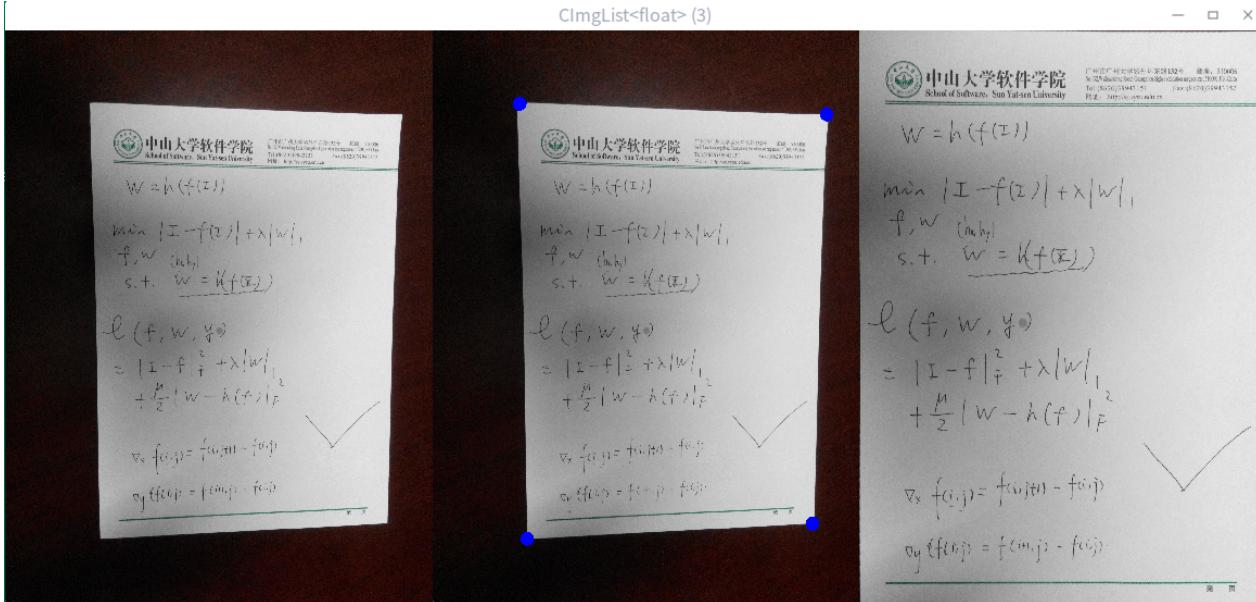
CImgList<float> (3)

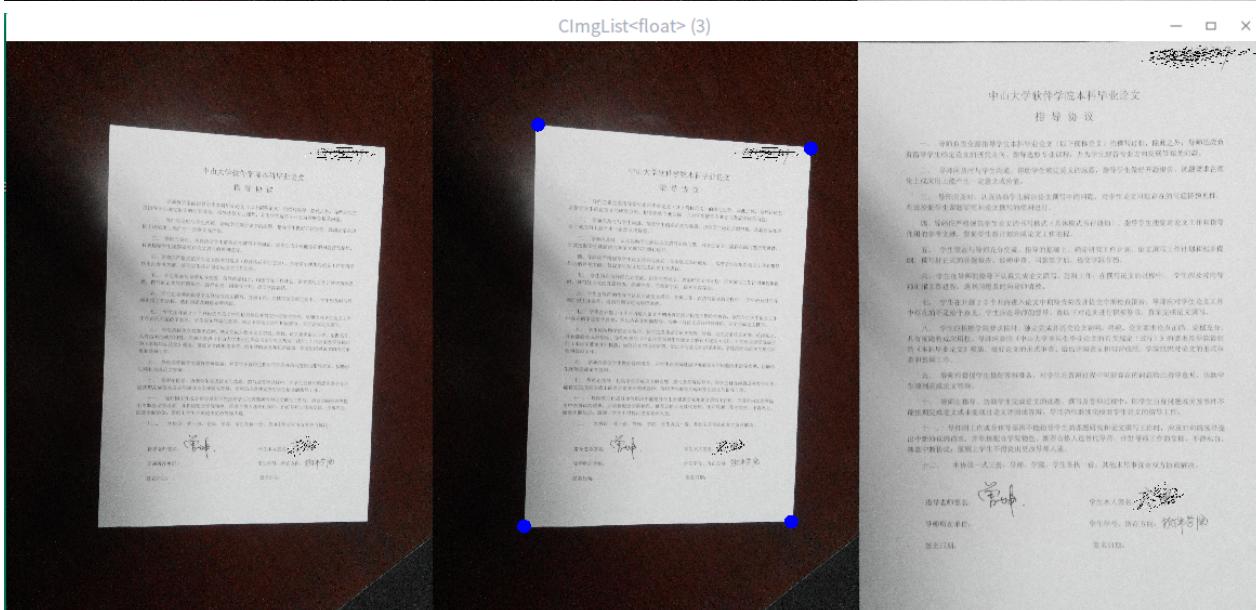
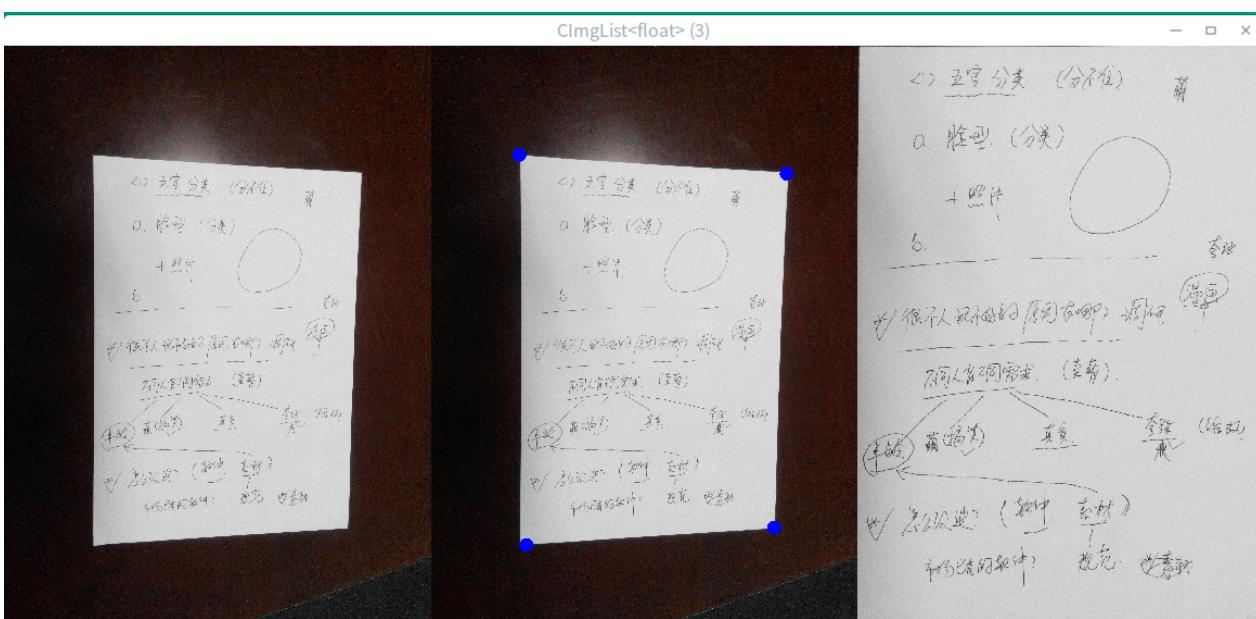
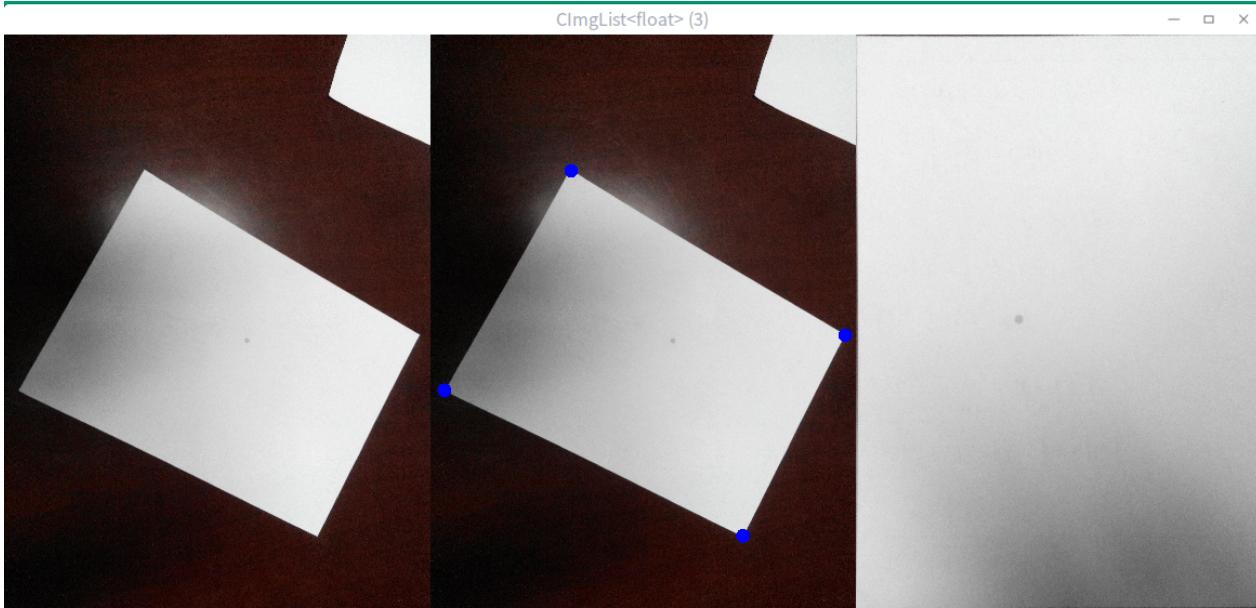


CImgList<float> (3)

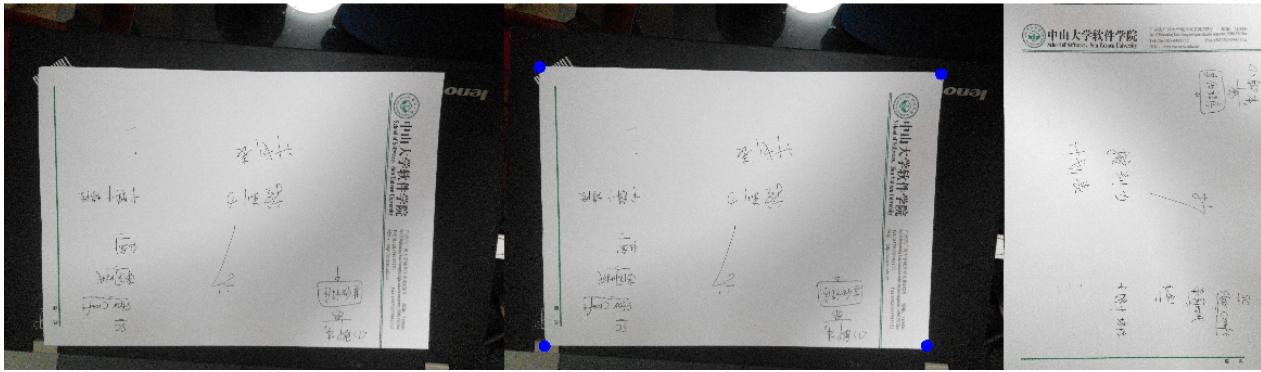


CImgList<float> (3)

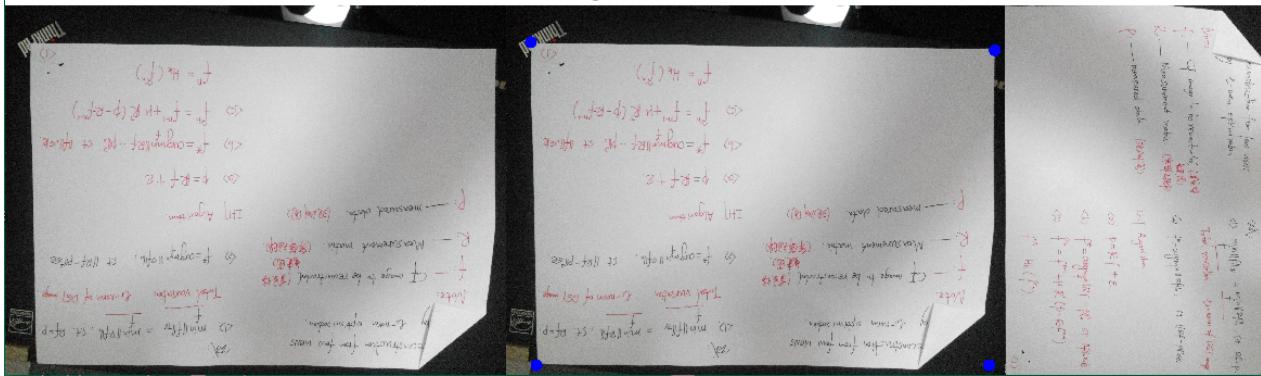




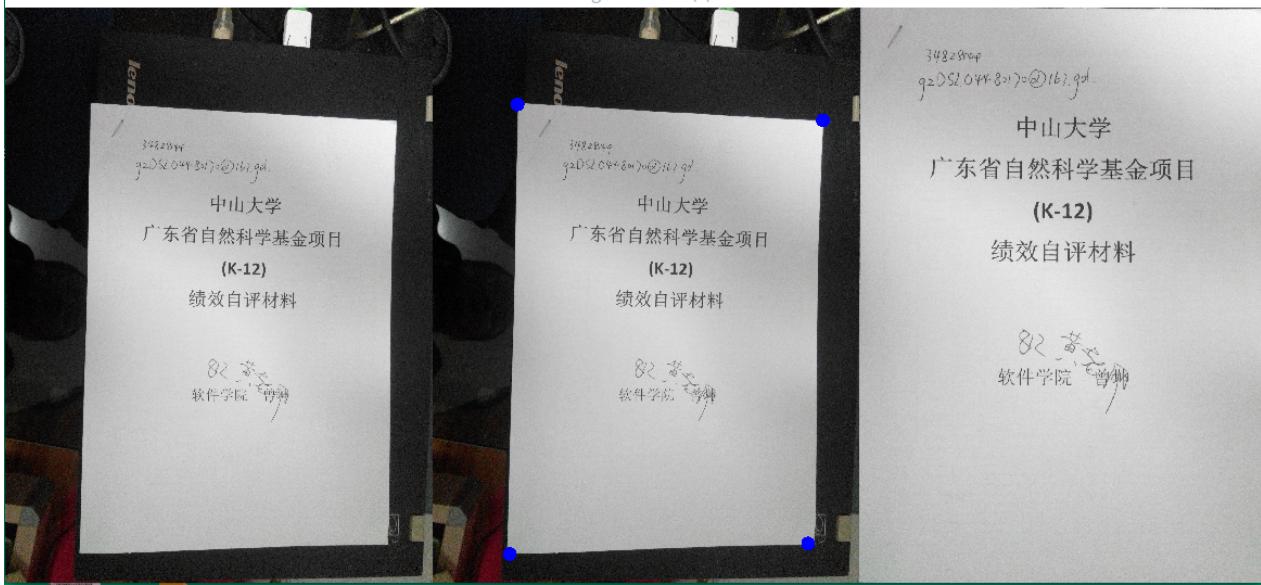
CImgList<float> (3)



CImgList<float> (3)



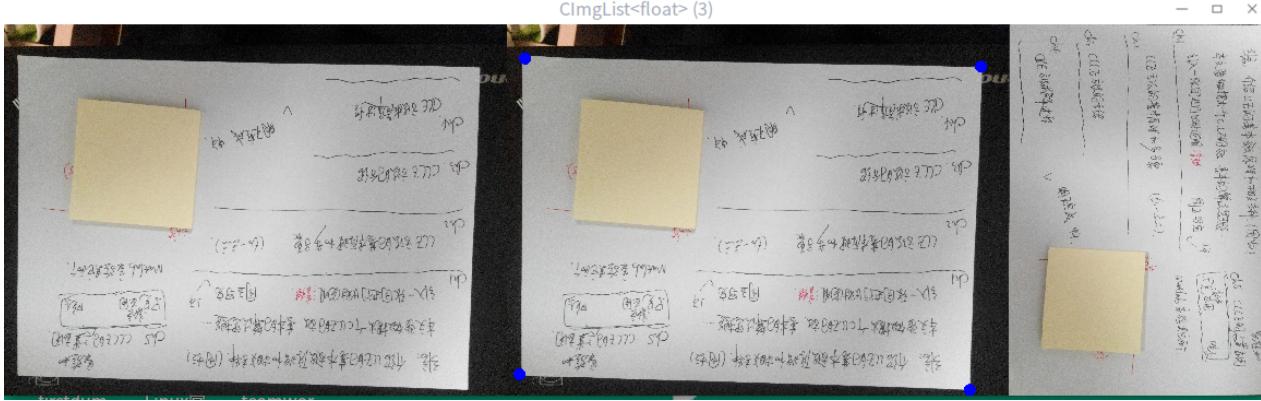
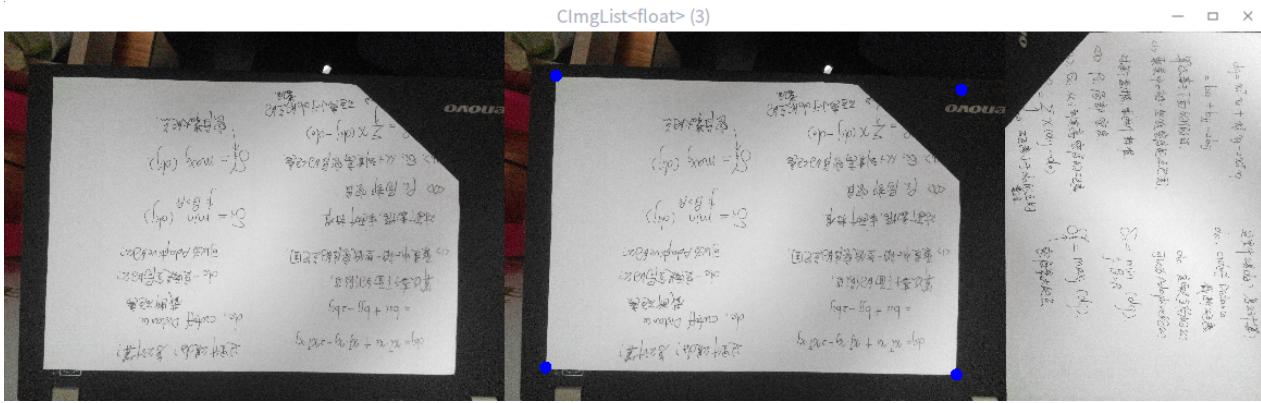
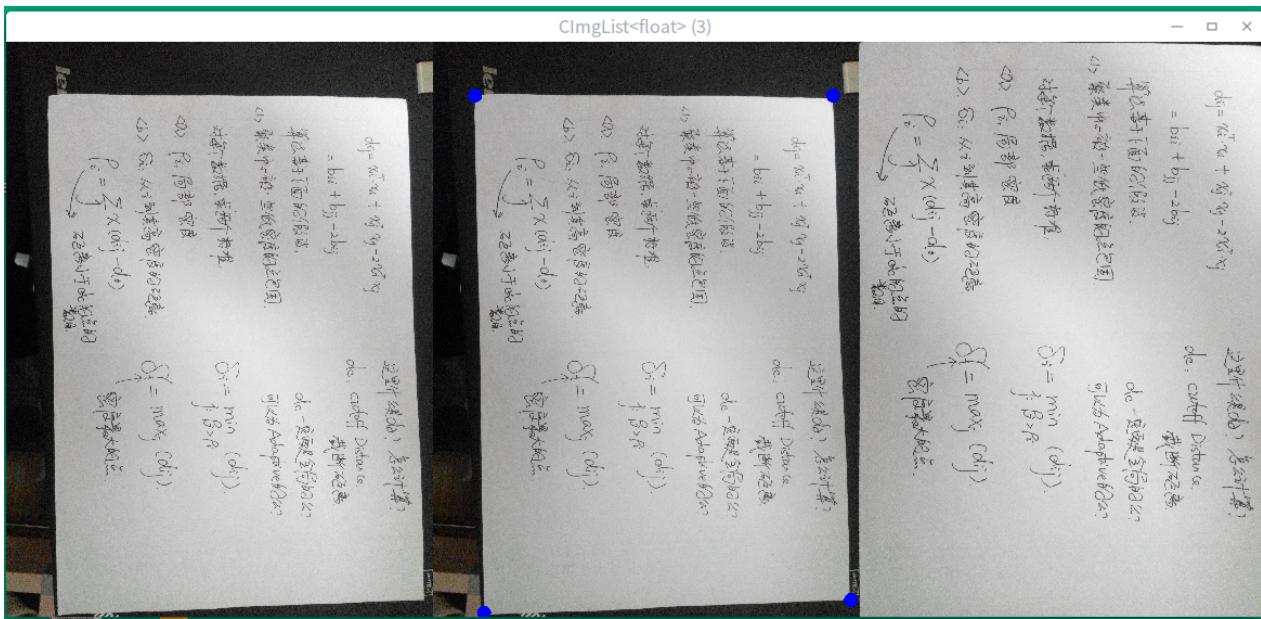
CImgList<float> (3)

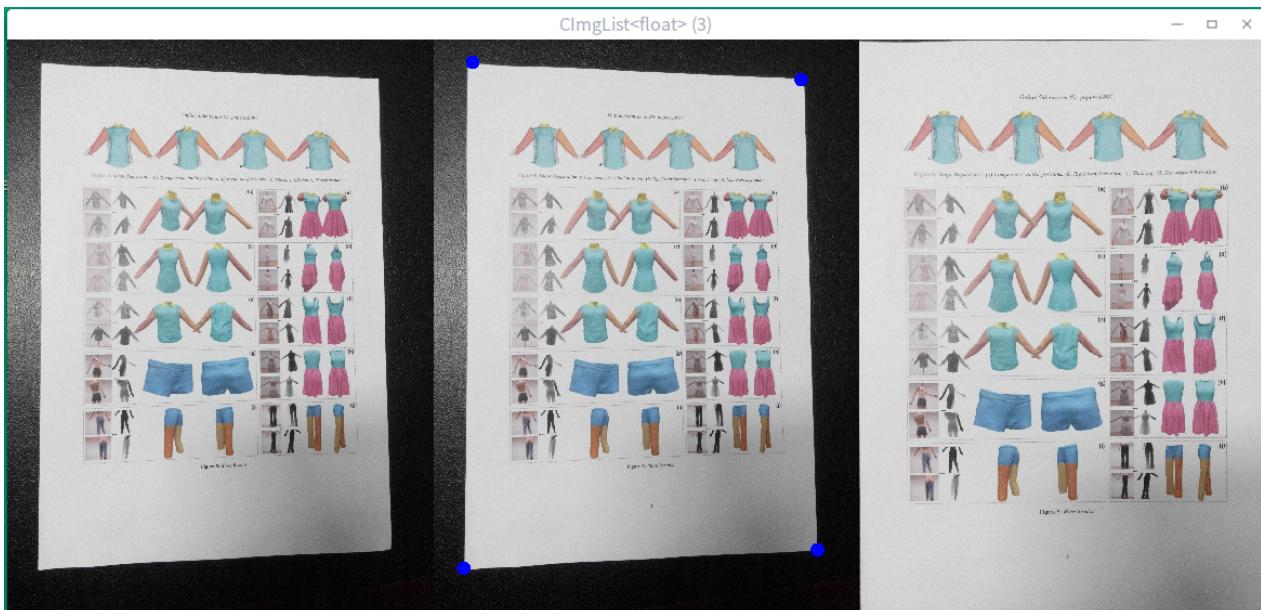
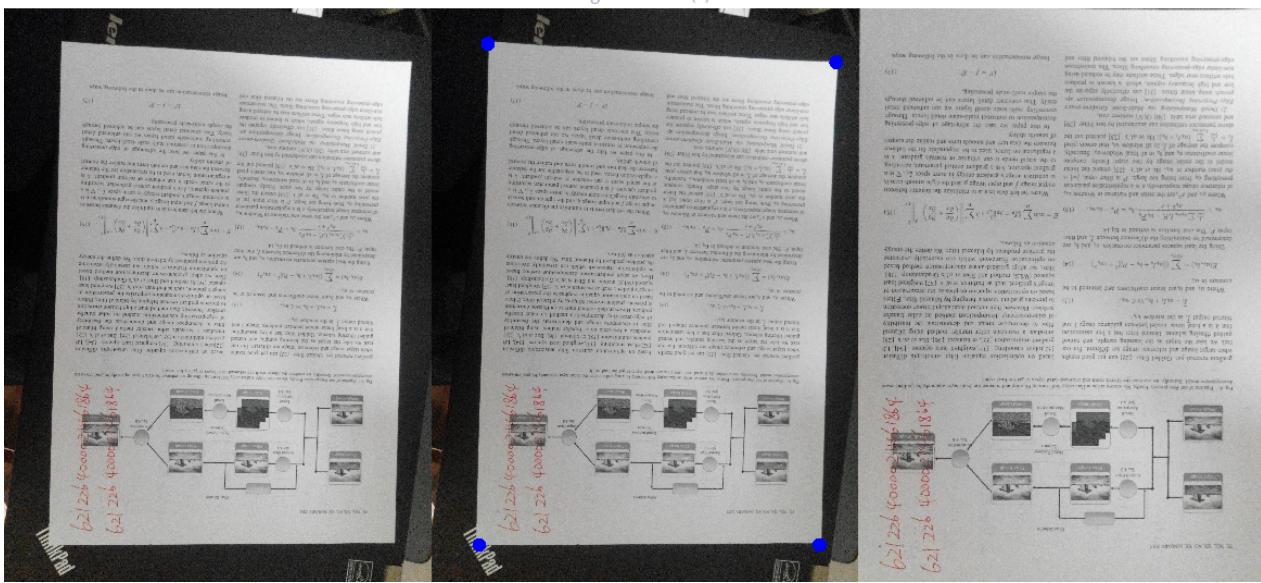


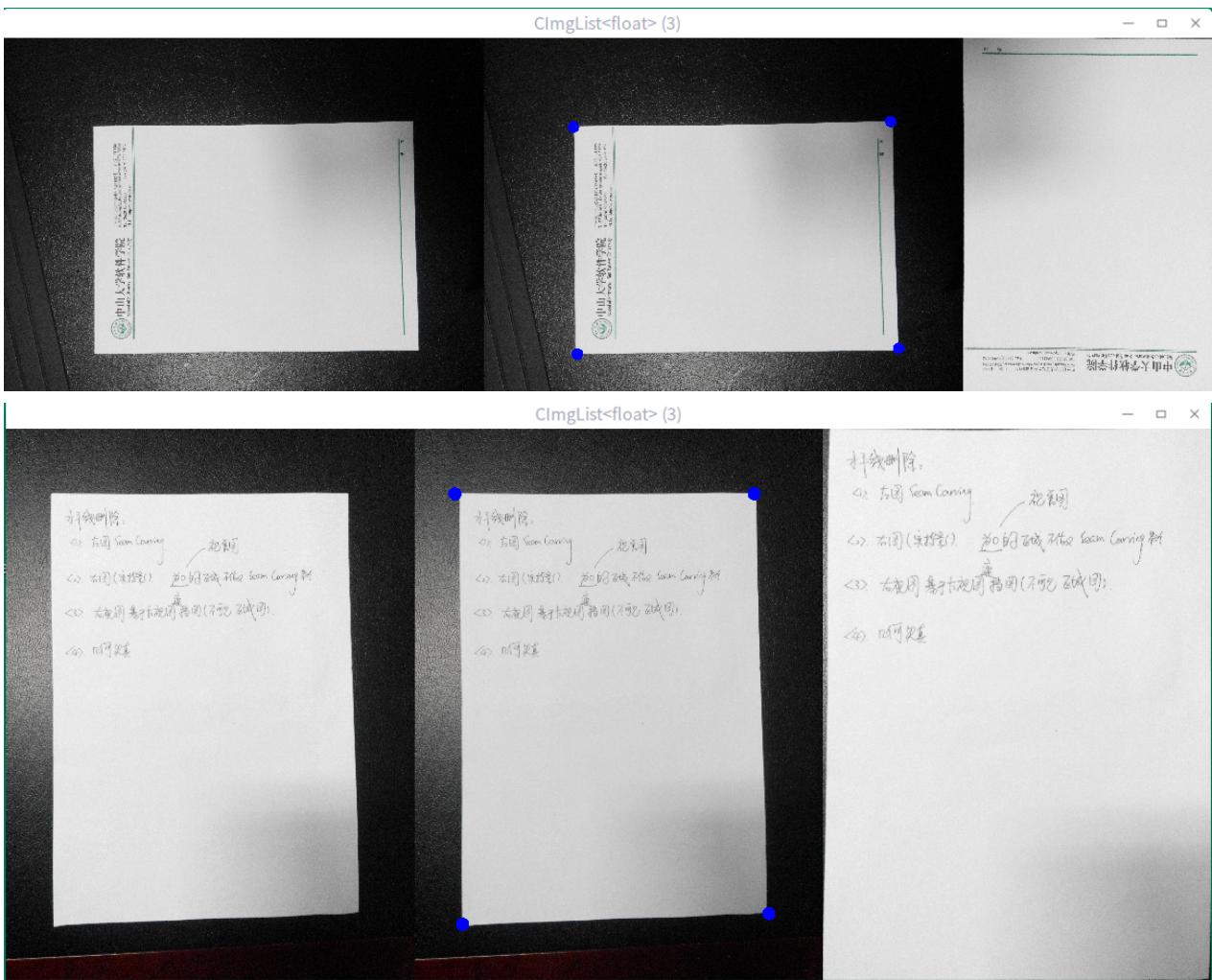
3422049
3422048

中山大学
广东省自然科学基金项目
(K-12)
绩效自评材料

82
软件学院
曾静



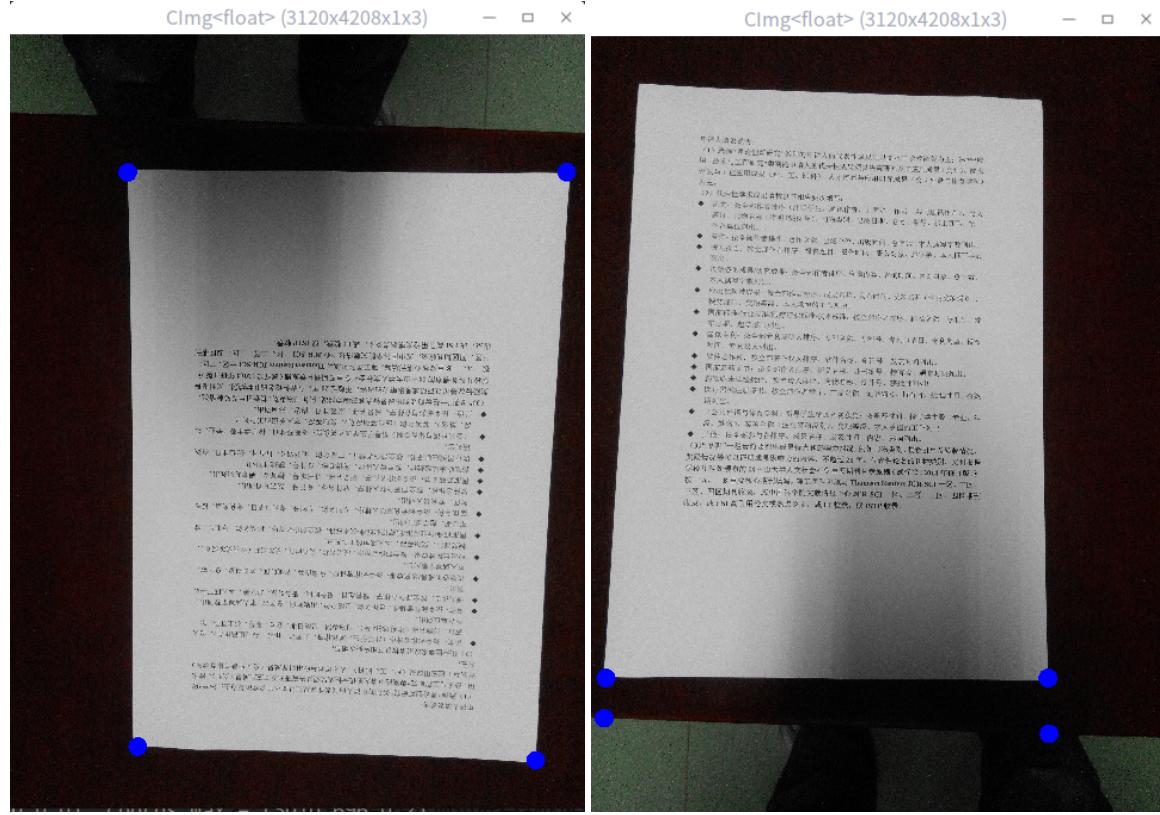




实验问题

- 实验中参数的设置遇到问题，无法使用固定的参数完成对所有图像的矫正，其中有两三张图片需要单独设置不同的参数，才能检测出4个角点并进行矫正。
- 纠正1.jpg时遇到一个很奇怪的问题，设置参数后只检测出A4纸底边的两个角点，另外两个角点检测成桌子的边缘，调了很多参数都不行，在查看源图像时不小心旋转了源图像，结果成功检测出A4纸4个角点，同一

份代码，相同的参数，因为输入图像旋转了180度就有不同的结果，想不懂原因。



实验思考

如何在保证精度的结果情况下加快运行速度。

- 缩小图片尺寸进行矫正后再进行放大，减少循环次数；
- 使用临近的点的像素值取代插值，减少运算次数