

# 基于 GUI 的 PPT 投影变换开发报告

姓名：程科

专业班级：光电 1307 班

学号：U201314221

## 实验目的

大学生经常用手机拍摄 PPT，但拍摄的 PPT 由于角度问题，大多不是矩形，也没有占满整个屏幕。通过投影变换，可以将倾斜和形变的 PPT 变换为矩形，方便了学生对 PPT 的阅读。

## 实验亮点

1. 利用图像校准技术，无需输入变换矩阵，而是通过在图像中采集特征点进行配准。
2. 利用 PPT 亮度高、面积大、边缘有黑框等特点，程序自动完成特征点的采集，实现了整个流程的自动化。
3. 对程序进行了大量检验，表明自动变换的正确率达到 80%以上。
4. 由于程序具有一定应用价值，我为其编写了 GUI 界面。

## 流程图



# 实验原理

整个程序分为 PPT 定位和 PPT 矫正两部分。

## PPT 定位算法原理

- 首先分析 PPT 的特征，提出一些假设：
  - a) 一般 PPT 为白色背景或者其他较亮的背景，黑色背景的 PPT 在教学中很少使用，因此我们程序将 PPT 默认为较亮的区域。
  - b) PPT 的投影屏有黑色边框，这使得 PPT 的区域是有黑色边框的。
  - c) 如果用户以拍摄 PPT 为目的，则一般 PPT 会占据图像中较大的面积。（这是考虑到环境中也有较亮的部分（比如灯光），但这些较亮的部分一般面积较小）
  - d) 如果用户以拍摄 PPT 为目的，则一般 PPT 会在图像中心。
- 根据这样的假设，我们构建如下算法：
  1. 读入图像，将其转换为灰度图。
  2. 利用最大类间差法确定图像中合适的阈值，将图像转换为二值图 BW，PPT 应属于二值图中白色区域。（这里利用了假设 a,b）
  3. 利用 matlab 自带函数求出二值图 BW 中最大的连通域，记为 ConnectMAX。（这利用了假设 c）
  4. 根据假设 d，判断图像中心的点是否包含在了 ConnectMAX 之内，若是，则进行第 6 步；若不是则进行第 5 步。
  5. ConnectMAX 不包含图像中心，说明环境中有些区域又亮又大（比如墙壁），但这些区域一般不在图像中心，应将这种不在图像中心的 ConnectMAX 舍去，采用减法： $BW = BW - \text{ConnectMAX}$ 。这样就将不在图像中心的又大又亮的区域减掉了，然后将新得到的二值图 BW 返回到第 3 步。
  6. 到这一步，ConnectMAX 就是 PPT 的区域了。接下来，我们求出 ConnectMAX 中的坐标点的  $x+y$  的最大值，也就得到了 PPT 右下角的坐标。同理可得 PPT 其他三个角的坐标，从而确定 PPT 四点坐标。
  7. PPT 定位算法完成。

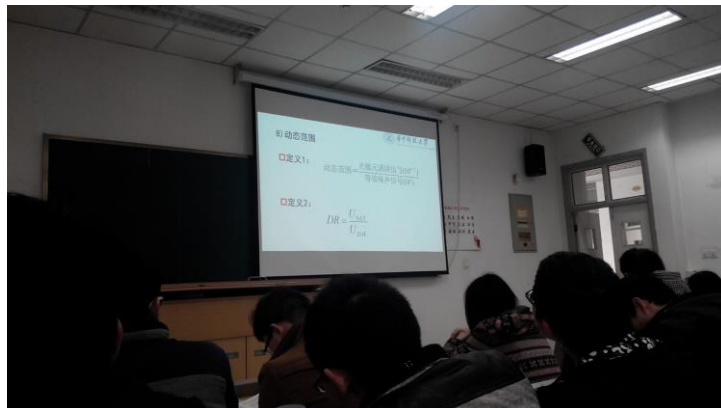
## PPT 矫正算法

教学 PPT 横竖比例一般为 4:3。因此将得到的 PPT 四个角的坐标与程序设定的 4:3 的矩形进行图像配准，直接调用 matlab 配准函数进行投影变换。

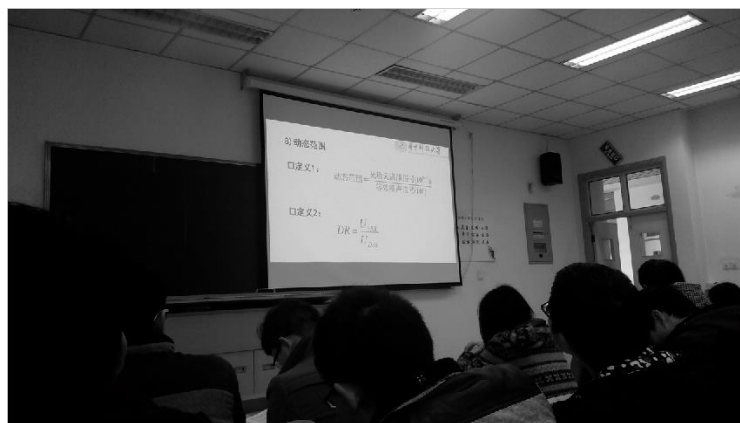
注意，这里变换的时候是拿原彩色图变换，而不是灰度图或者二值图。

## 例子

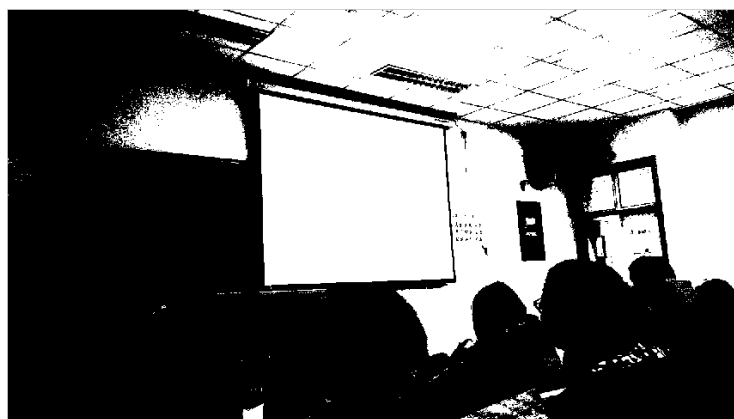
为了方便理解，我们以一副图像的矫正过程为例：  
读入图像 19.jpg 如下



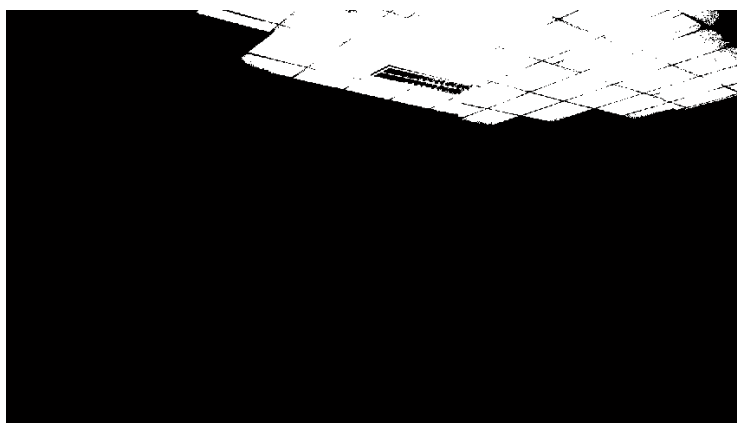
灰度化后如下



二值化后如下：

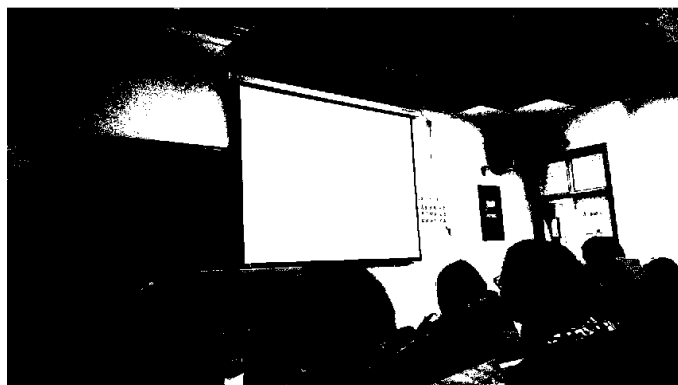


找到其中最大的连通域如下：

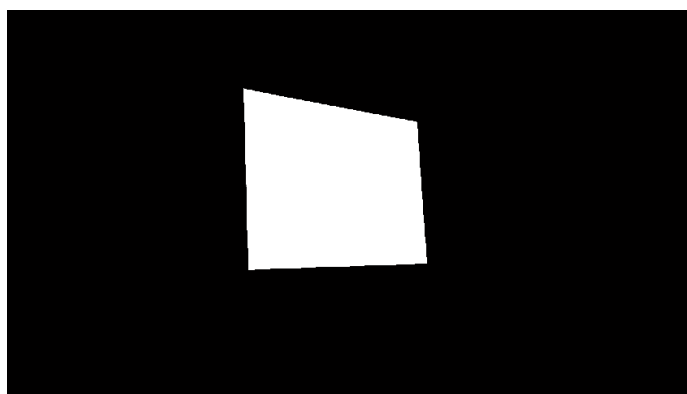


判断此连通域是否包含了图像的中心点（实际做的时候是取图像中心 100 个点，只要连通域包含其中一个点，就认为连通域在图像中心）。

经过判断，上图不在图像中心，因此用二值化图像减去最大连通域，得到新的待处理二值图如下。

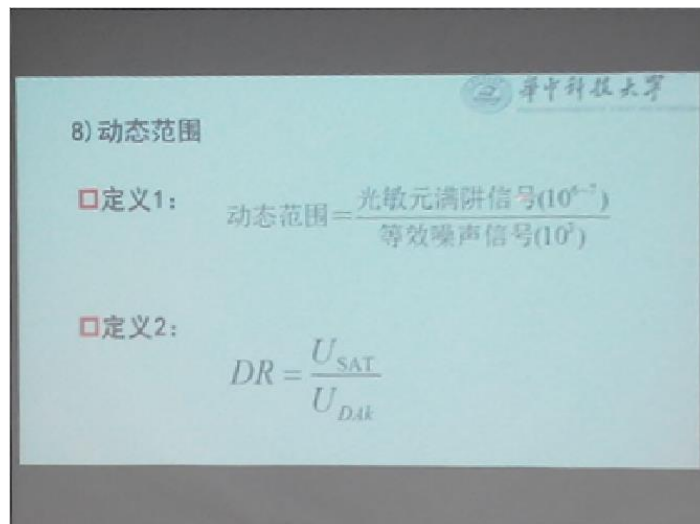


求此二值图中最大连通域，如下



用上文方法判断此区域是否在图像中心，经过判断，此区域在图像中心。于是将其认定为 PPT。

用上文算法求出其四角坐标。调用图像配准程序，得到转换后图像如下：



## GUI 设计

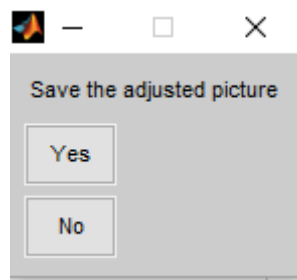
由于本程序有一定使用价值，因此为其制作 GUI 界面方便使用（运行程序文件夹中“PPT\_PHOTO\_GUI.m 即可”）：



其中设置两个可调选项：

1. 加速选项：在 PPT 定位过程中，对图像清晰度不关心，可以对图像进行压缩，界面中的压缩比从 1:1 调节到 20:1.
2. 微调类间差法所求得的阈值，可对一些阈值不佳的特殊情况使用。

求出图像后，弹出窗口询问用户是否保存：



为了方便演示，程序文件夹中附带了 20 个 PPT 照片，均为我在大学课堂中拍摄。事实证明，矫正效果较好，除 14.jpg 外，其他图片均矫正成功，成功率达 95%。（在合理调整阈值后 14.jpg 也可成功矫正）

## 展望

1. 可考虑采用点运算等方法改善最后输出图像的对比度。
2. 可将算法采用 JAVA 实现，开发 APP 方便手机客户使用。