学习过照片的梯形修正成矩形这节课以后，我对这个题材非常的感兴趣，首先感觉这个技术十分的使用，而且我自己喜欢到处走，拍些喜欢的照片，结果就是有时候高度不够，角度不对，拍出来的东西，比如窗户，就是梯形的，没有我想要的那种效果，于是我在学过这节课以后感觉这个创意十分的有用，贴近生活，而且我自己也能用到，于是想实现这个东西。

在看过p06的题目要求以后，现在首先来研究老师发的p06的代码。

经过研究后发现，我想彻底把它放到android代码里，不太想去写网页。在activity里做，感觉方便更实现我的需求。（有一部分原因是因为这个电脑上没有写html5和js趁手的兵器）。开始去研究那个网页是如何实现这些功能的。

在了解了网页的功能实现后，我发现了本次课设最大的问题：如何将不规则四边形规则成一个矩形？

在网页中，是一个梯形，于是可以只需要水平方向等比例拉伸即可，而像素点也是选取了周围的几个作为参考，可问题来了，如果是换成一个不规则四边形，该按什么比例拉伸呢，像素点又如何对应呢？这个问题我百思不得其解，想去网上寻找线索。

首先找到了很多OpenCV实现需求的例子，通过了解，我发现了一个路子：

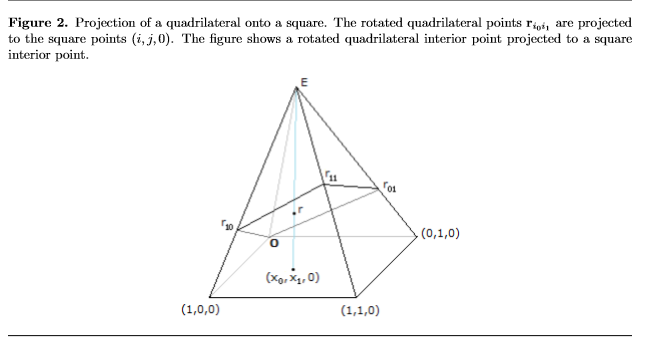
1.霍夫圆检测

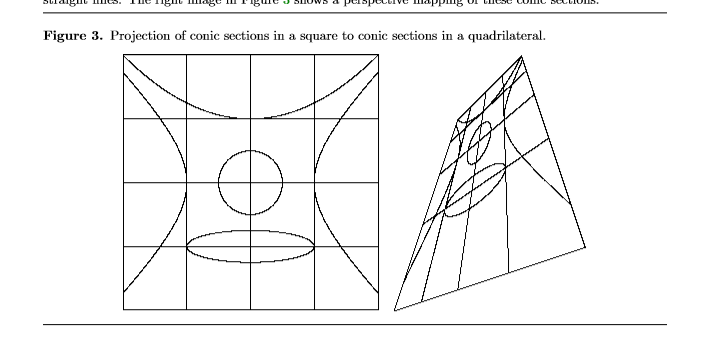
2.圆心所形成的外包矩形

3.利用四个圆心与矩形四角坐标进行校正

其实我是不太想用这个办法的，首先，在android studio上在装上OpenCV，然后学习OpenCV这门语言，是稍微有点复杂的，打算再学习学习，这个方法作为最后的方法。

功夫不负有心人，在我把视线转移到数学相关的时候，终于发现了数学解法：Perspective Mappings。





这是一篇论文，通过线性代数，利用中心投影的办法，巧妙的证明了任何四边形可以投影成矩形，并且给出了计算方法，我开始连夜研究这篇论文。

太困了就睡了，今天开始码代码，首先遇到的问题就是调用相机和读取本地图片的问题，从 Android 7.0 开始，Android SDK 中的 StrictMode 策略禁止开发人员在应用外部公开 file:// URI，我通过在网上的搜索，找到了一个大神写的图片选取的库：<https://github.com/GitLqr/LQRNativePicSelect>。

学习这个库。

在了解了如何使用后，在获得了权限以后实现了拍照和选取照片的功能。

下一步是选择不规则四边形，首先考虑到利用计算机图形学，卷积神经去进行识别要矫正的区域，简单思考后确认实现不了，于是想让用户手动改点击四边形的四个顶点，将围起来的图形截图截出来，然后再去对这个截图解出来的图操作。

找了半天文档，也没发现onclick()里怎么得到点击的x坐标和y坐标，转换思路，发现ontouch里面有motionevent 这个参数，里面有getx()和gety()这两个函数得到点击的位置，哇，舒服了，开始写点击选取四边形的功能。

又到了半夜了，总的来说还没有实现这个功能，解决的问题有使用canvas画图，可以根据点击的四个顶点画出任意四边形，并且给用户看到，目前遇到的问题是canvas使用的坐标系貌似和点击事件获得的getX(),getY()里面的坐标不太一样，在图片上圈出的四边形虽然是四边形，形状也和用户点击的一样，但是位置不一样，覆盖不上，点的是一个位置，画出来的是另一个位置，形状是对的，但是位置不对。继续研究。

我吐了，canvas(bitmap)里的bitmap大小是980\*1280的，我在x=980 y=1280画一圆圈，圆圈不是在右下角，而是在图片的一个地方，而且还是到头了，再加x和y，圆圈也不动了，再去查查是因为什么。

发现问题，因为getx()和gety()得到的是错误的。再研究研究。

我吐了，图片的位置和像素px单位不一样，弄到凌晨1点才弄好，我最后还是采用了最简单的数学办法：比例代换。先求出图片右边到父亲左边的距离 getRight()当作宽，图片底部的距离到父亲顶端的距离getBottom()当作高，和图片的width()和height()求比例系数，这样，每次点击一个点，求出这个点对应的像素的位置，在那个位置画图。Nice。

我的天！！！！！！！！！！！

我终于实现这个功能了！！！！！！！太舒服了。

现在是4月29日傍晚。说一下后来遇到的问题和解决的办法吧。

求出点击的像素点以后，我就知道了点击的四边形的四个顶点的像素坐标。根据前面从论文中学到的Perspective Mappings，知道了四个顶点的坐标，知道了要生成的图片的宽高，根据投影的原理，利用矩阵可以求出一些需要的系数，根据这些系数可以求出四边形中各个点对应在生成的图片中的位置。这里遇到了一些问题，主要是数学上的问题，对坐标的运算刚开始出了一点问题，通过不断的Log坐标，比对自己记录的之前的点击的地方，修改了刚开始写的有点错误的公式，解决了这个问题。自此，大部分问题就解决了。再就是如何填充新图了，我才用的是比较简单的最近邻域插值的办法，相对比较容易实现一点。最终实现了这个功能。剩下的老照片，负片等模式，就只需要按位改改颜色就可以了。相对来说没有什么难度了。

心得：

通过这次的项目，我有了很大的收获：

1. 确实这个功能是我想要的，感觉非常实用。
2. 知道了别人写的论文的功力，有的问题确实可以通过学习大神的论文而得到方便的解决。
3. 线性代数是真的有用啊，有点后悔当初学的时候没有认认真真的学，只是应付过去了。现在看来是真的有用。
4. 对于如何找自己需要的jar包，以及如何引用这些jar包有了些认识，比如矩阵，刚开始自己建了MyMatrix类，准备自己写矩阵左除之类的东西，在实现的过程中感觉这不是这个问题的主要问题，同时也发现了java的jama线性代数jar包，通过引入这个包，解决了自己matrix的问题，算是给自己的编程解决了不少代码上的问题。
5. 对于调试错误更加的得心应手了，debug越来越熟练。
6. 对于自己的学习能力有了很大的提升，要先明确我需要干什么，要学什么，从哪里学，明确了这些以后，具体的学习过程倒是耗时间最少的过程了。

附截图

