

# 扫描全能王？原来图像技术可以这样子玩

原创 神秘的铁头娃 AI算法与图像处理 2019-07-02

收录于话题

#图像技术应用实战

10个 >

点击上方“[AI算法与图像处理](#)”，选择加“星标”或“置顶”

重磅干货，第一时间送达

研究好玩又有用的技术第 001 期

不知道大家会不会有这种感觉，经常是觉得自己学的技术没有用，担心自己能不能胜任工作。因为我们通常学的都是基础的，老师教完之后做几道题目会做考试过了代表学过了。

因此我想找一些“好玩又有用的技术”，我们共同成长。



## 1 APP 测评

首先推荐简单介绍一下这款实用的app，扫描全能王，当然还有其他类似功能的产品。下图是它的一个主要功能介绍：



今天要讲解的技术就是这款app的“手机扫描仪”的功能，具体是可以实现对证件，银行卡，资料等进行扫描，与普通相机最大不同是可以实现几何变形的**自动矫正**，同时还能够实现对文字内容**增强（显示效果更佳）**。

## 2 分析原理

举个例子，对于公交卡进行扫描，由于拍摄人员的技术以及客观的一些原因导致拍摄出来的图片一般情况下存在这一定的几何畸变和其他一些背景的干扰。

如下图所示，拍摄一张公交卡，背景是一张存在条纹的纸张。而我们不希望有这么大的**几何变形以及背景干扰**，会影响下一步的处理（如OCR识别）。



存在的问题：

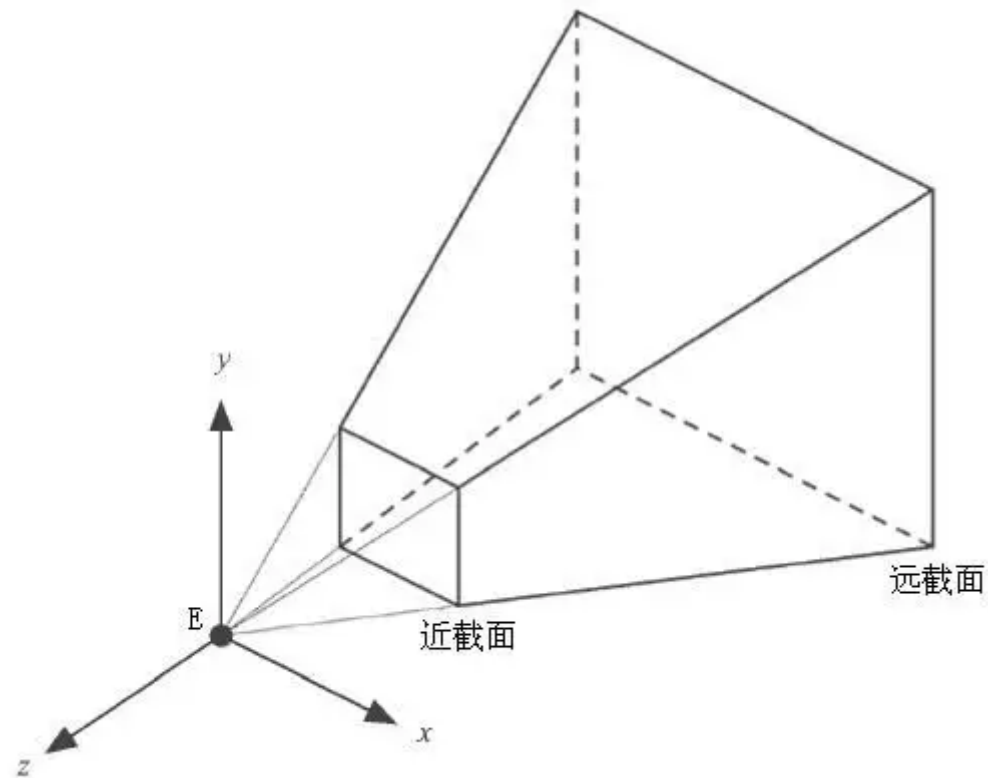
- (1) 几何变形
- (2) 背景干扰

思路：

对于由于拍摄视角导致的变换，一般可采用 **透视变换** 进行矫正。

**什么是透视变换？**

透视变换是将图片投影到一个新的视平面，也称为投影映射。

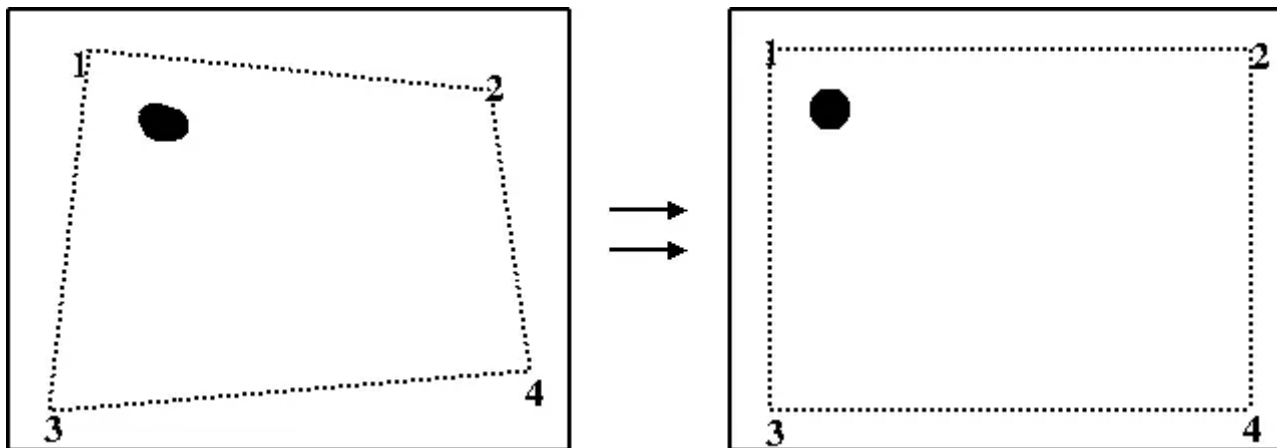


透视投影的标准模型

这里不详细展开透视变换矩阵系数的求法，有兴趣的可以参考下面的链接有详细的推导过程。

我们需要明白一点，求解透视变换的矩阵需要**四组对应点**即可。

下图展示由发生畸变的四边形矫正成一个长方形的示意图



参考资料：

<https://www.cnblogs.com/jsxyhelu/p/4219564.html>

[https://blog.csdn.net/wong\\_judy/article/details/6283019](https://blog.csdn.net/wong_judy/article/details/6283019)

此时，我们已经明白我们需要完成的任务是什么——**找四组对应点**。

### 怎么找？

从上面的一些分析的图，我们可以大概猜到，**是否可以通过变换前后的四个角点构造四组对应点？**

好，对于拍摄图求角点的方法如何求？

我们知道，上图的四个标号序号角点都是相邻两条线段之间相交的点。因此可以通过

- (1) **霍夫检测（直线）+求解直接角点** 来查找拍摄图的四个角点，
- (2) **寻找四边形的轮廓的四个顶点** 来确定拍摄图的四个角点。
- (3) 当然还有其他办法，眼神好的小伙伴可以**手动输入四个顶点的坐标**。

这里，我们已经完成了寻找拍摄图的四个定点。

## 那么对应的点如何获取？

对于你需要扫描的文件，一般都会事先[选型或者说选参数](#)，实际上就是在设置对应点的坐标（变换后的四边形尺寸）。

例如，此时我们想要扫描公交卡，由百度可知，公交卡的尺寸比例是16：9，因此我们只要按照比例进行设置坐标点，公交卡矫正后的形状就能基本满足我们的要求。

这里，我们基本上已经知道如何矫正 **几何变换** 了。

还剩下的问题是 背景干扰？

但是由于我们使用透视变换时，所用的点是四边形的四个角点，这个时候已经将背景剔除在外了，所以背景的干扰自然而然就无需考虑了。

## 3 代码实现

### 思路：

- 1、读入图片并进行预处理
- 2、寻找拍摄图的四个角点
- 3、根据预设的尺寸，设置对应的四个角点，并计算透视变换的矩阵参数
- 4、对拍摄图进行透视变换

为了让大家阅读起来更舒服，就不讲一些非常基础的东西，讲解里面的关键步骤即可

环境：python3.6, opencv 3.4.2

关键步骤一：找四个角点

```
1 # 寻找拍摄图的四个角点
2 # 1、找到处理后图片（边缘处理）的轮廓
3 cnts = cv2.findContours(Image, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)
4 # 2、寻找轮廓四个顶点
5 if len(cnts) > 0:
6     cnts = sorted(cnts, key=cv2.contourArea, reverse=True)
7     for c in cnts:
8         peri = cv2.arcLength(c, True) # 轮廓按大小降序排序
9         approx = cv2.approxPolyDP(c, 0.02 * peri, True) # 获取近似的轮廓
10        if len(approx) == 4: # 近似轮廓有四个顶点
11            docCnt = approx
12            break
```

关键步骤二：根据预设尺寸计算透视变换矩阵

公交卡的比例是16:9，这里假设长高（320,180）

注意：由于并不是正方形，所以这里要先确定长和高的对应

假设，拍摄导致变形不会大到使长和高的尺寸发生变化

顶点的顺序是 左上、左下、右下、右上

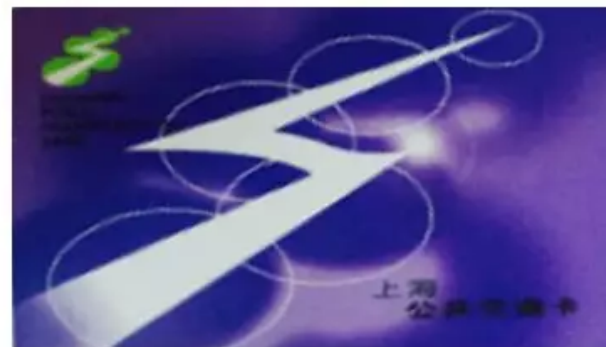


分布计算并比较 左上到左下的距离 左上到右上的距离 确定长高

```
1  if calculate_distance(p[0],p[1])<calculate_distance(p[0],p[3]):
2      pts2 = np.float32([[0,0],[0,180],[320,180],[320,0]])
3      M = cv2.getPerspectiveTransform(pts1,pts2)
4      dst = cv2.warpPerspective(image,M,(320,180))
5  else:
6      pts2 = np.float32([[0,0],[0,320],[180,320],[180,0]])
7      M = cv2.getPerspectiveTransform(pts1,pts2)
8      dst = cv2.warpPerspective(image,M,(180,320))
```

此时即可完成我们的任务。

看一下最终的效果吧



完整代码链接请看

[https://github.com/DWCTOD/AI\\_study/tree/master/%E5%90%88%E6%A0%BC%E7%9A%84CV%E5%B7%A5%E7%A8%8B%E5%B8%88/%E5%AE%9E%E6%88%98%E7%AF%87/opencv/%EF%BC%88%E4%B8%80%E7%BC%89%E5%9B%BE%E5%83%8F%E6%89%AB%E6%8F%8F%E5%8A%9F%E8%83%BD%E2%80%94%E2%80%94%E5%87%A0%E4%BD%95%E7%9F%AB%E6%AD%A3](https://github.com/DWCTOD/AI_study/tree/master/%E5%90%88%E6%A0%BC%E7%9A%84CV%E5%B7%A5%E7%A8%8B%E5%B8%88/%E5%AE%9E%E6%88%98%E7%AF%87/opencv/%EF%BC%88%E4%B8%80%E7%BC%89%E5%9B%BE%E5%83%8F%E6%89%AB%E6%8F%8F%E5%8A%9F%E8%83%BD%E2%80%94%E2%80%94%E5%87%A0%E4%BD%95%E7%9F%AB%E6%AD%A3)

END

非常感谢大家的每一次阅读和在看，大家的每一次点击都是对我的一个鼓励，看到阅读量一个一个上涨是一件非常开心的事情，如果能有一个小小的“在看”，我会更加的激动和努力创造，感谢感谢 >\_<



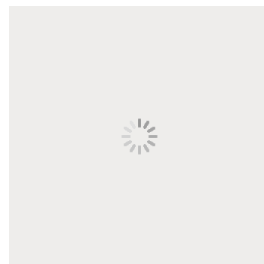
### 好文推荐：

- 《[坚信自己的选择，毕竟自己的路是我自己走出来的](#)》
- 《[阿里|格灵深瞳|地平线|小鹏汽车计算机视觉岗实习面经助您斩获offer](#)》
- 《[强烈推荐的这些Keras资料——让你少走很多弯路](#)》
- 《[推荐!\\_github上四个与100有关的优质项目](#)》
- 《[干货|\\_目标检测入门，看这篇就够了（上）](#)》



加群交流

扫码添加助手，可申请加入AI\_study交流群。一定要备注：研究方向+地点+学校/公司+昵称（如目标检测+上海+上交+卡卡西），不根据格式申请，一律不通过。



喜欢就点个"在看"呗^\_^