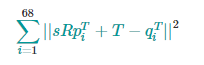
函数get\_landmarks(im)从图片im中获得68 \* 2的矩阵，对应68个点的坐标，不处理多个脸或没有脸的图片，过程：

获取人脸大致范围，用一个矩形区域rect表示：rect = detector(im,1),在rect中使用predictor提取出68个坐标.

其中predictor需要预先训练好，数据集从https://sourceforge.net/projects/dclib/files/dlib/v18.10/shape\_predictor\_68\_face\_landmarks.dat.bz2/download取得。使用的ERT方法，源自论文One Millisecond Face Alignment with an Ensemble of RegressionTrees

获取到两张图的68个点坐标之后需要将两张图的人脸对齐，需要对其中一张图进行旋转(旋转矩阵R)、缩放(缩放因子s)、平移(加上矩阵T)，使得:



最小，其中pi,qi为两幅图对应的点。采用[Ordinary Procrustes Analysis](https://en.wikipedia.org/wiki/Procrustes_analysis#Ordinary_Procrustes_analysis)，具体实现依靠函数transformation\_from\_points(points1, points2),返回仿射变换矩阵M。

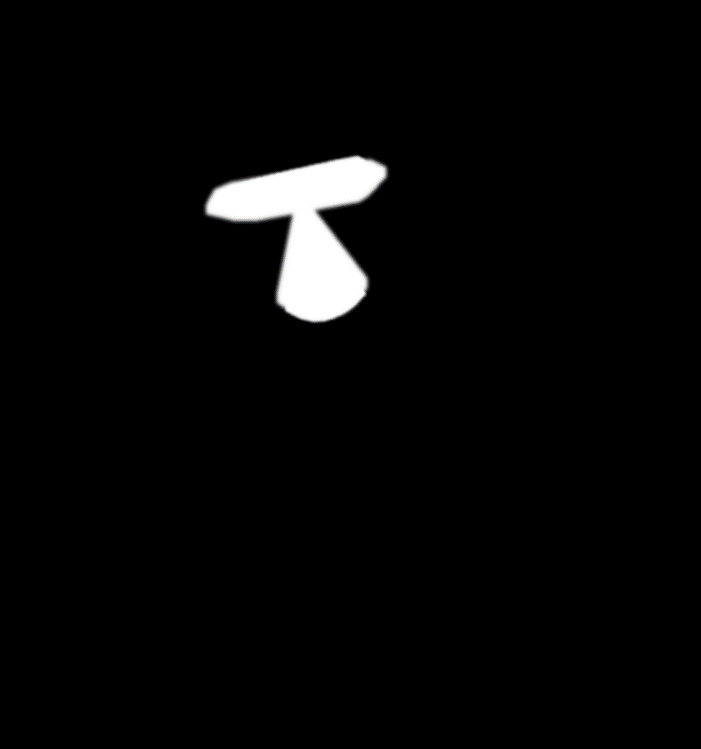
将得到的M作为参数传入opencv的函数warp\_im即可得到与第二张图像位置吻合的图像，效果如下：



使用函数correct\_colours修正覆盖后的图像颜色使得两张脸的肤色相近,效果如下：



制作面具：



使得最终图像中白色部分显示第二张图（脸的图）的对应部分，黑色部分显示第一张图（头的图）的对应部分。

过程：

分别计算眼睛和（鼻子+嘴巴）的凸包，将图片凸包的对应部分变为白色(调用函数draw\_convex\_hull(im, points, color)，points为眼睛或（鼻子+嘴巴）的特征点集，调用后im的凸包部分变为color（设为1表示白色）)，其余部分为黑色，再经过高斯滤波使得面具图像平滑。

两张图都要制作对应的面具(实质为3维的0 1阵)，最终的面具为两幅面具的并，使得第二张图中的脸能够完全覆盖第一张图中的脸。