

墨迹了一天，想明白一件事，其实不需要第二次的特征提取、匹配对应点来定位两张图片对应的位置。

已证明在correct\_H函数中计算的min\_out\_w, min\_out\_h就是对应的像素值。如果是负数说明超出左、上的边界，需要平移调整。之前由于计算单应矩阵H的时候都减去了最小值，导致忽略了min\_out\_w, min\_out\_h为正，不需要调整的情况。平移的时候还是按照min\_out\_w, min\_out\_h的绝对值去平移，自然有的拼接会出错有的则正确。

## 思路



image A

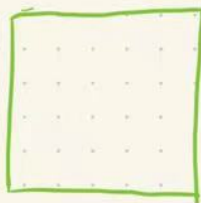


image B

主要耗时

1. 提取 A, B 的特征点

算法: SIFT, SURF, ORB

2. 特征点匹配, 即找到 A, B 对应(或者说同一位置)的特征点

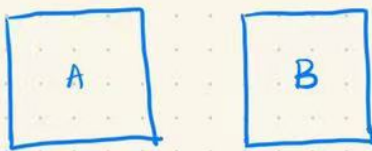
算法:  $\left\{ \begin{array}{l} \text{KnnMatcher} \\ \text{FlannBasedMatcher} \end{array} \right.$

找到后, 筛选良好的特征点

3. 根据对应的点(至少 4 对)计算单应矩阵  $H$

计算  $A \rightarrow B$  的单应矩阵, 把 A 转换到 B 的坐标系

难点在于:



先说一下 warpPerspective 的参数

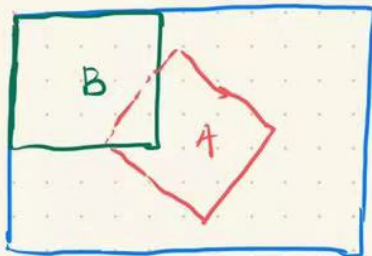
cv2.warpPerspective ( image A , H , ( res ) )

对A进行 H 单位变换后,  
生成图像的 size  $\leftarrow$  res

对A变换后, 其位置在 res 的位置不定, 但若 A, B 拼接, B 一定在左上角, 可能有如下几种情况

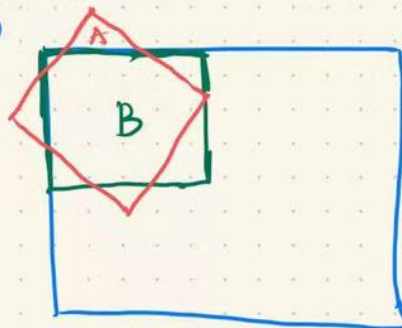
说明: 蓝色框为 res, A 变换后的结果

①



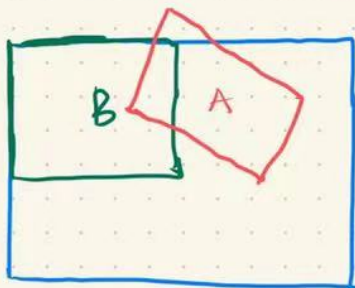
①这是极度理想的情况

②

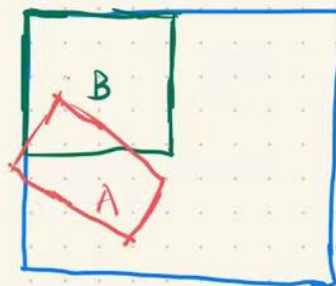


②极度不理想情况

③

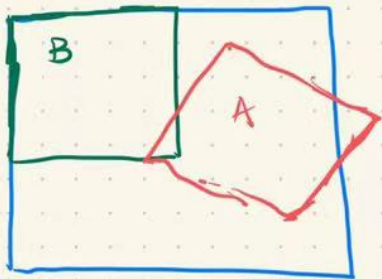


④

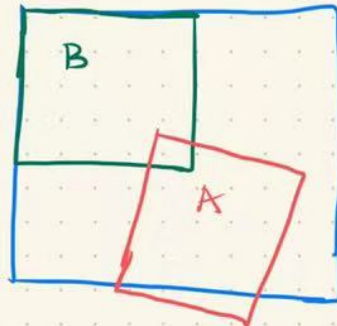


③④ 一个为上边界超出, 一个为左边界超出

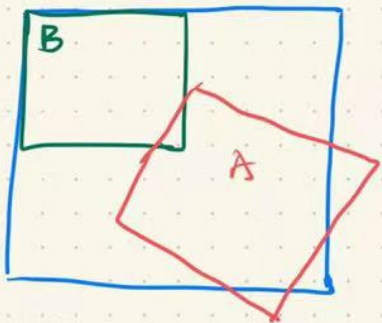
⑤



⑥



⑦



⑤⑥⑦ 都是 `warpPerspective` 的 `res` size 设置小了造成的。

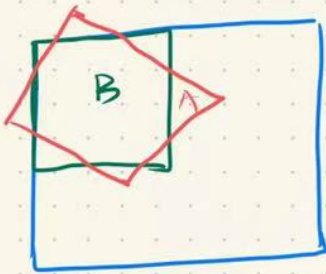
综上所述 7 种情况：②③④ 解决需要去调整 `H`，即对 `A` 进行平移，同时对 `B` 也平移，要分情况讨论。

⑤⑥⑦ 本质是一样的，只需要调整 `warpPerspective` 最后一个 `size` 参数即可，而且两张图像拼接基本不会出这三种情况，可能会出现在多张拼接的情况中。所以在 `Re-Sample` 中设置了一个 `warpedSize` 参数解决。



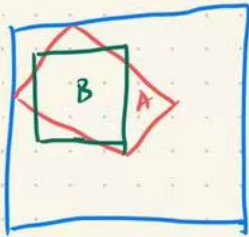
②③④情况, 分别讨论如下:

②

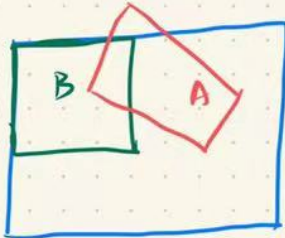


在②中, A图左上都超出边界,  $x, y$  最小值均为负, 所以调整H, 减去最小值 $x, y$ 即平移 同时 B 也做向右, 向下, 对应的平移

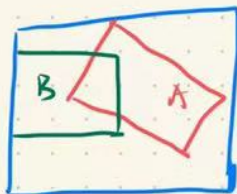
多余黑边裁掉.



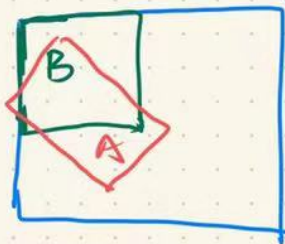
③



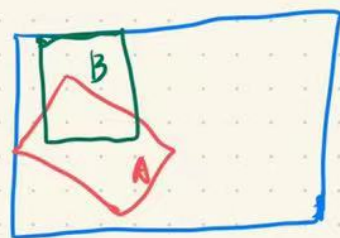
③这种情况下, 只有上超出边界, 只要调整H, 让A向下平移即可 同时 B 也向下平移



④



④ 同理



②③④ 这三种情况均在 `correct-H` 函数中实现，  
根据  $x, y$  最小值是否为负判断属于哪种情况。

在 A 做 `correct-H` 单应变换后，原 A 进行平移，  
B 也要做相应平移，即可拼接成功。B 平移也是按  $x, y$  最小值是否为负来判断哪种情况，并做  $\text{abs}(x \text{ 或 } y)$  的平移。

这样，我们少做一次特征提取和匹配，速度大大增加。

目前，我们得到了第二张图像边界的位置，那么如何进行边界的融合呢？

还可以优化的内容：

1. 新增几个特征提取的算法
2. 添加一个参数实现暴力匹配和flann的切换，但要考虑到例如ORB算法不能用flann去匹配
3. 多张图像拼接，如何保存上一次拼接的结果？新增两个参数，一个为是否保存imageA（默认情况下不保存），另一个是保存的路径，直接保存imageA。如何实现回退？
4. 可以实现增量拼接，result为第一个参数，新图为第二个参数。

需要增添功能

1. 调整两张图的亮度一致，多张图如何实现
2. 畸变调整？新增参数？
- 3.