

## تمرین سوم بینایی - بخش عملی

### سوال ششم - گزارش کار

مهرشاد فلاح اسطخزیر - 401521462

منطق کد:

تابع `sift_feature_matching`:

این تابع وظیفه این را دارد که از SIFT استفاده کند و نقاط کلیدی را در دو تصویر پیدا کند. از یک `BFMatcher` (Brute Force Matcher) هم برای حساب کردن فاصله بین توصیفگرها استفاده شده است (از نرم `L2` استفاده شده است). برای هر توصیفگر در تصویر اول نزدیکترین دو توصیفگر انتخاب می‌شود با توجه به استفاده از `knnMatch`. از قانون Lowe در مرحله بعد استفاده شده است که این قانون بیان می‌کند که اگر فاصله نزدیکترین تطابق ( $m$ ) خیلی بهتر از دومی ( $n$ ) بود ( `m.distance < 0.75 * n.distance:` ) این تطابق، تطابق خوب است. در نهایت نقاط متناظر را با توجه به این تطابق می‌سازیم. در خروجی هم نقاط متناظر تصویر اول و دوم و همچنین عکس تطابق یافته بازگردانده می‌شود.

تابع `ransac_homography`:

این تابع وظیفه این را دارد که بهترین تبدیل را با استفاده از `ransac` پیدا کند. برای این کار ابتدا چهار نقطه رندوم انتخاب می‌شوند و با توجه به آن‌ها یک تبدیل انتخاب می‌کند و تعداد `inlier`ها را با توجه به اعمال تبدیل مشخص می‌کند، اگر از بهترین حالت حال حاضر بهتر باشد به عنوان تبدیل بهتر انتخاب می‌شود در غیر اینصورت انتخاب نمی‌شود. در نهایت بعد از چندین اجرا به تبدیل بهینه می‌رسیم و خروجی تابع می‌شود.

تابع `stitch_images`:

این تابع وظیفه این را دارد که گوشه‌های دو تصویر را پیدا کند و از تبدیل بهینه بر روی عکس‌ها استفاده کند. برای این کار از `cv2.perspectiveTransform` استفاده می‌کنم. در مرحله بعد گوشه تصویر دوم را به تصویر اول می‌چسبانم و نقاط عرض و طول مینیمم و ماکسیمم را پیدا می‌کنم و آرایه `T` را با استفاده از آن‌ها پیدا می‌کنم و در نهایت ضربه نقطه‌ای بین تبدیل `H` و `T` را بر روی تصویر با استفاده از `warpPerspective` اعمال می‌کنم.



در نهایت تصویر خروجی به شکل عکس بالا است.