



## تمرین سری اول

شماره دانشجویی: ۹۸۱۰۸۱۲۴

نام و نام خانوادگی: نیما کلیدری

## پرسش ۱ - Harris Corner Detection and Matching

در تابع اصلی کد پاسخ، تصاویر مورد نظر سوال را دریافت کرده و آنها را به همراه پارامترهای دیگر سوال به تابع  $q1$  می‌دهیم. در این تابع ابتدا مشتقات هر دو تصویر را در دو راستای  $x$  و  $y$  به ترتیب با توابع  $x\_deretive\_get$  و  $y\_deretive\_get$  به روشی که در صورت سوال گفته شده (بدست آوردن مشتق در هر سه کانال رنگی و استفاده از حداکثر آنها) بدست می‌آوریم. در ادامه توان دوی این ماتریس ها و ضرب درایه به درایه آن ها را بدست می‌آوریم و در متغیرهای مربوط قرار می‌دهیم. سپس بنابر رابطه ی معمول گرادیان دو تصویر را بدست می‌آوریم و آنها را ذخیره می‌کنیم.

در ادامه یک فیلتر گاوسی روی این دو ماتریس گرادیان به اندازه  $size$  و سیگمای  $sigma$  اعمال می‌کنیم. سپس دترمینان و تریس را بنابر رابطه ی گفته شده مربوط به روش هریس، بدست آورده و  $R1$  و  $R2$  را از آن استخراج کرده و ذخیره می‌کنیم. با استفاده از یک مقدار آستانه  $Threshhold$ ، کل مقادیر موجود در  $R$  ها را که از حدی کوچکترند برابر با ۰ قرار می‌دهیم. سپس با استفاده از یک تابع دستی و یک مقدار همسایگی  $s$ ، عمل  $non - maximum\ supression$  را برای این ماتریس انجام می‌دهیم که در هر تکه از نقاط کنار هم، تنها نقطه پررنگ تر باقی بماند. پس از اینکار نیز نتیجه را ذخیره می‌کنیم.

پس از ذخیره کردن کل ایندکس های ماتریس های بدست آمده، وارد مرحله ۲ حل سوال می‌شویم. در این مرحله در ابتدا یک همسایگی  $n * n$  حول هر یک از نقاط باقی مانده می‌گیریم و هر یک از این همسایگی ها را، بعنوان یک بردار  $3n^2$  بعدی (که ۳ بدلیل ۳ کانال رنگی متفاوت تصویر است)، در دو لیست  $1\_vector\_properties$ ،  $2\_vector\_properties$  ذخیره می‌کنیم. سپس همانطور که گفته شده است، با طراحی دو حلقه تو در تو، هر کدام از این بردار ها را از تمام بردار های تصویر دیگر کم کرده و نرم آنها را بدست می‌آوریم (نرم  $L_2$ ).

عدد به دست آمده در حقیقت اختلاف کلی این دو قطعه است. سپس کمترین فاصله را تقسیم بر دومین کمترین فاصله کرده و اگر از مقدار آستانه  $astaneh$  کمتر بود، آن را ذخیره می‌کنیم. برای نقاط بدست آمده از تصویر اول به دوم و سپس دوم به اول اینکار را می‌کنیم و با استفاده از یک حلقه، تمام نقاطی را که هم از تصویر یک بر دو منطبق شده و هم از تصویر دو بر یک را جدا کرده و در انتها نیز با استفاده از ۴ عدد حلقه، تمام نقاطی را که با دو نقطه از تصویر دیگر متناظر شده اند را حذف می‌کنیم. در پایان نیز تمام نقاط بدست آمده را با استفاده از دایره نشان داده و در مرحله آخر، آنها را به هم متصل می‌کنیم.

موارد خواسته شده :

۱. انحراف معیار: ۵

۲. اندازه فیلتر: ۲۰

۳. مقدار عدد  $k$ : 0.005۴. مقدار  $n$ : ۱۵۳۵. نوع تابع فاصله:  $L_2$ ۶. مقدار آستانه اول برای حذف نقاط کمرنگ:  $4 \times 10^7$

۷. مقدار آستانه دوم برای بدست آوردن نقاط متناظر مناسب: 0.971

## پرسش ۲ - Perspective

در این سوال تصویر مورد نظر را دریافت کرده و به تابع  $q2$  میبریم. بنابر نتیجه گیری نهایی تئوری دوربین pinhole، داریم:

$$x = K[R|t]X$$

که  $K$  ماتریس کالیراسیون،  $R$  ماتریس دوران و  $t$  بردار انتقال است. ما برای ساده تر شدن،  $t$  را بصورت یک ماتریس  $1 \times 3$  یکه با ستون آخر  $t$  نوشته و در  $R$  ضرب میکنیم:

$$x = KRTX$$

که  $KRT$  را برابر با  $H$  در نظر میگیریم. از آنجا که دو زاویه برای دوربین داریم، پس دو  $H$  مختلف داریم که با  $X$  های ثابت، دو  $x$  متفاوت به ما میدهد. حال ما با داشتن دو تا  $H$  متفاوت و یکی از  $x$  ها، با توجه به  $X$  یکسان، میتوانیم  $x$  دیگر را بدست بیاوریم.

$$x_1 = H_1X, \quad x_2 = H_2X$$

$$H_1^{-1}x_1 = H_2^{-1}x_2$$

$$H_2H_1^{-1}x_1 = x_2$$

پس ما در پاسخ این سوال، به راحتی با استفاده از  $KRT$  کافیت  $H$  ها را ساخته، و سپس ضرب دو ماتریس  $H_2$  و  $H_1^{-1}$  را در هم ضرب کرده و در تصویر اول اعمال میکنیم. (برای خارج نشدن تصویر از کادر، لازم است یک ماتریس شیفت عادی هم به کار ببریم، یعنی آن را ضرب میکنیم. این شیفت از مقدار بیرون زده ی تصویر از قاب (که منفی شده است) بدست می آید که به راحتی از اعمال نگاشت بر ۴ گوشه تصویر بدست می آید؛ بدین ترتیب تصویر دوم بدست می آید، این همان تصویری است که اگر از دوربین جدید نگاه کنیم، آنرا میبینیم. پس با اعمال یک معکوس کلی در مرحله آخر و اعمال آن روی تصویر اصلی، سپس اعمال شیفت گفته شده، سپس با نصب قاب با اندازه مناسب (که از اختلاف سمت های مخالف تصویر پس از نگاشت ۴ گوشه تصویر بدست می آید) و استفاده از تابع آماده برای اعمال این نگاشت، تصویری را بدست میاوریم که اگر بر زمین نصب شود، از دوربین جدید، لوگوی دانشگاه را به شکل درستی میبینیم. پس این تصویر را ساخته و ذخیره میکنیم.

### پرسش ۳ - هموگرافی

در کد پاسخ، ابتدا تصاویر مربوط را دریافت کرده و به تابع  $q3$  می‌دهیم. در این تابع، ابتدا با استفاده از روش SIFT در کتابخانه Open-cv، نقاط کلیدی و دسکریپتورهای آن‌ها استخراج شده و رسم میشوند و ذخیره میشوند. سپس با استفاده از تابع BFMatcher و فاصله  $L_1$  و همچنین دسکریپتورهای بدست آمده، نقاط متناظر را پیدا کرده و آنها را در  $matches$  قرار می‌دهیم. با مرتب سازی  $matches$ ، تنها ۱۰۰۰ تطابق برتر را در نظر می‌گیریم، (زیرا  $match$  های بسیار شده بسیار زیاد هستند) با تبدیل این نقاط به آرایه، فقط نقاط را (بدون رسم خط)، نمایان کرده، و سپس با دادن  $matches$  به تابع  $drawMatches$ ، خط‌ها را رسم کرده و خروجی می‌دهیم.

در ادامه از بین نقاط بدست آمده، ۲۰ تا را بصورت تصادفی انتخاب کرده و نمایش می‌دهیم. حال کل نقاط مبدا و مقصد را از  $matches$  خوانده و به تابع  $findHomography$  و با استفاده از متد RANSAC می‌دهیم. ترشهولد رنسک را در اینجا ۴۰، حداکثر ایتريشن ۳۰ هزار (به علت تعداد بسیار بالای نقاط)، و درصد اطمینان نیز ۹۵ درصد است. این تابع پس از اجرا شدن، دو خروجی  $M$  و  $mask$  را به ما می‌دهد.  $M$ ، یعنی همان ماتریس هموگرافی، در نمونه‌ی اجرا شده در پاسخ با ۳ رقم اعشار، برابر است با:

$$M = \begin{bmatrix} 4.066 & 0.276 & -2626.157 \\ 0.052 & 2.378 & -1170.385 \\ 0 & 0 & 1 \end{bmatrix}$$

$mask$  هم یک ماسک از نقاط متناظری که به تابع داده بودیم می‌باشد که مشخص میکند کدام نقاط این لایر هستند و کدام نقاط اوت لایر. با استفاده از یک حلقه روی نقاط بررسی شده و ماسک دریافت شده، خطوط و نقاط این لایر را در تصور به رنگ قرمز نمایش می‌دهیم. (از آنجا که نقطه پرتی در این نمونه نداشتیم، تصویر ۱۸ همان دو تصویر اصلی در کنار هم هستند) در نقاط گوشه‌ی تصویر اول را به معکوس ماتریس به دست آمده می‌دهیم و نقاط به دست آمده را روی تصویر دوم به هم وصل می‌کنیم که یک چند ضلعی تشکیل شود. حال با اعمال نگاشت معکوس بدست آمده بر روی تصویر اول توسط تابع  $warpPerspective$ ، تصویر مشابه قطعه مشخص شده در تصویر دوم (که با بدست آوردن نقاط چپ و راست و بالا و پایین، آن را به اندازه مناسب می‌بریم) بدست آمده که این دو را کنار هم قرار می‌دهیم و ذخیره می‌کنیم.

سپس نگاشت را بر روی نقاط گوشه‌ی تصویر دوم اعمال کرده و منفی ترینشان را هم عرضی و هم طولی بدست آورده و با استفاده از این دو عدد، یک نگاشت شیفت به سمت مثبت ساخته و با ضرب آن در نگاشت اصلی، نگاشتی را بدست می‌آوریم که کل تصویر دوم را تغییر می‌دهد و چیزی از بین نمی‌رود. پس از اعمال این نگاشت روی تصویر دوم، نتیجه تغییر یافته تصویر دوم را بدست آورده، گوشه‌های خالی اطرافش را بریده و ذخیره می‌کنیم (در این نتیجه خط مشخص کننده بخش اول نداریم). سپس همین تصویر را در حالت خط دار کنار تصویر اول نمایش می‌دهیم و نتیجه‌ی مورد نظر آخر هم بدست می‌آوریم.

## پرسش ۴ - محاسبه هموگرافی و پیاده سازی RANSAC

در کد پاسخ، ابتدا تصاویر مربوط را دریافت کرده و به تابع  $q4$  میدهم. در این تابع، ابتدا با استفاده از روش SIFT در کتابخانه Open-cv، نقاط کلیدی و دسکریپتورهای آن‌ها استخراج شده و رسم میشوند و ذخیره میشوند. سپس با استفاده از تابع BFMatcher و فاصله  $L_1$  و همچنین دسکریپتورهای بدست آمده، نقاط متناظر را پیدا کرده و آنها را در `matches` قرار میدهم. با مرتب سازی `matches`، تنها ۱۰۰۰ تطابق برتر را در نظر میگیریم، (زیرا `match` های بسیار شده بسیار زیاد هستند) با تبدیل این نقاط به آرایه، فقط نقاط را (بدون رسم خط)، نمایان کرده، و سپس با دادن `matches` به تابع `drawMatches`، خط‌ها را رسم کرده و خروجی میدهم. در ادامه از بین نقاط بدست آمده، ۲۰ تا را بصورت تصادفی انتخاب کرده و نمایش میدهم.

در ادامه، آرایه ای از کل نقاط موجود در `match` میسازیم که به دو دسته `src` و `dst` تقسیم میشوند. لیست نقاط مبدأ و `dst` لیست نقاط مقصد میباشند. سپس با استفاده از دو حلقه تو در تو، الگوریتم RANSAC را پیاده میکنیم. در اینجا تعداد `Iteration` های ما ۸ هزار، ترشهود ۶۰ و درصد اطمینان ۷۰ درصد را در نظر میگیریم. حال بعد از ساختن یک آرایه دو بعدی به نام `stack` با مقادیر اولیه صفر، وارد حلقه اول میشویم. در این حلقه ۴ ایندکس رندوم انتخاب کرده و با دادن این لیست ۴ تایی تصادفی از نقاط به آرایه های نقاط مبدأ و مقصد، ۴ نقطه مبدأ و ۴ نقطه مقصد تصادفی بدست می‌آوریم. اکنون اگر این دو لیست از نقاط به همراه تعداد آن‌ها را به تابع `matrix_make` بدهیم، به ما یک ماتریس هموگرافی برمیگرداند. این تابع، با استفاده از ساختن  $A$  و  $b$  و حل آن،  $h$  را بدست می‌آوریم. (این ماتریس‌ها از راه DLT برای ۴ نقطه بدست می‌آید؛ در حقیقت ۴ معادله ۲ معادله ۸ مجهول را با هم ترکیب کرده ایم).

$$A = \begin{bmatrix} x_1 & y_1 & 1 & 0 & 0 & 0 & -x'_1x_1 & -x'_1y_1 \\ 0 & 0 & 0 & x_1 & y_1 & 1 & -y'_1x_1 & -y'_1y_1 \\ x_2 & y_2 & 1 & 0 & 0 & 0 & -x'_2x_2 & -x'_2y_2 \\ 0 & 0 & 0 & x_2 & y_2 & 1 & -y'_2x_2 & -y'_2y_2 \\ x_3 & y_3 & 1 & 0 & 0 & 0 & -x'_3x_3 & -x'_3y_3 \\ 0 & 0 & 0 & x_3 & y_3 & 1 & -y'_3x_3 & -y'_3y_3 \\ x_4 & y_4 & 1 & 0 & 0 & 0 & -x'_4x_4 & -x'_4y_4 \\ 0 & 0 & 0 & x_4 & y_4 & 1 & -y'_4x_4 & -y'_4y_4 \end{bmatrix}$$

$$b^T = [x'_1 \quad y'_1 \quad x'_2 \quad y'_2 \quad x'_3 \quad y'_3 \quad x'_4 \quad y'_4]$$

$$Ah = b, h^T = [h_1 \quad h_2 \quad h_3 \quad h_4 \quad h_5 \quad h_6 \quad h_7 \quad h_8]$$

که با تغییر شکل آن، ماتریس هموگرافی از تصویر ۲ به ۱ میشود:

$$H = \begin{bmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & 1 \end{bmatrix}$$

امکان دارد نقاط انتخاب شده یک نگاشت واحد به ما ندهند که در اینصورت با استفاده از `try` و `except`، بجای  $H$ ، ماتریس یک خروجی می‌دهیم. (که به باگ بیشتری در ادامه نخوریم و این ایتريشن را نادیده بگیریم).

بعد از خروجی دادن این ماتریس، با عوض کردن جای `src` و `dst`، معکوس آنرا نیز حساب میکنیم. سپس وارد حلقه ای میشویم که نگاشت بدست آمده را روی نقاط مقصد و وارون آن را روی نقاط مبدأ اعمال میکند و فاصله ی خود آن نقاط با نگاشت یافته ی متناظرشان را حساب و جمع میکند. اگر این مجموع بزرگتر از ترشهود بود، مقدار ستون  $i$  ام و سطر  $j$  ام از استک را ۱ میگذارد. در پایان این حلقه، اگر درصد نقاط از ضریب اطمینان بیشتر بودند، حلقه اولی را پایان میدهم. در غیر اینصورت، تا پایان حلقه صبر میکنیم و ستونی که بیشترین ۱ را داشت را ذخیره میکند. این ستون همان مقدار `mask` قبلی است و با آن نقاط پیدا شده را با تابع مشابه سوال ۳ رسم میکنیم.

حال کل نقاط مناسب بدست آمده را جدا میکنیم و با استفاده از تابع `homography_make`، ماتریس هموگرافی اصلی پاسخ سوال را بدست می‌آوریم. این تابع نیز با روش تجزیه `svd`، پاسخ مسئله بهینه سازی

$$||Hx' - x||$$

را بدست می آورد. پاسخ مسئله بنابر مطالب تدریس شده، به این صورت است که ماتریس  $A$  را بصورت مقابل میسازیم:

$$A = \begin{bmatrix} x_1 & y_1 & 1 & 0 & 0 & 0 & -x'_1 x_1 & -x'_1 y_1 & -x'_1 \\ 0 & 0 & 0 & x_1 & y_1 & 1 & -y'_1 x_1 & -y'_1 y_1 & -x'_1 \\ & & & & & \vdots & & & \\ x_n & y_n & 1 & 0 & 0 & 0 & -x'_n x_n & -x'_n y_n & -x'_n \\ 0 & 0 & 0 & x_n & y_n & 1 & -y'_n x_n & -y'_n y_n & -x'_n \end{bmatrix} = U \Sigma V$$

که باید  $\|Ah - 0\|^2$  را حداقل کنیم. برای اینکار تجزیه svd را انجام داده و ستون آخر ماتریس  $d$ ، نه تا درایه دارد که این ۹ تا، بعنوان یک ماتریس آپ تو اسکیل برای  $H$  حساب می شوند. این ماتریس را تغییر شکل داده و خروجی می دهیم.

$$H = \begin{bmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{bmatrix}$$

پس از خروجی دادن تابع، با تقسیم همه درایه ها بر  $H_{22}$ ، نگاشت اصلی بدست می آید که آن را هم وارونه می کنیم تا نگاشت تصویر اول به دوم بدست بیاید. ماتریس تصویر دوم به اول بدست آمده بصورت زیر می باشد:

$$M = \begin{bmatrix} 3.313 & 0.312 & -2187.426 \\ 0.029 & 2.063 & -996.490 \\ 0 & 0 & 1 \end{bmatrix}$$

mask که در مرحله قبل بدست آوردیم هم یک ماسک از نقاط متناظری که به تابع داده بودیم می باشد که مشخص میکند کدام نقاط این لایر هستند و کدام نقاط اوت لایر. با استفاده از یک حلقه روی نقاط بررسی شده و ماسک دریافت شده، خطوط و نقاط این لایر را در تصور به رنگ قرمز نمایش می دهیم. (از آنجا که نقطه پرتی در این نمونه نداشتیم، تصویر ۲۷ همان دو تصویر اصلی در کنار هم هستند) در نقاط گوشه ی تصویر اول را به معکوس ماتریس به دست آمده می دهیم و نقاط به دست آمده را روی تصویر دوم به هم وصل می کنیم که یک چند ضلعی تشکیل شود. حال با اعمال نگاشت معکوس بدست آمده بر بروی تصویر اول توسط تابع  $\text{warpPerspective}$ ، تصویر مشابه قطعه مشخص شده در تصویر دوم (که با بدست آوردن نقاط چپ و راست و بالا و پایین، آن را به اندازه مناسب می بریم) بدست آمده که این دو را کنار هم قرار می دهیم و ذخیره می کنیم.

سپس نگاشت را بر روی نقاط گوشه ی تصویر دوم اعمال کرده و منفی ترینشان را هم عرضی و هم طولی بدست آورده و با استفاده از این دو عدد، یک نگاشت شیفت به سمت مثبت ساخته و با ضرب آن در نگاشت اصلی، نگاشتی را بدست می آوریم که کل تصویر دوم را تغییر می دهد و چیزی از بین نمی رود. پس از اعمال این نگاشت روی تصویر دوم، نتیجه تغییر یافته تصویر دوم را بدست آورده، گوشه های خالی اطرافش را بریده و ذخیره می کنیم (در این نتیجه خط مشخص کننده بخش اول نداریم). سپس همین تصویر را در حالت خط دار کنار تصویر اول نمایش می دهیم و نتیجه ی مورد نظر آخر هم بدست می آوریم.