

۱. در این سوال ابتدا مختصات وسط ستاره را پیدا کرده و رئوس ستاره را با استفاده از فاصله‌ی نقاط سیاه از مرکز ستاره بدست می‌آوریم.

در ابتدا `BASE_DIR` را برابر با فولدری که در آن هستید قرار دهید و `img_name` را برابر با نام عکس ستاره قرار دهید و در نهایت `new_img_name` را برابر با نام تصویری که می‌خواهید عکس جدید ذخیره شود قرار دهید و سپس دستور `python hw1_q1.py` را اجرا کنید.

در ابتدا کد عکس را می‌خواند و با استفاده از `get_star_center` مختصات مرکز ستاره را پیدا می‌کند، این تابع علاوه بر مختصات مرکز ستاره یک `threshold` بر می‌گرداند که متناسب با پیکسل‌های سیاه عکس می‌باشد که از آن در تابع `normalize_corners` استفاده می‌شود برای حذف نقاط نزدیک به هم.

سپس در تابع `get_corners` نقاط سیاه را به ترتیب فاصلشان از مرکز ستاره مرتب می‌کنیم و پس از نرمالایز کردن آن‌ها و حذف نقاط نزدیک به هم ۵ تا نقاط با بیشترین فاصله و ۵ تا نقاط با کمترین فاصله را به عنوان رئوس ستاره بر می‌گرداند.

در نهایت رئوس را به ترتیب زاویه‌ای که دارند مرتب می‌کنیم.

عکس جدید ستاره‌ها با نام‌های `outfile1.jpg` و `outfile2.jpg` و `outfile3.jpg` ضمیمه شده‌اند و رنگ رئوس به ترتیب نوشته شده در زیر تیره می‌شوند.

(I) star ۱ :

(y, x)

(۰, ۱۹۱)

(۲۰۱, ۱۹۳)

(۲۶۳, ۱)

(۳۲۲, ۱۹۲)

(۵۲۵, ۱۹۱)

(۳۶۱, ۳۰۸)

(۴۲۵، ۴۹۹)

(۲۶۲، ۳۸۰)

(۱۰۰، ۴۹۹)

(۱۶۴، ۳۰۹)

star۲: (ب)

(y, x)

(۳۳، ۲۱۴)

(۱۶۴، ۱۷۹)

(۱۷۰، ۴۶)

(۲۴۱، ۱۵۹)

(۳۷۰، ۱۲۳)

(۲۸۶، ۲۲۵)

(۳۵۹، ۳۳۸)

(۲۳۳، ۲۹۰)

(۱۵۱، ۳۹۴)

(۱۵۸، ۲۶۰)

star۳: (ج)

(y, x)

(۱۱، ۳۷)

(۳۷، ۳۲)

(۴۰، ۶)

(۵۳، ۲۹)

(۷۹، ۲۴)

(۶۱، ۴۳)

(۷۳، ۶۶)

(۵۰، ۵۵)

(۳۲، ۷۴)

(۳۵، ۴۸)

۲. در ابتدا `BASE_DIR` را برابر با فولدري که در آن هستيد قرار دهيد و سپس دستور `python hw\q\py` را اجرا کنيد. فيلم قسمت الف با نام `corners.mp4` در ضميمه وجود دارد و همچنين عکس قسمت ب با نام `trajectory.jpg` در ضميمه وجود دارد.

در اين سوال با استفاده از الگوريتم `inter est points, Harris` را پيدا کرده و به ازای هر فریم، نقاط انتخاب شده‌ی آن فریم با فریم بعدی آن را مقایسه می‌کند و نقاط شبیه به هم و نزدیک به هم را به عنوان نقاط `match` شده انتخاب می‌کند و سپس گرافي تشکیل می‌دهد به این صورت از هر نقطه به نقطه‌ای که با فریم بعدیش `match` شده است یال می‌کشیم و سپس با کمک این گراف رئوسی که عمق بیشتری دارند را به عنوان نقاط انتخاب شده بر می‌گرداند، این کار را روی ۵۰ فریم اول انجام می‌دهیم که می‌توان با تغییر دادن متغیر `TRAJECTORY_FRAMES_COUNT` فریم های بیشتر و یا کمتری را تست کرد.

توضیح کد:

(آ) `extract_video`: در این تابع فریم های فیلم به عکس تبدیل می‌شود.

(ب) `corner_detection`: در این تابع با استفاده از الگوريتم `intrest points, Harris` را بدست می‌آوریم.

(ج) `normalize_corners`: در این تابع نقاط نزدیک بهم را حذف می‌کنیم تا محاسبات راحت تر و سریع تر شود.

(د) `match_corners`: در این تابع نقاط ۲ فریم متوالی را با استفاده از فاصله‌ی آن‌ها و میزان شباهتشان به هم `match` می‌کند.

(ه) `make_grap`: این تابع گراف بین فریم‌های `start` و `end` که ورودی می‌گیرد را می‌سازد، به این صورت که با استفاده از `corner_detection` و `normalize_corners` و `match_corners` که پیش از این گفته شده است گراف مجاورت را می‌سازد.

(و) `get_intrest_points`: در این تابع نقاط به ترتیب طولانی ترین مسیرشان در گراف مرتب می‌شوند

(ز) `normalize_intrest_points`: نقاط نزدیک به هم را حذف می‌کند.

(ح) $get_trajectory$: در این تابع $make_graph$ صدا زده می‌شود و با استفاده از گراف ساخته شده و استفاده از توابع $get_interest_points$ و $normalize_interest_points$ نقاطی از گراف که بلندترین مسیرها را دارند انتخاب می‌شوند (نه لزوماً نقاطی که طولانی‌ترین مسیر را دارند زیرا ممکن است آن‌ها نقاط ثابت باشند) و با راه رفتن روی یال‌های این رئوس $trajectory$ محاسبه می‌شود.

۳. در ابتدا مختصات ۴ گوشه‌ی عکس را طوری محاسبه می‌کنیم که $principal_line$ از مرکز تصویر عبور کند، برای این کار مختصات وسط عکس را از مختصات نقاط ۴ گوشه کم می‌کنیم و چون در ابتدا نقاط پایین عکس مختصات بیشتر از نقاط بالای عکس دارند y آن‌ها را در منفی ۱ ضرب می‌کنیم.

$$corners_skew(:, 1) = corners_skew(:, 1) - principal_point_skew(1);$$

$$corners_skew(:, 2) = corners_skew(:, 2) - principal_point_skew(2);$$

چون عکس روی فاصله کانونی تشکیل می‌شود پس مختصات z آن‌ها برابر با فاصله کانونی می‌باشد.

$$corners_skew = [corners_skew, focal_length_y * ones(4, 1)];$$

ماتریس جابجایی برابر است با جابجایی مرکز دستگاه عام نسبت به مرکز دوربین اول پس برابر است با مختصات دوربین اول نسبت به وسط زمین.

$$t_skew = view_skew;$$

برای به دست آوردن ماتریس جابجایی باید جهت بردارهای مختصات دوربین اول را بدست آوریم و چون محور z دوربین اول به سمت مرکز دستگاه عام است و جهت آن به سمت مرکز عام است پس:

$$z_axis_skew = -view_skew;$$

چون x دوربین با x عام در یک صفحه قرار دارند و x و y و z دو به دو برهم عمودند پس داریم:

$$x_axis_skew = cross(y_axis_skew, z_axis_skew)$$

حال با توجه با خواص ماتریس دوران این بردارها را نرمال می‌کنیم و سپس ماتریس دوران را می‌سازیم. برای بدست آوردن بردار نرمال عکس دوربین اول باید z آن را بدست آوریم که برابر است با:

$$n_skew = R_skew * [0; 0; 1];$$

حال عمق نقاط را بدست می‌آوریم و با استفاده از آن مختصات ۳ بعدی این ۴ گوشه را نسبت به مختصات دوربین اول بدست می‌آوریم که برابر است با حاصلضرب عمق آن نقطه و مختصاتش.

$$\text{corners}^3 D_{\text{skew}} = \text{repmat}(\text{depth}, [1, 3]) . * \text{corners_skew};$$

حال مختصات تصویر ۴ گوشه روی زمین را در دستگاه مختصات دوربین اول بدست می‌آوریم برای این کار وارون ماترسی دوران را در آن ضرب می‌کنیم و آن را با بردار جابجایی جمع می‌کنیم پس داریم:

$$\text{corners}^3 D = R_{\text{skew}}(-1) * \text{corners}^3 D_{\text{skew}};$$

حال ماتریس‌های دوران و جابجایی دوربین دوم را نسبت به دستگاه عام بدست می‌آوریم (همانند آنچه برای دوربین اول انجام دادیم)

$$t_up = \text{view_up};$$

$$z_axis_up = -\text{view_up};$$

$$x_axis_up = \text{cross}(y_axis_up, z_axis_up);$$

سپس مختصات نقاط روی زمین را از دستگاه عام به دستگاه دوربین دوم می‌بریم برای این کار مختصات نقاط را منهای بردار جابجایی می‌کنیم و سپس آن را در ماتریس دوران ضرب می‌کنیم.

$$\text{corners}^3 Dup = \text{corners}^3 D - \text{repmat}(t_up', [4, 1]);$$

واضح است که نسبت فاصله کانونی به فاصله زمین با دوربین دم برابر است با نسبت مختصات در عکس به مختصات در زمین پس داریم:

$$\text{corners_up} = \text{corners}^3 Dup ./ \text{repmat}(\text{corners}^3 Dup(:, 3), [1, 3]) * \text{focal_length_y};$$

حال سائز عکس جدید را بدست می‌آوریم و سپس متناظر هر نقطه در عکس دوم را در عکس اول پیدا می‌کنیم، برای این کار مختصات ۴ گوشه‌ی عکس دوم را همانند عکس اول پیدا می‌کنیم.

$$\text{pixels_up}(:, 1) = \text{pixels_up}(:, 1) - \text{principal_point_up}(1);$$

$$\text{pixels_up}(:, 2) = \text{pixels_up}(:, 2) - \text{principal_point_up}(2);$$

$$pixels_up = [pixels_up focal_length_y * ones(size(pixels_up, 1), 1)];$$

همانند بالا مختصات هر نقطه روی زمین را در دستگاه مختصات دپرین دوم بدست می‌آوریم:

$$depth = -d_up ./ sum((pixels_up .* repmat(n_up', size(pixels_up, 1), 1)), 2);$$

حال مختصات این ۴ گوشه را به دستگاه عام می‌بریم.

$$pixels^3D = R_up^{(1)} * pixels^3D_up';$$

حال با استفاده از مختصات نقاط در دستگاه عام، مختصات هر نقطه در عکس دوربین اول را بدست می‌آوریم.

$$pixels^3D_skew = pixels^3D - repmat(t_skew', [size(pixels^3D, 1) 1]);$$

$$pixels^3D_skew = R_skew * pixels^3D_skew';$$

حال با استفاده از نقاط متناظر رنگ‌های عکس دوم را بدست می‌آوریم.