

به نام خدا



دانشگاه صنعتی شریف  
دانشکده مهندسی برق

## دید کامپیوتری

پاییز ۱۴۰۲

مدرس: دکتر محمدزاده

گردآورندگان: میثم امیرسرداری، زینب شریفی (تئوری) /

مهرداد مرسلی، زینب شریفی (عملی)

تمرین چهارم

### نکات مهم پیش از انجام تمرین:

۱. تمرین‌ها پیش از تقدیم شدن به دانشجویان گرامی، پیاده‌سازی و بررسی شده‌اند. با این حال در صورت وجود هرگونه ابهام در تمرین، آن را صرفاً در گروه تلگرام مربوط به درس مطرح فرمایید.
۲. هرگونه مشابهت بین تمرین‌های دانشجویان غیرمجاز است؛ لذا از کپی کردن کد از منابع اینترنتی یا سایر دانشجویان خودداری فرمایید. در صورت مشاهده، نمره آن سوال به هیچکدام از عزیزان تعلق نخواهد گرفت. توجه شود که مشابهت کدهای شما با ابزارهای مشابهت سنج بررسی خواهند شد.
۳. دقت کنید کدهای شما باید قابلیت اجرای دوباره داشته باشند، در صورت دادن خطا هنگام اجرا، حتی بدلیل اشتباه تایپی، نمره صفر به آن بخش تعلق خواهد گرفت.
۴. لطفاً تصویری واضح از پاسخ سوالات تئوری خود آپلود کنید. در غیر اینصورت تمرین شما تصحیح نخواهد شد.
۵. تمیزی و خوانایی کدهای شما نمره امتیازی سوالات عملی خواهد بود.
۶. کدها، نتایج و گزارشکار بخش عملی را می‌توانید به فرمت ipynb تحویل دهید. کدهای خود را در نهایت به صورت py نیز تبدیل کنید و هر دو فایل را بفرستید.
۷. فایل‌های مربوط به تمرین را به فرمت HW#\_StudentNumber\_StudentName.zip آپلود کنید.

۲۰ + ۱۱۰ نمره

سوالات تئوری

### تمرین اول: (۱۵ نمره)

(الف)

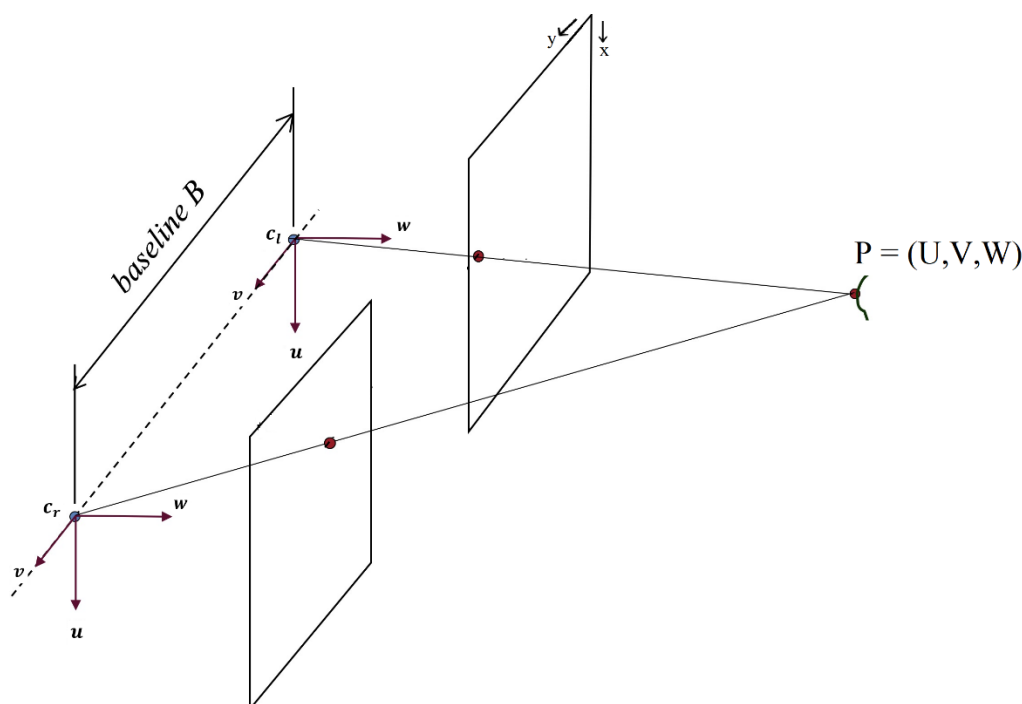
۱. درباره رابطه مورد نیاز برای تبدیل نمایش نقاط از pixel coordinates به normalized image coordinates در یک تصویر دو بعدی تحقیق کنید و تفاوت این دو نوع مختصات را ذکر کنید. (۳ نمره)
۲. فرض کنید که یک جفت دوربین stereo بر روی یک نقطه ثابت شده اند به طوری که optical axis آن‌ها در این نقطه با یکدیگر برخورد میکند. نشان دهید برای مختصات نرمال شده (normalized image coordinates) در ماتریس Fundamental تعریف شده بین این دو دید، المان  $F_{33}$  همواره صفر است. (۷ نمره)

ب) برای هر دو نقطه  $x_l$  و  $x_r$  متناظر در تصویر دوربین راست و چپ، خطوط epipolar متناظر با آن ها  $l_l$  و  $l_r$  و خط دلخواه  $k$  در تصویر دوربین چپ که از  $e_l$  نمیگذرد نشان دهید:

$$l_r = F[k]_x l_l$$

که در آن  $[k]_x$  بیان ماتریسی ضرب خارجی در بردار  $k$  است. (راهنمایی: صفحه ۶ اسلاید Two-View Geometry) (۵ نمره)

تمرین دوم: (۴۵ نمره)



شکل بالا نمایی از نحوه قرارگیری دو دوربین stereo را نشان میدهد. فرض کنید مختصات جهان بیرون دقیقاً مطابق بر مختصات مرکز دوربین چپ است (بدون دوران و جابجایی) و مرکز دوربین راست مطابق شکل در مختصات  $(0, B, 0)^T$  جهان بیرون قرار دارد. (دوربین راست دورانی نسبت به دوربین چپ ندارد)

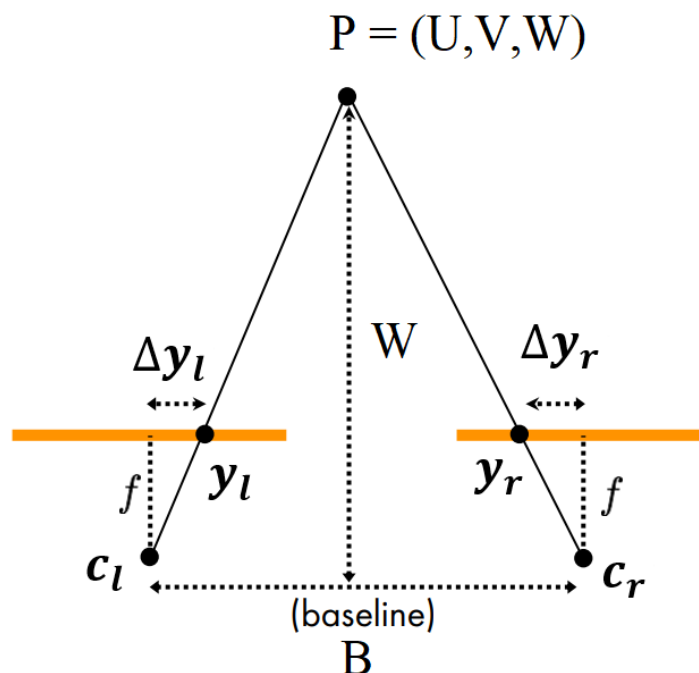
الف) با نوشتن فرم کلی ماتریس دوربین چپ و راست و جایگذاری مقادیر در فرمول ماتریس  $F$  فرم بسته ساده شده آن را بیابید. (۵ نمره)

ب) مختصات همگن epipole راست ( $e_r$ ) را بدست بیاورید. (راهنمایی: صفحه ۴ اسلاید Two-View Geometry) با توجه به اعداد بدست آمده، این نقطه در مختصات ناهمگن تصویر دو بعدی در کجا قرار دارد؟ با توجه به این که تمام خطوط epipolar از epipole عبور میکنند، حدس میزنید در صفحه تصویر دوربین راست خطوط epipolar چه حالت خاصی دارند؟ آیا همین استدلال ها برای دوربین چپ نیز قابل قبول است؟ (۱۰ نمره)

ج) با جایگذاری  $e_r$  در فرم بسته ماتریس  $F$  که در قسمت الف بدست آوردید، فرم  $3 \times 3$  این ماتریس را بنویسید. سپس با نوشتن رابطه  $x_r^T F x_l = 0$  برای دو نقطه متناظر دلخواه نشان دهید نقاط متناظر در این حالت خاص بر روی خطوط افقی موازی با  $x$  یکسان قرار دارند. (توجه کنید  $x$  طبق قرارداد محور عمودی تصویر است). (۵ نمره)

د) میخواهیم فاصله میان دو نقطه متناظر دلخواه در تصویر راست و چپ را اندازه بگیریم. با توجه به قسمت ج میدانیم این دو نقطه بر روی خطوط موازی با  $x$  یکسان قرار دارند و تنها تفاوت آن ها در مقدار  $y$  است. با توجه به شکل زیر ( که همان ساختار شکل قبل است و تنها برای راحتی محور  $y$  تصاویر نشان داده شده) و نوشتن روابط مثلثاتی نشان دهید: (۱۰ نمره)

$$\Delta y = \Delta y_r + \Delta y_l = \frac{fB}{s_y W} \text{ (in pixels)}$$



و) عبارت بدست آمده در قسمت قبل که به disparity ( $d$ ) میان هر دو پیکسل متناظر شناخته شده است در کاربردهای تخمین عمق مورد استفاده قرار میگیرد. با توجه به این عبارت به سوالات زیر پاسخ دهید: (۱۵ نمره)

۱. میزان جابجایی نقاط متناظر در دو تصویر برای نقاط نزدیک به دوربین ها و دور از دوربین ها چگونه است؟ نمودار  $W$  بر حسب  $d$  را رسم کنید.
۲. رابطه میان میزان خطا محاسبه  $d$  و میزان خطای عمق تخمین زده شده را بدست بیاورید و بیان کنید برای یک الگوریتم مشخص برقراری تناظر میان نقاط (مثلا SIFT feature matching) و خطای ثابت آن برای تمام پیکسل ها، خطای تخمین عمق نقاط نزدیک به دوربین بیشتر است یا نقاط دور از دوربین؟ (راهنمایی: از  $\frac{\partial d}{\partial W}$  استفاده کنید).
۳. اگر چشم های انسان را به عنوان یک سیستم stereo در نظر بگیریم، در نگاه کردن به یک صحنه یکسان کدام یک از دو شخص زیر قادر به مشاهده اشیاء دورتر (با عمق زیادتر) است؟ کدام یک فاصله اشیاء نزدیک را با دقت بهتری تشخیص میدهد؟



### تمرین سوم: (۲۰ نمره)

در این تمرین میخواهیم با ماتریس Essential (E) که حالت خاصی از ماتریس F در مختصات نرمال شده تصاویر (تمرین اول قسمت الف) است آشنا شویم. اگر ماتریس دوربین در حالت کلی به فرم  $M = K[R : t]$  باشد آنگاه:

الف) برای نوشتن معادلات دوربین با مختصات نرمال شده،  $K^{-1}$  را در دو طرف عبارت تبدیل مختصات جهان بیرون به مختصات پیکسلی (صفحه ۱۹ اسلاید Camera Model) ضرب کنید و عبارت جدید بین مختصات نرمال شده تصویر و جهان بیرون به واسطه اعمال ماتریس دوربین نرمال شده را بنویسید. (X برای مختصات جهان بیرون، x برای مختصات پیکسلی و  $\hat{x}$  برای مختصات نرمال شده) (۳ نمره)

ب) با فرض این که مختصات جهان بیرون برای دو دوربین stereo دلخواه بر مختصات دوربین چپ منطبق است و برای دوربین راست ماتریس پارامترهای خارجی به صورت  $[R|t]$  است، با جایگذاری ماتریس های دوربین نرمال شده در فرمول ماتریس F و محاسبه  $e_r$  در مختصات نرمال شده، نشان دهید ماتریس E به فرم زیر نمایش داده میشود: (۷ نمره)

$$E = [t]_{\times} R$$

ج) در رابطه معرف ماتریس E یعنی  $\hat{x}_r^T E \hat{x}_l = 0$  با جایگذاری  $x_l$  و  $x_r$  و مقایسه با رابطه معرف ماتریس F، رابطه ماتریس E را بر حسب ماتریس F بدست آورید. (۵ نمره)

د) نشان دهید ماتریس E یک مقدار تکین صفر و دو مقدار تکین غیر صفر برابر دارد. (یادآوری: مقادیر تکین ماتریس E ریشه دوم مقادیر ویژه ماتریس  $EE^T$  هستند). (۵ نمره)

### تمرین چهارم: (۳۰ نمره)

دو دوربین stereo کاملاً یکسان ( $K=K'$ ) در فاصله h از یکدیگر به صورت متقارن قرار گرفته اند به نحوی که optical axis آن ها در وسط حد فاصل دو دوربین با یکدیگر زاویه ۴۵ درجه تشکیل میدهند و صفحه v-w دو دوربین یکی هستند. برای راحتی فرض کنید مختصات جهان بیرون بر مختصات دوربین چپ کاملاً منطبق است. (بدون دوران) و مختصات دوربین راست از دوران ۴۵ درجه ای پادساعتگرد حول محور u دوربین چپ بدست می آید.

الف) پس از ترسیم آرایش کلی دوربین ها و مشخص کردن مختصات، با استفاده از روابط مثلثاتی ابتدا محل قرارگیری epipole راست و چپ را در مختصات ۲ بعدی صفحه تصویر (صفحه ۴ اسلاید Camera Model) هر کدام از دوربین ها بر حسب f (فاصله کانونی) بدست بیاورید. سپس شمایی کلی از نحوه قرارگیری epipole و خطوط epipolar در صفحه تصویر دوربین راست و چپ رسم کنید. (۱۵ نمره)

ب) اکنون با توجه به نحوه قرارگیری دوربین ها، ماتریس Essential میان این دو دید را تشکیل دهید. (در تمرین ۳ این ماتریس معرفی شد). (۱۰ نمره)

ج) درنهایت با استفاده از رابطه معرف ماتریس E که برای مختصات نرمال شده تعریف شده است، معادله کلی خطوط epipolar راست و چپ را بدست بیاورید. (۵ نمره)

### تمرین پنجم: (امتیازی) (۲۰ نمره)

در یک ساختار stereo مختصات جهان بیرون بر مختصات دوربین چپ کاملاً منطبق است و مختصات دوربین راست از جابجایی و دوران نسبت به دوربین چپ با  $t$  و  $R$  بدست می آید. ( $M_l = K_l[I : 0]$  و  $M_r = K_r[R : t]$ )

الف) با فرض داشتن ماتریس Essential میان این دو دید و با تجزیه SVD این ماتریس نشان دهید که میتوان چهار حالت مختلف برای ترکیب دوتایی  $R$  و  $t$  متصور شد. عبارت  $R$  و  $t$  در هر چهار حالت را بنویسید. (۱۰ نمره)

ب) شمایی کلی از نحوه قرار گیری ۲ دوربین نسبت به یکدیگر در ۴ حالت بدست آمده رسم کنید و حالت درست را مشخص کنید. (۵ نمره)

ج) اگر با استفاده از روش های بازسازی سه بعدی، برای نقاط متناظر بدست آمده از دو تصویر چپ و راست، مختصات سه بعدی این نقاط را در جهان بیرون برای هر ۴ حالت تخمین بزنیم، چطور میتوانیم با داشتن این اطلاعات حالت قرارگیری صحیح دو دوربین نسبت به یکدیگر را تشخیص دهیم؟ ( $R$  و  $t$  صحیح را انتخاب کنیم) (۵ نمره)

۱۰ + ۱۶۰ نمره

### سوالات عملی

### تمرین اول: (۱۰۰ نمره)

در این تمرین میخواهیم با کالیبراسیون دوربین های استریو و کاربردهای آن آشنا شویم. (استفاده از منابع اینترنتی جهت یادگیری مفاهیم و نحوه کار با توابع – و نه صرفاً کپی کردن بدون تفکر! – در این تمرین توصیه میشود).

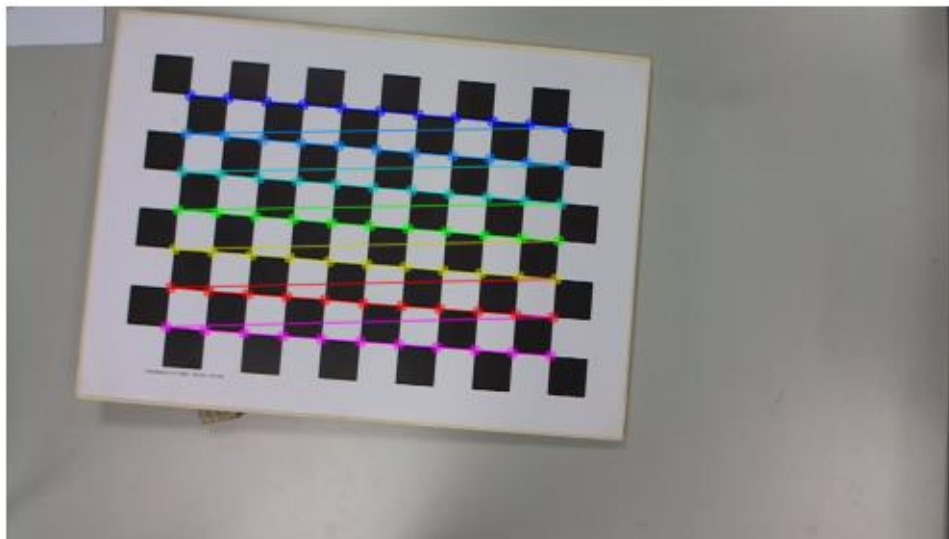
۱. تا به اینجا در درس، از مدل دوربین pinhole ایده آل استفاده کردیم، اما امروزه در عمل از این دوربین ها به ندرت استفاده میشود و دوربین های لنزدار جایگزین آن ها شده اند! تحقیق کنید که دوربین pinhole چه مزایا و معایبی نسبت به دوربین های لنزدار دارد. (۵ نمره)

۲. یکی از معایب استفاده از لنز در عکسبرداری، اضافه شدن **distortion** ناخواسته به تصاویر است یعنی خطوطی که در واقعیت مستقیم هستند در تصویر دچار خمیدگی میشوند. بنابراین در هنگام کالیبراسیون دوربین نیاز است این عوامل نیز مدل شوند. درباره انواع **distortion** پیاده سازی شده در OpenCV برای کالیبراسیون دوربین تحقیق کنید و روابط تبدیل مختصات جهان به مختصات پیکسلی را با حضور **distortion** بازنویسی کنید. (راهنمایی: ۳ نوع **distortion** در OpenCV نظر گرفته شده است). (۱۰ نمره)

۳. کالیبراسیون دوربین های استریو به صورت مستقیم با تابع **stereoCalibrate** در OpenCV پیاده سازی شده است. اما از آنجا که در این روش تمامی ماتریس ها به صورت همزمان تخمین زده میشوند برای **جلوگیری از انباشتگی خطا** بهتر است ابتدا هر دوربین به صورت جداگانه و دقیق کالیبره شود و سپس با فرض ثابت نگه داشتن پارامترهای داخلی، پارامترهای خارجی دو دوربین نسبت به یکدیگر و ماتریس های  $E$  و  $F$  تخمین زده شوند. پس در گام اول نیاز است دوربین راست و چپ را به صورت جداگانه کالیبره کنیم. به همین منظور تصاویر مختلفی از یک صفحه شطرنجی توسط دو دوربین stereo در زوایای مختلف (با فرض ثابت بودن دوربین ها و جابجایی صفحه شطرنجی) گرفته شده است. این تصاویر در دو پوشه **leftcamera** و **rightcamera** در پوشه Q1 در اختیار شما قرار گرفته اند. برای آشنایی با شیوه کلی کالیبراسیون تک دوربین به آموزش OpenCV مراجعه کنید.<sup>۱</sup> در پیاده سازی به چند نکته توجه کنید:

<sup>۱</sup>[https://docs.opencv.org/4.x/dc/dbb/tutorial\\_py\\_calibration.html](https://docs.opencv.org/4.x/dc/dbb/tutorial_py_calibration.html)

- اندازه هر مربع صفحه شطرنجی 30\*30 mm است.
- سائز الگو شطرنجی (11,7) میباشد.
- مبدا جهان بیرون در تمام تصاویر گوشه بالا چپ صفحه شطرنجی است به طوری که محور X افقی و محور Y عمودی است. نقاط پیدا شده در نهایت باید به فرم زیر باشند:



حال مراحل زیر را برای هر دوربین طی کنید: (با توجه به در اختیار بودن آموزش OpenCV از توضیحات جزئی در مراحل زیر استفاده نشده است. برای جزئیات بیشتر به لینک داده شده مراجعه کنید.)

A. ابتدا با استفاده از توابع `cv2.findChessboardCorners` و `cv2.cornerSubPix` مختصات گوشه های صفحه شطرنجی را برای تمام تصاویر پوشه پیدا کرده و در آرایه ای ذخیره کنید. آرایه مختصات گوشه ها در جهان بیرون را نیز با توجه به اطلاعات مسئله تشکیل دهید.

B. تابعی بنویسید که خروجی آن دیکشنری از پارامترهای تخمین زده شده دوربین است. درون این تابع ابتدا با دستور `cv2.calibrateCamera` دوربین را کالیبره کنید. با استفاده از خروجی های `rvecs` و `tvecs` و به کمک تابع `cv2.Rodrigues` ماتریس پارامترهای خارجی را برای هر تصویر محاسبه کنید. همچنین به کمک تابع `cv2.getOptimalNewCameraMatrix` ( $\alpha=1$ ) ماتریس پارامترهای داخلی مورد نیاز برای رفع `distortion` و ROI قابل قبول در تصاویر را نیز بدست بیاورید. در نهایت بردارهای `rvecs` و `tvecs`، ماتریس پارامترهای خارجی تصاویر، ماتریس پارامترهای داخلی دوربین قبل و بعد از رفع `distortion` و ناحیه ROI را درون دیکشنری دوربین ذخیره و برگردانید.

C. برای ارزیابی کیفیت کالیبراسیون دوربین میتوان با استفاده از پارامترهای تخمین زده شده در مرحله قبل و به کمک تابع `cv2.projectPoints` نقاط صفحه شطرنجی را از جهان بیرون بر روی صفحه تصویر دوربین انداخت و نتیجه را با نقاط اصلی (که در قسمت A محاسبه شده بود) مقایسه کرد. در حالت ایده آل نقاط اصلی و نقاط re-project شده باید کاملاً منطبق باشند! اما در عمل این حالت ممکن نیست. به همین دلیل خطای جذر میانگین مربعات (RMSE) بین نقاط اصلی و نقاط re-project شده معیار خوبی برای ارزیابی کیفیت کالیبراسیون است. طبیعی است این عدد هرچه به صفر نزدیکتر باشد، کالیبراسیون دقیق تر بوده است و معمولاً مقادیر زیر ۱ قابل قبول هستند. تابعی بنویسید که معیار RMSE را برای هر تصویر

محاسبه کرده و در آرایه rmsePerView ذخیره کند. این تابع باید RMSE کلی میان خطاهای محاسبه شده برای هر تصویر را نیز محاسبه و تک مقدار rmseTotal را هم ذخیره کند و در نهایت این دو متغیر را برگرداند. این مقادیر را نیز در دیکشنری پارامترهای دوربین ذخیره کنید. (راهنمایی: تابع cv2.calibrateCamera علاوه بر پارامترهای دوربین مقادیر rmseTotal و rmsePerView را نیز برمیگرداند. برای بررسی صحت روش محاسبه خود میتوانید مقادیر بدست آمده را با خروجی های این تابع تطبیق دهید.)

(۲۰ نمره)

اکنون موارد خواسته شده زیر را انجام دهید:

الف) برای اطمینان از درستی روش خود در قسمت A با استفاده از تابع cv2.drawChessboardCorners گوشه های پیدا شده برای تصویر شماره ۵ پوشه را نمایش دهید. (Im\_L\_05.png و Im\_R\_05.png) (۵ نمره)

ب) برای دو تصویر Im\_L\_18.png و Im\_R\_18.png به ازای  $\alpha=0$  و  $\alpha=1$  در تابع cv2.getOptimalNewCameraMatrix مقادیر ماتریس پارامترهای داخلی را حساب کنید و سپس با استفاده از تابع cv2.undistort به ازای هر دو حالت، تصویر اصلاح شده را نمایش دهید. درباره پارامتر  $\alpha$  و تاثیر آن در تصویر خروجی تابع undistort تحقیق کنید. (۵ نمره)

ج) ماتریس پارامترهای داخلی قبل و بعد از رفع distortion، ضرایب distortion و مقادیر RMSE بدست آمده را برای هر دو دوربین پرینت کنید. (۱۰ نمره)

د) در این قسمت میخواهیم با استفاده از نمایش سه بعدی کتابخانه matplotlib موقعیت کلی صفحه شطرنج و دو دوربین را برای تصاویر (Im\_L\_18.png و Im\_R\_18.png) بر مبنای مختصات جهان بیرون (بر روی صفحه شطرنج) نشان دهیم. به همین منظور تابعی بنویسید که مکان یک نقطه در مختصات سه بعدی دوربین را گرفته و با توجه به ماتریس پارامترهای خارجی آن صحنه (که در قسمت B محاسبه شده است) مکان آن نقطه در مختصات سه بعدی جهان بیرون را محاسبه کند. با استفاده از این تابع مکان مبدا دوربین های راست و چپ (C) و سه نقطه روی محورهای مختصات آن ها به فاصله ۳۰ میلی متر (X\_camera, Y\_camera, Z\_camera) را از مختصات محلی دوربین به مختصات جهان بیرون منتقل (C\_right/left\_world, X\_right/left\_world, Y\_right/left\_world, Z\_right/left\_world) و مقادیر را پرینت کنید. سپس با استفاده از قطعه کد زیر موقعیت صفحه شطرنج و مرکز دوربین ها را در سه بعد نمایش دهید. (در کد زیر objp مختصات گوشه ها در جهان بیرون هستند که در ابتدا ایجاد کردیم.) (۱۰ نمره)

```
from mpl_toolkits import mplot3d

X = objp[:,0].reshape((BOARD_SIZE[1],BOARD_SIZE[0])) # BOARD_SIZE = (11, 7)
Y = objp[:,1].reshape((BOARD_SIZE[1],BOARD_SIZE[0]))
Z = objp[:,2].reshape((BOARD_SIZE[1],BOARD_SIZE[0]))

fig = plt.figure()
ax = plt.axes(projection='3d')
ax.contour3D(X, Z, Y, 49, cmap='viridis')

colors = ['red', 'blue', 'green']
```

```

for i, point in enumerate([X_left_world, Y_left_world, Z_left_world]):
    ax.plot([C_left_world[0], point[0]], [C_left_world[2], point[2]],
            zs=[C_left_world[1], point[1]], color=colors[i])

for i, point in enumerate([X_right_world, Y_right_world, Z_right_world]):
    ax.plot([C_right_world[0], point[0]], [C_right_world[2], point[2]],
            zs=[C_right_world[1], point[1]], color=colors[i])

ax.set_xlabel('x')
ax.set_ylabel('z')
ax.set_zlabel('y')
ax.set_zlim(-50, 250)
ax.zaxis.labelpad=-1
ax.invert_zaxis()

```

۴. پس از کالیبره کردن هر کدام از دوربین ها به صورت جداگانه، حال نوبت کالیبراسیون دو دوربینه است. با توجه به محاسبه ماتریس پارامترهای داخلی دوربین ها در بخش قبل، در این بخش تنها  $R$  و  $T$  دوربین راست نسبت به دوربین چپ و ماتریس های  $E$  و  $F$  محاسبه میشوند. به همین منظور مراحل زیر را طی کنید:

A. تابعی بنویسید که با گرفتن مختصات گوشه های تمام تصاویر چپ و راست و پارامترهای دوربین چپ و راست، دوربین های stereo را کالیبره کند. خروجی این تابع دیکشنری از ماتریس های  $E$ ،  $F$ ، ماتریس  $R$ ، بردار  $t$  و خطای RMSE حاصل از Re-Projection نقاط است. از تابع `cv2.stereoCalibrate` با ورودی پارامترهای داخلی ورودی این تابع همان ماتریس های خروجی تابع `cv2.getOptimalNewCameraMatrix` هستند. (خطای Re-Projection خروجی تابع `cv2.stereoCalibrate` میباشد و نیازی به محاسبه جدا نیست). (۵ نمره)

B. خروجی های تابع خود را پرینت کنید. (۵ نمره)

C. برای تصاویر ۱۸ هر دو پوشه، ابتدا با تابع `cv2.undistort` تصاویر را تصحیح کنید. حال با تابع `cv2.undistortPoints` نقاط گوشه هایی که در تصاویر قبلا پیدا کرده بودید را نیز تصحیح کنید. با استفاده از تابع `cv2.computeCorrespondEpilines` خطوط epipolar را برای هر کدام از تصاویر بدست بیاورید. سپس تابعی بنویسید که این خطوط و نقاط متناظر دو تصویر را گرفته و این نقاط و خطوط را رسم کند. در نهایت تصاویر اصلاح شده به همراه نقاط گوشه ها و خطوط epipolar را نشان دهید. برای آشنایی بیشتر با این مبحث میتوانید به آموزش OpenCV مراجعه کنید. (۱۰ نمره)

D. همانطور که در بخش تئوری تمرین دیدید، یکی از کاربردهای اساسی دوربین های stereo تشخیص عمق است. همچنین دیدید که برای تخمین عمق از disparity بین تصاویر دوربین هایی که محورهای مختصات آن ها با یکدیگر موازی هستند (بدون دوران) استفاده میشود. در عمل دوربین هایی که برای این امر استفاده میشوند لزوما در حالت موازی قرار ندارند و یا پس از تخمین پارامترهای آن ها، به حالت ایده آل نمیرسیم! لذا لازم است با توجه به اطلاعاتی که از موقعیت دو دوربین بدست آورده ایم، صفحه تصویر آن ها را جوری در فضا جابجا و scale کنیم که در نهایت به دو صفحه تصویر موازی برسیم. این عملیات



Stereo Rectification نام دارد. تحقیق کنید که در حالت کالیبره بودن دوربین ها فرآیند این کار چگونه است و از چه تبدیل هایی روی صفحه تصویر دو دوربین استفاده میشود. (۵ نمره)

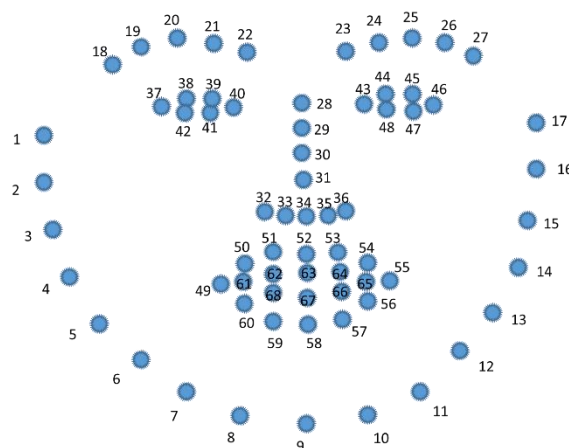
E. با استفاده از توابع cv2.stereoRectify، cv2.initUndistortRectifyMap و cv2.remap تصاویر ۱۸ دو پوشه را اصلاح کنید و نتیجه نهایی را نشان دهید. (۱۰ نمره)

#### تمرین دوم: (۴۰ نمره)

می‌خواهیم به عنوان کاربردی در حوزه Augmented Reality نسخه ساده‌ای از نرم‌افزار Snapchat را پیاده‌سازی کنیم. برای این کار مراحل زیر را طی نمایید: (فایل های مربوط به این سوال در پوشه Q2 قرار دارند).

A. کدی بنویسید که تصویر را خوانده و چهره‌های موجود در آن را به کمک تابع get\_frontal\_face\_detector از کتابخانه dlib پیدا کند.

B. نقاط کلیدی چهره فرد را به کمک تابع shape\_predictor از کتابخانه dlib پیدا کنید. آرگومان ورودی این تابع فایل با نام shape\_predictor\_68\_face\_landmarks.dat خواهد بود که در ضمیمه آمده است. این تابع، همواره نقاط کلیدی ثابتی را در چهره پیدا خواهد کرد که در نقشه زیر نشان داده شده‌اند:



C. ماسکی که قرار است بر چهره فرد قرار گیرد در تصویر joker.png به چشم می‌خورد. فایل joker.csv حاوی مختصات نقاط کلیدی ماسک مذکور است که مطابق نقشه نشان داده در بند قبلی برای تصویر ماسک به دست آمده‌اند. برای هر فرد در تصویر، ماتریس هموگرافی بین ماسک و چهره فرد را به کمک تابع findHomography از کتابخانه OpenCV بیابید. اکنون به کمک ماتریس هموگرافی، ماسک را به صفحه چهره فرد منتقل کنید.

D. ماسک منتقل شده به صفحه چهره فرد را با روش مناسب روی چهره فرد قرار دهید.

(۲۰ نمره)

اکنون به سوالات زیر پاسخ دهید.

۱. چرا نقاط کلیدی را با روش‌هایی نظیر SIFT و غیره استخراج نکردیم؟ (۵ نمره)

۲. ماسک را روی تصویر 1.png قرار داده و خروجی را نمایش دهید. (۵ نمره)

۳. ماسک را روی تصویر 2.png قرار داده و خروجی را نمایش دهید. اگر همه چیز درست باشد، ماسک به خوبی روی صورت فرد قرار نخواهد گرفت! با تغییر آرگومان‌های تابع findHomography سعی کنید مسأله را برای این حالت حل کنید. (۵ نمره)

۴. ماسک را روی تصویر 3.png قرار داده و خروجی را نمایش دهید. اگر همه چیز درست باشد، ماسک حتی با تغییر با تغییر آرگومان‌های تابع findHomography نیز به خوبی روی صورت فرد قرار نخواهد گرفت! ضمن در نظر گرفتن نقشه نقاط کلیدی و نواحی چهره، راه حلی را برای رفع این موضوع پیشنهاد دهید (نیازی به پیاده‌سازی راه حل پیشنهادی نیست) (۵ نمره)

### تمرین سوم: (۲۰ نمره)

در این سوال میخواهیم با استفاده از Agglomerative Clustering از پکیج sklearn به segmentation بر روی یک تصویر بپردازیم.

۱. در مورد پارامتر Linkage تابع AgglomerativeClustering و انواع آن (Ward, Average, Complete, Single) به صورت مختصر توضیح دهید. (۵ نمره)

۲. برای تصویر موجود در پوشه Q3 با linkage نوع ward و  $n\_clusters = 2$  به segmentation بپردازید و مرز میان دو کلاستر را با استفاده از plt.contour() رسم کنید. (۵ نمره)

۳. حال تعداد کلاسترها را بین ۲ تا ۹ تغییر دهید و هر بار مرزها را رسم کنید. به ازای چه تعداد کلاستر، بهترین segmentation از لحاظ جداسازی مرز فرد موجود در تصویر از محیط صورت می‌گیرد؟ (فرد به طور کامل از محیط اطراف جدا شده باشد) (۱۰ نمره)