

# گزارش تمرین سری اول

اصول بینایی کامپیوتر

رضا اکبریان بافقی - ۹۵۱۰۰۰۶۱

ابتدا عکس را نصف می‌کنیم و سپس فیلتر گاوسی به ابعاد ۱۰ در ۱۰ را با سیگمای ۳ بر می‌داریم و از آن مشتق سوئل می‌گیریم، و سپس در عکس کانوالو می‌کنیم. برای محاسبه مقدار مؤلفه‌های تنسور ساختار نیز از همان فیلتر گاوسی قبل استفاده می‌کنیم. برای به‌دست آوردن تابع هریس  $k$  را برابر ۰.۰۶ در نظر می‌گیریم و پس از به‌دست آوردن آن و بردن مقدار آن در بازه‌ی ۰ تا ۲۵۵، آستانه ۲.۵ را برای هر دو تصویر در نظر می‌گیریم که بالاتر از آن را به عنوان نقاط کرنر در نظر می‌گیریم.

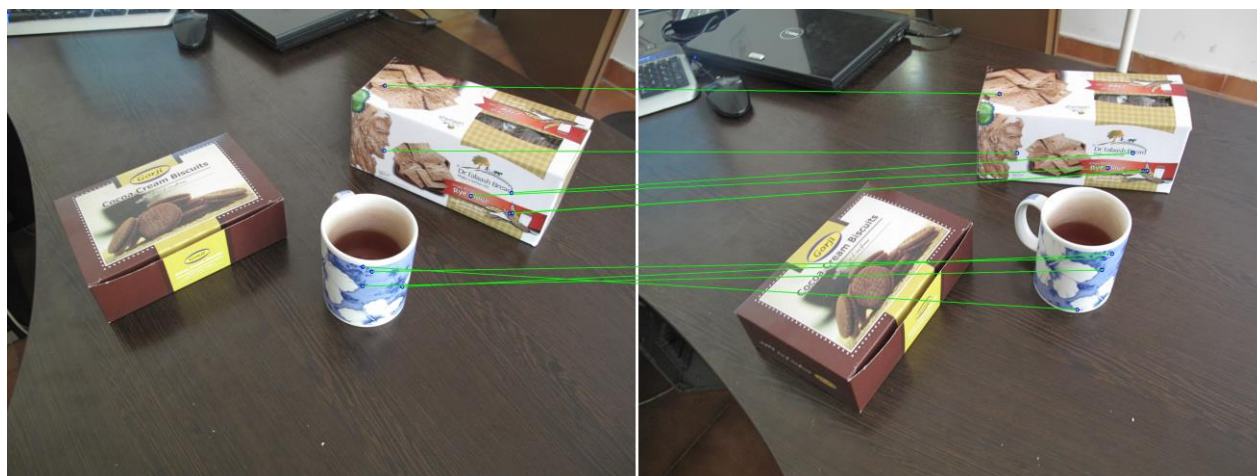
حال برای به دست آوردن نقاط  $non\text{-}maximal$ ، از تابع آماده `cv2.connectedComponents()` استفاده می‌کنیم تا بخش‌های به هم متصل را در عکسی که در آن آستانه اعمال شده را به ما بدهد. سپس در این بخش‌ها به دنبال نقطه‌ای که بیش‌ترین مقدار هریس دارد می‌گردیم. این نقاط را مشخص می‌کنیم و سپس یک بردار ویژگی به اندازه  $7 \times 7$  از نقاط همسایه آن بر می‌داریم. این بردار ویژگی را نرمال کرده و به بازه ۰ تا ۱ می‌بریم و ذخیره می‌کنیم. این مراحل ذکر شده را برای دو تصویر انجام می‌دهیم تا بردار ویژگی متناظر هر نقطه به دست بیاید. که می‌توانید این نقاط در شکل ۱ ببینید.



شکل ۱ نقاط خوب هریس در دو تصویر

حال از هر نقطه بردار ویژگی عکس اول فاصله آن را با هر نقطه بردار ویژگی عکس دوم به این صورت که اختلاف دو مقدار را به توان دو می‌رسانم و با هم جمع می‌کنم، محاسبه می‌کنم. حال دو نقطه‌ای که از تصویر دیگر کم‌ترین فاصله را با نقطه تصویر اولی داشته‌اند را در نظر می‌گیریم. اگر نسبت این دو فاصله کم‌تر از ۰.۶ باشد آن دو نقطه را به عنوان نقاط خوب در نظر می‌گیریم.

برای تصویر دوم هم همین کار را انجام می‌دهم با این تفاوت که در این مرحله بررسی می‌کنم که اگر نقطه‌ای خوب از تصویر اول برای تصویر دوم انتخاب شد، آیا قبلاً این نقطه تصویر دوم برای نقطه تصویر اول متناظر بوده است یا خیر. اگر از متناظر بوده باشد آن نقطه را ذخیره می‌کنم، در غیر این صورت این کار را انجام نمی‌دهم.



شکل ۲ نقاط متناظر در دو تصویر

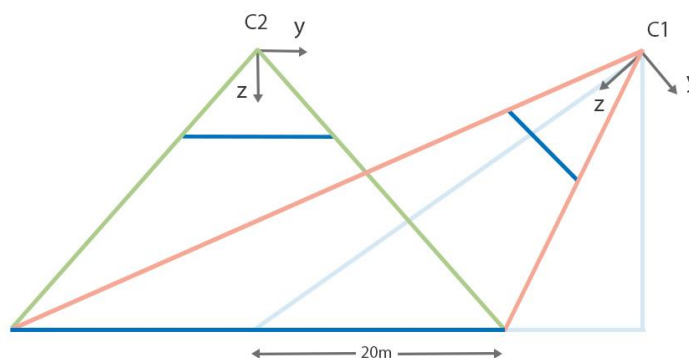
سپس در این نقاط که با هم متناظر شده‌اند به دنبال این می‌گردیم که یک نقطه با چند نقطه متناظر نشده باشد، در صورت وجود چنین نقطه‌ای، همه آن‌ها را حذف می‌کنیم. حال تعداد ۱۰ خطی که به نظر تناظر بهتری داشته باشند را انتخاب کرده و آن‌ها را رسم می‌کنیم.

## ۲

ما در این سوال یک فرضی می‌کنیم که خط گذرنده از دوربین به پایین تصویر، زمین فوتبال را در فاصله ۲۰ متری قطع می‌کند. که در شکل ۲ این فاصله مشخص شده است. سپس با استفاده از این فرض زاویه‌ها را به دست می‌آوریم. در واقع ما به دنبال این هستیم که با استفاده از رابطه زیر یک هموگرافی از دوربین اول به دوربین دوم به دست آوریم.

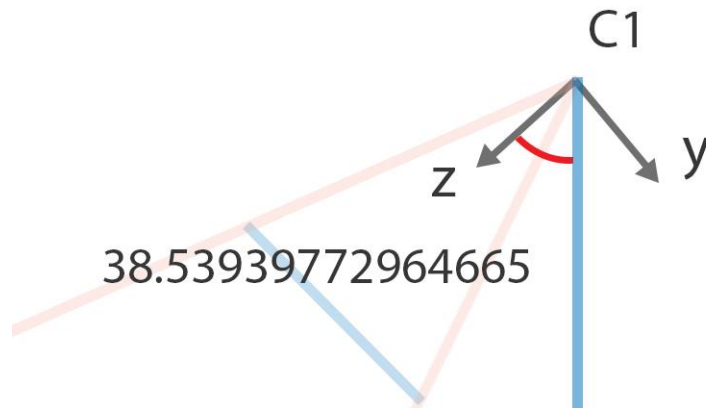
$$H = K' \left( R - \frac{tn^t}{d} \right) K^{-1}$$

که همه مختصات در این جا باید نسبت به دوربین اول نوشته شود. در شکل ۳ می‌توانید محل قرارگیری دو دوربین را نسبت به هم مشاهده کنید.



شکل ۳ محل قرارگیری دو دوربین نسبت به هم

ابتدا محاسبه می‌کنیم که دوربین C2 نسبت به دوربین C1 چه تغییری کرده است. همانطور که می‌توانیم در شکل ۳ مشاهده کنیم دوربین C2 به اندازه ۳۸,۵۴ درجه چرخش محور x هم راستا با دوربین C1 می‌شود.



شکل ۴ زاویه چرخش محور x دوربین اول نسبت به دوربین دوم

پس ماتریس R برابر مقدار زیر می‌شود:

$$R = \begin{bmatrix} \cos(38.53) & \sin(-38.53) \\ \sin(38.53) & \cos(38.53) \end{bmatrix}$$

حال همانطور که می‌دانیم باید بردار عمود بر صفحه زمین فوتبال را نسبت به دوربین C1 به دست آوریم. با به‌دست آوردن زوایا می‌توانیم سه نقطه در زمین را مختصاتشان را نسبت به مرکز مختصات C1 به‌دست آوریم. حال با استفاده از این سه نقطه معادله صفحه را به‌دست می‌آوریم که برابر زیر می‌باشد:

$$-40.3 \times Y - 295.9 \times Z + 10610.3 = 0$$

که از این‌جا می‌فهمیم که d برابر ۱۰۶۱۰.۳ و بردار n را برابر [۰. -۰.۸. -۰.۵۹] در نظر می‌گیریم. حال برای محاسبه t باید جابجایی دوربین دوم را نسبت به دوربین اول طبق مختصات دوربین اول به‌دست آوریم. می‌دانیم که  $t = -RC$  می‌باشد که C همان بردار جابجایی می‌باشد که من اینجا برابر مقدار زیر به‌دست آوردم که حاصل ضرب آن با R نتیجه مورد نظر ما را می‌دهد.

$$C = [0. \quad 24.42. \quad -19.45]$$

ماتریس کالیبراسیون دوربین اول و دوم نیز برابر مقدارهای زیر می‌باشد:

$$K_1 = \begin{bmatrix} 500 & 0 & 128 \\ 0 & 500 & 128 \\ 0 & 0 & 1 \end{bmatrix} \quad K_2 = \begin{bmatrix} 500 & 0 & 400 \\ 0 & 500 & 800 \\ 0 & 0 & 1 \end{bmatrix}$$

و ماتریس هموگرافی که از رابطه قبل داشتیم برابر مقدار زیر می‌شود:

$$h = \begin{bmatrix} 9.95721699e - 0.1 & 4.94163800e - 0.1 & 1.20027473e + 0.2 \\ -7.13050219e - 0.3 & 2.02115467e + 0.0 & 2.09293261e + 0.2 \\ -4.75366812e - 0.6 & 1.24135158e - 0.3 & 6.21518551e - 0.1 \end{bmatrix}$$

حال ماتریس هموگرافی را در تصویر در دوربین اول اعمال می‌کنیم و تصویری که به ما می‌دهد، تصویر از نگاه دوربین دوم می‌باشد. اندازه تصویر را در نهایت برابر ۸۰۰ در ۱۶۰۰ در نظر می‌گیریم.

### ۳

ابتدا تصویر ۰۴.jpg را به عنوان مرجع در نظر می‌گیریم سپس آن را به مرکز یک صفحه سیاه بزرگ‌تری انتقال می‌دهیم. سپس برای تصویر مرجع و تصاویر دیگر توسط SIFT نقاط و بردار دیکسریپتور را برای هر کدام از آن‌ها را برمی‌داریم. سپس به وسیله تابع آماده cv2.BFMatcher() نقاط خوب را در هر تصویر نسبت به تصویر مرجع انتخاب می‌کنیم. بدین صورت که اگر فاصله دو نقطه‌ای که کمترین فاصله را داشتند نسبت به هم کم‌تر از ۰,۶ باشند، نقاط خوب محسوب می‌شوند.

اگر تعداد نقاط خوب تصویر با تصویر مرجع بیش‌تر از ۲۰۰ عدد باشد ما با استفاده از آن نقاط یک هموگرافی با استفاده از RANSAC به‌دست می‌آوریم و در صورتی که این تصویر تعداد نقاط کافی برای تشکیل هموگرافی را نداشته باشد ما به دنبال تصویر سومی از بین تصاویری که هموگرافی آن‌ها را به تصویر مرجع پیدا کرده‌ایم می‌گردیم و آن تصویری که بیش‌ترین تعداد نقطه خوب را با آن داشته باشد را انتخاب می‌کنیم و سپس هموگرافی را بین آن دو محاسبه می‌کنیم. در واقع هموگرافی بین آن تصویر و تصویر مرجع را می‌توانیم با حاصل ضرب زیر محاسبه کنیم:

$$h_{31} = h_{21}h_{32}$$

که در اینجا  $h_{21}$  را از قبل داشتیم و  $h_{32}$  را هم می‌توانیم محاسبه کنیم و در نهایت  $h_{31}$  به‌دست می‌آید که تصویر ما را به تصویر مرجع می‌برد.

در این‌جا RANSAC ای که پیاده‌سازی کردم با ترشهولد ۲۰، نقاطی که داخل این ترشهولد باشند را به عنوان inlier می‌شمارد و پس از پیدا کردن هموگرافی اولیه آن‌قدر ایتريشن انجام می‌دهد تا هموگرافی ما کانورج شود.



شکل ۵ تصویر سمت چپ تصویر مرجع و تصویر سمت راست تصویری است که تعداد نقاط خوب بیشتر از ۲۰۰ تا دارد.

در مرحله **sitiching** ما می‌آییم و هموگرافی را روی تصاویر به ترتیب انجام می‌دهیم و سپس با استفاده از ماسک دو تصویر را با هم ترکیب می‌کنیم. به این صورت که در قسمت‌هایی که مشترک می‌باشد عمل میانگین‌گیری را انجام می‌دهیم سپس آن قسمتی از صفحه بزرگ اولیه را که با تصاویر پوشانده شده است را کراب می‌کنیم. تصویر کراب نشده را می‌توانید در شکل ۴ مشاهده کنید.



شکل ۶ تصویر بریده نشده از چسباندن تصاویر به هم

## ۴

ابتدا توسط تابع آماده `cv2.xfeatures2d.SIFT_create()` یک آبجکت از **SIFT** می‌سازیم و سپس نقاط را با استفاده از `sift.detectAndCompute(img1_gray, None)` به‌دست می‌آوریم. سپس با استفاده از تابع آماده مانند سوال قبل نقاط خوب را پیدا می‌کنیم. این بار نسبت را برای فاصله  $0.75$  در نظر می‌گیریم. یعنی دو نقطه‌ای که کم‌ترین فاصله را با نقطه مورد نظر دارد اگر نسبت فاصله‌شان کم‌تر از  $0.75$  باشد نقطه خوبی به حساب می‌آیند.

تعداد نقاطی که در این مرحله به عنوان نقاط خوب پیدا می‌کند  $73$  عدد می‌باشد. که من  $20$  عدد از این نقاط خوب را به هم وصل کردم و در تصویر **r17** ذخیره کرده‌ام.

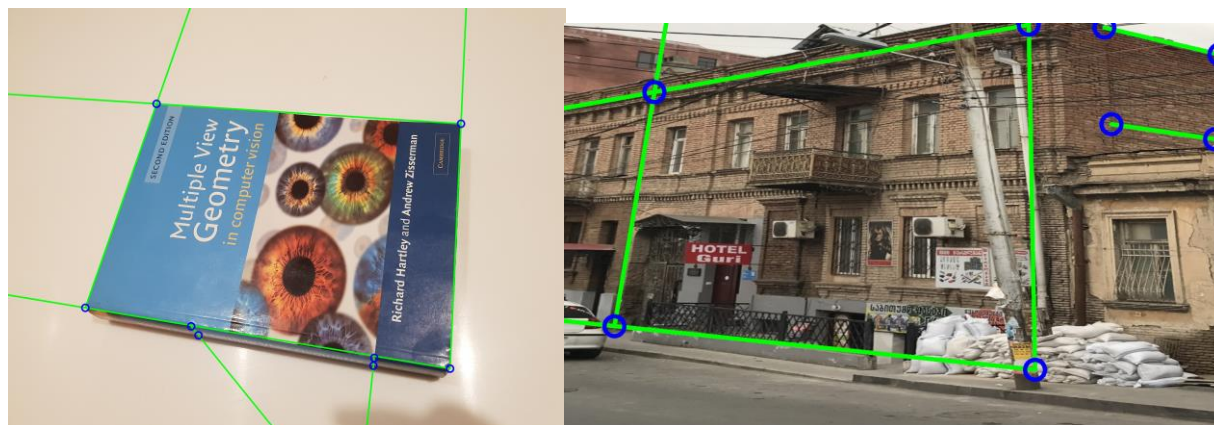
حال این جفت نقاط خوب متناظر را به **RANSAC** می‌دهیم تا بتواند نقاط **inlier** را پیدا کند. روش کار بدین صورت است که ابتدا در هر مرحله  $4$  جفت نقطه را به طور رندوم انتخاب می‌کند. از آن‌ها هموگرافی را پیدا می‌کند و در این‌جا هم ترشهولد را برابر  $20$  قرار می‌دهم. نقاطی که فاصله‌ای کم‌تر از  $20$  دارند را به عنوان **inlier** انتخاب می‌کند و  $w$  و  $N$  را آپدیت می‌کند. تابع هموگرافی‌ای که از این مرحله به‌دست می‌آید دوباره طی یک ایتريشن تا به کانورج رسیدن آن تابع ادامه می‌دهد.

**۱۷** نقطه به عنوان **inlier** در نهایت به‌دست می‌آید که ما از آن‌ها یک تابع هموگرافی به‌دست آورده‌ایم. آن را روی تصویر اعمال می‌کنیم و سپس نقاط سیاه را از تصویر می‌بریم.

از این نقاط **inlier** ای که به‌دست آمده است، هیچ کدام **outlier** نمی‌باشد و همه درست انتخاب شده‌اند.



در این سوال ابتدا ونیشینگ پوینت‌ها را مشخص می‌کنیم. که  $A$  و  $B$  دو ونیشینگ پوینت افقی و  $C$  ونیشینگ پوینت عمودی می‌باشد. می‌خواهیم با استفاده از این سه ونیشینگ پوینت ماتریس کالیبریشن دوربینی که این عکس با آن گرفته شده است را به دست آوریم. خطی که  $A$  و  $B$  را به هم وصل می‌کند می‌تواند بگوید دوربین چقدر کج و جابجا شده است.



شکل ۷ نقاط آبی نقاطی است که به صورت دستی انتخاب کردم و خطوط سبز تقاطع‌شان باعث ایجاد ونیشینگ پوینت می‌شود.

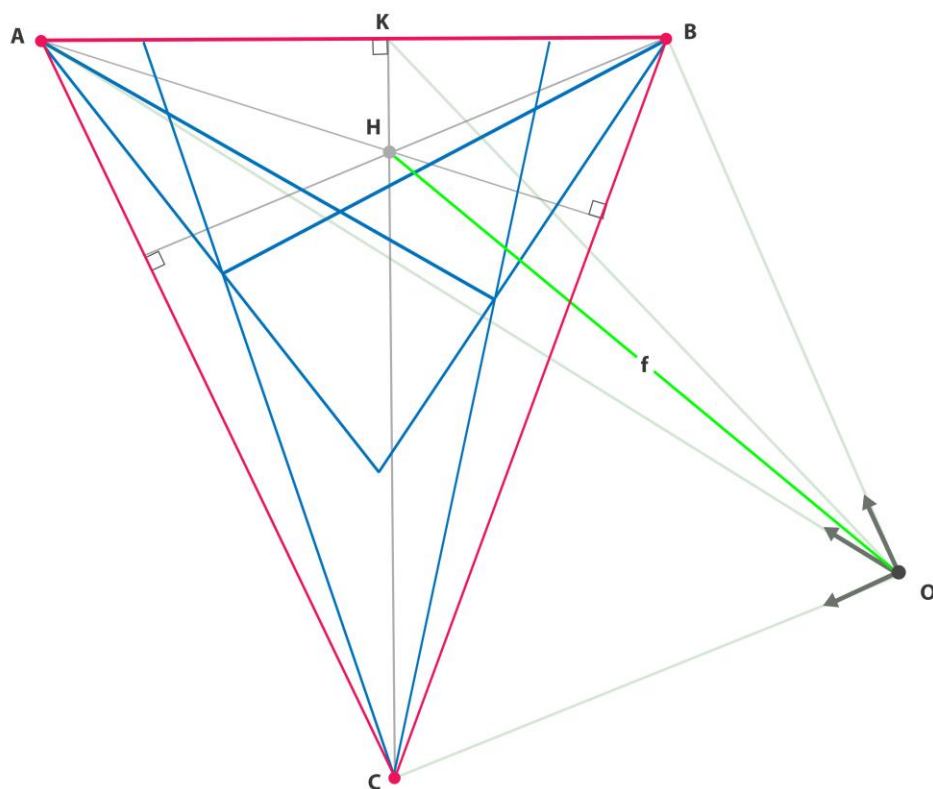
چون این سه نقطه‌ای که به عنوان ونیشینگ پوینت به دست می‌آوریم از سه دسته خطوط موازی با هم که بر هم عمود هستند به دست آورده شده است، پس همانطور که در شکل مشخص شده است، زمانی که از  $O$  به  $A$ ،  $B$  و  $C$  خطی وصل می‌کنیم، این سه خط بر هم به صورت دو به دو عمود هستند. در واقع ما در این جا  $O$  را مکان دوربین در نظر گرفته‌ایم که از آن جا عمل پروژکشن را انجام می‌دهیم. حال می‌خواهیم فاصله  $O$  از صفحه  $ABC$  را به دست آوریم.

می‌دانیم که فاصله  $O$  از صفحه  $ABC$  کوتاه‌ترین خط گذرنده از صفحه  $ABC$  می‌باشد. بنابراین این خط باید بر صفحه  $ABC$  عمود باشد. طبق قضیه‌ای می‌دانیم این خط از محل تقاطع عمودهای مثلث  $ABC$  می‌گذرد. پس مکان تقاطع عمودهای مثلث  $ABC$  مرکز تصویر و فاصله این نقطه تا  $O$  فاصله کانونی را مشخص می‌کند. در شکل زیر می‌توانید موقعیت  $O$  نسبت به  $A$ ،  $B$  و  $C$  را مشاهده فرمایید.

حال باقی مراحل برای به دست آوردن طول  $OH$  آسان است. می‌دانیم مثلث  $OAB$  قائم‌الزاویه می‌باشد، پس در آن  $OK^2$  برابر  $AK * KB$  می‌باشد و در آخر  $OH = \sqrt{OK^2 - HK^2}$  به دست می‌آید.

حال با استفاده از مختصات  $H$  و طول  $OH$  که برابر فاصله کانونی دوربین می‌شود می‌توانیم دوربین‌مان را کالیبره بکنیم. این کالیبراسیون برای عکس کتاب برابر مقدار زیر شد:

$$K_{\text{کتاب}} = \begin{bmatrix} 3758.2 & 0 & 1767.3 \\ 0 & 3758.2 & 1588.2 \\ 0 & 0 & 1 \end{bmatrix}$$

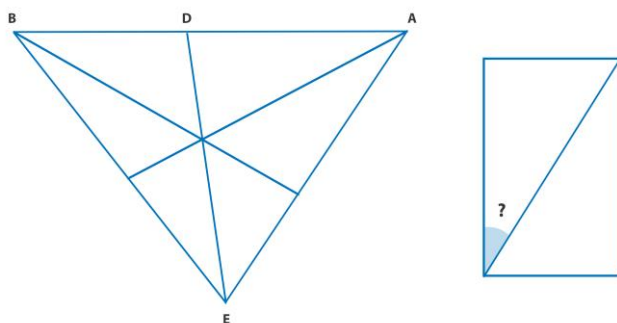


شکل ۸ موقعیت دوربین و نقاط ونیشینگ

حال می‌توانیم با استفاده از ماتریس کالیبراسیون، ماتریس  $\omega$  را به دست بیاوریم. می‌دانیم که  $\omega = (KK^T)^{-1}$  است. با استفاده از  $\omega$  می‌توانیم زاویه بین دو دسته از خطوط موازی متناظر با دو ونیشینگ پوینت را به دست بیاوریم. در واقع برای دو ونیشینگ پوینت  $v_1$  و  $v_2$  داریم:

$$\cos(\theta) = \frac{v_1^t \omega v_2}{\sqrt{(v_1^t \omega v_1)(v_2^t \omega v_2)}}$$

حال همان‌طور که در شکل ۹ مشخص است زاویه خطوط متناظر بین دو ونیشینگ پوینت B و D را به دست می‌آوریم. که همان‌طور که در شکل ۹ مشخص است. زاویه متناظر در مستطیل را می‌دهد.



شکل ۹ با استفاده از رابطه بالا زاویه واقعی این قطر مستطیل با ضلعش به دست می‌آید.



حال با استفاده از این زاویه می توان نسبت واقعی دو ضلع مستطیل در عکس کتاب و هتل به دست آورد.

برای کتاب این نسبت برابر  $۰,۷۴۱$  و برای هتل این نسبت برابر  $۲,۸۹۵$  می باشد.

برای این سوال از کتاب **Multiple View Geometry in Computer Vision** و از کورس آنلاین **How to Compute Intrinsic from Vanishing Points** از سایت coursera استفاده کرده ام.