

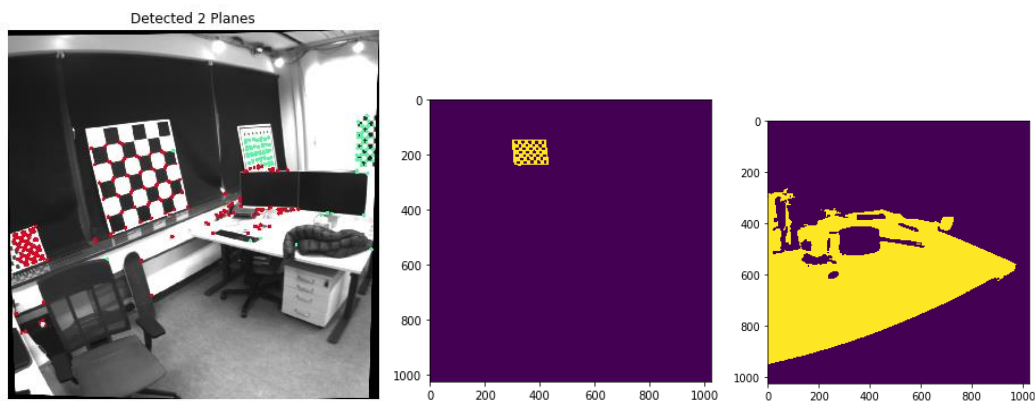
به نام خدا

آرزو رحمتی سلطانتلی ۴۰۳۱۶۶۲۰۹۲

تسک اول:

- سگمنت و هموگرافی انتهایی:

با توجه به وجود پترن های تکرار شونده ی فراوان در دیتاست، تصمیم گرفتیم اول کیفیت را در اولویت قرار دهیم و ببینیم چگونه میتوانم دقت خوبی داشته باشیم. برای همین برای هر فریم بعد از اصلاح اعوجاج ها، یک segmentation پردازش تصویری انجام دادیم. بدین صورت که ابتدا کنتراست تصویر را بهبود داده و با فیلتر گوسی هموار می کنیم. سپس با یک استانه گذاری آن را باینری کرده و با عملیات مورفولوژی نویز ها را از بین برده و connected component ها را بصورت یک شی در نظر گرفتیم. با این روش حدودا ۳۵ تا ۴۰ شی در تصویر بدست آوردیم و با یک استانه گذاری روی تعداد پیکسل های شی (که از استانه کمتر نباشد) به زیر ۱۰ سگمنت رسیدیم. پس از آن این سگمنت ها را با سگمنت های بدست آمده از فریم دوم AND کرده و هرکدام بیشتر تعداد پیکسل اشتراک را داشت بعنوان متناظر آن انتخاب شد (که مچ ها را تقریبا ۱۰۰ در ۱۰۰ درست بدست آورد) سپس برای هر دو mask بدست آمده در دو تصویر متوالی یک هموگرافی محاسبه کردیم (با همان روش استخراج ویژگی با ORB و مچ کردن و پیدا کردن بهترین نقاط مچ و سپس محاسبه ی هموگرافی). تا این لحظه برای هر دو فریم کمتر از ۱۰ هموگرافی برای قسمت های متناظر بدست آوردیم و آنها را با هم مقایسه کردیم و اگر کمتر از یک مقدار استانه با هم اختلاف داشتند (در یک صفحه بوده اند)، یکی را حذف کرده و میانگین آن دو را قرار دادیم. در این مرحله هم چند هموگرافی دیگر حذف شده و تا اینجا معمولا بین ۲ تا ۵ هموگرافی (صفحه متفاوت) برای ما باقی ماند. در انتها ویژگی ها را برای هر دو صفحه بدست آورده و مشخص کردیم هر ویژگی با کدام هموگرافی متناظر است و همه ی نقاط مربوط به آن هموگرافی را به یک رنگ کشیدیم. ویدیو این روش با نام maskH.mp4 و کد این روش با نام seg2.py پیوست شده است. نمونه ای از خروجی segment و خروجی انتهایی برای تشخیص plane ها: تصاویر بیشتر از یک قسمت خاص انتخاب شده است تا متوجه شویم توانسته دو صفحه تمایز مقابل و چپ را درست تشخیص دهد یا خیر

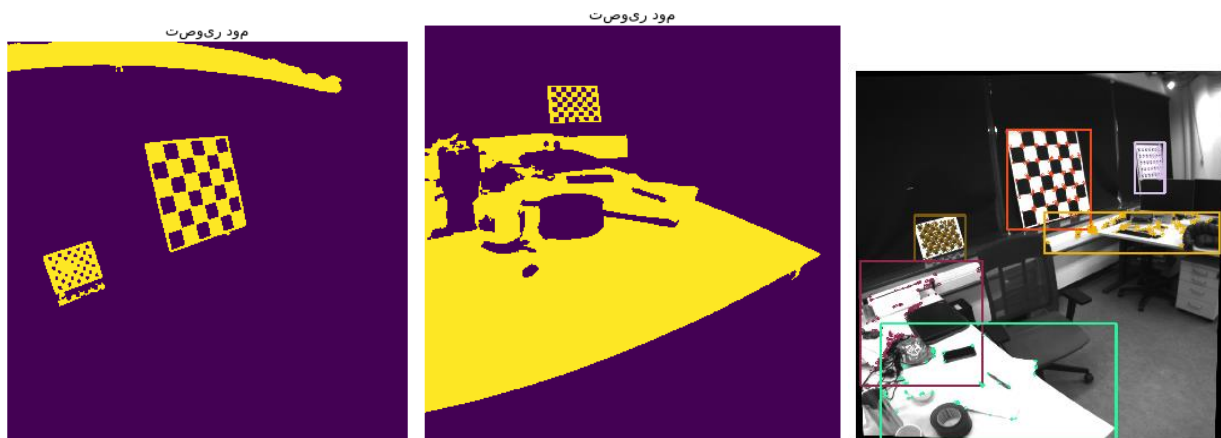




- سگمنت بدون هموگرافی انتهایی

همانطور که مشخص است این روش بسیار کند است و سعی کردم روش را بصورت زیر تغییر دهم تا زمان (و شاید کیفیت بهتر شود) در این روش مرحله ی اخر یعنی نسبت دادن هر ویژگی به یکی از هموگرافی های بدست آمده را نداریم و بجای آن در زمان حذف هموگرافی های نزدیک به هم mask های آن ها را با هم ادغام می کردم و سپس برای نمایش ویژگی ها، فریم و هر کدام از این mask ها را داده تا ویژگی های متناظر با آن با رنگ مشخص در تصویر کشیده شود.

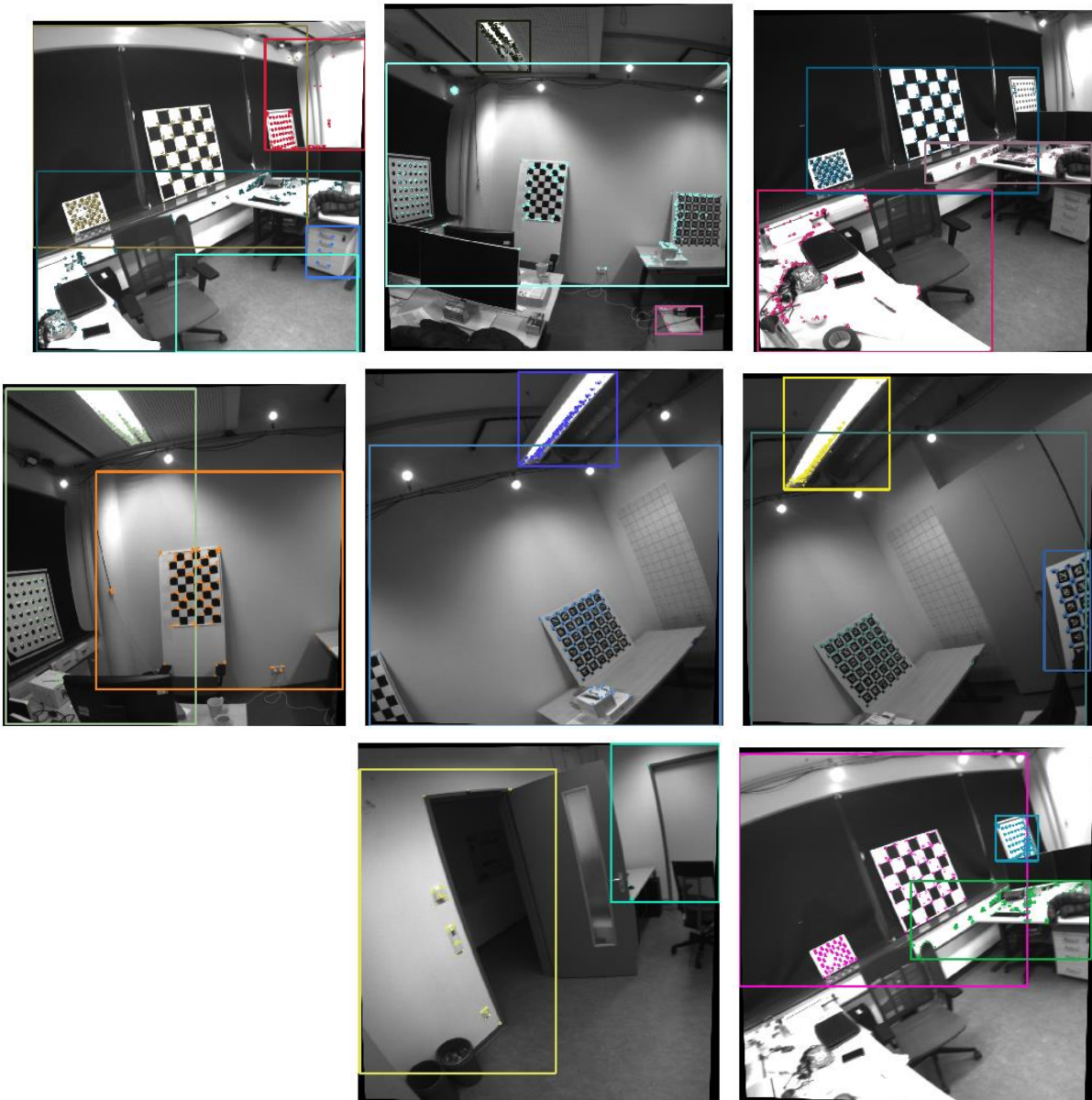
نمونه ای از تصاویر و mask های ادغام شده در زیر آمده است:



سه تا سگمنت بدلیل نزدیک بودن هموگرافی یک صفحه تشخیص داده شدند

نمونه ای دیگر از ادغام سگمنت ها

خوب ادغام نشده و جدا جدا نشان داده شده



این روش نیز مانند روش قبلی کند است ولی از قبلی سریعتر است. فایل ویدیو مربوط به این کد با نام maskWOH.mp4 و کد با نام segWithH.py پیوست شده است.

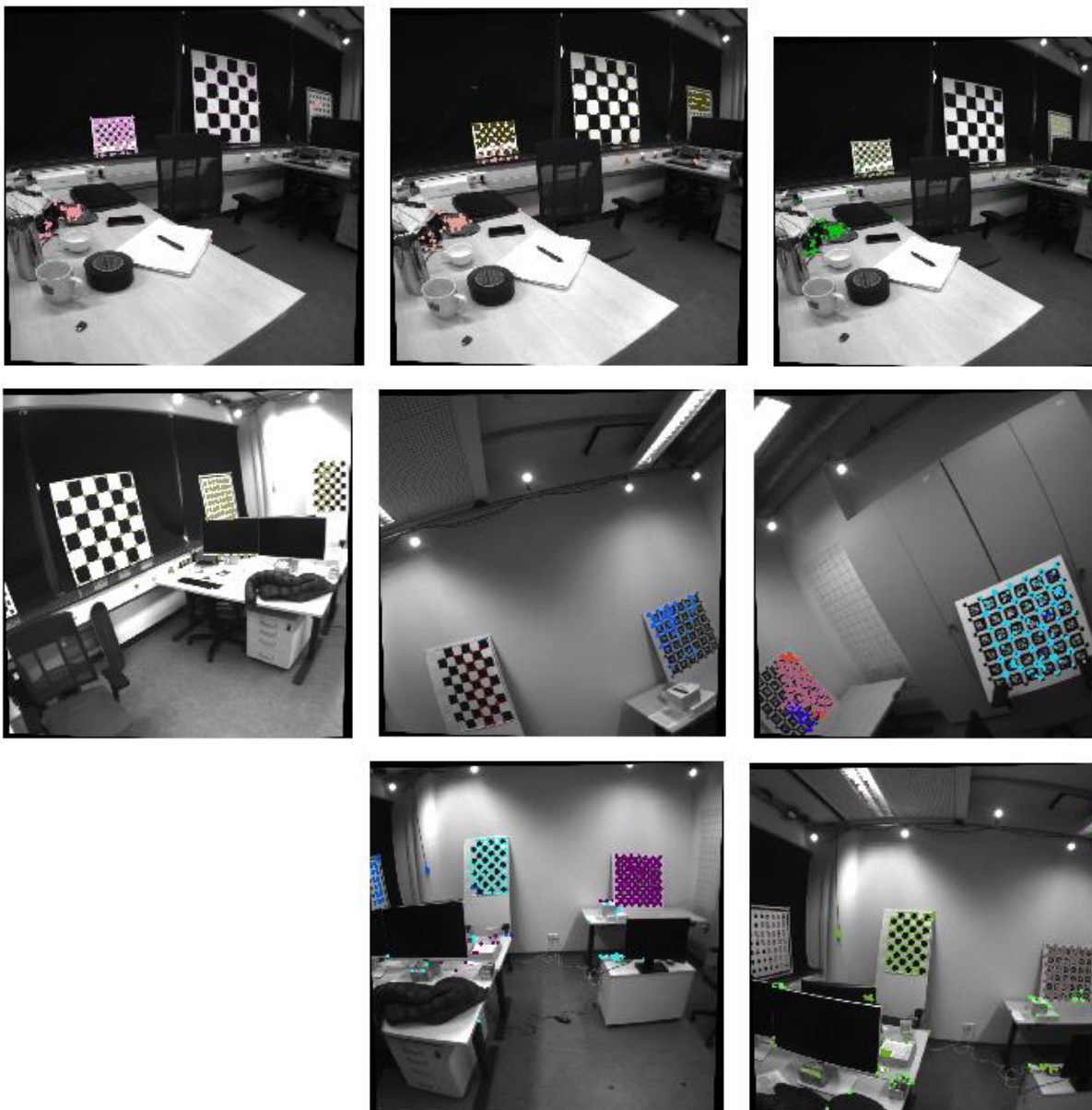
نسبت به روش قبلی بهتر عمل کرده است. این دو روش برای بدست آوردن قسمت هایی که ویژگیهای زیادی ندارند مثل دیوار و زمین مناسب اند:



- GoodFeaturesToTrack + KLT

روشی که در متن صورت سوال تمرین آمده است نیز پیاده شده است. برای بالا بردن سرعت و همچنین بدلیل داشتن پترن های تکرار شونده از روش GoodFeaturesToTrack + KLT استفاده شده است که بجای مچ کردن ویژگی ها، از optical flow تصویر برای track کردن ویژگی ها استفاده می کند و از نقاط بدست آمده با استفاده از ransac ابتدا یک هموگرافی بدست می آورد سپس با نقاط باقیمانده دوباره همین کار را تکرار می کند تا زمانی که دیگر نشود با نقاط یک صفحه درست کرد (کمتر ۲۰) سپس نقاط مربوط به هر plane را به یک رنگ روی صفحه نمایش می دهیم. خروجی با نام KLT.mp4 و کد با نام KLTF.py پیوست شده است.

این روش بسیار سریع بود. نمونه ای از این تصاویر این روش را در زیر میبینید:



تسک دوم:

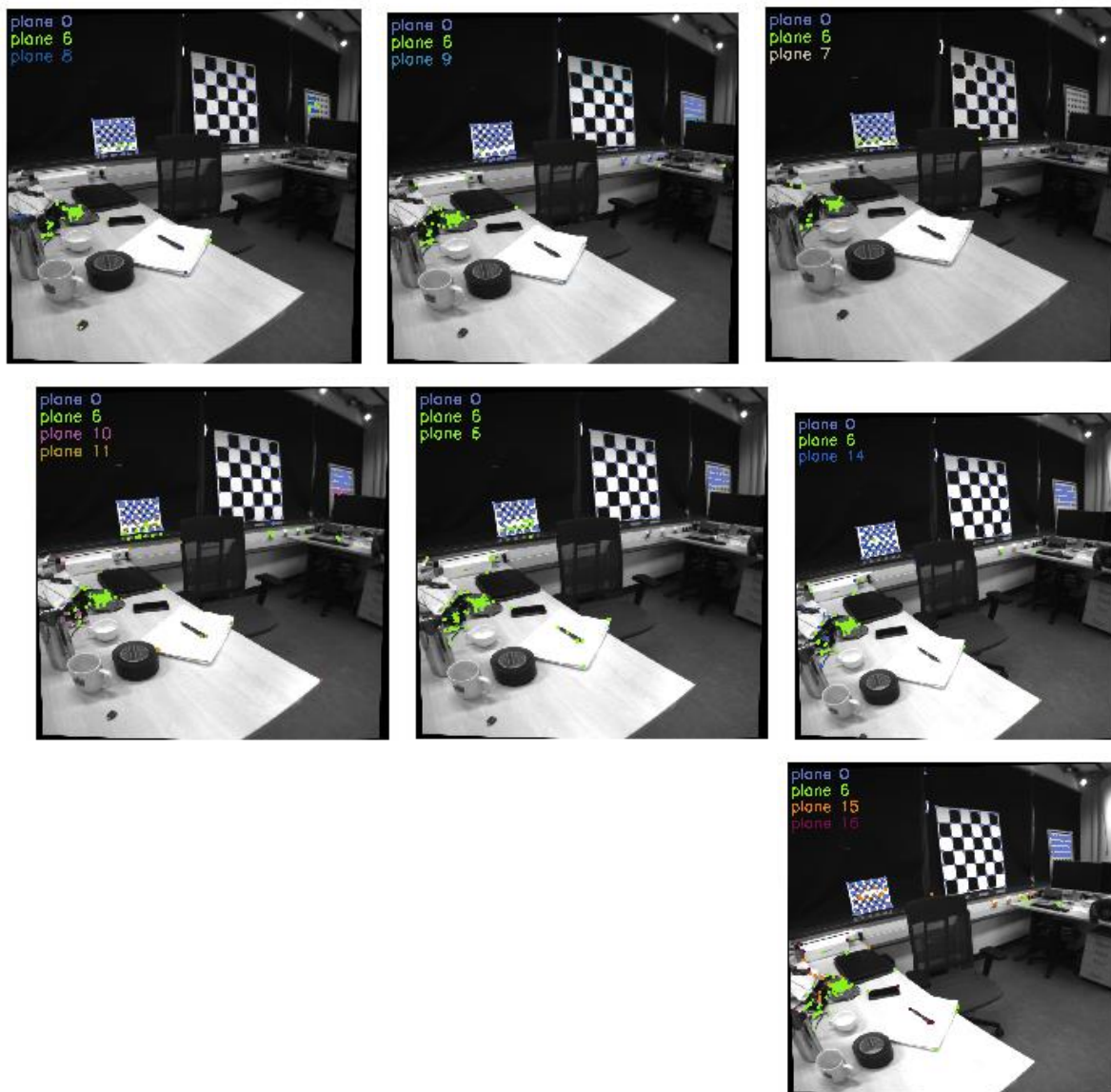
قسمت tracker را با حالت سوم یعنی KLT پیاده سازی کردم. دقت در این قسمت را دقت انتخاب درست plane ها مشخص می کند زیرا اگر plane ای اشتباه مشخص شود (دارای پیکسل های اضافه از یک صفحه ی دیگر باشد) باعث میشود بدلیل شباهت در قسمت های اشتباه، صفحه ی دوم نیز جزو صفحه ی اول شناخته شود. با توجه به اینکه در قسمت قبل و حالت سوم جواب ها کامل درست نبود این روش نیز در قسمت های میانی نتیجه ی قابل قبولی نداشته باشد در حالیکه عملکرد آن در حدود ۳۰ فریم ابتدایی خوب است. البته عملکرد مرحله ی قبل تا حدی بدلیل سختی دیتاست بوده است.

Tracker بدین صورت پیاده سازی شد که بعد از شناسایی صفحات بین دو فریم متوالی، هر کدام از صفحات جدید را با صفحات موجود در tracker مقایسه می کنیم و این مقایسه از طریق استخراج ویژگی و مچ کردن بین آنها است. صفحه ای که بیشترین مچ را با صفحه ی جدید داشته باشد و مچ ها از یک مقدار استانه بیشتر باشد، مشابه در نظر گرفته شده و رنگ آن درون رنگ چاپ قرار می گیرد و مقادیر keypoint های صفحه و یک لیست دیگر به نام recency بروز میشود که نشان میدهد آخرین فریمی که صفحه

ی مچی با این صفحه ی درون لیست داشته کدام بوده است. زیرا ما بر اساس همین recency صفحاتی که ۳۰ فریم دیده نشده باشند را دیگر در مقایسه وارد نمی کنیم. keypoint های صفحه ی جدید نیز به keypoint های قبلی صفحه الحاق میشود. ان مقدار استانه ای که باید تعداد مچ ها از ان بیشتر باشد را هم متغیر در نظر گرفتیم و در ابتدا که تعداد نقاط کم است کم و در ادامه این استانه را بیشتر می کنیم.

اگر صفحه ی مشابهی پیدا نکردیم این صفحه را به لیست صفحات خود اضافه می کنیم.

پس ما برای track کردن لیستی از صفحات ، لیستی از رنگ متناظر با صفحات و لیستی از آخرین بار دیده شدن این صفحه نگه می داریم. ایندکس ذخیره شده در هر کدام از این لیست ها را هم ID در نظر میگیریم که در کنار صفحه چاپ می شود. نمونه ای از نتایج این روش برای لحظات ابتدایی:



علت چاپ مثلا چندین plane 0 این است که روش ما در این فریم جاری ۲ تا صفحه پیدا کرده است ولی هر دوی این صفحات بیشترین اشتراک را با صفحه ی یک داشته اند.

این روش همانطور که گفته شد بسیار سریع است. ویدیو این روش با نام KLTTracking.mp4 و کد این روش با نام klt2.py پیوست شده است.