گروه كامپيوتر - دانشكده مهندسي - دانشگاه فردوسي مشهد

مستندات گزارش تمرین هفتم

درس مبانی بینایی کامپیوتر

**ارائه ‌دهنده:**

**صالح شیروانی**

**استاد درس:**

**دکتر امیرحسین طاهری نیا**

**پاییز1401**

چکیده

در این گزارش به الگوریتم های مختلف استخراج نقاط کلیدی (key points) از تصاویر می‌پردازیم. در مسائلی که در این تمرین قرار گرفته الگئریتم های Harris،SIFT و SURF را مورد بررسی قرار می‌دهیم . همانطور که می‌دانیم Harris تنها برای تشخیص نقاط کلیدی به کار می‌رود ولی SIFT و SURF علاوه بر تشخیص برای توصیف (describe) و انطباق نقاط کلیدی (matching) هم به کار می‌روند. در نهایت، نتایج حاصل از اعمال این تصاویر را خواهیم دید و درباره ی آن بحث خواهیم کرد.

Technical Description

**7- Features**

7-1- Harris corner detector

7-1-1- در این سوال به طور کلی خواسته شده که الگوریتم Harris corner detector را پیاده کنیم و نتیجه اعمال الگوریتم روی یک تصویر در 4 اسکیل متفاوت را گزارش دهیم.

با توجه به الگوریتم مطرح شده در کلاس، گام های زیر را به ترتیب پیاده می‌کنیم:

* ابتدا تصویر رنگی را به تصویر grayscale تبدیل می‌کنیم.
* روی تصویر یک فیلر گوسین اعمال می‌کنیم تا تصویر نرم شود و جزئیات اضافه از تصویر حذف گردد.
* گرادیان تصویر را در دوجهت x و y حساب کرده و آنها را به ترتیب و می‌نامیم. برای محاسبه گرادیان از ماسک سوبل استفاده می‌کنیم.
* با استفاده از مقادیر گام قبل، مقادیر ، و را محاسبه می‌کنیم.
* ماتریس M(در کد H نام دارد) را به صورت زیر محاسبه می‌کنیم:
* حال دترمینان و تریس(trace) ماتریس M را محاسبه می‌کنیم تا به وسیله ی آن مقدار R را محاسبه کنیم.
* در نهایت روی مقادیر R محاسبه شده threshold می‌زنیم تا براساس آن نقاطی که نسبت به بقیه ضعیف تر هستند حذف شوند و key point های نهایی مشخص شود.
* در نهایت این الگوریتم را روی تصویر مورد نظر در 4 اسکیل مختلف اجرا می‌کنیم.

7-2- Scene stitching with SIFT/SURF features

7-2-1- در این تمرین از ما خواسته شده تا به کمک ماژول های موجود در پایتون به بررسی الگوریتم های SIFT و SURF بپردازیم و به کمک آن ها نقاط کلیدی را یافته و correspond ها را مشخص کنیم. ما برای پیاده سازی این تمرین از OpenCV استفاده می‌کنیم. الگوریتم های SIFT و SURF در این کتابخانه به طور کامل پیاده شده است و ما تنها از آنها استفاده می‌کنیم. البته شایان ذکر است که این توابع در پیاده سازی های جدیدتر OpenCV موجود نیستند و به طور کلی از این کتابخانه حذف شده اند و ما در این تمرین از ورژن های قدیمی تر OpenCV استفاده می‌کنیم.

7-2-1-1- برای بررسی SIFT گام های زیر را طی می‌کنیم:

* ابتدا تصویر رنگی را به تصویر grayscale تبدیل می‌کنیم.
* به کمک تابع SIFT\_create در ماژول xfeatures2d موجود در OpenCV، یک شی SIFT می‍سازیم و سپس با استفاده از متد detect نقاط کلیدی تصویر را پیدا می‌کنیم.
* در نهایت به کمک متد drawKeypoints نقاط کلیدی یافته شده در گام قبل را روی تصویر نمایش می‌دهیم.

7-1-1-2- برای بررسی SURF گام های زیر را طی می‌کنیم:

* ابتدا تصویر رنگی را به تصویر grayscale تبدیل می‌کنیم.
* به کمک تابع SURF\_create در ماژول xfeatures2d موجود در OpenCV، یک شی SURF می‌‍سازیم و سپس با استفاده از متد detectAndCompute نقاط کلیدی تصویر به همراه descriptor را پیدا می‌کنیم.
* در نهایت به کمک متد drawKeypoints نقاط کلیدی یافته شده در گام قبل را روی تصویر نمایش می‌دهیم.
* برای یافتن correspond ها کافیست ابتدا الگوریتم های توضیح داده شده برای SIFT و SURF را روی تصویر اعمال کنیم و در ادامه از متد match استفاده کنیم که به صورت brute force، descriptor تصویرها را که قبلا به دست آورده ایم با هم match می‌کند و correspond ها را مشخص می‌کند. توجه کنید که برای استفاده از متد match ابتدا باید از BFMatcher موجود در OpenCV استفاده کرد و یک شی ساخت. متد match از طریق این شی قابل دسترسی است.

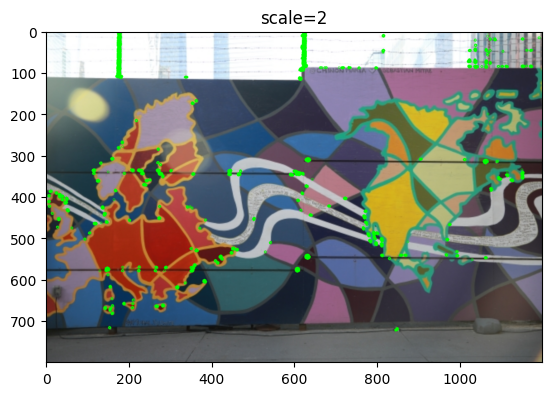
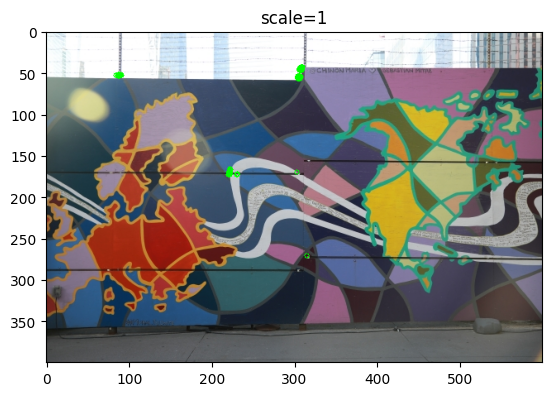
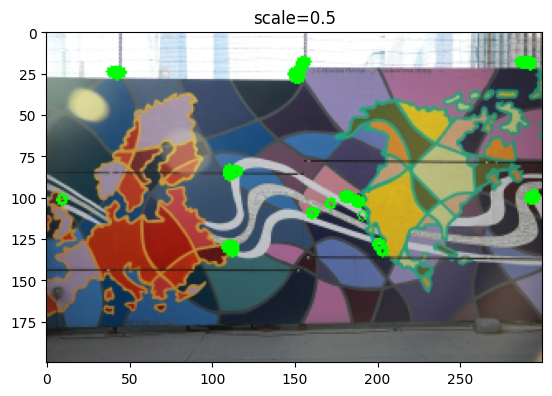
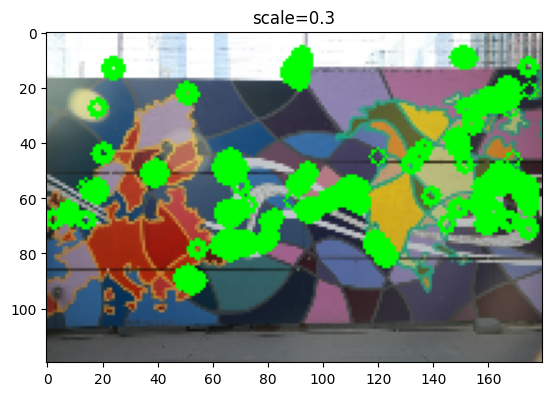
7-2-2- این تمرین عینا مثل تمرین های توضیح داده شده هستند فقط باید الگوریتم های ذکر شده را روی تصاویری که با دوربین خودمان گرفته این اعمال کنیم.

**Results**

**7- Features**

7-1- Harris corner detector

7-1-1- چهار scale برای تصویر در نظر می‌گیریم و سپس تصویر را scale می‌کنیم و الگوریتم توضیح داده شده را روی آن اجرا می‌کنیم. خروجی به شکل زیر است:

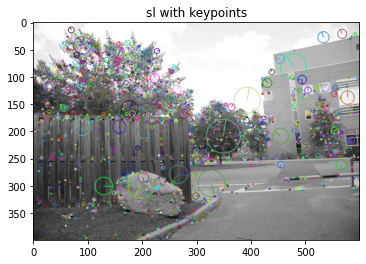


همانطور که مشاهده می‌شود نقاطی که توسط الگوریتم مشخص شده، نقاط گوشه هستند. همچنین وقتی عکس را به اسکیل های بزرگ و کوچک تر از اندازه ی اصلی resize می‌کنیم نقاط تشخیص داده شده فرق می‌کند. که انتظار چنین نتیجه ای هم می‌رفت زیرا می‌دانیم Harris به اسکیل شدن تصویر حساس است. به طور کلی وقتی تصویر scale down می‌شود تغییرات شدت پیکس ها در جهات x و y کوچکتر می‌شود و کار Harris را برای تشخیص نقاط گوشه سخت می‌کند. و به طریق مشابه وقتی تصویر بزرگ می‌شود تغییرات شدت نیز چشمگیرتر بوده و الگوریتم بهتر عمل می‌کند. همچنین وقتی تصویر کوچک می‌شود، تعداد پیکسل ها کاهش می‌یابد که می‌تواند تصویر را مستعد نویز گرداند و جزئیات را کاهش دهد و بدین وسیله روی عملکرد Harris تاثیر بگذارد. برعکس وقتی تصویر بزرگ می‌شود و تعداد پیکسل ها افزایش می‌یابد جزئیات تصویر واضح تر می‌شود که می‌تواند تشخیص گوشه ها را برای Harris آسان کند.

همچنین scale کردن تصویر می‌تواند روی کیفیت کلی تصویر نیز تاثیر بگذارد که می‌تواند عملکرد Harris را تحت تاثیر قرار دهد.

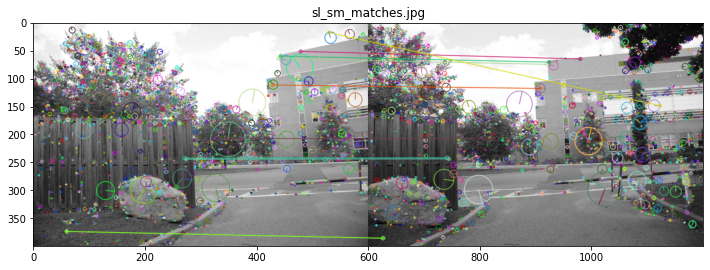
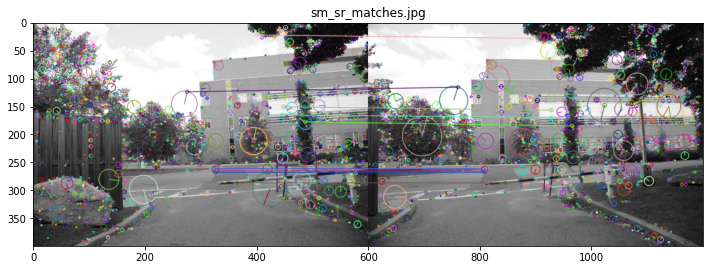
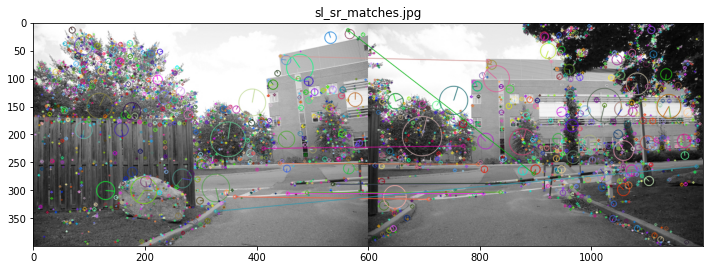
7-2- Scene stitching with SIFT/SURF features

7-2-1- key point های حاصل از اعمال SIFT روی تصویر به صورت زیر است:









*همه ی این تصاویر از یک منظره هستند و SIFT به خوبی در قسمت های مشترک عکس ها keypoint ها را به خوبی شناسایی کرده و نقاط متناظر را در دو عکس متفاوت تشخیص داده و آنها را با هم match کرده است. البته در برخی موارد هم اشتباهاتی رخ داده است که خطوطی که خطوط موازی را قطع کرده اند حاکی از این خطاهاست. یعنی الگوریتم در تشخیص نقاط متناظر دچار خطا شده است.*

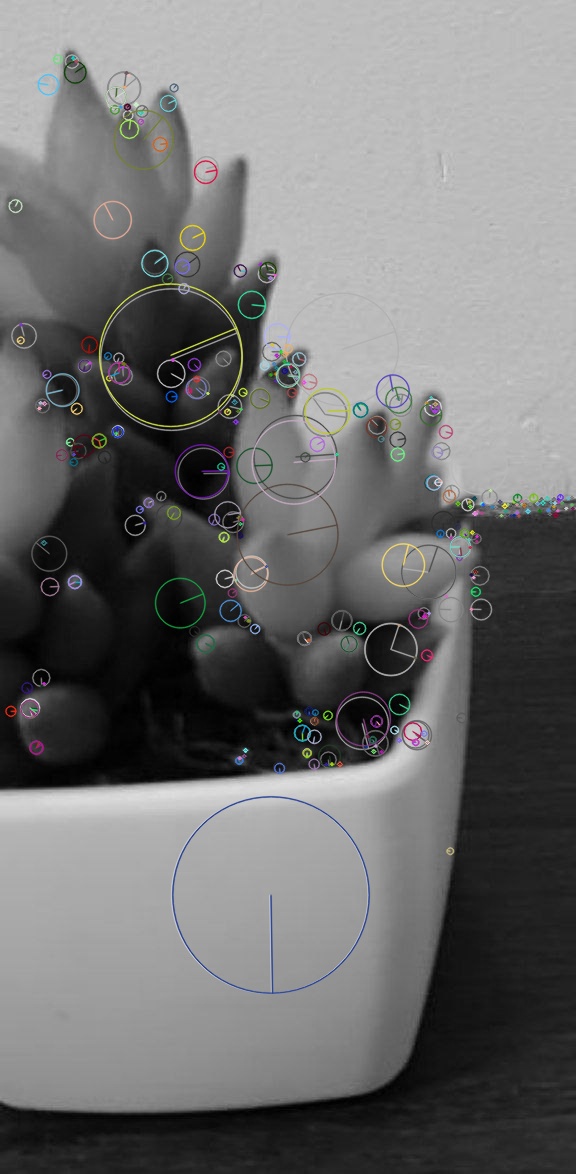
*با توجه به نتایج می‌توان اینطور استنباط کرد که با تغییر زاویه دید و پرسپکتیو تصویر، SIFT همچنان عملکرد قابل قبولی از خود نشان می‌دهد که البته این به بافت و جزئیات تصویر نیز بستگی دارد. همچنین عملکرد SIFT می‌تواند تحت تاثیر عوامل مختلفی از جمله اندازه و کیفیت تصاویرو میزان شباهت بین تصاویر شود.*

key point های حاصل از اعمال SIFT روی تصویر به صورت زیر است:

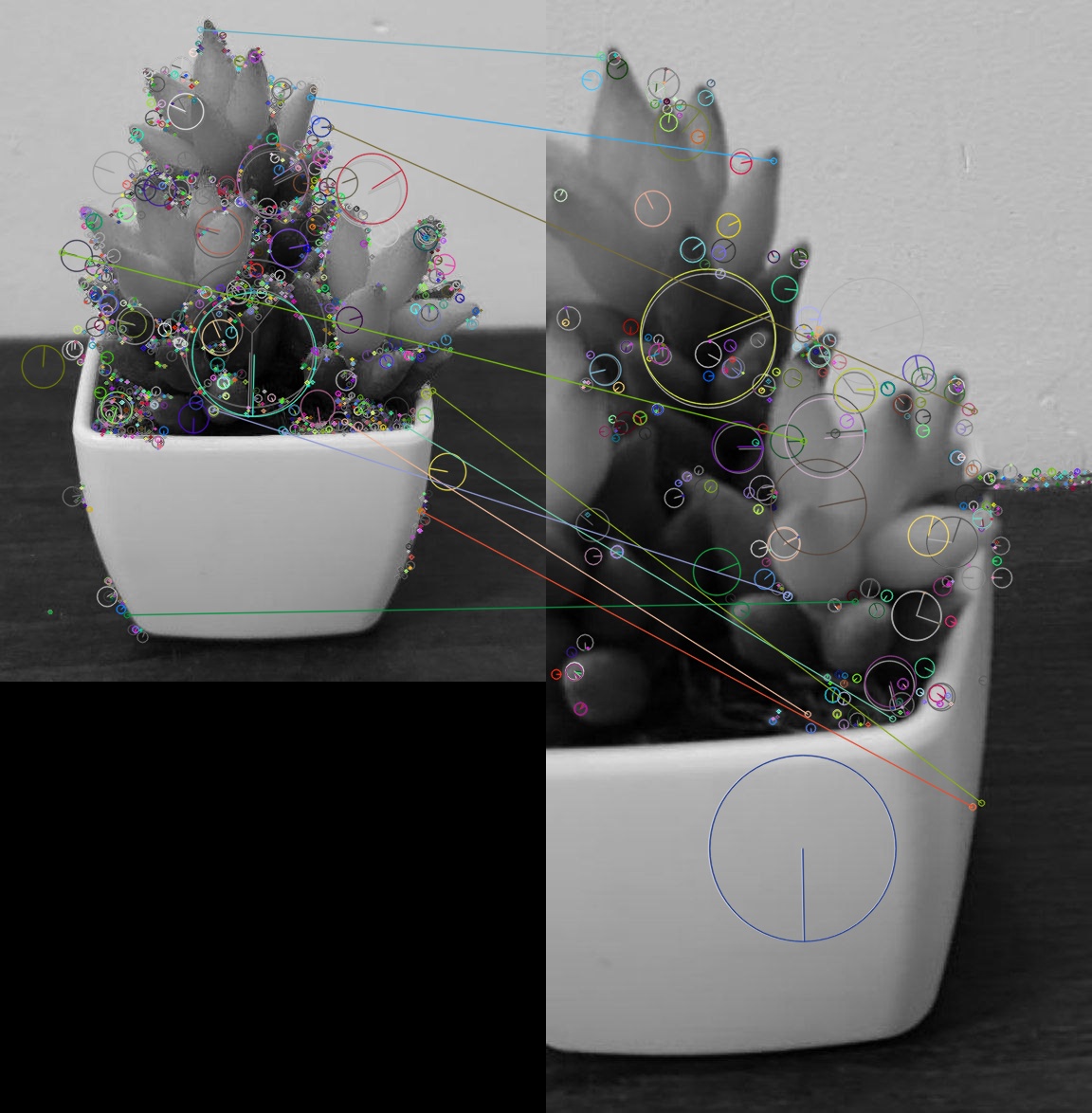
*7-2-2- تصاویر خروجی این سوال به صورت زیر هستند:*

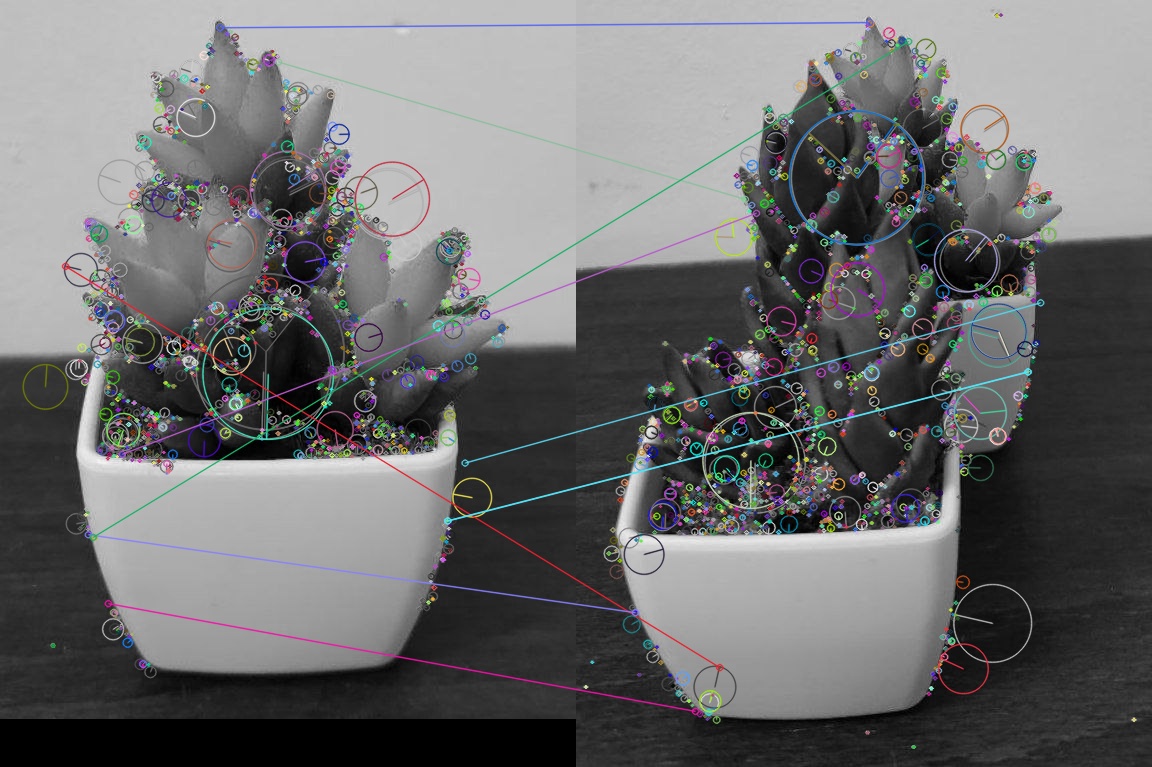
*تصاویر مربوط به اعمال SIFT*

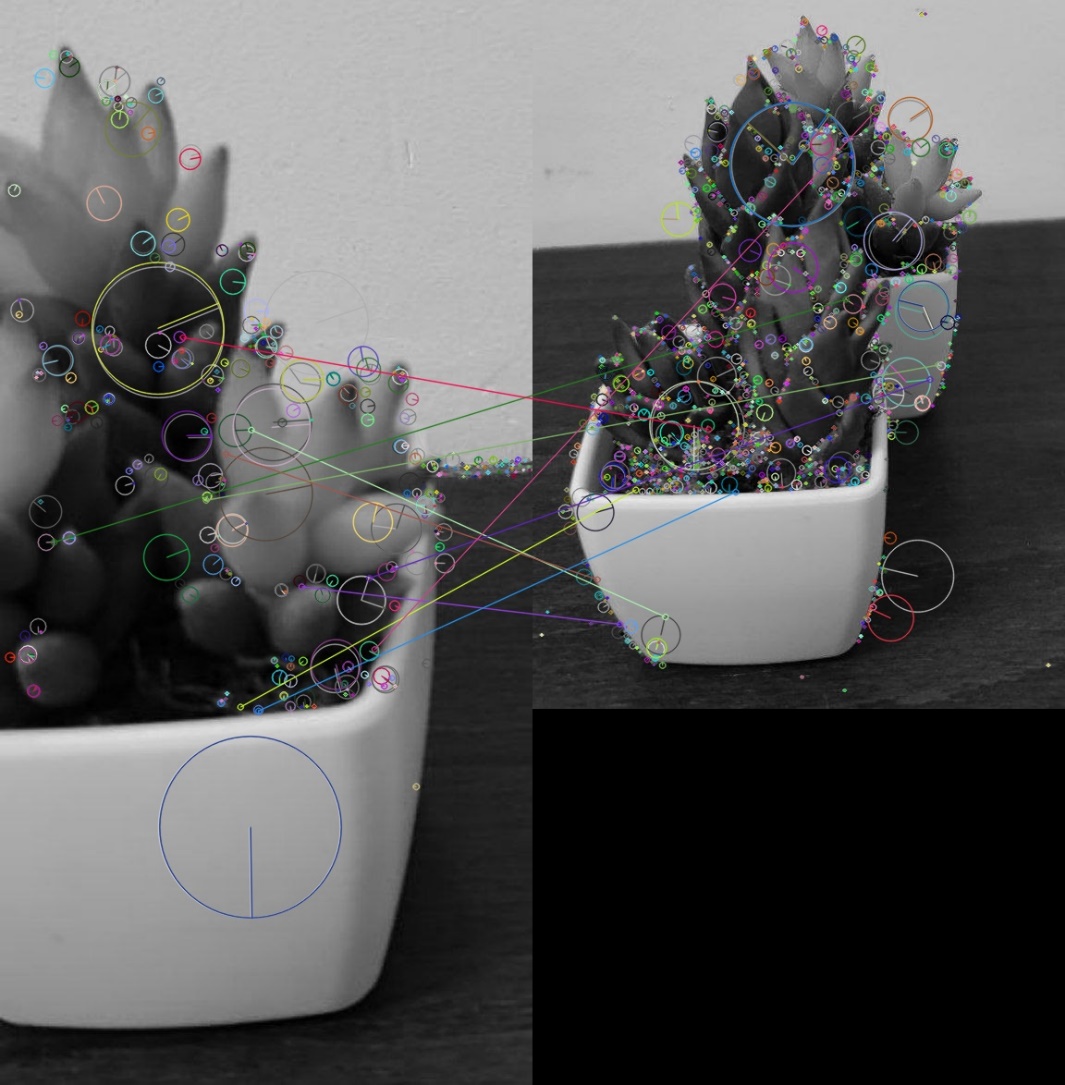
******

******

******

******

******

******

*در رابطه با نتایج اعمال SIFT و SURF روی تصاویر در سوال قبل به طور مفصل بحث کردیم. همان نتایج را می‌توان به خروجی های مربوط به این سوال نیز نسبت داد.*

***پیوست***

* *کدها و تصاویر مربوط به این تمرین در*[*GitHub*](https://github.com/saleh-sh/HW7) *در دسترس است.*