سید محمد طاها طباطبایی – تمرین سری هفتم

9812762838

چکیده:

توضیحات فنی:

7.1.1

7.2.1

برای پیاده‌سازی این بخش، از الگوریتم sift استفاده کرده ایم. در سلول اول، یک آبجکت از کلاس sift ساخته می‌شود. در سلول بعد، با استفاده از تابع detectAndCompute ، نقاط کلیدی و بردار های دیسکریپشن تصویر sl را محاسبه می‌کنیم. در نهایت با استفاده از تابع drawKeyPoints ، نقاط کلیدی به‌دست آمده را روی تصویر نمایش می‌دهیم. در دو سلول دیگر، این عملیات را برای تصاویر sm و sr نیز انجام دادم.

در گام بعدی، با استفاده از تابع BruteForceMatcher ، نقاط توجه(interest points) متناظر بین دو تصویر را با کمک بردار های دیسکریپشن دو تصویر محاسبه می‌کنیم. برای پیدا کردن نقاط، 2 نقطه با بیشترین تشابه از تصویر test ، نسبت به تصویر src را با تابع knnMatch پیدا می‌کنیم. در ادامه یک حلقه گوجود دارد که طبق روش پیشنهادی lowe، یک ترشولد اعمال می‌کند تا برخی نقاط که همچنان فاصله زیادی از هم دارند(تشابه کمی دارند) را نادیده بگیرد. با اینکار نسبتا مطمئن می‌شویم، نقاط توجه پیدا شده بین دو تصویر، نقاط با کیفیت بهتری هستند.

در گام آخر، درصد مشابهت بین دو تصویر را با کمک تابع howSimilar محاسبه کرده ایم. همانطور که مشخص است، درصد تشابه بین تصاویر sl و sm و درصد تشابه بین تصاویر sm و sr ، نسبت به درصد تشابه sl و sr ، تقریبا دو برابر است. دلیل این اتفاق تغییر زیاد زاویه دید بین دو تصویر sl و sr است. البته ممکن است به نظر بیاید، درصد مشابهت تصاویر به طور کلی کم است، اما دلیل این اتفاق، ترشولدی است که در مرحله قبل اعمال کردیم. با توجه به اینکه بسیاری از نقاط توجه پیدا شده ضعیف، در مرحله قبل کنار گذاشته شده اند، اگر ترشولد را آسان تر تعیین کنیم(عدد بالاتر) تعداد نقاط انتخاب شده بیشتر می‌شود، و دو تصویر از لحاظ آماری شبیه تر محسوب می‌شوند. این انتخاب ترشولد نسبی است. درصد مشابهت حدود 30 برای دو تصویر که تغییر زاویه دید دارند، همچنان به نظر عدد مناسبی است. برای تست، ترشولد 0.99 بر روی مقایسه sm و sr تست شد که نتیجه مشابهت 96.8 حاصل شد، یعنی اگر تقریبا تمام نقاط توجه پیدا شده را در نظر بگیریم، تشابه مورد انتظار نزدیک به 100 خواهد بود، هرچند طبیعتا 100 نخواهد بود.

7.2.2