

تمرین پنجم

هدف: آشنایی با شار نوری.

کد: کد این فعالیت را با کمک OpenCV (به جز در مواردی که صریحا در صورت سوال ذکر شده باشد) و به یکی از زبان‌های پایتون، متلب یا C/C++ بنویسید.

گزارش: ملاک اصلی انجام فعالیت گزارش آن است و ارسال کد بدون گزارش فاقد ارزش است. برای این فعالیت یک فایل گزارش در قالب pdf تهیه کنید و در آن برای هر سوال، تصاویر ورودی، تصاویر خروجی و توضیحات مربوط به آن را ذکر کنید. سعی کنید توضیحات کامل و جامعی تهیه کنید.

تذکر: مطابق قوانین دانشگاه هر نوع کپی برداری و اشتراک کار دانشجویان غیر مجاز بوده و شدیداً برخورد خواهد شد. استفاده از کدها و توضیحات اینترنت به منظور یادگیری بلامانع است، اما کپی کردن غیرمجاز است.

راهنمایی: در صورت نیاز میتوانید سوالات خود را در خصوص پروژه از تدریس یار درس، از طریق ایمیل زیر بپرسید.

E-mail: cv.ceit.aut@gmail.com

ارسال: فایل‌های کد و گزارش خود را در قالب یک فایل فشرده با فرمت StudentID_HW۰۵.zip تا تاریخ ۱۳۹۹/۱۰/۲۰ ارسال نمایید. شایان ذکر است هر روز تاخیر باعث کسر ۱۰٪ نمره خواهد شد.

توجه ۱: تمامی سوالات در این تمرین به هم مرتبط هستند. لطفاً قبل از شروع به پیاده‌سازی ابتدا تمامی سوالات را تا انتها مطالعه بفرمایید.

توجه ۲: ویدئو مورد نیاز در فایل زیپ شامل تعریف تمرین قرار گرفته است. در صورت نیاز می‌توانید تصاویر رنگی را به تصاویر سطح خاکستری تبدیل نموده و از پیش‌پردازش‌های مختلف به صلاح‌دید خود استفاده نمایید. در صورت پیش‌پردازش تصاویر، فرایند پیاده‌سازی شده را به طور کامل در فایل گزارش توضیح دهید.

در این تمرین قصد داریم با استفاده از شار نوری، بردار حرکت نقاط تصویر را استخراج نماییم. یک ویدئو از حرکت خودروها در شرایط مختلف به فایل تعریف تمرین پیوست شده است. با استفاده از اپن‌سی‌وی این ویدئو را بارگذاری نموده و فریم‌های این ویدئو را در طول تمرین مورد استفاده قرار دهید.

۱- ابتدا با استفاده از الگوریتم Shi-Tomasi گوشه‌ها را در فقط در اولین فریم از ویدئو پیدا کنید. برای این کار کافیست از تابع `cv2.goodFeaturesToTrack` در اپن‌سی‌وی استفاده نمایید. پارامترهای این تابع را به طور کامل در فایل گزارش توضیح دهید و بگویید هر کدام از آن‌ها چه تاثیری بر عملکرد این تابع می‌گذارند؟

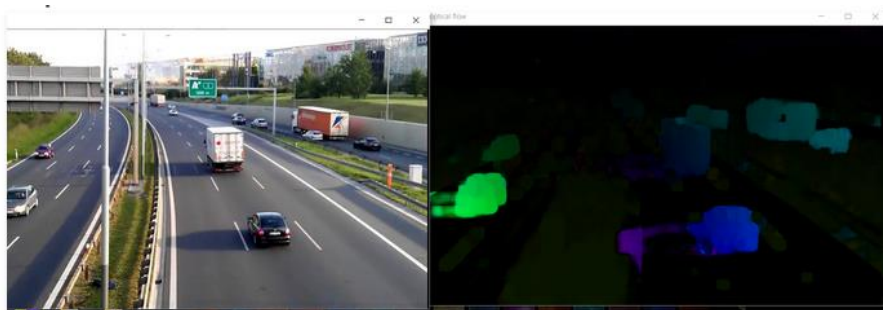
۲- سپس روی تک‌تک فریم‌های ویدئو پیش رفته و با استفاده از تابع `calcOpticalFlowPyrLK` شار نوری Lukas-kanade را برای نقاط انتخاب شده محاسبه نمایید. توجه کنید این تابع، بردار حرکت را برای هر کدام از نقاط ورودی تولید می‌کند. با استفاده از همین بردار حرکت، محل هر یک از گوشه‌های انتخاب شده در فریم بعدی را محاسبه نمایید تا بتوانید در فریم‌های بعدی نقاط را به طور کامل دنبال کنید. بردارهای حرکت این نقاط را روی تصاویر رسم نمایید تا شکلی

مشابه با شکل زیر حاصل شود. این کار را در ۵ ثانیه ابتدایی ویدیو انجام دهید. سپس در ۵ ثانیه دوم، گوشه‌ها را دوباره پیدا کنید و فرایند را تکرار نمایید تا بتوانید مسیر حرکت خودروهایی که تازه به تصویر وارد شده‌اند را نیز دنبال نمایید.



شکل ۱ خروجی مورد انتظار برای الگوریتم Lukas-kanade

- ۳- به نظر شما چرا نمی‌توان الگوریتم Lukas-kanade را روی تمام نقاط موجود در تصویر اجرا کرد؟
- ۴- الگوریتم Gunner Farneback که به شار نوری متراکم^۱ معروف است، نقطه ضعف الگوریتم قبلی را برطرف کرده و قادر است شار نوری را برای تک تک نقاط موجود در تصویر محاسبه نماید. در مورد این الگوریتم مطالعه نموده و توضیح دهید که چگونه توانسته مشکل الگوریتم Lukas-kanade را برطرف نماید؟ سپس با استفاده از تابع `calcOpticalFlowFarneback` در اپن‌سی‌وی، شار نوری متراکم را برای فریم‌های این ویدئو محاسبه نموده و نقشه حرارتی حاصل را مطابق با شکل زیر در فایل گزارش نمایش دهید.



شکل ۲ نقشه حرارتی نقاط متحرک ویدئو

موفق باشید

^۱ Dense Optical Flow