



دانشگاه صنعتی امیر کبیر  
(پلی تکنیک تهران)

## درس پیوپی کامپیوونر

استاد درس جناب آقای دکتر صفابخش

(تمرین سری هشتم)

محسن عبادپور | ۴۰۰۱۳۱۰۸۰ | [m.ebadpour@aut.ac.ir](mailto:m.ebadpour@aut.ac.ir)

نیمسال اول سال تحصیلی ۱۴۰۱-۱۴۰۲



---

## فهرست پاسخ ها

---

- ۳ ..... **goodFeaturesToTrack** تابع پارامترهای تاثیر و بررسی
- ۸ ..... **Lukas-Kanade** حرکت با الگوریتم تعیین مسیر
- ۱۲ ..... **gunner-farneback** حرکت با الگوریتم تعیین مسیر
- ۱۴ ..... **روش ها** مقایسه روش ها

## مسئله اول

### مسئله اول: تأثیر و بررسی پارامترهای تابع goodFeaturesToTrack

الگوریتم Shi-Tomasi یک الگوریتم تشخیص ویژگی و گوشه‌یاب می‌باشد که نسخه بهبود یافته‌ی گوشه‌یاب هریس بوده و برای تصویر ورودی گوشه‌ها را نمایان می‌کند. این الگوریتم با ساختن یک ماتریس به نام structure tensor و استفاده از گرادیان پیکسل‌های تصویر اقدام به گوشه‌یابی می‌کند که این عمل با محاسبه‌ی مقدار ویژه‌ی ماتریس فوق انجام می‌پذیرد([لینک](#)). گوشه‌های حاصل از این الگوریتم مناسب بوده و سعی می‌شود نقاط حاصل یکتا باشد تا بدین صورت فرآیند رهگیری (track) منحصر‌بفرد و سریع‌تر انجام گیرد.

حال به بررسی پارامترهای تابع goodFeaturesToTrack می‌پردازیم؛ این تابع در پیاده‌سازی خود و بصورت کامل‌ترین حالت ممکن دارای ۸ پارامتر ورودی می‌باشد.

- پارامتر اول image نام دارد که سطح خاکستری تصویر مد نظر را ورودی می‌گیرد.
- پارامتر دو maxCorners نام دارد که حداقل تعداد گوشه‌های مدنظر را نشان می‌دهد. این یک کران بالا بوده و گوشه‌های بازگشتی می‌توانند بیشتر از این باشند. اگر تعداد گوشه‌های محاسبه شده بیشتر از این حد آستانه باشد، توسط کیفیت نقاط حاصل عمل فیلتر و مرتب سازی انجام شده و تعداد مد نظر صرفا بازگردانده می‌شود.
- پارامتر سوم qualityLevel نام دارد که یک حد آستانه برای فیلتر کردن گوشه‌ها و ویژگی‌های استخراجی می‌باشد؛ ویژگی‌هایی که از این حد آستانه کیفیت پایین‌تری داشته باشند حذف می‌شوند.(این حد آستانه در زمان محاسبه و انتخاب مقادیر ویژه در ماتریس ذکر شده استفاده می‌شود)
- پارامتر چهارم minDistance نام دارد که نشان میدهد فاصله اقلیدسی گوشه‌ها و ویژگی‌های استخراجی حداقل باید چقدر باشد. این پارامتر به ما کمک می‌کند نقاط بسیار نزدیک به هم که در تصویر کنار هم قرار داشته و اطلاعات مفید جدیدی را تامین نمی‌کند را نادیده بگیریم.

- پارامتر پنجم mask نام دارد که به ما کمک می کند توسط آن، ناحیه ای مدنظر را برای استخراج ویژگی مشخص کنیم تا صرفا در آن ناحیه الگوریتم به پردازش و محاسبه ای ویژگی اقدام کند. برای کاهش بار محاسباتی می توانیم از آن استفاده کنیم.
  - پارامتر ششم blockSize نام دارد که اندازه بلوک متوسط را برای محاسبه تشکیل ماتریس هموارسازی مشتق در همسایگی هر پیکسل تعیین می کند.
  - پارامتر هفتم useHarrisDetector نام دارد که تعیین می کند به جای روش موجود آیا از خود گوشیاب هریس استفاده شود یا خیر.
  - پارامتر هشتم K نام دارد که در روش هریس مورد استفاده قرار می گیرد.
- از بین پارامترهای فوق، تصویر ورودی که بدیهی بوده و از مابقی نیز بهتر است چهار پارامتر بعدی که عملکرد الگوریتم را مشخص می کند را تعیین کنیم. برای این منظور در دو گام اقدام می کنیم؛ در گام اول پارامترهای مستقل از تاثیرگذاری در استخراج ویژگی (minDistance=5 و maxCorners=1000) که مستقل از ویژگی استخراجی بوده و صرفا فیلتر می کنند) را بسیار آزاد انتخاب کرده و سپس مقادیر مناسب را برای پارامترهای blockSize و qualityLevel بدست می آوریم. در گام بعدی، دو پارامتر قبلی را به ازای پارامترهای تعیین مقدار شده بدست می آوریم. در تصویر صفحه پنجم گام اول قابل مشاهده است؛ از چپ به راست مقدار blockSize از ۵ تا ۵۰ تغییر پیدا می کند و از بالا به پایین مقدار qualityLevel از ۰.۰ تا ۰.۶ کاهش پیدا می کند.
- در بحث رهگیری اشیاء، ویژگی های مورد استفاده در عین حال که بایستی مناسب و مفید و موثر باشند بایستی تعداد خیلی زیادی نیز نداشته باشند تا باز محاسبات زیاد نشده و فرآیند رهگیری واقعا بتواند بصورت real-time انجام بپذیرد. با توجه به این نکته مقادیر ۰.۳ و ۰.۱۵ بترتیب برای blockSize و qualityLevel انتخاب می شود.(تصویر دوم از چپ در ردیف دوم از بالا)
- در تصویر صفحه ششم گام دوم قابل مشاهده است؛ از چپ به راست مقدار minDistance از ۵ تا ۱۰۰ تغییر پیدا می کند و از بالا به پایین مقدار maxCorners از ۱۰۰۰ تا ۵۰ کاهش پیدا می کند. بر حسب توضیحات ارائه شده در بند قبل، مقادیر ۱۰۰ و ۱۰ بترتیب برای minDistnace و maxCorners انتخاب می شود(ردیف آخر و ستون دوم از چپ)
- حال بر اساس نتایج بدست آمده، تصویر نهایی برای عکس ورودی بصورت صفحه هفتم حاصل می شود.







قابل مشاهده است که طبق پارامترهای بدست تعیین شده، نقاط و ویژگی‌های استخراجی کاملاً برای رهگیری شی مناسب هستند زیرا علاوه بر اینکه تعداد زیادی نقطه استخراج نشده، نقاط ویژگی استخراج شده در تمامی قسمت‌های مهم شیء وجود دارد مانند حروف بزرگ انگلیسی، چنگال و هویج روی چنگال، قارچ‌های تصویر، لوگوی الیت و... .

## مسئله دوم

### مسئله دوم: تعیین مسیر حرکت با الگوریتم Lukas-Kanade

برای این منظور، از تابع معرفی شده استفاده شده است. برای دریافت فریم‌های ویدیو دو مکانیزم پیاده‌سازی شده است؛ یکی دریافت فریم از ویدیوی ضبط شده و دیگری دریافت فریم از وبکم می‌باشد که برای سهولت و ثبات در نتایج، از ویدیوی ضبط شده استفاده شده است.

پیاده‌سازی به این صورت انجام شده است که در یک لیست پایتونی همواره سه فریم آخر دریافتی به همراه سطوح خاکستری منتظر و ویژگی‌های حاصل از goodToTrack نگهداری می‌شود. وقتی فریم جدید وارد می‌شود آن را به لیست append می‌کنیم و قدیمی ترین فریم را پاک می‌کنیم.

حال در لیست حاصل مسیر مستقیم حرکت نقاط را از طریق تابع calcOpticalFlowPyrLK و فریم اول و سوم بدست آورده و با یک فلش آن را نمایش می‌دهیم. برای جستجوی نقاط منتظر در یک فریم‌ها از پنجره‌ی ۲۰\*۲۰ استفاده می‌کنیم که نسبت به پنجره‌ی minDistance و blockSize ما بزرگتر بوده و به ما این امکان را می‌دهد که بتوانیم در مقابل حرکت‌های سریع عملکرد مطلوبتری داشته باشیم.

ویدیوی خروجی به پیوست در سایت کورسز آپلود شده است. از جایی که حجم ویدیو بالا بوده و ممکن است در زمان تصحیح و دریافت فایل، دسترسی به شبکه داخلی دانشگاه وجود نداشته باشد و سرعت بارگیری بالا نباشد، این ویدیو در داخل فایل زیپ قرار نگرفته است. صرفا تعدادی فریم نمونه استخراج شده و در صفحات بعدی مسیر حرکت شی در جهات گوناگون آورده شده است.

در صفحه نهم و دهم بعد مسیر مستقیم حرکت شی در سه فریم متوالی (فریم یک به سه) آورده شده است. فرآیند رسم مسیر حرکت را می‌توان به جای مسیر مستقیم طور دیگه‌ای نیز انجام داد بطوری که نقاط مسیر حرکت را ابتدا از نقطه اول به نقطه دوم و سپس از نقطه دوم به نقطه سوم خط کشید. این مورد نیز پیاده‌سازی شده و تصاویر نمونه آن در صفحه یازدهم آورده شده است.





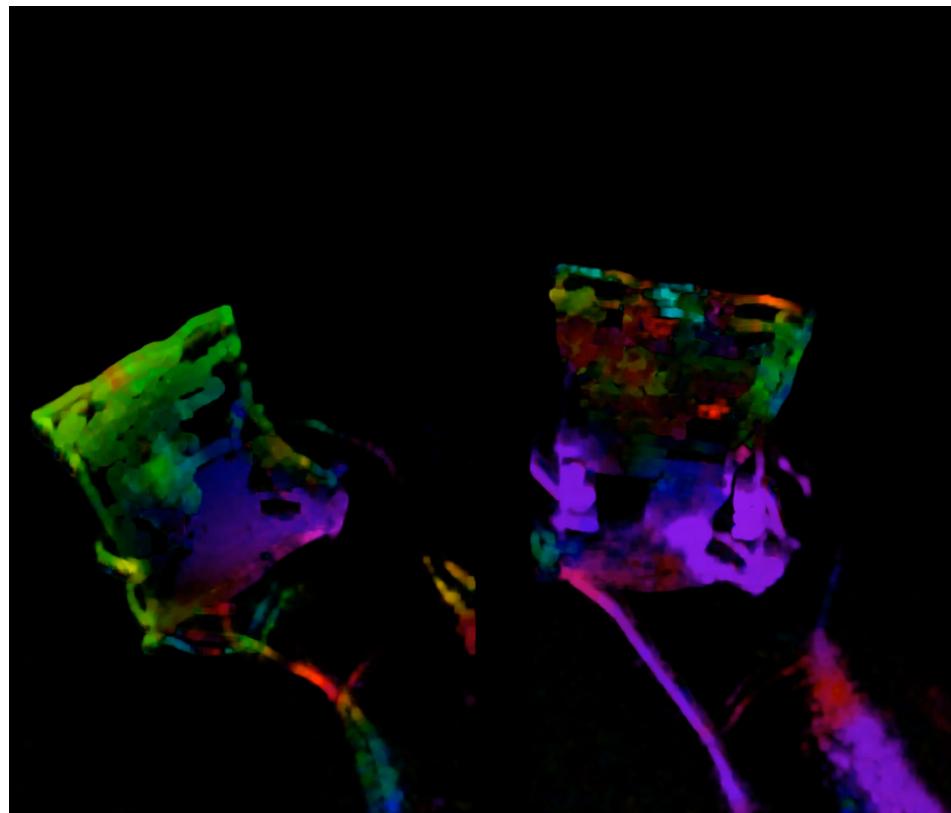


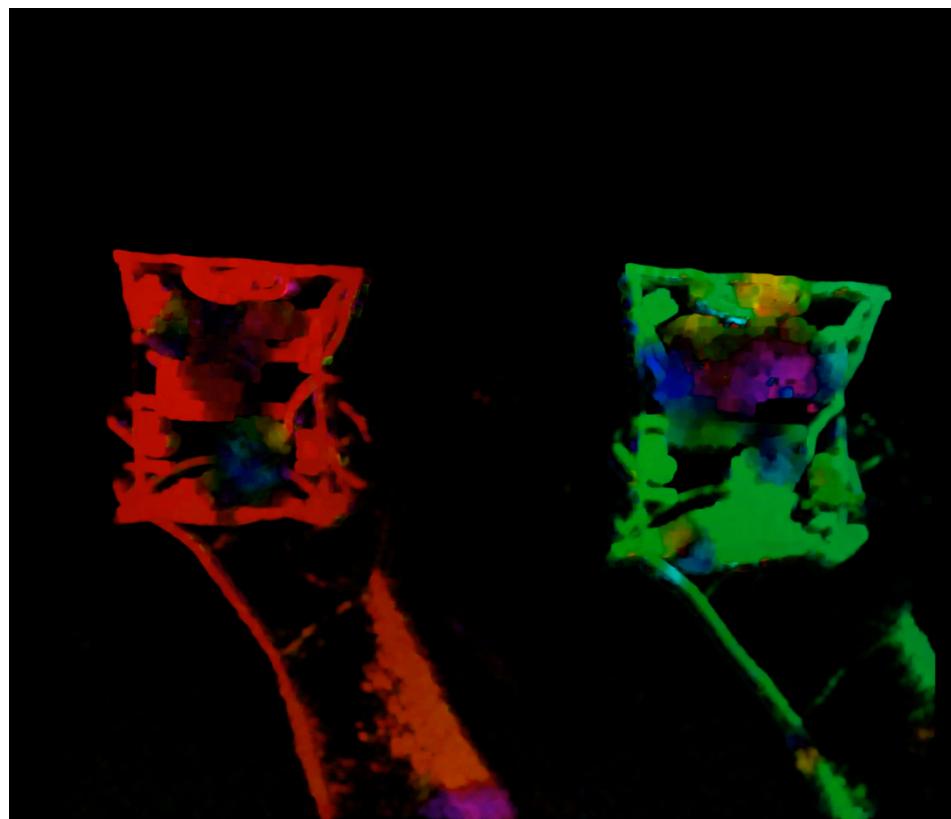
### مسئله سوم

#### مسئله سوم: تعیین مسیر حرکت با الگوریتم gunner-farneback

برای این منظور، از تابع و لینک معرفی شده استفاده شده است. خروجی تابع calcOpticalFlowFarneback که حاصل از الگوریتم gunner-farneback می‌باشد بر اساس جریان و شار نوری حاصل از نسخه‌های هموارشده‌ی گوسی در یک هرم از تصویر ورودی بوده و این جریان را بین هر دو تصویر متوالی محاسبه می‌کند و این پردازش و محاسبه به ازای هر پیکسل انجام شده و همچنین فرض می‌شود که روشنایی ثابت است.

خروجی مربوط به این قسمت در قالب ویدیو ضمیمه شده و در زیر نمونه‌هایی از آن آورده شده است.





## مسئله چهارم

### مسئله چهارم: مقایسه روش‌ها

معمولًا در بحث tracking اشیاء ما میدانیم که چه شیء ای و تقریباً در چه ناحیه‌ای حضور دارد (پس از اولین شناسایی) و در فریم بعدی نمیتواند بصورت غیرممکن و ماورایی جا به جا شود لذا صرف هزینه‌ی مجدد برای شناسایی و تطبیق تمامی ویژگی کار بیهوده‌ای بوده و منابع مصرفی را بالا می‌برد اما گاهی وجود محیط و شرایط پیچیده باعث می‌شود که از این مورد استفاده کنیم.

با توجه به نکته فوق، در زمانهایی که جسم ما با سایر اجسام همپوشانی یا اشتراک داشته باشد، یا قسمتی از آن در حال پنهان شدن یا نمایان شدن باشد (گم کردن یا پیدا کردن نقاط)، یا محیط پس زمینه پیچیده و چندین جسم در حال تغییر باشد، یا شرایط نور و اخذ تصویر متغیر باشد، یا سرعت تغییر واقعاً بسیار بالا باشد، یا تغییرات مربوط به مقیاس و چرخش وجود داشته باشد (نظریه دوران و نزدیک شدن به دورین) یا ... برای بالا بردن دقت رهگیری و اطمینان از آن بهتر است از روش‌های مبتنی بر نقاط کلیدی و تطبیق آن نظریه SIFT استفاده شود هر چند سرعت کاهش و بار محاسبات بسیار زیاد شده و شاید دیگر نتوان بصورت real-time رهگیری را انجام داد. (سرعت فدای دقت بالا می‌شود) با عنایت به توضیحات فوق، میتوان گفت که روش SIFT در بحث بار محاسباتی و زمان پردازشی در مقابل دو روش دیگر مورد بررسی در این تمرین نقطه ضعف داشته اما در مقابل در بحث صحت و دقت رهگیری نقطه مزیت دارد. (تغییرات روشنایی و چرخش و اندازه، نقاط گم شده، تغییرات با سرعت بالا و ...) لازم به ذکر است که SIFT برای رهگیری اشیایی میتواند موثر باشد که سطح توصیف شونده ای داشته باشند در غیر اینصورت دقت آن نیز با چالش مواجه می‌شود. (رهگیری یک کاغذ صاف و ساده و سرراست برای روش SIFT همچین کار ساده ای نیست)

دو روش مورد بررسی در این تمرین نیز با وجود اینکه به اندازه روش‌های مبتنی بر استخراج و توصیف نقاط کلیدی دقت بالایی ندارند اما در زمینه‌هایی که سرعت پردازش و real-time بودن رهگیری برای ما اولویت بوده و همچنین رهگیری اشیای مختلف را میخواهیم انجام دهیم این دو روش بهتر می‌باشد. حال اگر بخواهیم این دو روش را باهم مقایسه کنیم بایستی گفت که روش اول چون با تعداد نقاط محدودی سر و کار دارد از حجم پردازش و بار محاسبات پایین و سرعت بالایی برخوردار بوده و برای مباحثه بلادرنگ بیشتر فیت است اما در مقابل در صورت گم شدگی نقاط اولیه یا تغییر با شدت و سرعت جا به جایی بالا کلا با مشکل مواجه شده و دیگر نمیتواند رهگیری را با دقت مناسب انجام دهد. روش دوم نیز برای هر پیکسل تخمین و شار نوری را محاسبه می‌کند لذا بار پردازشی و محاسباتی آن بیشتر از روش اول بوده و در نتیجه برای بلادرنگ مناسب نمی‌باشد اما مزیت آن نسبت

به روش اول در این است که در صورت گم شدگی نقاط، سختی در استخراج ویژگی و... با مشکل مواجه نشده و میتواند رهگیری را از سر گیرد و ادامه دهد لذا صحت رهگیری اشیای آن بهتر است.

در کل میتوان نتیجه گرفت مقایسه انواع روش ها برای رهگیری اشیا وابسته به موضوع، مسئله، شرایط و محیط و خواسته اصلی(دقت/سرعت) داشته و نیمتوان برای همه موارد نسخه واحد تجویز کرد.