

## Shi-Tomasi Corner Detection

روش **Shi-Tomasi Corner Detection** که در سال 1994 معرفی شد، یک نسخه بهبود یافته از الگوریتم Harris است. هر دو روش بر پایه‌ی ماتریس ساختار (Structure Tensor) کار می‌کنند و با محاسبه مقادیر ویژه (Eigenvalues) آن به تشخیص نقاط گوشه می‌پردازند. تفاوت اصلی در این است که Shi-Tomasi به جای استفاده از فرمول پاسخ Harris، مستقیماً از مقادیر ویژه برای تصمیم‌گیری استفاده می‌کند. این تغییر ساده، دقت و پایداری الگوریتم را در عمل به طور قابل توجهی افزایش می‌دهد.

در الگوریتم Shi-Tomasi، هر نقطه بر اساس مقدار کمینه‌ی دو مقدار ویژه ماتریس  $M$  ارزیابی می‌شود:

$$\min(\lambda_1, \lambda_2) = R$$

اگر این مقدار از یک آستانه مشخص بیشتر باشد، نقطه یک گوشه محسوب می‌شود. دلیل این کار این است که اگر هر دو مقدار ویژه بزرگ باشند، یعنی تغییرات شدت روشنایی در هر دو جهت زیاد است (شرط اصلی برای گوشه بودن). به همین دلیل، نتایج Shi-Tomasi نسبت به Harris مطمئن‌تر و با خطای کمتر هستند.

این الگوریتم در OpenCV از طریق تابع `cv2.goodFeaturesToTrack()` پیاده‌سازی شده است. این تابع به طور خودکار بهترین نقاط کلیدی (گوشه‌ها) را در تصویر انتخاب می‌کند. علاوه بر تشخیص دقیق گوشه‌ها، امکان محدود کردن تعداد ویژگی‌ها و تنظیم حداقل فاصله بین آن‌ها نیز وجود دارد. به همین دلیل Shi-Tomasi انتخاب اول در بسیاری از الگوریتم‌های ردیابی ویژگی‌ها (Feature Tracking) مانند **KLT Tracker** است.

از نظر کاربردی، Shi-Tomasi به دلیل دقت بالا در پیدا کردن گوشه‌های پایدار، در مسائلی مثل ردیابی حرکت در ویدئو، ساخت پانوراما و بینایی رباتیک بسیار استفاده می‌شود. این روش نسبت به تغییرات نور و نویز مقاوم‌تر از Harris عمل می‌کند و گوشه‌های بهتری برای تطبیق (Matching) انتخاب می‌کند. در نتیجه، کیفیت کلی سیستم‌های مبتنی بر ویژگی‌ها (Feature-Based Systems) بهبود پیدا می‌کند.

ورودی های الگوریتم:

```
cv2.goodFeaturesToTrack(image, maxCorners, qualityLevel, minDistance[, mask[, blockSize[, useHarrisDetector[, k]]]])
```

image

- تصویر ورودی باید خاکستری (Grayscale) و از نوع float32 یا 8-bit باشد.
- این همان تصویری است که الگوریتم روی آن گوشه ها (Corners) را جستجو می کند.

maxCorners

- حداکثر تعداد گوشه هایی که می خواهید الگوریتم پیدا کند.
- اگر بیشتر از این تعداد گوشه وجود داشته باشد، الگوریتم بهترین ها را برمی گزیند.
- مقدار رایج: 100، 200 یا بیشتر بسته به نیاز.

qualityLevel

- عددی بین 0 و 1 که نشان دهنده ی آستانه کیفیت گوشه هاست.
- مثلاً اگر  $qualityLevel = 0.01$  باشد، یعنی فقط گوشه هایی انتخاب شوند که مقدار پاسخشان بیشتر از 1٪ بهترین گوشه است.
- هرچه مقدار بزرگ تر باشد → فقط گوشه های قوی تر انتخاب می شوند.

minDistance

- حداقل فاصله بین دو گوشه انتخابی (برحسب پیکسل).
- اگر دو گوشه خیلی به هم نزدیک باشند، فقط یکی از آن ها نگه داشته می شود.
- این پارامتر باعث می شود گوشه ها در تصویر پراکندگی بهتری داشته باشند.

## mask.اختیاری

- یک تصویر ماسک با همان اندازه تصویر ورودی.
- فقط گوشه‌هایی در مناطقی پیدا می‌شوند که مقدار ماسک غیرصفر باشد.
- کاربرد: اگر فقط می‌خواهید در ناحیه خاصی از تصویر گوشه پیدا کنید.

## blockSize.اختیاری، پیش فرض = 3

- اندازه بلوک (پنجره همسایگی) که برای محاسبه ماتریس گرادیان استفاده می‌شود.
- مقدار رایج: 3، 5 یا 7.
- بلوک بزرگ‌تر → نتایج پایدارتر ولی حساسیت کمتر به گوشه‌های کوچک.

## useHarrisDetector.اختیاری، پیش فرض False

- اگر مقدارش **True** باشد، الگوریتم از **Harris** به جای Shi-Tomasi استفاده می‌کند.
- اگر **False** باشد، همان روش Shi-Tomasi اجرا می‌شود.

## k.اختیاری، فقط برای Harris

- پارامتر حساسیت در فرمول Harris معمولاً بین 0.04 و 0.06.
- فقط زمانی استفاده می‌شود که `useHarrisDetector=True` باشد.

## خروجی

این تابع یک آرایه از نقاط برمی‌گرداند (مختصات  $(x, y)$ ,  $(y, x)$  گوشه‌هایی که پیدا شده‌اند).