

SIFT:

الگوریتم **SIFT** یا **Scale-Invariant Feature Transform** یکی از پرکاربردترین روش‌ها در حوزه بینایی ماشین و پردازش تصویر است که در سال ۱۹۹۹ توسط **David Lowe** معرفی شد. هدف اصلی این الگوریتم استخراج ویژگی‌هایی از تصویر است که نسبت به تغییر مقیاس (بزرگ‌نمایی و کوچک‌نمایی)، چرخش، تغییر زاویه دید و حتی تغییرات نسبی در روشنایی و نویز مقاوم باشند. به همین دلیل SIFT به عنوان یک الگوریتم مقاوم (Robust) شناخته می‌شود و در بسیاری از کاربردها مانند شناسایی اشیاء، ردیابی، ساخت مدل سه‌بعدی و هم‌ترازی تصاویر مورد استفاده قرار می‌گیرد.

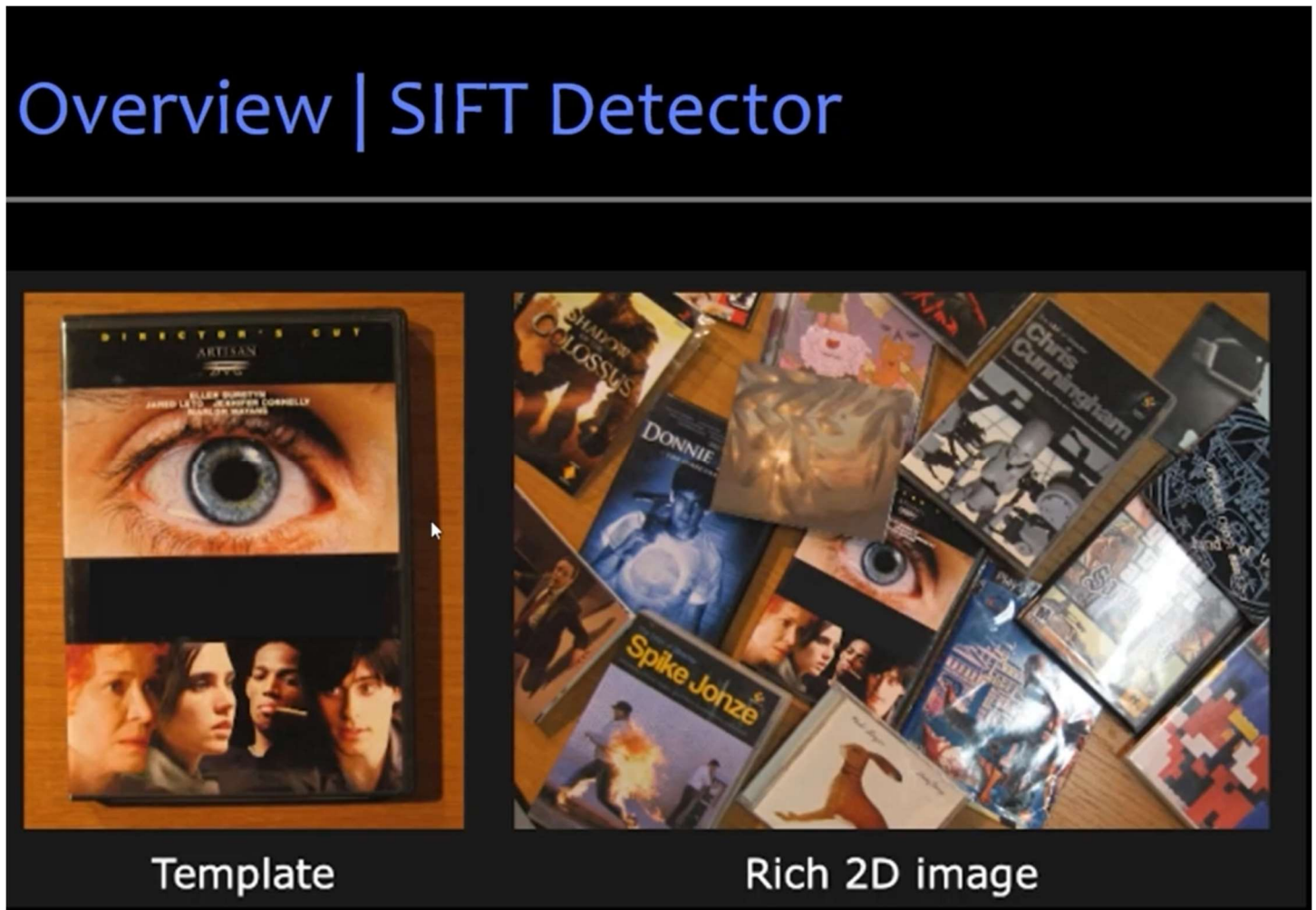
فرایند SIFT با یافتن نقاط کلیدی (Keypoints) در تصویر آغاز می‌شود. این نقاط معمولاً گوشه‌ها، لبه‌ها یا نواحی‌ای هستند که تغییرات شدت روشنایی در آن‌ها به گونه‌ای است که نسبت به تغییرات محیطی پایدار می‌مانند. برای یافتن این نقاط، از روش **Difference of Gaussian (DoG)** در مقیاس‌های مختلف استفاده می‌شود. ایده این است که با محاسبه اختلاف بین تصاویر محوشده (Blurred) در مقیاس‌های متفاوت، مناطقی که تغییرات شدیدی دارند برجسته می‌شوند و به عنوان نقاط کلیدی در نظر گرفته می‌شوند.

بعد از استخراج نقاط کلیدی، هر نقطه باید به صورت توصیف‌گر (Descriptor) نمایش داده شود تا امکان مقایسه بین تصاویر مختلف وجود داشته باشد. در این مرحله، برای هر نقطه کلیدی یک ناحیه محلی در اطراف آن انتخاب می‌شود و گرادیان شدت روشنایی پیکسل‌ها محاسبه می‌گردد. سپس یک هیستوگرام جهت‌گیری (Orientation Histogram) ساخته می‌شود که جهت غالب (Dominant Orientation) و تغییرات محلی را نمایش می‌دهد. این کار باعث می‌شود توصیف‌گر نهایی نسبت به تغییرات زاویه‌ای و چرخش مقاوم باشد.

در نهایت، توصیف‌گرهای به‌دست‌آمده برای مقایسه بین تصاویر به کار می‌روند. برای مثال اگر بخواهیم دو تصویر از یک شیء را با زوایای متفاوت شناسایی کنیم، توصیف‌گرهای SIFT از هر دو تصویر استخراج می‌شوند و سپس با استفاده از الگوریتم‌هایی مثل **Nearest Neighbor** یا **RANSAC** نقاط متناظر شناسایی و تطبیق داده می‌شوند. همین خاصیت باعث شده است که SIFT یکی از محبوب‌ترین الگوریتم‌ها در پروژه‌های شناسایی اشیاء و بینایی ماشین باشد، هرچند به دلیل پیچیدگی محاسباتی و ثبت پتنت آن، الگوریتم‌های ساده‌تر یا آزادتر مانند **SURF** و **ORB** نیز در برخی کاربردها جایگزین شده‌اند.

یکی از نکات مهم در مورد الگوریتم SIFT این است که با وجود دقت و پایداری بسیار بالا، از نظر محاسباتی نسبتاً سنگین محسوب می‌شود. اجرای آن روی تصاویر بزرگ یا مجموعه‌داده‌های حجیم زمان‌بر است و به همین دلیل در کاربردهای بلادرنگ (Real-Time) مانند پردازش ویدئو یا سیستم‌های رباتیک ممکن است محدودیت ایجاد کند. با این حال، به دلیل قابلیت اعتماد بالای آن در شرایط پیچیده، همچنان به عنوان الگوریتم مرجع برای مقایسه و ارزیابی دیگر روش‌های استخراج ویژگی مورد استفاده قرار

می‌گیرد. علاوه بر این، پیاده‌سازی‌های بهینه‌سازی‌شده و نسخه‌های توسعه‌یافته‌ی SIFT نیز ارائه شده‌اند تا کارایی آن در شرایط واقعی بهبود یابد.



نقاط مورد علاقه:

در الگوریتم **SIFT**، نقاط مورد علاقه یا همان **Keypoints** نقاطی از تصویر هستند که تغییرات شدیدی در شدت روشنایی دارند و در مقیاس‌های مختلف پایدار باقی می‌مانند. این نقاط معمولاً گوشه‌ها، لبه‌ها و یا نواحی با بافت غنی (High Texture) هستند که به راحتی در تصاویر مختلف اگر چرخانده، کوچک یا بزرگ شوند قابل تشخیص‌اند.

SIFT برای پیدا کردن این نقاط از روش **Difference of Gaussian (DoG)** استفاده می‌کند. ابتدا تصویر در مقیاس‌های مختلف (با فیلتر گاوسی) محو می‌شود، سپس اختلاف بین تصاویر محوشده محاسبه می‌گردد. نقاطی که در این اختلاف‌ها به عنوان

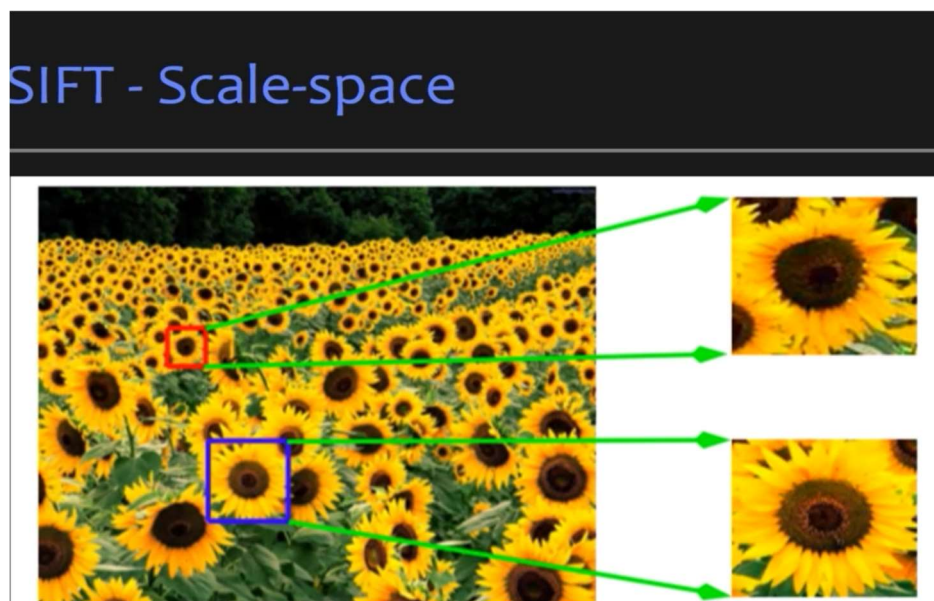
ماکزیمم یا مینیمم محلی ظاهر شوند به عنوان کاندیدای نقاط کلیدی در نظر گرفته می شوند. این فرایند باعث می شود نقاط انتخاب شده نسبت به تغییر اندازه تصویر مقاوم باشند.

همچنین، برای هر نقطه کلیدی یک **جهت غالب (Dominant Orientation)** تعیین می شود که با استفاده از گرادیان شدت در اطراف آن محاسبه می گردد. این جهت باعث می شود نقاط کلیدی علاوه بر مقیاس، نسبت به چرخش هم مقاوم باشند. به این ترتیب، حتی اگر تصویر چرخانده شود، الگوریتم همچنان همان نقاط کلیدی را شناسایی می کند.

در مجموع، نقاط مورد علاقه در SIFT همان بخش هایی از تصویر هستند که خاص، پایدار و متمایزکننده محسوب می شوند و می توان از آن ها برای تطبیق بین تصاویر مختلف استفاده کرد.

Scale Space:

مفهوم **Scale Space** یکی از پایه های اصلی الگوریتم SIFT است و برای شناسایی ویژگی هایی به کار می رود که در مقیاس های مختلف تصویر پایدار باقی می مانند. ایده اصلی این است که اشیاء یا جزئیات در تصاویر ممکن است با تغییر اندازه یا فاصله دوربین به شکل متفاوتی دیده شوند؛ بنابراین باید ویژگی ها را به گونه ای استخراج کرد که مستقل از بزرگنمایی یا کوچکنمایی باشند. برای این منظور، تصویر در سطوح مختلف مقیاس (Scale Levels) بررسی می شود و ویژگی ها در تمام این مقیاس ها تحلیل می شوند.



برای ایجاد فضای مقیاس (Scale Space)، تصویر اصلی با فیلتر گاوسی (Gaussian Filter) در درجات مختلف محو (Blur) می شود. این عمل باعث می شود جزئیات ریز به تدریج حذف شوند و تنها ساختارهای بزرگتر باقی بمانند. با محو کردن تصویر در

سطوح مختلف و مقایسه آن‌ها، می‌توان نقاطی را شناسایی کرد که در مقیاس‌های گوناگون پایدار هستند. این نقاط همان کاندیداهای **Keypoint** محسوب می‌شوند.

در عمل، الگوریتم SIFT به جای استفاده مستقیم از تصاویر گاوسی، از **Difference of Gaussian (DoG)** بهره می‌برد که با محاسبه اختلاف دو تصویر محو شده در مقیاس‌های متفاوت به دست می‌آید. این روش سریع‌تر و از نظر محاسباتی کارا تر است. نقاطی که در فضای سه‌بعدی مقیاس-مکان (Scale-Space) به عنوان ماکزیمم یا مینیمم محلی ظاهر می‌شوند، به عنوان نقاط کلیدی نهایی انتخاب می‌گردند. به این ترتیب، Scale Space امکان می‌دهد ویژگی‌ها در برابر تغییر مقیاس مقاوم شوند و همین ویژگی یکی از دلایل قدرت بالای الگوریتم SIFT است.

در الگوریتم SIFT چندین متد به صورت مرحله‌ای به کار می‌روند تا نقاط کلیدی استخراج و توصیف شوند. مهم‌ترین متدهای مورد استفاده عبارت‌اند از:

1. ساخت فضای مقیاس (Scale-Space Construction)

برای مقاوم بودن نسبت به تغییر اندازه، تصویر در مقیاس‌های مختلف با فیلتر **Gaussian** محو (Blur) می‌شود. سپس با استفاده از متد **Difference of Gaussian (DoG)** اختلاف بین تصاویر محو شده محاسبه می‌گردد. این کار پایه‌ای برای شناسایی نقاط کلیدی پایدار است.

2. یافتن نقاط کلیدی (Keypoint Detection)

در فضای سه‌بعدی مقیاس-مکان، نقاطی که به عنوان ماکزیمم یا مینیمم محلی ظاهر می‌شوند به عنوان نقاط کلیدی انتخاب می‌شوند. این کار با متد **اکسترمم محلی (Local Extrema Detection)** انجام می‌شود. سپس با متدهای ریاضی مثل تقریب تیلور و محاسبه هسین (Hessian) نقاط ناپایدار (مانند لبه‌ها یا نقاط پر نویز) حذف می‌شوند.

3. اختصاص جهت (Orientation Assignment)

برای مقاوم بودن نسبت به چرخش، به هر نقطه کلیدی یک یا چند جهت غالب داده می‌شود. این کار با استفاده از متد

محاسبه هیستوگرام گرادیان (Gradient Orientation Histogram) در ناحیه اطراف نقطه کلیدی انجام می گیرد.

4. ایجاد توصیفگر (Descriptor Generation)

برای نمایش هر نقطه کلیدی، یک توصیفگر برداری ساخته می شود. این کار با تقسیم ناحیه اطراف نقطه کلیدی به سلول های کوچک و محاسبه توزیع گرادیان در هر سلول انجام می شود. متدی که در این مرحله استفاده می شود **Local Gradient Histogram** است. نتیجه نهایی یک بردار ۱۲۸ بعدی است که ویژگی های محلی تصویر را توصیف می کند.

به طور خلاصه، الگوریتم SIFT از **DoG** برای استخراج نقاط، **Local Extrema Detection** برای انتخاب پایدارترین نقاط، **Gradient Orientation** برای مقاوم بودن در برابر چرخش، و **Local Histogram Descriptors** برای ساخت بردار ویژگی استفاده می کند.

:Keypoints vs Descriptor

در الگوریتم SIFT، ابتدا نقاطی خاص به نام **Keypoints** شناسایی می شوند. این نقاط همان بخش هایی از تصویر هستند که پایدار و متمایز کننده اند، مانند گوشه ها، لبه ها یا نواحی بافت دار. هر **Keypoint** دارای ویژگی هایی مثل مختصات (x, y)، مقیاس و جهت غالب است. به کمک این اطلاعات، الگوریتم می تواند تضمین کند که حتی در صورت تغییر مقیاس یا چرخش تصویر، همان نقاط کلیدی دوباره شناسایی شوند.

پس از شناسایی **Keypoint** ها، مرحله ی بعدی ساخت **Descriptor** برای هر یک از آن ها است. **Descriptor** یک بردار عددی است) در SIFT معمولاً ۱۲۸ بعدی که ویژگی های محلی اطراف آن نقطه کلیدی را توصیف می کند. این توصیف بر اساس گرادیان ها و جهت های شدت روشنایی در اطراف **Keypoint** ساخته می شود. به کمک این بردارها می توان **Keypoint** های یک تصویر را با **Keypoint** های تصویر دیگر مقایسه و تطبیق داد. به بیان ساده، " **Keypoints** کجا را باید نگاه کنیم " را مشخص می کنند، و " **Descriptors** چه چیزی آنجاست " را توضیح می دهند.

```
bf = cv2.BFMatcher(cv2.NORM_L1, crossCheck = False)
```

این کد یک **Brute-Force Matcher** در OpenCV می سازد. وظیفه ی آن مقایسه ی **Descriptor** های استخراج شده (مثلاً از SIFT) بین دو تصویر است. معیار شباهت اینجا **L1 norm** یا همان فاصله ی منتهن است که با جمع قدر مطلق اختلاف عناصر دو بردار محاسبه می شود. همچنین چون **crossCheck=False** تنظیم شده، تطبیق یک طرفه کافی است؛ یعنی نیازی

insta: kahkeshani_mohammad

دوره پردازش تصویر و بینایی کامپیوتر با open cv

youtube: <https://www.youtube.com/@mohammadkahkeshani>

• مدرس محمد کهکشانی (مدرس رسمی دانشگاه هاروارد)

نیست که هر دو Descriptor همدیگر را به عنوان بهترین جفت انتخاب کنند، در نتیجه Matches بیشتری ولی با دقت کمتر به دست می آید.