

: HDR

HDR یا **High Dynamic Range** در عکاسی به تکنیکی گفته می‌شود که برای ثبت جزئیات بیشتر در قسمت‌های خیلی روشن و خیلی تاریک یک تصویر به کار می‌رود. چشم انسان توانایی بالایی در دیدن دامنه‌ی نوری گسترده دارد؛ به این معنی که همزمان می‌تواند جزئیات موجود در سایه‌ها و بخش‌های روشن را تشخیص دهد. اما سنسور دوربین‌ها چنین توانایی محدودی دارند و معمولاً یا نوردهی به سمت تاریکی‌ها تنظیم می‌شود یا به سمت روشنایی‌ها، و همین باعث از دست رفتن بخشی از جزئیات تصویر می‌گردد HDR راه‌حلی برای این مشکل است.

برای ایجاد یک تصویر HDR، عکاس معمولاً چند عکس با نوردهی‌های مختلف (Exposure) از یک صحنه می‌گیرد؛ مثلاً یکی با نوردهی کم برای ثبت بخش‌های روشن، یکی با نوردهی معمولی، و یکی با نوردهی زیاد برای جزئیات در سایه‌ها. سپس این عکس‌ها در نرم‌افزارهای ویرایش یا حتی مستقیماً در خود دوربین ترکیب می‌شوند تا تصویری با دامنه‌ی دینامیکی وسیع‌تر به دست آید. نتیجه آن است که در یک تصویر، هم جزئیات در بخش‌های روشن آسمان و هم در نقاط تاریک زمین دیده می‌شوند.

HDR می‌تواند بسیار طبیعی یا به‌طور اغراق‌شده استفاده شود. در حالت طبیعی، هدف فقط نزدیک کردن تصویر به آن چیزی است که چشم ما واقعاً می‌بیند. اما برخی عکاسان از HDR به شکل هنری بهره می‌برند و با تنظیمات افراطی، تصاویری فراواقعی و سورئال خلق می‌کنند. این نوع تصاویر به دلیل رنگ‌های شدید و کنتراست غیرمعمول، گاهی بیش از اندازه مصنوعی به نظر می‌رسند. بنابراین چگونگی استفاده از HDR بستگی زیادی به سبک عکاس دارد.

امروزه بسیاری از دوربین‌های دیجیتال و حتی گوشی‌های هوشمند، حالت HDR خودکار دارند. این قابلیت بدون نیاز به ویرایش دستی، چندین عکس با نوردهی مختلف می‌گیرد و بلافاصله آن‌ها را ترکیب می‌کند. استفاده درست از HDR به‌ویژه در عکاسی مناظر (Landscape) و صحنه‌هایی با تضاد نوری شدید، مانند عکاسی در برابر نور خورشید یا داخل ساختمان با پنجره‌های روشن، می‌تواند به خلق عکس‌هایی جذاب‌تر و نزدیک‌تر به تجربه واقعی چشم انسان کمک کند.



در کتابخانه **OpenCV** قابلیت‌های متنوعی برای کار با تصاویر HDR وجود دارد. همان‌طور که در عکاسی HDR چندین تصویر با نوردهی‌های مختلف گرفته می‌شود، در **OpenCV** نیز می‌توان این تصاویر را با استفاده از توابع خاص ترکیب کرد. این فرایند معمولاً شامل هم‌تراز کردن تصاویر (**Alignment**) برای جلوگیری از جابجایی و سپس ادغام آن‌ها برای تولید یک نگاشت HDR است. به این ترتیب، تصویر نهایی شامل دامنه‌ی دینامیکی گسترده‌تری خواهد بود.

یکی از ماژول‌های کلیدی در **OpenCV** برای این کار، ماژول **photo** است. این ماژول شامل الگوریتم‌هایی مانند `createMergeRobertson` و `createMergeDebevec` برای ساخت تصاویر HDR از چند نوردهی مختلف است. همچنین برای نمایش تصویر HDR روی مانیتورهای معمولی (که توانایی نمایش محدوده دینامیکی بالا را ندارند) باید از تکنیک **Tone Mapping** استفاده کرد. توابعی مانند `createTonemapDrago` یا `createTonemap` در **OpenCV** به همین منظور طراحی شده‌اند تا تصویر HDR به حالتی با دامنه دینامیکی استاندارد اما با جزئیات بیشتر در سایه و روشنایی تبدیل شود.

کاربرد HDR در **OpenCV** بسیار گسترده است؛ از بهبود کیفیت عکس‌های گرفته‌شده با دوربین گرفته تا پروژه‌های بینایی ماشین که نیازمند استخراج جزئیات در شرایط نوری دشوار هستند. برای مثال، در رباتیک یا پردازش تصاویر پزشکی، استفاده از HDR می‌تواند اطلاعاتی را که در تصاویر معمولی از دست می‌روند، آشکار کند. به این ترتیب، HDR در **OpenCV** نه تنها در عکاسی خلاقانه، بلکه در حوزه‌های علمی و صنعتی نیز اهمیت زیادی دارد.

Exposure:

مفهوم **Exposure** یا نوردهی در عکاسی به مقدار نوری گفته می‌شود که در مدت زمان مشخصی به سنسور یا فیلم دوربین می‌رسد. نوردهی تعیین می‌کند که تصویر نهایی چقدر روشن یا تاریک ثبت شود. اگر نوردهی بیش از حد باشد (**Overexposure**)، تصویر خیلی روشن شده و جزئیات بخش‌های روشن از بین می‌رود. برعکس، اگر نوردهی خیلی کم باشد (**Underexposure**)، تصویر تاریک می‌شود و جزئیات قسمت‌های سایه یا تیره از دست می‌رود.

نوردهی توسط سه عامل اصلی که به مثلث نوردهی (**Exposure Triangle**) معروف است کنترل می‌شود:

1. **دیافراگم (Aperture):** میزان باز یا بسته بودن دریچه‌ی لنز که مقدار نور ورودی را تعیین می‌کند.
2. **سرعت شاتر (Shutter Speed):** مدت زمانی که شاتر باز می‌ماند و نور به سنسور می‌رسد.
3. **حساسیت (ISO):** میزان حساسیت سنسور دوربین به نور.

با ترکیب درست این سه پارامتر، می‌توان نوردهی متعادل ایجاد کرد. در واقع **Exposure** تعیین می‌کند که تعادل بین روشنایی و تاریکی تصویر چگونه باشد و در عین حال بر عمق میدان (از طریق دیافراگم)، وضوح حرکت (از طریق شاتر) و میزان نویز از

طریق ISO نیز تأثیر مستقیم می‌گذارد. به همین دلیل، درک Exposure یکی از پایه‌ای‌ترین مفاهیم در یادگیری عکاسی به شمار می‌رود.

برای کار با HDR در open cv چند مرحله وجود دارد. ابتدا باید تصاویری که قرار است با هم ترکیب شوند را آماده کنیم.

```
img_fn = ["images/img_15.jpg", "images/img_2.5.jpg", "images/img_0.25.jpg",
"images/img_0.033.jpg"]
```

در اینجا نام چهار تصویر با نوردهی‌های مختلف Exposure متفاوت در یک لیست ذخیره شده. برای مثال:

- تصویر اول با نوردهی 15 ثانیه
- تصویر دوم با نوردهی 2.5 ثانیه
- تصویر سوم با نوردهی 0.25 ثانیه
- تصویر چهارم با نوردهی 0.0333 ثانیه

```
img_list = [cv2.imread(fn) for fn in img_fn]
```

اینجا با یک **list comprehension** هر تصویر از مسیر مربوطه توسط OpenCV (`cv2.imread`) خوانده می‌شود و همه در یک لیست ذخیره می‌شوند. در نهایت `img_list` شامل چهار تصویر است.

```
exposure_times = np.array([15.0, 2.5, 0.25, 0.0333], dtype=np.float32)
```

در این بخش زمان‌های نوردهی متناظر با تصاویر تعریف شده است. این مقادیر برای الگوریتم‌های HDR لازم هستند تا بدانند هر تصویر با چه مدت نور ثبت شده.

1 مفهوم مقادیر داخل آرایه

- این اعداد در واقع زمان نوردهی (Exposure Time) هر تصویر هستند.
- واحد آن‌ها ثانیه است.
- یعنی:

- تصویر اول با ۱۵ ثانیه نوردهی ثبت شده → نور زیادی وارد شده → تصویر روشن‌تر.
- تصویر دوم با ۲.۵ ثانیه نوردهی → کمی روشن.
- تصویر سوم با ۰.۲۵ ثانیه نوردهی → متوسط.
- تصویر چهارم با ۰.۰۳۳۳ ثانیه (تقریباً ۳۰/۱ ثانیه) → نوردهی خیلی کوتاه → تصویر تیره‌تر.

این تنوع در نوردهی همان چیزی است که برای ساخت HDR لازم داریم؛ چون هر تصویر بخشی از محدوده‌ی روشنایی صحنه را بهتر ثبت می‌کند.

چرا np.array استفاده شده؟

- این مقادیر به شکل یک آرایه‌ی NumPy تعریف می‌شوند تا بتوان آن‌ها را راحت‌تر در محاسبات عددی و الگوریتم‌های OpenCV به کار برد.
- NumPy سرعت بالا و سازگاری خوبی با توابع پردازش تصویر دارد.

دلیل انتخاب dtype=np.float32

- مقدار پیش‌فرض آرایه‌های NumPy ممکن است float64 یا انواع دیگر باشد.
- در اینجا صراحتاً float32 انتخاب شده، چون توابع HDR در OpenCV انتظار دارند که زمان نوردهی‌ها به شکل عدد اعشاری ۳۲-بیتی باشند.
- اگر نوع داده‌ی دیگری استفاده شود مثلاً int یا float64، ممکن است خطا رخ دهد یا الگوریتم به درستی کار نکند.

```
plt.figure(figsize=[18,4])
for i, (image, exposure) in enumerate(zip(img_list, exposure_times)):
    plt.subplot(1,4,i+1);plt.imshow(image[...::-1]);plt.title(exposure);plt.xticks([]);plt.yticks([]);
```

نمایش با matplotlib :

یک شکل (Figure) با ابعاد 4×18 ساخته می‌شود.

با حلقه‌ی for، هر تصویر به همراه زمان نوردهی‌اش (zip) پیمایش می‌شود.

plt.subplot(1,4,i+1) تصاویر را در یک ردیف و ۴ ستون قرار می‌دهد.

image[...::-1] رنگ‌ها را از BGR که فرمت پیش‌فرض OpenCV است به RGB فرمت مورد انتظار matplotlib تبدیل می‌کند.

عنوان هر تصویر (plt.title(exposure)) زمان نوردهی آن است.

plt.xticks([]); plt.yticks([]) باعث می‌شود محورهای X و Y نمایش داده نشوند.

```
merge_mertens = cv2.createMergeMertens()
res_mertens = merge_mertens.process(img_list)
```

مرحله بعد نوبت به merge کردن تصاویر می رسد:

دستور اول یک آجکت از الگوریتم **Merge Mertens** در OpenCV می سازد.

الگوریتم Mertens یک روش برای ترکیب چند تصویر با نوردهی متفاوت است.

بر خلاف روش های کلاسیک HDR مثل Debevec یا Robertson ، این الگوریتم به زمان نوردهی **Exposure Times** احتیاج ندارد.

در واقع، Mertens یک تکنیک **Exposure Fusion** است:

- چند عکس با نوردهی های مختلف گرفته می شوند.
- به جای ساختن یک فایل HDR واقعی با فرمت خاص مثل hdr یا exr ، تصاویر مستقیماً ترکیب می شوند.
- نتیجه یک تصویر معمولی (LDR) است که جزئیات سایه و روشنایی را همزمان حفظ می کند.

چرا از این تکنیک استفاده می کنیم؟ چونکه ممکن است بازه تصاویر بیش از 255 باشد اما opencv به به بازه ای حداکثر تا 255 احتیاج دارد بنابراین این تکنیک تصاویر را به یک شبه HDR قابل پردازش برای open cv تبدیل میکند.

پس از این عمل در خط دوم:

لیست تصاویر (img_list) به متد process() داده می شود.

الگوریتم Mertens روی این تصاویر اعمال می شود.

خروجی res_mertens یک تصویر ترکیبی است که ظاهر HDR دارد، ولی خودش HDR واقعی نیست چون محدود به بازه [0,1] است.

تصویر LDR :

تصویر **LDR (Low Dynamic Range)** به تصاویری گفته می شود که دامنه ی روشنایی محدودی دارند و معمولاً در قالب های رایجی مثل JPEG یا PNG ذخیره می شوند؛ در این تصاویر مقدار هر پیکسل در بازه ی ۰ تا ۲۵۵ قرار دارد، بنابراین توانایی ثبت جزئیات در نواحی بسیار روشن یا بسیار تاریک محدود است و بخشی از اطلاعات از بین می رود. برعکس، تصاویر HDR دامنه ی

دینامیکی بسیار وسیع‌تری دارند و می‌توانند همزمان جزئیات سایه و هایلایت را نمایش دهند، اما نیاز به فرمت‌ها و پردازش خاص دارند؛ به همین دلیل خروجی بسیاری از الگوریتم‌ها مانند **MergeMertens** در OpenCV به‌صورت یک تصویر LDR با ظاهر HDR ذخیره می‌شود تا روی نمایشگرهای معمولی قابل مشاهده باشد.

```
res_mertens_8bit = np.clip(res_mertens*255, 0, 255).astype('uint8')
```

خروجی الگوریتم **MergeMertens** در OpenCV به صورت یک تصویر **float32** است که مقادیر پیکسل‌هایش بین ۰ و ۱ قرار دارند. این نوع داده برای پردازش داخلی مناسب است، اما مستقیماً قابل ذخیره‌سازی به شکل یک فایل تصویری معمولی مثل JPEG یا PNG نیست. به همین دلیل ابتدا این مقادیر در ۲۵۵ ضرب می‌شوند تا به محدوده‌ی استاندارد روشنایی تصویر (۰ تا ۲۵۵) برسند. سپس با استفاده از `np.clip` تضمین می‌شود که هیچ مقداری خارج از این بازه قرار نگیرد؛ یعنی مقادیر منفی صفر شوند و مقادیر بالاتر از ۲۵۵ به ۲۵۵ محدود شوند.

در نهایت با دستور `astype('uint8')` نوع داده‌ی آرایه از اعشاری ۳۲ بیتی (**float32**) به صحیح ۸ بیتی بدون علامت (**uint8**) تغییر می‌کند. این تبدیل ضروری است زیرا اغلب فرمت‌های تصویری و توابع ذخیره‌سازی/نمایش (مانند `plt.imshow` یا `cv2.imwrite`) تنها داده‌های **uint8** در محدوده‌ی ۰ تا ۲۵۵ را پشتیبانی می‌کنند. به این ترتیب، تصویر ترکیب‌شده‌ی **Mertens** به فرمتی استاندارد و قابل استفاده در نمایشگرها و نرم‌افزارهای رایج تبدیل می‌شود.