

: Panorama

عکس پانوراما نوعی تصویر عریض است که زاویه دید بسیار بیشتری نسبت به عکس‌های معمولی پوشش می‌دهد. این عکس‌ها معمولاً با کنار هم قرار دادن چندین تصویر گرفته‌شده از یک صحنه ساخته می‌شوند تا چشم‌اندازی وسیع‌تر از آنچه در یک فریم دوربین جا می‌گیرد به دست آید. نتیجه کار تصویری است که می‌تواند تا ۳۶۰ درجه یا حتی بیشتر از محیط اطراف را نمایش دهد و تجربه‌ای نزدیک‌تر به دید طبیعی انسان ایجاد کند.

روش‌های مختلفی برای ایجاد عکس پانوراما وجود دارد. ساده‌ترین راه، استفاده از قابلیت داخلی دوربین‌های دیجیتال یا گوشی‌های هوشمند است که هنگام حرکت آرام دوربین، چند تصویر را به‌طور خودکار ثبت و سپس به هم متصل می‌کند. در روش‌های پیشرفته‌تر، عکاس چندین عکس با هم‌پوشانی کافی می‌گیرد و سپس با نرم‌افزارهای ویرایش عکس مانند Photoshop یا PTGui آن‌ها را به صورت دقیق و بدون خطا به یک تصویر یکپارچه تبدیل می‌کند. تکنیک‌هایی مثل تصحیح پرسپکتیو و تنظیم نور هم نقش مهمی در طبیعی شدن نتیجه دارند.

کاربرد عکس پانوراما بسیار گسترده است؛ از ثبت مناظر طبیعی و شهری گرفته تا معماری داخلی و حتی عکاسی نجومی. پانوراما به عکاس اجازه می‌دهد جزئیاتی را نشان دهد که در قاب معمولی جا نمی‌گیرند، و به همین دلیل برای ایجاد حس غوطه‌وری (Immersion) در صحنه بسیار محبوب است. این نوع عکس‌ها در پروژه‌های گردشگری، طراحی معماری و حتی واقعیت مجازی (VR) هم به کار می‌روند و تجربهٔ بصری منحصر به فردی برای مخاطب فراهم می‌کنند.

کتابخانه glob در پایتون:

کتابخانه‌ی **glob** در پایتون یکی از ماژول‌های استاندارد است که برای جست‌وجو و یافتن فایل‌ها و پوشه‌ها بر اساس الگوهای متنی (wildcards) استفاده می‌شود. این کتابخانه به شما اجازه می‌دهد با الگوهای مثل * برای هر تعداد کاراکتر، ? برای یک کاراکتر یا [] برای مجموعه‌ای از کاراکترها به راحتی مسیر فایل‌ها را فیلتر کنید. برای مثال، با دستور `glob.glob("**/*.py")` می‌توان تمام فایل‌های پایتون موجود در پوشه فعلی را لیست کرد یا با الگوی `recursive=True` همراه با پارامتر `**/*.txt` همه فایل‌های متنی در پوشه و زیرپوشه‌ها را به دست آورد. این ابزار ساده و سریع است و در بسیاری از اسکریپت‌ها برای مدیریت فایل‌ها و پردازش دسته‌جمعی آن‌ها کاربرد دارد.

```
# Read Images

imagefiles = glob.glob("images/boat*.jpg")
imagefiles.sort()

images = []
for filename in imagefiles:
```

```
img = cv2.imread(filename)
images.append(img)

num_images = len(images)
num_images
```

این کد ابتدا با استفاده از کتابخانه‌ی **glob** همه فایل‌هایی را که در پوشه‌ی `images` قرار دارند و نام آن‌ها با `boa` شروع شده و پسوند `.jpg` دارند پیدا می‌کند. سپس با دستور `sort()` آن‌ها را مرتب می‌کند تا ترتیب فایل‌ها مشخص و ثابت باشد مثلاً `boat1.jpg, boat2.jpg,` به این شکل برنامه مطمئن می‌شود که تصاویر به همان ترتیب دلخواه بارگذاری خواهند شد و هیچ اختلالی در ترتیب پردازش پیش نمی‌آید.

در ادامه یک لیست خالی به نام `images` ساخته می‌شود و با یک حلقه، هر نام فایل از لیست خوانده شده و با استفاده از تابع `cv2.imread()` از کتابخانه **OpenCV** به عنوان یک تصویر بارگذاری می‌شود. هر تصویر خوانده شده به لیست `images` اضافه می‌گردد. در پایان، تعداد کل تصاویری که موفق به بارگذاری شده‌اند با دستور `len(images)` محاسبه می‌شود و در متغیر `num_images` ذخیره می‌شود. این مقدار معمولاً برای بررسی تعداد ورودی‌ها یا استفاده در مراحل بعدی پردازش تصاویر به کار می‌رود.

```
# Display Images
plt.figure(figsize=[30,10])
num_cols = 3
num_rows = math.ceil(num_images / num_cols)
for i in range(0, num_images):
    plt.subplot(num_rows, num_cols, i+1)
    plt.axis('off')
    plt.imshow(images[i][...,:-1])
```

این قطعه کد برای نمایش مجموعه‌ای از تصاویر در قالب یک جدول (`grid`) با استفاده از کتابخانه‌ی **Matplotlib** نوشته شده است. در خط اول، با `plt.figure(figsize=[30,10])` یک فضای نمایش با ابعاد بزرگ تعریف می‌شود تا تصاویر با کیفیت و اندازه مناسب در کنار هم نمایش داده شوند. سپس تعداد ستون‌ها برابر با ۳ تعیین شده و تعداد ردیف‌ها (`num_rows`) بر اساس تقسیم تعداد کل تصاویر (`num_images`) بر تعداد ستون‌ها محاسبه می‌شود. چون ممکن است تعداد تصاویر مضرب دقیقی از ۳ نباشد، از تابع `math.ceil` استفاده شده تا تعداد ردیف‌ها به سمت بالا گرد شود و همه تصاویر جا بگیرند.

در ادامه، با یک حلقه `for` روی همه تصاویر پیمایش انجام می‌شود. در هر تکرار، دستور `plt.subplot(num_rows, num_cols, i+1)` مشخص می‌کند که تصویر باید در کدام جایگاه شبکه

(grid) نمایش داده شود. مقدار $i+1$ برای تعیین اندیس مکان تصویر در جدول ضروری است، چون اندیس گذاری subplot از ۱ (و نه ۰) شروع می شود.

برای جلوگیری از شلوغی، دستور `plt.axis('off')` محورها را حذف می کند تا فقط خود تصویر نمایش داده شود. سپس با `plt.imshow(images[i][..., :-1])` تصویر به نمایش درمی آید. بخش `[..., :-1]` برای معکوس کردن کانال های رنگی استفاده شده است، چون OpenCV تصاویر را در قالب BGR می خواند، اما Matplotlib انتظار فرمت RGB دارد. با این کار رنگ ها به درستی نمایش داده می شوند و تصاویر دقیقاً همان طور که باید دیده می شوند.

```
# Stitch Images
stitcher = cv2.Stitcher_create()
status, result = stitcher.stitch(images)
if status == 0:
    plt.figure(figsize=[30,10])
    plt.imshow(result[..., :-1])
```

این قطعه کد مربوط به دوختن یا ترکیب چند تصویر (Image Stitching) برای ایجاد یک تصویر پانوراما است. ابتدا با دستور `cv2.Stitcher_create()` یک شیء از کلاس `Stitcher` در کتابخانه OpenCV ساخته می شود. این شیء ابزار اصلی برای فرآیند دوختن تصاویر است و الگوریتم های داخلی OpenCV را برای هم تراز کردن (alignment) و ادغام (blending) تصاویر به کار می گیرد. این مرحله اساس کار را آماده می کند تا مجموعه تصاویر خوانده شده در مراحل قبلی بتوانند به یک تصویر واحد تبدیل شوند.

در خط بعدی، دستور `stitcher.stitch(images)` روی لیست تصاویر اعمال می شود. این تابع تلاش می کند نقاط کلیدی (keypoints) مشترک بین تصاویر را پیدا کند، آن ها را تطبیق دهد و سپس تصاویر را به گونه ای به هم بچسباند که یک چشم انداز پیوسته ایجاد شود. خروجی این تابع شامل دو مقدار است: یکی `status` که وضعیت عملیات را نشان می دهد و دیگری `result` که در صورت موفقیت، تصویر نهایی پانوراما خواهد بود.

کد سپس بررسی می کند که آیا `status` برابر با صفر است یا خیر. در OpenCV، مقدار صفر به معنی موفقیت آمیز بودن عملیات است. اگر وضعیت برابر با صفر باشد، یعنی تصاویر با موفقیت کنار هم قرار گرفته اند و یک تصویر کامل ساخته شده است. در غیر این صورت، معمولاً عددی غیر از صفر برمی گردد که نشان دهنده وجود خطاهایی مانند عدم همپوشانی کافی بین تصاویر یا مشکلات در شناسایی ویژگی های مشترک است.

در نهایت، اگر عملیات موفقیت آمیز بود، تصویر نهایی با استفاده از Matplotlib نمایش داده می شود. ابتدا با `plt.figure(figsize=[30,10])` یک بوم بزرگ ایجاد می شود و سپس دستور `plt.imshow(result[..., :-1])` تصویر را نمایش می دهد. همان طور که در کدهای قبلی هم دیده بودیم،

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)

[1:-,:,:] برای تبدیل فرمت رنگی از BGR فرمت پیش فرض (OpenCV به RGB) فرمت مورد انتظار Matplotlib به کار می رود تا رنگ ها به شکل طبیعی و صحیح دیده شوند. به این ترتیب، خروجی نهایی یک عکس پانورامای ترکیب شده از چند تصویر مجزا خواهد بود.