

## ApproxPoly

الگوریتم `approxPolyDP` در کتابخانه OpenCV برای تقریب زدن یک منحنی یا چندضلعی به شکل چندضلعی ساده‌تر استفاده می‌شود. هدف اصلی این الگوریتم کاهش تعداد نقاط موجود در یک کانتور (contour) یا منحنی، بدون تغییر زیاد شکل کلی آن است.

توضیح کلی الگوریتم:

- ورودی اصلی: یک آرایه از نقاط که یک منحنی یا کانتور را مشخص می‌کند.
- یک پارامتر کلیدی به نام `epsilon` که میزان دقت تقریب را تعیین می‌کند. این مقدار نشان‌دهنده‌ی بیشترین فاصله‌ای است که نقاط تقریبی می‌توانند از منحنی اصلی دور شوند. هر چه `epsilon` کوچکتر باشد، تقریب به شکل اصلی نزدیک‌تر است و تعداد نقاط بیشتر می‌ماند. بالعکس، `epsilon` بزرگتر باعث کاهش نقاط بیشتر و تقریب ساده‌تر می‌شود.
- نوع تقریب: این الگوریتم از روش داینو (Douglas-Peucker) برای حذف نقاط اضافی استفاده می‌کند.

عملکرد الگوریتم:

1. ابتدا و انتهای کانتور حفظ می‌شود.
2. فاصله هر نقطه از خط مستقیم بین دو نقطه حفظ شده اندازه‌گیری می‌شود.
3. اگر فاصله نقطه‌ای بیشتر از `epsilon` باشد، آن نقطه به عنوان یک نقطه کلیدی نگه داشته می‌شود.
4. این روند بازگشتی ادامه پیدا می‌کند تا همه نقاط کانتور با دقت مشخص شده، تقریب زده شوند.

کاربردها:

- ساده‌سازی شکل‌ها و خطوط پیچیده برای کاهش حجم داده‌ها.
- استخراج چندضلعی‌های ساده‌تر از کانتورهای اشیاء در تصویر (مثل یافتن مستطیل یا مثلث).
- پیش‌پردازش در تشخیص اشکال هندسی.

ورودی و خروجی:

- ورودی: کانتور به صورت آرایه‌ای از نقاط (معمولاً نقاط مرز یک شیء در تصویر).
- خروجی: آرایه‌ای از نقاط تقریب زده شده که تعدادشان کمتر است و شکل کلی حفظ شده است.

برای استفاده از این الگوریتم و تخمین کانتورها می توان به روش زیر عمل نمود. ابتدا اولین کانتور پیدا شده را در یک متغیر ذخیره می کنیم. این موضوع بسته به عکس و تعداد کانتورها متفاوت است چراکه ما لیستی از کانتورها را در اختیار داریم. اگر بخواهیم از کانتورهای دیگر استفاده کنیم میتوانیم از ویژگی های خاصی مثل بزرگترین کانتور، کوچکترین کانتور یا سایر ویژگی ها استفاده کنیم.

```
cnt = contours[0]
```

در مرحله بعد باید محیط کانتور را محاسبه کنیم. در این متد ورودی True به معنی بسته بودن کانتور است

```
perimeter = cv2.arcLength(cnt, True)
```

epsilon یکی از مهمترین پارامترهای الگوریتم approxPolyDP است و وظیفه آن مشخص نمود میزان تقریب یک منحنی است. به عبارتی این پارامتر میزان خطای مجاز یا حداکثر فاصله بین شکل اصلی و شکل تقریب زده شده را نشان می دهد. اگر فرض کنیم که الگوریتم approxPolyDP مجموعه ای از نقاط روی یک منحنی را با چند نقطه کمتر و به شکل یک منحنی تقریب می زند پارامتر epsilon مشخص می کند که هر نقطه جدید چه مقدار می تواند از شکل اصلی دور باشد تا در شکل اصلی قرار نگیرد. در این مثال ما میزان این پارامتر را 10 درصد خطا و 1 درصد خطا در نظر گرفته ایم.

اثرات مقدار epsilon:

- اگر epsilon کوچک باشد:
  - تقریب دقیق تر است.
  - تعداد نقاط بیشتر حفظ می شود.
  - شکل نهایی بسیار شبیه شکل اصلی است.
  - پیچیدگی و جزئیات شکل حفظ می شود.
- اگر epsilon بزرگ باشد:
  - تقریب ساده تر است.
  - تعداد نقاط کمتر می شود.
  - شکل نهایی ساده و هندسی تر می شود.
  - جزئیات ریز شکل حذف می شود و فقط شکل کلی حفظ می شود.

فرض کن یک مسیر پیچ در پیچ با 1000 نقطه داری:

- با epsilon کوچک، تقریب زده شده ممکن است 900 نقطه داشته باشد و مسیر تقریباً همان مسیر اصلی باشد.

- با epsilon بزرگ، شاید فقط 10 نقطه باقی بماند که مسیر را به شکل چندضلعی خیلی ساده تبدیل می کند.

```
epsilon = 0.1 * cv2.arcLength(cnt, True)
```

پس از تعیین میزان خطا نوبت به استفاده از متد اصلی است

```
approx = cv2.approxPolyDP(cnt, epsilon, True)
```

این تابع برای گرفتن یک کانتور پیچیده (یک منحنی بسته یا باز که از نقاط تشکیل شده) و تبدیل آن به یک کانتور ساده تر و کم نقطه تر به کار می رود، به طوری که شکل کلی تقریباً حفظ شود.

ورودی ها

### 1. cnt

- ورودی اول، آرایه ای از نقاط کانتور است.
- کانتور یعنی یک لیست از نقاط (x, y) که دور یک شیء یا شکل را در تصویر مشخص می کنند.
- می تواند یک شکل پیچیده باشد با تعداد زیاد نقاط.

### 2. epsilon

- پارامتر مهم که تعیین می کند چقدر اجازه داریم شکل کانتور اصلی را تغییر بدهیم تا ساده شود.
- به معنای بیشترین فاصله مجاز بین شکل اصلی و شکل تقریب زده شده است.
- هر چه این مقدار کمتر باشد، شکل خروجی به شکل اصلی نزدیک تر است و تعداد نقاط بیشتر حفظ می شود.
- هر چه مقدار بیشتر باشد، شکل ساده تر و تعداد نقاط کمتر می شود.
- معمولاً به صورت درصدی از طول محیط کانتور (arcLength) انتخاب می شود.

### 3. True

- این پارامتر نشان دهنده بسته بودن کانتور است.
- True یعنی کانتور یک شکل بسته (مانند چندضلعی) است.
- False یعنی کانتور باز است (مثلاً یک منحنی باز).
- اگر کانتور بسته باشد، تابع فرض می کند ابتدا و انتهای کانتور به هم وصل هستند.

عملکرد داخلی تابع

- این تابع بر پایه الگوریتم داگلاس-پکر (Douglas-Peucker) کار می کند که یک الگوریتم معروف برای کاهش نقاط در منحنی هاست.

- الگوریتم ابتدا نقاط ابتدایی و انتهایی کانتور را نگه می‌دارد.
- سپس نقاطی که بیشترین فاصله را از خط مستقیم بین این دو نقطه دارند پیدا می‌کند.
- اگر فاصله این نقاط بزرگ‌تر از  $\epsilon$  باشد، آن نقاط به لیست نقاط حفظ شده اضافه می‌شوند و فرآیند بازگشتی انجام می‌شود.
- اگر فاصله کمتر باشد، آن نقاط حذف می‌شوند (یعنی می‌توانند با خط مستقیم بین نقاط دیگر جایگزین شوند).
- نتیجه، چندضلعی ساده‌تر با تعداد کمتر نقطه است که شکل اصلی را تقریباً حفظ کرده است.

### خروجی

- خروجی `approx`. آرایه‌ای از نقاط کانتور است که تقریب زده شده است.
- این آرایه نسبت به `cnt` معمولاً تعداد نقاط کمتری دارد.
- این نقاط نشان‌دهنده چندضلعی ساده‌ای هستند که می‌تواند شکل شیء اصلی را نمایش دهد.

### موارد استفاده:

- برای ساده کردن شکل‌ها و کاهش پیچیدگی داده‌ها.
- برای شناسایی اشکال هندسی (مثل مثلث، مربع، مستطیل و غیره) با تعداد رأس مشخص.
- کاهش حجم داده برای پردازش سریع‌تر.
- حذف نویز و نقاط اضافی روی کانتور.