

EKSTRAKSI KONTUR

Kontur menyatakan kurva yang menghubungkan piksel-piksel yang mempunyai intensitas atau warna yang sama.

Listing program **kontur**

```
1  # Contoh untuk mendapatkan tepi objek menggunakan findContours()
2
3  import cv2
4  import numpy as np
5
6  # Baca citra biner huruf_bin.png
7  citra = cv2.imread("C:\Pandas_Experiment\golf.JPG", 0)
8
9  # Bentuk citra berwarna dengan latar belakang hitam
10 jumBaris = citra.shape[0]
11 jumKolom = citra.shape[1]
12 citraKontur = np.zeros((jumBaris, jumKolom, 3),
13                        np.uint8)
14
15 # Memperoleh kontur
16 citra2, kontur, hierarki = cv2.findContours(citra,
17      cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_NONE)
18
19 # Tambahkan kontur
20 cv2.drawContours(citraKontur, kontur, -1,
21      (230, 216, 173), 2)
22
23 # Tampilkan hasilnya
24 citraRGB = cv2.merge((citra, citra, citra))
25 hasil = np.hstack((citraRGB, citraKontur))
26 cv2.imshow("Hasil", hasil)
27 cv2.waitKey()
```

Kontur dapat ditampilkan dengan menggunakan warna yang berbeda. Listing programnya sebagai berikut :

```
1  # Pengaturan warna kontur
2
3  import cv2
4  import numpy as np
5
6  # Baca citra biner huruf_bin.png
7  citra = cv2.imread("huruf_bin.png", 0)
8
9  # Bentuk citra berwarna dengan latar belakang hitam
10 jumBaris = citra.shape[0]
11 jumKolom = citra.shape[1]
12 citraKontur = np.zeros((jumBaris, jumKolom, 3),
13                        np.uint8)
14
15 # Peroleh kontur
16 citra2, kontur, hierarki = cv2.findContours(citra,
17      cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_NONE)
18
19 # Tambahkan kontur dengan warna berlainan
20 warna = [(230, 216, 173), (255, 255, 255),
21      (255, 255, 0), (255, 0, 255),
22      (0, 255, 255)]
23
24 for indeks in range(len(kontur)):
25     cv2.drawContours(citraKontur, kontur, indeks,
26         warna[indeks % 5], 2)
27
28 # Tampilkan hasilnya
29 citraRGB = cv2.merge((citra, citra, citra))
30 hasil = np.hstack((citraRGB, citraKontur))
31 cv2.imshow("Hasil", hasil)
32 cv2.waitKey()
```

PENGHAMPIRAN KURVA POLIGON

Suatu kontur dapat disederhanakan sehingga kurva yang dihasilkan mengandung titik-titik yang lebih sedikit. Misalkan objek berbentuk persegi panjang dapat diwakili oleh empat titik. Contoh implementasinya sebagai berikut :

```
1  # Contoh penggunaan approxPolyDB()
2
3  import cv2
4
5  # Baca citra biner
6  citra = cv2.imread("guppi-5.png", 0)
7
8  citraBerwarna = cv2.merge((citra, citra, citra))
9
10 # Peroleh kontur
11 citra2, kontur, hierarki = cv2.findContours(citra,
12      cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_NONE)
13
14 print("Jumlah piksel semula:", len(kontur[0]))
15
16 epsilon = 0.01 * cv2.arcLength(kontur[0], True)
17 approx = cv2.approxPolyDP(kontur[0], epsilon, True)
18
19 print("Jumlah piksel sekarang:", len(approx))
20
21 cv2.drawContours(citraBerwarna, approx, -1,
22      (255, 0, 255), 8)
23
24 cv2.imshow("Hasil", citraBerwarna)
25 cv2.waitKey()
```

PERIMETER

Perimeter sering digunakan sebagai bahan untuk mendapatkan fitur suatu objek. Perimeter atau keliling menyatakan panjang tepi suatu objek. Listing program untuk menentukan perimeter sebagai berikut :

```
1  # Perimeter
2
3  import cv2
4
5  # Baca citra biner
6  citra = cv2.imread("guppi-5.png", 0)
7
8  # Peroleh kontur
9  citra2, kontur, hierarki = cv2.findContours(citra,
10      cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_NONE)
11
12 # Peroleh perimeter
13 keliling = cv2.arcLength(kontur[0], True)
14
15 print("Keliling = ", keliling)
```

MENGHITUNG LUAS

Luas menyatakan jumlah piksel yang terkandung dalam suatu objek. Untuk menghitung luas dapat dilakukan sebagai berikut :

```
1  # Perhitungan Luas
2
3  import cv2
4
5  # Baca citra biner
6  citra = cv2.imread("guppi-5.png", 0)
7
8  # Peroleh kontur
9  citra2, kontur, hierarki = cv2.findContours(citra,
10      cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_NONE)
11
12 # Peroleh Luas
13 luas = cv2.contourArea(kontur[0], False)
14
15 print("Luas = ", luas)
```

KOTAK PEMBATA

Kotak pembatas (bounding box) adalah kotak terkecil yang dapat melingkupi sebuah objek. Kotak pembatas ada 2 jenis yaitu kotak pembatas langsung yaitu kotak pembatas yang berorientasi citra dan kotak pembatas terputar yaitu kotak pembatas yang berorientasi objek.

```
1  # Kotak pembatas
2
3  import cv2
4  import numpy as np
5
6  # Baca citra biner
7  citra = cv2.imread("guppi-5.png", 0)
8
9  citraBerwarna = cv2.merge((citra, citra, citra))
10
11 # Peroleh kontur
12 citra2, kontur, hierarki = cv2.findContours(citra,
13      cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_NONE)
14
15 # Peroleh kotak pembatas langsung
16 x, y, lebar, tinggi = cv2.boundingRect(kontur[0])
17
18 # Gambar kotak pembatas
19 cv2.rectangle(citraBerwarna, (x, y),
20      (x + lebar, y + tinggi),
21      (255, 255, 0), 2)
22
23 # Peroleh kotak pembatas terputar
24 persegiPanjang = cv2.minAreaRect(kontur[0])
25 kotak = cv2.boxPoints(persegiPanjang)
26 kotak = np.int32(kotak)
27 cv2.drawContours(citraBerwarna, [kotak], 0,
28      (128, 128, 128), 2)
29
30 # Tampilkan hasilnya
31 cv2.imshow("Hasil", citraBerwarna)
32 cv2.waitKey()
```

LINGKARAN PEMBATAS

Lingkaran pembatas merupakan lingkaran terkecil yang melingkupi suatu objek.

```
1 # Lingkaran pembatas
2
3 import cv2
4 import numpy as np
5
6 # Baca citra biner
7 citra = cv2.imread("guppi-5.png", 0)
8
9 citraBerwarna = cv2.merge((citra, citra, citra))
10
11 # Peroleh kontur
12 citra2, kontur, hierarki = cv2.findContours(citra,
13      cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_NONE)
14
15 # Peroleh lingkaran pembatas
16 (x, y), radius = cv2.minEnclosingCircle(kontur[0])
17
18 pusat = (int(x), int(y))
19 radius = int(radius)
20 cv2.circle(citraBerwarna, pusat,
21      radius,
22      (255, 255, 0), 2)
23
24 # Tampilkan hasilnya
25 cv2.imshow("Hasil", citraBerwarna)
26 cv2.waitKey()
```

ELLIPS

Ellips untuk objek merupakan ellips di atas objek dan tidak benar-benar untuk ellips yang membatasi objek.

```
1 # Elips untuk objek
2
3 import cv2
4 import numpy as np
5
6 # Baca citra biner
7 citra = cv2.imread("guppi-5.png", 0)
8
9 citraBerwarna = cv2.merge((citra, citra, citra))
10
11 # Peroleh kontur
12 citra2, kontur, hierarki = cv2.findContours(citra,
13      cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_NONE)
14
15 # Peroleh elips
16 elips = cv2.fitEllipse(kontur[0])
17 cv2.ellipse(citraBerwarna, elips,
18      (255, 255, 0), 2)
19
20 # Tampilkan hasilnya
21 cv2.imshow("Hasil", citraBerwarna)
22 cv2.waitKey()
```

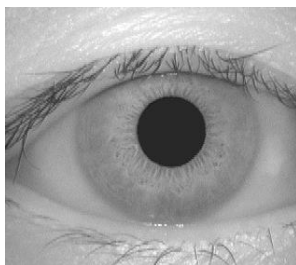
CONVEX HULL

Convex hull dapat dibayangkan seperti karet gelang yang melingkupi suatu benda.

```
1 # Convex hull
2
3 import cv2
4
5 # Baca citra biner huruf_bin.png
6 citra = cv2.imread("guppi-5.png", 0)
7
8 citraBerwarna = cv2.merge((citra, citra, citra))
9
10 # Peroleh kontur
11 citra2, kontur, hierarki = cv2.findContours(citra,
12      cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_NONE)
13
14 # Peroleh convex hull
15 convex = cv2.convexHull(kontur[0])
16
17 # Tampilkan convex hull
18 cv2.drawContours(citraBerwarna, [convex], -1,
19      (255, 255, 0), 2)
20
21 # Tampilkan hasilnya
22 cv2.imshow("Hasil", citraBerwarna)
23 cv2.waitKey()
```

TRANSFORMASI HOUGH LINGKARAN

Gunakanlah citra mata berikut ini untuk mendeteksi lingkaran iris mata



```
from matplotlib import pyplot as plt
import numpy as np
import cv2

gambar = cv2.imread('C:\Python_Experiment\irismata.JPG', 0)
cimg = cv2.cvtColor(gambar, cv2.COLOR_GRAY2BGR)
canny = cv2.Canny(cimg, 50, 50)

circles = cv2.HoughCircles(canny, cv2.HOUGH_GRADIENT, 1, 1000,
      param1=10, param2=15, minRadius=20, maxRadius=40)
circles = np.uint16(np.around(circles))
for i in circles[0,:]:
    # draw the outer circle
    cv2.circle(cimg, (i[0], i[1]), i[1], (0, 255, 0), 2)
    # draw the center of the circle
    cv2.circle(cimg, (i[0], i[1]), 2, (10, 10, 255), 3)
plt.imshow(cimg, cmap = 'gray')
plt.show()
```