

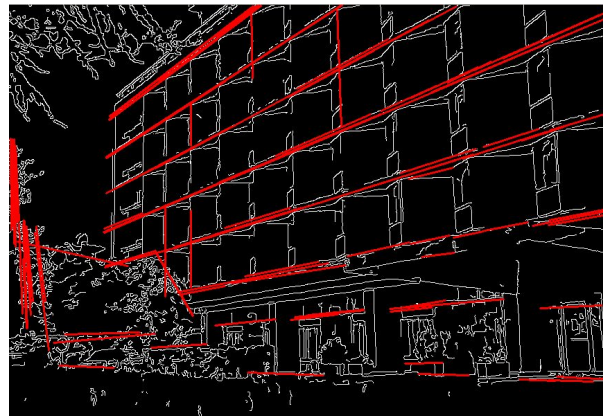
Extracción de características y clasificación

Extracción básica

- Área
- Perímetro
- Centroide
- Radios (Medidas de tendencia central)
- Cuerdas
- Cantidad de: esquinas, huecos

Extracción avanzada

- Shape features based on Hough transform
- Gabor features (textures)
- Haar-Like features (face detection)
- Histogram of oriented gradients - HOG description (People detection)



Extracción de área y centroide

M = cv2.moments(imagen)

$$M_{pq} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x^p y^q f(x, y) dx dy$$

Los momentos son una medida tanto de la estadística como de la física mecánica, asociados a fenómenos de naturaleza propia de cada área.

Extracción de área y centroide

$$M = \text{cv2.moments(imagen)} \quad M_{pq} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x^p y^q f(x, y) dx dy$$

Cuando p y q son 0 la fórmula cambiaría a $M_{00} = \iint f(x,y) dx dy$. Este Momento es conocido como el Momento central y en física mecánica se refiere a la masa (explicación), y representaría algo similar en la visión artificial, por otro lado en estadística representa la probabilidad total.

Extracción de área y centroide

$$M = \text{cv2.moments(imagen)} \quad M_{pq} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x^p y^q f(x, y) dx dy$$

Cuando p y q son 1 y 0, la fórmula cambiaría a $M_{10} = \iint x f(x, y) dx dy$. En física este momento permite calcular el centroide en X del objeto, se calcula como: centro x = M_{10}/M_{00} , de forma similar se utiliza en la visión artificial para calcular la ubicación del centroide, por otro lado en estadística representa el promedio

Extracción de área y centroide

$$M = \text{cv2.moments(imagen)} \quad M_{pq} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x^p y^q f(x, y) dx dy$$

Una vez se aplica moments y se almacena el resultado en una variable (M en este caso), es posible acceder a la información específica de M usando [nombre].

Ejemplo

`A = M["m00"]`

Extracción de contornos

Es altamente utilizado en el procesamiento de imágenes la detección de píxeles conectados para encontrar el contorno de uno o varios objetos.

Una vez el contorno es encontrado este se almacena en un arreglo 2D de puntos, este arreglo contiene la información de los puntos relevantes para representar el contorno y son además la base para otras funciones

Extracción de contornos

Es posible extraer los contornos con la función:

```
cv2.findContours(imagen, mode=cv2.RETR_EXTERNAL ,  
method=cv2.CHAIN_APPROX_SIMPLE).
```

Los contornos quedan almacenados en el segundo parámetro retornado.

El modo RETR_EXTERNAL permite que se encuentren sólo los contornos externos y por lo tanto se ignoran los contornos internos o huecos

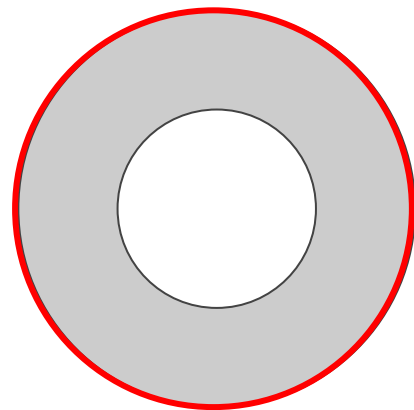
Extracción de contornos

`mode=cv2.RETR_EXTERNAL`

Si la imagen es como la del ejemplo sólo se almacenarían los contornos asociados al borde rojo.

Es importante entender que almacena los contornos de TODOS los objetos de la imagen, por lo tanto si sólo existe un objeto en la imagen se debe buscar el elemento en la posición 0. Algo como: `contorno = contornos[0]`

Donde `contornos` es el segundo valor retornado por la función `findContours`



Extracción de las distancias radiales

Conociendo las posiciones $[x_i, y_i]$ de cada punto del contorno y la posición del centroide $[x_0, y_0]$, es posible calcular las distancias radiales a través de la ecuación de la distancia euclidiana

Extracción de perímetro

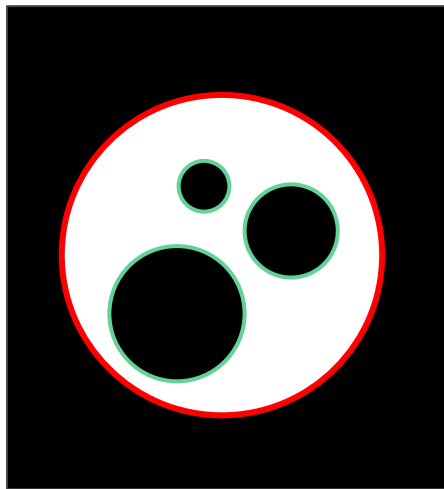
Teniendo los contornos de la imagen es posible:

- Encontrar un rectángulo que se ajusta a su forma con: `cv2.boundingRect`
- Encontrar el perímetro con `cv2.arcLength`
- Es posible rellenar los huecos de la imagen de forma más precisa que con una dilatación usando `cv2.drawContours`

Extracción de la cantidad de huecos

Es posible además encontrar la cantidad de huecos reemplazando el modo de detección de contornos por: `cv2.RETR_CCOMP`. En este caso, y suponiendo que sólo existe UN único objeto, el primer contorno detectado es el del objeto más grande y los siguientes contornos son los contornos interiores.

En una imagen como la del ejemplo `contornos[0]` contendría los contornos del objeto con bordes rojos y `contornos[1]`, `contornos[2]`, `contornos[3]` contendrían la información de los huecos de borde verde



Clasificación

- Clasificación básica (umbrales)
- Clasificadores estadísticos (supervisados y no supervisados)
- Clasificadores basados en aprendizaje de máquina (supervisados y no supervisados)

Clasificación por umbrales

Se establecen umbrales que deben cumplir las características extraídas para pertenecer a un grupo u a otro.

Ejemplo: se quiere clasificar entre objetos pequeños, medianos y grandes

- Si el área es menor a 20: el objeto es pequeño
- Si el área está entre 20 y 40: el objeto es mediano
- Si el área es mayor a 40: el objeto es grande

Clasificación por umbrales

Se establecen umbrales que deben cumplir las características extraídas para pertenecer a un grupo u a otro.

Ejemplo: se quiere clasificar entre círculos y no círculos

1. Se extraen las distancias radiales y se obtiene la desviación estándar de estas
2. Si la desviación es menor a 10 (ejemplo) se dice que es un círculo (¿por qué?), en los casos opuestos se dice que no es un círculo

Clasificación por umbrales

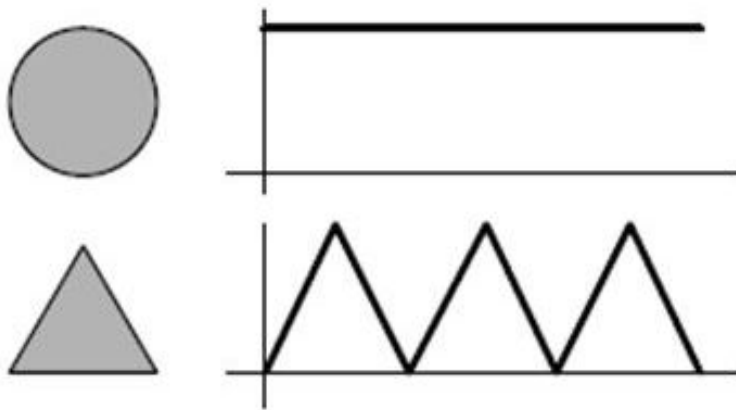
Se establecen umbrales que deben cumplir las características extraídas para pertenecer a un grupo u a otro.

Ejemplo: se quiere clasificar entre estrellas y no estrellas

1. Se calcula el perímetro y el área
2. Se calcula $c = p^2/A$
3. Si C es mayor a 20 (ejemplo) se dice que es una estrella (¿por qué?), en los casos opuestos se dice que no es una estrella

Clasificadores con base en el problema

Existen algunos clasificadores que existen para problemas específicos. Uno de ellos es el método de la firma que permite clasificar objetos con base en su forma



Retos

Los retos de esta página pueden ser realizados con las imágenes de la A0 a la A7 del enlace en las notas

1. Crear un programa que carga una imagen y extrae: área, perímetro y centroide. (5 puntos)
2. Crear un programa que carga una imagen y decide si es: una estrella, un círculo o un cuadro. Puede o no que se requiera información adicional a la extraída en el punto anterior (10 puntos)

Retos

Los retos de esta página deben ser probados con FOTOS reales como las de los ejemplos B0 y B1

1. Crear un programa que carga una imagen, realiza una segmentación y decide si es: una estrella, un círculo o un cuadro. Puede o no que se requiera información adicional a la extraída en el punto anterior (10 puntos)

Retos

Los retos de esta página deben ser probados con las imágenes de la C0 a la C4

1. Crear un programa que carga una imagen y cuenta la cantidad de huecos en esta (10 puntos)