Isabel Manzaneque Núñez – 53902577F Imanzaneq3@alumno.uned.es

Visión Artificial

PEC 2 curso 2023-2024

Índice de contenido

[Segmentación con conocimiento del dominio 2](#_Toc153344794)

[1. Apartado a. 2](#_Toc153344795)

[2. Apartado b. 2](#_Toc153344796)

[3. Apartado c. 2](#_Toc153344797)

[4. Apartado d. 2](#_Toc153344798)

[5. Apartado e. 2](#_Toc153344799)

[6. Apartado f. 2](#_Toc153344800)

[Descriptores de puntos característicos 2](#_Toc153344801)

[**1.** Apartado a. 2](#_Toc153344802)

[2. Apartado b. 2](#_Toc153344803)

[3. Apartado c. 2](#_Toc153344804)

[4. Apartado d. 2](#_Toc153344805)

[5. Apartado e. 2](#_Toc153344806)

[Reconocimiento de objetos 2](#_Toc153344807)

[**1.** Apartado a. 2](#_Toc153344808)

[2. Apartado b. 2](#_Toc153344809)

[3. Apartado c. 2](#_Toc153344810)

[4. Apartado d. 2](#_Toc153344811)

[Aplicación de reconocimiento de objetos 3](#_Toc153344812)

[1. Apartado 4.1 3](#_Toc153344813)

[2. Apartado 4.2 3](#_Toc153344814)

# Segmentación con conocimiento del dominio

## Apartado a.

**cv2.HoughCircles** se utiliza para detectar círculos en imágenes. En la documentación de OpenCV podemos encontrar información sobre los parámetros que se utilizan para configurar la función:

* **image**: Imagen de entrada en escala de grises.
* **method**: Método de detección. Los métodos disponibles son HOUGH\_GRADIENT y HOUGH\_GRADIENT\_ALT.
* **dp**: Relación inversa de la resolución del acumulador. Por ejemplo, un valor de 1 significa que el acumulador tiene la misma resolución que la imagen de entrada, mientras que un valor de 2 significa que el acumulador tiene la mitad de ancho y altura.
* **minDist**: Distancia mínima entre los centros de los círculos detectados.
* **param1**: Umbral superior para el detector de bordes Canny.
* **param2**: Umbral para la detección de centros del círculo.
* **minRadius**: Radio mínimo del círculo a detectar.
* **maxRadius**: Radio máximo del círculo a detectar.

**cv2.HoughLines** se utiliza para detectar líneas en una imagen. En la documentación de OpenCV podemos encontrar también toda la información relativa a sus parámetros:

* **image**: Imagen de entrada, que debe ser una imagen binaria.
* **rho**: Resolución de la distancia del acumulador en píxeles.
* **theta**: Resolución del ángulo del acumulador en radianes.
* **threshold**: Umbral de votos mínimo para decidir si se trata de una línea. Como el número de votos depende del número de puntos de la línea, representa el largo mínimo de la línea a detectar.

**cv2.HoughLinesP** es una variante de cv2.HoughLines que utiliza la transformada de Hough probabilística. Cuenta con los parámetros de cv2.HoughLines y añade dos parámetros extra:

* **image**: Imagen de entrada, que debe ser una imagen binaria.
* **rho**: Resolución de la distancia del acumulador en píxeles.
* **theta**: Resolución del ángulo del acumulador en radianes.
* **threshold**: Umbral de votos mínimo para decidir si se trata de una línea. Como el número de votos depende del número de puntos de la línea, representa el largo mínimo de la línea a detectar.
* **minLineLength**: Longitud mínima de la línea. Las líneas más cortas que esto son rechazadas.
* **maxLineGap**: Máximo espacio permitido entre segmentos de línea para tratarlos como una sola línea.

## Apartado b.

La transformada de Hough es una técnica que se puede utilizar para detectar cualquier tipo de forma como elipses, polígonos y curvas arbitrarias.

La condición es que la forma que se desea detectar debe ser definible matemáticamente mediante una ecuación paramétrica. Por tanto, si tenemos conocimiento de la ecuación paramétrica del contorno que se está buscando, la transformada de Hough podrá detectar la forma incluso si presentan algo de discontinuidad o distorsión).

## Apartado c.

Para resolver el apartado c se han realizado las siguientes funciones:

**drawHoughLines()**

Esta función aplica la transformada de Hough a la detección de la dirección del texto en la figura textoMolinos.png. Me he basado en el código del tutorial recomendado en el enunciado, el cual he adaptado a mis necesidades. La elección de los parámetros inicial ha sido la siguiente:

* **rho**: 1
* **theta**: pi/180
* **threshold**: 228

A screenshot of a computer screen

Description automatically generated

Con estos parámetros, se encuentra un total de 76 líneas. Como se había indicado antes, el threshold representa el largo mínimo de la línea a detectar. Parece que las líneas que no se han detectado tienen como característica el ser más cortas que las que sí se han detectado. Sin embargo, reducir el threshold a un valor inferior a 228 hace que se generen líneas ruidosas que no se corresponden con el texto.

Si cambiamos la resolución angular a pi/200, se detectarán un total de 92 líneas incluyendo la primera (que estaba siendo ignorada anteriormente). Esto puede deberse a que exista una ligera variación angular entre esta línea y el resto, por lo que una precisión angular más fina sería suficiente para detectarla. Hay que tener en cuenta que una resolución más fina puede también resultar en más falsos positivos, detectando líneas debido al ruido o variaciones menores en la imagen.

A screenshot of a computer screen

Description automatically generated

El valor de theta devuelto por cv2.HoughLines() es la inclinación de la línea respecto a la horizontal en radianes. Se puede utilizar para calcular la media de la inclinación en grados y así realizar una transformación afín para que las líneas del texto queden en horizontal. Esta inclinación media se le va a pasar a una función que rotará la imagen.

**rotate(meanAngle)**

Esta función recibe un ángulo “meanAngle” y rota la imagen esos grados en torno a su centro

A screenshot of a computer

Description automatically generated

**segment(rotatedImg)**

Esta función va a segmentar el texto de la imagen rotada. Comienza realizando una umbralización binaria para transformar la imagen rotada a una imagen en blanco y negro.

Se realizara una detección de bordes y contornos en la imagen, lo cual también detectará los bordes de la bounding box, lo cual queremos evitar. Tras realizar un filtrado de los contornos detectados, para eliminar aquellos mayores a un cierto umbral, el resultado de la segmentación es el siguiente:

A screenshot of a computer screen

Description automatically generated

## Apartado d.

## Apartado e.

## Apartado f.

# Descriptores de puntos característicos

## Apartado a.

## Apartado b.

## Apartado c.

## Apartado d.

## Apartado e.

# Reconocimiento de objetos

## Apartado a.

## Apartado b.

## Apartado c.

## Apartado d.

# Aplicación de reconocimiento de objetos

## Apartado 4.1

## Apartado 4.2