

Imagen y Vídeo por Computador



Práctica 2: Extracción de contornos y esquinas



ÍNDICE

1. Sesión 1: Contornos

- ⊙ Extracción de contornos mediante la función 'edge'
- ⊙ Extracción de contornos mediante convoluciones

2. Sesión 2: Esquinas

ÍNDICE



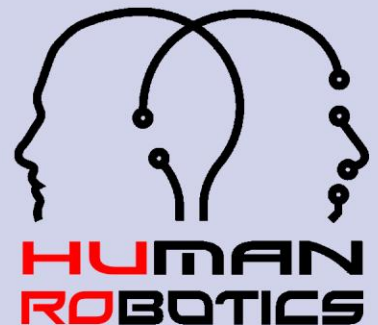
1. Extracción de contornos mediante la función 'edge'
2. Extracción de contornos mediante convoluciones
3. Esquinas
4. Extracción de esquinas con la función 'corner'



Vicente Morell
vicente.morell@ua.es



Universitat d'Alacant
Universidad de Alicante



Extracción de contornos mediante la función edge

- La función **edge** extrae los contornos de la imagen de entrada obteniendo una imagen binaria, donde los 1 representan los bordes encontrados.
- La **imagen de entrada** debe ser una imagen binaria o una imagen en escala de grises.

```
>> I = imread('mano_ua_gris.jpg');
>> BW = edge(I);
```

```
>> help edge;
```



Extracción de contornos mediante la función edge



● Métodos de detección de contornos

- **'sobel'**: Método de Sobel (opc. defecto)

Parámetros:

- Umbral.
- Dirección: vertical, horizontal o ambas.
- Opciones.

- **'prewitt'**: Método de Prewitt. Parámetros:

- Umbral.
- Dirección: vertical, horizontal o ambas.

- **'roberts'**: Método de Roberts. Parámetros:

- Umbral.
- Opciones.

- **'log'**: Laplaciana del filtro gaussiano.

Parámetros:

- Umbral.
- Desviación estándar.

- **'zerocross'**: Método de paso por cero.

Parámetros:

- Umbral.
- Filtro.

- **'canny'**: Método de Canny. Parámetros:

- Umbral.
- Desviación estándar.



Detección de contornos por convolución

- ⊙ **fspecial** para definir el tipo de filtro que se va a realizar, así como el tamaño de la máscara de convolución (por defecto de 3x3):
 - ⊙ Filtro de media: **'average'**.
 - ⊙ Filtro Gaussiano: **'gaussian'**.
 - ⊙ **'laplacian'**: Filtro de 3x3 que aproxima la forma de un operador bidimensional Laplaciano.
 - ⊙ **'log'**: Filtro de Laplaciana de Gaussiana.
 - ⊙ **'prewitt'**: Filtro de Prewitt, un filtro de 3x3 que enfatiza los contornos horizontales aproximando un gradiente vertical.
 - ⊙ **'sobel'**: Filtro de Sobel, un filtro de 3x3 que enfatiza los contornos horizontales aproximando un gradiente vertical.
- ⊙ **imfilter**: realiza la convolución con los filtros definidos mediante fspecial.

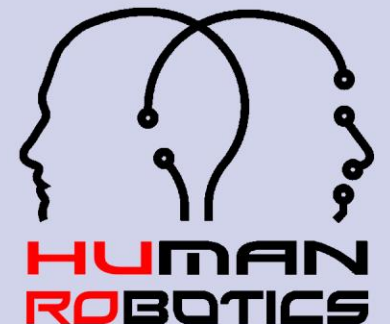
ÍNDICE

1. Sesión 1: Contornos
2. **Sesión 2: Esquinas**
 - ⊙ Esquinas
 - ⊙ Extracción de esquinas con la función 'corner'

Vicente Morell
vicente.morell@ua.es

Ingeniería
Multimedia

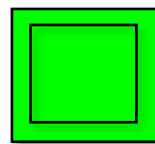
Universitat d'Alacant
Universidad de Alicante



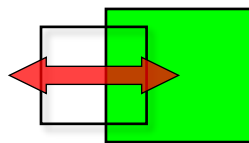
Esquinas

- La detección de esquinas funciona bajo el principio de que si se coloca una pequeña ventana sobre una esquina de una imagen y se mueve en cualquier dirección, habrá un gran **cambio en la intensidad**. Esto se ilustra a continuación con algunos diagramas.

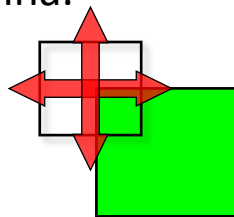
- Si la ventana se sitúa sobre un **área plana** de la imagen entonces no se verá ningún cambio de intensidad cuando se mueve la ventana



- Si la ventana se encuentra sobre un **borde** sólo habrá un cambio de intensidad si la ventana se mueve en una dirección.



- Si la ventana se sitúa sobre una **esquina**, entonces habrá un cambio en todas las direcciones, y por lo tanto, sabemos que debe ser una esquina.



Función corner



- ⦿ La función **corner** extrae las esquinas de la imagen de entrada obteniendo una matriz de $M \times 2$, donde M indica el número de esquinas encontradas.
- ⦿ La **imagen de entrada** debe ser una imagen binaria o una imagen en escala de grises.
- ⦿ Métodos:
 - ⦿ '**Harris**': Detector de esquinas de Harris (por defecto).
 - ⦿ '**MinimumEigenvalue**': Método de Shi & Tomasi.
- ⦿ Otros parámetros:
 - ⦿ **N**: el número de esquinas máximo que busca la función (por defecto 200).
 - ⦿ '**FilterCoefficients**'.
 - ⦿ '**QualityLevel**'.
 - ⦿ '**SensitivityFactor**'.

```
>> I = imread('mano_ua_gris.jpg');
>> C = corner(I);
```

```
>> help corner;
```

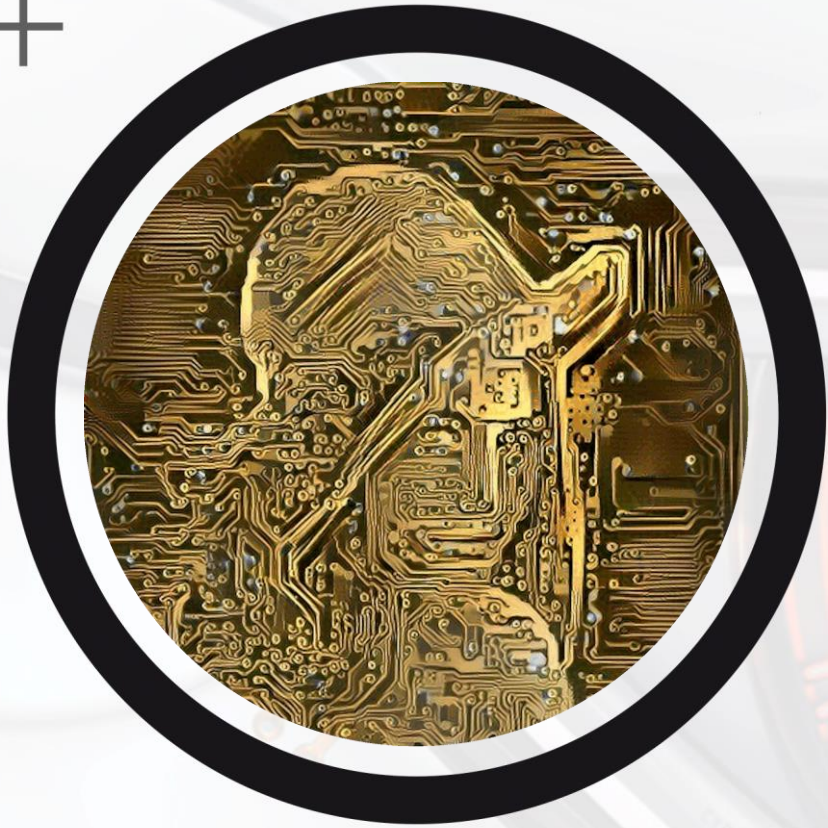


Imagen y Vídeo por Computador



Práctica 2: Extracción de contornos y esquinas

