

Aula de Inteligencia Artificial

Contornos y Segmentación



Contornos

- Contornos se corresponden con cambios de nivel de gris bruscos en la imagen



Máscaras contornos

X – Direction Kernel

-1	0	1
-2	0	2
-1	0	1

Contornos verticales

Y – Direction Kernel

-1	-2	-1
0	0	0
1	2	1

Contornos horizontales

100	100	200	200
100	100	200	200
100	100	200	200
100	100	200	200

-1	0	1
-2	0	2
-1	0	1

-100
-200
-100
200
400
<u>+200</u>
=400

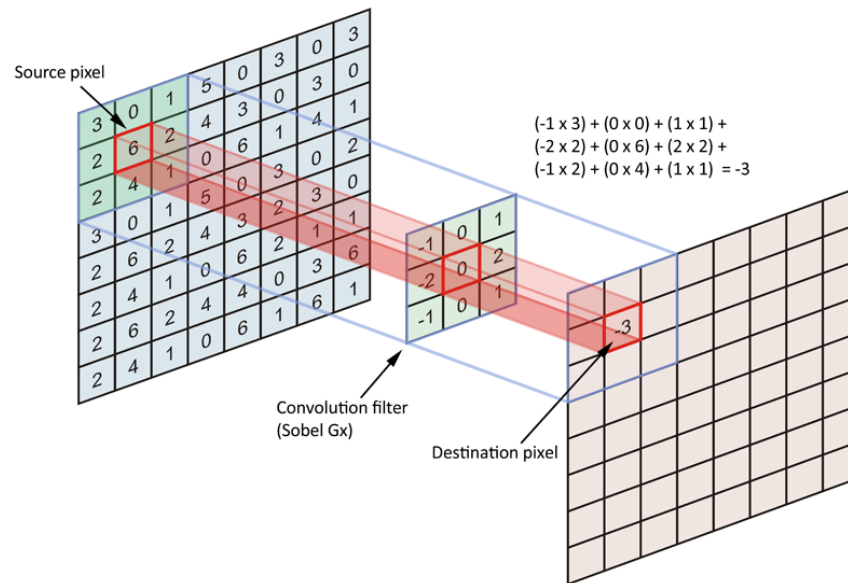
Kernel Convolution: The bigger the value at the end, the more noticeable the edge will be.

Fuente: An Implementation of Sobel Edge Detection.

https://www.projectrhea.org/rhea/index.php/An_Implementation_of_Sobel_Edge_Detection

Convolución

- Aplicar la máscara a toda la imagen desplazando la máscara pixel a pixel



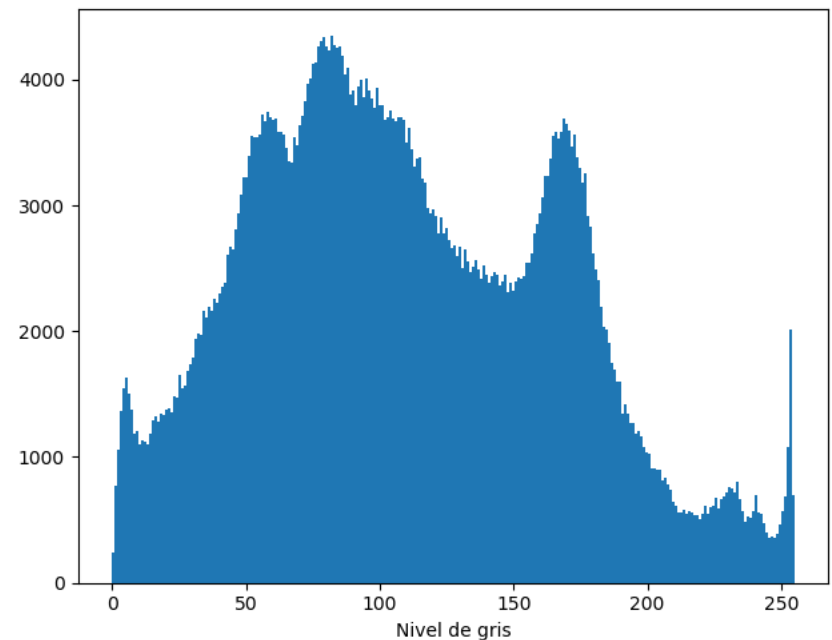
contornos image = contornos verticales + contornos horizontales

Tarea operador sobel

- Analizar y probar el ejemplo “test-filtrado-sobel.py” y ejecutarlo.
- Modificar el ejemplo:
 - Calcular los contornos horizontales.
 - Combinar los contornos horizontales y verticales en una sola imagen.
 - Modificar mostrar los contornos de las imágenes de la cámara utilizando la función `cv2.Canny(imagenGrises, 100, 300, 3)`

Histogramas

- Representación del número de píxeles de una imagen que tiene un determinado valor (nivel de gris o componente de color)



Separación fondo-objeto basada en histograma - Umbralizado

- Marcar los píxeles como pertenecientes al fondo frente a los objetos usando el nivel de gris
 - Fondo con un color uniforme y claro (oscuro)
 - Objetos con un color oscuro (claro)
- Píxeles nivel gris $>$ umbral \rightarrow Cambiar a blanco
- Píxeles nivel gris \leq umbral \rightarrow Cambiar a negro



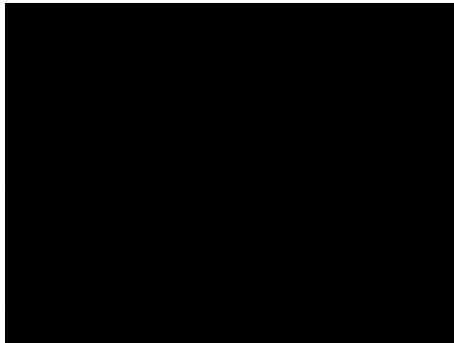
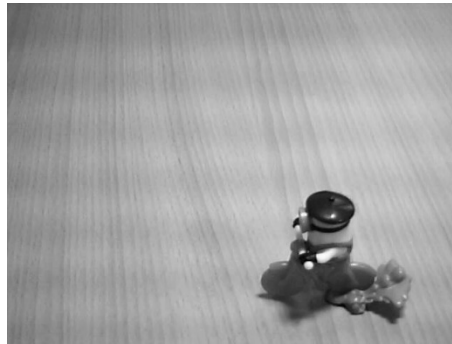
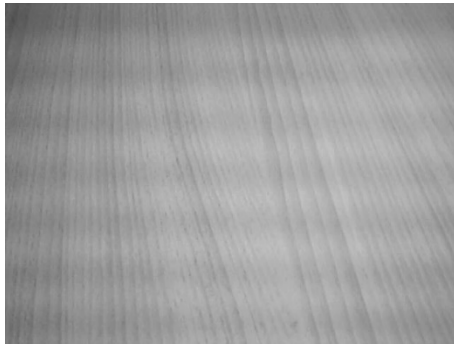
Tarea umbralizado

- Analizar y ejecutar el script “test-segmentacion-histograma.py”.
- Cambiar el valor del umbral y comprobar los resultados.
- Modificar el script para abrir la imagen. BioID_0086.jpg y encontrar un umbral que permita separar la cara de la persona del fondo.

Detección objeto en movimiento mediante en diferencia de imágenes

- Detecta los objetos que se mueven en un vídeo.
- Diferencia entre fotogramas consecutivos.
 - Si el objeto se mueve la diferencia entre fotogramas es mayor que cero
 - Si el objeto no se mueve la diferencia entre fotogramas es cero (muy baja debido al ruido)

Detección objeto en movimiento mediante en diferencia de imágenes



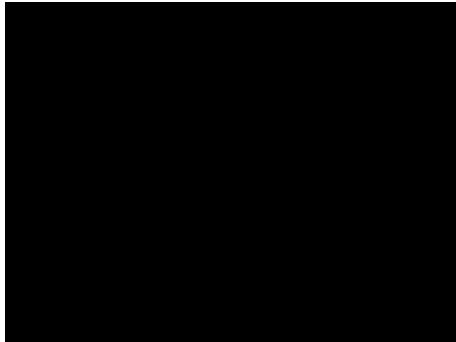
Tarea diferencia imágenes

- Analizar y ejecutar el script “test-diferencia-`imagenes.py`”.
- Modificar cambiando el valor mínimo para considerar movimiento (valor 30 en la función `cv2.threshold`) y ver resultado.

Separación fondo-objeto basada sustracción del fondo

- Utilizado cuando hay objetos en movimiento en un video.
- Marcar como fondo los píxeles que no cambian su nivel de gris con respecto al fondo estático.
- Diferencia con fondo estático $>$ umbral \rightarrow objeto
- Diferencia con fondo estático \leq umbral \rightarrow fondo

Separación fondo-objeto basada sustracción del fondo



Tarea sustracción del fondo

- Analizar y ejecutar el script “test-sustraccion-fondo.py”.
- Modificar el ejemplo para dibujar una línea vertical en que siga al objeto por la imagen.



[My little piece of privacy](#), Niklas Roy, 2010