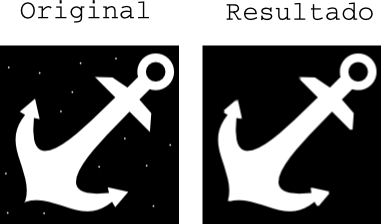
**3- Morphological Gradient**

[](http://robologs.net/wp-content/uploads/2015/07/resultado_gradient.png)No es más que la diferencia entre el ‘Dilation’ y el ‘Erode’. ¡Es muy útil para encontrar la silueta de los objetos!

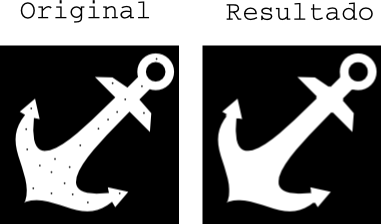
|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16 | import numpy as np  import cv2    #Cargar la mascara  imagen = cv2.imread('mask.png',0)    #Crear un kernel de '1' de 3x3  kernel = np.ones((3,3),np.uint8)    #Se aplica la transformacion: Morphological Gradient  transformacion = cv2.dilate(imagen,kernel,iterations = 1) - cv2.erode(imagen,kernel,iterations = 1)    #Mostrar el resultado y salir  cv2.imshow('resultado',transformacion)  cv2.waitKey(0)  cv2.destroyAllWindows() |

**4- Opening**

[](http://robologs.net/wp-content/uploads/2015/07/resultado_opening.png)Es un ‘Erotion’ seguido de un ‘Dilation’. Nos sirve para eliminar el ruido blanco sobre las zonas negras.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16 | import numpy as np  import cv2    #Cargar la mascara  imagen = cv2.imread('mask.png',0)    #Crear un kernel de '1' de 3x3  kernel = np.ones((3,3),np.uint8)    #Se aplica la transformacion: Opening  transformacion = cv2.morphologyEx(imagen,cv2.MORPH\_OPEN,kernel)    #Mostrar el resultado y salir  cv2.imshow('resultado',transformacion)  cv2.waitKey(0)  cv2.destroyAllWindows() |

**5- Closing**

[](http://robologs.net/wp-content/uploads/2015/07/resultado_closing.png)Justo lo contrario que el Opening: el Closing es un ‘Dilation’ seguido de un ‘Erosion’. Si hay ruido negro en las áreas blancas, el Closing lo limpiará.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16 | import numpy as np  import cv2    #Cargar la mascara  imagen = cv2.imread('mask.png',0)    #Crear un kernel de '1' de 3x3  kernel = np.ones((3,3),np.uint8)    #Se aplica la transformacion: Closing  transformacion = cv2.morphologyEx(imagen,cv2.MORPH\_CLOSE,kernel)    #Mostrar el resultado y salir  cv2.imshow('resultado',transformacion)  cv2.waitKey(0)  cv2.destroyAllWindows() |