

Procesamiento de Imágenes

Departamento de Ciencias e Ingeniería de la Computación

Universidad Nacional del Sur

2020

Actividad 6:

Detección de bordes

Imagen original

La imagen sobre la cual se aplican todas las técnicas de detección de bordes para que contribuyan al efecto *cartoon* es la siguiente:



En todos los casos, se muestra el resultado final, no la detección de bordes *per se*, sino cómo estos contribuyen al efecto *cartoon* en general, ya que el resto del procesamiento de la imagen es el mismo para todas.

Laplaciano (actividad 3)

Aplicando el Laplaciano con un kernel de 1×1 sobre la imagen en escala de grises, se obtiene el siguiente resultado:



Si bien el contorno del delfín y el horizonte del agua quedan bien relativamente definidos, aunque partes del primero quedan sin remarcan en la parte baja de la imagen, las olas, las gotas de agua y la espuma que provoca la cola del animal contienen ruido y muchos píxeles negros que empeoran el resultado del efecto.

Laplaciano sobre Gaussiano (LoG)

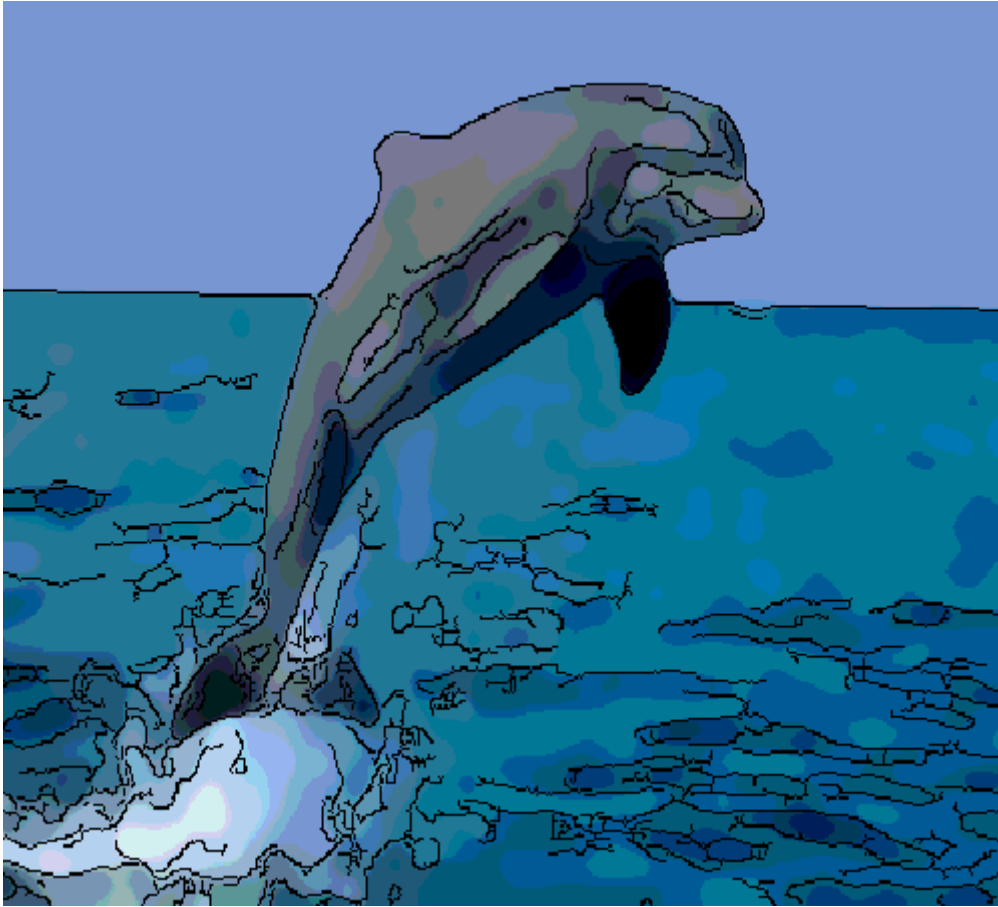
Aplicando un filtro Gaussiano con un kernel de 3x3 y luego el Laplaciano con un kernel 3x3, se obtiene el siguiente resultado:



Igual que el anterior, el contorno del delfín y el horizonte del agua quedan relativamente bien definidos, pero ahora la cantidad de ruido se reduce y la espuma en la cola del animal ahora parece una única región, como debería ser según el efecto que se busca.

Canny

Aplicando el algoritmo de Canny con un primer umbral de 10 y un segundo umbral de 100, se obtiene el siguiente resultado:



En este caso, el contorno del delfín y el horizonte del agua quedan perfectamente definidos. Sin embargo, algunas características del primero quedan sobre enfatizadas.

Sobel

En este caso, se aplica dos veces el algoritmo de Sobel a la imagen original: una con un orden 1 de derivada x y otra con un orden 1 de derivada y . Luego se procede combinar el resultado de ambas en una única foto, para luego pasarla a la siguiente etapa de procesamiento. El resultado es el siguiente:



Este es probablemente el peor resultado: no todo el contorno del delfín está re-marcado, el horizonte no tiene borde y, los bordes que sí se detectan, quedan sobre saturados y con un grosor muy grande. Sin embargo, es posible que este sea un caso particular, ya que con otras imágenes, este algoritmo obtiene muy buenos resultados, como se pueden observar con el siguiente ejemplo:



Conclusión

Todos los algoritmos utilizados en esta actividad obtienen mejores resultados que el empleado en la actividad 3, a muy bajo costo de implementación, ya que en todos los casos lo único que hay que hacer es agregar unas pocas líneas más de código para utilizar los métodos requeridos. Aún así, cabe destacar que esto puede ser una ventaja o consecuencia de utilizar OpenCV, ya que esta librería implementa todas las operaciones que se mostraron. Esto puede no ser así utilizando alguna otra librería o lenguaje de programación.

Sin embargo, el costo de ejecución es indudablemente mayor, agregando más pasadas y procesamiento a cada uno de los píxeles de la imagen, lo cual se puede notar cuando se los ejecuta con alguna imagen relativamente grande (1000x1000 píxeles), ya que ahora demoran uno o dos segundos más, en promedio.