

## Informe Técnico Tarea 5

Carla Mardones y Paula Mardones  
Universidad Adolfo Ibáñez  
camardones@alumnos.uai.cl  
paumardones@alumnos.uai.cl

### Abstract

Se utilizó la imagen “rombo.png” para tratar de aislar el objeto. En un primer instante se le agrega ruido a esta imagen y se trata de disminuirlo a través del filtro gaussiano. Luego, se busca la manera más adecuada que encontrar los bordes del objeto incrustado en la imagen. Para esto, se probaron los filtros sato, roberts, prewitt y sobel y canny. A través de este análisis se eligió el filtro canny que era el más adecuado para el objetivo. Con esto determinado, se continuó a buscar el procesamiento morfológico que más iba a ayudar a obtener un borde del adecuado. Para esto se ocupó la clausura, lo cual involucra una dilatación y luego una erosión. Por último, se ocupó la función floodFill para rellenar el objetivo incrustado.

### Key Words

Filtro gaussiano, filtro Sato, filtro Roberts, filtro Prewitt, filtro Sobel, clausura, dilatación, erosión, kernel, borde, floodFill, relleno.

## 1. Introducción

El siguiente trabajo tiene como objetivo analizar diferentes tipos filtros, procesamientos morfológicos y algoritmos aprendidos en clases, para poder identificar y aplicar sobre una imagen con ruido la cual tiene un objeto que queremos aislar. El trabajo se dividió en cuatro partes, la primera para reducir el ruido, la segunda para buscar bordes dentro de la imagen, la tercera para unir estos bordes, y la última para rellenar el espacio contenido por los bordes. Cada parte involucró un análisis de diferentes filtros o procesamientos para poder elegir el más adecuado dado las circunstancias de la imagen con ruido entregada.

## 2. Antecedentes

### 2.1. Material Complementario

#### Tipos de Filtros

Existen dos tipos de filtros: los del dominio espacial y los del dominio frecuencial.

#### Dominio Espacial

Las operaciones de filtrado en los filtros espaciales se llevan a cabo directamente sobre los píxeles de la imagen. De aquí se pueden separar en 3 categorías: de orden estadístico, adaptativos o lineales.

- Los filtros de **orden estadísticos** operan sobre la base de un orden o ranking de los datos contenidos en una región o ventana. El resultado obtenido como respuesta del filtro opera únicamente a base del orden.
  - Mediana
  - Moda
  - Max-Min
  - Punto Medio
  - Promedio Alfa-Acotado
- Los filtros **adaptivos** cambian según las propiedades de la región o ventana analizada. Esto permite obtener un mayor rendimiento, pero al mismo tiempo implica un mayor costo computacional.
  - Ruido-Local
  - Mediana Adaptativa

- Los filtros **lineales** realizan una combinación lineal sobre los píxeles de la ventana, determinando un único valor en cada máscara. La operación matemática sobre la cual se aplica a la imagen es conocida como convolución.
  - Media
  - Media geométrica
  - Media armónica
  - Media contra-armónica
  - Gaussiana

### Dominio Frecuencial

Los filtros de frecuencia procesan una imagen trabajando sobre el dominio de la frecuencia en la *Transformada de Fourier* de la imagen. Para ello, esta se modifica siguiendo el Teorema de la Convolución correspondiente:

1. se aplica la Transformada de Fourier,
2. se multiplica posteriormente por la función del filtro que ha sido escogido,
3. para concluir re-transformándola al dominio espacial empleando la Transformada Inversa de Fourier.

Estos pueden ser de tipo:

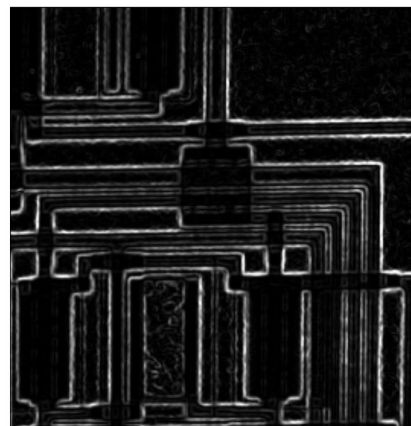
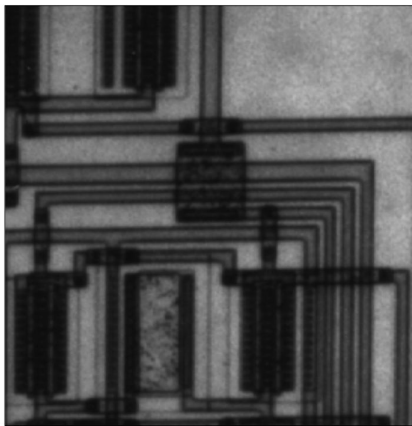
- Ideal
- Gaussiano
- Butterworth
- Homomórfico
- Puntuales
- Puntuales óptimos
- Filtro de Wiener
- Filtro Paramétrico

Usando:

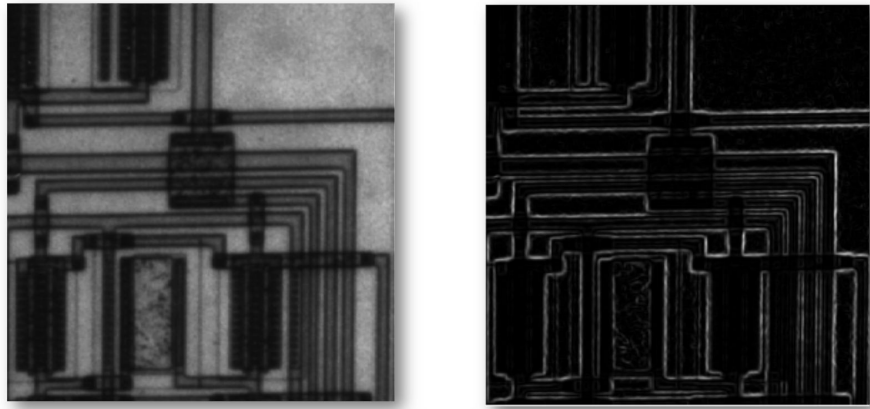
- Pasa alto
- Pasa bajo
- Pasa banda
- Para band

Filtros de Gradiente:

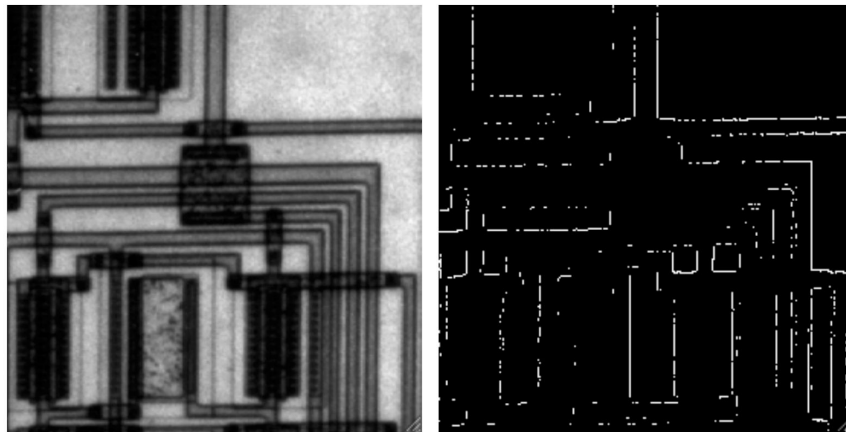
- Sobel: El operador de sobel utiliza dos máscaras direccionales de 3x3 que recorre la imagen a través de una operación en bloques.



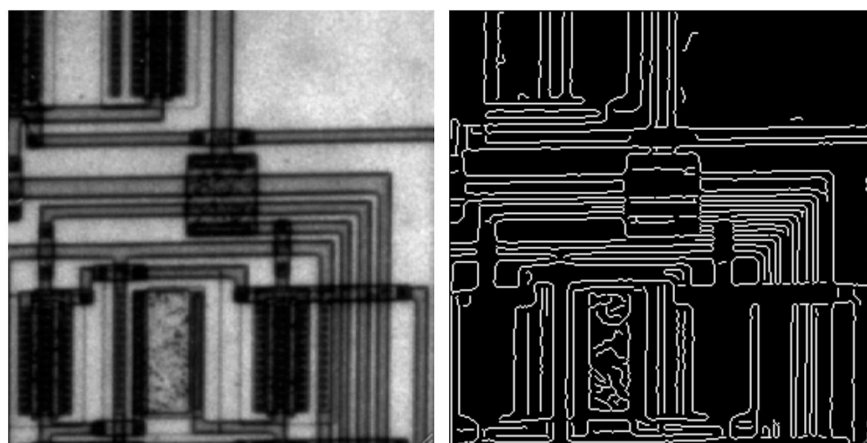
- Prewitt: El operador prewitt utiliza dos máscaras direccionales de 3x3 que recorre la imagen a través de una operación en bloques.



- Roberts: Roberts. El operador de sobel utiliza dos máscaras direccionales de 2x2 que recorre la imagen a través de una operación en bloques.

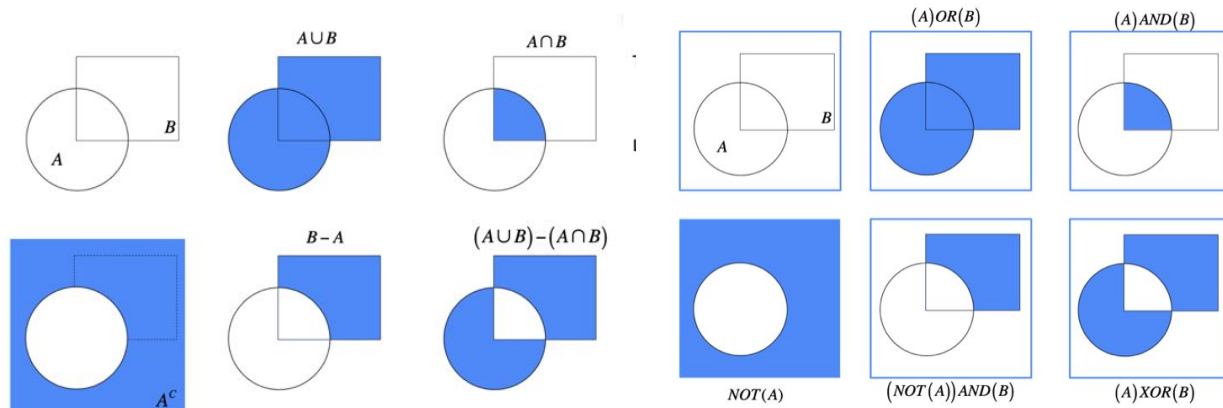


- Canny: El operador de Canny es el más complejo ya que encuentra los bordas buscando el máximo local del gradiente en un proceso denominado Histérisis.



## Procesamientos Morfológicos

La idea de los procesamientos morfológicos es que se aplican los procesos de las teorías de conjuntos y operadores lógicos sobre la imagen, usando un kernel (un objeto que representa una matriz).

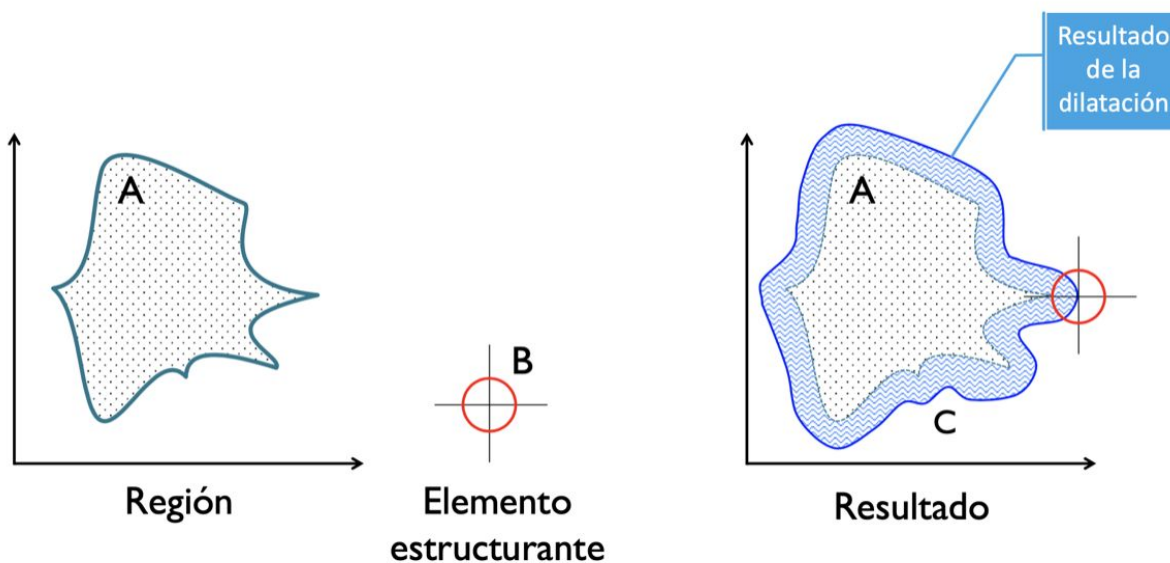


### Dilatación y Erosión

Comúnmente son empleados para la reducción del ruido, realce de la imagen, detección de bordes, segmentación, análisis de textura, etc.

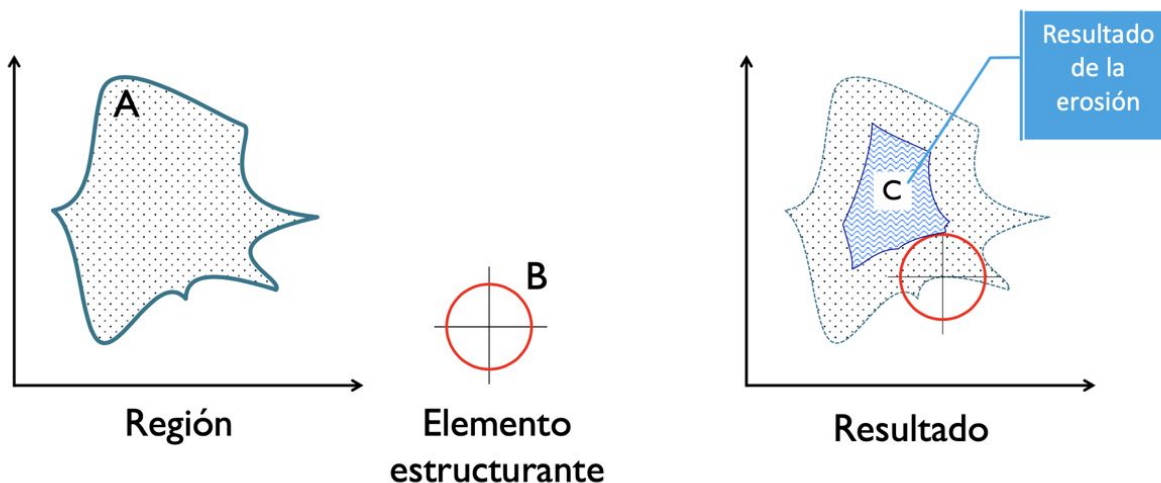
#### Dilatación

Uno de los operadores más importantes en la matemática morfológica es la dilatación. Desde un punto de vista geométrico, la dilatación consiste en el crecimiento de la región (o conjunto) por un elemento estructurante. El elemento puede ser cualquier objeto o forma, no necesariamente círculo, pero el círculo dilata uniformemente por todos los lados.



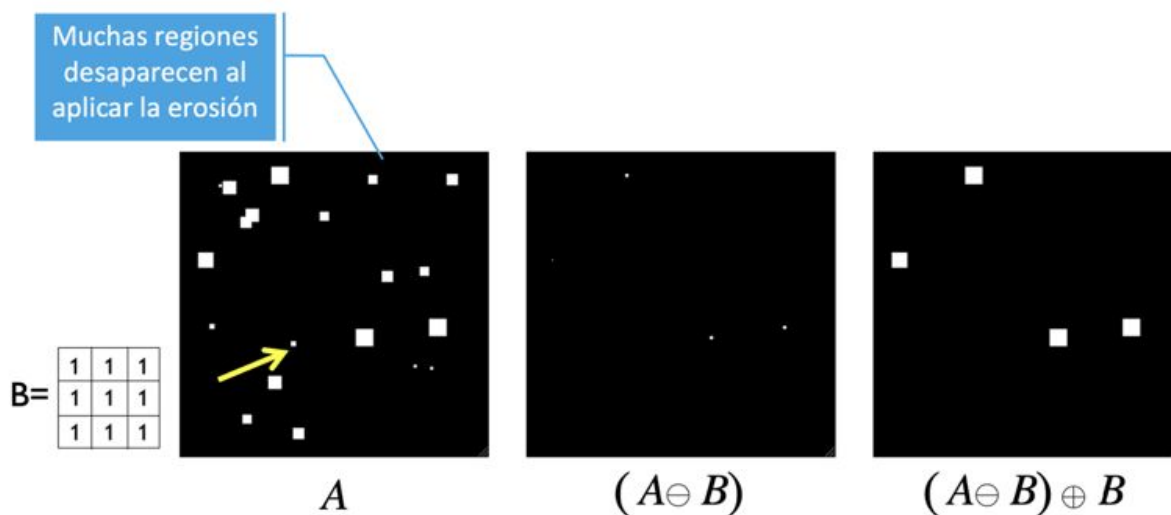
## Erosión

El segundo operador más importante en la matemática morfológica es la erosión. Desde un punto de vista geométrico, la erosión consiste en la reducción de parte de la región (o conjunto) por un elemento estructurante.



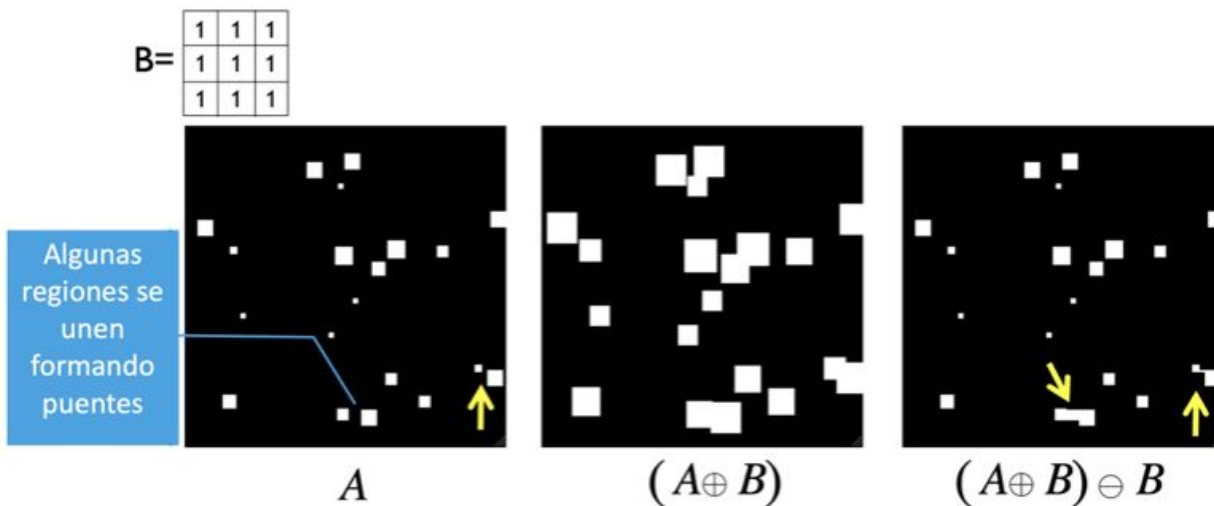
## Apertura

La apertura es una función que tiene dos pasos. Primero aplicamos **erosión** y luego a dicho resultado una **dilatación**. El efecto más relevante es la reducción del ruido. Matemáticamente se representa con la siguiente expresión



## Cierre/Clausura

La clausura es una función que tiene dos pasos. Primero aplicamos una **dilatación** y luego a dicho resultado una **erosión**. Esta técnica permite unir regiones que están muy cercanas. Matemáticamente se representa con la siguiente expresión:



## 2.2. Trabajos Relacionados

En la parte 2 de la tarea 3 se tuvo que probar con muchos filtro hasta al final utilizar el filtro Gaussiano para mejorar el ruido de una imagen. Este borra los detalles desenfocando la imagen, pero de lo que queda se puede ver a grandes rasgos lo principal de la imagen. Por esto, es un buen filtro para borrar la matriz de puntos. Sin embargo, también se puede ver que se perdieron detalles en toda la imagen, no solo la matriz.



Además en la tarea 4, en la primera parte, se tuvo que hacer algo parecido, donde tuvimos que probar muchos tipos de filtros espaciales con diferentes parámetros para ver cuales eran los más adecuados según los respectivos ruidos. Luego en la parte b, también se tuvo que hacer pruebas de diferentes parámetros de los filtros frecuenciales para elegir los adecuados según los respectivos ruidos.



**Ruido Gaussiano**



**Filtro Ruido Local**



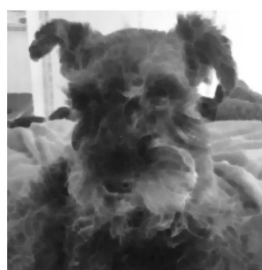
**Ruido Uniforme**



**Filtro Media**



**Ruido Impulsional-Sal**



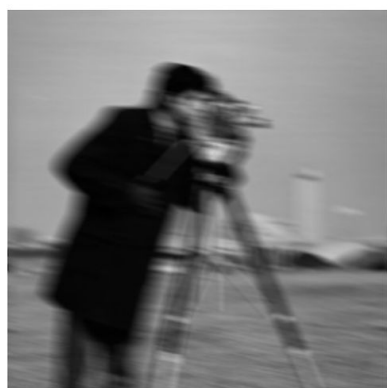
**Filtro Min**



**Ruido Impulsional-Pimienta**



**Filtro Media Contra-armónica**



**Imagen con Ruido**



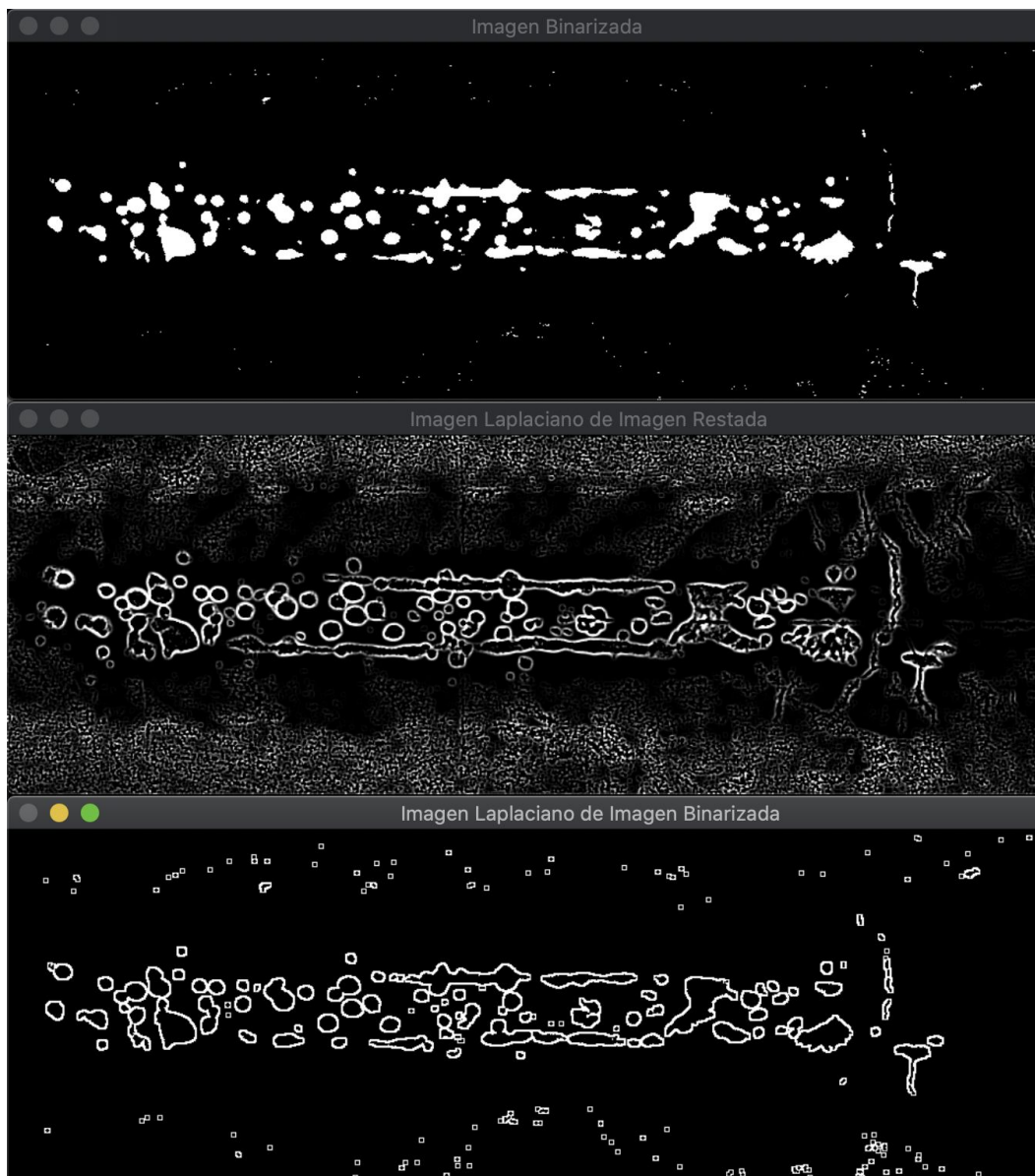
**Ruido + Filtro de Wiener**



**Ruido + Filtro Paramétrico**

Con esta práctica de poder idear formas para disminuir los ruidos probando con diferentes filtros, avanzamos a la tarea actual donde tuvimos que repetir este proceso de prueba y error.

Por último, la tarea que más nos aportó ayuda fue la tarea 2. Esta nos portó en entender el proceso para aislar ciertos elementos de una imagen.





### 3. Método Propuesto

La imagen original que utilizamos es la entregada por el profesor: “rombo.png”



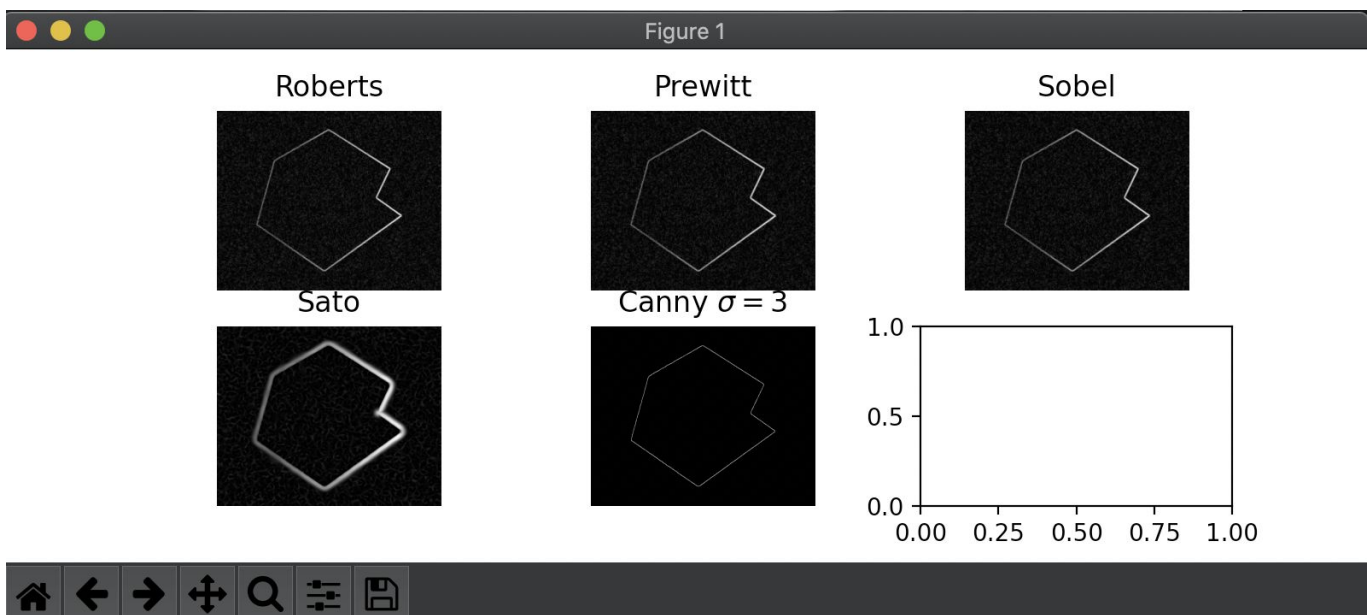
Imagen Original: “rombo.png”

**1. Utilice un filtro gaussiano para reducir el ruido generado en la variable output. Para ello utilice el filtro gaussiano y los comandos apropiados para definir una máscara gaussiana.**

Se pidió que se utilizara un filtro gaussiano para disminuir el ruido generado por el código entregado por el profesor. Aquí se experimentó con los parámetros de la máscara, de  $t$  y  $\sigma$ .

**2. Determine el borde a la imagen del resultado del paso 2. Utilice alguna de las técnicas vistas en clases para generar bordes. Puede experimentar modificar los parámetros del paso 2.**

Utilizamos matplotlib para ver cómo eran los bordes del rombo dependiendo del modelo. Decidimos utilizar **canny** porque es el borde es más homogéneo, es decir, es más parejo que el borde de sato, roberts, prewitt y sobel. Estos últimos tienen un borde más blanco en el lado derecho.



**3. Utilice los algoritmos morfológicos vistos en clases para unir los bordes del paso 3. Se recomienda que realice una clausura con estructuras de distinto tamaño, es decir, que la estructura para erosionar sea menor a la dilatación con un kernel definido por usted.**

Se realizó una clausura para obtener el borde del rombo. Para esto, se hizo una dilatación y luego una erosión sobre esa dilatación. En un principio solo utilizamos un kernel = 15 para la dilatación y luego erosión, pero se veía un poco extraño el borde:

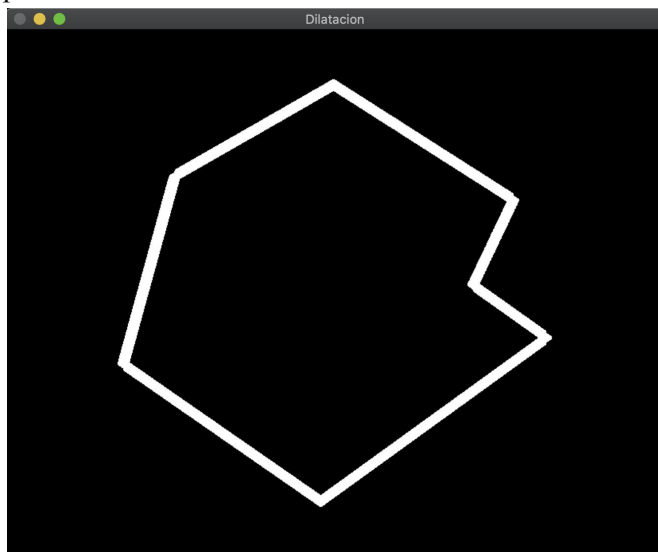


Imagen Dilatación Kernel=15

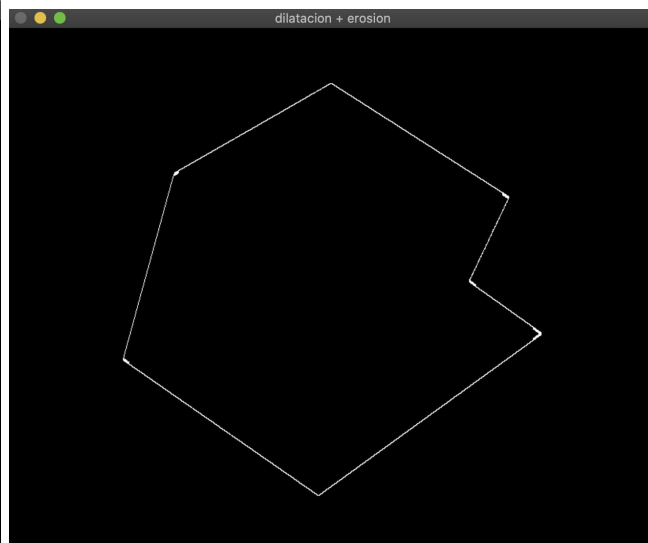


Imagen Erosión Kernel=15

Por esto decidimos tener 2 kernel distintos, uno para la dilatación  $k = 15$ , y otro para la erosión  $k = 5$

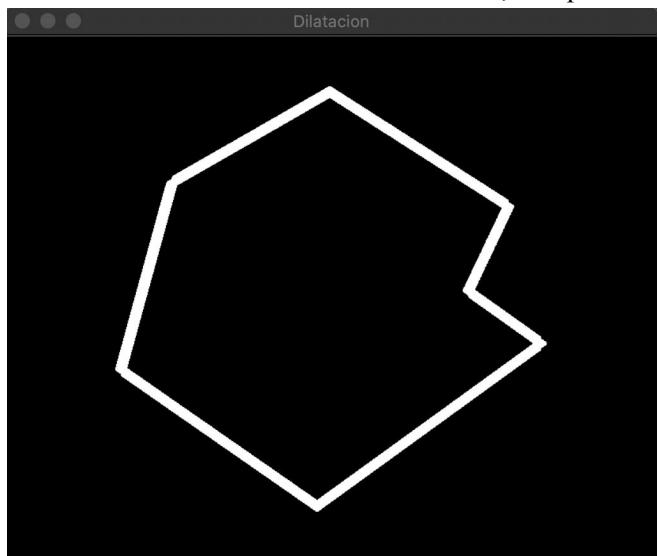


Imagen Dilatación Kernel=15

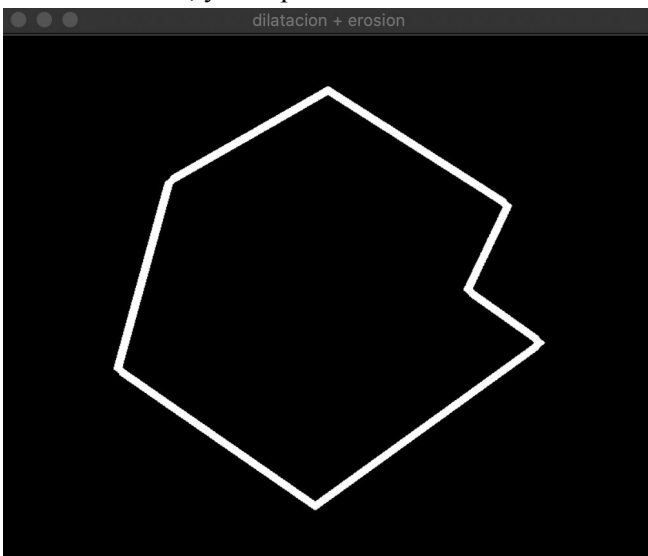
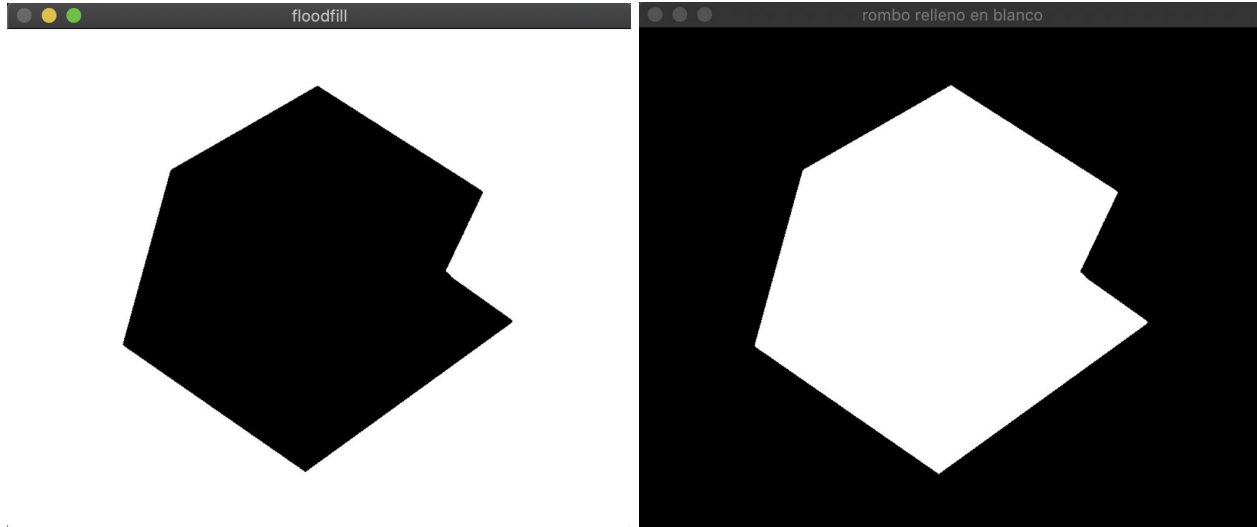


Imagen Erosión Kernel=5

**4. Utilice la función `cv2.floodFill` para cerrar la región del paso 4. Recuerde que dicha región debe estar cerrada para que el algoritmo rellene la región.**

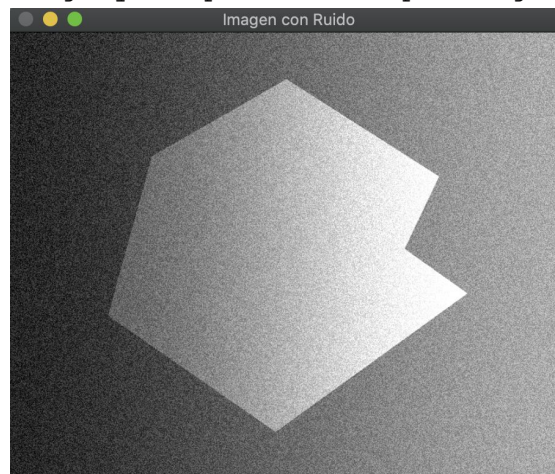
Utilizamos `floodfill` para que llene la región afuera de la línea en blanco. De esta manera, se obtuvo un rombo negro con fondo blanco. Sin embargo, el objetivo es rellenar el rombo por dentro, por lo que sacamos el negativo de la imagen y obtuvimos un rombo blanco con fondo negro.



#### 4. Resultados

1. Utilice el siguiente código para abrir su imagen. La imagen `rombo.png` se encuentra en la intranet junto a este enunciado.

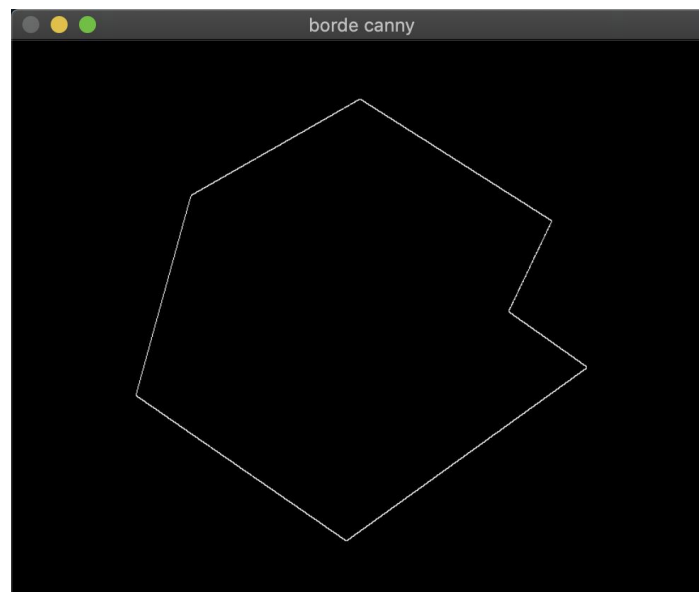
```
img = cv2.imread(rombo.png')
gray = cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)
gray_norm = cv2.normalize(gray.astype('float'), None, 0.0, 1.0,
cv2.NORM_MINMAX)
noise = np.random.random(gray.shape)*0.3 output = gray_norm + noise
```



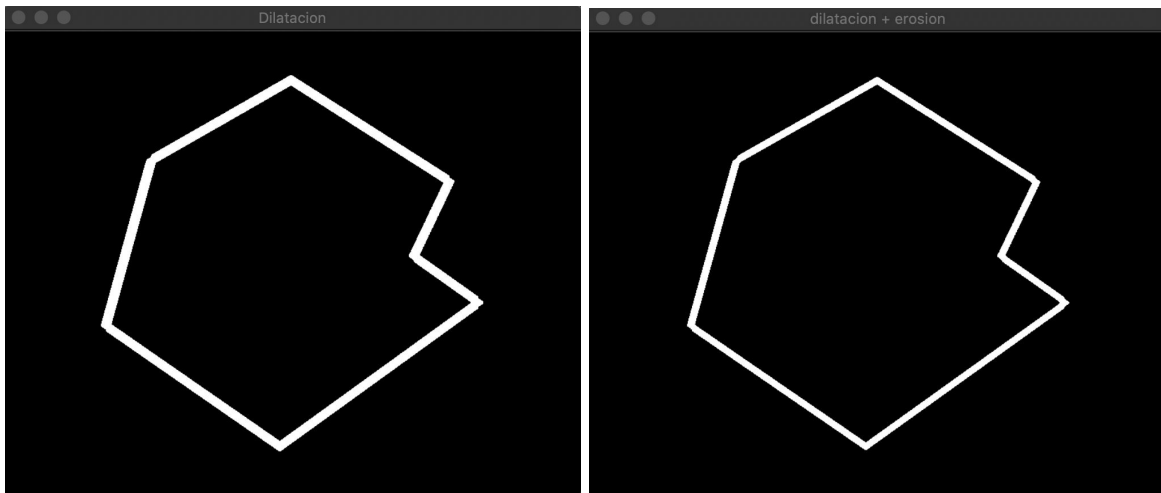
2. Utilice un filtro gaussiano para reducir el ruido generado en la variable output. Para ello utilice el filtro `ndi.generic_filter` los comandos apropiados para definir una máscara gaussiana.



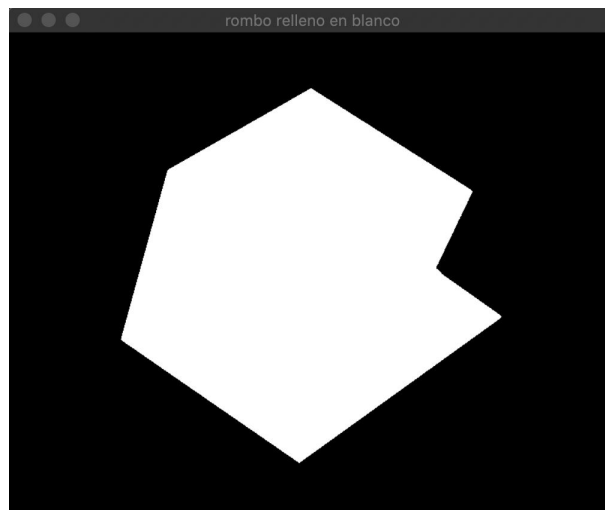
3. Determine el borde a la imagen del resultado del paso 2. Utilice alguna de las técnicas vistas en clases para generar bordes. Puede experimentar modificar los parámetros del paso 2.



4. Utilice los algoritmos morfológicos vistos en clases para unir los bordes del paso 3. Se recomienda que realice una clausura con estructuras de distinto tamaño, es decir, que la estructura para erosionar sea menor a la dilatación con un kernel definido por usted.



5. Utilice la función `cv2.floodFill` para cerrar la región del paso 4. Recuerde que dicha región debe estar cerrada para que el algoritmo rellene la región.



## 5. Conclusiones

En conclusión, es importante saber el objetivo del desafío para ver qué se puede hacer para transformar la imagen en la deseada. Para esta tarea, se pudo identificar el borde del rombo utilizando el filtro Canny y luego a partir de algoritmos morfológicos se mejoró este borde con clausura (una dilatación y luego erosión). De esta manera, se aseguraba que el borde estuviese cerrado completamente. Por último, con el borde cerrado, se utilizó floodfill para rellenar la figura con un color. Es parecido a utilizar el color bucket en paint, el cual cambia de color todo un sector que tenga el mismo color. En este caso, flood fill cambio el fondo de negro a blanco, es decir se obtuvo un rombo negro y fondo blanco. Sin embargo, para lograr el objetivo: rellenar el rombo en blanco, simplemente sacamos el negativo de la imagen.



## **6. Referencias**

- [1] Procesamiento digital de imágenes “07\_ruido\_y\_filtrado.pdf”
- [2] Procesamiento digital de imágenes “08\_filtrado.pdf”
- [3] Procesamiento digital de imágenes “09\_filtrado.pdf”
- [4] Procesamiento digital de imágenes “10\_procesamiento\_morfologico.pdf”
- [5] Procesamiento digital de imágenes “11\_segmentacion.pdf”