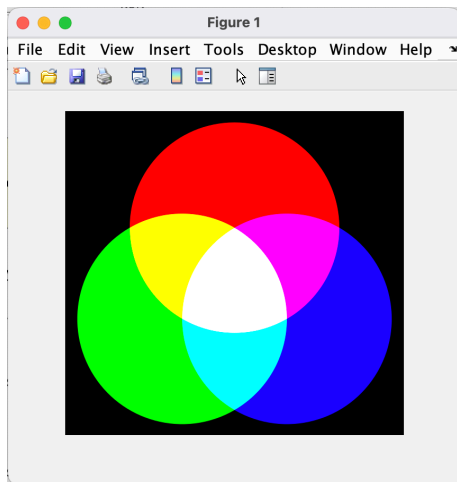


# Reporte de práctica

## Ruido y filtros

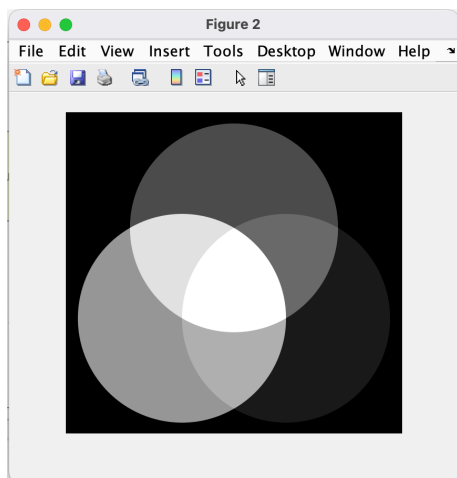
## 1) Cargar una imagen RGB

Con el comando `imread('Archivo de imagen.jpg')` se carga la imagen, y con el comando `imshow(I)` esta se muestra en una figura.



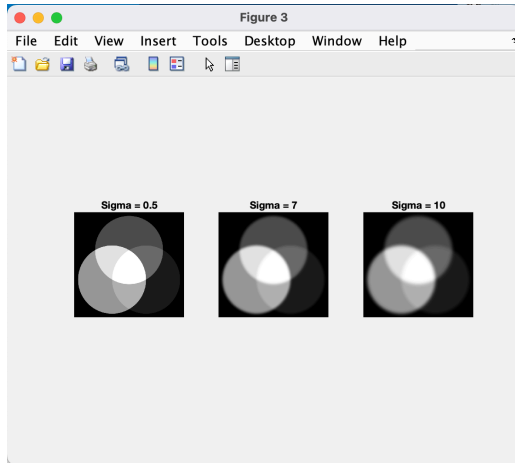
## 2) Convertir a escala de grises

Con el comando `rgb2gray(I)` se logra convertir la imagen original a una en escala de grises, para después mostrarla en una figura.



## 3) Aplicar un filtro Gaussiano

Partiendo de la imagen que se obtuvo anteriormente, se aplica el comando `imgaussfilt(I)`, este filtra la imagen  $I$  con un núcleo de suavizado gaussiano 2D con desviación estándar de 0,5 [1]. Si a este comando se le agrega el valor de sigma (`imgaussfilt(I, SIGMA)`), la desviación estándar cambia al valor que se ha escrito. De esta manera, se cambió el valor de sigma en tres ocasiones, resultando en lo siguiente.

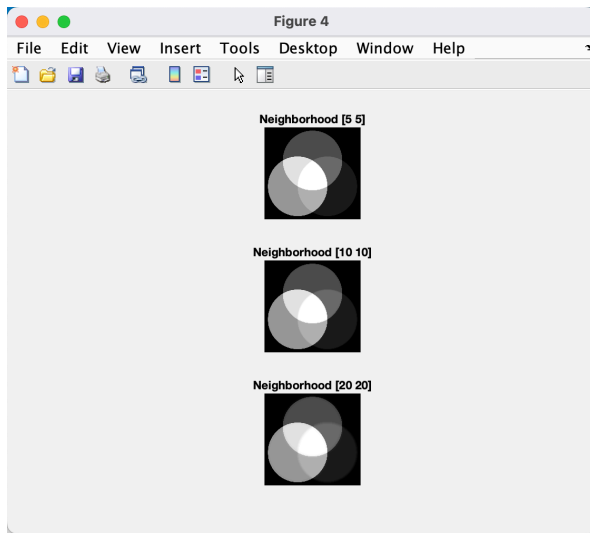


De esto se concluye que, al aumentar el valor de la desviación estándar, el resultado se ve reflejado como una imagen con menos contraste, haciendo que se vea borrosa.

#### 4) Aplicar un filtro Wiener

Nuevamente utilizando la imagen en escala de grises como base, se utiliza el comando `wiener2(I, [m n])`, que filtra la imagen en escala de grises  $I$  utilizando un filtro de Wiener adaptativo de paso bajo a nivel de píxel.  $[m n]$  especifica el tamaño ( $m$  por  $n$ ) del entorno utilizado (neighborhood) para estimar la media y la desviación estándar de la imagen local [2].

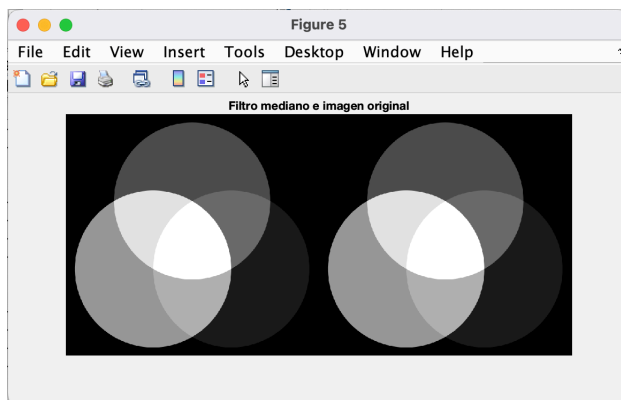
Se realizan filtrados con diferentes valores de entorno, resultando en la imagen que se muestra a continuación.



De estas diferentes pruebas se concluye que al aumentar el entorno, el contraste disminuye, haciendo que ya no se puedan distinguir algunos objetos.

## 5) Aplicar un filtro mediano

Utilizando el comando `medfilt2(I)` se aplica ahora un filtro mediano a la imagen en escala de grises original. Este es una operación no lineal que se utiliza a menudo en el procesamiento de imágenes para reducir el ruido "sal y pimienta" [3].

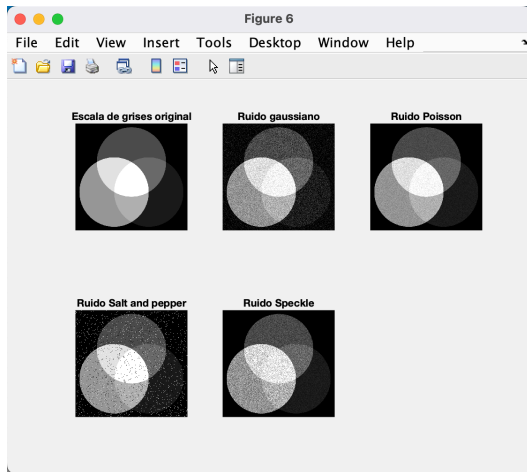


Para dar el formato que se presenta en la figura se utilizó el comando `montage({Imagen 1, Imagen 2})`.

Como la imagen original no presenta ruido, se ven muy parecidas.

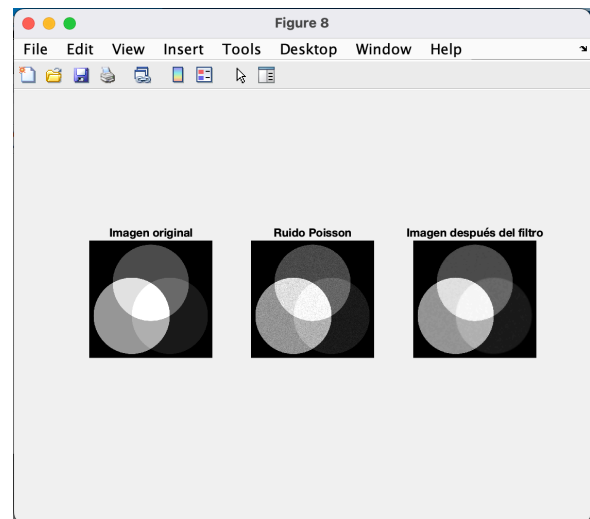
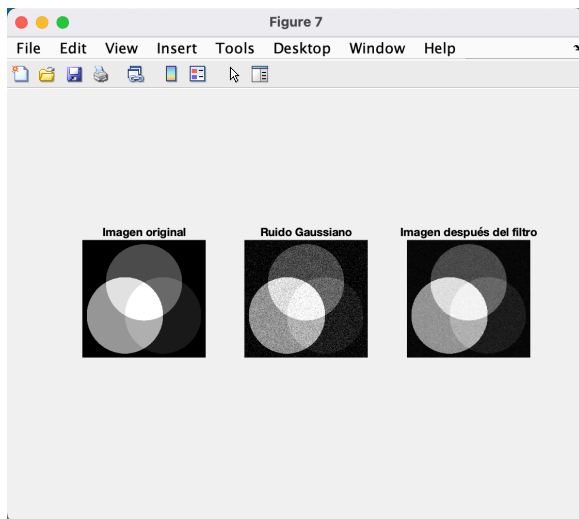
## 6) Añadir ruido a la imagen en escala de grises

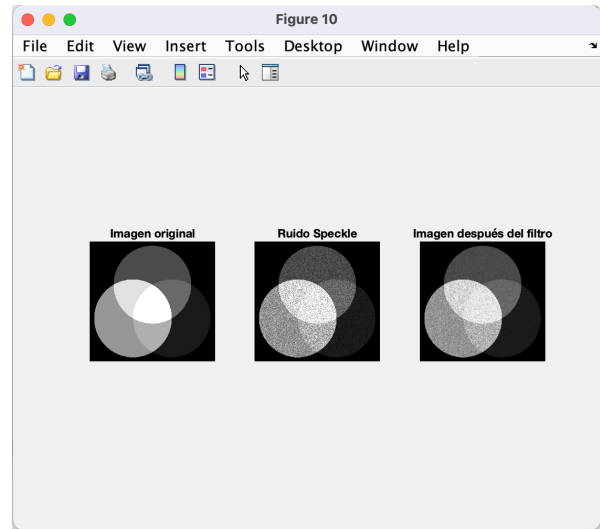
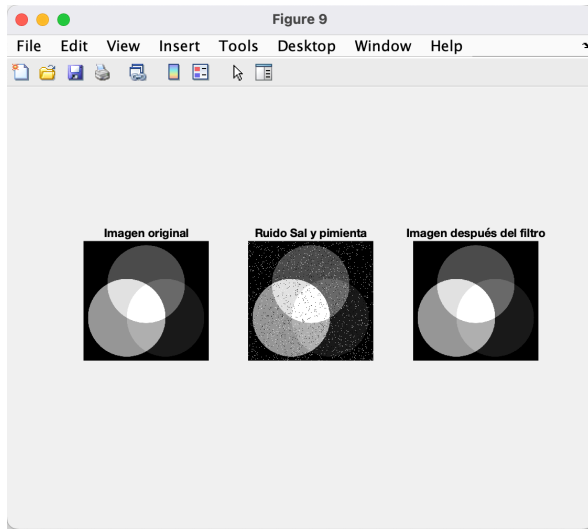
Para este paso se utiliza el comando `imnoise(I, TYPE,...)` en donde `TYPE` es el tipo de ruido que se desea añadir a la imagen. En este caso, se utilizaron los tipos gaussiano, Poisson, sal y pimienta y Speckle. En la misma figura se muestran los diferentes tipos de ruido, así como la comparación con la imagen sin ruido.



## 7) Aplicar filtros según el tipo de ruido

En este último punto se debía de escoger un tipo de filtro, adecuado para cada tipo de ruido. En primer lugar, como ya se sabía que el filtro mediano es de utilidad para el ruido de tipo sal y pimienta, se utiliza este para la figura 9. Después, se probaron todos los filtros para las demás imágenes, y se concluyó que todos los demás tipos de ruido se atenúan bastante bien a partir del filtro Wiener, así que este fue el que se eligió.





## Referencias

[1] Filtrado gaussiano 2D de imágenes - MATLAB imgaussfilt - MathWorks América Latina. (s. f.). <https://la.mathworks.com/help/images/ref/imgaussfilt.html>

[2] *Filtrado adaptativo 2D de eliminación de ruido - MATLAB wiener2 - MathWorks América Latina.* (s. f.). <https://la.mathworks.com/help/images/ref/wiener2.html>

[3] Filtrado de mediana de 2D - MATLAB medfilt2 - MathWorks América Latina. (s. f.). <https://la.mathworks.com/help/images/ref/medfilt2.html>