

Procesamiento de Imágenes y Visión por Computadora

TP2

Magalí Abigail Dumit

1. Implementar el detector de bordes por el método del gradiente utilizando los siguientes operadores de gradiente:



Original



Prewitt



Sobel

Implementar el detector de bordes por el método del gradiente utilizando los siguientes operadores de gradiente:

1. Los operadores de Prewitt y Sobel detectan bordes buscando el máximo cambio de intensidad (la mayor "pendiente" o gradiente) en la imagen.
2. Ambos funcionan bien, detectando los bordes principales. Sobel da un poco más de peso a los píxeles centrales, resultando en bordes ligeramente más robustos.



Prewitt

A histogram showing the distribution of edge detection results for the Prewitt operator. The x-axis represents the intensity of the detected edges, and the y-axis represents the frequency. The distribution is highly skewed to the right, with a very high peak at the lowest intensity values, indicating that most edges detected by Prewitt have low intensity. The frequency drops rapidly as intensity increases, forming a long tail.



Sobel

A histogram showing the distribution of edge detection results for the Sobel operator. The x-axis represents the intensity of the detected edges, and the y-axis represents the frequency. The distribution is highly skewed to the right, with a very high peak at the lowest intensity values, indicating that most edges detected by Sobel have low intensity. The frequency drops rapidly as intensity increases, forming a long tail.

2. Aplicar los detectores de borde del punto anterior a las mismas imágenes contaminadas con ruido.



Ruido Gaussiano Aditivo (σ
 ≈ 15 , porcentaje ≈ 0.2)



Prewitt

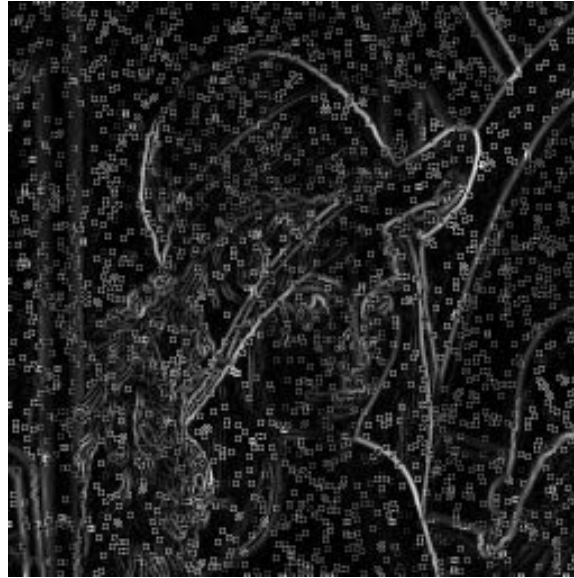


Sobel

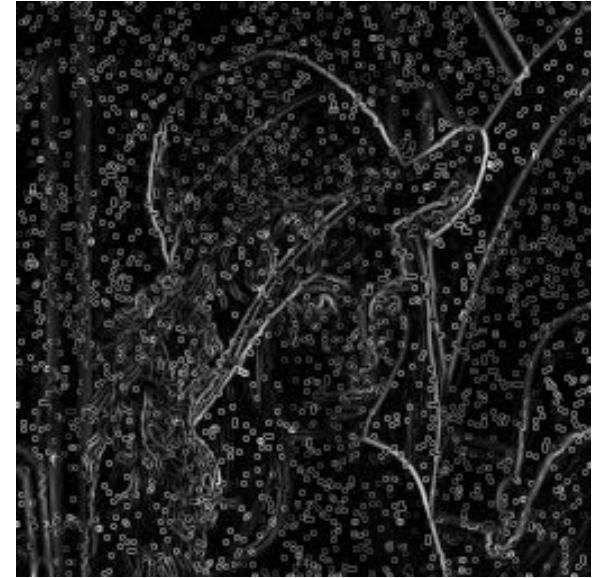
2. Aplicar los detectores de borde del punto anterior a las mismas imágenes contaminadas con ruido.



Ruido Sal y Pimienta
(densidad ≈ 0.05)



Prewitt



Sobel

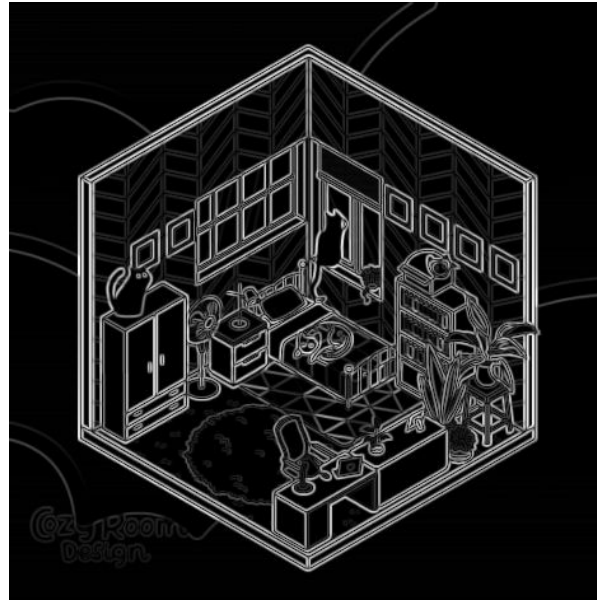
2. Aplicar los detectores de borde del punto anterior a las mismas imágenes contaminadas con ruido.

1. El resultado es un fracaso. Los ruidos Gaussiano y Sal & Pimienta crea miles de "pendientes" falsas, y los operadores los detectan todos.
2. Esto demuestra que los detectores de gradiente simples son **extremadamente sensibles al ruido** y no pueden usarse solos en una imagen contaminada.

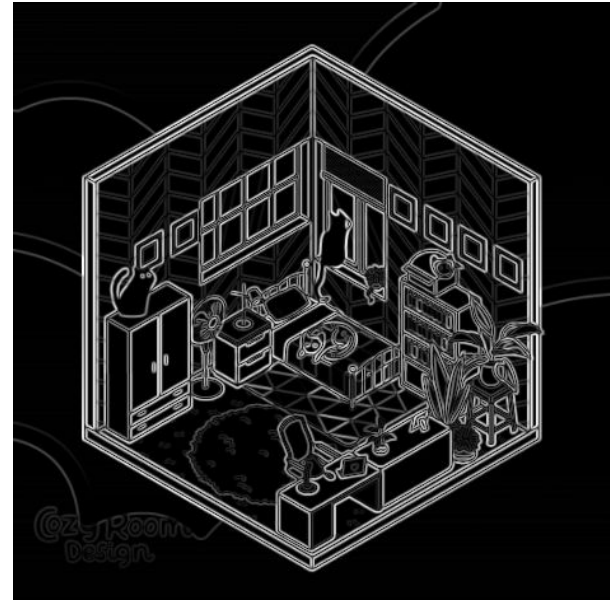
3. Aplicar los detectores de borde del punto anterior a imágenes en color.



Original

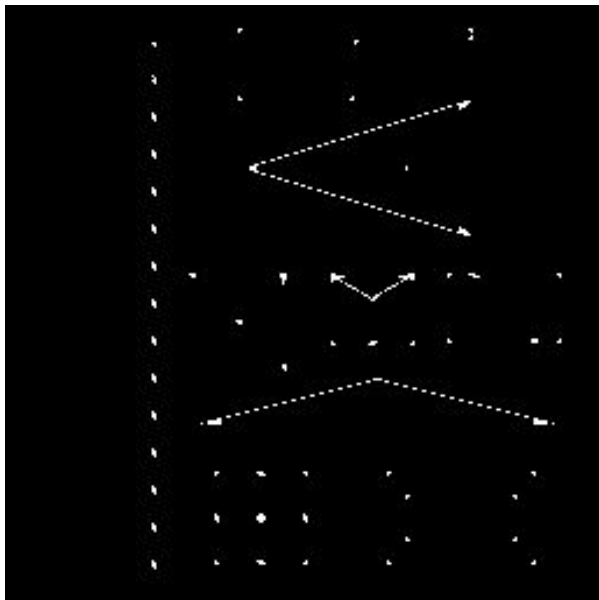


Prewitt

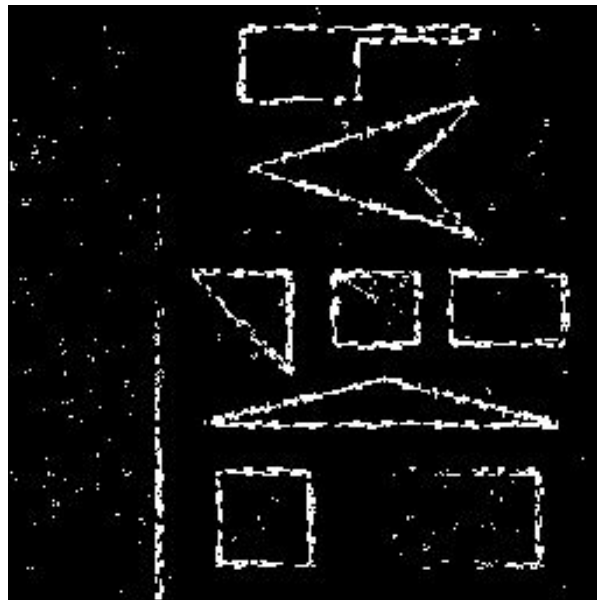


Sobel

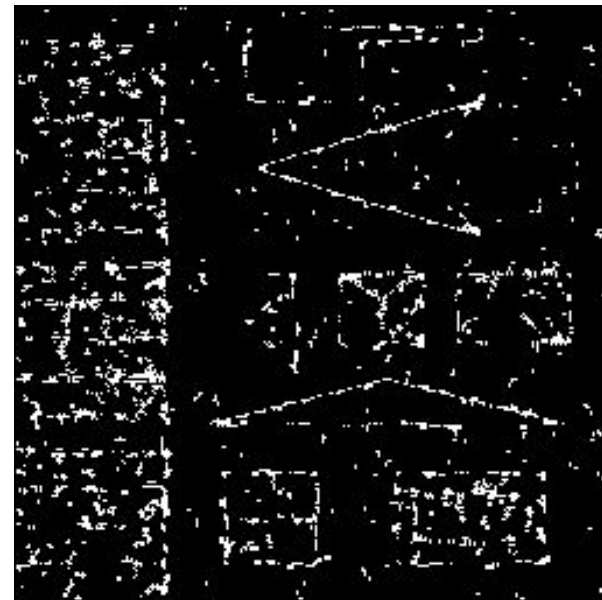
4. Implementar los siguientes detectores de borde y aplicarlos a dos imágenes y a sus versiones contaminadas:



Método del Laplaciano
(umbral: 15)

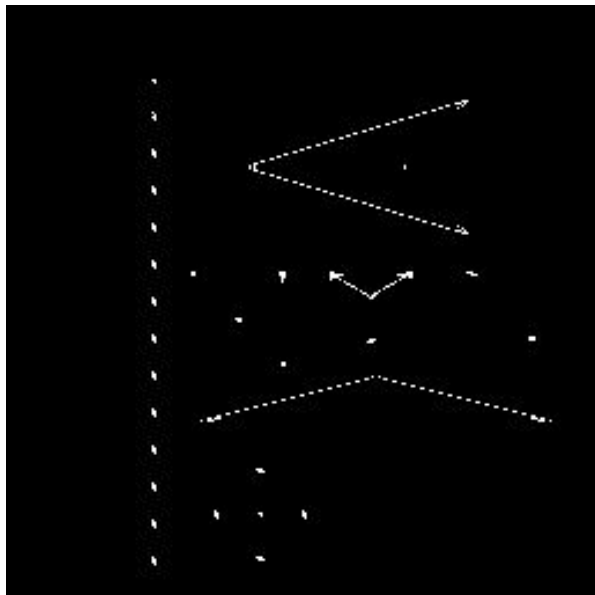


Ruido Gaussiano Aditivo
($\sigma \approx 15$, porcentaje ≈ 0.2)
umbral:170

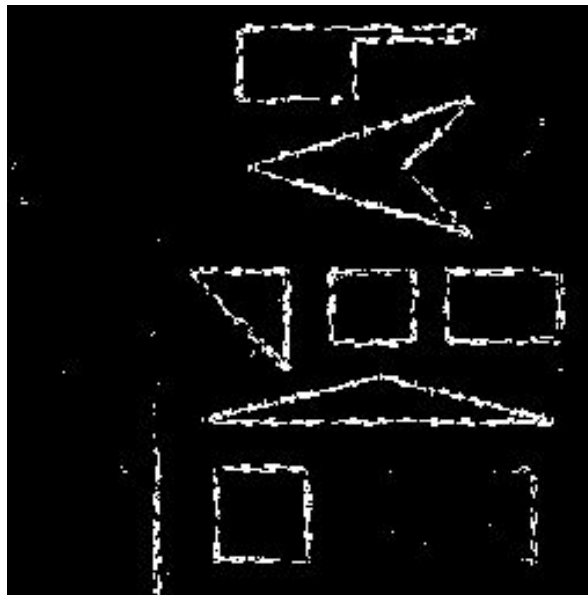


Ruido Sal y Pimienta
(densidad ≈ 0.05)
umbral: 100

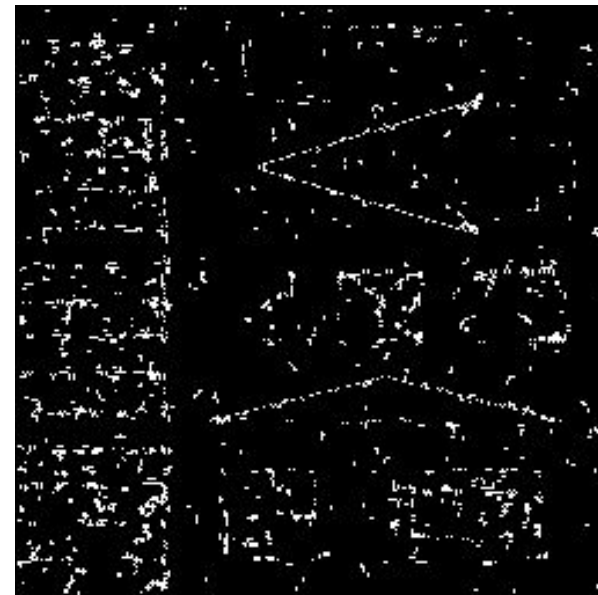
4. Implementar los siguientes detectores de borde y aplicarlos a dos imágenes y a sus versiones contaminadas:



**Método del Laplaciano con
evaluación de la pendiente
(umbral: 15)**

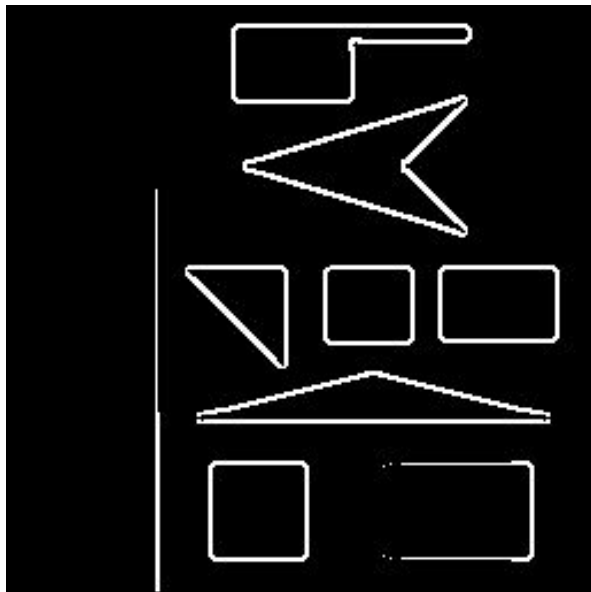


**Ruido Gaussiano Aditivo
($\sigma \approx 15$, porcentaje ≈ 0.2)
Umbral : 170**



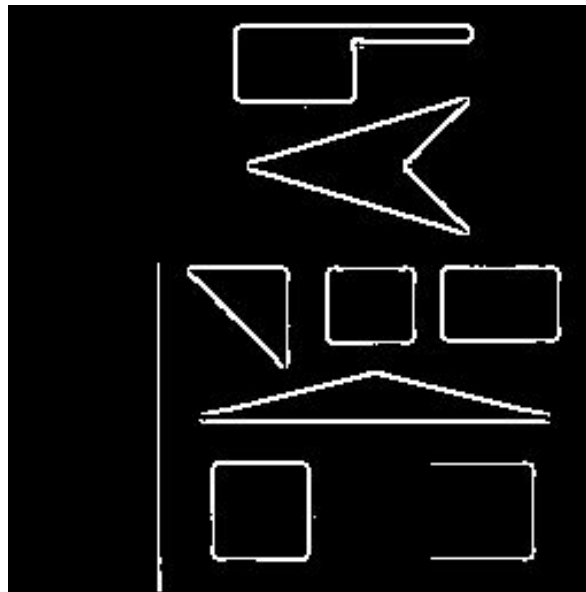
**Ruido Sal y Pimienta
(densidad ≈ 0.05)
Umbral : 100**

4. Implementar los siguientes detectores de borde y aplicarlos a dos imágenes y a sus versiones contaminadas:



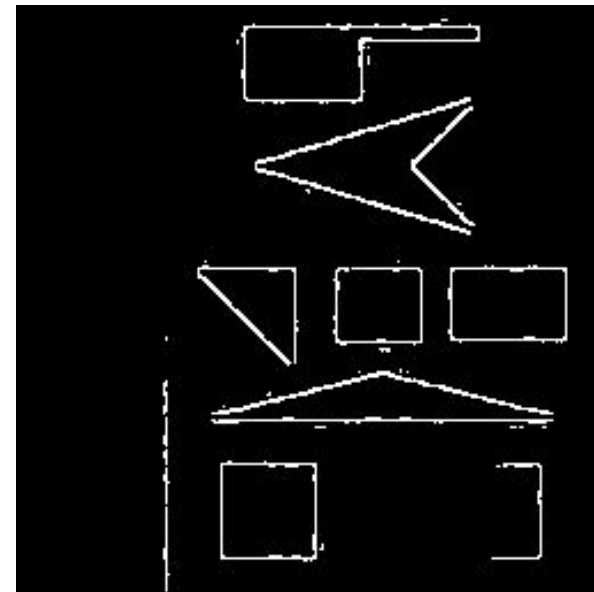
Método del Laplaciano del Gaussiano

Sigma: 2 y umbral : 15



Ruido Gaussiano Aditivo
($\sigma \approx 15$, porcentaje ≈ 0.2)

Sigma: 2 y umbral : 20



Ruido Sal y Pimienta
(densidad ≈ 0.05)

Sigma: 2 y umbral : 30

4. Implementar los siguientes detectores de borde y aplicarlos a dos imágenes y a sus versiones contaminadas:

1. Laplaciano: Busca el "cruce por cero", no el máximo. Esto localiza mejor el borde, pero es aún más sensible al ruido que Sobel/Prewitt.
2. Laplaciano con Pendiente: Una gran mejora. Filtra los bordes "débiles". Solo marca un cruce por cero si la pendiente (gradiente) en ese punto es alta.
3. LoG (Marr-Hildreth): El más robusto. Primero aplica un suavizado Gaussiano (para eliminar ruido) y *luego* busca los cruces por cero.

El LoG es el método superior aquí, ya que **ataca el problema del ruido *antes* de detectar el borde.**

5. Implementar los métodos de Difusión Isotrópica y Anisotrópica. Aplicarlos a imágenes con sal y pimienta. Comparar con el filtro de la mediana.



Difusión Isotrópica
Núm interacciones: 1

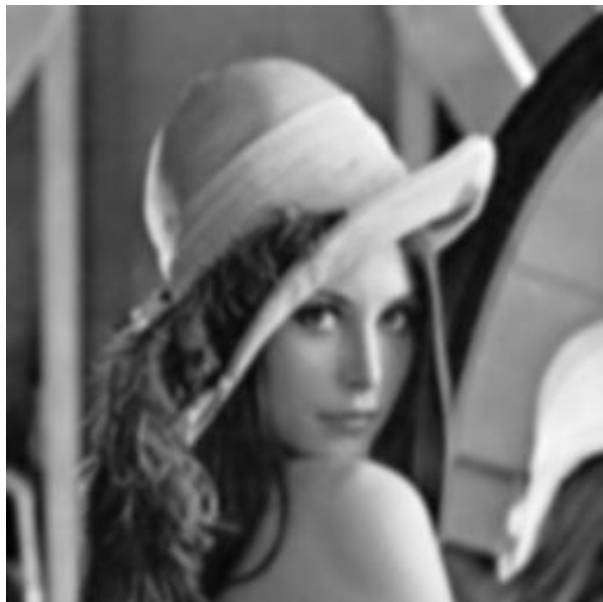


Difusión Anisotrópica
Núm interacciones: 15, $k : 15$



Filtro de la Mediana
Máscara : 3

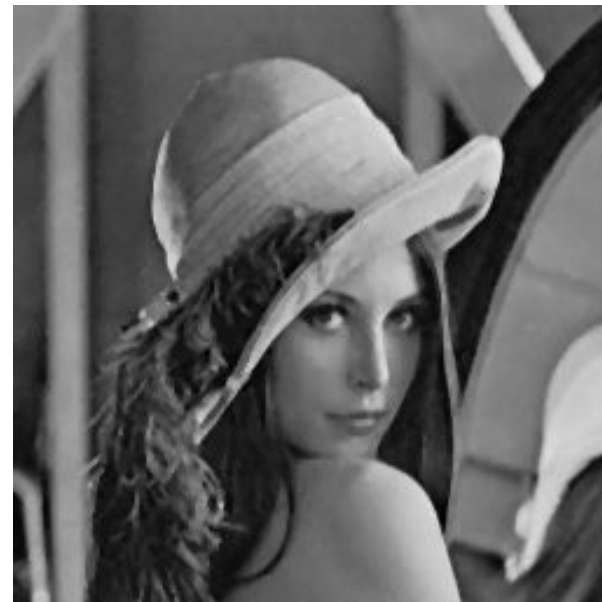
5. Implementar los métodos de Difusión Isotrópica y Anisotrópica. Aplicarlos a imágenes con ruido gaussiano. Comparar con el filtro de la mediana.



Difusión Isotrópica
Núm interacciones: 1



Difusión Anisotrópica
Núm interacciones: 15, $k : 15$



Filtro de la Mediana
Máscara : 3

Filtro de Mediana:

- **Excelente** para ruido **Sal y Pimienta (S&P)**. Elimina los píxeles atípicos (0 y 255) limpiamente.
- Regular para ruido Gaussiano (crea un efecto "plástico").

Difusión Isotrópica:

- Suaviza el ruido Gaussiano, pero **destruye los bordes** (los vuelve borrosos).

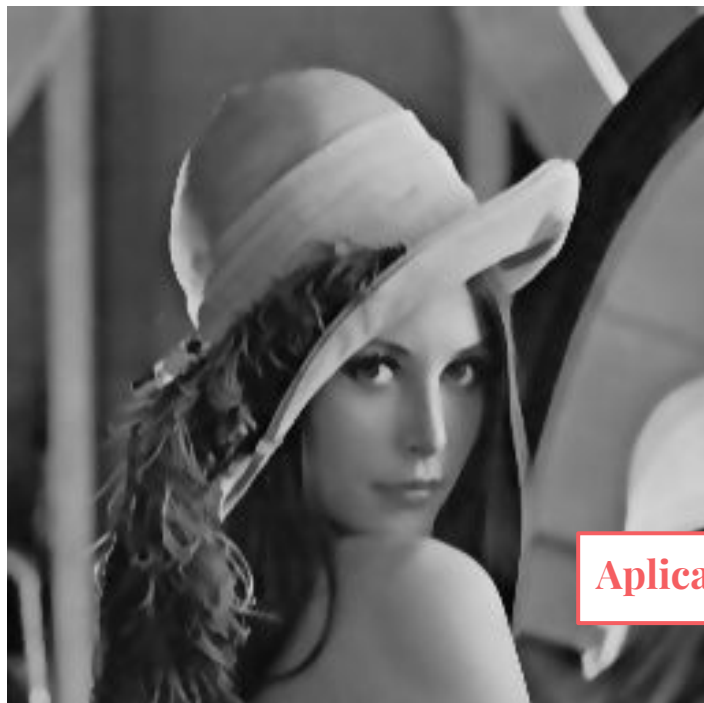
Difusión Anisotrópica:

- **Excelente** para ruido **Gaussiano**. Es un filtro "inteligente": suaviza las regiones planas pero **respeto y preserva los bordes**.

No hay un filtro "mejor". Se usa **Mediana para S&P** y **Anisotrópico para Gaussiano** si se quieren preservar los bordes.

6. Implementar el filtro bilateral. Aplicarlo a imágenes con ruido gaussiano y con ruido sal y pimienta. Comparar con el filtro de difusión anisotrópica.

Filtro Bilateral



Difusión Anisotrópica



Aplicados sobre LENA.raw

6. Implementar el filtro bilateral. Aplicarlo a imágenes con ruido gaussiano y con ruido sal y pimienta. Comparar con el filtro de difusión anisotrópica.

Filtro Bilateral



Difusión Anisotrópica



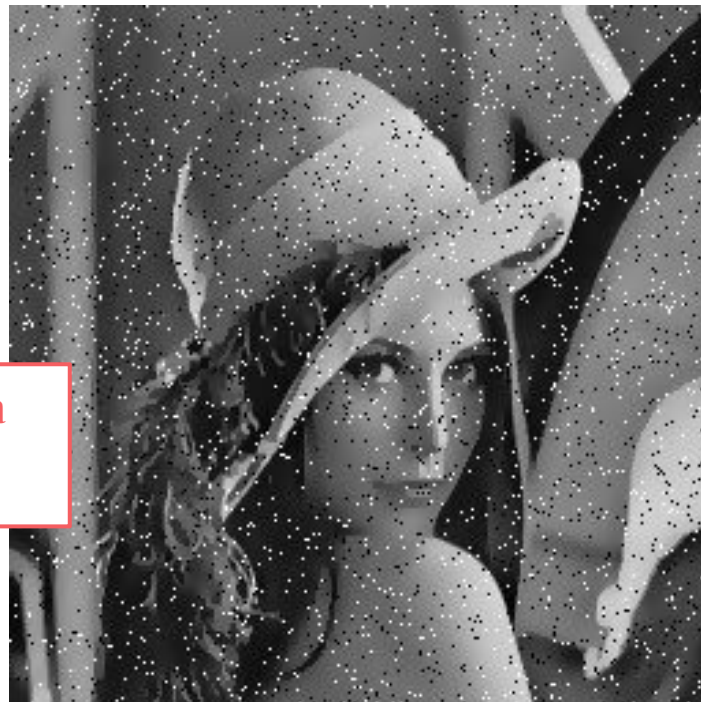
Ruido Gaussiano Aditivo
($\sigma \approx 15$, porcentaje ≈ 0.2)

6. Implementar el filtro bilateral. Aplicarlo a imágenes con ruido gaussiano y con ruido sal y pimienta. Comparar con el filtro de difusión anisotrópica.

Filtro Bilateral



Difusión Anisotrópica



Ruido Sal y Pimienta
(porcentaje ≈ 0.02)

6. Implementar el filtro bilateral. Aplicarlo a imágenes con ruido gaussiano y con ruido sal y pimienta. Comparar con el filtro de difusión anisotrópica.

Se aplican los dos filtros avanzados (Bilateral y Anisotrópico) a una imagen con ruido Sal y Pimienta.

Ambos filtros fallan. Tratan los puntos de ruido (0 y 255) como bordes reales y los preservan, en lugar de eliminarlos.

Esto demuestra que para ruido S&P, el filtro de **Mediana (del Punto 5)** es la **única solución correcta**. Estos filtros avanzados son específicos para ruido Gaussiano.

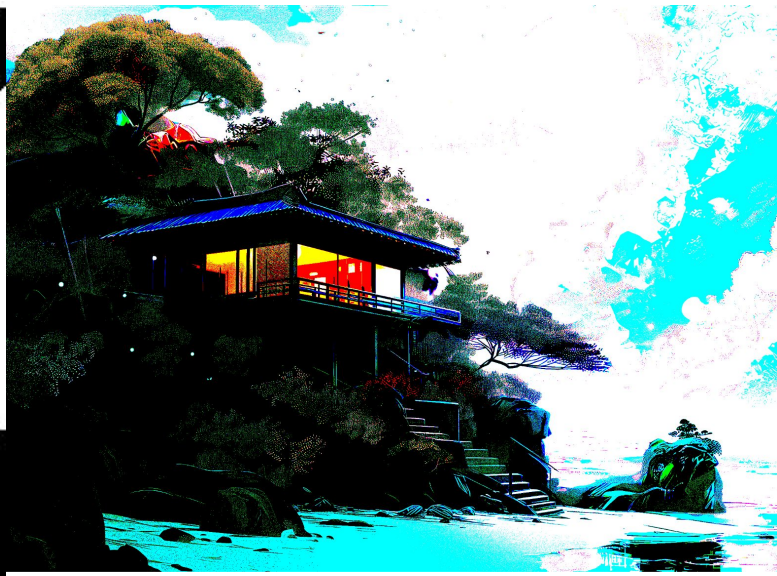
7. Implementar los siguientes algoritmos de umbralización y aplicarlos a dos imágenes y a sus versiones contaminadas: Umbralización sin filtros



Umbralización óptima
iterativa.

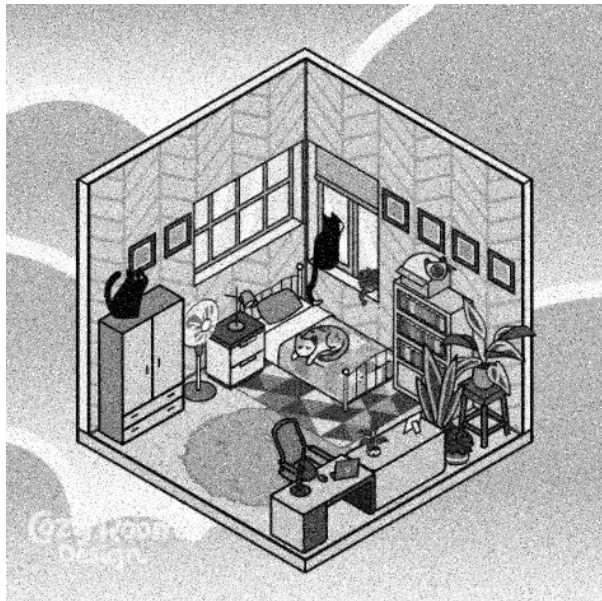


Umbralización de Otsu

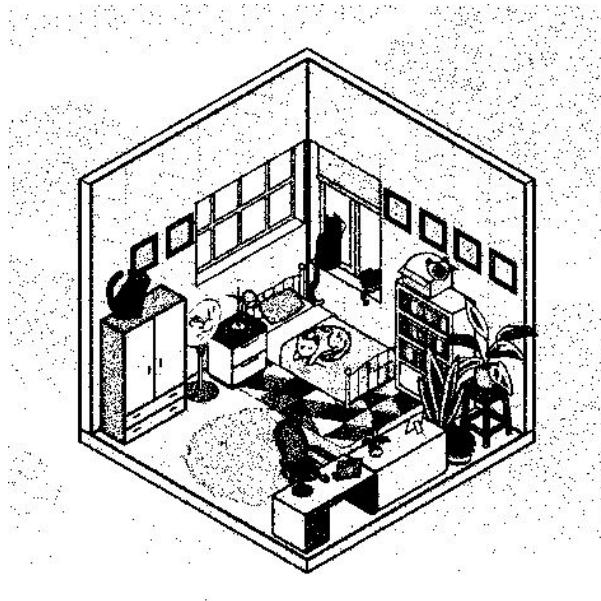


Umbralización de Otsu (RGB)

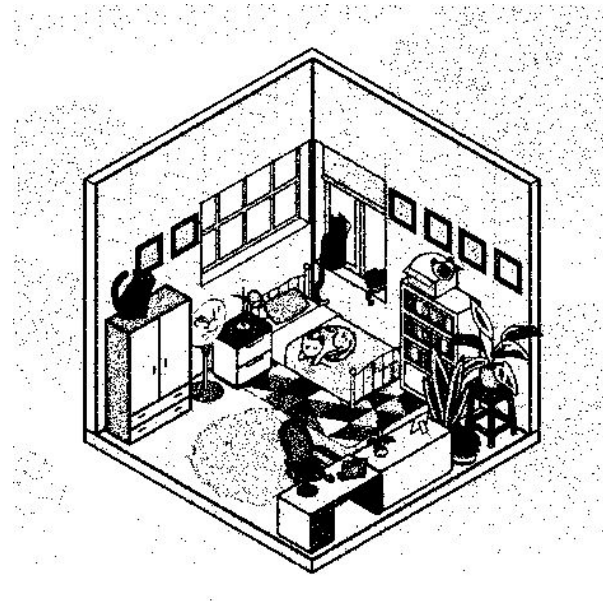
7. Implementar los siguientes algoritmos de umbralización y aplicarlos a dos imágenes y a sus versiones contaminadas: Umbralización con Ruido Gaussiano



Con ruido gaussiano

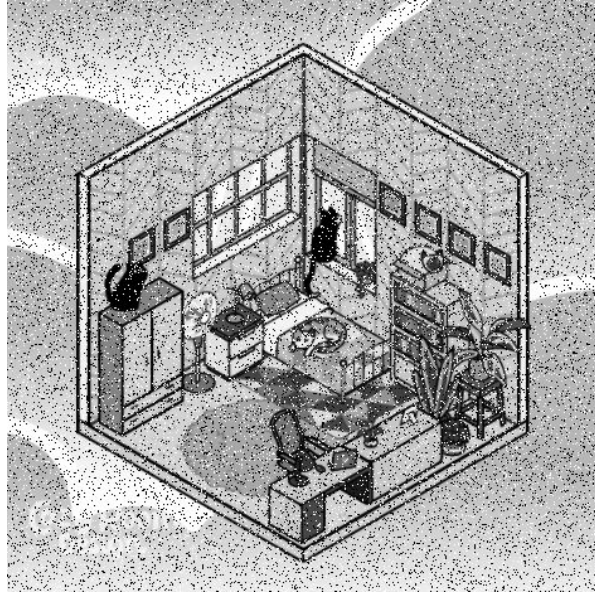


Umbralización óptima
iterativa.

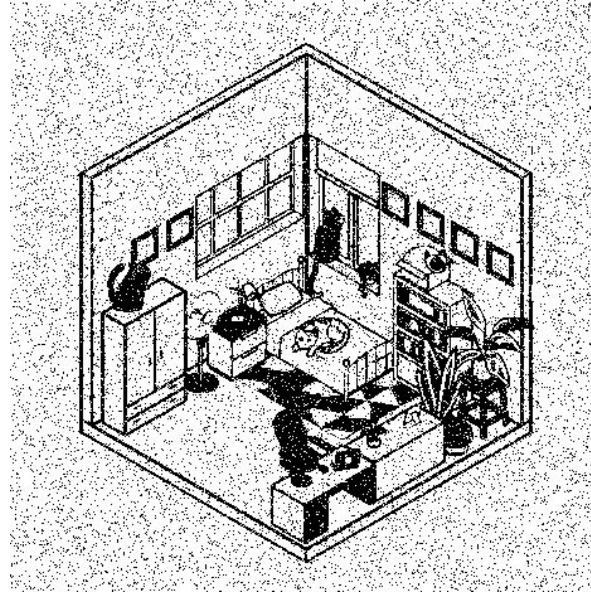


Umbralización de Otsu

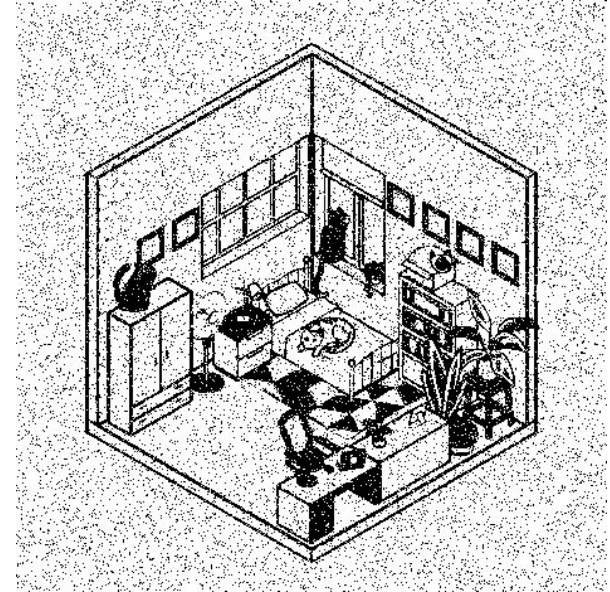
7. Implementar los siguientes algoritmos de umbralización y aplicarlos a dos imágenes y a sus versiones contaminadas: Umbralización con Sal y Pimienta



Con ruido sal y pimienta



Umbralización óptima
iterativa.



Umbralización de Otsu

Se intenta segmentar (Otsu e Iterativo) una imagen contaminada con ruido S&P.

Es un fracaso total. Los píxeles de ruido se clasifican incorrectamente, resultando en una segmentación "sucia" e inutilizable.

La umbralización NUNCA debe aplicarse sobre ruido S&P.

Primero se debe aplicar un **Filtro de Mediana** para limpiar la imagen, y recién después umbralizar.