E.T.S. de Ingeniería Industrial,
Informática y de Telecomunicación

Filtrado de imagen utilizando el algoritmo Non-Local Means. Aplicación Android

Grado en Ingeniería Informática

Trabajo Fin de Grado

Iosu Rodríguez Alfaro

Daniel Paternain Dallo, Aránzazu Jurío Munarriz

Pamplona, 27 de Junio de 2014

upna Universidad Pública de Navarra
Nafarroako Unibertsitate Publikoa

Índice general

Resumen

En procesamiento de imagen uno de los tópicos más estudiados es el de eliminación de ruido. El ruido puede aparecer en la captura o en la transmisión de la imagen. El objetivo de los algoritmos de filtrado consiste en la eliminación de ruido tratando de preservar las características más importantes de la imagen (bordes, textura, intensidad). Uno de los algoritmos que más atención están recibiendo en la actualidad es el de Non-local Means (Buades et al.). La particularidad de este filtro es que, a diferencia de los filtros 'locales' de suavizado, el valor de los píxeles se modifica por un promedio de los píxeles considerados más similares independientemente de su localización en la imagen.

En la actualidad, los Smartphones (teléfonos inteligentes) están muy presentas en la sociedad. Estos teléfonos trabajan con sistemas operativos propios como Android, iOS, Windows Phone, etc. Asimismo, los teléfonos integran cámaras cada vez de mayor resolución. Pero, pese a ello, muchas de las imágenes tomadas son de mala calidad debido a problemas de iluminación, movimiento o ruido en general.

El objetivo de este trabajo es estudiar en profundidad el algoritmo Non-Local Means y su posible implementación en una aplicación de Android.

Capítulo 1: Introducción y objetivos

1.1. Procesamiento digital de imagen (PDI):

En la actualidad el procesamiento digital de imagen es un campo empleado en numerosas áreas, pero ¿qué es el procesamiento digital de imagen?

El procesamiento digital de imagen es el conjunto de técnicas que permiten mejorar la calidad o facilitar la búsqueda de información en una imagen por medio de una computadora. Se emplea principalmente para mejorar la calidad de la información para la interpretación humana y para procesar los datos de la imagen para su almacenamiento, transmisión y representación.

Medicina, astronomía, seguridad, diagnóstico médico, biología, industria, radares, inspección industrial, etc., son ejemplos de algunas de las áreas donde se emplea el PDI.

El PDI se compone de una serie de pasos que nos permiten obtener imágenes de buena calidad, con respecto a la original. Estos pasos son:

1. Adquisición de la imagen.
2. Realce de la imagen. Consiste en resaltar detalles de nuestro interés.
3. Restauración de la imagen. Consiste en mejorar la apariencia de la imagen.
4. Procesamiento de imagen en color.
5. Compresión. Consiste en reducir el tamaño de la imagen para su almacenamiento o reducir su ancho de banda para su transmisión.
6. Procesos morfológicos. Consiste en extraer los componentes de una imagen que son útiles para la representación y descripción de formas.
7. Segmentación. Se utiliza para extraer o aislar objetos de una imagen para su posterior análisis.
8. Representación y descripción.
9. Reconocimiento.

En este trabajo nos centramos en la fase de restauración de la imagen. Dentro de esta fase los algoritmos más comunes son los algoritmos de filtrado. Concretamente, nosotros vamos a trabajar sobre uno de los algoritmos más estudiados en la literatura: el algoritmo Non Local Means.

1.2. Smartphones y aplicaciones móviles:

Los Smartphone o teléfonos inteligentes están a la orden del día. Cualquier persona que tenga uno a su disposición puede navegar por internet desde cualquier lado, llamar por teléfono e incluso saber la localización de sus amigos. Actualmente existe una amplia gama de dispositivos móviles, aptos para todos los públicos. Cada vez hay más competencia en este sector y las empresas compiten por tener el Smartphone más completo. Estos teléfonos móviles inteligentes constan de un sistema operativo además de procesador, memoria RAM, etc. Los sistemas operativos más conocidos para tecnologías móviles son Android, iOS y Windows phone. Sin embargo en las últimas fechas se están dando a conocer más sistemas operativos. Cada uno de los sistemas operativos cuenta con algún tipo de aplicación que permite descargar e instalar todo tipo de aplicaciones en el dispositivo móvil.

Android es sin duda uno de los sistemas operativos más empleados. Múltiples marcas conocidas mundialmente fabrican móviles Android. Esta es la gran diferencia con Apple, cuyo sistema operativo iOS solo se utiliza en los teléfonos móviles que él fabrica, los iPhone.

También una de las especificaciones más demandadas por los usuarios es que el Smartphone tenga una buena cámara. Muchos de ellos poseen cámaras de alta resolución, pero pese a ello la calidad de las imágenes no es siempre la deseada. En muchas ocasiones aparece ruido debido a problemas de iluminación, movimiento, etc. Por esta razón, el desarrollo de aplicaciones móviles para tratamiento de imagen es de vital importancia.

1.3. Objetivos:

Los objetivos de este trabajo fin de grado son los siguientes:

- Aprender nuevas técnicas para el filtrado de imágenes con ruidos.
- Estudiar el funcionamiento del algoritmo Non Local Means para el filtrado de imágenes.
- Realizar varias implementaciones en Matlab de dicho algoritmo y seleccionar la más eficiente y la que mejor resultados aporte. Aplicarla tanto a imágenes en escala de grises como en imágenes a color.
- Realizar un estudio para obtener la mejor combinación de parámetros para el algoritmo Non Local Means y aplicarlo a imágenes con distintos tipos de ruido.
- Comparar los resultados obtenidos con los resultados de aplicar otros filtros comunes.
- Sacar conclusiones sobre las distintas implementaciones.
- Familiarizarse con el entorno de Android y aprender a programar aplicaciones Android.
- Realizar una pequeña aplicación para Android que implemente el algoritmo Non Local Means.
- Ver posibles mejoras para la aplicación Android.

Capítulo 2: Conceptos previos

2.1. Tipos de ruido

2.1.1. Ruido Gaussiano

El valor final del píxel es el ideal más una cierta cantidad de error. Este error viene dado por la desviación típica de la distribución gaussiana.
Produce pequeñas variaciones en la imagen.
Suele ser debido a los componentes electrónicos (sensores, digitalizadores...)
Afecta a la imagen completa. La intensidad de todos los píxeles se ve alterada.
Su función de densidad de probabilidad esta expresada por la curva gaussiana definida como:

$$p(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(x-\mu)^2}{2\sigma^2}}$$

μ es la media y σ es la desviación típica de la variable aleatoria.

Ejemplo de imagen con ruido gaussiano:

Imagen original

Imagen con ruido gaussiano con σ=30

2.1.2. Ruido Rician

Suele aparecer en imágenes de resonancia magnética. Tiene un comportamiento parecido al ruido gaussiano.
La distribución rician viene dada por la siguiente expresión:

$$p(m) = \frac{m}{s_n^2} \exp\left(-\frac{m^2 + A^2}{2s_n^2}\right) \mathrm{I}_0\left(\frac{Am}{s_n^2}\right)$$

Donde Sn es la desviación típica del ruido gaussiano en el dominio complejo.

m es el tamaño de la imagen,

I0 es la función de Bessel de orden 0 modificada,

A es la amplitud de la señal sin ruido.

Ejemplo de imagen con ruido rician:

Imagen con ruido rician con σ=40

2.1.3. Ruido impulsivo Sal y Pimienta

Es un tipo de ruido impulsivo.
El valor que toma el píxel no tiene relación con el valor ideal sino que toma valores muy altos o muy bajos. Es decir, píxeles aleatorios de colores completamente blancos o completamente negros. Toma el valor máximo (sal) o el mínimo (pimienta).
Es producido por el mal funcionamiento de los píxeles en los sensores de una cámara, en situaciones de memoria defectuosas en el hardware o por transmisión de un canal de ruido.

Ejemplo de imagen con ruido sal y pimienta:

Imagen con ruido sal y pimienta

2.1.4. Ruido Uniforme

El ruido que afecta a la imagen sigue una distribución uniforme.
La probabilidad de tomar cualquier valor de gris de dentro de un intervalo definido es constante.

Existen dos tipos de ruido uniforme:

Ruido uniforme frecuencia:

La imagen obtenida es la real más una interferencia de señal periódica, (seno, coseno...).

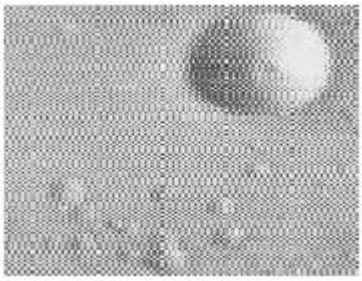


Imagen original

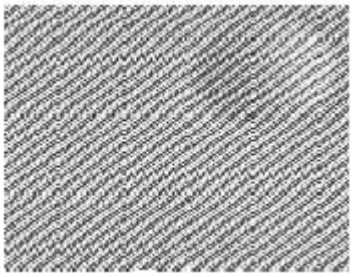


Imagen con ruido frecuencia

Ruido uniforme multiplicativo:

La señal obtenida es fruto de la multiplicación de dos señales.

Imagen original

Imagen con ruido multiplicativo

2.1.5. Ruido Speckle

Las imágenes de ultrasonido presentan un tipo especial de ruido llamado Speckle. Este tipo de ruido degrada significativamente la calidad de la imagen, aumentando de esta forma la dificultad de observar detalles finos en las imágenes durante un examen de diagnóstico. También dificulta el procesado de las imágenes, como la segmentación y la detección de bordes.

Los tipos de ruido que se van a tratar son el gaussiano por ser el más común, y el rician por tener un comportamiento similar al gaussiano. También porque es común en imágenes de resonancia magnética y me parecía interesante ver si era posible eliminar ruido en ese tipo de imágenes con el algoritmo Non Local Means.

2.2. Tipos de filtros

Podemos clasificar los filtros según su dominio, en dos clases:
Filtros de dominio espacial y filtros de dominio frecuencia. Nos centraremos en los filtros de dominio espacial.

2.2.1. Filtros de dominio espacial

Definimos un filtro de dominio espacial como

$$G(x,y)=T(f(x,y))$$

Donde f(x,y) es un píxel de la imagen de entradla, g(x,y) es un píxel de la imagen de salida, T es una operación en f definida sobre la vecindad del píxel (x,y).

Generalmente, entenderemos el vecindario de (x,y) como una ventana de mxn píxeles centrada en el píxel (x,y).
Para el filtrado espacial, se definen máscaras o Kernel del mismo tamaño que el vecindario.
Se utilizan principalmente para reducir ruido en imágenes. Este suele ser el paso previo a la detección de borde, segmentación, etc. Estos filtros operan sobre la vecindad del píxel que se está tratando. Dos categorías de filtros de dominio espacial muy comunes son la de los filtros promedios y los filtros de orden.

2.2.1.1. Filtros promedios

Operan sobre la vecindad mediante sumas y multiplicaciones. La idea es que al cambiar cada píxel por el promedio de los vecinos reducimos considerablemente los detalles y eliminamos cambios abruptos de intensidad. Algunos de estos filtros son:

Filtro de la media aritmética:

El valor de cada píxel se calcula según la siguiente expresión:

$$g(x,y) = \frac{1}{mn} \sum_{(r,c)\epsilon Sxy} f(r,c)$$

donde r,c son las coordenadas de los píxeles que forman la vecindad del píxel (x,y). La máscara para este filtro, es una máscara donde todos los valores son iguales. Es decir se ponderan todos los píxeles del vecindario con el mismo valor.

Filtro Media ponderada:

La media ponderada se basa en que los valores pueden multiplicarse por distintos coeficientes. Es decir, podemos dar más importancia a algunos píxeles.
Un ejemplo de máscara puede ser el siguiente:

$\frac{1}{16} \times$

1	2	1
2	4	2
1	2	1

En general podemos expresar un filtro de estas características como:

$$g(x,y) = \frac{\sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t) f(x+s,y+t)}{\sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t)}$$

Donde w representa la máscara, s y t son cada elemento de la máscara y a y b son el número de filas y columnas respectivamente, que tiene la máscara.

Filtro gaussiano:

Permiten definir máscaras de filtro con forma de gaussiana discreta.
Cada elemento w(x,y) de la máscara viene dada por:

$$w(x,y) = e^{-\frac{x^2+y^2}{2\sigma^2}}$$

Este filtro tiene las siguientes propiedades:

- En 2 dimensiones, las funciones gaussianas son simétricas respecto a rotación. Es decir, la cantidad de filtrado va a ser la misma en todas las direcciones.
- Actúa como una media ponderada dando mucha más importancia al píxel central. Además, la importancia de la vecindad decrece con la distancia al centro.
- La anchura de la gaussiana y, en consecuencia, la cantidad de suavizado está parametrizada por el valor de sigma.

2.2.1.2. Filtros de orden

También operan sobre la vecindad pero no mediante sumas y/o multiplicaciones.
Los más utilizados son los basados en proyecciones ordenadas. Funcionan de la siguiente manera:

1. Para cada píxel, tomamos los píxeles vecinos.
2. Ordenamos todos los píxeles.
3. Tomamos como resultado uno de los píxeles (según el orden) o una combinación de ellos.

Algunos de los filtros de orden más conocidos son:

Filtro de la mediana:

Reemplazamos cada píxel por la mediana de los píxeles del vecindario. Muy útil para eliminar ruido impulsivo. Además preserva los bordes mejor que los filtros promedio.

Filtro del mínimo / máximo:

Útiles para detectar puntos brillantes / oscuros en la imagen.

$$g(x,y) = \min_{(s,t)\in Sxy}\{f(s,t)\}$$
$$g(x,y) = \max_{(s,t)\in Sxy}\{f(s,t)\}$$

2.3. Métodos de medición de calidad de imagen:

Se utilizan para saber cuánta información se ha perdido durante el procesamiento de la imagen, o para saber lo que se parecen dos imágenes. Algunos de los métodos más comunes son:

2.3.1. Error Cuadrático Medio (Mean Square Error)

Viene dado por la siguiente expresión:

$$MSE = \frac{1}{MN}\sum_{x=1}^{M}\sum_{y=1}^{N}(f'(x,y) - f(x,y))^2$$

Donde f'(x,y) es el valor del píxel en la imagen original y f(x,y) es el valor del píxel en la imagen obtenida tras aplicar una técnica de procesamiento de imágenes.
Cuanto menor valor más parecida es la imagen resultante a la imagen original.
En el límite MSE = 0.

2.3.2. Peak Signal-to-Noise Ratio (PSNR)

Viene dado por la siguiente expresión:

$$PSNR = 10\log_{10}\frac{(L-1)^2}{MSE}$$

Donde MSE es el error cuadrático medio y L es el rango dinámico. Por ejemplo si trabajamos con imágenes cuyas intensidades toman valores entre 0 y 255, L seria 256.
Cuanto mayor valor más parecida es la imagen resultante a la imagen original.

En el límite PSNR = +∞.

2.3.3. Índice de Similitud Estructural (SSIM)

Es un método para medir la similitud entre dos imágenes. En otras palabras, la medición de la calidad de la imagen basada en una imagen sin comprimir o sin distorsión inicial como referencia. SSIM está diseñado para mejorar los métodos tradicionales, como el PSNR y el MSE, que han demostrado ser incompatibles con la percepción del ojo humano.
Viene dado por la siguiente expresión:

$$\mathrm{SSIM}(x,y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}$$

Donde μ_x es la media de x , μ_y es la media de y , σ_x^2 es la varianza de x , σ_y^2 es la varianza de y , σ_{xy} es la covarianza de x e y , y $c_1=(k_1 L)^2$, $c_2=(k_2 L)^2$ dos variables para estabilizar la división con denominadores bajos. L es el rango dinámico de los valores de los píxeles y $k_1=0.01$ y $k_2=0.03$ de forma predeterminada.

El resultado del SSIM será un valor entre 0 y 1. A mayor valor, más similares son las imágenes.

En los distintos experimentos se obtendrá tanto el PSNR como el SSIM, pero únicamente se tendrá en cuenta el SSIM por lo citado anteriormente. Los mejores valores del PSNR no correspondían realmente con las imágenes de mejor resultado visual. Sin embargo, empleando el SSIM se ha solucionado este problema.

Capítulo 3: Filtro Non Local Means

3.1. Introducción

El filtro Non Local Means (NLMs) o medias no locales es un algoritmo de procesamiento de imagen para la eliminación de ruido. La particularidad de este filtro es que, a diferencia de los filtros 'locales' de suavizado, el valor de los píxeles se modifica por un promedio de los píxeles considerados más similares independientemente de su localización en la imagen. Esto se traduce en mucha mayor claridad tras el filtrado, y menos pérdida de detalle en la imagen.

Descripción del algoritmo:

Dada una imagen con ruido discreto $v = \{ v(i) | \, i \in I \}$, el valor estimado $NL[v](i)$, para un píxel i, es computado como una media ponderada de todos los píxeles de la imagen,

$$NL[v](i) = \sum_{j \in I} w(i,j) v(j) \text{ (1)},$$

donde el conjunto de pesos $\{w(i,j)\}j$, depende de la similitud entre el píxel i y el píxel j, y satisface las la condición de $0 \leq w(i,j) \leq 1$ y $\sum j \, w(i,j) = 1$.

La similitud entre 2 píxeles i y j depende de la similitud de la intensidad de los niveles de gris del vector $v(N_i)$ y $v(N_j)$, donde N_k es el vecindario cuadrado de tamaño fijado y centrado en el píxel k. Esta similitud es medida como una función decreciente de la distancia Euclídea ponderada, $||v(N_i) - v(N_j)||^2_{2,a}$, donde $a > 0$ es la desviación estándar del kernel Gausiano,
Los píxeles con un vecindario similar en niveles de grises a $v(N_i)$ tienen mayor peso en el promedio.
Estos pesos están definidos como,

$$w(i,j) = \frac{1}{Z(i)} e^{\frac{-||v(N_i) - v(N_j)||^2_{2,a}}{h^2}} \text{ (2)},$$

donde $Z(i)$ es la constante de normalización

$$Z(i) = \sum_j e^{\frac{-||v(N_i) - v(N_j)||^2_{2,a}}{h^2}} \text{ (3)},$$

y el parámetro h actúa como grado de filtrado. Este parámetro controla la decadencia de la función exponencial y por lo tanto el de los pesos.

En nuestras implementaciones, se introducirá también un radio de búsqueda, de forma que para cada píxel no será necesario volver a recorrer todos los píxeles de la imagen, sino que bastará con aquellos que estén dentro de ese radio. Por tanto la expresión (1) seria modificada de la siguiente forma:

$$NL[v](i) = \sum_{j \in R} w(i,j) v(j), \text{ (4)}$$

donde R es la ventana cuyo píxel central es el píxel i, y todos los píxeles j, son aquellos que están a una distancia de i menor o igual que el radio de búsqueda dado.

En algunas de las implementaciones, se trabajará con la distancia euclídea sin ponderar, es decir, se les dará el mismo peso a cada uno de los píxeles del vecindario. En otras se hará uso de la distancia euclídea ponderada por un Kernel gaussiano.
Para la distancia euclídea sin ponderar, la expresión $||v(N_i) - v(N_j)||^2_{2,a}$ viene dada por

$$\sum_{j=1}^{k} \left(x_j - y_j\right)^2, \text{ (5)}$$

y para la distancia euclídea ponderada viene dada por,

$$\sum_{j=1}^{k} a_j \left(x_j - y_j\right)^2, \text{ (6)}$$

donde k es el número de píxeles de la vecindad, x e y corresponden con píxeles de cada vecindad y a corresponde con el Kernel gaussiano.

Complejidad del algoritmo:

Cuando no empleamos radio de búsqueda, la complejidad del algoritmo para una imagen de tamaño NxM píxeles y una vecindad v es, $O(NMxNMxV)$, donde V es una ventana de tamaño $(2v+1)^2$ píxeles.

Cuando empleamos radio de búsqueda, la complejidad del algoritmo para una imagen de tamaño NxM píxeles, una vecindad v y un radio de búsqueda r, es, $O(NMxRxV)$, donde V es una ventana de tamaño $(2v+1)^2$ píxeles y R es una ventana de tamaño $(2r+1)^2$ píxeles.

3.2. Algoritmo en pseudocódigo:

Entrada (imagen (v), grado de filtrado (h), tamaño vecindario (vt), radio de búsqueda (r)).

Salida (imagen_filtrada (u)).

1) Para cada píxel i de la imagen:
   a. Coger vecindario $v(N_i)$ del píxel i de tamaño $((2 * vt + 1)\text{^}2)$
   b. Inicializar $\mathrm{z} = 0$ y $\mathrm{sumw} = 0$
   c. Para cada píxel j dentro del radio de búsqueda de tamaño $((2 * r + 1)\text{^}2)$ (o de la imagen, en caso de no haber radio de búsqueda):
      i. Coger vecindario $v(N_j)$ del píxel j de tamaño $((2 * vt + 1)\text{^}2)$.
      ii. $w = e^{\frac{-||v(N_i)-v(N_j)||_{2,a}^2}{h*h}}$
      iii. $sumw = sumw + w * v(j)$
      iv. $z = z + w$
   d. $u(i) = sumw/z$

2) Devolver imagen_filtrada u.

3.3. Versiones del Non Local Means:

3.3.1. Version1: NLMs con distancia euclídea sin radio de búsqueda para imágenes en escala de gris.

Esta es la versión más básica del filtro Non Local Means. A la hora de filtrar cada píxel de la imagen, se vuelven a recorrer todos y cada uno de los píxeles de la imagen, midiendo la similitud de las vecindades de esos píxeles. Para medir esa similitud, en esta versión se hace uso de la distancia euclídea (sin ponderar). Este valor obtenido, representará el peso que tendrá cada uno de los píxeles de la imagen sobre el píxel que estamos filtrando. De esta forma, el nuevo valor del píxel estudiado, vendrá dado por una suma ponderada de todos los píxeles de la imagen.

En la implementación de esta versión se recibe como parámetros de entrada:

- La imagen en escala de grises a filtrar.
- El grado de filtrado (h).
- El tamaño del vecindario (v). Ventanas de tamaño (2v+1)x(2v+1).
- El radio de búsqueda (r). Para esta versión corresponde a la imagen completa.

Como resultado se obtiene la imagen filtrada y el tiempo de ejecución del algoritmo.

3.3.2. Version2: NLM con distancia euclídea ponderada sin radio de búsqueda para imágenes en escala de gris.

En esta versión, modificamos la forma de calcular la similitud entre dos vecindades. En lugar de emplear la distancia euclídea, vamos a utilizar la distancia euclídea ponderada por un Kernel gaussiano con desviaciónσ. De esta forma, no todos los píxeles de la vecindad tienen la misma importancia. Se dará más peso a aquellos píxeles que sean más cercanos al píxel central de la vecindad. Una vez calculada la distancia euclídea entre las vecindades, a cada píxel de la vecindad resultante, se le ponderará con los valores del Kernel gaussiano.

En la implementación de esta versión se recibe como parámetros de entrada:

- La imagen en escala de grises a filtrar.
- El grado de filtrado (h).
- El tamaño del vecindario (v). Ventanas de tamaño (2v+1)x(2v+1).
- El radio de búsqueda (r). Para esta versión corresponde a la imagen completa.

Como argumentos de salida tenemos la imagen filtrada y el tiempo de ejecución del algoritmo.

3.3.3. Version3: NLM con distancia euclídea y con radio de búsqueda para imágenes en escala de gris.

Es esta versión, en vez de comparar cada píxel con todos los píxeles de la imagen, se comparará solo con aquellos píxeles que estén dentro de un radio de búsqueda, establecido como parámetro. De esta forma estamos suponiendo que las vecindades más similares van a estar relativamente cerca de ese píxel.

De esta forma, también suponemos (y comprobaremos en la parte experimental), que el tiempo de ejecución del algoritmo será menor, ya que para cada píxel no tenemos que estudiar todos los píxeles de la imagen, sino solamente un conjunto más reducido.

En la implementación de esta versión se recibe como parámetros de entrada:

- La imagen en escala de grises a filtrar.
- El grado de filtrado (h).
- El tamaño del vecindario (v). Ventanas de tamaño (2v+1)x(2v+1).
- El radio de búsqueda (r). Ventanas de tamaño (2r+1)x(2r+1).

Como argumentos de salida tenemos la imagen filtrada y el tiempo de ejecución del algoritmo.

3.3.4. Version4: NLM con distancia euclídea ponderada y con radio de búsqueda para imágenes en escala de gris.

Esta versión combina las versiones 2 y 3. Por tanto incorpora la mejora de buscar solamente en un radio cercano al píxel tratado y para calcular la similitud, hace uso de la distancia euclídea ponderada, teniendo más en cuenta los píxeles cercanos de la vecindad.

En la implementación de esta versión se recibe como parámetros de entrada:

- La imagen en escala de grises a filtrar.
- El grado de filtrado (h).
- El tamaño del vecindario (v). Ventanas de tamaño (2v+1)x(2v+1).
- El radio de búsqueda (r). Ventanas de tamaño (2r+1)x(2r+1).

Como argumentos de salida tenemos la imagen filtrada y el tiempo de ejecución del algoritmo.

3.3.5. Versión COLOR: NLM con distancia euclídea ponderada y con radio de búsqueda para imágenes a color (y escala de grises).

Para esta versión, hemos decidido adaptar la versión 4 para poder trabajar también con imágenes a color. Para ello se trabaja sobre cada canal de la imagen a color.

Capítulo 4: Estudio experimental

En esta sección vamos a denominar h al grado de filtrado, v a la vecindad y r al radio de búsqueda.

El objetivo de estas pruebas es observar cómo se comporta cada una de las versiones anteriores y escoger la mejor de ellas. Además, queremos estudiar qué combinación de parámetros es la que mejor se adapta en función de la cantidad de ruido que tenga la imagen.

Esta sección se divide en cuatro partes:

1. En primer lugar, debido a la complejidad del algoritmo ya explicada, el tiempo de ejecución es muy importante. Será preferible escoger un resultado un poco peor que otro, si este tiene un tiempo de ejecución bastante inferior. El tiempo de ejecución, como más adelante se verá viene marcado por el radio de búsqueda, así como por el tamaño de la imagen. En primer lugar se hará una comparativa de tiempos de ejecución entre las versiones 1 y 2 con las versiones 3 y 4. La única diferencia es que en las versiones 1 y 2 el radio de búsqueda corresponde a la imagen completa y en las versiones 3 y 4 es marcado como parámetro. A priori consideramos que las versiones 3 y 4 al no tener que recorrer toda la imagen completa de nuevo para cada píxel (sino parte de ella), serán más rápidas y permitirán trabajar con imágenes más grandes de forma más eficiente y rápida. Las pruebas se realizaran con imágenes en escala de grises.

2. En este punto se realizara un estudio de los parámetros tanto para la V3 como para la V4. El objetivo es obtener una tabla que permita saber cuáles son los parámetros más adecuados dependiendo de la cantidad de ruido que tenga la imagen, para ambas versiones. Se va a trabajar con imágenes en escala de gris. Estas imágenes tendrán ruido gaussiano, por ser el más común, con distinta desviación, de 10 a 100, aumentada de diez en diez. Es decir, imágenes con ruido gaussiano de desviación 10, 20, 30, 40, 50, 60, 70, 80,90 y 100. Una vez obtenidas estas tablas, se aplicarán, a imágenes con ruido rician. Finalmente se compararan los resultados y se escogerá una de las dos versiones para aplicarla a imágenes a color.

   Los tipos de ruido que se van a tratar son el gaussiano por ser el más común, y el rician por tener un comportamiento similar al gaussiano. También porque es común en imágenes de resonancia magnética y me parecía interesante ver si era posible eliminar ruido en ese tipo de imágenes con el algoritmo Non Local Means.
   Para las imágenes obtenidas se obtendrán el PSNR y SSIM. Sin embargo, durante todo el experimento, solo tendremos en cuenta el SSIM, ya que el PSNR daba buenos resultados en imágenes que visualmente eran malas y el SSIM daba los mejores resultados para las imágenes con mejor resultado visual.

3. A continuación se trabajará con la versión a color. En primer lugar se hará un estudio para observar el tiempo de ejecución para distintos radios y tamaños de imágenes. A continuación se realizara el estudio de los parámetros de forma idéntica al punto anterior. Se aplicaran los resultados de la tabla final también a imágenes con ruido rician.

4. Para finalizar se comparará los resultados obtenidos con el filtro NLMs con otros filtros comunes empleados para eliminar ruido en imágenes.

Las diferentes versiones han sido implementadas en Matlab y las pruebas han sido realizadas en un ordenador con las siguientes características:

- Tipo CPU: Intel® Core™ i7-2670qm CPU @ 2.20GHz
- Velocidad reloj: 100MHz
- Capacidad caché L1: 32KB
- Capacidad caché L2: 256KB
- Capacidad RAM: 6GB

4.1. Estudio del tiempo de ejecución de las distintas versiones

A continuación se va a realizar un estudio del tiempo de ejecución de las distintas versiones implementadas del algoritmo. La versiones 1 y 2 recorren la imagen completa por cada píxel de nuevo. Por tanto el tiempo de ejecución de ambas será aproximadamente el mismo. Las versiones 3 y 4 recorren por cada píxel una ventana de tamaño (2r+1)* (2r+1), donde r es el radio de búsqueda establecido como parámetro. Estas 2 versiones también tendrán tiempo de ejecución semejante.

Para las versiones 1 y 2 vamos a realizar las pruebas con fragmentos la imagen Lena de tamaños 2^i, para $4 \leq i \leq 9$. El radio de búsqueda en este caso es la imagen completa. Se obtendrán los tiempos de ejecución. De esta forma veremos cómo varía el tiempo de ejecución en función del tamaño de la imagen.

Para las versiones 3 y 4 se van a utilizar las imágenes de Lena de tamaño 512x512 y de Lena de tamaño 256x256. Se utilizaran radios de búsqueda de tamaño 2, 5, 7, 10, 15 y 17. Se observará como varía el tiempo de ejecución en función del radio y se compararan los resultados obtenidos para estos radios con el resultado obtenido de las versiones 1 y 2. Observaremos si es viable emplear las versiones 1 y 2.

El objetivo es demostrar que es preferible utilizar el radio de búsqueda puesto que reduce considerablemente el tiempo de ejecución permitiendo trabajar con imágenes de tamaños más grandes.

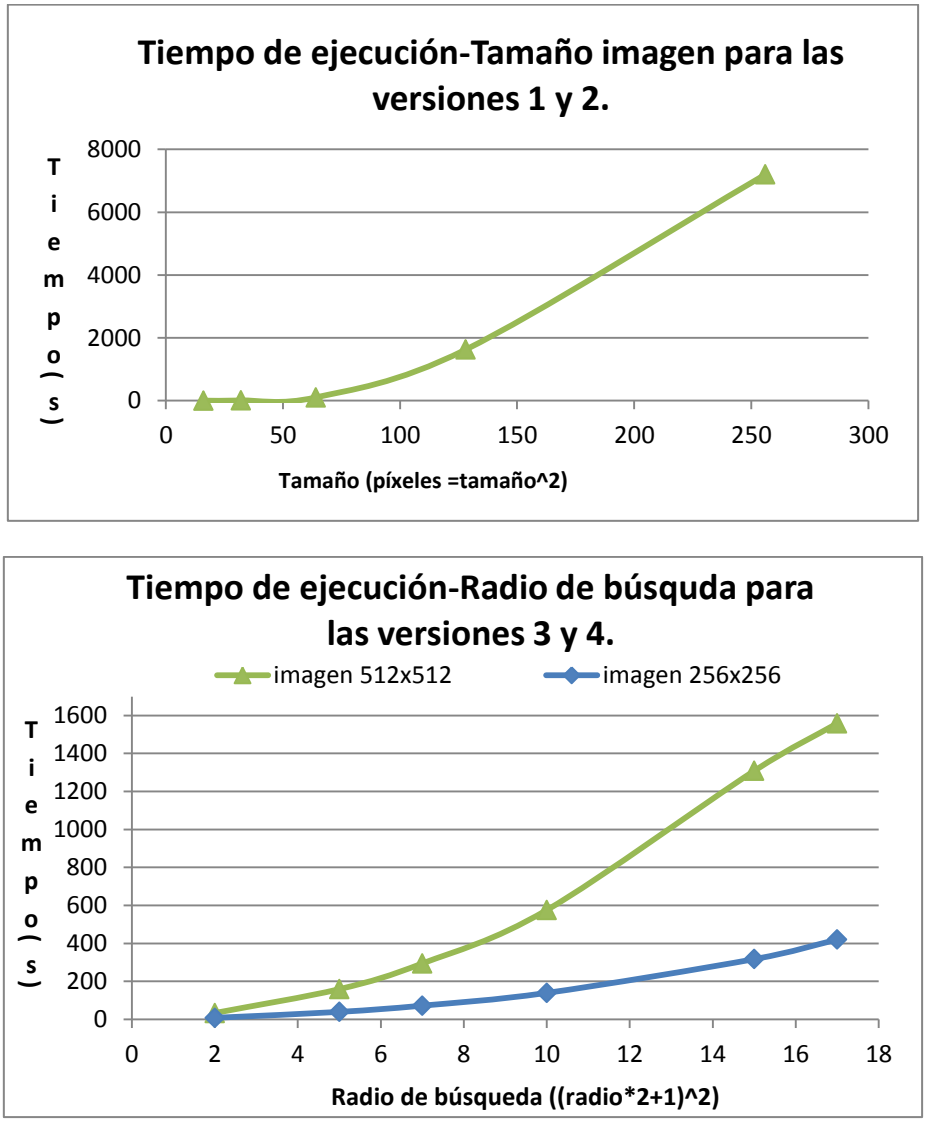
Los resultados obtenidos son los siguientes:

Versiones 1 y 2	
Tamaño imagen	Tiempo(s)
16x16	0,45
32x32	6
64x64	102
128x128	1624
256x256	7200
512x512	Inf.

Versiones 3 y 4		
	Tiempo (s)	
Radio búsqueda	Imagen 512x512	Imagen 256x256
2 (5x5)	34	9
5 (11x11)	160	40
7 (15x15)	295	73
10 (21x21)	576	140
15 (31x31)	1309	318
17 (35x35)	1559	420

TABLA1: Tablas tiempo de ejecución según tamaño de imagen para las versiones 1 y 2 (a la izquierda) y tiempo de ejecución según radio de búsqueda para las versiones 3 y 4 (a la derecha).

A continuación se muestran las gráficas correspondientes a las tablas anteriores:



GRÁFICA 1: Gráficas Tiempo de ejecución-Tamaño imagen para las versiones 1 y 2 (arriba) y tiempo de ejecución-Radio de búsqueda para las versiones 3 y 4 (abajo).

Resulta evidente la diferencia. Mientras que con las versiones 1 y 2 procesar una imagen de 512x512 resulta casi imposible en tiempo de ejecución, con las versiones 3 y 4 en función del radio podemos trabajar con este tamaño sin ningún problema. De la misma forma, para procesar con las v1y2 una imagen de 256x256 necesitamos 3 horas, mientras que con las v3y4 con un radio de búsqueda de 10 (ventana de 21x21) son suficientes unos 3 minutos.

También se muestra un ejemplo de los resultados visuales obtenidos con la versión 1 y la versión 3:



(a) *(b)* *(c)*

La imagen a) corresponde con la imagen de Lena de tamaño 128x128 con ruido gaussiano de desviación 20. La imagen b) corresponde con el resultado de aplicar la versión 1 con h=120, v=3 y SSIM=0,7969. La imagen c) corresponde con el resultado de aplicar la versión 3 con h=120, v=3, r=7 y SSIM=0,9220.

Definitivamente el uso de las versiones 1 y 2 queda descartado para el resto de pruebas.

4.2. Estudio de los parámetros para las versiones 3 y 4. Selección de la mejor versión

En este experimento vamos a comprobar cómo se comportan las versiones 3 y 4 del algoritmo con diferentes combinaciones de parámetros. Para ello, se va a utilizar la imagen de Lena en escala de grises con una resolución de 512x512. Con esta resolución nos es suficiente para observar bien los detalles.

La imagen de Lena es la siguiente:

El objetivo de esta prueba es observar cómo influye cada uno de los parámetros en el filtrado de la imagen y obtener una tabla con la combinación más adecuada dependiendo de la cantidad de ruido que tenga la imagen.

En primer lugar vamos a aplicarle ruido gaussiano con distinta desviación, de 10 a 100, aumentado de diez en diez, a la imagen de Lena. Las imágenes con ruido con las que trataremos las llamaremos input_*cantidadRuido*R.png. Obtendremos su PSNR y su SSIM que nos servirán de referencia para ver cuánto mejora tras aplicar el filtrado. Para añadir ruido realizamos lo siguiente (ejemplo para ruido gaussiano con desviación 10):

input_10R=imgOriginal+randn(f,c)*10 (ruido)

A continuación se muestra la tabla de imágenes de Lena con ruido gaussiano con las que trataremos:

Imagen con ruido	PSNR(dB)	SSIM
input_10R.png	28,11	0,8708
input_20R.png	22,14	0,6788
input_30R.png	18,69	0,5352
input_40R.png	16,35	0,4361
input_50R.png	14,61	0,3651
input_60R.png	13,29	0,3142
input_70R.png	12,23	0,2724
input_80R.png	11,40	0,2412
input_90R.png	10,73	0,2159
input_100R.png	10,13	0,1913

TABLA 2: PSNR y SSIM de las imágenes de Lena con ruido gaussiano añadido de desviación 10, 20, 30, 40, 50, 60, 70, 80, 90 y 100.

El estudio de los parámetros consistirá en lo siguiente:

Para cada una de las 10 imágenes seleccionaremos distintos valores para el grado de filtrado (parámetro h). También utilizaremos vecindades de 3x3,5x5,7x7,9x9 (v=1,2,3,4) y radios de 5x5,11x11,15x15,21x21 y en algunos casos de 31x31 (r=2,5,7,10,15). Aplicaremos todas las posibles combinaciones de valores de parámetros y los usaremos para filtrar las imágenes con ruido.

A las imágenes obtenidas tras el filtrado las llamaremos de la siguiente forma: input_cantidadRuido-h-v-r.png. Por ejemplo la imagen input_20R-20-2-7.png corresponde al filtrado de la imagen de Lena con ruido gaussiano de desviación 20, aplicando un h=20, una vecindad de 5x5 (v=2) y un radio de 15x15 (r=7).

Para cada imagen obtendremos su PSNR y su SSIM.

Tras estudiar las mejores combinaciones para cada cantidad de ruido presente en la imagen, aplicaremos estos resultados a diferentes imágenes con ruido gaussiano y rician.

Finalmente se escogerá una de las 2 versiones con la que seguir trabajando para imágenes a color.

4.2.1. Estudio de la versión 3: distancia euclídea

4.2.1.1. Estudio de parámetros para imágenes en escala de gris con ruido gaussiano

Resultados:

RUIDO 10:

Al ser las primeras pruebas se emplearán más valores para el grado de filtrado (h). Estos valores van a ser 25, 50, 75,100.

IMAGEN	PSNR (dB)	SSIM
input_10R-25-1-2	30,97	0,9267
input_10R-25-2-2	28,18	0,8727
input_10R-25-3-2	28,11	0,8708
input_10R-25-4-2	28,11	0,8708
input_10R-25-1-5	32,73	0,9506
input_10R-25-2-5	28,41	0,8803
input_10R-25-3-5	28,11	0,8708
input_10R-25-4-5	28,11	0,8708
input_10R-25-1-7	33,16	0,9525
input_10R-25-2-7	28,57	0,8854
input_10R-25-3-7	28,11	0,8708
input_10R-25-4-7	28,11	0,8708
input_10R-25-1-10	33,46	0,9522
input_10R-25-2-10	28,78	0,8922
input_10R-25-3-10	28,11	0,8708
input_10R-25-4-10	28,11	0,8708

IMAGEN	PSNR (dB)	SSIM
input_10R-50-1-2	34,74	0,9559
input_10R-50-2-2	32,85	0,9499
input_10R-50-3-2	29,51	0,9059
input_10R-50-4-2	28,22	0,8740
input_10R-50-1-5	34,35	0,9512
input_10R-50-2-5	34,21	0,9615
input_10R-50-3-5	31,06	0,9440
input_10R-50-4-5	28,55	0,8860
input_10R-50-1-7	34,01	0,9458
input_10R-50-2-7	34,33	0,9592
input_10R-50-3-7	31,60	0,9528
input_10R-50-4-7	28,77	0,8936
input_10R-50-1-10	33,59	0,9408
input_10R-50-2-10	34,32	0,9561
input_10R-50-3-10	34,32	0,9561
input_10R-50-4-10	29,04	0,9033

IMAGEN	PSNR (dB)	SSIM
input_10R-75-1-2	34,01	0,9506
input_10R-75-2-2	34,82	0,9570
input_10R-75-3-2	33,57	0,9558
input_10R-75-4-2	31,02	0,9336
input_10R-75-1-5	32,82	0,9326
input_10R-75-2-5	34,31	0,9478
input_10R-75-3-5	34,36	0,9590
input_10R-75-4-5	32,61	0,9607
input_10R-75-1-7	32,33	0,9226
input_10R-75-2-7	33,87	0,9406
input_10R-75-3-7	34,28	0,9541
input_10R-75-4-7	32,91	0,9611
input_10R-75-1-10	31,81	0,9142
input_10R-75-2-10	33,34	0,9337
input_10R-75-3-10	34,10	0,9487
input_10R-75-4-10	33,08	0,9587

IMAGEN	PSNR (dB)	SSIM
input_10R-100-1-2	33,14	0,9450
input_10R-100-2-2	34,33	0,9523
input_10R-100-3-2	34,84	0,9575
input_10R-100-4-2	33,84	0,9573
input_10R-100-1-5	31,61	0,9153
input_10R-100-2-5	32,99	0,9322
input_10R-100-3-5	34,28	0,9462
input_10R-100-4-5	34,34	0,9566
input_10R-100-1-7	31,06	0,9012
input_10R-100-2-7	32,38	0,9211
input_10R-100-3-7	33,80	0,9372
input_10R-100-4-7	34,17	0,9500
input_10R-100-1-10	30,51	0,8901
input_10R-100-2-10	31,76	0,9113
input_10R-100-3-10	33,24	0,9284
input_10R-100-4-10	33,89	0,9429

TABLA 3: Tablas del PSNR y SSIM de las imágenes resultantes del filtrado, con la versión 3, de la imagen de Lena con ruido gaussiano con desviación 10 para cada combinación de parámetros con h=25 (arriba a la izquierda), h=50 (arriba a la derecha), h=75(abajo a la izquierda) y h=100 (abajo a la derecha).

RUIDO 20:

A partir del experimento con ruido de desviación 10 hemos comprobado que los mejores resultados vienen dados cuando h=5*ruido más o menos. Por tanto, a partir de ahora vamos a reducir los experimentos y utilizar valores para el grado de filtrado alrededor de h=5*ruido, algo superiores y algo inferiores.

IMAGEN	PSNR (dB)	SSIM
input_20R-80-1-2	30,43	0,8859
input_20R-80-2-2	25,73	0,8038
input_20R-80-3-2	22,55	0,6957
input_20R-80-4-2	22,14	0,6789
input_20R-80-1-5	31,31	0,9139
input_20R-80-2-5	28,90	0,9025
input_20R-80-3-5	23,44	0,7416
input_20R-80-4-5	22,16	0,6801
input_20R-80-1-7	31,10	0,9082
input_20R-80-2-7	29,73	0,9142
input_20R-80-3-7	24,01	0,7707
input_20R-80-4-7	22,19	0,6816
input_20R-80-1-10	30,76	0,9001
input_20R-80-2-10	30,30	0,9166
input_20R-80-3-10	24,72	0,8060
input_20R-80-4-10	22,24	0,6842

IMAGEN	PSNR (dB)	SSIM
input_20R-100-1-2	31,09	0,8929
input_20R-100-2-2	28,86	0,8697
input_20R-100-3-2	24,30	0,7618
input_20R-100-4-2	22,39	0,6894
input_20R-100-1-5	31,03	0,9069
input_20R-100-2-5	31,27	0,9208
input_20R-100-3-5	26,94	0,8731
input_20R-100-4-5	23,00	0,7218
input_20R-100-1-7	30,63	0,8975
input_20R-100-2-7	31,40	0,9176
input_20R-100-3-7	28,86	0,8697
input_20R-100-4-7	23,43	0,7449
input_20R-100-1-10	30,16	0,8873
input_20R-100-2-10	31,23	0,9105
input_20R-100-3-10	28,70	0,9138
input_20R-100-4-10	23,99	0,7761

IMAGEN	PSNR (dB)	SSIM
input_20R-120-1-2	31,10	0,8932
input_20R-120-2-2	30,63	0,8909
input_20R-120-3-2	26,97	0,8373
input_20R-120-4-2	23,49	0,7331
input_20R-120-1-5	30,50	0,8968
input_20R-120-2-5	31,52	0,9148
input_20R-120-3-5	29,98	0,9198
input_20R-120-4-5	25,53	0,8338
input_20R-120-1-7	30,01	0,8842
input_20R-120-2-7	31,20	0,9065
input_20R-120-3-7	30,53	0,9220
input_20R-120-4-7	26,44	0,8717
input_20R-120-1-10	29,47	0,8718
input_20R-120-2-10	30,71	0,8965
input_20R-120-3-10	30,78	0,9168
input_20R-120-4-10	27,28	0,8982

TABLA 4: Tablas del PSNR y SSIM de las imágenes resultantes del filtrado, con la versión 3, de la imagen de Lena con ruido gaussiano con desviación 20 para cada combinación de parámetros con h=80 (arriba a la izquierda), h=100 (arriba a la derecha) y h=120 (abajo).

RUIDO 30:

Los valores elegidos de h son 120, 150,180.

IMAGEN	PSNR (dB)	SSIM
input_30R-120-1-2	27,93	0,8066

IMAGEN	PSNR (dB)	SSIM
input_30R-150-1-2	28,75	0,8198

input_30R-120-2-2	22,89	0,6919
input_30R-120-3-2	19,22	0,5573
input_30R-120-4-2	18,70	0,5355
input_30R-120-1-5	29,20	0,8687
input_30R-120-2-5	26,94	0,8421
input_30R-120-3-5	20,40	0,6173
input_30R-120-4-5	18,75	0,5379
input_30R-120-1-7	29,01	0,8638
input_30R-120-2-7	27,99	0,8642
input_30R-120-3-7	21,16	0,6575
input_30R-120-4-7	18,80	0,5403
input_30R-120-1-10	28,67	0,8544
input_30R-120-2-10	28,65	0,8707
input_30R-120-3-10	22,13	0,7087
input_30R-120-4-10	18,88	0,5440

input_30R-150-2-2	26,38	0,7814
input_30R-150-3-2	21,31	0,6391
input_30R-150-4-2	19,04	0,5496
input_30R-150-1-5	29,06	0,8645
input_30R-150-2-5	29,35	0,8760
input_30R-150-3-5	24,78	0,7987
input_30R-150-4-5	19,87	0,5923
input_30R-150-1-7	28,64	0,8543
input_30R-150-2-7	29,46	0,8757
input_30R-150-3-7	26,38	0,7814
input_30R-150-4-7	20,45	0,6242
input_30R-150-1-10	28,11	0,8412
input_30R-150-2-10	29,26	0,8689
input_30R-150-3-10	27,28	0,8680
input_30R-150-4-10	21,25	0,6694

IMAGEN	PSNR (dB)	SSIM
input_30R-180-1-2	28,90	0,8228
input_30R-180-2-2	28,22	0,8135
input_30R-180-3-2	24,41	0,7370
input_30R-180-4-2	20,39	0,6038
input_30R-180-1-5	28,61	0,8546
input_30R-180-2-5	29,46	0,8726
input_30R-180-3-5	28,40	0,8724
input_30R-180-4-5	23,09	0,7438
input_30R-180-1-7	28,05	0,8395
input_30R-180-2-7	29,16	0,8653
input_30R-180-3-7	29,06	0,8798
input_30R-180-4-7	24,37	0,8028
input_30R-180-1-10	27,37	0,8221
input_30R-180-2-10	28,72	0,8549
input_30R-180-3-10	29,23	0,8747
input_30R-180-4-10	25,62	0,8468

TABLA 5: Tablas del PSNR y SSIM de las imágenes resultantes del filtrado, con la versión 3, de la imagen de Lena con ruido gaussiano con desviación 30 para cada combinación de parámetros con h=120 (arriba a la izquierda), h=150 (arriba a la derecha) y h=180 (abajo).

RUIDO 40:

Los valores elegidos de h son 170,200 y 230.

IMAGEN	PSNR (dB)	SSIM
input_40R-170-1-2	26,41	0,7379
input_40R-170-2-2	22,13	0,6390
input_40R-170-3-2	17,46	0,4792
input_40R-170-4-2	16,40	0,4384

IMAGEN	PSNR (dB)	SSIM
input_40R-200-1-2	26,97	0,7482
input_40R-200-2-2	24,67	0,7042
input_40R-200-3-2	19,45	0,5528
input_40R-200-4-2	16,82	0,4543

input_40R-170-1-5	27,69	0,8264
input_40R-170-2-5	26,55	0,8083
input_40R-170-3-5	19,65	0,5849
input_40R-170-4-5	16,61	0,4474
input_40R-170-1-7	27,39	0,8211
input_40R-170-2-7	27,34	0,8279
input_40R-170-3-7	20,91	0,6461
input_40R-170-4-7	16,78	0,4548
input_40R-170-1-10	26,91	0,8074
input_40R-170-2-10	27,61	0,8309
input_40R-170-3-10	22,38	0,7122
input_40R-170-4-10	17,04	0,4670

input_40R-200-1-5	27,60	0,8246
input_40R-200-2-5	27,83	0,8322
input_40R-200-3-5	23,59	0,7395
input_40R-200-4-5	17,92	0,5068
input_40R-200-1-7	27,12	0,8144
input_40R-200-2-7	27,91	0,8359
input_40R-200-3-7	25,18	0,7942
input_40R-200-4-7	18,69	0,5456
input_40R-200-1-10	26,46	0,7962
input_40R-200-2-10	27,68	0,8295
input_40R-200-3-10	26,43	0,8240
input_40R-200-4-10	19,73	0,6005

IMAGEN	PSNR (dB)	SSIM
input_40R-230-1-2	27,18	0,7523
input_40R-230-2-2	26,15	0,7346
input_40R-230-3-2	21,97	0,6346
input_40R-230-4-2	17,90	0,4956
input_40R-230-1-5	27,35	0,8187
input_40R-230-2-5	27,98	0,8335
input_40R-230-3-5	26,60	0,8160
input_40R-230-4-5	20,68	0,6332
input_40R-230-1-7	26,74	0,8039
input_40R-230-2-7	27,74	0,8304
input_40R-230-3-7	27,48	0,8361
input_40R-230-4-7	22,16	0,7041
input_40R-230-1-10	25,94	0,7810
input_40R-230-2-10	27,32	0,8198
input_40R-230-3-10	27,74	0,8364
input_40R-230-4-10	23,73	0,7697

TABLA 6: Tablas del PSNR y SSIM de las imágenes resultantes del filtrado, con la versión 3, de la imagen de Lena con ruido gaussiano con desviación 40 para cada combinación de parámetros con h=170 (arriba a la izquierda), h=200 (arriba a la derecha) y h=230 (abajo).

RUIDO 50:

Se va a introducir un radio de búsqueda de 15 (ventana de 31x31) en algunos momentos ya que el ruido empieza a ser considerable y radios pequeños empiezan a funcionar peor.

Los valores elegidos de h son 225,250 y 275.

IMAGEN	PSNR (dB)	SSIM
input_50R-225-1-2	25,29	0,6785
input_50R-225-2-2	21,91	0,6010
input_50R-225-3-2	16,64	0,4376
input_50R-225-4-2	14,82	0,3728

IMAGEN	PSNR (dB)	SSIM
input_50R-250-1-2	25,62	0,6853
input_50R-250-2-2	23,45	0,6393
input_50R-250-3-2	18,22	0,4913
input_50R-250-4-2	15,24	0,3878

input_50R-225-1-5	26,53	0,7871
input_50R-225-2-5	26,11	0,7759
input_50R-225-3-5	20,09	0,5914
input_50R-225-4-5	15,38	0,3966
input_50R-225-1-7	26,14	0,7818
input_50R-225-2-7	26,54	0,7938
input_50R-225-3-7	21,85	0,6637
input_50R-225-4-7	15,81	0,4156
input_50R-225-1-10	25,48	0,7618
input_50R-225-2-10	26,51	0,7933
input_50R-225-3-10	23,62	0,7264
input_50R-225-4-10	16,45	0,4451
input_50R-230-3-15	25,48	0,7780

input_50R-250-1-5	26,49	0,7871
input_50R-250-2-5	26,65	0,7902
input_50R-250-3-5	22,93	0,6921
input_50R-250-4-5	16,67	0,4519
input_50R-250-1-7	25,97	0,7781
input_50R-250-2-7	26,70	0,7988
input_50R-250-3-7	23,45	0,6393
input_50R-250-4-7	17,66	0,4974
input_50R-250-1-10	25,19	0,7540
input_50R-250-2-10	26,42	0,7921
input_50R-250-3-10	25,81	0,7838
input_50R-250-4-10	18,99	0,5586
input_50R-250-3-15	26,28	0,7945

IMAGEN	PSNR (dB)	SSIM
input_50R-275-1-2	25,80	0,6893
input_50R-275-2-2	24,47	0,6626
input_50R-275-3-2	19,99	0,5468
input_50R-275-4-2	16,01	0,4152
input_50R-275-1-5	26,37	0,7850
input_50R-275-2-5	26,79	0,7945
input_50R-275-3-5	25,05	0,7536
input_50R-275-4-5	18,71	0,5382
input_50R-275-1-7	25,76	0,7726
input_50R-275-2-7	26,62	0,7974
input_50R-275-3-7	26,13	0,7890
input_50R-275-4-7	20,29	0,6090
input_50R-275-1-10	24,89	0,7447
input_50R-275-2-10	26,20	0,7866
input_50R-275-3-10	26,52	0,7981
input_50R-275-4-10	22,10	0,6839

TABLA 7: Tablas del PSNR y SSIM de las imágenes resultantes del filtrado, con la versión 3, de la imagen de Lena con ruido gaussiano con desviación 50 para cada combinación de parámetros con h=225 (arriba a la izquierda), h=250 (arriba a la derecha) y h=275 (abajo).

RUIDO 60:

Los valores elegidos de h son 275, 300 y 330.

IMAGEN	PSNR (dB)	SSIM
input_60R-275-1-2	24,21	0,6244
input_60R-275-2-2	21,42	0,5631
input_60R-275-3-2	16,05	0,4075
input_60R-275-4-2	13,68	0,3276

IMAGEN	PSNR (dB)	SSIM
input_60R-300-1-2	24,45	0,6294
input_60R-300-2-2	22,56	0,5903
input_60R-300-3-2	17,52	0,4538
input_60R-300-4-2	14,16	0,3440

input_60R-275-1-5	25,46	0,7470
input_60R-275-2-5	25,24	0,7380
input_60R-275-3-5	20,35	0,5835
input_60R-275-4-5	14,62	0,3665
input_60R-275-1-7	25,04	0,7429
input_60R-275-2-7	25,51	0,7561
input_60R-275-3-7	22,28	0,6551
input_60R-275-4-7	15,32	0,3961
input_60R-275-1-10	24,32	0,7195
input_60R-275-2-10	25,35	0,7544
input_60R-275-3-10	23,89	0,7093
input_60R-275-4-10	16,34	0,4395

input_60R-300-1-5	25,43	0,7478
input_60R-300-2-5	25,56	0,7491
input_60R-300-3-5	22,63	0,6598
input_60R-300-4-5	16,05	0,4235
input_60R-300-1-7	24,92	0,7409
input_60R-300-2-7	25,57	0,7603
input_60R-300-3-7	24,21	0,7171
input_60R-300-4-7	17,32	0,4767
input_60R-300-1-10	24,12	0,7142
input_60R-300-2-10	25,24	0,7532
input_60R-300-3-10	25,08	0,7467
input_60R-300-4-10	18,99	0,5445
input_60R-300-3-15	25,23	0,7550

IMAGEN	PSNR (dB)	SSIM
input_60R-330-1-2	24,63	0,6331
input_60R-330-2-2	23,47	0,6108
input_60R-330-3-2	19,35	0,5072
input_60R-330-4-2	15,09	0,3753
input_60R-330-1-5	25,36	0,7468
input_60R-330-2-5	25,68	0,7541
input_60R-330-3-5	24,41	0,7152
input_60R-330-4-5	18,42	0,5153
input_60R-330-1-7	24,76	0,7367
input_60R-330-2-7	25,50	0,7596
input_60R-330-3-7	25,27	0,7504
input_60R-330-4-7	20,30	0,5896
input_60R-330-1-10	23,88	0,7064
input_60R-330-2-10	25,02	0,7478
input_60R-330-3-10	25,44	0,7594
input_60R-330-4-10	22,29	0,6633
input_60R-330-3-15	25,14	0,7552

TABLA 8: Tablas del PSNR y SSIM de las imágenes resultantes del filtrado, con la versión 3, de la imagen de Lena con ruido gaussiano con desviación 60 para cada combinación de parámetros con h=275 (arriba a la izquierda), h=300 (arriba a la derecha) y h=330 (abajo).

RUIDO 70:

Los valores elegidos de h son 320, 350, 380.

IMAGEN	PSNR (dB)	SSIM
input_70R-320-1-2	23,27	0,5754
input_70R-320-2-2	20,87	0,5226
input_70R-320-3-2	15,53	0,3759

IMAGEN	PSNR (dB)	SSIM
input_70R-350-1-2	23,49	0,5802
input_70R-350-2-2	21,91	0,5473
input_70R-350-3-2	17,17	0,4236

input_70R-320-4-2	12,77	0,2898
input_70R-320-1-5	24,65	0,7154
input_70R-320-2-5	24,48	0,7054
input_70R-320-3-5	20,34	0,5658
input_70R-320-4-5	14,06	0,3394
input_70R-320-1-7	24,22	0,7148
input_70R-320-2-7	24,67	0,7273
input_70R-320-3-7	22,25	0,6391
input_70R-320-4-7	15,00	0,3766
input_70R-320-1-10	23,45	0,6882
input_70R-320-2-10	24,41	0,7236
input_70R-320-3-10	23,61	0,6890
input_70R-320-4-10	16,34	0,4293
input_70R-330-3-15	24,28	0,7205

input_70R-350-4-2	13,40	0,3097
input_70R-350-1-5	24,64	0,7173
input_70R-350-2-5	24,74	0,7167
input_70R-350-3-5	22,48	0,6388
input_70R-350-4-5	15,87	0,4070
input_70R-350-1-7	24,13	0,7139
input_70R-350-2-7	24,70	0,7319
input_70R-350-3-7	23,83	0,6956
input_70R-350-4-7	17,46	0,4697
input_70R-350-1-10	23,29	0,6843
input_70R-350-2-10	24,27	0,7224
input_70R-350-3-10	24,39	0,7208
input_70R-350-4-10	19,44	0,5448
input_70R-350-3-15	24,31	0,7258

IMAGEN	PSNR (dB)	SSIM
input_70R-380-1-2	23,63	0,5833
input_70R-380-2-2	22,59	0,5630
input_70R-380-3-2	18,77	0,4676
input_70R-380-4-2	14,34	0,3390
input_70R-380-1-5	24,60	0,7174
input_70R-380-2-5	24,83	0,7218
input_70R-380-3-5	23,76	0,6826
input_70R-380-4-5	18,15	0,4904
input_70R-380-1-7	24,03	0,7117
input_70R-380-2-7	24,64	0,7321
input_70R-380-3-7	24,48	0,7217
input_70R-380-4-7	20,18	0,5691
input_70R-380-1-10	23,13	0,6792
input_70R-380-2-10	24,09	0,7180
input_60R-330-3-10	25,44	0,7594
input_70R-380-4-10	22,19	0,6429
input_70R-380-4-15	23,61	0,6975

TABLA 9: Tablas del PSNR y SSIM de las imágenes resultantes del filtrado, con la versión 3, de la imagen de Lena con ruido gaussiano con desviación 70 para cada combinación de parámetros con h=320 (arriba a la izquierda), h=350 (arriba a la derecha) y h=380 (abajo).

RUIDO 80:

Los valores elegidos de h son 370, 400, 430. A partir de ahora las pruebas para radios de búsqueda de 2 y 5 los eliminamos puesto que vemos que no están dando resultados buenos.

IMAGEN	PSNR (dB)	SSIM
input_80R-370-1-7	23,36	0,6859
input_80R-370-2-7	23,74	0,6980
input_80R-370-3-7	22,45	0,6376

IMAGEN	PSNR (dB)	SSIM
input_80R-400-1-7	23,31	0,6861
input_80R-400-2-7	23,73	0,7012
input_80R-400-3-7	23,29	0,6749

input_80R-370-4-7	15,64	0,3886
input_80R-370-1-10	22,61	0,6591
input_80R-370-2-10	23,37	0,6912
input_80R-370-3-10	23,21	0,6746
input_80R-370-4-10	17,49	0,4556
input_80R-370-3-15	23,25	0,6855

input_80R-400-4-7	18,15	0,4785
input_80R-400-1-10	22,50	0,6571
input_80R-400-2-10	23,25	0,6901
input_80R-400-3-10	23,50	0,6931
input_80R-400-4-10	20,28	0,5567
input_80R-400-3-15	23,18	0,6902
input_80R-400-4-15	22,07	0,6284

IMAGEN	PSNR (dB)	SSIM
input_80R-430-1-7	23,24	0,6851
input_80R-430-2-7	23,68	0,7017
input_80R-430-3-7	23,62	0,6930
input_80R-430-4-7	20,48	0,5630
input_80R-430-1-10	22,40	0,6542
input_80R-430-2-10	23,12	0,6868
input_80R-430-3-10	23,51	0,6989
input_80R-430-4-10	22,18	0,6313
input_80R-430-3-15	23,01	0,6880
input_80R-430-4-15	23,00	0,6734

TABLA 10: Tablas del PSNR y SSIM de las imágenes resultantes del filtrado, con la versión 3, de la imagen de Lena con ruido gaussiano con desviación 80 para cada combinación de parámetros con h=370 (arriba a la izquierda), h=400 (arriba a la derecha) y h=430 (abajo).

RUIDO 90:

Los valores elegidos de h son 410, 450, 480.

IMAGEN	PSNR (dB)	SSIM
input_90R-410-1-7	22,79	0,6699
input_90R-410-2-7	23,09	0,6788
input_90R-410-3-7	22,16	0,6244
input_90R-410-4-7	15,95	0,3851
input_90R-410-1-10	22,06	0,6418
input_90R-410-2-10	22,69	0,6692
input_90R-410-3-10	22,67	0,6564
input_90R-410-4-10	18,02	0,4584
input_90R-410-3-15	22,54	0,6620

IMAGEN	PSNR (dB)	SSIM
input_90R-450-1-7	22,73	0,6709
input_90R-450-2-7	23,08	0,6831
input_90R-450-3-7	22,85	0,6626
input_90R-450-4-7	19,04	0,4959
input_90R-450-1-10	21,96	0,6406
input_90R-450-2-10	22,56	0,6683
input_90R-450-3-10	22,85	0,6740
input_90R-450-4-10	20,95	0,5729
input_90R-450-3-15	22,40	0,6649

IMAGEN	PSNR (dB)	SSIM
input_90R-480-1-7	22,69	0,6705
input_90R-480-2-7	23,03	0,6838
input_90R-480-3-7	23,02	0,6760
input_90R-480-4-7	20,82	0,5669
input_90R-480-1-10	21,88	0,6388

input_90R-480-2-10	22,46	0,6659
input_90R-480-3-10	22,81	0,6776
input_90R-480-4-10	22,10	0,6279
input_90R-480-4-15	22,44	0,6573

TABLA 11: Tablas del PSNR y SSIM de las imágenes resultantes del filtrado, con la versión 3, de la imagen de Lena con ruido gaussiano con desviación 90 para cada combinación de parámetros con h=410 (arriba a la izquierda), h=450 (arriba a la derecha) y h=480 (abajo).

RUIDO 100:

Los valores elegidos de h son 470, 500, 540.

IMAGEN	PSNR (dB)	SSIM
input_100R-470-1-7	22,01	0,6452
input_100R-470-2-7	22,23	0,6521
input_100R-470-3-7	21,89	0,6230
input_100R-470-4-7	17,82	0,4342
input_100R-470-1-10	21,33	0,6185
input_100R-470-2-10	22,69	0,6396
input_100R-470-3-10	21,92	0,6387
input_100R-470-4-10	19,78	0,5120
input_100R-470-4-15	21,01	0,5777

IMAGEN	PSNR (dB)	SSIM
input_100R-500-1-7	21,98	0,6463
input_100R-500-2-7	22,21	0,6545
input_100R-500-3-7	22,10	0,6401
input_100R-500-4-7	19,60	0,5059
input_100R-500-1-10	21,28	0,6183
input_100R-500-2-10	21,69	0,6392
input_100R-500-3-10	21,93	0,6456
input_100R-500-4-10	20,99	0,5738
input_100R-500-3-15	21,38	0,6305
input_100R-500-4-15	21,44	0,6131

IMAGEN	PSNR (dB)	SSIM
input_100R-540-1-7	21,94	0,6466
input_100R-540-2-7	22,17	0,6555
input_100R-540-3-7	22,20	0,6517
input_100R-540-4-7	21,08	0,5775
input_100R-540-1-10	21,22	0,6173
input_100R-540-2-10	21,59	0,6371
input_100R-540-3-10	21,87	0,6482
input_100R-540-4-10	21,67	0,6205

TABLA 12: Tablas del PSNR y SSIM de las imágenes resultantes del filtrado, con la versión 3, de la imagen de Lena con ruido gaussiano con desviación 100 para cada combinación de parámetros con h=470 (arriba a la izquierda), h=500 (arriba a la derecha) y h=540 (abajo).

Análisis de los resultados:

A continuación se muestra la tabla resumen con las mejores combinaciones de parámetros para las diferentes cantidades de ruido que hemos estudiado anteriormente:

	h	vecindad	radio
0<ruido<=15	5*ruido	2	5

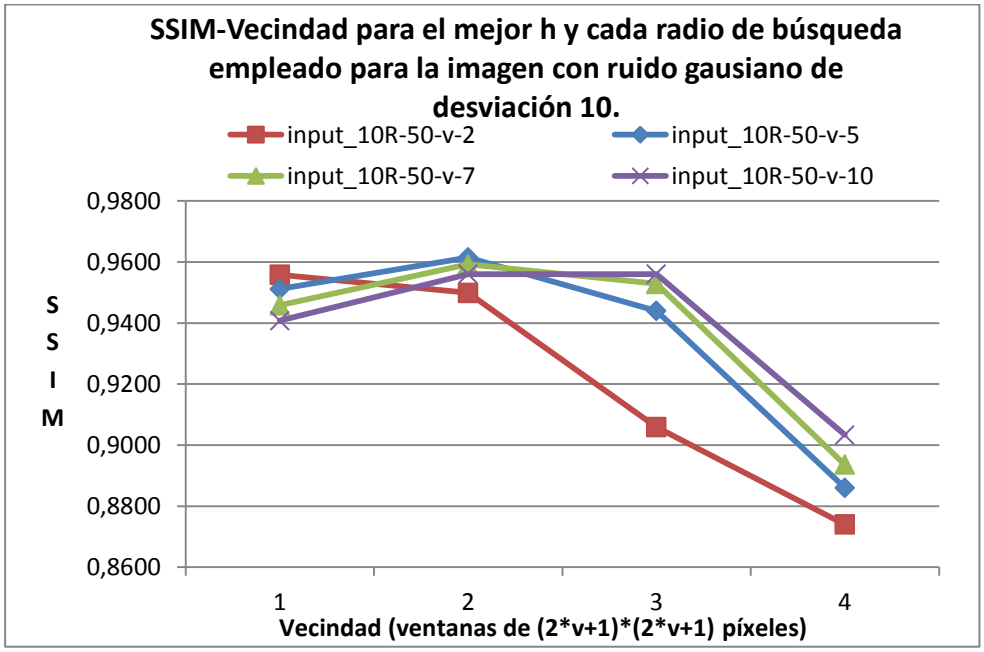
15<ruido<=30	6*ruido	3	7
30<ruido<=45	5,75*ruido	3	10
45<ruido<=75	5,5*ruido	3	10
75<ruido<=100	5,35*ruido	3	10

TABLA 12: Tabla resumen de parámetros en función del ruido para la versión 3.

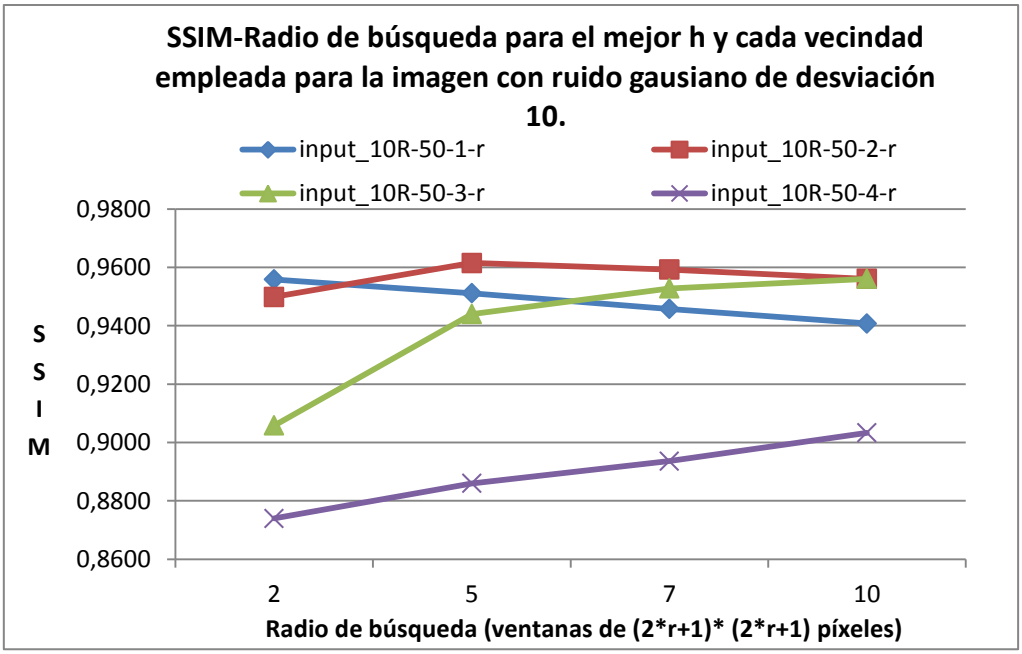
ruido	h	vecindad	radio
10	50	2	5
20	120	3	7
30	180	3	7
40	230	3	10
50	275	3	10
60	330	3	10
70	385	3	10
80	428	3	10
90	481,5	3	10
100	535	3	10

TABLA 13: Tabla de parámetros exactos para ruidos con desviación 10, 20, 30, 40, 50, 60, 70, 80, 90 y 100 para la versión 3.

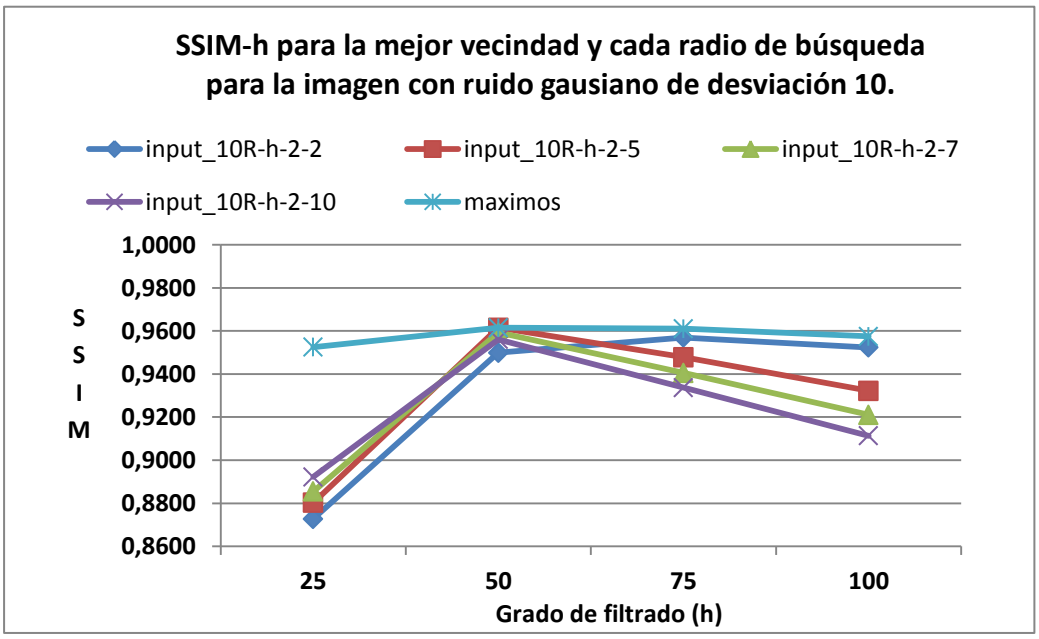
Para clarificar la cantidad de resultados obtenidos hasta ahora, mostramos en diferentes gráficas cómo evoluciona la calidad de las imágenes obtenidas según variamos cada uno de los parámetros de estudio.



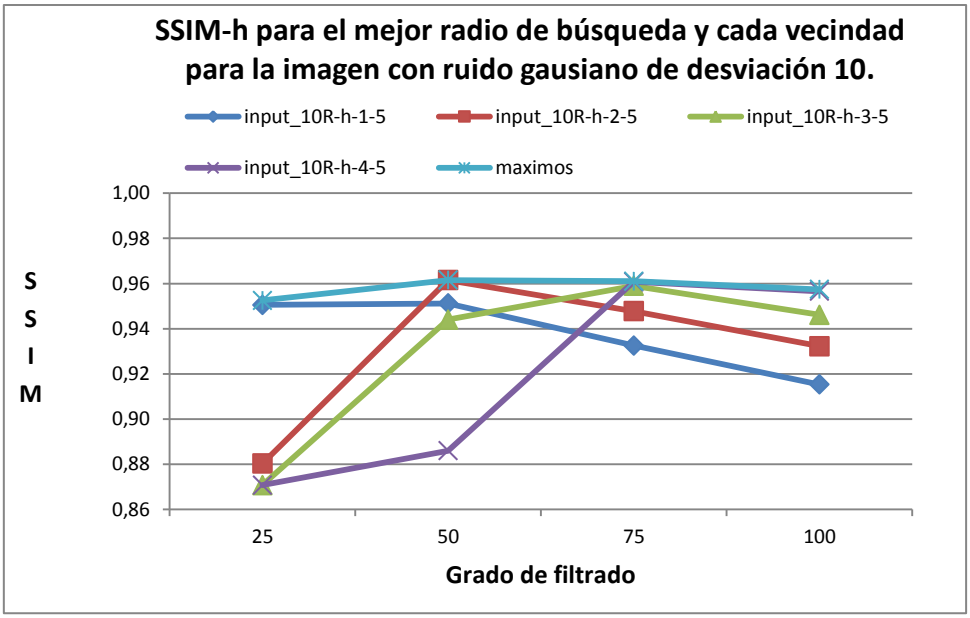
(a)


SSIM-Radio de búsqueda para el mejor h y cada vecindad empleada para la imagen con ruido gausiano de desviación 10.
input_10R-50-1-r
input_10R-50-2-r
input_10R-50-3-r
input_10R-50-4-r
SSIM
0,9800
0,9600
0,9400
0,9200
0,9000
0,8800
0,8600
2
5
7
10
Radio de búsqueda (ventanas de (2*r+1)* (2*r+1) píxeles)

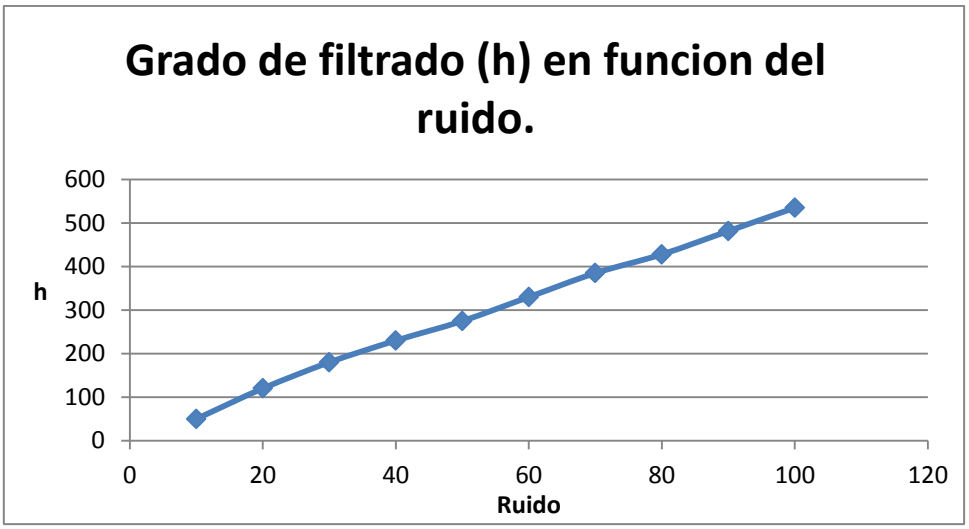
(b)


SSIM-h para la mejor vecindad y cada radio de búsqueda para la imagen con ruido gausiano de desviación 10.
input_10R-h-2-2
input_10R-h-2-5
input_10R-h-2-7
input_10R-h-2-10
maximos
SSIM
1,0000
0,9800
0,9600
0,9400
0,9200
0,9000
0,8800
0,8600
25
50
75
100
Grado de filtrado (h)

(c)



(d)



(e)

GRÁFICA 2: Gráficas para la versión 3. ***a)****SSIM-Vecindad para el mejor h y cada radio de búsqueda empleado para la imagen con ruido gaussiano de desviación 10.* ***b)****SSIM-Radio de búsqueda para el mejor h y cada vecindad empleada para la imagen con ruido gaussiano de desviación 10.* ***c)****SSIM-h para la mejor vecindad y cada radio de búsqueda para la imagen con ruido gaussiano de desviación 10.* ***d)****SSIM-h para el mejor radio de búsqueda y cada vecindad para la imagen con ruido gaussiano de desviación 10.* ***e)****Grado de filtrado (h) en función del ruido.*

Además, comprobamos visualmente los resultados obtenidos con estos parámetros.

La siguiente sucesión de imágenes es un ejemplo para la gráfica a), tomando como imagen la imagen de Lena con ruido gaussiano de desviación 50:

Imagen con ruido gaussiano desviación 50. SSIM=0,3651

v=1, SSIM=0,7447 *v=2, SSIM=0,7866* ***v=3, SSIM=0,7981*** *v=4, SSIM=0,6839*

Sucesion de imágenes para cada vecindad (v), para el mejor h (h=275) y para radio de búsqueda = 10 (ventana de 21x21), para la imagen de lena con ruido gausiano de desviación 50.

La siguiente sucesión de imágenes es un ejemplo para la gráfica b), tomando como imagen la imagen de Lena con ruido gaussiano de desviación 50:

r=2, SSIM= 0,5468 *r=5, SSIM= 0,7536* *r=7, SSIM= 0,7890* ***r=10, SSIM= 0,7981***

Sucesion de imágenes para vecindad=3 (ventana de 7x7), para el mejor h (h=275) y para cada radio de búsqueda (r), para la imagen de lena con ruido gausiano de desviación 50.

La siguiente sucesión de imágenes es un ejemplo para la gráfica c) y d), tomando como imagen la imagen de Lena con ruido gaussiano de desviación 10:

Imagen con ruido gaussiano desviación 10. SSIM= 0,8708

h=25, SSIM=0,8803 ***h=50, SSIM=0,9615*** *h=100, SSIM=0,9322*

Sucesion de imágenes para la mejor vecindad =2 (ventana de 5x5), para h=25, 50, 100 y para el mejor radio de búsqueda = 5 (ventana de 11x11), para la imagen de lena con ruido gausiano de desviación 10.

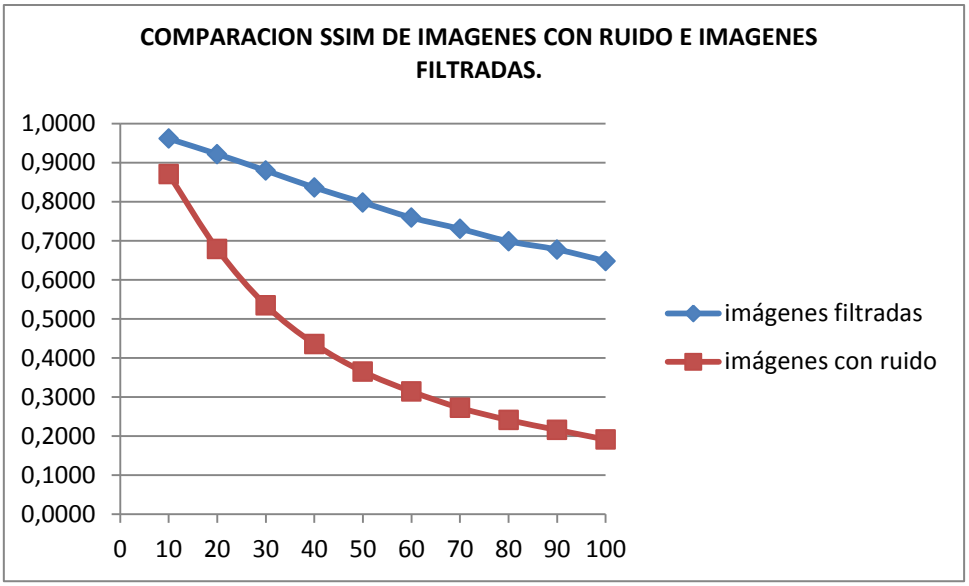
A continuación se analizan los resultados:

En general se observa que para una determinada vecindad, tras alcanzar el máximo SSIM, este empeora a medida que aumenta el radio. Este máximo SSIM se alcanza en función del ruido que tenga la imagen. Cuanto más ruido tenga, mayor es la vecindad en la que se alcanza. Por tanto a más ruido, combinaciones de mayores vecindades, y mayores radios funcionan mejor. También según aumenta la vecindad, tras alcanzar el máximo SSIM, los resultados empeoran filtrando menos ruido.

Tras alcanzar el máximo SSIM, a medida que aumenta grado de filtrado (h), el SSIM empeora. Se observa que para alcanzar este máximo SSIM, se necesita más grado de filtrado cuanto mayor sea la vecindad para el mejor radio. También para la mejor vecindad y distinto radio, el máximo SSIM se alcanza para el mismo valor del grado de filtrado.

Los resultados visuales muestran que para una buena combinación de valores para la vecindad y el radio, a mayor grado de filtrado más ruido se filtra y más detalles de la imagen se pierden. Por el contrario si el grado de filtrado es pequeño se filtra menos ruido. Estos valores de la vecindad y el radio dependerán también del ruido. Si el ruido es grande trabajaremos con combinaciones de valores para el radio y la vecindad con mayor valor que si el ruido es poco, que bastara con vecindades y radios pequeños. Podemos representar el valor del grado de filtrado en función del ruido, visto de la forma h=f(ruido)=x*ruido. A más ruido con la combinación adecuada de valores para la vecindad y el radio, x será menor que si el ruido es pequeño.

La siguiente grafica representa una comparación entre el SSIM obtenido en la mejor imagen de las imágenes filtradas de Lena ruido gaussiano para cada desviación, con el SSIM que tenían las imágenes con ruido:



GRÁFICA 3: Comparación del SSIM entre las imágenes con ruido gaussiano de desviación 10, 20, 30, 40, 50, 60, 70, 80, 90 y 100, y las imágenes obtenidas tras aplicar la mejor combinación de parámetros a cada una.

4.2.1.2. Aplicación de los resultados a imágenes en escala de gris con ruido gaussiano y rician

Aplicación de resultados a imágenes con ruido gaussiano:

A continuación vamos a aplicar el filtro Non Local Means con los parámetros que en el estudio anterior hemos considerado que son los más adecuados, a varias imágenes con ruido gaussiano:

Imagen Original: mandril.png

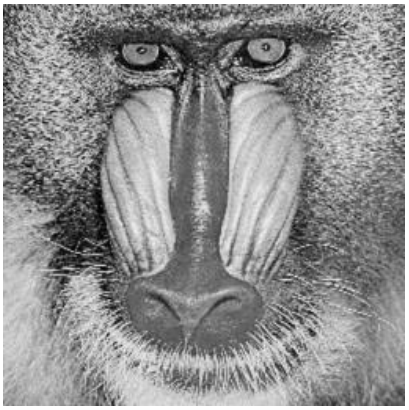
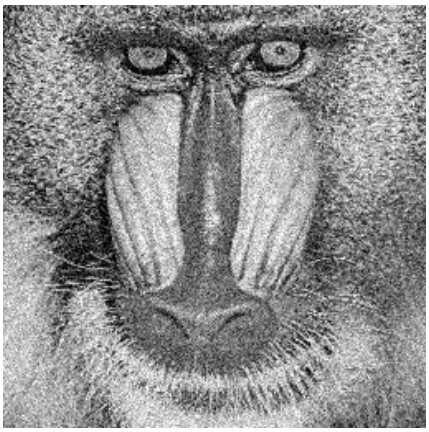
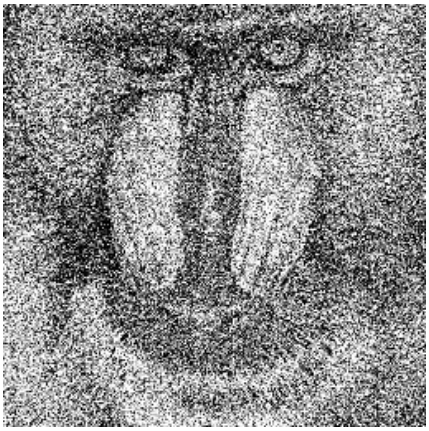


Imagen con ruido gaussiano Imagen filtrada

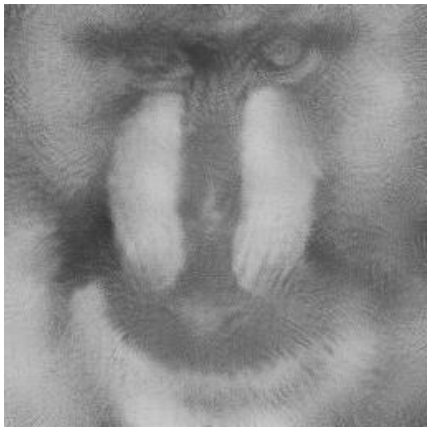


Desviación=25

h=150,v=2,r=5,SSIM= 0,6556



Desviación=85



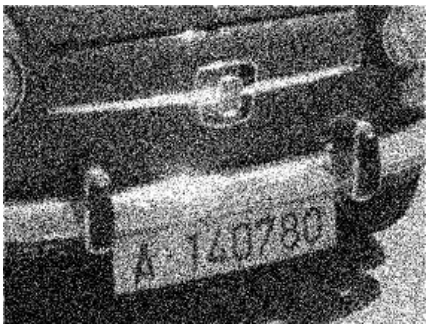
h=455,v=3,r=10,SSIM= 0,2870

Imagen Original: matricula.png

Imagen con ruido gaussiano

Imagen filtrada

Desviación=15 *h=75,v=2,r=5,SSIM= 0,9005*



Desviación=55 *h=303,v=3,r=10,SSIM= 0,6817*

Aplicación de resultados a imágenes con ruido rician:

Se aplicaran los parámetros siguiendo la tabla resumen obtenida para ruido gaussiano. Para añadir ruido rician utilizamos el fichero ricernd.m de la siguiente forma (por ejemplo con desviación 50):

input_50R=ricernd(double(original),50);

A continuación se muestran los resultados obtenidos para la imagen de Lena de tamaño 256x256, a la que se le ha añadido ruido rician con distinta desviación y los resultados del filtrado tras aplicar los parámetros obtenidos para ruido gaussiano:

IMAGEN	PSNR (dB)	SSIM
input_10R	28,17	0,7015
input_20R	22,2	0,4532
input_30R	18,82	0,3206
input_40R	16,42	0,2399
input_50R	14,66	0,1828
input_60R	13,28	0,1425
input_70R	12,15	0,1174
input_80R	11,31	0,0979
input_90R	10,5	0,0834
input_100R	9,91	0,0675

IMAGEN	PSNR (dB)	SSIM
input_10R-50-2-5	32,52	0,8971
input_20R-120-3-7	28,24	0,8081
input_30R-180-3-7	26,12	0,7388
input_40R-230-3-10	23,59	0,6707
input_50R-275-3-10	21,28	0,6029
input_60R-330-3-10	19,21	0,5486
input_70R-385-3-10	17,42	0,5051
input_80R-428-3-10	15,99	0,4806
input_90R-481-3-10	14,66	0,4524
input_100R-535-3-10	13,66	0,4383

TABLA 14: Tablas de PSNR y SSIM para la imagen lena512 con ruido rician de desviación 10, 20, 30, 40, 50, 60, 70, 80, 90 y100 (a la izquierda) y de resultados obtenidos tras aplicar los parámetros exactos a cada una(a la derecha).

A continuación se muestran varias imágenes:

Imagen Original: lena256.png

Imagen con ruido rician | Imagen filtrada

Desviación=30 | *h=180,v=3,r=7,SSIM= 0,7388*

Desviación=60 | *h=330,v=3,r=10,SSIM=0,5486*

Imagen Original: cameraman.png

Imagen con ruido rician | Imagen filtrada

Desviación=25

h=150,v=3,r=7,SSIM= 0,7455

Desviación=55

h=330,v=3,r=10,SSIM= 0,5030

Imagen Original: radiografia.png

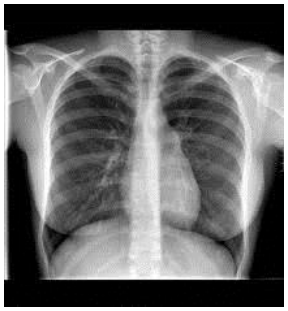
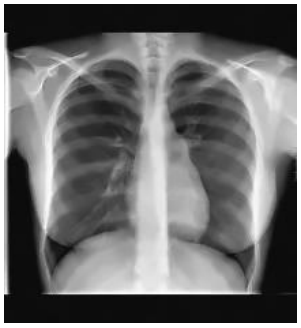


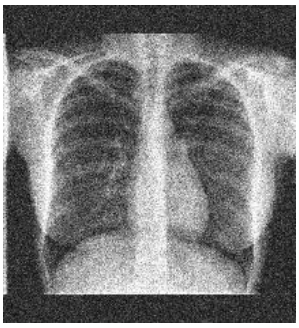
Imagen con ruido rician

Imagen filtrada

Desviación=10



h=50,v=2,r=5,SSIM= 0,7954



Desviación=40 *h=230,v=3,r=10,SSIM= 0,5630*

Los resultados del filtrado dan “peores” resultados de SSIM que para el ruido gaussiano. Principalmente es debido a que las imágenes con ruido rician también tienen un SSIM menor que las imágenes con ruido gaussiano. Evidentemente no es lo mismo ruido gaussiano con desviación 10 que ruido rician con desviación 10, puesto que la distribución de probabilidad es diferente. Las imágenes filtradas, visualmente, pierden contraste. Pese a esto, el ruido queda eliminado. Para solucionar el problema del contraste bastaría con realzarlo. Aun así los resultados obtenidos son aceptables.

4.2.2. Estudio de la versión 4: distancia euclídea ponderada

4.2.2.1. Estudio de parámetros para imágenes en escala de gris con ruido gaussiano

Esta versión introduce una modificación respecto a la versión 3. Se calcula la distancia euclídea ponderada por un Kernel gaussiano con cierta desviación típica.

En este apartado hemos estudiado diferentes desviaciones típicas, lo que se traduce en kernels que tratan de diferente forma la vecindad de los píxeles.

El primer Kernel que utilizamos tiene desviación típica igual a 1. En este caso, los resultados obtenidos, tanto numéricos como visuales, no son los esperados. Consideramos que este mal resultado es debido a que este Kernel, cuando la vecindad tiene tamaños más grandes, pondera los pixeles más lejanos con valor próximos a 0, por lo que casi no se tienen en cuenta los píxeles más lejanos de la vecindad del pixel correspondiente. Para solucionar este problema hemos decidido estudiar otros kernels con una desviación mayor. En primer lugar se ha utilizado un Kernel gaussiano con desviación 5. Así conseguimos dar más peso a los píxeles más lejanos del vecindario, reduciendo por tanto el peso que se le da a los más cercanos. Para obtener un Kernel más ajustado, se han hecho las pruebas con dos kernels gaussianos más, uno con desviación 3 y otro 7.

A continuación se muestran las tablas con los resultados para cada Kernel. Tras este primer análisis también mostramos las tablas resúmenes para cada Kernel junto con una gráfica.

Resultados:

KERNEL Gaussiano de desviación 1

RUIDO=10

IMAGEN	PSNR (dB)	SSIM

IMAGEN	PSNR (dB)	SSIM

input_10R-15-1-2	34,30	0,9545	input_10R-30-1-2	33,81	0,9492
input_10R-15-2-2	34,51	0,9559	input_10R-30-2-2	33,69	0,9482
input_10R-15-3-2	34,52	0,9560	input_10R-30-3-2	33,69	0,9482
input_10R-15-4-2	34,52	0,9560	input_10R-30-4-2	33,69	0,9482
input_10R-15-1-5	34,58	0,9564	input_10R-30-1-5	32,50	0,9275
input_10R-15-2-5	34,72	0,9563	input_10R-30-2-5	32,24	0,9233
input_10R-15-3-5	34,73	0,9563	input_10R-30-3-5	32,23	0,9230
input_10R-15-4-5	34,73	0,9563	input_10R-30-4-5	32,23	0,9230
input_10R-15-1-7	34,41	0,9528	input_10R-30-1-7	31,98	0,9159
input_10R-15-2-7	34,52	0,9522	input_10R-30-2-7	31,69	0,9107
input_10R-15-3-7	34,52	0,9522	input_10R-30-3-7	31,67	0,9102
input_10R-15-4-7	34,52	0,9522	input_10R-30-4-7	31,67	0,9102
input_10R-15-1-10	34,15	0,9489	input_10R-30-1-10	31,44	0,9065
input_10R-15-2-10	34,21	0,9481	input_10R-30-2-10	31,12	0,9003
input_10R-15-3-10	34,22	0,9480	input_10R-30-3-10	31,10	0,8998
input_10R-15-4-10	34,22	0,9480	input_10R-30-4-10	31,10	0,8998

TABLA 15: Tablas del PSNR y SSIM de las imágenes resultantes del filtrado, con la versión 4, de la imagen de Lena con ruido gaussiano con desviación 10 para cada combinación de parámetros con h=15 (izquierda) y h=30 (derecha) empleando un Kernel gaussiano con desviación 1.

RUIDO 20:

Visto que con radio 2 no da buenos resultados, a partir de ahora vamos a eliminar esta prueba.

IMAGEN	PSNR (dB)	SSIM
input_20R-20-1-5	28,89	0,8820
input_20R-20-2-5	29,39	0,8939
input_20R-20-3-5	29,43	0,8948
input_20R-20-4-5	29,43	0,8948
input_20R-20-1-7	29,17	0,8835
input_20R-20-2-7	29,73	0,8960
input_20R-20-3-7	29,77	0,8969
input_20R-20-4-7	29,77	0,8969
input_20R-20-1-10	29,27	0,8800
input_20R-20-2-10	29,87	0,8931
input_20R-20-3-10	29,91	0,8939
input_20R-20-4-10	29,91	0,8939

IMAGEN	PSNR (dB)	SSIM
input_20R-30-1-5	30,92	0,9094
input_20R-30-2-5	31,14	0,9110
input_20R-30-3-5	31,15	0,9111
input_20R-30-4-5	31,15	0,9111
input_20R-30-1-7	30,70	0,9031
input_20R-30-2-7	30,86	0,9040
input_20R-30-3-7	30,62	0,8877
input_20R-30-4-7	30,87	0,9039
input_20R-30-1-10	30,36	0,8946
input_20R-30-2-10	30,48	0,8952
input_20R-30-3-10	30,48	0,8951
input_20R-30-4-10	30,48	0,8951

IMAGEN	PSNR (dB)	SSIM
input_20R-40-1-5	30,68	0,9016
input_20R-40-2-5	30,58	0,8986
input_20R-40-3-5	30,57	0,8983
input_20R-40-4-5	30,57	0,8983

input_20R-40-1-7	30,23	0,8905
input_20R-40-2-7	30,10	0,8864
input_20R-40-3-7	30,08	0,8860
input_20R-40-4-7	30,08	0,8859
input_20R-40-1-10	29,71	0,8789
input_20R-40-2-10	29,56	0,8742
input_20R-40-3-10	29,54	0,8737
input_20R-40-4-10	29,54	0,8737

TABLA 16: Tablas del PSNR y SSIM de las imágenes resultantes del filtrado, con la versión 4, de la imagen de Lena con ruido gaussiano con desviación 20 para cada combinación de parámetros con h=20 (arriba a la izquierda), h=30(arriba a la derecha) y h=40 (abajo) empleando un Kernel gaussiano con desviación 1.

RUIDO 30

IMAGEN	PSNR (dB)	SSIM
input_30R-35-1-5	27,51	0,8347
input_30R-35-2-5	28,14	0,8496
input_30R-35-3-5	28,19	0,8506
input_30R-35-4-5	28,19	0,8506
input_30R-35-1-7	27,60	0,8342
input_30R-35-2-7	28,22	0,8493
input_30R-35-3-7	28,27	0,8502
input_30R-35-4-7	28,27	0,8502
input_30R-35-1-10	27,52	0,8272
input_30R-35-2-10	28,12	0,8430
input_30R-35-3-10	28,16	0,8439
input_30R-35-4-10	28,16	0,8440

IMAGEN	PSNR (dB)	SSIM
input_30R-50-1-5	28,80	0,8610
input_30R-50-2-5	28,93	0,8629
input_30R-50-3-5	28,93	0,8629
input_30R-50-4-5	28,93	0,8629
input_30R-50-1-7	28,47	0,8526
input_30R-50-2-7	28,56	0,8536
input_30R-50-3-7	28,52	0,8157
input_30R-50-4-7	28,56	0,8534
input_30R-50-1-10	28,01	0,8403
input_30R-50-2-10	28,07	0,8411
input_30R-50-3-10	28,06	0,8409
input_30R-50-4-10	28,06	0,8409

IMAGEN	PSNR (dB)	SSIM
input_30R-65-1-5	28,49	0,8534
input_30R-65-2-5	28,38	0,8505
input_30R-65-3-5	28,37	0,8501
input_30R-65-4-5	28,37	0,8501
input_30R-65-1-7	27,91	0,8378
input_30R-65-2-7	27,77	0,8335
input_30R-65-3-7	27,75	0,8330
input_30R-65-4-7	27,75	0,8329
input_30R-65-1-10	27,19	0,8192
input_30R-65-2-10	27,05	0,8143
input_30R-65-3-10	27,03	0,8136
input_30R-65-4-10	27,03	0,8136

TABLA 17: Tablas del PSNR y SSIM de las imágenes resultantes del filtrado, con la versión 4, de la imagen de Lena con ruido gaussiano con desviación 30 para cada combinación de parámetros con h=35 (arriba a la izquierda), h=50 (arriba a la derecha) y h=65 (abajo) empleando un Kernel gaussiano con desviación 1.

RUIDO 40

IMAGEN	PSNR (dB)	SSIM
input_40R-45-1-5	25,45	0,7651
input_40R-45-2-5	26,17	0,7871
input_40R-45-3-5	26,22	0,7886
input_40R-45-4-5	26,22	0,7886
input_40R-45-1-7	25,57	0,7663
input_40R-45-2-7	26,30	0,7895
input_40R-45-3-7	26,35	0,7910
input_40R-45-4-7	26,35	0,7910
input_40R-45-1-10	25,49	0,7578
input_40R-45-2-10	26,20	0,7824
input_40R-45-3-10	26,25	0,7839
input_40R-45-4-10	26,25	0,7839

IMAGEN	PSNR (dB)	SSIM
input_40R-60-1-5	27,04	0,8101
input_40R-60-2-5	27,36	0,8185
input_40R-60-3-5	27,37	0,8189
input_40R-60-4-5	27,38	0,8189
input_40R-60-1-7	26,80	0,8044
input_40R-60-2-7	27,07	0,8124
input_40R-60-3-7	27,08	0,8127
input_40R-60-4-7	27,08	0,8127
input_40R-60-1-10	26,36	0,7898
input_40R-60-2-10	26,59	0,7983
input_40R-60-3-10	26,60	0,7986
input_40R-60-4-10	26,60	0,7986

IMAGEN	PSNR (dB)	SSIM
input_40R-75-1-5	27,27	0,8170
input_40R-75-2-5	27,31	0,8180
input_40R-75-3-5	27,30	0,8179
input_40R-75-4-5	27,30	0,8179
input_40R-75-1-7	26,76	0,8051
input_40R-75-2-7	26,75	0,8049
input_40R-75-3-7	26,74	0,8047
input_40R-75-4-7	26,74	0,8047
input_40R-75-1-10	26,05	0,7842
input_40R-75-2-10	26,02	0,7838
input_40R-75-3-10	26,01	0,7835
input_40R-75-4-10	26,01	0,7835

TABLA 18: Tablas del PSNR y SSIM de las imágenes resultantes del filtrado, con la versión 4, de la imagen de Lena con ruido gaussiano con desviación 40 para cada combinación de parámetros con h=45 (arriba a la izquierda), h=60 (arriba a la derecha) y h=75 (abajo) empleando un Kernel gaussiano con desviación 1.

RUIDO 50

IMAGEN	PSNR (dB)	SSIM
input_50R-75-1-5	25,82	0,7636
input_50R-75-2-5	26,17	0,7754
input_50R-75-3-5	26,19	0,7760
input_50R-75-4-5	26,19	0,7760
input_50R-75-1-7	25,56	0,7591
input_50R-75-2-7	25,85	0,7708
input_50R-75-3-7	24,89	0,6689
input_50R-75-4-7	25,87	0,7713
input_50R-75-1-10	25,05	0,7398
input_50R-75-2-10	25,28	0,7518

IMAGEN	PSNR (dB)	SSIM
input_50R-85-1-5	26,10	0,7740
input_50R-85-2-5	26,27	0,7804
input_50R-85-3-5	26,28	0,7806
input_50R-85-4-5	26,28	0,7807
input_50R-85-1-7	25,68	0,7659
input_50R-85-2-7	25,80	0,7716
input_50R-85-3-7	25,41	0,6801
input_50R-85-4-7	25,80	0,7717
input_50R-85-1-10	24,99	0,7425
input_50R-85-2-10	25,06	0,7482

input_50R-75-3-10	25,29	0,7524
input_50R-75-4-10	25,29	0,7524

input_50R-85-3-10	25,06	0,7482
input_50R-85-4-10	25,06	0,7482

IMAGEN	PSNR (dB)	SSIM
input_50R-100-1-5	26,15	0,7782
input_50R-100-2-5	26,17	0,7797
input_50R-100-3-5	26,17	0,7797
input_50R-100-4-5	26,17	0,7797
input_50R-100-1-7	25,54	0,7646
input_50R-100-2-7	25,52	0,7651
input_50R-100-3-7	25,51	0,7649
input_50R-100-4-7	25,51	0,7649
input_50R-100-1-10	24,66	0,7352
input_50R-100-2-10	24,61	0,7349
input_50R-100-3-10	24,60	0,7346
input_50R-100-4-10	24,60	0,7345

TABLA 19: Tablas del PSNR y SSIM de las imágenes resultantes del filtrado, con la versión 4, de la imagen de Lena con ruido gaussiano con desviación 50 para cada combinación de parámetros con h=75 (arriba a la izquierda), h=85 (arriba a la derecha) y h=100 (abajo) empleando un Kernel gaussiano con desviación 1.

RUIDO 60

IMAGEN	PSNR (dB)	SSIM
input_60R-80-1-5	24,27	0,6998
input_60R-80-2-5	24,78	0,7193
input_60R-80-3-5	24,81	0,7206
input_60R-80-4-5	24,82	0,7206
input_60R-80-1-7	24,15	0,6982
input_60R-80-2-7	24,62	0,7190
input_60R-80-3-7	24,64	0,7202
input_60R-80-4-7	24,65	0,7202
input_60R-80-1-10	23,77	0,6800
input_60R-80-2-10	24,18	0,7017
input_60R-80-3-10	24,19	0,7030
input_60R-80-4-10	24,20	0,7030

IMAGEN	PSNR (dB)	SSIM
input_60R-95-1-5	24,92	0,7256
input_60R-95-2-5	25,18	0,7365
input_60R-95-3-5	25,19	0,7370
input_60R-95-4-5	25,20	0,7370
input_60R-95-1-7	24,59	0,7203
input_60R-95-2-7	24,79	0,7314
input_60R-95-3-7	24,80	0,7319
input_60R-95-4-7	24,80	0,7319
input_60R-95-1-10	23,97	0,6970
input_60R-95-2-10	24,12	0,7080
input_60R-95-3-10	24,12	0,7085
input_60R-95-4-10	24,12	0,7085

IMAGEN	PSNR (dB)	SSIM
input_60R-110-1-5	25,14	0,7361
input_60R-110-2-5	25,24	0,7413
input_60R-110-3-5	25,24	0,7415
input_60R-110-4-5	25,24	0,7415
input_60R-110-1-7	24,64	0,7272
input_60R-110-2-7	24,69	0,7319
input_60R-110-3-7	24,69	0,7320

input_60R-110-4-7	24,69	0,7320
input_60R-110-1-10	23,84	0,6989
input_60R-110-2-10	23,86	0,7029
input_60R-110-3-10	23,85	0,7028
input_60R-110-4-10	23,85	0,7028

TABLA 20: Tablas del PSNR y SSIM de las imágenes resultantes del filtrado, con la versión 4, de la imagen de Lena con ruido gaussiano con desviación 60 para cada combinación de parámetros con h=80 (arriba a la izquierda), h=95 (arriba a la derecha) y h=110 (abajo) empleando un Kernel gaussiano con desviación 1.

RUIDO 70

IMAGEN	PSNR (dB)	SSIM
input_70R-90-1-5	23,31	0,6554
input_70R-90-2-5	23,87	0,6784
input_70R-90-3-5	23,90	0,6800
input_70R-90-4-5	23,90	0,6800
input_70R-90-1-7	23,23	0,6557
input_70R-90-2-7	23,73	0,6809
input_70R-90-3-7	23,76	0,6825
input_70R-90-4-7	23,76	0,6825
input_70R-90-1-10	22,83	0,6350
input_70R-90-2-10	23,27	0,6612
input_70R-90-3-10	23,29	0,6627
input_70R-90-4-10	23,29	0,6627

IMAGEN	PSNR (dB)	SSIM
input_70R-105-1-5	24,00	0,6848
input_70R-105-2-5	24,33	0,6996
input_70R-105-3-5	24,34	0,7004
input_70R-105-4-5	24,34	0,7004
input_70R-105-1-7	23,73	0,6830
input_70R-105-2-7	23,99	0,6988
input_70R-105-3-7	24,00	0,6996
input_70R-105-4-7	24,00	0,6996
input_70R-105-1-10	23,12	0,6579
input_70R-105-2-10	23,32	0,6738
input_70R-105-3-10	23,33	0,6746
input_70R-105-4-10	23,33	0,6747

IMAGEN	PSNR (dB)	SSIM
input_70R-120-1-5	24,31	0,7001
input_70R-120-2-5	24,47	0,7087
input_70R-120-3-5	24,48	0,7092
input_70R-120-4-5	24,48	0,7092
input_70R-120-1-7	23,88	0,6959
input_70R-120-2-7	23,99	0,7048
input_70R-120-3-7	24,00	0,7052
input_70R-120-4-7	23,99	0,7052
input_70R-120-1-10	23,13	0,6667
input_70R-120-2-10	23,19	0,6752
input_70R-120-3-10	23,19	0,6755
input_70R-120-4-10	23,19	0,6755

TABLA 21: Tablas del PSNR y SSIM de las imágenes resultantes del filtrado, con la versión 4, de la imagen de Lena con ruido gaussiano con desviación 70 para cada combinación de parámetros con h=90 (arriba a la izquierda), h=105 (arriba a la derecha) y h=120 (abajo) empleando un Kernel gaussiano con desviación 1.

RUIDO 80

IMAGEN	PSNR (dB)	SSIM
input_80R-100-1-7	22,39	0,6180
input_80R-100-2-7	22,88	0,6451
input_80R-100-3-7	22,90	0,6468
input_80R-100-4-7	22,90	0,6469
input_80R-100-1-10	22,00	0,5965
input_80R-100-2-10	22,42	0,6248
input_80R-100-3-10	22,44	0,6265
input_80R-100-4-10	22,44	0,6266

IMAGEN	PSNR (dB)	SSIM
input_80R-120-1-7	22,96	0,6532
input_80R-120-2-7	23,18	0,6693
input_80R-120-3-7	23,19	0,6702
input_80R-120-4-7	23,19	0,6702
input_80R-120-1-10	22,36	0,6274
input_80R-120-2-10	22,52	0,6440
input_80R-120-3-10	22,53	0,6448
input_80R-120-4-10	22,53	0,6448

IMAGEN	PSNR (dB)	SSIM
input_80R-140-1-7	23,12	0,6691
input_80R-140-2-7	23,20	0,6776
input_80R-140-3-7	23,20	0,6780
input_80R-140-4-7	23,20	0,6780
input_80R-140-1-10	22,37	0,6397
input_80R-140-2-10	22,41	0,6481
input_80R-140-3-10	22,41	0,6485
input_80R-140-4-10	22,41	0,6485

TABLA 22: Tablas del PSNR y SSIM de las imágenes resultantes del filtrado, con la versión 4, de la imagen de Lena con ruido gaussiano con desviación 80 para cada combinación de parámetros con h=100 (arriba a la izquierda), h=120 (arriba a la derecha) y h=140 (abajo) empleando un Kernel gaussiano con desviación 1.

RUIDO 90

IMAGEN	PSNR (dB)	SSIM
input_90R-95-1-7	21,02	0,5496
input_90R-95-2-7	21,73	0,5864
input_90R-95-3-7	21,77	0,5889
input_90R-95-4-7	21,78	0,5889
input_90R-95-1-10	20,81	0,5303
input_90R-95-2-10	21,46	0,5683
input_90R-95-3-10	21,50	0,5710
input_90R-95-4-10	21,50	0,5711

IMAGEN	PSNR (dB)	SSIM
input_90R-115-1-7	22,05	0,6053
input_90R-115-2-7	22,44	0,6316
input_90R-115-3-7	22,46	0,6333
input_90R-115-4-7	22,46	0,6333
input_90R-115-1-10	21,61	0,5814
input_90R-115-2-10	21,93	0,6083
input_90R-115-3-10	21,95	0,6100
input_90R-115-4-10	21,95	0,6100

IMAGEN	PSNR (dB)	SSIM
input_90R-135-1-7	22,47	0,6367
input_90R-135-2-7	22,65	0,6531
input_90R-135-3-7	22,65	0,6540
input_90R-135-4-7	22,65	0,6540
input_90R-135-1-10	21,86	0,6094
input_90R-135-2-10	21,98	0,6258
input_90R-135-3-10	21,99	0,6267
input_90R-135-4-10	21,99	0,6267

TABLA 23: Tablas del PSNR y SSIM de las imágenes resultantes del filtrado, con la versión 4, de la imagen de Lena con ruido gaussiano con desviación 90 para cada combinación de parámetros con h=95 (arriba a la izquierda), h=115 (arriba a la derecha) y h=135 (abajo) empleando un Kernel gaussiano con desviación 1.

RUIDO 100

IMAGEN	PSNR (dB)	SSIM
input_100R-125-1-7	21,35	0,5747
input_100R-125-2-7	21,70	0,6020
input_100R-125-3-7	21,72	0,6037
input_100R-125-4-7	21,72	0,6038
input_100R-125-1-10	20,93	0,5508
input_100R-125-2-10	21,22	0,5791
input_100R-125-3-10	21,24	0,5809
input_100R-125-4-10	21,24	0,5810

IMAGEN	PSNR (dB)	SSIM
input_100R-150-1-7	21,78	0,6122
input_100R-150-2-7	21,92	0,6283
input_100R-150-3-7	21,93	0,6293
input_100R-150-4-7	21,93	0,6293
input_100R-150-1-10	21,21	0,5857
input_100R-150-2-10	21,30	0,6023
input_100R-150-3-10	21,31	0,6032
input_100R-150-4-10	21,31	0,6033

IMAGEN	PSNR (dB)	SSIM
input_100R-175-1-7	21,90	0,6302
input_100R-175-2-7	21,94	0,6390
input_100R-175-3-7	21,94	0,6394
input_100R-175-4-7	21,94	0,6394
input_100R-175-1-10	21,23	0,6018
input_100R-175-2-10	21,25	0,6105
input_100R-175-3-10	21,25	0,6109
input_100R-175-4-10	21,25	0,6109

TABLA 24: Tablas del PSNR y SSIM de las imágenes resultantes del filtrado, con la versión 4, de la imagen de Lena con ruido gaussiano con desviación 100 para cada combinación de parámetros con h=125 (arriba a la izquierda), h=150 (arriba a la derecha) y h=175 (abajo) empleando un Kernel gaussiano con desviación 1.

Kernel Gaussiano de desviación 5:

RUIDO 10

IMAGEN	PSNR (dB)	SSIM
input_10R-10-1-2	32,48	0,9435
input_10R-10-2-2	32,81	0,9495
input_10R-10-3-2	32,93	0,9520
input_10R-10-4-2	32,93	0,9529
input_10R-10-1-5	33,98	0,9579
input_10R-10-2-5	34,20	0,9616
input_10R-10-3-5	34,08	0,9613
input_10R-10-4-5	33,95	0,9609
input_10R-10-1-7	34,21	0,9572
input_10R-10-2-7	34,33	0,9595
input_10R-10-3-7	34,13	0,9578

IMAGEN	PSNR (dB)	SSIM
input_10R-15-1-2	34,64	0,9560
input_10R-15-2-2	34,82	0,9570
input_10R-15-3-2	34,87	0,9570
input_10R-15-4-2	34,86	0,9567
input_10R-15-1-5	34,62	0,9547
input_10R-15-2-5	34,36	0,9484
input_10R-15-3-5	34,18	0,9446
input_10R-15-4-5	34,10	0,9426
input_10R-15-1-7	34,36	0,9504
input_10R-15-2-7	33,92	0,9414
input_10R-15-3-7	33,66	0,9353

input_10R-10-4-7	33,97	0,9565
input_10R-10-1-10	34,30	0,9551
input_10R-10-2-10	34,33	0,9565
input_10R-10-3-10	34,07	0,9536
input_10R-10-4-10	33,88	0,9514

input_10R-15-4-7	33,55	0,9320
input_10R-15-1-10	34,00	0,9461
input_10R-15-2-10	33,41	0,9346
input_10R-15-3-10	33,07	0,9263
input_10R-15-4-10	32,92	0,9215

TABLA 25: Tablas del PSNR y SSIM de las imágenes resultantes del filtrado, con la versión 4, de la imagen de Lena con ruido gaussiano con desviación 10 para cada combinación de parámetros con h=10 (izquierda) y h=15 (derecha) empleando un Kernel gaussiano con desviación 5.

RUIDO 20

IMAGEN	PSNR (dB)	SSIM
input_20R-20-1-5	30,28	0,9054
input_20R-20-2-5	31,24	0,9206
input_20R-20-3-5	31,35	0,9214
input_20R-20-4-5	31,33	0,9211
input_20R-20-1-7	30,54	0,9057
input_20R-20-2-7	31,38	0,9178
input_20R-20-3-7	28,80	0,8686
input_20R-20-4-7	31,28	0,9127
input_20R-20-1-10	30,58	0,9013
input_20R-20-2-10	31,24	0,9110
input_20R-20-3-10	31,09	0,9052
input_20R-20-4-10	30,93	0,9003

IMAGEN	PSNR (dB)	SSIM
input_20R-30-1-5	31,24	0,9113
input_20R-30-2-5	30,87	0,9018
input_20R-30-3-5	30,46	0,8937
input_20R-30-4-5	30,18	0,8885
input_20R-30-1-7	30,91	0,9035
input_20R-30-2-7	30,36	0,8897
input_20R-30-3-7	29,84	0,8775
input_20R-30-4-7	29,49	0,8696
input_20R-30-1-10	30,49	0,8944
input_20R-30-2-10	29,80	0,8774
input_20R-30-3-10	29,19	0,8614
input_20R-30-4-10	28,79	0,8507

TABLA 26: Tablas del PSNR y SSIM de las imágenes resultantes del filtrado, con la versión 4, de la imagen de Lena con ruido gaussiano con desviación 20 para cada combinación de parámetros con h=20 (izquierda) y h=20 (derecha) empleando un Kernel gaussiano con desviación 5.

RUIDO 30

IMAGEN	PSNR (dB)	SSIM
input_30R-30-1-5	27,95	0,8461
input_30R-30-2-5	29,31	0,8754
input_30R-30-3-5	29,50	0,8781
input_30R-30-4-5	29,49	0,8774
input_30R-30-1-7	28,24	0,8496
input_30R-30-2-7	29,44	0,8756
input_30R-30-3-7	26,31	0,7798
input_30R-30-4-7	29,28	0,8685
input_30R-30-1-10	28,27	0,8452
input_30R-30-2-10	29,26	0,8691
input_30R-30-3-10	29,06	0,8619
input_30R-30-4-10	28,80	0,8541

IMAGEN	PSNR (dB)	SSIM
input_30R-40-1-5	29,18	0,8683
input_30R-40-2-5	29,24	0,8675
input_30R-40-3-5	28,89	0,8596
input_30R-40-4-5	28,60	0,8538
input_30R-40-1-7	29,00	0,8635
input_30R-40-2-7	28,85	0,8580
input_30R-40-3-7	28,36	0,8450
input_30R-40-4-7	27,99	0,8358
input_30R-40-1-10	28,66	0,8542
input_30R-40-2-10	28,37	0,8461
input_30R-40-3-10	27,78	0,8295
input_30R-40-4-10	27,36	0,8170

TABLA 27: Tablas del PSNR y SSIM de las imágenes resultantes del filtrado, con la versión 4, de la imagen de Lena con ruido gaussiano con desviación 30 para cada combinación de parámetros con h=30 (izquierda) y h=40 (derecha) empleando un Kernel gaussiano con desviación 5.

RUIDO 40

IMAGEN	PSNR (dB)	SSIM
input_40R-30-1-5	22,91	0,6878
input_40R-30-2-5	24,25	0,7498
input_40R-30-3-5	24,64	0,7699
input_40R-30-4-5	24,80	0,7783
input_40R-30-1-7	23,73	0,7107
input_40R-30-2-7	25,59	0,7904
input_40R-30-3-7	26,09	0,8127
input_40R-30-4-7	26,24	0,8207
input_40R-30-1-10	24,28	0,7199
input_40R-30-2-10	26,53	0,8105
input_40R-30-3-10	27,04	0,8308
input_40R-30-4-10	27,14	0,8358

IMAGEN	PSNR (dB)	SSIM
input_40R-40-1-5	26,25	0,7900
input_40R-40-2-5	27,79	0,8310
input_40R-40-3-5	27,97	0,8355
input_40R-40-4-5	27,92	0,8347
input_40R-40-1-7	26,54	0,7961
input_40R-40-2-7	27,89	0,8354
input_40R-40-3-7	27,84	0,8336
input_40R-40-4-7	27,65	0,8283
input_40R-40-1-10	26,53	0,7909
input_40R-40-2-10	27,67	0,8293
input_40R-40-3-10	27,45	0,8226
input_40R-40-4-10	27,15	0,8134

IMAGEN	PSNR (dB)	SSIM
input_40R-50-1-5	27,46	0,8208
input_40R-50-2-5	27,89	0,8311
input_40R-50-3-5	27,62	0,8251
input_40R-50-4-5	27,37	0,8197
input_40R-50-1-7	27,35	0,8192
input_40R-50-2-7	27,55	0,8255
input_40R-50-3-7	27,14	0,8140
input_40R-50-4-7	26,80	0,8046
input_40R-50-1-10	27,02	0,8084
input_40R-50-2-10	27,07	0,8131
input_40R-50-3-10	26,57	0,7975
input_40R-50-4-10	26,16	0,7844

TABLA 28: Tablas del PSNR y SSIM de las imágenes resultantes del filtrado, con la versión 4, de la imagen de Lena con ruido gaussiano con desviación 40 para cada combinación de parámetros con h=30 (arriba a la izquierda), h=40 (arriba a la derecha) y h=50 (abajo) empleando un Kernel gaussiano con desviación 5.

RUIDO 50

IMAGEN	PSNR (dB)	SSIM
input_50R-35-1-5	20,71	0,5899
input_50R-35-2-5	21,92	0,6500
input_50R-35-3-5	22,25	0,6689
input_50R-35-4-5	22,40	0,6771
input_50R-35-1-7	21,60	0,6172
input_50R-35-2-7	23,47	0,7051

IMAGEN	PSNR (dB)	SSIM
input_50R-40-1-5	22,59	0,6569
input_50R-40-2-5	24,59	0,7341
input_50R-40-3-5	25,21	0,7572
input_50R-40-4-5	25,46	0,7663
input_50R-40-1-7	23,30	0,6774
input_50R-40-2-7	25,63	0,7685

input_50R-35-3-7	17,71	0,4742
input_50R-35-4-7	24,20	0,7429
input_50R-35-1-10	22,23	0,6291
input_50R-35-2-10	24,68	0,7396
input_50R-35-3-10	25,36	0,7713
input_50R-35-4-10	25,55	0,7813
input_50R-35-3-15	26,08	0,7882
input_50R-35-4-15	26,17	0,7947

input_50R-40-3-7	19,83	0,5422
input_50R-40-4-7	26,34	0,7952
input_50R-40-1-10	23,67	0,6803
input_50R-40-2-10	26,11	0,7801
input_50R-40-3-10	26,53	0,7972
input_50R-40-4-10	26,50	0,7978
input_50R-40-3-15	26,35	0,7940
input_50R-40-4-15	26,18	0,7900

IMAGEN	PSNR (dB)	SSIM
input_50R-50-1-5	25,02	0,7373
input_50R-50-2-5	26,60	0,7887
input_50R-50-3-5	26,76	0,7953
input_50R-50-4-5	26,68	0,7950
input_50R-50-1-7	25,28	0,7464
input_50R-50-2-7	26,68	0,7978
input_50R-50-3-7	23,37	0,6374
input_50R-50-4-7	26,39	0,7923
input_50R-50-1-10	25,23	0,7389
input_50R-50-2-10	26,42	0,7914
input_50R-50-3-10	26,19	0,7867
input_50R-50-4-10	25,87	0,7768

TABLA 29: Tablas del PSNR y SSIM de las imágenes resultantes del filtrado, con la versión 4, de la imagen de Lena con ruido gaussiano con desviación 50 para cada combinación de parámetros con h=35 (arriba a la izquierda), h=40 (arriba a la derecha) y h=50 (abajo) empleando un Kernel gaussiano con desviación 5.

RUIDO 60

IMAGEN	PSNR (dB)	SSIM
input_60R-45-1-5	20,87	0,5799
input_60R-45-2-5	22,86	0,6613
input_60R-45-3-5	23,51	0,6868
input_60R-45-4-5	23,78	0,6971
input_60R-45-1-7	21,62	0,6015
input_60R-45-2-7	24,08	0,7052
input_60R-45-3-7	24,76	0,7332
input_60R-45-4-7	24,95	0,7419
input_60R-45-1-10	22,05	0,6057
input_60R-45-2-10	24,73	0,7253
input_60R-45-3-10	25,29	0,7522
input_60R-45-4-10	25,32	0,7568

IMAGEN	PSNR (dB)	SSIM
input_60R-60-1-5	24,02	0,6916
input_60R-60-2-5	25,52	0,7474
input_60R-60-3-5	25,64	0,7542
input_60R-60-4-5	25,56	0,7540
input_60R-60-1-7	24,24	0,7008
input_60R-60-2-7	25,55	0,7589
input_60R-60-3-7	25,47	0,7598
input_60R-60-4-7	25,26	0,7545
input_60R-60-1-10	24,14	0,6917
input_60R-60-2-10	25,24	0,7522
input_60R-60-3-10	25,02	0,7485
input_60R-60-4-10	24,73	0,7388

IMAGEN	PSNR (dB)	SSIM
input_60R-70-1-5	24,93	0,7243
input_60R-70-2-5	25,67	0,7545

input_60R-70-3-5	25,55	0,7532
input_60R-70-4-5	25,39	0,7497
input_60R-70-1-7	24,89	0,7285
input_60R-70-2-7	25,41	0,7574
input_60R-70-3-7	25,15	0,7505
input_60R-70-4-7	24,90	0,7426
input_60R-70-1-10	24,53	0,7144
input_60R-70-2-10	24,88	0,7432
input_60R-70-3-10	24,53	0,7321
input_60R-70-4-10	24,23	0,7203

TABLA 30: Tablas del PSNR y SSIM de las imágenes resultantes del filtrado, con la versión 4, de la imagen de Lena con ruido gaussiano con desviación 60 para cada combinación de parámetros con h=45 (arriba a la izquierda), h=60 (arriba a la derecha) y h=70 (abajo) empleando un Kernel gaussiano con desviación 5.

RUIDO 70

IMAGEN	PSNR (dB)	SSIM
input_70R-55-1-5	20,92	0,5665
input_70R-55-2-5	23,17	0,6570
input_70R-55-3-5	23,79	0,6832
input_70R-55-4-5	23,99	0,6926
input_70R-55-1-7	21,54	0,5848
input_70R-55-2-7	24,00	0,6952
input_70R-55-3-7	24,48	0,7207
input_70R-55-4-7	24,52	0,7262
input_70R-55-1-10	21,78	0,5827
input_70R-55-2-10	24,23	0,7057
input_70R-55-3-10	24,52	0,7284
input_70R-55-4-10	24,40	0,7287
input_70R-55-3-15	24,14	0,7230
input_70R-55-4-15	23,92	0,7193

IMAGEN	PSNR (dB)	SSIM
input_70R-65-1-5	22,71	0,6330
input_70R-65-2-5	24,48	0,7052
input_70R-65-3-5	24,72	0,7184
input_70R-65-4-5	24,70	0,7209
input_70R-65-1-7	23,03	0,6456
input_70R-65-2-7	24,65	0,7260
input_70R-65-3-7	24,68	0,7339
input_70R-65-4-7	24,52	0,7311
input_70R-65-1-10	22,97	0,6361
input_70R-65-2-10	24,37	0,7214
input_70R-65-3-10	24,26	0,7250
input_70R-65-4-10	24,01	0,7172

IMAGEN	PSNR (dB)	SSIM
input_70R-70-1-5	23,31	0,6557
input_70R-70-2-5	24,71	0,7149
input_70R-70-3-5	24,79	0,7227
input_70R-70-4-5	24,71	0,7227
input_70R-70-1-7	23,49	0,6662
input_70R-70-2-7	24,69	0,7303
input_70R-70-3-7	24,59	0,7324
input_70R-70-4-7	24,40	0,7275
input_70R-70-1-10	23,30	0,6542
input_70R-70-2-10	24,27	0,7211
input_70R-70-3-10	24,06	0,7189

input_70R-70-4-10	23,80	0,7094

TABLA 31: Tablas del PSNR y SSIM de las imágenes resultantes del filtrado, con la versión 4, de la imagen de Lena con ruido gaussiano con desviación 70 para cada combinación de parámetros con h=55 (arriba a la izquierda), h=65 (arriba a la derecha) y h=70 (abajo) empleando un Kernel gaussiano con desviación 5.

RUIDO 80

IMAGEN	PSNR (dB)	SSIM
input_80R-65-1-7	21,34	0,5701
input_80R-65-2-7	23,45	0,6772
input_80R-65-3-7	23,70	0,6976
input_80R-65-4-7	23,66	0,7005
input_80R-65-1-10	21,43	0,5632
input_80R-65-2-10	23,38	0,6805
input_80R-65-3-10	23,47	0,6978
input_80R-65-4-10	23,32	0,6961
input_80R-65-3-15	22,89	0,6833

IMAGEN	PSNR (dB)	SSIM
input_80R-80-1-7	22,75	0,6359
input_80R-80-2-7	23,72	0,6996
input_80R-80-3-7	23,63	0,7021
input_80R-80-4-7	23,48	0,6981
input_80R-80-1-10	22,50	0,6217
input_80R-80-2-10	23,26	0,6887
input_80R-80-3-10	23,07	0,6875
input_80R-80-4-10	22,87	0,6796

IMAGEN	PSNR (dB)	SSIM
input_80R-60-1-10	20,79	0,5343
input_80R-60-2-10	23,22	0,6669
input_80R-60-3-10	23,51	0,6945
input_80R-60-4-10	23,44	0,6975
input_80R-60-3-15	23,10	0,6869
input_80R-60-4-15	22,92	0,6862

TABLA 32: Tablas del PSNR y SSIM de las imágenes resultantes del filtrado, con la versión 4, de la imagen de Lena con ruido gaussiano con desviación 80 para cada combinación de parámetros con h=65 (arriba a la izquierda), h=80 (arriba a la derecha) y h=60 (abajo) empleando un Kernel gaussiano con desviación 5.

RUIDO 90

IMAGEN	PSNR (dB)	SSIM
input_90R-60-2-10	22,17	0,6139
input_90R-60-3-10	22,70	0,6562
input_90R-60-4-10	22,74	0,6673
input_90R-60-4-15	22,42	0,6654

IMAGEN	PSNR (dB)	SSIM
input_90R-65-2-10	22,55	0,6389
input_90R-65-3-10	22,83	0,6709
input_90R-65-4-10	22,77	0,6759
input_90R-65-3-15	22,39	0,6617
input_90R-65-4-15	22,24	0,6626

TABLA 33: Tablas del PSNR y SSIM de las imágenes resultantes del filtrado, con la versión 4, de la imagen de Lena con ruido gaussiano con desviación 90 para cada combinación de parámetros con h=60 (izquierda) y h=65 (derecha) empleando un Kernel gaussiano con desviación 5.

RUIDO 100

IMAGEN	PSNR (dB)	SSIM
input_100R-70-1-10	19,46	0,4619
input_100R-70-2-10	21,68	0,6043
input_100R-70-3-10	21,92	0,6393

IMAGEN	PSNR (dB)	SSIM
input_100R-75-1-10	20,00	0,4879
input_100R-75-2-10	21,81	0,6190
input_100R-75-3-10	21,91	0,6450

input_100R-70-4-10	21,87	0,6456
input_100R-70-3-15	21,32	0,6271
input_100R-70-4-15	21,44	0,6305

input_100R-75-4-10	21,81	0,6472

IMAGEN	PSNR (dB)	SSIM
input_100R-85-1-10	20,75	0,5310
input_100R-85-2-10	21,84	0,6337
input_100R-85-3-10	21,78	0,6457
input_100R-85-4-10	21,65	0,6431

TABLA 34: Tablas del PSNR y SSIM de las imágenes resultantes del filtrado, con la versión 4, de la imagen de Lena con ruido gaussiano con desviación 100 para cada combinación de parámetros con h=70 (arriba a la izquierda), h=75 (arriba a la derecha) y h=85 (abajo) empleando un Kernel gaussiano con desviación 5.

Kernel Gaussiano de desviación 3:

RUIDO 10

IMAGEN	PSNR (dB)	SSIM
input_10R-10-1-5	33,93	0,9576
input_10R-10-2-5	34,16	0,9617
input_10R-10-3-5	34,07	0,9618
input_10R-10-1-7	34,16	0,9569
input_10R-10-2-7	34,31	0,9598

IMAGEN	PSNR (dB)	SSIM
input_10R-20-1-5	34,63	0,9549
input_10R-20-2-5	34,43	0,9493
input_10R-20-3-5	34,30	0,9463
input_10R-20-1-7	34,38	0,9507
input_10R-20-2-7	34,02	0,9426

TABLA 35: Tablas del PSNR y SSIM de las imágenes resultantes del filtrado, con la versión 4, de la imagen de Lena con ruido gaussiano con desviación 10 para cada combinación de parámetros con h=10 (izquierda) y h=15 (derecha) empleando un Kernel gaussiano con desviación 3.

RUIDO 20

IMAGEN	PSNR (dB)	SSIM
input_20R-20-2-5	31,16	0,9201
input_20R-20-3-5	31,30	0,9220
input_20R-20-4-5	31,33	0,9222
input_20R-20-2-7	31,34	0,9179
input_20R-20-3-7	31,39	0,9170
input_20R-20-4-7	31,36	0,9158
input_20R-20-2-10	31,24	0,9117
input_20R-20-3-10	31,18	0,9081
input_20R-20-4-10	31,10	0,9053

IMAGEN	PSNR (dB)	SSIM
input_20R-30-2-5	30,95	0,9032
input_20R-30-3-5	30,64	0,8968
input_20R-30-4-5	30,46	0,8933
input_20R-30-2-7	30,46	0,8917
input_20R-30-3-7	30,04	0,8819
input_20R-30-4-7	29,81	0,8765
input_20R-30-2-10	29,91	0,8798
input_20R-30-3-10	29,42	0,8670
input_20R-30-4-10	29,14	0,8594

TABLA 36: Tablas del PSNR y SSIM de las imágenes resultantes del filtrado, con la versión 4, de la imagen de Lena con ruido gaussiano con desviación 20 para cada combinación de parámetros con h=20 (izquierda) y h=30 (derecha) empleando un Kernel gaussiano con desviación 3.

RUIDO 30

IMAGEN	PSNR (dB)	SSIM

IMAGEN	PSNR (dB)	SSIM

input_30R-30-2-5	29,21	0,8738
input_30R-30-3-5	29,46	0,8781
input_30R-30-4-5	29,51	0,8787
input_30R-30-2-7	29,39	0,8750
input_30R-30-3-7	29,47	0,8747
input_30R-30-4-7	29,42	0,8727
input_30R-30-2-10	29,25	0,8691
input_30R-30-3-10	29,18	0,8654
input_30R-30-4-10	29,04	0,8610

input_30R-40-2-5	29,30	0,8687
input_30R-40-3-5	29,04	0,8628
input_30R-40-4-5	28,85	0,8590
input_30R-40-2-7	28,92	0,8597
input_30R-40-3-7	28,54	0,8496
input_30R-40-4-7	28,29	0,8434
input_30R-40-2-10	28,46	0,8483
input_30R-40-3-10	27,99	0,8353
input_30R-40-4-10	27,70	0,8267

TABLA 37: Tablas del PSNR y SSIM de las imágenes resultantes del filtrado, con la versión 4, de la imagen de Lena con ruido gaussiano con desviación 30 para cada combinación de parámetros con h=30 (izquierda) y h=40 (derecha) empleando un Kernel gaussiano con desviación 3.

RUIDO 40

IMAGEN	PSNR (dB)	SSIM
input_40R-30-2-7	25,33	0,7819
input_40R-30-3-7	25,81	0,8039
input_40R-30-4-7	25,96	0,8119
input_40R-30-2-10	26,27	0,8024
input_40R-30-3-10	26,81	0,8241
input_40R-30-4-10	26,95	0,8307

IMAGEN	PSNR (dB)	SSIM
input_40R-40-2-7	27,83	0,8338
input_40R-40-3-7	27,90	0,8351
input_40R-40-4-7	27,81	0,8328
input_40R-40-2-10	27,65	0,8284
input_40R-40-3-10	27,57	0,8258
input_40R-40-4-10	27,39	0,8208

TABLA 38: Tablas del PSNR y SSIM de las imágenes resultantes del filtrado, con la versión 4, de la imagen de Lena con ruido gaussiano con desviación 40 para cada combinación de parámetros con h=30 (izquierda) y h=40 (derecha) empleando un Kernel gaussiano con desviación 3.

RUIDO 50

IMAGEN	PSNR (dB)	SSIM
input_50R-40-2-7	25,39	0,7601
input_50R-40-3-7	25,99	0,7824
input_50R-40-4-7	26,17	0,7896
input_50R-40-2-10	25,90	0,7720
input_50R-40-3-10	26,41	0,7924
input_50R-40-4-10	26,49	0,7964

IMAGEN	PSNR (dB)	SSIM
input_50R-50-2-7	26,61	0,7952
input_50R-50-3-7	26,66	0,7985
input_50R-50-4-7	26,56	0,7965
input_50R-50-2-10	26,39	0,7894
input_50R-50-3-10	26,30	0,7893
input_50R-50-4-10	26,11	0,7841

TABLA 39: Tablas del PSNR y SSIM de las imágenes resultantes del filtrado, con la versión 4, de la imagen de Lena con ruido gaussiano con desviación 50 para cada combinación de parámetros con h=40 (izquierda) y h=50 (derecha) empleando un Kernel gaussiano con desviación 3.

RUIDO 60

IMAGEN	PSNR (dB)	SSIM
input_60R-50-2-7	24,88	0,7311
input_60R-50-3-7	25,35	0,7512
input_60R-50-4-7	25,44	0,7561
input_60R-50-2-10	25,08	0,7380

IMAGEN	PSNR (dB)	SSIM
input_60R-60-2-7	25,50	0,7558
input_60R-60-3-7	25,52	0,7602
input_60R-60-4-7	25,41	0,7582
input_60R-60-2-10	25,21	0,7495

input_60R-50-3-10	25,39	0,7555
input_60R-50-4-10	25,36	0,7572

input_60R-60-3-10	25,11	0,7505
input_60R-60-4-10	24,93	0,7455

TABLA 40: Tablas del PSNR y SSIM de las imágenes resultantes del filtrado, con la versión 4, de la imagen de Lena con ruido gaussiano con desviación 60 para cada combinación de parámetros con h=50 (izquierda) y h=60 (derecha) empleando un Kernel gaussiano con desviación 3.

RUIDO 70

IMAGEN	PSNR (dB)	SSIM
input_70R-55-2-7	23,78	0,6850
input_70R-55-3-7	24,33	0,7122
input_70R-55-4-7	24,44	0,7197
input_70R-55-2-10	24,05	0,6954
input_70R-55-3-10	24,45	0,7217
input_70R-55-4-10	24,45	0,7263

IMAGEN	PSNR (dB)	SSIM
input_70R-65-2-7	24,56	0,7206
input_70R-65-3-7	24,69	0,7319
input_70R-65-4-7	24,62	0,7324
input_70R-65-2-10	24,32	0,7164
input_70R-65-3-10	24,32	0,7247
input_70R-65-4-10	24,17	0,7219

TABLA 41: Tablas del PSNR y SSIM de las imágenes resultantes del filtrado, con la versión 4, de la imagen de Lena con ruido gaussiano con desviación 70 para cada combinación de parámetros con h=55 (izquierda) y h=65 (derecha) empleando un Kernel gaussiano con desviación 3.

RUIDO 80

IMAGEN	PSNR (dB)	SSIM
input_80R-60-3-7	23,33	0,6740
input_80R-60-4-7	23,46	0,6831
input_80R-60-3-10	23,44	0,6858
input_80R-60-4-10	23,46	0,6925

IMAGEN	PSNR (dB)	SSIM
input_80R-70-3-7	23,72	0,6987
input_80R-70-4-7	23,70	0,7011
input_80R-70-3-10	23,39	0,6940
input_80R-70-4-10	23,29	0,6935

TABLA 42: Tablas del PSNR y SSIM de las imágenes resultantes del filtrado, con la versión 4, de la imagen de Lena con ruido gaussiano con desviación 80 para cada combinación de parámetros con h=60 (izquierda) y h=70 (derecha) empleando un Kernel gaussiano con desviación 3.

RUIDO 90

IMAGEN	PSNR (dB)	SSIM
input_90R-65-3-7	22,69	0,6489
input_90R-65-4-7	22,81	0,6600
input_90R-65-3-10	22,76	0,6605
input_90R-65-4-10	22,79	0,6693

IMAGEN	PSNR (dB)	SSIM
input_90R-75-3-7	22,96	0,6674
input_90R-75-4-7	23,00	0,6749
input_90R-75-3-10	22,80	0,6693
input_90R-75-4-10	22,75	0,6739

TABLA 43: Tablas del PSNR y SSIM de las imágenes resultantes del filtrado, con la versión 4, de la imagen de Lena con ruido gaussiano con desviación 90 para cada combinación de parámetros con h=65 (izquierda) y h=75 (derecha) empleando un Kernel gaussiano con desviación 3.

RUIDO 100

IMAGEN	PSNR (dB)	SSIM
input_100R-70-3-7	21,87	0,6176
input_100R-70-4-7	21,98	0,6293
input_100R-70-3-10	21,87	0,6280
input_100R-70-4-10	21,89	0,6376

IMAGEN	PSNR (dB)	SSIM
input_100R-80-3-7	22,19	0,6458
input_100R-80-4-7	22,19	0,6515
input_100R-80-3-10	21,87	0,6414
input_100R-80-4-10	21,81	0,6445

TABLA 44: Tablas del PSNR y SSIM de las imágenes resultantes del filtrado, con la versión 4, de la imagen de Lena con ruido gaussiano con desviación 100 para cada combinación de parámetros con h=70 (izquierda) y h=80 (derecha) empleando un Kernel gaussiano con desviación 3.

Kernel Gaussiano de desviación 7:

RUIDO 10

IMAGEN	PSNR (dB)	SSIM
input_10R-10-1-5	34,00	0,9580
input_10R-10-2-5	34,20	0,9616
input_10R-10-3-5	34,08	0,9611
input_10R-10-1-7	34,22	0,9572
input_10R-10-2-7	34,33	0,9594
input_10R-10-3-7	34,13	0,9575

IMAGEN	PSNR (dB)	SSIM
input_10R-15-1-5	34,62	0,9546
input_10R-15-2-5	34,34	0,9481
input_10R-15-3-5	34,14	0,9441
input_10R-15-1-7	34,35	0,9503
input_10R-15-2-7	33,90	0,9410
input_10R-15-3-7	33,62	0,9346

TABLA 45: Tablas del PSNR y SSIM de las imágenes resultantes del filtrado, con la versión 4, de la imagen de Lena con ruido gaussiano con desviación 10 para cada combinación de parámetros con h=10 (izquierda) y h=15 (derecha) empleando un Kernel gaussiano con desviación 7.

RUIDO 20

IMAGEN	PSNR (dB)	SSIM
input_20R-15-2-5	27,83	0,8840
input_20R-15-3-5	27,79	0,8919
input_20R-15-4-5	27,70	0,8946
input_20R-15-2-7	28,76	0,9033
input_20R-15-3-7	28,70	0,9116
input_20R-15-4-7	28,55	0,9136
input_20R-15-2-10	29,51	0,9115
input_20R-15-3-10	29,39	0,9183
input_20R-15-4-10	29,18	0,9191

IMAGEN	PSNR (dB)	SSIM
input_20R-20-2-5	31,25	0,9207
input_20R-20-3-5	31,35	0,9214
input_20R-20-4-5	31,33	0,9206
input_20R-20-2-7	31,39	0,9177
input_20R-20-3-7	31,35	0,9147
input_20R-20-4-7	31,24	0,9117
input_20R-20-2-10	31,24	0,9108
input_20R-20-3-10	31,06	0,9042
input_20R-20-4-10	30,87	0,8987

TABLA 46: Tablas del PSNR y SSIM de las imágenes resultantes del filtrado, con la versión 4, de la imagen de Lena con ruido gaussiano con desviación 20 para cada combinación de parámetros con h=15 (izquierda) y h=20 (derecha) empleando un Kernel gaussiano con desviación 7.

RUIDO 30

IMAGEN	PSNR (dB)	SSIM
input_30R-25-2-5	27,56	0,8531
input_30R-25-3-5	27,93	0,8664
input_30R-25-4-5	28,05	0,8713
input_30R-25-2-7	28,46	0,8696
input_30R-25-3-7	28,77	0,8788
input_30R-25-4-7	28,81	0,8806
input_30R-25-2-10	28,93	0,8723
input_30R-25-3-10	29,13	0,8768
input_30R-25-4-10	29,06	0,8746

IMAGEN	PSNR (dB)	SSIM
input_30R-30-2-5	29,33	0,8757
input_30R-30-3-5	29,50	0,8779
input_30R-30-4-5	29,47	0,8768
input_30R-30-2-7	29,45	0,8756
input_30R-30-3-7	29,40	0,8720
input_30R-30-4-7	29,22	0,8670
input_30R-30-2-10	29,26	0,8691
input_30R-30-3-10	29,01	0,8607
input_30R-30-4-10	28,72	0,8519

TABLA 47: Tablas del PSNR y SSIM de las imágenes resultantes del filtrado, con la versión 4, de la imagen de Lena con ruido gaussiano con desviación 30 para cada combinación de parámetros con h=25 (izquierda) y h=30 (derecha) empleando un Kernel gaussiano con desviación 7.

RUIDO 40

IMAGEN	PSNR (dB)	SSIM
input_40R-30-2-7	25,59	0,7904
input_40R-30-3-7	26,09	0,8127
input_40R-30-4-7	26,24	0,8207
input_40R-30-2-10	26,53	0,8105
input_40R-30-3-10	27,04	0,8308
input_40R-30-4-10	27,14	0,8358

IMAGEN	PSNR (dB)	SSIM
input_40R-35-2-7	27,50	0,8303
input_40R-35-3-7	27,82	0,8385
input_40R-35-4-7	27,81	0,8379
input_40R-35-2-10	27,66	0,8311
input_40R-35-3-10	27,77	0,8334
input_40R-35-4-10	27,59	0,8280

TABLA 48: Tablas del PSNR y SSIM de las imágenes resultantes del filtrado, con la versión 4, de la imagen de Lena con ruido gaussiano con desviación 40 para cada combinación de parámetros con h=30 (izquierda) y h=35 (derecha) empleando un Kernel gaussiano con desviación 7.

RUIDO 50

IMAGEN	PSNR (dB)	SSIM
input_50R-40-2-7	25,70	0,7707
input_50R-40-3-7	26,25	0,7911
input_50R-40-4-7	26,36	0,7960
input_50R-40-2-10	26,16	0,7820
input_50R-40-3-10	26,54	0,7981
input_50R-40-4-10	26,48	0,7973

IMAGEN	PSNR (dB)	SSIM
input_50R-50-2-7	26,69	0,7983
input_50R-50-3-7	26,59	0,7971
input_50R-50-4-7	26,33	0,7907
input_50R-50-2-10	26,42	0,7918
input_50R-50-3-10	26,16	0,7857
input_50R-50-4-10	25,80	0,7744

TABLA 49: Tablas del PSNR y SSIM de las imágenes resultantes del filtrado, con la versión 4, de la imagen de Lena con ruido gaussiano con desviación 50 para cada combinación de parámetros con h=40 (izquierda) y h=50 (derecha) empleando un Kernel gaussiano con desviación 7.

RUIDO 60

IMAGEN	PSNR (dB)	SSIM
input_60R-40-2-7	22,07	0,6387
input_60R-40-3-7	22,66	0,6670
input_60R-40-4-7	22,88	0,6774
input_60R-40-2-10	23,38	0,6797
input_60R-40-3-10	24,14	0,7163
input_60R-40-4-10	24,36	0,7280

IMAGEN	PSNR (dB)	SSIM
input_60R-50-2-7	25,12	0,7409
input_60R-50-3-7	25,49	0,7578
input_60R-50-4-7	25,45	0,7593
input_60R-50-2-10	25,27	0,7474
input_60R-50-3-10	25,42	0,7594
input_60R-50-4-10	25,22	0,7552

TABLA 50: Tablas del PSNR y SSIM de las imágenes resultantes del filtrado, con la versión 4, de la imagen de Lena con ruido gaussiano con desviación 60 para cada combinación de parámetros con h=40 (izquierda) y h=50 (derecha) empleando un Kernel gaussiano con desviación 7.

RUIDO 70

IMAGEN	PSNR (dB)	SSIM
input_70R-55-2-7	24,06	0,6978
input_70R-55-3-7	24,51	0,7225

IMAGEN	PSNR (dB)	SSIM
input_70R-45-2-7	21,16	0,5906
input_70R-45-3-7	21,77	0,6217

input_70R-55-4-7	24,52	0,7271
input_70R-55-2-10	24,28	0,7084
input_70R-55-3-10	24,52	0,7297
input_70R-55-4-10	24,37	0,7284

input_70R-45-4-7	22,00	0,6330
input_70R-45-2-10	22,48	0,6335
input_70R-45-3-10	23,28	0,6757
input_70R-45-4-10	23,50	0,6891

TABLA 51: Tablas del PSNR y SSIM de las imágenes resultantes del filtrado, con la versión 4, de la imagen de Lena con ruido gaussiano con desviación 70 para cada combinación de parámetros con h=55 (izquierda) y h=45 (derecha) empleando un Kernel gaussiano con desviación 7.

RUIDO 80

IMAGEN	PSNR (dB)	SSIM
input_80R-55-3-7	22,83	0,6527
input_80R-55-4-7	23,03	0,6646
input_80R-55-3-10	23,34	0,6800
input_80R-55-4-10	23,39	0,6894

IMAGEN	PSNR (dB)	SSIM
input_80R-60-3-7	23,52	0,6863
input_80R-60-4-7	23,56	0,6930
input_80R-60-3-10	23,52	0,6963
input_80R-60-4-10	23,41	0,6977

TABLA 52: Tablas del PSNR y SSIM de las imágenes resultantes del filtrado, con la versión 4, de la imagen de Lena con ruido gaussiano con desviación 80 para cada combinación de parámetros con h=55 (izquierda) y h=60 (derecha) empleando un Kernel gaussiano con desviación 7.

RUIDO 90

IMAGEN	PSNR (dB)	SSIM
input_90R-60-3-7	22,35	0,6335
input_90R-60-4-7	22,52	0,6470
input_90R-60-3-10	22,73	0,6596
input_90R-60-4-10	22,74	0,6694

IMAGEN	PSNR (dB)	SSIM
input_90R-70-3-7	23,04	0,6771
input_90R-70-4-7	23,00	0,6817
input_90R-70-3-10	22,80	0,6770
input_90R-70-4-10	22,65	0,6759

TABLA 53: Tablas del PSNR y SSIM de las imágenes resultantes del filtrado, con la versión 4, de la imagen de Lena con ruido gaussiano con desviación 90 para cada combinación de parámetros con h=60 (izquierda) y h=70 (derecha) empleando un Kernel gaussiano con desviación 7.

RUIDO 100

IMAGEN	PSNR (dB)	SSIM
input_100R-65-3-7	21,65	0,6063
input_100R-65-4-7	21,78	0,6201
input_100R-65-3-10	21,86	0,6292
input_100R-65-4-10	21,86	0,6396

IMAGEN	PSNR (dB)	SSIM
input_100R-75-3-7	22,17	0,6466
input_100R-75-4-7	22,14	0,6524
input_100R-75-3-10	21,90	0,6465
input_100R-75-4-10	21,79	0,6474

TABLA 54: Tablas del PSNR y SSIM de las imágenes resultantes del filtrado, con la versión 4, de la imagen de Lena con ruido gaussiano con desviación 100 para cada combinación de parámetros con h=65 (izquierda) y h=75 (derecha) empleando un Kernel gaussiano con desviación 7.

Selección del Kernel gaussiano:

A continuación se muestran las combinaciones de parámetros con las que se obtienen los mejores resultados con cada Kernel gaussiano, para cada imagen de Lena con ruido gaussiano empleada:

DESVIACION 1				
ruido	h	vecindad	radio	SSIM

DESVIACION 3				
ruido	h	vecindad	radio	SSIM

10	15	1	5	0,9564
20	30	2	5	0,9110
30	50	2	5	0,8629
40	60	3	7	0,8127
50	85	4	7	0,7717
60	110	4	7	0,7320
70	120	4	7	0,7052
80	140	4	7	0,6780
90	135	4	7	0,6540
100	175	4	7	0,6394

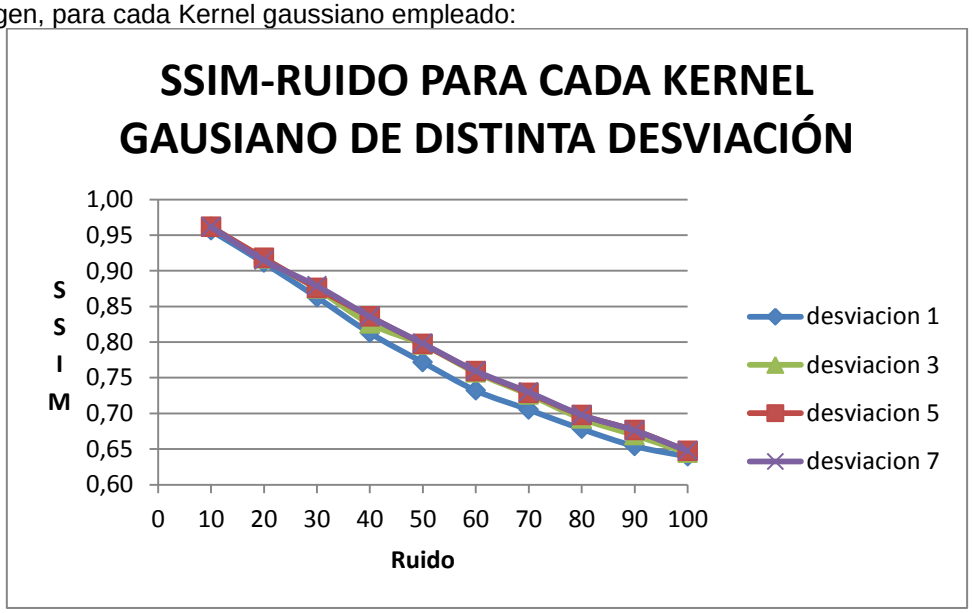
10	10	2	5	0,9617
20	20	3	7	0,9170
30	30	3	7	0,8747
40	40	3	10	0,8258
50	40	4	10	0,7964
60	50	4	10	0,7572
70	55	4	10	0,7263
80	60	4	10	0,6925
90	65	4	10	0,6693
100	80	4	10	0,6445

DESVIACION 5				
ruido	h	vecindad	radio	SSIM
10	10	2	5	0,9616
20	20	2	7	0,9178
30	30	2	7	0,8756
40	30	4	10	0,8358
50	40	3	10	0,7972
60	50	3	10	0,7591
70	55	3	10	0,7284
80	60	4	10	0,6975
90	70	4	10	0,6761
100	75	4	10	0,6472

DESVIACION 7				
ruido	h	vecindad	radio	SSIM
10	10	2	5	0,9616
20	20	3	7	0,9147
30	25	3	7	0,8788
40	30	4	10	0,8358
50	40	3	10	0,7981
60	50	3	10	0,7594
70	55	3	10	0,7297
80	60	4	10	0,6977
90	70	4	10	0,6759
100	75	4	10	0,6474

TABLA 55: Tablas de los mejores resultados para cada Kernel gaussiano empleado.

La siguiente grafica representa el SSIM obtenido en función de la desviación del ruido que tiene la imagen, para cada Kernel gaussiano empleado:



GRÁFICA 4: Comparación del SSIM las imágenes obtenidas tras aplicar la mejor combinación de parámetros a la imagen de Lena con ruido gaussiano de desviación 10, 20, 30, 40, 50, 60, 70, 80, 90 y 100, empleando la versión 4 y kernels gaussianos de desviación 1,3,5 y 7.

Las siguientes imágenes muestran los resultados obtenidos para la imagen de Lena con ruido gaussiano de desviación 50 empleando cada uno de los kernels gaussianos:

Imagen con ruido gaussiano desviación 50. SSIM=0,3651

Desviacion kernel=1

h=85,v=4,r=7

SSIM= 0,7717

Desviacion kernel=3

h=40,v=4,r=10

SSIM= 0,7964

Desviacion kernel=5

h=40,v=3,r=10

SSIM= 0,7972

Desviacion kernel=7

h=40,v=3,r=10

SSIM= 0,7981

Se observa que los kernels con desviación 5 y 7 son los que mejor resultados dan. Sería difícil elegir decidir cuál funciona mejor. Si se aumentaría más la desviación del Kernel, realmente para las vecindades que se usan no habría mucho cambio en los resultados. Al final la ponderación se estaría haciendo con valores semejantes si aumentaríamos la desviación, ya que variaría en cifras decimales pequeñas. Sin embargo si se precisara de vecindades aún mayores, habría que considerar el aumentar la desviación del Kernel para así dar más peso a aquellos píxeles más lejanos de la vecindad (y no darle un valor próximo a 0). Decir que aumentar mucho la desviación lo único que se conseguiría sería ponderar con el mismo valor (o valores muy parecidos) a todos los píxeles del vecindario.
Finalmente el Kernel elegido es el de desviación 5, por dar buenos resultados y además ponderar con valores algo más altos los píxeles centrales del vecindario que el Kernel de desviación 7.

Análisis de los resultados:

A continuación se muestra la tabla resumen de parámetros y los parámetros exactos para cada ruido:

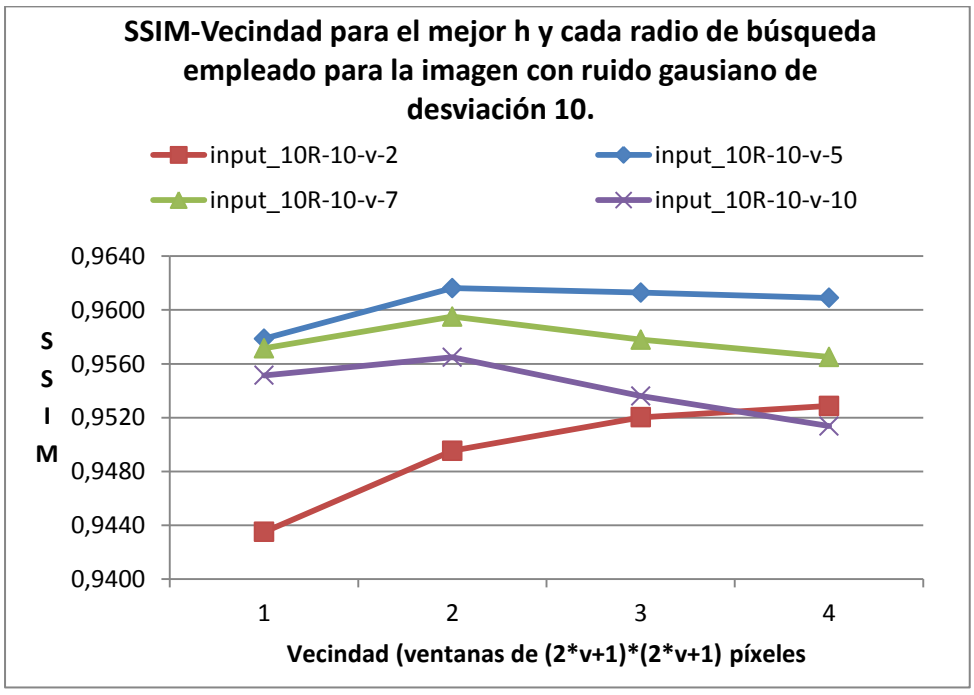
	Kernel desviación 5		
	h	v	r
0<ruido<=15	ruido	2	5
15<ruido<=30	ruido	2	7
30<ruido<=45	0,75*ruido	4	10
45<ruido<=75	0,8*ruido	3	10
75<ruido<=100	0,75*ruido	4	10

TABLA 56: Tabla resumen de parámetros en función del ruido para la versión 4.

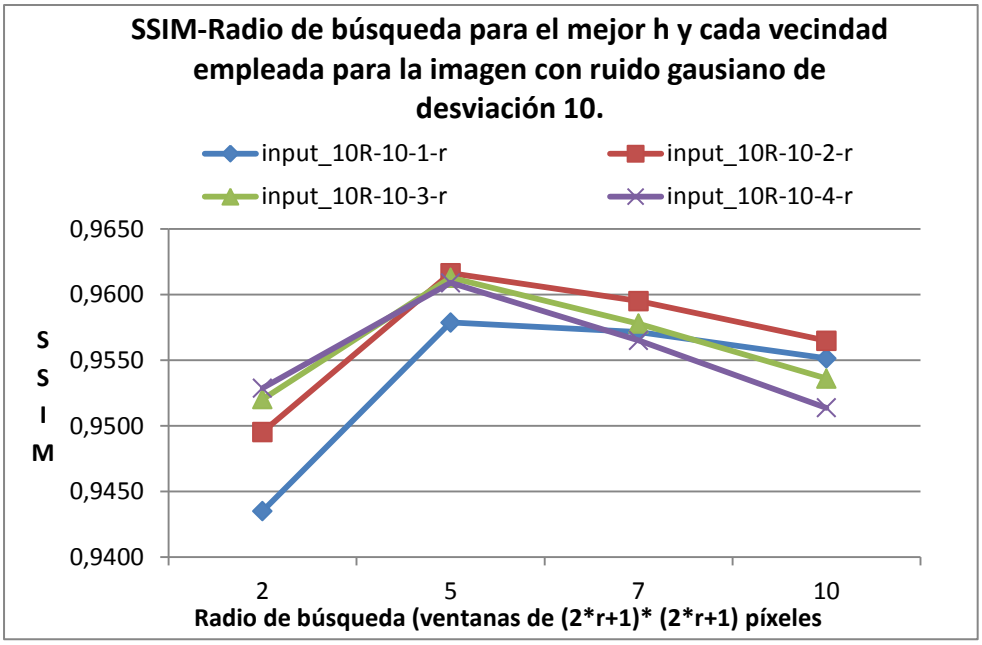
ruido	h	vecindad	radio
10	10	2	5
20	20	2	7
30	30	2	7
40	30	4	10
50	40	3	10
60	48	3	10
70	56	3	10
80	60	4	10
90	67,5	4	10
100	75	4	10

TABLA 57: Tabla de parámetros exactos para ruidos con desviación 10, 20, 30, 40, 50, 60, 70, 80, 90 y 100 para la versión 4.

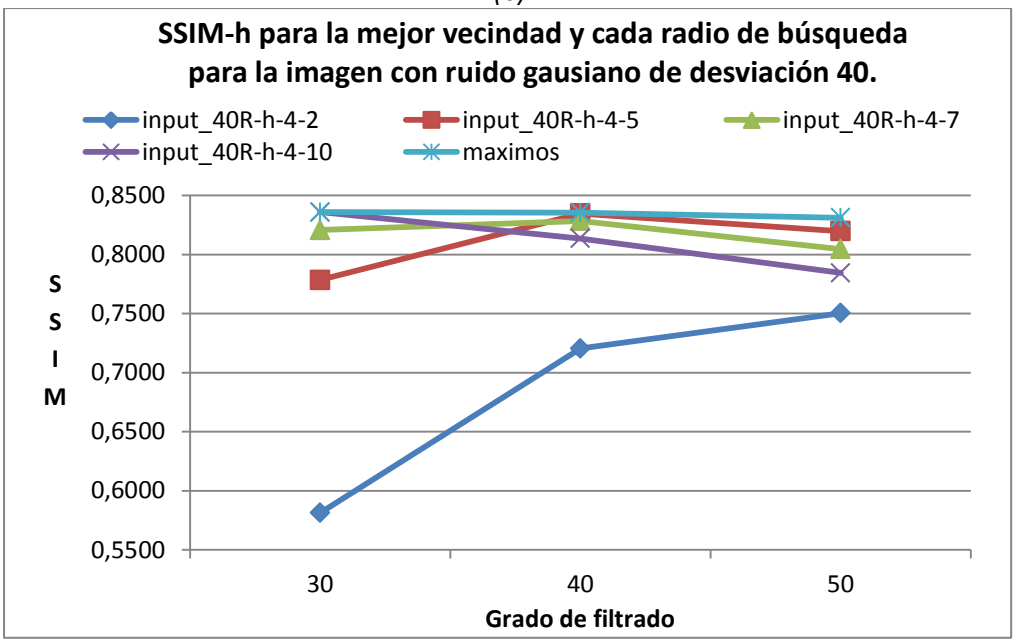
Al igual que hemos hecho con la versión 3, a continuación se muestran diferentes gráficas que muestran la evolución de la calidad de las imágenes obtenidas dependiendo del valor de los parámetros estudiados.



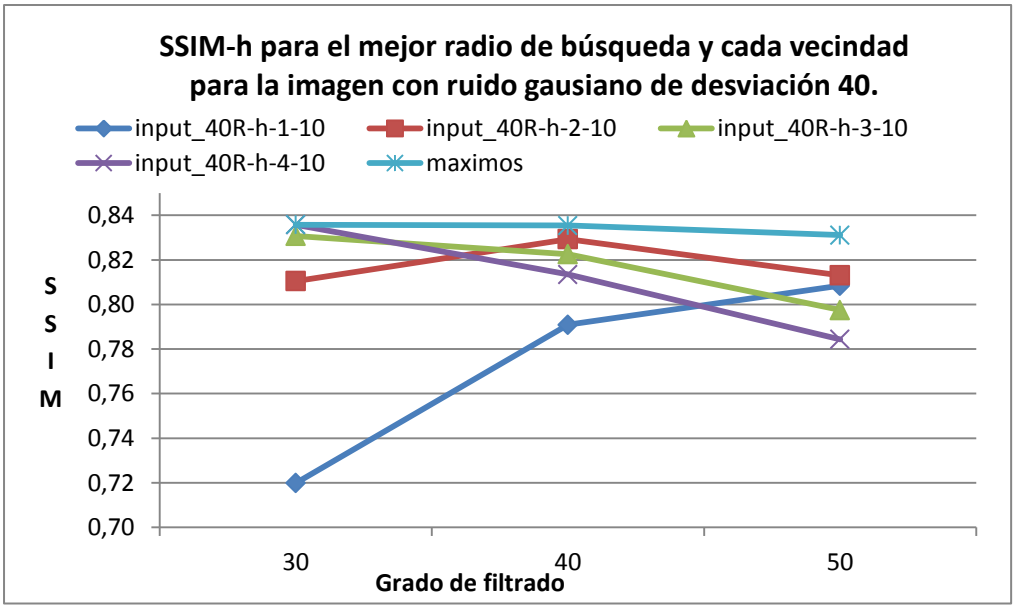
(a)


SSIM-Radio de búsqueda para el mejor h y cada vecindad empleada para la imagen con ruido gausiano de desviación 10.
input_10R-10-1-r
input_10R-10-2-r
input_10R-10-3-r
input_10R-10-4-r
0,9650
0,9600
0,9550
0,9500
0,9450
0,9400
SSIM
2
5
7
10
Radio de búsqueda (ventanas de (2*r+1)* (2*r+1) píxeles

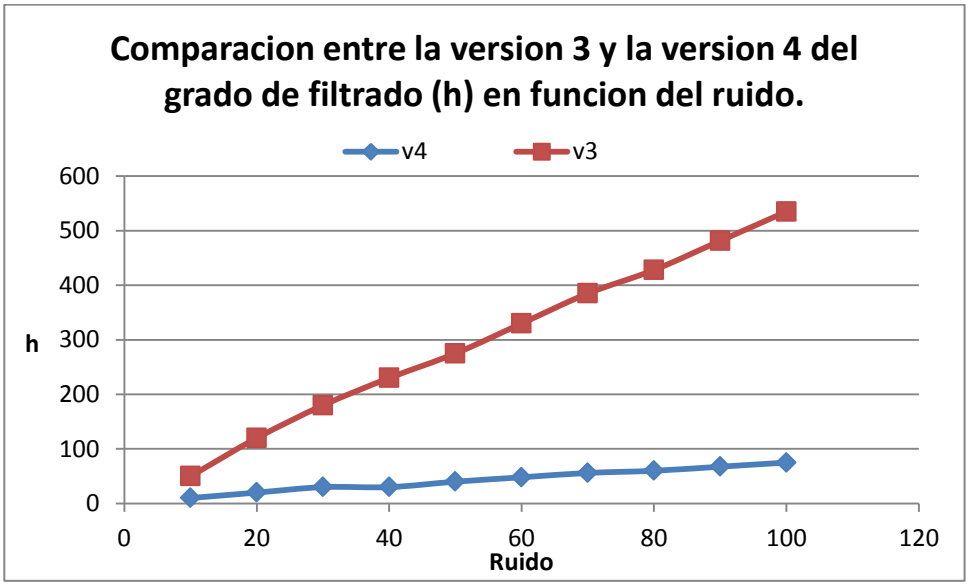
(b)


SSIM-h para la mejor vecindad y cada radio de búsqueda para la imagen con ruido gausiano de desviación 40.
input_40R-h-4-2
input_40R-h-4-5
input_40R-h-4-7
input_40R-h-4-10
maximos
0,8500
0,8000
0,7500
0,7000
0,6500
0,6000
0,5500
SSIM
30
40
50
Grado de filtrado

(c)



(d)



(e)

GRÁFICA 5: Gráficas para la versión 4: ***a)*** *SSIM-Vecindad para el mejor h y cada radio de búsqueda empleado para la imagen con ruido gaussiano de desviación 10.* ***b)*** *SSIM-Radio de búsqueda para el mejor h y cada vecindad empleada para la imagen con ruido gaussiano de desviación 10.* ***c)*** *SSIM-h para la mejor vecindad y cada radio de búsqueda para la imagen con ruido gaussiano de desviación 40.* ***d)*** *SSIM-h para el mejor radio de búsqueda y cada vecindad para la imagen con ruido gaussiano de desviación 40.* ***e)*** *Comparación entre la versión 3 y la versión 4 del grado de filtrado (h) en función del ruido.*

La siguiente sucesión de imágenes es un ejemplo para la gráfica a), tomando como imagen la imagen de Lena con ruido gaussiano de desviación 50:

Imagen con ruido gaussiano desviación 50. SSIM=0,3651

v=1, SSIM=0,6803 *v=2, SSIM=0,7801* ***v=3, SSIM=0,7972*** *v=4, SSIM=0,7970*

Sucesion de imágenes para cada vecindad (v), para el mejor h (h=40), para radio de búsqueda = 10 (ventana de 21x21) y kernel gausiano de desviacion 5, para la imagen de lena con ruido gausiano de desviación 50.

La siguiente sucesión de imágenes es un ejemplo para la gráfica b), tomando como imagen la imagen de Lena con ruido gaussiano de desviación 50:

r=2, SSIM= 0,5598 *r=5, SSIM= 0,7663* *r=7, SSIM= 0,7952* ***r=10, SSIM= 0,7970***

Sucesion de imágenes para vecindad=4 (ventana de 9x9), para el mejor h (h=40), para cada radio de búsqueda (r) y kernel gausiano de desviacion 5, para la imagen de lena con ruido gausiano de desviación 50.

La siguiente sucesión de imágenes es un ejemplo para la gráfica c) y d), tomando como imagen la imagen de Lena con ruido gaussiano de desviación 40:

Imagen con ruido gaussiano desviación 40. SSIM= 0,4361

h=30, SSIM=0,8358 *h=40, SSIM=0,8134* *h=50, SSIM=0,7844*

Sucesion de imágenes para la mejor vecindad =4 (ventana de 9x9), para h=30,40,50 y para el mejor radio de búsqueda = 10 (ventana de 21x21) y kernel gausiano de desviacion 5, para la imagen de lena con ruido gausiano de desviación 40.

A continuación se analizan los resultados:

Cabe destacar que quizás se observen combinaciones de parámetros cuyo valor SSIM es mejor que el de la combinación escogida. Si no se escoge ese resultado es porque realmente al visualizar la imagen, la calidad no es la esperada y es peor que la de la combinación escogida. El tiempo de ejecución también es decisivo a la hora de escoger una combinación de parámetros, ya que es preferible por ejemplo escoger una imagen con un peor SSIM si su tiempo de ejecución es mucho menor que otra con más SSIM.

Al igual que en la v3, para una determinada vecindad, tras alcanzar el máximo SSIM, este empeora a medida que aumenta el radio. Este empeoramiento es mayor cuanto mayor es la vecindad. El máximo SSIM se alcanza en función del ruido que tenga la imagen. Cuanto más ruido tenga, mayor es la vecindad en la que se alcanza. Se observa que para un radio determinado, tras alcanzar el máximo SSIM, el decrecimiento que se produce a medida que aumenta la vecindad es menor que en la versión 3. Esto indica que las distintas vecindades funcionan mejor que en la versión 3 y que por lo tanto el resultado del filtrado no esta tan influenciado por el tamaño de la vecindad. Cuanto mayor es el valor del grado de filtrado más ruido se filtra y más detalles se pierden de la imagen.

La principal diferencia que se observa con respecto a la versión 3, es que el valor necesario del grado de filtrado es bastante inferior. Esto es debido a que al ponderar los píxeles cuando se calcula la distancia euclídea ponderada, obtenemos valores menores que en la versión 3, por lo que el valor del grado de filtrado, que es el que divide esto, también tendrá que ser más pequeño. Esto nos permite aumentar el rango de valores de grado de filtrado para los que funciona el algoritmo, siendo estos más pequeños que en la versión 3. Es decir, es más fácil encontrar un valor del grado de filtrado que reduzca algo de ruido, a diferencia de la v3, en la que los valores del grado de filtrado empezaban a funcionar cuando eran más altos. También decir que funciona en general mejor que la versión 3. En la v3 había combinaciones de parámetros que directamente no eliminaban ruido. Esto sucedía a medida que se aumentaba la vecindad cuando la cantidad de ruido era baja por ejemplo. En la versión 4 se consigue reducir ruido trabajando con cualquier vecindad (con un valor más o menos adecuado del grado de filtrado). Por tanto emplear esta versión puede darnos más posibilidades a la hora de elegir una combinación de parámetros que más o menos funcione, ya que aun siendo mala esta combinación, funcionará mejor que una combinación mala de parámetros de la versión 3.

En los resultados finales para la imagen Lena (se muestran a continuación), con la versión 4, se obtienen valores del SSIM algo más bajos que con la versión 3. Sin embargo, el hecho de que la versión 4 funciona casi igual o mejor que la versión 3 con distintas combinaciones de parámetros (para los cuales la versión 3 no funciona), se trabajará con la versión 4 a partir de ahora.

Las siguientes imágenes se corresponden con las de Lena con ruido gaussiano y con la mejor imagen obtenida tras aplicar los parámetros exactos correspondientes, tanto para la versión 3 como para la versión 4 (con un Kernel gaussiano de desviación 5):

Imagen con ruido gaussiano	Imagen Filtrada v3	Imagen Filtrada v4
		
Desviacion=10, SSIM=0,8708	*h=50,v=2,r=5,SSIM=0,9615*	*h=10,v=2,r=5,SSIM=0,9616*
		
Desviacion=20, SSIM=0,6788	*h=120,v=3,r=7,SSIM=0,9220*	*h=20,v=2r=7,SSIM=0,9178*
		
Desviacion=30, SSIM=0,5352	*h=180,v=3,r=7,SSIM=0,8798*	*h=30,v=2,r=7,SSIM=0,8756*

Desviacion=40, SSIM=0,4361

h=230,v=3,r=10,SSIM=0,8364

h=30,v=4,r=10,SSIM=0,8358

Desviacion=50, SSIM=0,3651

h=275,v=3,r=10,SSIM=0,7981

h=40,v=3,r=10,SSIM=0,7972

Desviacion=60, SSIM=0,3142

h=330,v=3,r=10,SSIM=0,7594

h=48,v=3,r=10,SSIM=0,7583

Desviacion=70, SSIM=0,2724

h=385,v=3,r=10,SSIM=0,7307

h=56,v=3,r=10,SSIM=0,7292

Desviacion=80, SSIM=0,2412

h=428,v=3,r=10,SSIM=0,6987

h=60,v=4,r=10,SSIM=0,6975

Desviacion=90, SSIM=0,2159

h=481,v=3,r=10,SSIM=0,6777

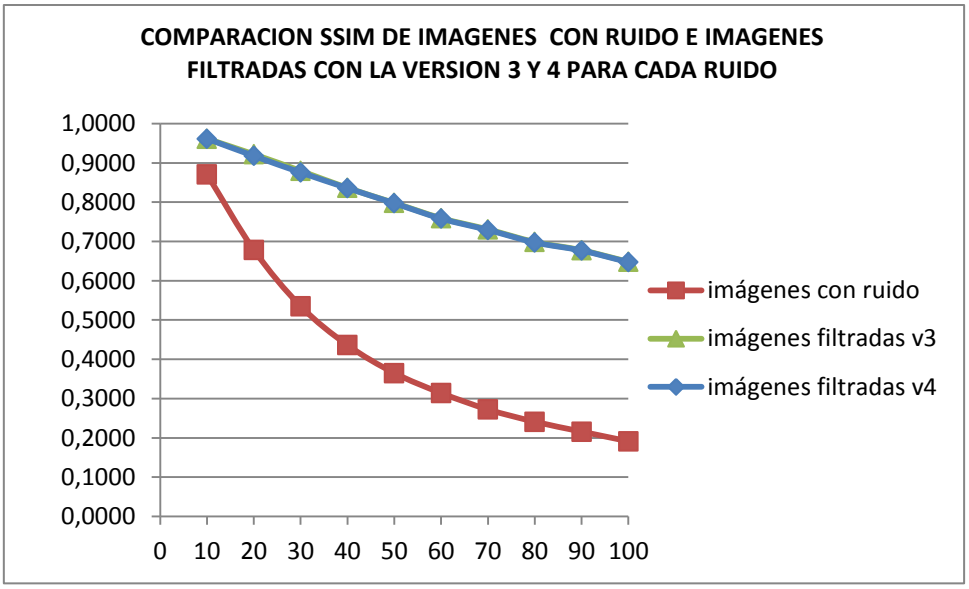
h=68,v=4,r=10,SSIM=0,6767

Desviacion=100, SSIM=0,1913

h=535,v=3,r=10,SSIM=0,6481

h=75,v=4,r=10,SSIM=0,6472

La siguiente grafica muestra los datos anteriores:



GRÁFICA 6: Comparación del SSIM entre las imágenes con ruido gaussiano de desviación 10, 20, 30, 40, 50, 60, 70, 80, 90 y 100, y las imágenes obtenidas tras aplicar la mejor combinación de parámetros de la versión 3 y de la versión 4 a cada una.

4.2.2.2. Aplicación de los resultados a imágenes en escala de gris con ruido gaussiano y rician

Aplicación de resultados a imágenes con ruido gaussiano:

A continuación vamos a aplicar el filtro Non Local Means con los parámetros que hemos considerado la mejor combinación en el estudio anterior a varias imágenes con ruido gaussiano:

Imagen Original: mandril.png

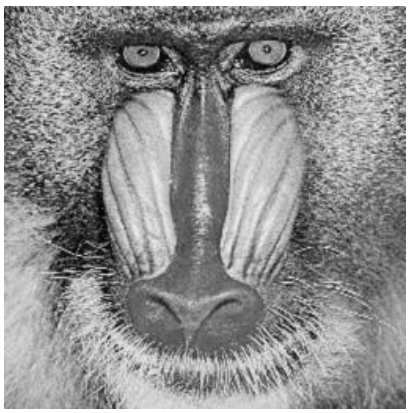
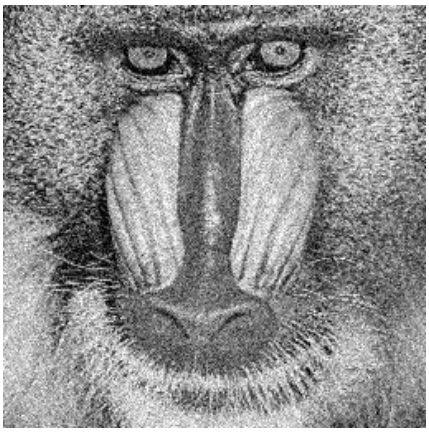
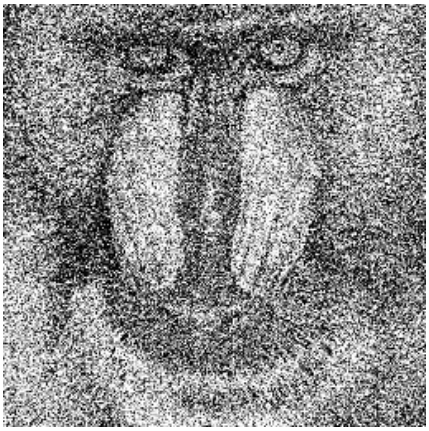


Imagen con ruido gaussiano Imagen filtrada



Desviación=25

h=20,v=2,r=5,SSIM= 0,7151



Desviación=85

h=40,v=3,r=10,SSIM= 0,2802

Imagen Original: avion.png

Imagen con ruido gaussiano

Imagen filtrada

Desviación=20 *h=20,v=2,r=7,SSIM= 0,8393*

Desviación=40 *h=30,v=4,r=10,SSIM= 0,7091*

Aplicación de resultados a imágenes con ruido rician:

A continuación se muestran los resultados para la imagen Lena256:

IMAGEN	PSNR	SSIM
input_10R-10-2-5	32,51	0,8970
input_20R-20-2-7	29,02	0,8182
input_30R-30-2-7	26,20	0,7394
input_40R-35-3-10	23,49	0,6735
input_50R-40-3-10	21,29	0,6011
input_60R-48-3-10	19,22	0,5458
input_70R-56-3-10	17,42	0,5019
input_80R-60-4-10	15,95	0,4818
input_90R-68-4-10	14,63	0,4549
input_100R-75-4-10	13,64	0,4401

TABLA 58: PSNR y SSIM para la imagen lena256 tras aplicar los parámetros exactos a cada imagen con ruido.

A continuación se muestran varias imágenes:

Imagen Original: lena256.png

Imagen con ruido rician Imagen filtrada

Desviación=20

h=20,v=2,r=7,SSIM= 0,8182

Desviación=70

h=56,v=3,r=10,SSIM=0,5019

Imagen Original: avion.png

Imagen con ruido rician

Imagen filtrada

Desviación=15

h=15,v=2,r=5,SSIM= 0,8663

Desviación=30 *h=30,v=2,r=7,SSIM= 0,7738*

Imagen Original: radiografia.png

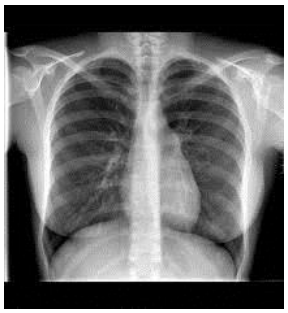
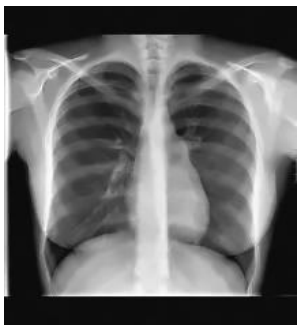
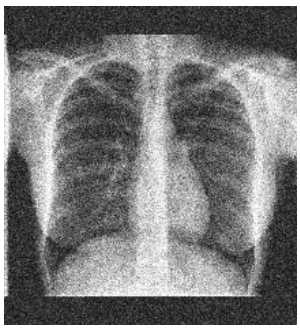


Imagen con ruido rician Imagen filtrada



Desviación=10 *h=10,v=2,r=5,SSIM= 0,7963*



Desviación=40 *h=30,v=4,r=10,SSIM= 0,5652*

Sucede lo mismo que con la versión 3. Los resultados son algo peores, debido a que el ruido rician no tiene el mismo comportamiento que el gaussiano. Para las imágenes en las que se pierde contraste, bastará con realzarlo.

4.3. Adaptación a imágenes a color

Para este apartado se ha utilizado la versión4 con un Kernel gaussiano de desviación 5, y la hemos adaptado para trabajar con a imágenes a color. En primer lugar se va a realizar un estudio del tiempo de ejecución según el radio de búsqueda, similar al realizado en el apartado 1 de las pruebas. A continuación se realizará el mismo procedimiento para el estudio de los parámetros que se ha realizado en el punto 2, esta vez para la versión ampliada a imágenes a color.

4.3.1. Tiempo de ejecución

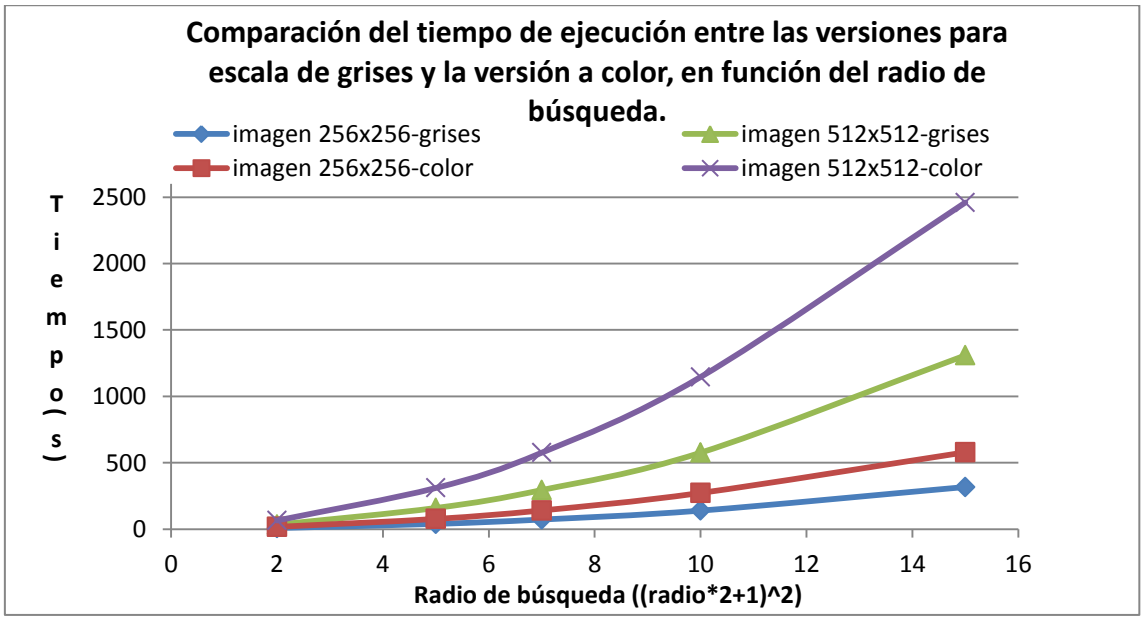
Se van a tomar tiempos de ejecución para una imagen de tamaño 256x256 y otra de tamaño 512x512, aplicándoles a cada una radios de búsqueda de tamaños 2, 5, 7,10 y 15.

Tabla de resultados de tiempo de ejecución:

Versión Color		
	Tiempo (s)	
Radio búsqueda	Imagen 512x512	Imagen 256x256
2 (5x5)	65	17
5 (11x11)	311	77
7 (15x15)	576	141
10 (21x21)	1146	273
15 (31x31)	2460	579

TABLA 59: Tiempos de ejecución de la versión a color, para imágenes de tamaño 256x256 y 512x512 empleando radios de búsqueda de 2, 5, 7, 10 y 15.

A continuación se muestra la gráfica del tiempo en función del radio de búsqueda. Se muestran también los tiempos para las versiones que trabajan con imágenes en escala de gris (v3y4), para ver la diferencia:



GRÁFICA 7: Comparación del tiempo de ejecución entre las versiones para escala de grises y la versión a color, para imágenes de tamaño 256x256 y 512x512 empleando radios de búsqueda de 2, 5, 7, 10 y 15.

Al tener que filtrar cada una de las capas de la imagen a color, el tiempo de ejecución se incrementa respecto al de una imagen en gris. Más adelante veremos que para obtener buenos resultados de filtrado de imágenes a color, los radios empleados son menores que para imágenes en escala de grises. Por tanto el tiempo de ejecución necesario para procesar correctamente una imagen a color, no será mucho mayor que el necesario para una imagen en escala de gris.

4.3.2. Estudio de parámetros para imágenes en color con ruido gaussiano.

De la misma forma que en apartado 2, hemos añadido ruido gaussiano a la imagen, en este caso, de Lena a color con resolución 225x225. La imagen de Lena a color es la siguiente:

La siguiente tabla muestra el PSNR y SSIM de las imágenes con ruido:

RUIDO	PSNR(dB)	SSIM
input_10R.png	28,32260061	0,8339
input_20R.png	22,49546963	0,6360
input_30R.png	19,17542215	0,5028
input_40R.png	16,93054379	0,4128
input_50R.png	15,18971236	0,3444
input_60R.png	13,86333706	0,2955
input_70R.png	12,77056087	0,2572
input_80R.png	11,8839833	0,2253
input_90R.png	11,09995494	0,1986
input_100R.png	10,4710696	0,1768

TABLA 60: PSNR y SSIM de las imágenes de Lena a color, con ruido gaussiano añadido, de desviación 10, 20, 30, 40, 50, 60, 70, 80, 90 y 100.

Para el estudio se emplearán distintas combinaciones de parámetros. Para distintos valores del grado de filtrado (h), se usarán distintas vecindades y radios de búsqueda, de la misma forma que en el apartado 2. Cuando se observe que ciertas combinaciones dan malos resultados, se dejarán de usar, para evitar realizar pruebas innecesarias.

Resultados:

RUIDO 10

IMAGEN	PSNR (dB)	SSIM
input_10R-10-1-5	32,48	0,9421
input_10R-10-2-5	32,42	0,9443
input_10R-10-3-5	32,24	0,9442
input_10R-10-4-5	32,06	0,9439
input_10R-10-1-7	32,57	0,9416
input_10R-10-2-7	32,46	0,9418
input_10R-10-3-7	32,26	0,9411
input_10R-10-4-7	32,08	0,9411
input_10R-10-1-10	32,59	0,9405
input_10R-10-2-10	32,42	0,9388
input_10R-10-3-10	32,21	0,9376
input_10R-10-4-10	32,03	0,9378

IMAGEN	PSNR (dB)	SSIM
input_10R-15-1-5	32,59	0,9278
input_10R-15-2-5	32,37	0,9168
input_10R-15-3-5	32,30	0,9148
input_10R-15-4-5	32,29	0,9170
input_10R-15-1-7	32,27	0,9233
input_10R-15-2-7	31,96	0,9086
input_10R-15-3-7	31,86	0,9055
input_10R-15-4-7	31,84	0,9075
input_10R-15-1-10	31,90	0,9197
input_10R-15-2-10	31,49	0,9014
input_10R-15-3-10	31,36	0,8968
input_10R-15-4-10	31,33	0,8985

TABLA 61: Tablas del PSNR y SSIM de las imágenes resultantes del filtrado, con la versión a color, de la imagen de Lena con ruido gaussiano con desviación 10 para cada combinación de parámetros con h=10 (izquierda) y h=15 (derecha).

RUIDO 20

IMAGEN	PSNR (dB)	SSIM
input_20R-20-1-5	28,51	0,8652
input_20R-20-2-5	28,74	0,8763
input_20R-20-3-5	28,63	0,8766
input_20R-20-4-5	28,50	0,8767
input_20R-20-1-7	28,59	0,8654
input_20R-20-2-7	28,72	0,8718
input_20R-20-3-7	28,57	0,8697
input_20R-20-4-7	28,41	0,8692
input_20R-20-1-10	28,51	0,8629
input_20R-20-2-10	28,52	0,8647
input_20R-20-3-10	28,31	0,8601
input_20R-20-4-10	28,13	0,8586

IMAGEN	PSNR (dB)	SSIM
input_20R-30-1-5	28,44	0,8549
input_20R-30-2-5	27,82	0,8321
input_20R-30-3-5	27,40	0,8201
input_20R-30-4-5	27,12	0,8141
input_20R-30-1-7	28,07	0,8478
input_20R-30-2-7	27,29	0,8195
input_20R-30-3-7	26,80	0,8045
input_20R-30-4-7	26,47	0,7969
input_20R-30-1-10	27,60	0,8415
input_20R-30-2-10	26,69	0,8073
input_20R-30-3-10	26,11	0,7885
input_20R-30-4-10	25,73	0,7789

TABLA 62: Tablas del PSNR y SSIM de las imágenes resultantes del filtrado, con la versión a color, de la imagen de Lena con ruido gaussiano con desviación 20 para cada combinación de parámetros con h=20 (izquierda) y h=30 (derecha).

RUIDO 30

IMAGEN	PSNR (dB)	SSIM
input_30R-30-1-5	26,26	0,7883
input_30R-30-2-5	26,63	0,8103
input_30R-30-3-5	26,53	0,8107
input_30R-30-4-5	26,40	0,8090
input_30R-30-1-7	26,31	0,7898

IMAGEN	PSNR (dB)	SSIM
input_30R-40-1-5	26,41	0,7998
input_30R-40-2-5	25,93	0,7855
input_30R-40-3-5	25,47	0,7715
input_30R-40-4-5	25,13	0,7620
input_30R-40-1-7	26,10	0,7931

input_30R-30-2-7	26,51	0,8051
input_30R-30-3-7	26,28	0,8010
input_30R-30-4-7	26,08	0,7972
input_30R-30-1-10	26,16	0,7873
input_30R-30-2-10	26,16	0,7968
input_30R-30-3-10	25,82	0,7881
input_30R-30-4-10	25,54	0,7816

input_30R-40-2-7	25,45	0,7725
input_30R-40-3-7	24,90	0,7546
input_30R-40-4-7	24,51	0,7430
input_30R-40-1-10	25,66	0,7869
input_30R-40-2-10	24,89	0,7607
input_30R-40-3-10	24,25	0,7389
input_30R-40-4-10	23,82	0,7247

TABLA 63: Tablas del PSNR y SSIM de las imágenes resultantes del filtrado, con la versión a color, de la imagen de Lena con ruido gaussiano con desviación 30 para cada combinación de parámetros con h=30 (izquierda) y h=40 (derecha).

RUIDO 40

IMAGEN	PSNR (dB)	SSIM
input_40R-30-1-5	22,65	0,6342
input_40R-30-2-5	23,33	0,6800
input_40R-30-3-5	23,39	0,6898
input_40R-30-4-5	23,36	0,6906
input_40R-30-1-7	23,17	0,6528
input_40R-30-2-7	24,00	0,7102
input_40R-30-3-7	24,02	0,7223
input_40R-30-4-7	23,93	0,7236
input_40R-30-1-10	23,46	0,6632
input_40R-30-2-10	24,34	0,7275
input_40R-30-3-10	24,30	0,7392
input_40R-30-4-10	24,14	0,7399

IMAGEN	PSNR (dB)	SSIM
input_40R-40-1-5	24,61	0,7190
input_40R-40-2-5	24,94	0,7515
input_40R-40-3-5	24,74	0,7510
input_40R-40-4-5	24,52	0,7468
input_40R-40-1-7	24,60	0,7220
input_40R-40-2-7	24,72	0,7485
input_40R-40-3-7	24,37	0,7419
input_40R-40-4-7	24,08	0,7341
input_40R-40-1-10	24,38	0,7192
input_40R-40-2-10	24,31	0,7411
input_40R-40-3-10	23,84	0,7293
input_40R-40-4-10	23,48	0,7183

IMAGEN	PSNR (dB)	SSIM
input_40R-50-1-5	24,77	0,7412
input_40R-50-2-5	24,37	0,7395
input_40R-50-3-5	23,90	0,7251
input_40R-50-4-5	23,57	0,7137
input_40R-50-1-7	24,48	0,7361
input_40R-50-2-7	23,92	0,7280
input_40R-50-3-7	23,37	0,7088
input_40R-50-4-7	22,98	0,6944
input_40R-50-1-10	24,04	0,7289
input_40R-50-2-10	23,39	0,7173
input_40R-50-3-10	22,79	0,6942
input_40R-50-4-10	22,35	0,6768

TABLA 64: Tablas del PSNR y SSIM de las imágenes resultantes del filtrado, con la versión a color, de la imagen de Lena con ruido gaussiano con desviación 40 para cada combinación de parámetros con h=30 (arriba a la izquierda), h=40 (arriba a la derecha) y h=50 (abajo).

RUIDO 50

IMAGEN	PSNR (dB)	SSIM
input_50R-35-1-5	20,72	0,5428
input_50R-35-2-5	21,43	0,5872
input_50R-35-3-5	21,50	0,5956
input_50R-35-4-5	21,49	0,5958
input_50R-35-1-7	21,32	0,5630
input_50R-35-2-7	22,27	0,6239
input_50R-35-3-7	22,34	0,6379
input_50R-35-4-7	22,29	0,6393
input_50R-35-1-10	21,69	0,5753
input_50R-35-2-10	22,80	0,6493
input_50R-35-3-10	22,83	0,6667
input_50R-35-4-10	22,71	0,6687

IMAGEN	PSNR (dB)	SSIM
input_50R-40-1-5	22,04	0,5961
input_50R-40-2-5	22,96	0,6568
input_50R-40-3-5	23,07	0,6716
input_50R-40-4-5	23,04	0,6748
input_50R-40-1-7	22,41	0,6095
input_50R-40-2-7	23,32	0,6778
input_50R-40-3-7	23,30	0,6917
input_50R-40-4-7	23,17	0,6933
input_50R-40-1-10	22,51	0,6149
input_50R-40-2-10	23,32	0,6856
input_50R-40-3-10	23,15	0,6960
input_50R-40-4-10	22,92	0,6944

IMAGEN	PSNR (dB)	SSIM
input_50R-50-1-5	23,25	0,6603
input_50R-50-2-5	23,49	0,7002
input_50R-50-3-5	23,22	0,6996
input_50R-50-4-5	22,97	0,6941
input_50R-50-1-7	23,20	0,6618
input_50R-50-2-7	23,24	0,6983
input_50R-50-3-7	22,85	0,6923
input_50R-50-4-7	22,53	0,6827
input_50R-50-1-10	22,95	0,6579
input_50R-50-2-10	22,84	0,6912
input_50R-50-3-10	22,36	0,6800
input_50R-50-4-10	21,97	0,6667

TABLA 65: Tablas del PSNR y SSIM de las imágenes resultantes del filtrado, con la versión a color, de la imagen de Lena con ruido gaussiano con desviación 40 para cada combinación de parámetros con h=35 (arriba a la izquierda), h=40 (arriba a la derecha) y h=50 (abajo).

RUIDO 60

IMAGEN	PSNR (dB)	SSIM
input_60R-40-1-7	20,00	0,4970
input_60R-40-2-7	21,11	0,5595
input_60R-40-3-7	21,25	0,5735
input_60R-40-4-7	21,23	0,5751
input_60R-40-1-10	20,38	0,5106
input_60R-40-2-10	21,70	0,5905
input_60R-40-3-10	21,79	0,6097
input_60R-40-4-10	21,69	0,6122

IMAGEN	PSNR (dB)	SSIM
input_60R-50-1-7	21,62	0,5711
input_60R-50-2-7	22,31	0,6425
input_60R-50-3-7	22,12	0,6524
input_60R-50-4-7	21,88	0,6498
input_60R-50-1-10	21,57	0,5733
input_60R-50-2-10	22,11	0,6466
input_60R-50-3-10	21,79	0,6516
input_60R-50-4-10	21,45	0,6437

IMAGEN	PSNR (dB)	SSIM
input_60R-60-1-7	22,03	0,6109
input_60R-60-2-7	21,99	0,6527

input_60R-60-3-7	21,57	0,6448
input_60R-60-4-7	21,24	0,6324
input_60R-60-1-10	21,73	0,6060
input_60R-60-2-10	21,57	0,6463
input_60R-60-3-10	21,09	0,6334
input_60R-60-4-10	20,70	0,6170

TABLA 66: Tablas del PSNR y SSIM de las imágenes resultantes del filtrado, con la versión a color, de la imagen de Lena con ruido gaussiano con desviación 60 para cada combinación de parámetros con h=40 (arriba a la izquierda), h=40 (arriba a la derecha) y h=60 (abajo).

RUIDO 70

IMAGEN	PSNR (dB)	SSIM
input_70R-45-1-7	18,96	0,4448
input_70R-45-2-7	20,15	0,5095
input_70R-45-3-7	20,32	0,5260
input_70R-45-4-7	20,31	0,5291
input_70R-45-1-10	19,32	0,4568
input_70R-45-2-10	20,71	0,5407
input_70R-45-3-10	20,83	0,5642
input_70R-45-4-10	20,73	0,5682

IMAGEN	PSNR (dB)	SSIM
input_70R-55-1-7	20,45	0,5105
input_70R-55-2-7	21,25	0,5914
input_70R-55-3-7	21,11	0,6076
input_70R-55-4-7	20,89	0,6071
input_70R-55-1-10	20,44	0,5131
input_70R-55-2-10	21,09	0,5991
input_70R-55-3-10	20,83	0,6115
input_70R-55-4-10	20,52	0,6060

IMAGEN	PSNR (dB)	SSIM
input_70R-65-1-7	20,95	0,5519
input_70R-65-2-7	21,07	0,6110
input_70R-65-3-7	20,71	0,6098
input_70R-65-4-7	20,41	0,5998
input_70R-65-1-10	20,69	0,5476
input_70R-65-2-10	20,69	0,6068
input_70R-65-3-10	20,26	0,6005
input_70R-65-4-10	19,91	0,5861

TABLA 67: Tablas del PSNR y SSIM de las imágenes resultantes del filtrado, con la versión a color, de la imagen de Lena con ruido gaussiano con desviación 70 para cada combinación de parámetros con h=45 (arriba a la izquierda), h=55 (arriba a la derecha) y h=65 (abajo).

RUIDO 80

IMAGEN	PSNR (dB)	SSIM
input_80R-50-1-7	18,22	0,4060
input_80R-50-2-7	19,52	0,4774
input_80R-50-3-7	19,69	0,4965
input_80R-50-4-7	19,67	0,5004
input_80R-50-1-10	18,52	0,4149
input_80R-50-2-10	19,97	0,5063
input_80R-50-3-10	20,06	0,5329
input_80R-50-4-10	19,94	0,5382

IMAGEN	PSNR (dB)	SSIM
input_80R-60-1-7	19,53	0,4634
input_80R-60-2-7	20,35	0,5506
input_80R-60-3-7	20,21	0,5691
input_80R-60-4-7	20,00	0,5699
input_80R-60-1-10	19,49	0,4637
input_80R-60-2-10	20,19	0,5588
input_80R-60-3-10	19,93	0,5751
input_80R-60-4-10	19,65	0,5712

IMAGEN	PSNR (dB)	SSIM
input_80R-70-1-7	20,02	0,5027
input_80R-70-2-7	20,21	0,5732
input_80R-70-3-7	19,89	0,5767
input_80R-70-4-7	19,64	0,5689
input_80R-70-1-10	19,76	0,4965
input_80R-70-2-10	19,84	0,5695
input_80R-70-3-10	19,46	0,5688
input_80R-70-4-10	19,15	0,5564

TABLA 68: Tablas del PSNR y SSIM de las imágenes resultantes del filtrado, con la versión a color, de la imagen de Lena con ruido gaussiano con desviación 80 para cada combinación de parámetros con h=50 (arriba a la izquierda), h=60 (arriba a la derecha) y h=70 (abajo).

RUIDO 90

IMAGEN	PSNR (dB)	SSIM
input_90R-60-1-7	19,28	0,4005
input_90R-60-2-7	19,29	0,4882
input_90R-60-3-7	19,00	0,5143
input_90R-60-4-7	18,76	0,5200
input_90R-60-1-10	18,96	0,4032
input_90R-60-2-10	18,89	0,5043
input_90R-60-3-10	18,54	0,5319
input_90R-60-4-10	18,26	0,5349

IMAGEN	PSNR (dB)	SSIM
input_90R-70-1-7	19,01	0,4448
input_90R-70-2-7	19,49	0,5304
input_90R-70-3-7	19,26	0,5461
input_90R-70-4-7	19,04	0,5444
input_90R-70-1-10	18,87	0,4401
input_90R-70-2-10	19,21	0,5309
input_90R-70-3-10	18,91	0,5430
input_90R-70-4-10	18,62	0,5366

IMAGEN	PSNR (dB)	SSIM
input_90R-80-1-7	18,24	0,4767
input_90R-80-2-7	19,37	0,5442
input_90R-80-3-7	19,37	0,5473
input_90R-80-4-7	19,23	0,5396
input_90R-80-1-10	18,34	0,4660
input_90R-80-2-10	19,40	0,5352
input_90R-80-3-10	19,27	0,5341
input_90R-80-4-10	19,04	0,5223

TABLA 69: Tablas del PSNR y SSIM de las imágenes resultantes del filtrado, con la versión a color, de la imagen de Lena con ruido gaussiano con desviación 90 para cada combinación de parámetros con h=60 (arriba a la izquierda), h=70 (arriba a la derecha) y h=80 (abajo).

RUIDO 100

IMAGEN	PSNR (dB)	SSIM
input_100R-65-1-7	17,58	0,3653
input_100R-65-2-7	18,61	0,4544
input_100R-65-3-7	18,57	0,4819

input_100R-65-4-7	18,44	0,4885
input_100R-65-2-10	17,63	0,3659
input_100R-65-3-10	18,57	0,4674
input_100R-65-4-10	18,42	0,4971

IMAGEN	PSNR (dB)	SSIM
input_100R-75-1-7	18,23	0,4041
input_100R-75-2-7	18,67	0,4934
input_100R-75-3-7	18,47	0,5122
input_100R-75-4-7	18,28	0,5128
input_100R-75-1-10	18,07	0,3982
input_100R-75-2-10	18,39	0,4923
input_100R-75-3-10	18,12	0,5083
input_100R-75-4-10	17,88	0,5045

IMAGEN	PSNR (dB)	SSIM
input_100R-85-1-7	18,47	0,4339
input_100R-85-2-7	18,52	0,5093
input_100R-85-3-7	18,27	0,5168
input_100R-85-4-7	18,08	0,5120
input_100R-85-1-10	18,16	0,4225
input_100R-85-2-10	18,12	0,4990
input_100R-85-3-10	17,83	0,5028
input_100R-85-4-10	17,60	0,4944

TABLA 70: Tablas del PSNR y SSIM de las imágenes resultantes del filtrado, con la versión a color, de la imagen de Lena con ruido gaussiano con desviación 100 para cada combinación de parámetros con h=65 (arriba), h=75 (abajo a la izquierda) y h=85 (abajo a la derecha).

Análisis de los resultados:

A continuación se muestra la tabla resumen donde, para cada cantidad de ruido, se ha seleccionado cuál es la combinación de parámetros más adecuada:

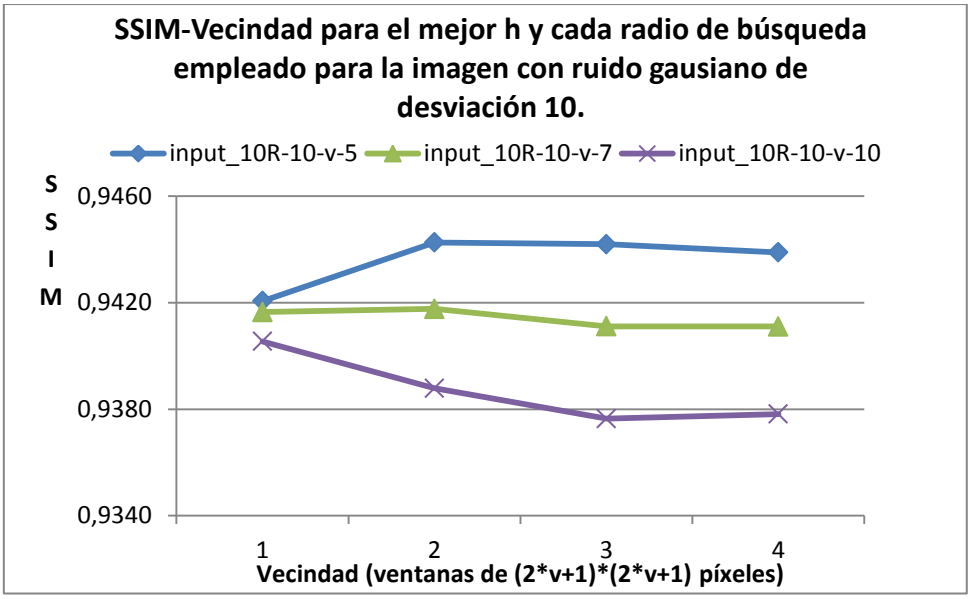
	h	vecindad	radio
0<ruido<=45	ruido	2	5
45<ruido<=75	0,8*ruido	3	7
75<ruido<=90	0,75*ruido	3	7
90<ruido<=100	0,75*ruido	4	7

TABLA 71: Tabla resumen de parámetros en función del ruido para la versión color.

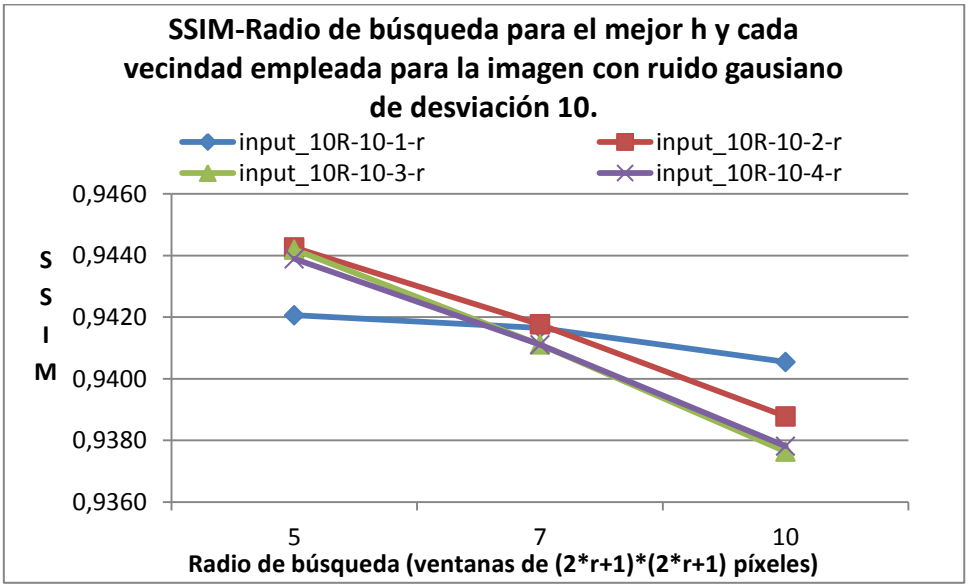
ruido	h	vecindad	radio
10	10	2	5
20	20	2	5
30	30	2	5
40	40	2	5
50	40	3	7
60	48	3	7
70	56	3	7
80	60	3	7
90	67,5	3	7
100	75	4	7

TABLA 72: Tabla de parámetros exactos para ruidos con desviación 10, 20, 30, 40, 50, 60, 70, 80, 90 y 100 para la versión a color.

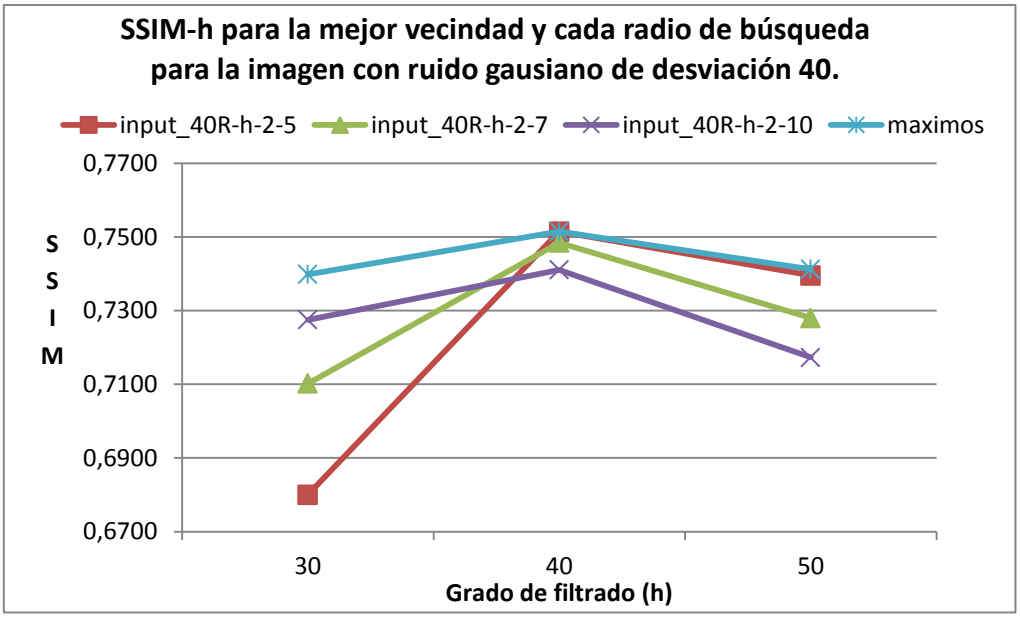
Al igual que en las versiones anteriores, a continuación se muestran diferentes gráficas que resumen el comportamiento de cada parámetro. Además, también se muestran varias sucesiones de imágenes para comprobar cómo varía el resultado visualmente.



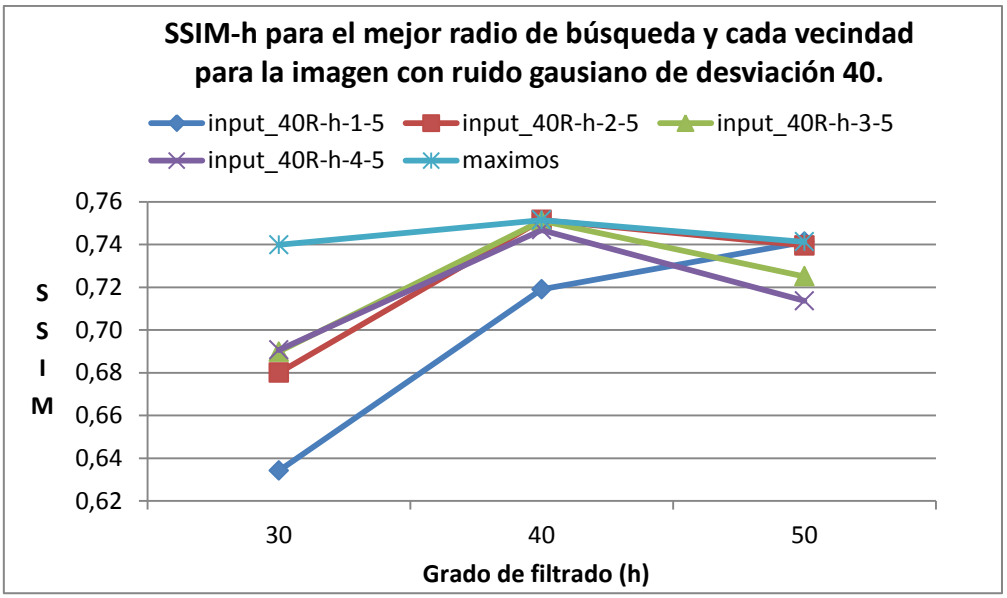
(a)



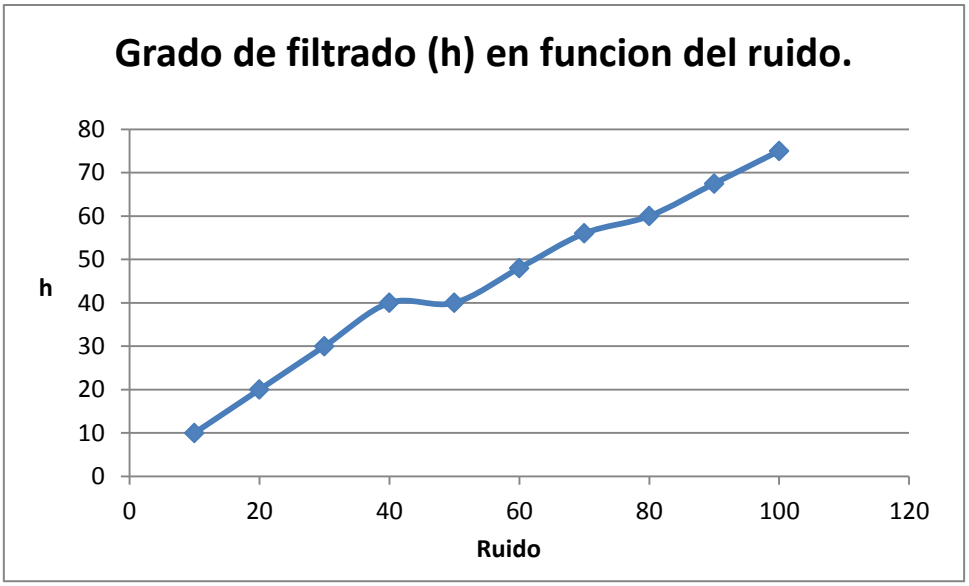
(b)


SSIM-h para la mejor vecindad y cada radio de búsqueda para la imagen con ruido gausiano de desviación 40.
input_40R-h-2-5
input_40R-h-2-7
input_40R-h-2-10
maximos
SSIM
0,7700
0,7500
0,7300
0,7100
0,6900
0,6700
30
40
50
Grado de filtrado (h)

(c)


SSIM-h para el mejor radio de búsqueda y cada vecindad para la imagen con ruido gausiano de desviación 40.
input_40R-h-1-5
input_40R-h-2-5
input_40R-h-3-5
input_40R-h-4-5
maximos
SSIM
0,76
0,74
0,72
0,70
0,68
0,66
0,64
0,62
30
40
50
Grado de filtrado (h)

(d)

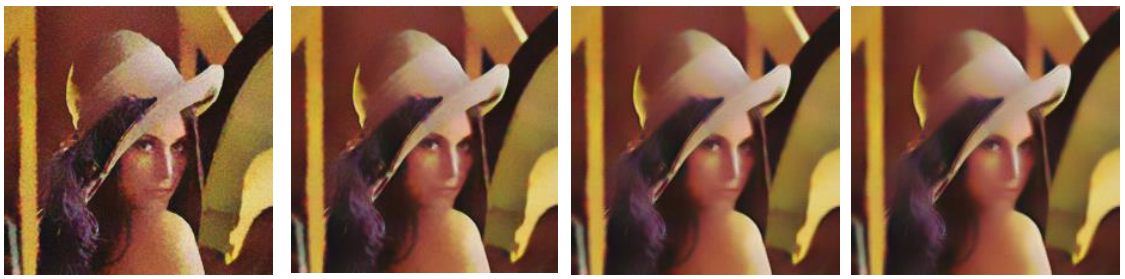


(e)

GRÁFICA 8: Gráficas para la versión a color: ***a)****SSIM-Vecindad para el mejor h y cada radio de búsqueda empleado para la imagen con ruido gaussiano de desviación 10.* ***b)****SSIM-Radio de búsqueda para el mejor h y cada vecindad empleada para la imagen con ruido gaussiano de desviación 10.* ***c)****SSIM-h para la mejor vecindad y cada radio de búsqueda para la imagen con ruido gaussiano de desviación 40.* ***d)****SSIM-h para el mejor radio de búsqueda y cada vecindad para la imagen con ruido gaussiano de desviación 40.* ***e)****Grado de filtrado(h) en función del ruido.*

La siguiente sucesión de imágenes es un ejemplo para la gráfica a), tomando como imagen la imagen de Lena a color con ruido gaussiano de desviación 40:

Imagen con ruido gaussiano desviación 40. SSIM=0,4128



v=1, SSIM=0,7192 ***v=2, SSIM=0,7411*** *v=3, SSIM=0,7293* *v=4, SSIM=0,7183*

Sucesion de imágenes para cada vecindad (v), para el mejor h (h=40) y para radio de búsqueda = 10 (ventana de 21x21), para la imagen de lena a color con ruido gausiano de desviación 40.

La siguiente sucesión de imágenes es un ejemplo para la gráfica b), tomando como imagen con ruido la imagen de Lena a color con ruido gaussiano de desviación 40:

r=5, SSIM= 0,7468 *r=7, SSIM= 0,7341* *r=10, SSIM= 0,7183*

Sucesion de imágenes para vecindad=4 (ventana de 9x9), para el mejor h (h=40) y para cada radio de búsqueda (r), para la imagen de lena a color con ruido gausiano de desviación 40.

La siguiente sucesión de imágenes es un ejemplo para la gráfica c) y d), tomando como imagen con ruido la imagen de Lena con ruido gaussiano de desviación 40:

h=30, SSIM=0,6800 ***h=40, SSIM=0,7515*** *h=50, SSIM=0,7395*

Sucesion de imágenes para la mejor vecindad =2 (ventana de 5x5), para h=30, 40, 50 y para el mejor radio de búsqueda = 5 (ventana de 11x11), para la imagen de lena con ruido gausiano de desviación 30.

A continuación se analizan los resultados:

Los resultados de las pruebas y las gráficas, muestran un comportamiento parecido al de la versión 4 aplicada en imágenes en escala de grises. El hecho de que la imagen sea a color no varía mucho en los resultados. Lo único destacado es que con radios menores que para imágenes en grises obtenemos buenos resultados. Para un radio dado, según aumenta la vecindad más detalles de la imagen se pierden, tras alcanzar el máximo SSIM. Lo mismo sucede al aumentar el radio para una vecindad dada. Valores altos de h también hacen que el ruido se elimine en exceso y se pierdan los detalles de la imagen.

Las siguientes imágenes se corresponden con las de Lena a color con ruido gaussiano y con la mejor imagen obtenida tras aplicar los parámetros exactos correspondientes a la versión a color:

Imagen con ruido gaussiano Imagen Filtrada con versión a color

Desviacion=10, SSIM=0,8339

h=10,v=2,r=5,SSIM=0,9443

Desviacion=20, SSIM=0,6360

h=20,v=2,r=5,SSIM=0,8763

Desviacion=30, SSIM=0,5028

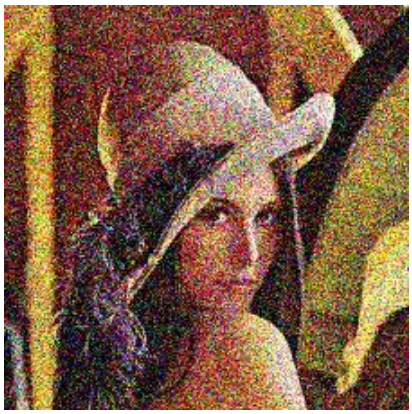
h=30,v=2,r=5,SSIM=0,8103

Desviacion=40, SSIM=0,4128

h=40,v=2,r=5,SSIM=0,7515

Desviacion=50, SSIM=0,3444

h=40,v=3,r=7,SSIM=0,6917

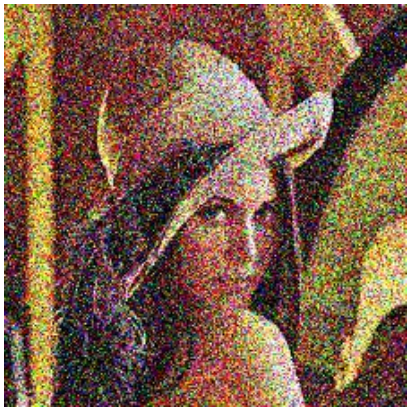


Desviacion=60, SSIM=0,2955

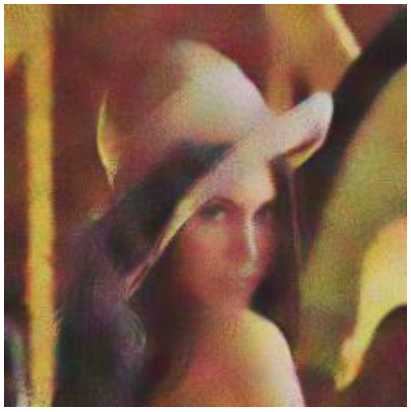
h=48,v=3,r=7,SSIM=0,6485

Desviacion=70, SSIM=0,2572

h=56,v=3,r=7,SSIM=0,6094

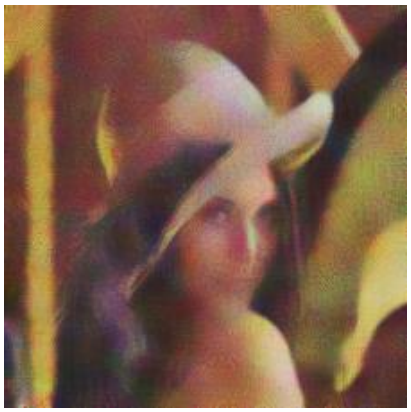


Desviacion=80, SSIM=0,2253

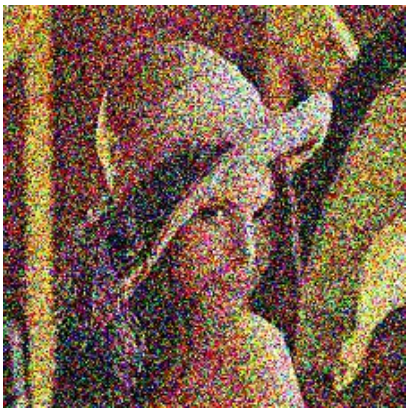


h=60,v=3,r=7,SSIM=0,5691

Desviacion=90, SSIM=0,1986

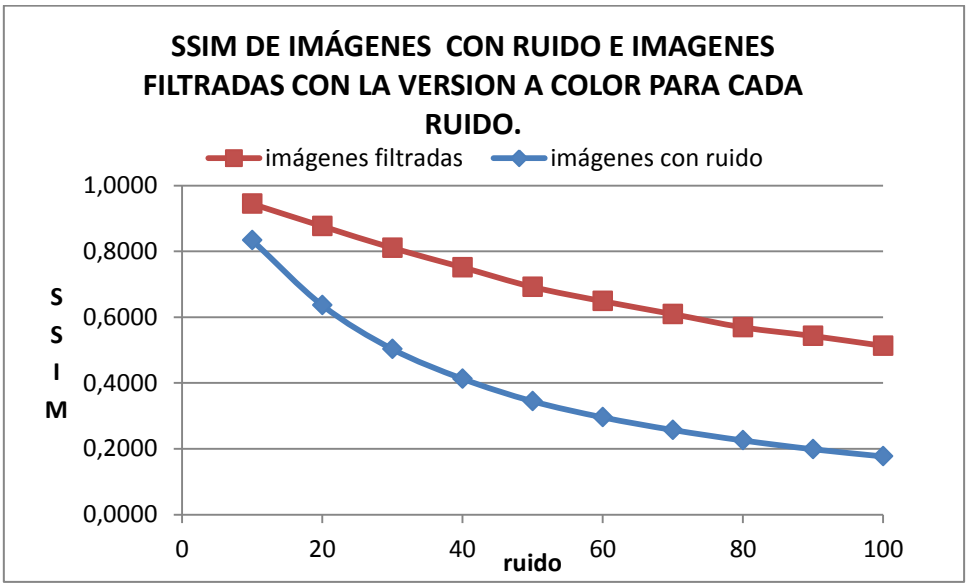


h=68,v=3,r=7,SSIM=0,5427



Desviacion=100, SSIM=0,1768 *h=75,v=4,r=7,SSIM=0,5128*

La siguiente grafica muestra los resultados anteriores:



GRÁFICA 9: Comparación del SSIM entre las imágenes con ruido gaussiano de desviación 10, 20, 30, 40, 50, 60, 70, 80, 90 y 100, y las imágenes obtenidas tras aplicar la mejor combinación de parámetros de la versión a color a cada una.

4.3.3. Aplicación de los resultados a imágenes en color con ruido gaussiano y rician

Aplicación de resultados a imágenes con ruido gaussiano:

A continuación vamos a aplicar el filtro Non Local Means con los parámetros que hemos determinado en el apartado anterior a varias imágenes con ruido gaussiano:

Imagen Original: barco.jpg

Imagen con ruido gaussiano Imagen filtrada

Desviación=10 *h=10,v=2,r=5,SSIM= 0,9474*

Desviación=30 *h=30,v=2,r=5,SSIM= 0,8373*

Imagen Original: flores.jpg

Imagen con ruido gaussiano Imagen filtrada

Desviación=25

h=25,v=2,r=5,SSIM= 0,8494

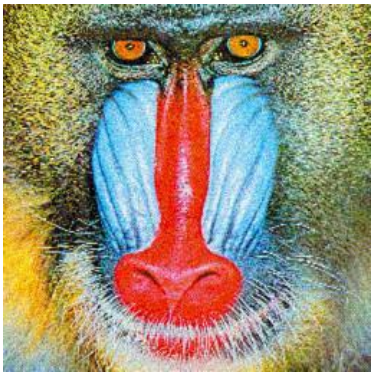
Desviación=70

h=56,v=3,r=7,SSIM= 0,6301

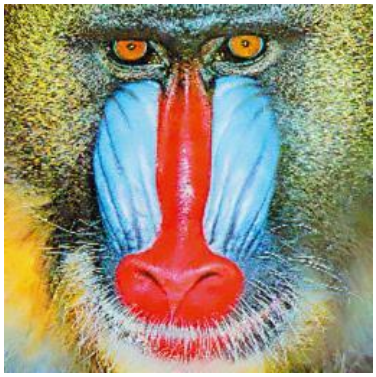
Imagen Original: mandril.jpg

Imagen con ruido gaussiano

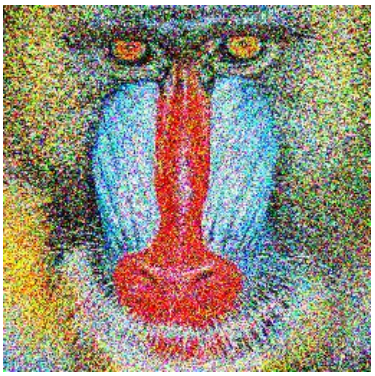
Imagen filtrada



Desviación=15



h=15,v=2,r=5,SSIM= 0,9042



Desviación=85

h=64,v=3,r=7,SSIM= 0,3467

Aplicación de resultados a imágenes con ruido rician:

A continuación se muestran los resultados obtenidos para la imagen de Lena a color, a la que se le ha añadido ruido rician con distinta desviación y los resultados del filtrado tras aplicar los parámetros obtenidos para ruido gaussiano:

IMAGEN	PSNR	SSIM
input_10R	28,06	0,8303
input_20R	22,06	0,6189
input_30R	18,51	0,4713
input_40R	16,03	0,3711
input_50R	14,08	0,2992
input_60R	12,47	0,2397
input_70R	11,23	0,2009
input_80R	10,11	0,1661
input_90R	9,16	0,1379
input_100R	8,39	0,1153

IMAGEN	PSNR	SSIM
input_10R-10-2-5	31,53	0,9345
input_20R-20-2-5	26,46	0,8416
input_30R-30-2-5	22,92	0,7547
input_40R-40-2-5	20,04	0,6755
input_50R-40-3-7	17,77	0,6107
input_60R-48-3-7	15,76	0,5514
input_70R-56-3-7	14,13	0,5031
input_80R-60-3-10	12,76	0,4562
input_90R-68-3-10	11,55	0,4178
input_100R-65-4-10	10,53	0,3934

TABLA 73: Tablas de PSNR y SSIM para la imagen Lena a color con ruido rician de desviación 10, 20, 30, 40, 50, 60, 70, 80, 90 y100 (a la izquierda) y de resultados obtenidos tras aplicar los parámetros exactos a cada una(a la derecha).

A continuación se muestran varias imágenes:

Imagen Original: lena256.png

Imagen con ruido rician

Desviación=20

Imagen filtrada

h=20,v=2,r=5,SSIM= 0,9375

Desviación=60

h=48,v=3,r=7,SSIM=0,5510

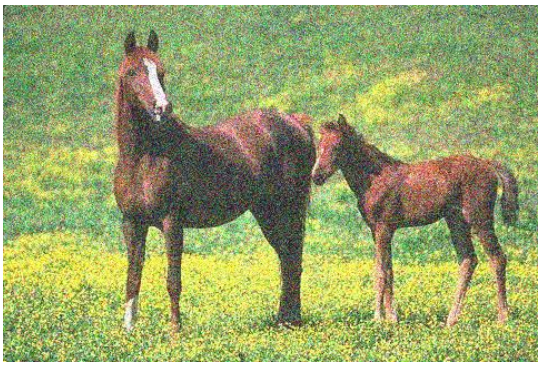
Imagen Original: caballo.jpg

Imagen con ruido rician

Desviación=10

Imagen filtrada

h=10,v=2,r=5,SSIM= 0,9298



Desviación=40

h=40,v=2,r=5,SSIM= 0,6172

Imagen Original: flores.jpg

Imagen con ruido rician

Imagen filtrada

Desviación=20

h=20,v=2,r=5,SSIM= 0,8277

Desviación=50

h=40,v=3,r=7,SSIM= 0,5609

Imagen Original: mandril.jpg

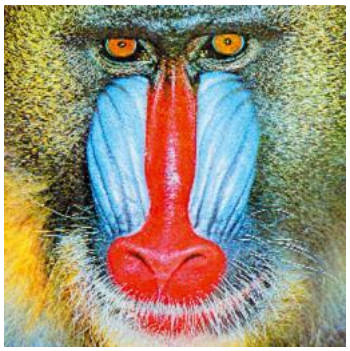
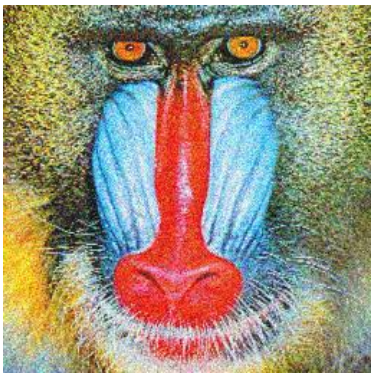
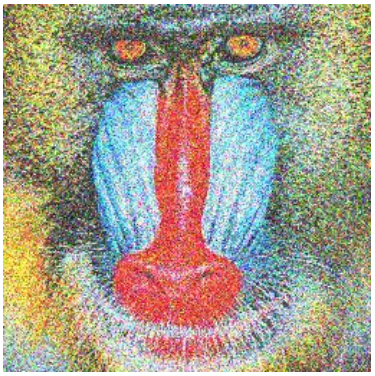


Imagen con ruido rician Imagen filtrada



Desviación=25 *h=25,v=2,r=5,SSIM= 0,7693*



Desviación=60 *h=48,v=3,r=7,SSIM= 0,3565*

Para este ruido rician, también se obtienen buenos resultados. Sin embargo, sucede como en las anteriores versiones. Para imágenes con una cantidad considerable de ruido, tras el filtrado, se pierde contraste. Empleando cualquier técnica para el realce del contraste sería suficiente para solucionarlo.

4.4. Comparación de resultados con otros filtros

En esta sección vamos a comparar los resultados que hemos obtenido con una de nuestras versiones del algoritmo Non Local Means frente a otros algoritmos de filtrado de la literatura. Para esta prueba vamos a utilizar la versión 4 (adaptada a imágenes en color), que utiliza la distancia euclídea ponderada y un radio de búsqueda. En concreto, vamos a compararla con el filtro de la media y el filtro gaussiano. Estas pruebas las haremos con imágenes en escala de grises y en color, que presenten tanto ruido gaussiano como ruido rician.

Comparación en imágenes en escala de grises con ruido gaussiano:


(a) *(b)SSIM=0,4429*



(c)SSIM=0,7476 *(d) SSIM=0,7013* *(e)SSIM=0,8393*

Comparacion de imágenes obtenidas de la imagen avion.jpg empleando diferentes filtros: a)imagen original. b)imagen con ruido gausiano de desviacion 20. c)imagen obtenida con filtro gausiano. d)imagen obtenida con el filtro de la media. e)imagen obtenida con filtro NLMs con h=20,v=2 y r=7.

(a) *(b)SSIM=0,5323*

(c)SSIM=0,8484 *(d)SSIM=0,8378* *(e)SSIM=0,8991*

Comparacion de imágenes obtenidas de la imagen matricula.bmp empleando diferentes filtros: a)imagen original. b)imagen con ruido gausiano de desviacion 15. c)imagen obtenida con filtro gausiano. d)imagen obtenida con el filtro de la media. e)imagen obtenida con filtro NLMs con h=15,v=2 y r=5.

Comparación en imágenes en escala de grises con ruido rician:

(a)SSIM=0,3294



(b)SSIM=0,6061 *(c)SSIM=0,6457* *(d)SSIM=0,7738*

Comparacion de imágenes obtenidas de la imagen avion.jpg empleando diferentes filtros: a) imagen con ruido rician de desviacion 30. b)imagen obtenida con filtro gausiano. c)imagen obtenida con el filtro de la media. b)imagen obtenida con filtro NLMs con h=30,v=2 y r=7.


(a) *(b)SSIM=0,7372*



(c)SSIM=0,7956 *(d) SSIM=0,7818* *(e) SSIM=0,9137*

Comparacion de imágenes obtenidas de la imagen radiografia.jpg empleando diferentes filtros: a)imagen original. b)imagen con ruido rician de desviacion 10. c)imagen obtenida con filtro gausiano. d)imagen obtenida con el filtro de la media. e)imagen obtenida con filtro NLMs con h=10,v=2 y r=5.

Comparación en imágenes en color con ruido gaussiano:


(a) *(b)SSIM=0,5293*



(c) *SSIM=0,7291* (d) *SSIM=0,6839* (e) *SSIM=0,8373*

Comparacion de imágenes obtenidas de la imagen barco.jpg empleando diferentes filtros: a)imagen original. b)imagen con ruido gausiano de desviacion 30. c)imagen obtenida con filtro gausiano. d)imagen obtenida con el filtro de la media. e)imagen obtenida con filtro NLMs con h=30,v=2 y r=5.


(a) *(b)SSIM= 0,1585*



(c) *SSIM= 0,6318* (d) *SSIM= 0,6338* (e) *SSIM= 0,6401*

Comparacion de imágenes obtenidas de la imagen flores.jpg empleando diferentes filtros: a)imagen original. b)imagen con ruido gausiano de desviacion 70. c)imagen obtenida con filtro gausiano. d)imagen obtenida con el filtro de la media. e)imagen obtenida con filtro NLMs con h=56,v=3 y r=7.

Comparación en imágenes en color con ruido rician:


(a) *(b)SSIM=0, 8520*



(c) *SSIM=0,8723* (d) *SSIM=0,8665* (e) *SSIM=0,9298*

Comparacion de imágenes obtenidas de la imagen caballo.jpg empleando diferentes filtros: a)imagen original. b)imagen con ruido gausiano de desviacion 10. c)imagen obtenida con filtro gausiano. d)imagen obtenida con el filtro de la media. e)imagen obtenida con filtro NLMs con h=10,v=2 y r=5.

(a)SSIM= 0, 5331



(b) *SSIM= 0,7985* (c) *SSIM= 0,7904* (d) *SSIM= 0,8277*

Comparacion de imágenes obtenidas de la imagen flores.jpg empleando diferentes filtros: a imagen con ruido gausiano de desviacion 20. b)imagen obtenida con filtro gausiano. c)imagen obtenida con el filtro de la media. d)imagen obtenida con filtro NLMs con h=20,v=2 y r=5.

Se observa que el filtro NLMs funciona mejor que el de la media y el gaussiano. Sin embargo dependiendo de la cantidad de ruido que tenga la imagen, debemos plantearnos el utilizar cualquier otro filtro que pueda dar resultados parecidos al NLMs y que sea más rápido en tiempo de ejecución.

Capítulo 5: Adaptación a Android. NLMFilter App

5.1. Tecnologías empleadas

Smartphone:

El Smartphone empleado para probar la aplicación ha sido un Sony Xperia P. Procesador de doble núcleo U8500 de 1 GHz. Memoria RAM de 1GB.

Android y su entorno

Android es un sistema operativo basado en Linux. Fue diseñado y desarrollado por Google para dispositivos con pantalla táctil principalmente, como lo son los Smartphone o las tablets. El desarrollo y la mayoría de las aplicaciones que tiene, se ejecutan sobre una máquina virtual parecida a la de java. Google ofrece la posibilidad de desarrollar empleando el SDK (Software Development Kit) y el NDK (Native Development Kit). El primero (el que he utilizado), es el que se basa en la máquina virtual (dalvik). La sintaxis de desarrollo es idéntica a la de java, por tanto es orientada a objetos.
Por otro lado el NDK permite trabajar sobre el Linux subyacente de Android. También permite implementar partes de la aplicación utilizando lenguajes de código nativo como C y C + +. El NDK es muy utilizado en aplicaciones de videojuegos, procesamiento de imágenes, etc.

ECLIPSE

Es un programa informático compuesto por un conjunto de herramientas de programación de código abierto multiplataforma. Permite crear y desarrollar proyectos en java, Android… Incorpora varios entornos de desarrollo como por ejemplo el JDK para java y el SDK para Android.

OpenCV

OpenCV (Open Source Computer Vision Library) es una librería de software de código abierto para visión por ordenador y para aprendizaje de máquina. OpenCV fue construido para proporcionar una infraestructura común para aplicaciones de visión por computador y de acelerar el uso de la percepción de la máquina en los productos comerciales. Al ser un producto de licencia BSD, OpenCV hace que sea fácil para las empresas utilizar y modificar el código.
La biblioteca cuenta con más de 2.500 algoritmos optimizados. Estos algoritmos se pueden utilizar para detectar y reconocer rostros, identificar objetos, clasificar las acciones humanas en los videos, etc.
Empresas como Google, Yahoo, Microsoft, Intel, IBM, Sony, Honda, Toyota emplean OpenCV.
Cuenta con interfaces de C++, C, Python, Java y MATLAB y es compatible con Windows, Linux, Android y Mac OS. OpenCV se inclina principalmente hacia las aplicaciones de visión en tiempo real. Hay más de 500 algoritmos y una gran cantidad de funciones que la componen o apoyan esos algoritmos.

5.2. Aplicación Android NLMFilter App

NLMFilter App, una aplicación para todos:

Tras el estudio de los parámetros en el apartado experimental, hemos llegado a diversas tablas resumen que indican los parámetros más adecuados en función de la cantidad de ruido que

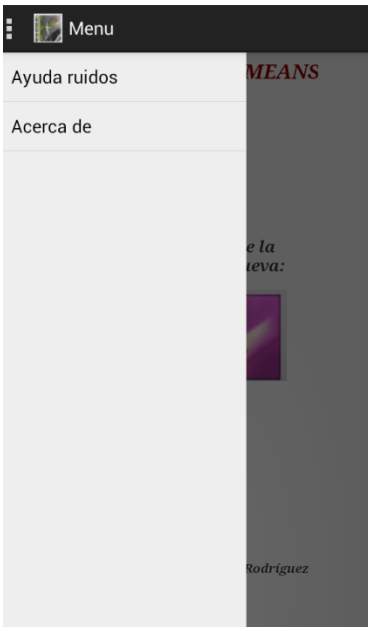
tenga la imagen. Con esta aplicación se pretende hacer uso de ese estudio. La idea es que cualquier persona pueda emplear la aplicación obteniendo buenos resultados en el filtrado de la imagen. La versión implementada del Non Local Means corresponde con la versión 4.

Si una persona conoce la cantidad exacta de ruido que tiene la imagen bastará con introducirla y se le indicarán los parámetros exactos obtenidos en la tabla resumen. Si por el contrario no lo conoce, se le dará la opción de elegir entre la cantidad de ruido que él cree que tiene la imagen. Podrá elegir entre diversas cantidades de ruido: nada, muy poco, poco, bastante, mucho y no veo la imagen. Para orientarse en esto, desde la actividad principal se puede acceder a la actividad "ayuda ruidos", donde se muestran ejemplos de imágenes con cada cantidad de ruido anteriormente citadas. Una vez escogido una de las cantidades de ruido, se mostrará un intervalo de valores donde el parámetro h puede ser el adecuado, así como la vecindad y el radio de búsqueda adecuado. Decir que la cantidad de ruido se refiere a la desviación del ruido, ya sea para imágenes con ruido gaussiano como con ruido rician.

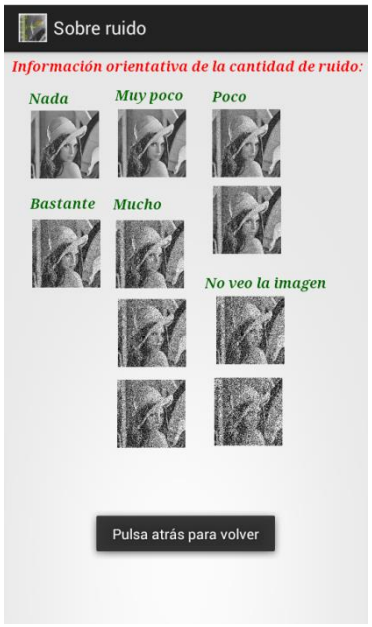
En lo correspondiente al procesamiento de la imagen se pueden emplear dos algoritmos. El que yo he implementado y la implementación del algoritmo Non Local Means de OpenCV. En caso de usar la mía, el tiempo de ejecución será mayor, pero los parámetros serán los adecuados. Debido a que el tiempo de ejecución puede ser elevado, para que el usuario pueda seguir utilizando el teléfono móvil, bastará con minimizarla y en el momento en que la imagen termine de ser procesada, se enviará una notificación al móvil, indicando que ya hemos terminado. Al pulsar sobre ella se abrirá la aplicación con la imagen obtenida. En caso de utilizar el de OpenCV, el tiempo de ejecución será más reducido, por lo que no será necesario mandar la notificación. Cuando termine también se pasará a otra actividad donde se muestre la imagen obtenida.

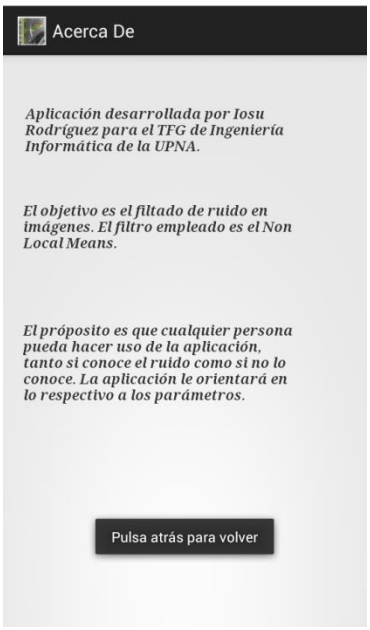
Manual de uso de NLMFilter App:

En primer lugar nos encontramos con la actividad principal. Si pulsamos sobre la imagen de la cámara, esta se activa y permite sacar una nueva imagen con la que trabajar. Si pulsamos sobre la imagen de la galería, podremos escoger una imagen. Una vez escogida, bajo el botón siguiente aparecerá la ruta de la imagen. Si arrastramos el dedo hacia la derecha, aparecerá el menú desplegable.



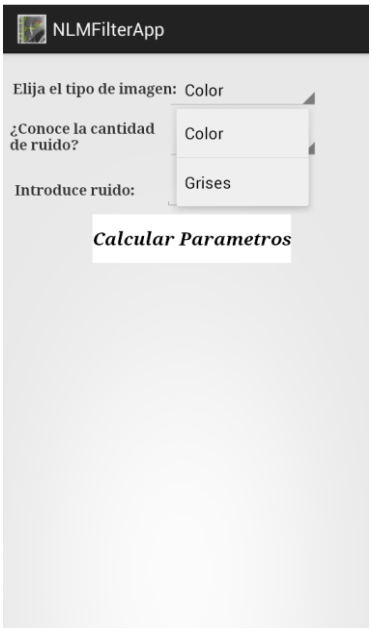
A continuación se muestran las actividades a las que llegamos desde cada una de las opciones del menú desplegable (a la izquierda Sobre ruido y a la derecha Acerca de):

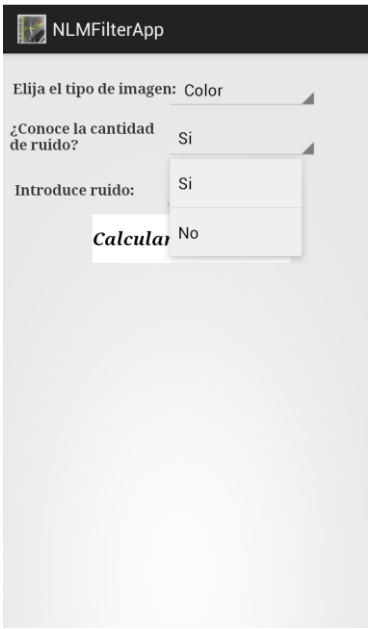




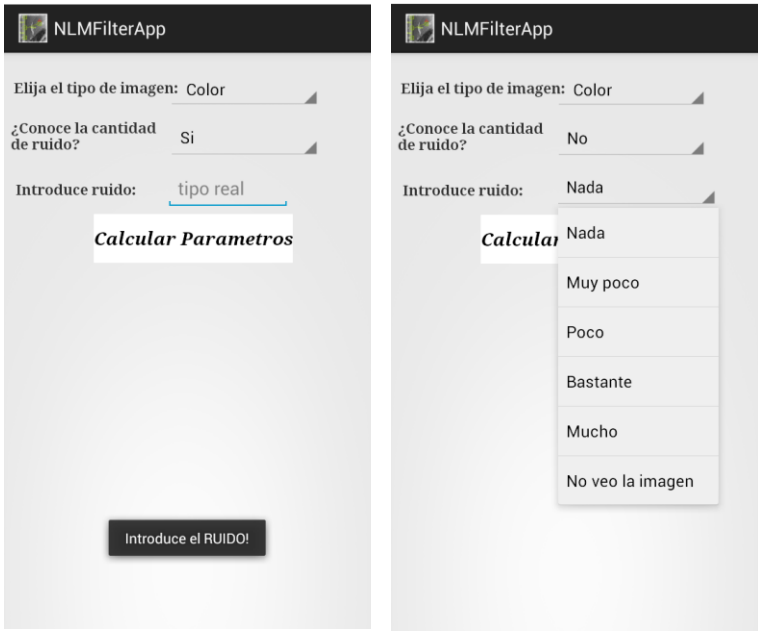
En la actividad sobre el ruido se muestran ejemplos de imágenes con cada cantidad de ruido, para orientar así a los usuarios que desconozcan la cantidad de ruido. En la actividad Acerca De, se muestra una breve descripción sobre la aplicación. Para volver a la actividad principal desde estas basta pulsar el botón del móvil de atrás. Al acceder a estas actividades se muestra un breve mensaje que indica como volver a la actividad principal.

Una vez escogida la imagen a filtrar, pulsando el botón siguiente, llegamos a la actividad donde indicar los parámetros. Cada menú desplegable se muestra también a continuación:



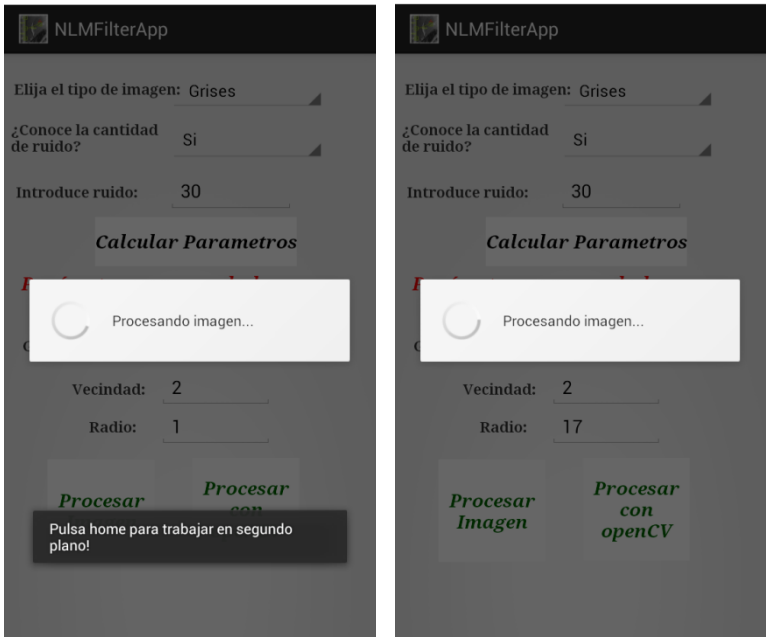


Podemos escoger entre si la imagen es a color o en escala de grises. Tambien podemos indicar si conocemos la cantidad de ruido. Si la conocemos basta con introducirla. Si no se introduce el ruido y pulsamos en el botón calcular parámetros, aparecerá un mensaje que indica que debemos introducir el ruido (imagen de la izquierda). Si no conocemos la cantidad de ruido se muestra otro menú desplegable (imagen de la derecha).

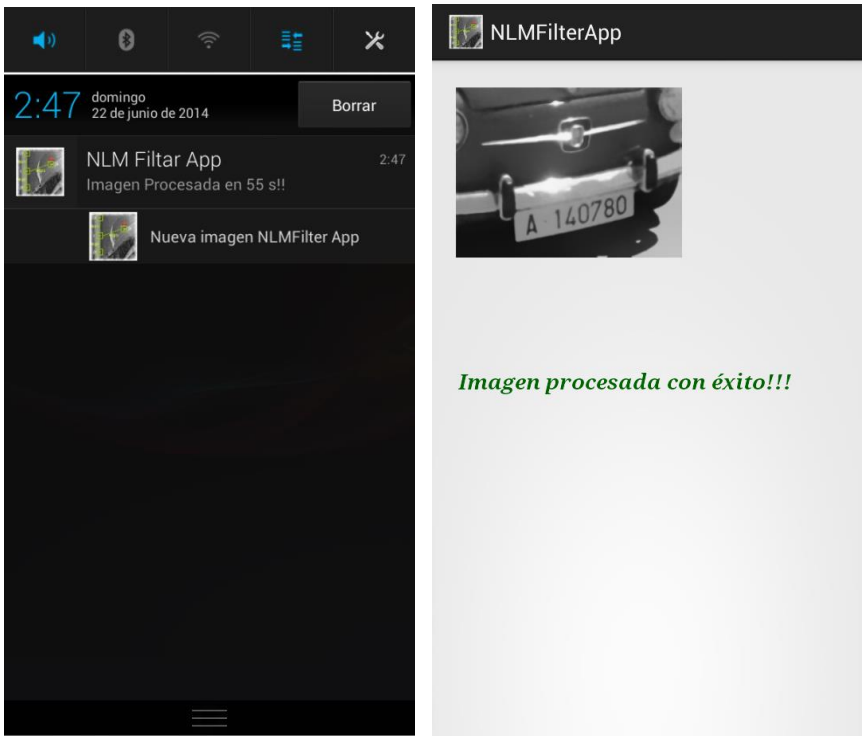


Si conocemos la cantidad de ruido, tras insertarla y pulsar el botón calcular parámetros, aparecerán los parámetros recomendados (imagen izquierda). Si no conocíamos la cantidad de ruido, aparecerá un intervalo de valores recomendados para h y el valor recomendado de la vecindad y del radio (imagen central). En cualquiera de los casos, si pulsamos cualquiera de los botones de filtrar y falta algún parámetro, se mostrará un mensaje (imagen derecha):

Una vez introducidos todos los parámetros, podremos elegir entre pocesar usando mi implementacion, o usando la de OpenCV. Si elegimos la primera opción, se mostrara un mensaje indicando que pulsemos home (para minimizar la aplicación), y así poder seguir usando el teléfono móvil mientras la imagen se procesa (imagen izquierda). Si elegimos la opcion de OpenCV habra que esperar unos segudos a que termine (imagen central):



Si habíamos pulsado el botón procesar imagen, una vez que termina, se mandará una notificación al móvil (imagen izquierda). Al pulsarle se abrirá de nuevo la aplicación mostrando el resultado (imagen derecha). Si habíamos pulsado sobre el botón procesar con OpenCV, se mostrará al terminar el resultado (imagen derecha):



Para salir de la aplicación bastará con darle al botón atrás hasta salir.

5.3. Problemas en el desarrollo de la aplicación

Antes de explicar la aplicación, veo adecuado explicar todos y cada uno de los problemas encontrados en el desarrollo.

En primer lugar decir que ya había trabajado con Android, por lo que no era completamente nuevo para mí. Por otro lado nunca había trabajado con Android para el procesamiento de imágenes, por lo que empecé a informarme sobre ello. Antes de comenzar con la aplicación empecé a hacer pequeñas pruebas para coger imágenes de la galería y tratar con ellas. Aquí es donde me encontré el primer problema. Cuando quería coger una imagen con una resolución grande, saltaba la excepción OutOFMemoryError. Esta excepción salta cuando la memoria dedicada para la aplicación se sobrepasa. Debido a la alta resolución de las imágenes, la aplicación no era capaz de soportar imágenes de ese tamaño. Tras investigar en internet, encontré la solución a esto en la página oficial de Android: http://developer.android.com/training/displaying-bitmaps/load-bitmap.html.

Ahí explica como cargar una imagen eficientemente en un bitmap (objetos de la clase bitmap que permiten cargar en ellos las imágenes). La solución consistió en hacer uso del código que ahí explicaba y adaptarlo finalmente a la aplicación.

Tras esto, empecé a implementar el algoritmo Non Local Means. Una vez implementado, lo instale en mi teléfono móvil Android. Aquí apareció el segundo problema, era muy lento. Esta lentitud era causada por distintas cosas: el procesador de un Smartphone no es tan potente como el de un computador, la complejidad del algoritmo, el lenguaje de programación, la forma de programa…

Buscando posibles soluciones, encontré unas librerías de código abierto hechas específicamente para temas relacionados con el procesamiento de imagen, OpenCV (Open Source Computer Vision Library). Esta escrito y optimizado en c/c++. También dispone de interfaces para java y Python y es compatible con Windows, Mac, Linux, iOs y Android. Para Android dispone de varias librerías muy útiles (en java). Estas librerías me han servido para realizar la implementación final del algoritmo algo más eficiente. Aun así, la aplicación no es lo rápida que quisiera.

Buscando más posibles soluciones me encontré con el NDK (native development kit) para Android. Esto permite desarrollar aplicaciones para Android empleando código nativo (en c y c++ por ejemplo). De esta forma podríamos hacer implementaciones tan eficientes como las propias de OpenCV pero exclusivamente en c o c++. También se podría hacer uso de las librerías hechas en c/c++ de OpenCV en vez de las escritas en java. Debido a las horas estipuladas para el proyecto, no ha sido posible realizar esto. La idea era implementar el algoritmo Non Local Means en c++ y haciendo uso del NDK y de las librerías de Android para utilizar código nativo, incorporarlo a la aplicación. De esta forma podría conseguir una aplicación realmente eficiente.

5.4. Líneas futuras y mejoras

Esta aplicación desarrollada es un primer boceto de una posible aplicación completa. Una buena idea sería hacer una aplicación que implemente diversos filtros y crear un pequeño editor de fotos.

Respecto al filtro Non Local Means, en caso de llevar adelante la aplicación, realizaría una implementación más eficiente empleando código nativo. También la parte en la que un usuario que no sepa qué es el ruido, pueda orientarse respecto a los parámetros más adecuados, se

podría realizar un sistema de recomendación. Para ello podría utilizarse una base de datos donde se hayan almacenado los parámetros que un usuario que haya quedado satisfecho con el resultado del filtrado, haya empleado.

Capítulo 6: Conclusiones y línea futuras

A continuación se resumen las conclusiones obtenidas con cada sección del experimento así como el futuro de la aplicación Android.

Conclusiones del estudio del tiempo de ejecución de las distintas versiones (sección 4.1.):

Es preferible trabajar con las versiones 3 y 4 que con las versiones 1 y 2. Las versiones 3 y 4 emplean radio de búsqueda mientras que la 1 y 2 recorren de nuevo la imagen completa por cada píxel de la imagen. Empleando las versiones 3 y 4 podemos trabajar con imágenes de tamaño más grande. Por ejemplo con un radio de búsqueda de 7, procesar una imagen de tamaño 512x512 tarda aproximadamente tres minutos. Sin embargo con las versiones 1 y 2, aplicadas a una imagen de tamaño 512x512, resulta imposible en tiempo de ejecución terminar el procesamiento de la imagen.

Conclusiones del estudio de los parámetros para las versiones 3 y 4. Selección de la mejor versión. (Sección 4.2.):

Para la versión 3, que emplea la distancia euclídea (sin ponderar), se sacan las siguientes conclusiones en lo referido a los parámetros:
Para una determinada vecindad, una vez alcanzamos el máximo SSIM, los resultados empeoran a medida que aumentamos el radio de búsqueda. Lo mismo sucede al aumentar la vecindad.
Cuanto más ruido tenga la imagen, necesitaremos vecindades y radios más grandes.
Valores muy altos de h, hacen que se filtre ruido en exceso, perdiendo detalles de la imagen. Si los valores son muy bajos, no se filtra apenas ruido.
El valor de h depende de la cantidad de ruido que tenga la imagen. A más ruido, se necesitan valores de h más grandes.
La tabla resumen de parámetros en función del ruido es la siguiente:

	h	vecindad	radio
0<ruido<=15	5*ruido	2	5
15<ruido<=30	6*ruido	3	7
30<ruido<=45	5,75*ruido	3	10
45<ruido<=75	5,5*ruido	3	10
75<ruido<=100	5,35*ruido	3	10

En imágenes con ruido rician se pierde contraste cuando la imagen tiene mucho ruido. Para solucionar esto, sería necesario realzar el contraste una vez filtrado el ruido.

Para la versión 4, que emplea la distancia euclídea ponderada por un Kernel gaussiano, se sacan las siguientes conclusiones:
Es preferible emplear un Kernel gaussiano con desviación 5 (o parecida), puesto que si la desviación del Kernel es muy baja, cuando la vecindad es grande, los píxeles más lejanos al central se ponderan con valores próximos a 0, obteniendo así imágenes peores.
Respecto a los parámetros, sucede lo mismo que con la versión 3. Algunas diferencias entre ambas versiones son:
Para la versión 4 los valores de h necesarios son muy inferiores. Esto es debido a que al ponderar los píxeles cuando se calcula la distancia euclídea ponderada, obtenemos valores menores que en la versión 3, por lo que el valor de h, que es el que divide esto, también tendrá que ser más pequeño.
Combinaciones de vecindades y radios que para la versión 3 funcionan mal, para la versión 4 se consiguen resultados aceptables.
En lo que respecta a los resultados finales de la versión 4, el SSIM es algo más pequeño (diferencias de milésimas) que en los resultados finales de la versión 3. Visualmente apenas se

nota diferencia. Pese a esto, la versión escogida es la versión 4, ya que es más versátil y se adapta mejor que la versión 3 a diversas combinaciones de parámetros.
La tabla resumen de parámetros es la siguiente:

	Kernel desviación 5		
	h	v	r
0<ruido<=15	ruido	2	5
15<ruido<=30	ruido	2	7
30<ruido<=45	0,75*ruido	4	10
45<ruido<=75	0,8*ruido	3	10
75<ruido<=100	0,75*ruido	4	10

Conclusiones de la adaptación a imágenes a color (sección 4.3.):

Esta versión funciona igual que la versión 4 pero está adaptada para trabajar también con imágenes a color.

El tiempo de ejecución aumenta con respecto a imágenes en escala de grises. Sin embargo para obtener los mejores resultados se necesitan radios menores que con imágenes en escala de grises. Por ello el tiempo de ejecución para obtener buenos resultados es semejante.
Respecto a los parámetros e imágenes con ruido rician sucede lo mismo que en las versiones anteriores.
Tabla resumen de parámetros:

	h	vecindad	radio
0<ruido<=45	ruido	2	5
45<ruido<=75	0,8*ruido	3	7
75<ruido<=90	0,75*ruido	3	7
90<ruido<=100	0,75*ruido	4	7

Conclusiones de la Comparación de resultados con otros filtros (sección 4.4.):

Tras la comparación del filtro NLMs con el filtro de la media aritmética y el filtro gaussiano, se concluye que el NLMs proporciona mejores resultados, pero el tiempo de ejecución también es mucho más elevado. Por lo tanto tendrá que ser el usuario quien decida que le compensa más, si unos buenos resultados invirtiendo más tiempo, o resultados peores en menor tiempo.

Futuro de la aplicación Android NLMFilter App:

Esta aplicación desarrollada es un primer boceto de una posible aplicación completa. Una buena idea sería hacer una aplicación que implemente diversos filtros y crear un pequeño editor de fotos.

Respecto al filtro Non Local Means, en caso de llevar adelante la aplicación, realizaría una implementación más eficiente empleando código nativo. También la parte en la que un usuario que no sepa qué es el ruido, pueda orientarse respecto a los parámetros más adecuados, se podría realizar un sistema de recomendación. Para ello podría utilizarse una base de datos donde se hayan almacenado los parámetros que un usuario que haya quedado satisfecho con el resultado del filtrado, haya empleado.

Bibliografía y referencias

Sobre algoritmo Non Local Means:

Artículo de referencia: http://bengal.missouri.edu/~kes25c/nl2.pdf

http://www.ipol.im/pub/art/2011/bcm_nlm

Para tipos de ruido y filtros:

Apuntes de la asignatura Visión Artificial, impartida en la UPNA.

Para desarrollo en Android:

http://developer.android.com/index.html

http://stackoverflow.com/

http://www.droiddraw.org/

http://www.nacho-alvarez.es/index.php/blog/2012/05/02/conectar-programas-cc-con-aplicaciones-android/

http://opencv.org/

http://docs.opencv.org/java/

http://www.aprendeandroid.com/menu.htm

http://www.javaya.com.ar/androidya/

http://www.maestrosdelweb.com/editorial/curso-android/

Para desarrollo en Matlab:

http://www.mathworks.es/products/matlab/