ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

FACULTAD DE INGENIERIA EN ELECTRICIDAD Y COMPUTACIÓN

“Análisis e implementación de algoritmos para distorsionar imágenes con distintos tipos de ruido y aplicación de filtros en dos dimensiones para restaurarlas”

REPORTE DE MATERIA DE GRADUACIÓN

Previo a la obtención del Título de:

INGENIERO EN ELECTRÓNICA Y TELECOMUNICACIONES

Presentado por:

Carlos César García Chilán

Juan Salomón Viteri Paredes

GUAYAQUIL – ECUADOR

AÑO: 2009

AGRADECIMIENTO

A Dios por la culminación de este trabajo, a nuestros profesores, en especial a la Ing. Patricia Chávez por su ayuda y consejos.

DIDECATORIA

A Dios

A mis padres por su sacrificio

A mis hermanos

Carlos García Chilán

A mis padres

A mis hermanos

Juan Viteri Paredes

TRIBUNAL DE SUSTENTACION

MSc. Patricia Chávez
PROFESOR DE LA MATERIA

Ing. Juan Carlos Avilés

PROFESOR DESIGNADO

DECLARACIÓN EXPRESA

“La responsabilidad del contenido de esta Tesis de Grado, nos corresponde exclusivamente; y el Patrimonio intelectual de la misma a la ESCUELA SUPERIOR POLITECNICA DEL LITORAL” (Reglamento de Graduación de la ESPOL)

Carlos García Chilán

Juan Viteri Paredes

RESUMEN

El presente trabajo analiza ciertos algoritmos que sirven para la degradación de imágenes (ruido) y a su vez la recuperación de las mismas por medio de filtros predefinidos en dos dimensiones. El filtrado se convierte en una herramienta muy importante en situaciones reales. Se realizará un análisis cualitativo utilizando el Error Cuadrático Medio (MSE) el cual nos ayudará a determinar cual filtro es más eficiente que otro, y por último, se procederá a realizar un análisis cuantitativo utilizando para esto la percepción visual de un grupo de personas que nos ayudarán a determinar cual filtro es mas conveniente para la recuperación de las imágenes y se procederá a comparar ambos análisis.

En este trabajo utilizaremos los siguientes ruidos: el Ruido Gaussiano, el Ruido Poisson, el Ruido Salt & Pepper, y el Ruido Speckle que degradarán las imágenes, los cuales serán analizados en el capítulo 2. Como filtros de recuperación utilizaremos: Filtro Average, Filtro Disk, Filtro Gaussiano, Filtro Motion y el Filtro Unsharp, analizados en el capitulo 3.

INDICE GENERAL

CAPÍTULO 1

Conceptos Generales

CAPÍTULO 2

Tipos de Ruido

CAPÍTULO 3

Filtros de Restauración

CAPÍTULO 4

Desarrollo del proyecto

ANEXOS

INDICE DE FIGURAS

INDICE DE TABLAS

INTRODUCCION

La supresión de ruido es importante ya que cualquier sistema de visión artificial comienza con el procesamiento de los valores de intensidades. En este trabajo veremos algunas modelos de ruido y su filtrado (eliminación).

La cantidad de ruido se mide normalmente por la media y la varianza del ruido, lo cual nos puede ayudar en el entendimiento de cómo afecta en una situación real debido a muchos fenómenos que se escapan a nuestra voluntad. Para simular un proceso de captación de imágenes con ruido utilizaremos la herramienta Matlab 7.7.0 (R2008b), el cual nos facilitará un entorno gráfico para la manipulación de la imagen, permitiendo añadirle ruido y su respectivo filtrado para recuperarla.

Se analizarán brevemente los distintos tipos de ruido que utilizaremos para degradar las imágenes y se incluirá un capítulo para estudiar los filtros que se utilizarán, estos algoritmos son parte de la librería de Matlab 7.0.0 por lo cual no es necesario implementarlos, y se puede proceder directamente a una verificación visual de sus resultados.

Como parámetro de medición teórica de eficiencia del proceso de restauración, se usará el Error Cuadrático Medio, y se obtendrán conclusiones acerca de cada proceso para la imagen, y luego se incluirá una encuesta a un grupo de por lo menos 40 personas para analizar la percepción visual de cada una y comparar con los resultados analíticos.

Se incluirá un manual de ayuda que permitirá a cualquier usuario manejar el programa teniendo conocimientos previos de procesamiento digital de señales ya que se permitirá escoger el algoritmo de degradación de imágenes y el filtro para su recuperación.

CAPÍTULO 1

Conceptos Generales

Las señales desempeñan un papel fundamental en la vida diaria la voz, la música, la temperatura, las imágenes y el vídeo son ejemplos de señales que se encuentran frecuentemente en la vida cotidiana. Una señal es una función de una o más variables independientes como el tiempo, la velocidad, la humedad y la presión entre otras.

El procesamiento digital de señales permite realizar análisis de imágenes, audio o video de tal forma que se pueda ejecutar diversas aplicaciones.

Sus principales objetivos son:

- Mejorar la calidad visual para permitir la interpretación humana.
- Extraer información de las imágenes para que pueda ser entendida por el ordenador.
- Manipular una imagen de tal forma que se ajuste a los requerimientos.

1.1 Ruido

Es la información no deseada que contamina una imagen.

$$g(x,y)=f(x,y)+r(x,y)$$

El origen puede estar tanto en el proceso de adquisición de la imagen (errores en los sensores), como en el de transmisión (debido a interferencias en el canal de transmisión).

El ruido es un problema propio de la naturaleza y está asociado a cualquier toma de datos de la realidad. El problema de su eliminación es muy común, tanto en la adquisición de señales, como en el de imágenes. El ruido en las señales electrónicas es ocasionado por diversas causas, las cuales muchas veces no se pueden evitar. Asimismo en el tratamiento de imágenes, se presenta como una fluctuación aleatoria de los valores en los pixeles que componen dicha imagen.

1.2 Degradación de imágenes

La degradación de una imagen se define como un proceso mediante el cual se puede introducir ruido a esta de alguna forma para alterar su estado natural.

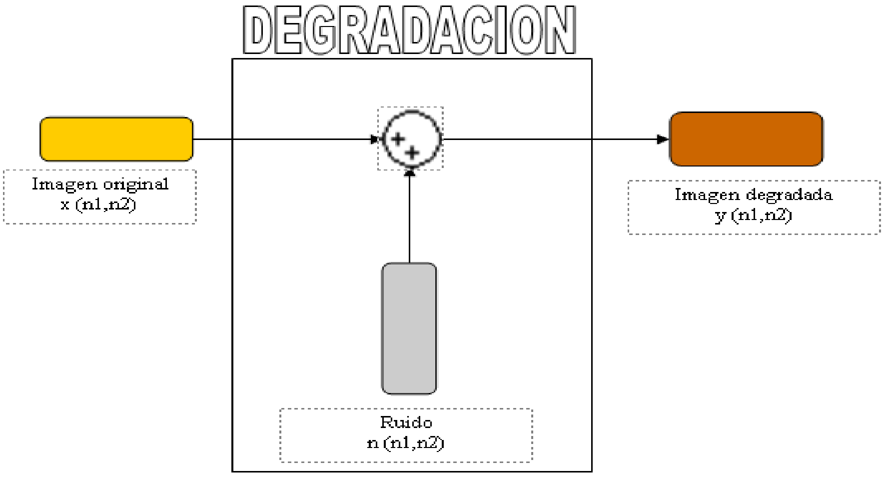


Fig. 1.2 Diagrama de bloques del proceso de distorsión

1.3 Restauración de imágenes

La restauración de imagen se define, como un proceso que nos permite recuperar una parte deteriorada de la imagen o que tiene algún objeto que la oculta, con el objetivo de mejorar su calidad.

Es un tipo de operación que altera el valor de un píxel en función de los valores de los píxeles que le rodean, también se le denomina procesamiento basado en la vecindad u operación de vecindad. En general, restaurar una imagen consiste en aplicar una transformación de forma que se acentúen o disminuyan ciertos aspectos que distorsionan la imagen original.

1.4 Error Cuadrático Medio (MSE)

El error cuadrático medio consiste en la suma de las diferencias al cuadrado entre lo real y lo proyectado por el modelo

p= valor proyectado, r = valor real N = tamaño de la muestra

En nuestro caso utilizaremos el Error Cuadrático Medio para proyectar que tanto se ha degradado la imagen original con respecto al ruido aplicado, y realizar el mismo procedimiento para la imagen filtrada (recuperada) y la imagen original.

CAPÍTULO 2

Tipos de Ruido

En esta sección vamos a estudiar brevemente los tipos de ruido más usuales que se pueden presentar en una imagen.

2.1 Ruido Gaussian

Un tipo de ruido que se presenta con frecuencia es el ruido Gaussiano cuya función de densidad de probabilidad está expresada por la curva gaussiana, la cual unidimensionalmente está definida como:

(1)

De la ecuación (1), μ es la media σ es la desviación típica de la variable aleatoria. Este ruido es muy usado ya que se asemeja mucho a los casos reales. Se caracteriza por tener un espectro de energía constante para todas las frecuencias. Cuando se presenta este problema, el valor exacto de cualquier píxel es diferente cada vez que se captura la misma imagen. Este efecto, suma o resta un determinado valor al nivel de gris real y es independiente de los valores que toma la imagen.

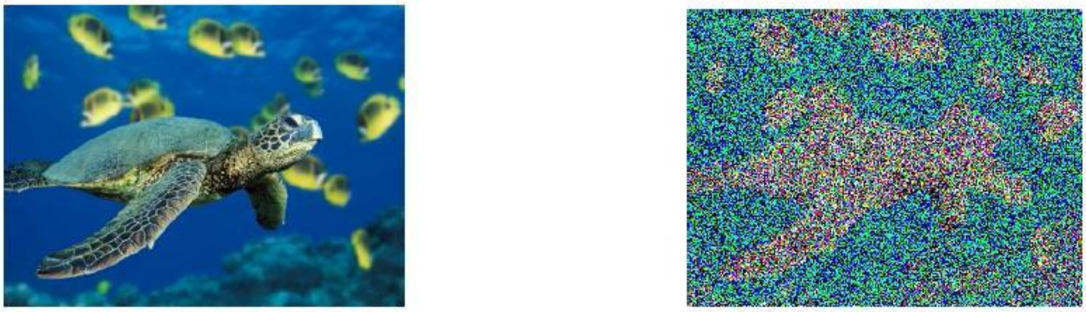


Imagen original

Imagen transformada

Fig. 2.1 Ejemplo de ruido Gaussiano

2.2 Ruido Poisson

En teoría de probabilidad y estadística, la distribución de Poisson es una distribución de probabilidad discreta que expresa la probabilidad de un número de eventos que ocurren en un período fijo de tiempo si estos eventos ocurren con una media proporción conocida e independientemente del tiempo desde el último evento.

$$f(k;\lambda)=\frac{\lambda^k e^{-\lambda}}{k!}, \quad (2)$$

Donde:

k es el número de ocurrencias de un evento.

λ un es un número real positivo, igual al número esperado de ocurrencias que suceden durante el intervalo dado.

Así, el número de ocurrencias observadas fluctúa sobre su media λ con una desviación estándar$\sigma_k = \sqrt{\lambda}$. Estas fluctuaciones se denotan como el ruido de Poisson o (particularmente en la electrónica) como el ruido del tiro (shot noise).

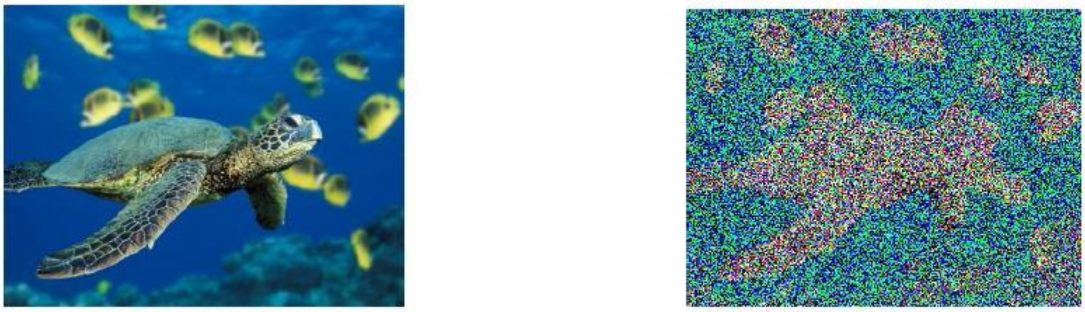


Imagen original

Imagen transformada

Fig. 2.2 Ejemplo de ruido Poisson

2.3 Ruido Salt & Pepper

Este ruido es producido por el mal funcionamiento de los pixeles en los sensores de una cámara, en situaciones de memoria defectuosas en el hardware o por transmisión de un canal de ruido.

Para las imágenes degradadas por ruido Salt & Pepper, los pixeles de ruido solo pueden tomar los valores máximo y mínimo (cualquier valor aleatorio) en el rango dinámico. Una forma de producirlo es tomando varios pixeles aleatoriamente y reemplazar este con valores de cero en la escala de color (color negro).

$$G(i,j) = \begin{cases} I(i,j) & si\ x < p \\ A & si\ p \leq x < q \\ B & si\ q \leq x < 1 \end{cases}$$

Donde x~U [0,1), 0≤p ≤q ≤1, A y B son número fijos, usualmente A=255 y B=0. Observemos que (q-p) es la probabilidad de pasar al valor A (255 y por tanto sal) y (1-q) es la probabilidad de pasar a B (0 y por tanto pimienta), p es la probabilidad de permanecer igual.

Imagen original

Imagen transformada

Fig. 2.3 Ejemplo de ruido Salt & Pepper

2.4 Ruido Speckle

El ruido *Speckle* aparece en imágenes con iluminación coherente, como las de scanner ultrasónico, sonar y radar de abertura sintética (SAR). Este ruido se desvía del modelo clásico, en el que se supone ruido Gaussiano, independiente de la señal y adicionado al verdadero valor. El *Speckle*, en cambio, es multiplicativo y no Gaussiano (en los formatos intensidad y amplitud), y dificulta la interpretación de las imágenes porque el "efecto de sal y pimienta" corrompe la información o verdad del terreno.

Existen numerosas técnicas para extraer información contenida en imágenes con *Speckle*, entre las cuales las estadísticas proveen los mejores modelos y herramientas para el procesamiento y análisis de imágenes ruidosas. El procedimiento usual consiste en proponer un modelo, contrastar la hipótesis con datos reales (etapa muchas veces evitada por la necesidad de obtener resultados rápidos y por la tradición de considerar verdadera la distribución Gaussiana) y, una vez aceptado el modelo, hacer inferencia acerca de los parámetros desconocidos del mismo. Ciertas decisiones médicas, ambientales y estratégicas están condicionadas por estos parámetros

Imagen original

Imagen transformada

Fig. 2.4 Ejemplo de ruido Speckle

CAPÍTULO 3

Filtros de Restauración

Los filtros se usan, principalmente, para eliminar altas o bajas frecuencias de la imagen, es decir, para suavizar la imagen, o bien, para realzar o detectar bordes.

Una imagen se puede filtrar en el dominio del espacio, trabajando directamente sobre los píxeles de la imagen, o en el dominio de la frecuencia, donde las operaciones se llevan a cabo en la transformada de Fourier de la imagen.

3.1 Filtro Average

El filtro moving Average es el filtro más común en los DSP, en la mayoría de las veces se debe a ser el filtro digital más fácil de entender y usar. A pesar de su simplicidad, el filtro de la media móvil es óptimo para una tarea común como puede ser la reducción de un ruido fortuito mientras retiene una respuesta de paso muy bien definida. Esto hace que este filtro predomine en el campo de las señales codificadas en el dominio del tiempo, con la posibilidad de separar una banda de frecuencias de otra. Relacionado con el este tipo de filtro se puede encontrar la media móvil Gaussiana, de Blackman y de múltiple-paso. Estos confieren una mejor ejecución en el dominio de la frecuencia, con un mayor gasto del tiempo de CPU.

Como su propio nombre indica, el filtro de la media móvil opera mediante el promedio de un número de puntos de la señal de entrada para producir cada punto de la señal de salida. En una formula matemática queda como:

$$y[i]=\frac{1}{M}\sum_{j=0}^{M-1}x[i+j] \qquad (3)$$

Donde se encuentra la señal de entrada, la señal de salida y M que es el número de puntos en la media.

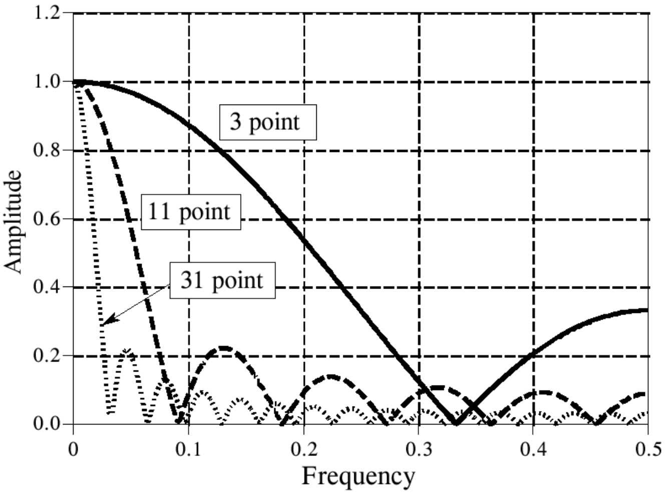


Fig. 3.1 Variación del filtro Average según la media M

3.2 Filtro Disk

Este es un tipo de filtro pasa-bajo llamado filtro de media, presenta la ventaja de que como medida estadística, la mediana es menos sensible a valores extremadamente desviados y se modifican menos los valores originales, ya que la mediana es en principio, uno de los valores concretos de la ventana de filtrado.

Este tipo de filtro asigna al pixel central la media de todos los pixeles incluidos en la ventana. La matriz de filtrado estaría compuesta por unos y el divisor sería el número total de elementos en la matriz.

Su efecto es el difuminado o suavizado de la imagen y se aplica junto con el de mediana para eliminar ruidos.

¿Cómo funciona? : Se visita cada píxel de la imagen y se reemplaza por la media de los píxeles vecinos. Se puede operar mediante convolución con una máscara determinada.

$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{9}$
$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{9}$
$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{9}$

Fig. 3.2 Ejemplo de máscara 3x3 para el filtro de la media (disk)

3.3 Filtro Gaussiano

En la electrónica y procesamiento de señales, un filtro Gaussiano es un filtro cuya respuesta al impulso es una función Gaussiana. Matemáticamente, un filtro Gaussiano modifica la señal de entrada por convolución con una función Gaussiana.

El filtro Gaussiano en una dimensión tiene la siguiente respuesta al impulso:

$$g(x) = \sqrt{\frac{a}{\pi}} \cdot e^{-a \cdot x^2} \quad (4)$$

O con la desviación estándar como parámetro:

$$g(x) = \frac{1}{\sqrt{2 \cdot \pi} \cdot \sigma} \cdot e^{-\frac{x^2}{2\sigma^2}}. \quad (5)$$

En dos dimensiones el filtro tiene la siguiente respuesta al impulso:

$$g(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (6)$$

El filtro Gaussiano se usa para suavizar la imagen: a mayor σ^2 el suavizado es mayor.

3.4 Filtro Motion

El filtro de movimiento nos permite detectar cambios de color en la imagen debido al desplazamiento de los objetos. La información extraída de la aplicación de este filtro permite a los algoritmos de localización 3D estimar la posición de objetos que se mueven.

La aplicación de este filtro consiste en restar a cada pixel su valor anterior de RGB para cada una de las imágenes recibidas. Para ello, es necesario almacenar la imagen que se vaya a analizar, de forma que se pueda contrastar cada pixel con el mismo en la imagen original. De esta forma, la diferencia entre ambas imágenes dará como resultado los pixeles que han sufrido variación debida al movimiento.

3.5 Filtro Unsharp

Los filtros anteriores son filtros en cierta medida extremos ya que enfatizan bien las componentes de frecuencia baja, o bien las componentes de frecuencia alta. Los primeros según hemos visto, reducen el ruido a costa de difuminar las transiciones de la imagen.

Los segundos, por su parte, enfatizan tales transiciones, pero a base de eliminar las componentes con intensidad homogénea.

Un filtro mezcla de ambos comportamiento es el que se ha llamado tradicionalmente *Unsharp masking*, debido a que opera eliminando de la imagen original una parte de su correspondiente imagen paso bajo (esta última sería la imagen con fronteras relajadas, de ahí el adjetivo de no afilado-Unsharp- del filtro).

Denominando O a la imagen original, HPO a la versión paso alto de la imagen original, LPO a su versión paso bajo, y UMO a la filtrada mediante este filtro, podemos formalizar el procedimiento de la manera siguiente:

$$O = HPO + LPO$$

$$UMO = A * O - LPO$$

$$UMO = A * O - (O - HPO)$$

Por tanto, si la constante A es mayor que 2 el filtro Unsharp masking equivale a añadir a la versión original amplificada de la imagen una parte de la componente paso alto de la misma.

En este caso, se conseguirán enfatizar las transiciones sin perder información de la estructura original de la imagen. Si concretamos esto en unas operaciones con máscaras podemos llegar a una que corresponde a este tipo de filtrado. Por ejemplo,

$$\frac{1}{9}\begin{bmatrix} 0 & 0 & 0 \\ 0 & 9A & 0 \\ 0 & 0 & 0 \end{bmatrix} - \frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} = \frac{1}{9}\begin{bmatrix} -1 & -1 & -1 \\ -1 & 9A-1 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

CAPÍTULO 4

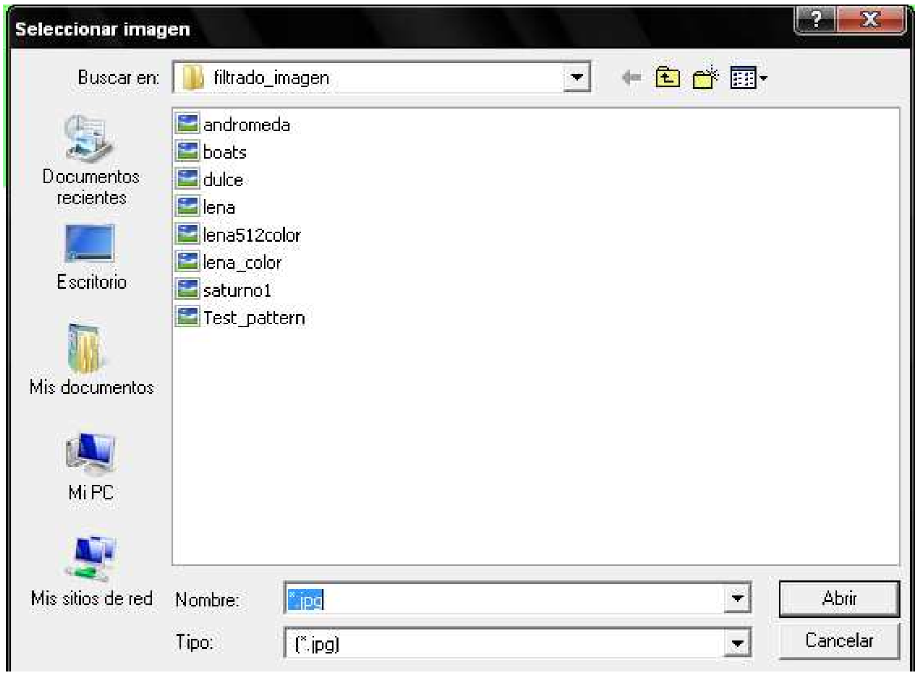
Desarrollo del Proyecto

Al inicio, se planteó usar un grupo determinado de imágenes que según sus características nos servirían para entender de una mejor manera cada ruido que se pensaba aplicar, así como los filtros de recuperación. Luego se llegó a la conclusión de que nuestro programa debía ser capaz de utilizar cualquier tipo de imagen sin importar sus características puesto que el objetivo de este trabajo es el análisis de los resultados para cada tipo de ruido y de la misma manera para cada filtro, sin centrarnos en las características individuales de cada imagen.

4.1 Manipulación de las imágenes

Como nuestro programa utilizará un entorno grafico que se creará utilizando MATLAB, herramienta computacional que entre sus prestaciones básicas se hallan: la manipulación de matrices, la representación de datos y funciones, la implementación de algoritmos, la creación de interfaces de usuario (GUI), etc. Se investigó una función que nos permita abrir cualquier imagen ubicada en cualquier ruta de la computadora en la que se vaya a trabajar y no necesariamente en la carpeta de trabajo en donde se encuentre los códigos fuentes de nuestro programa.

Esta función es `uigetfile` que en el momento de ser ejecutada nos mostrará un cuadro de dialogo ABIR, tal como indica la figura:



Seleccionar imagen

Buscar en: filtrado_imagen

andromeda
boats
dulce
lena
lena512color
lena_color
saturno1
Test_pattern

Nombre: *.jpg

Tipo: (*.jpg)

Fig. 4 Cuadro de dialogo para abrir archivos

Se guardarán en dos variables el nombre y la ruta en donde está ubicada la imagen que se desea abrir.

Luego de que obtenemos nuestra imagen a ser procesada, procederemos a describir cómo funciona nuestro programa.

4.2 Funcionamiento del Programa

El programa esta diseñado con una interfaz gráfica, en la cual el usuario tiene como primera opción, abrir una imagen ubicada en cualquier directorio. Luego de esto se podrá seleccionar el tipo de imagen (a color o blanco y negro).

El programa le mostrará al usuario la imagen a ser analizada, luego de esto podrá escoger el tipo de ruido que le desea aplicar a la imagen y los parámetros que cada uno de estos requieren para funcionar (parámetros como la media y la varianza). Al mismo tiempo se mostrará en pantalla el Error Cuadrático Medio, que le indicará al usuario que tan degradada ha sido la imagen original con respecto al ruido aplicado. Esta nueva imagen con ruido será mostrada en pantalla.

Luego de esto el usuario podrá seleccionar el tipo de filtro con el que desea recuperar la imagen, junto a esto se mostrará el Error Cuadrático Medio que compara la imagen recuperada con la original. Esta imagen también será mostrada en pantalla.

4.3 Resultados obtenidos

Realizaremos el análisis teórico utilizando el programa en seis fotos (seis a color y sus respectivas en blanco y negro), pero por cuestiones de espacio, solo presentaremos el análisis de dos imágenes (una a color y otra a blanco y negro) de la imagen de Lena proporcionada por la Ing. Patricia Chávez en clases. Se aplicaron todos los ruidos a las imágenes calculándose con el programa el Error Cuadrático Medio (MSE) de las imágenes originales con las imágenes distorsionadas. En este caso, el ruido que menor distorsión generó de acuerdo al MSE fue el ruido tipo Salt & Pepper (usando una densidad de 0.05).

Luego se procedió a aplicarle los filtros a todas las imágenes distorsionadas calculando para cada imagen el MSE respectivo, observándose que los filtros que mejor recuperan las imágenes son los filtros tipo Average y Disk (observar tabla).

El resultado analítico se muestra en la Tabla 1 con los nombres de las imágenes utilizadas, todos los MSE para cada ruido y así mismo para los filtros aplicados.

		FILTROS DE RESTAURACION					
Imágenes	Ruido aplicado	MSE	Average MSE	Disk MSE	Gaussiano MSE	Motion MSE	Unsharp MSE
baboon	Gaussiano	86	65	77	71	76	117
baboon_bn		87	66	77	72	76	118
baboon	Poisson	46	54	75	38	68	113
baboon_bn		47	54	76	39	68	112
baboon	Salt & Pepper	6	64	78	31	74	98
baboon_bn		6	65	78	32	75	98
baboon	Speckle	90	71	81	75	82	123
baboon_bn		94	72	78	79	81	124
airplane	Gaussiano	86	52	60	70	61	116
airplane_bn		87	54	62	71	61	116
airplane	Poisson	57	34	57	38	45	111
airplane_bn		56	35	59	38	45	111
airplane	Salt & Pepper	6	60	74	30	68	65
airplane_bn		6	61	76	30	68	65
airplane	Speckle	104	78	73	95	85	121
airplane_bn		103	76	72	93	83	121
dark_stream	Gaussiano	73	27	34	46	34	103
dark_stream_bn		72	27	35	44	34	101
dark_stream	Poisson	16	17	38	13	28	77
dark_stream_bn		15	18	39	13	28	75
dark_stream	Salt & Pepper	5	17	31	11	26	65
dark_stream_bn		5	17	32	11	27	63
dark_stream	Speckle	25	23	39	20	33	82
dark_stream_bn		23	24	39	20	33	79

Tabla I. Resultados analíticos para las imágenes a color y a blanco y negro

		FILTROS DE RESTAURACION					
Imágenes	Ruido aplicado	MSE	Average MSE	Disk MSE	Gaussiano MSE	Motion MSE	Unsharp MSE
fruits	Gaussiano	86	79	58	70	61	113
fruits_bn		87	48	54	71	59	115
fruits	Poisson	51	28	51	34	41	104
fruits_bn		54	26	49	36	40	108
fruits	Salt & Pepper	6	49	62	25	59	65
fruits_bn		6	52	63	27	61	61
fruits	Speckle	97	74	75	88	82	117
fruits_bn		100	72	68	91	80	119
peppers	Gaussiano	83	46	59	65	60	111
peppers_bn		85	46	61	67	61	114
peppers	Poisson	41	28	59	28	46	99
peppers_bn		43	27	59	29	46	103
peppers	Salt & Pepper	6	42	62	21	56	67
peppers_bn		6	43	64	22	58	67
peppers	Speckle	79	53	63	66	65	112
peppers_bn		84	53	63	71	67	116
Lena	Gaussiano	86	39	31	67	48	117
Lena_bn		86	43	46	65	61	118
Lena	Poisson	45	18	28	26	31	107
Lena_bn		38	26	46	26	49	105
Lena	Salt & Pepper	6	38	36	19	47	68
Lena_bn		6	39	48	19	58	75
Lena	Speckle	88	45	40	70	54	118
Lena_bn		76	42	49	59	61	117

4.4 Resultados visuales

Se procedió a consultar a 50 personas entre 18 y 50 años para que observen según su apreciación visual, cual es el filtro que mejor recupera la imagen.

Para esto se procedió, primeramente a cargar todas las imágenes a color y aplicarle todos los ruidos, y luego recuperarla con todos los tipos de filtro existentes en el programa. Los resultados se muestran en la Tabla II.

Por ultimo, se repitió el mismo procedimiento para las imágenes a blanco y negro. A continuación se muestra lo que las personas consultadas observaron solo para la foto de Lena:

Fig. 4.1 Imágenes originales

Imagen Original Lena a Color

Imagen Original Lena a Blanco y Negro

Fig. 4.2 Ruido Gaussiano aplicado a las dos Imágenes

Fig. 4.3 Recuperación con Filtro Average

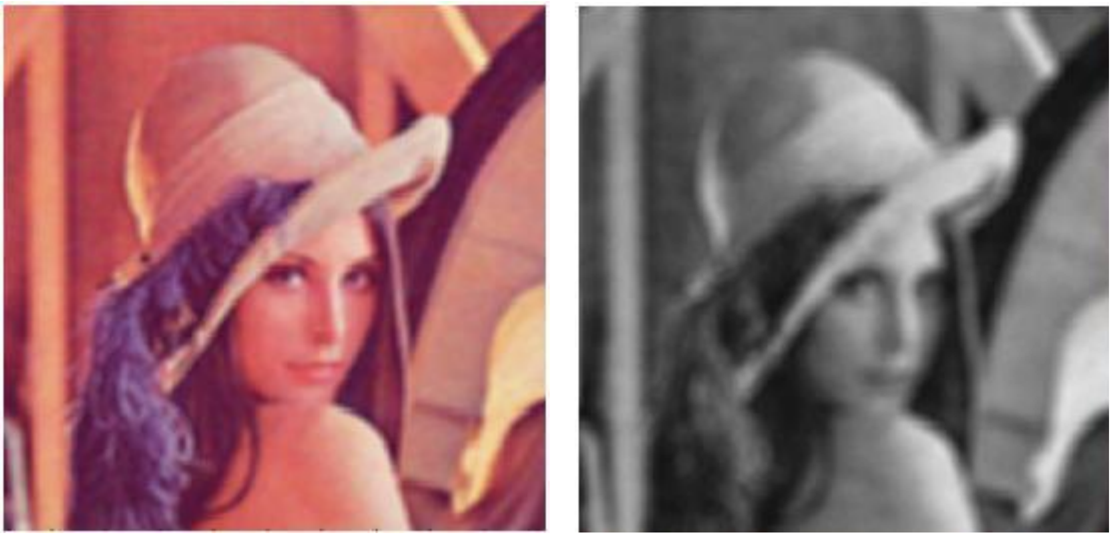


Fig. 4.4 Recuperación con Filtro Disk

Fig. 4.5 Recuperacion con Filtro Gaussiano

Fig. 4.6 Recuperación con Filtro Motion

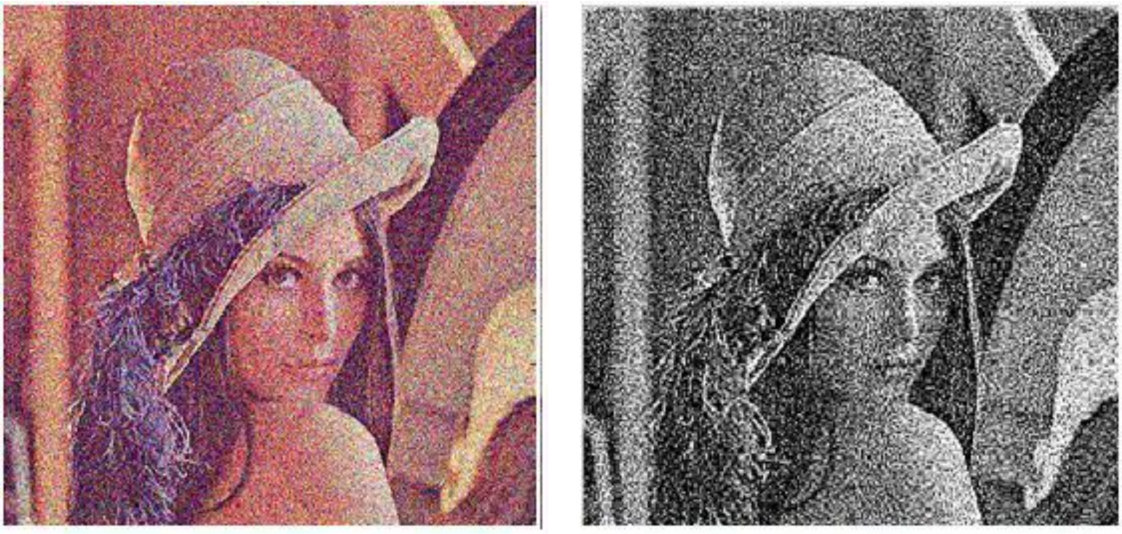


Fig. 4.7 Recuperación con Filtro Unsharp

Fig. 4.8 Ruido Poisson aplicado a las dos Imágenes

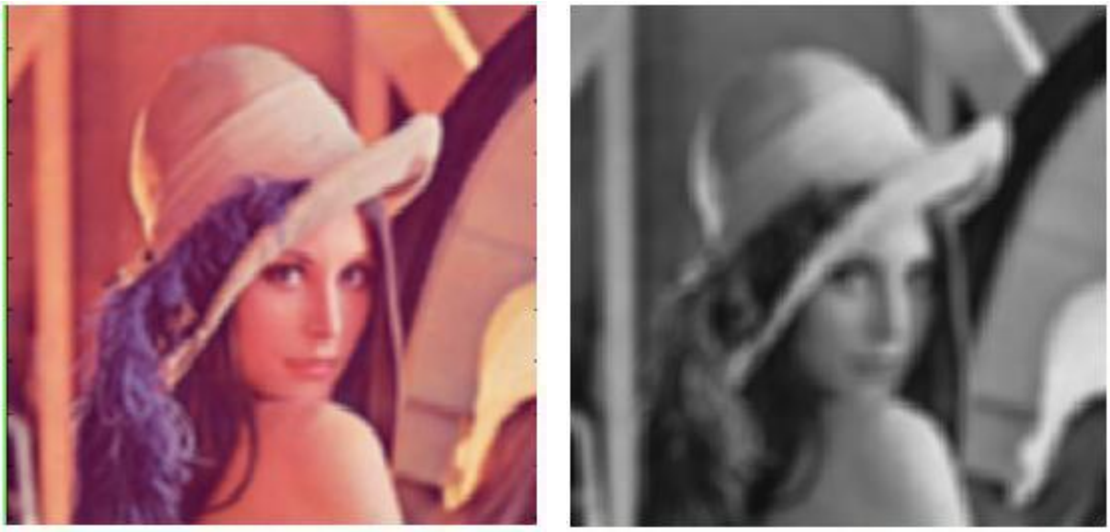


Fig. 4.9 Recuperación con Filtro Average (para ruido Poisson)

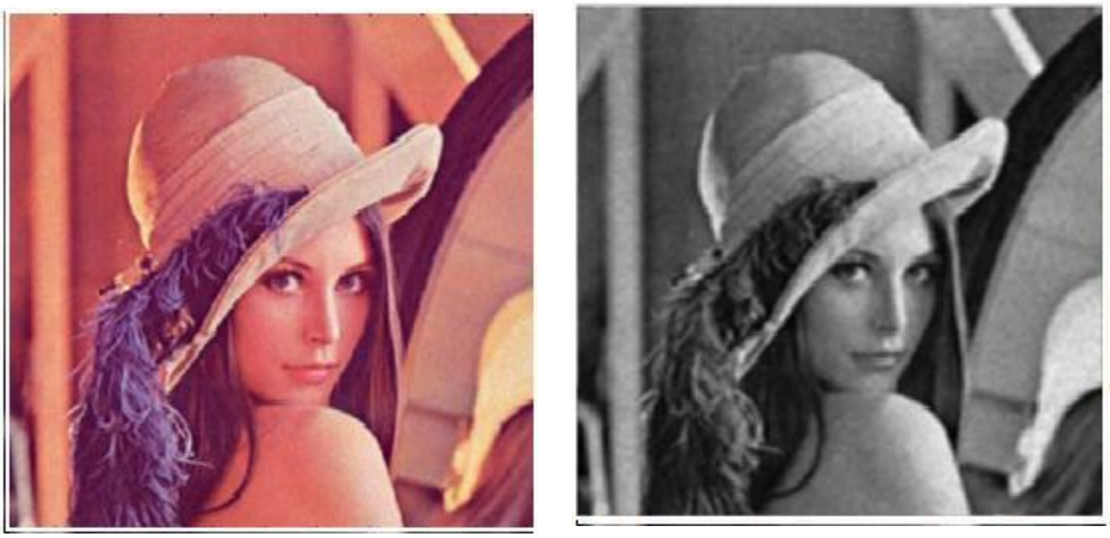


Fig. 4.10 Recuperación con Filtro Disk (para ruido Poisson)

Fig. 4.11 Recuperación con Filtro Gaussiano (para ruido Poisson)

Fig. 4.12 Recuperación con Filtro Motion (para ruido Poisson)

Fig. 4.13 Recuperación con Filtro Unsharp (para ruido Poisson)

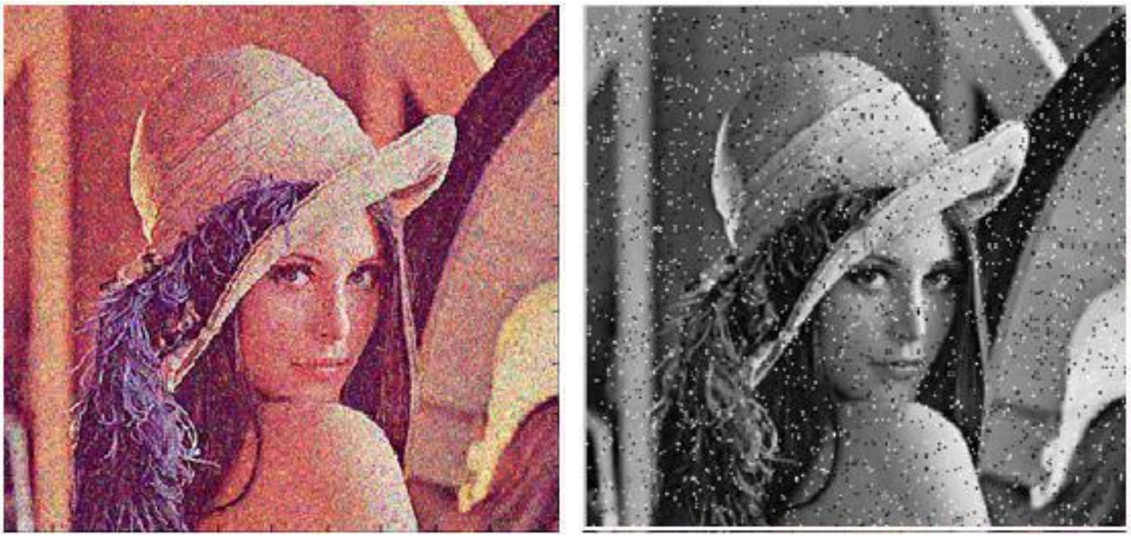


Fig. 4.14 Ruido Salt & Pepper aplicado a las dos Imágenes

Fig. 4.15 Recuperación con Filtro Average (para ruido Salt & Pepper)

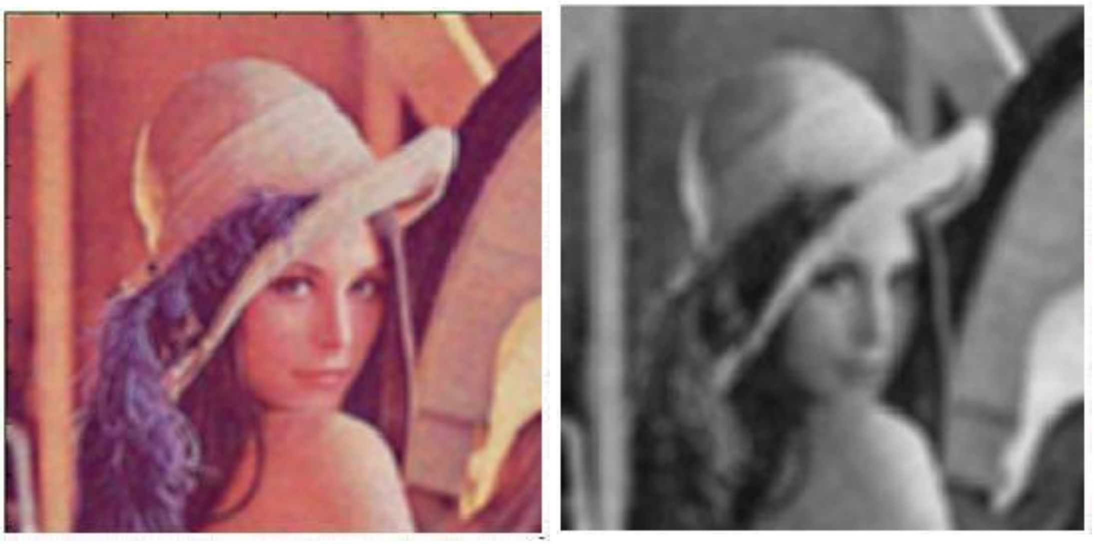


Fig. 4.16 Recuperación con Filtro Disk (para ruido Salt & Pepper)

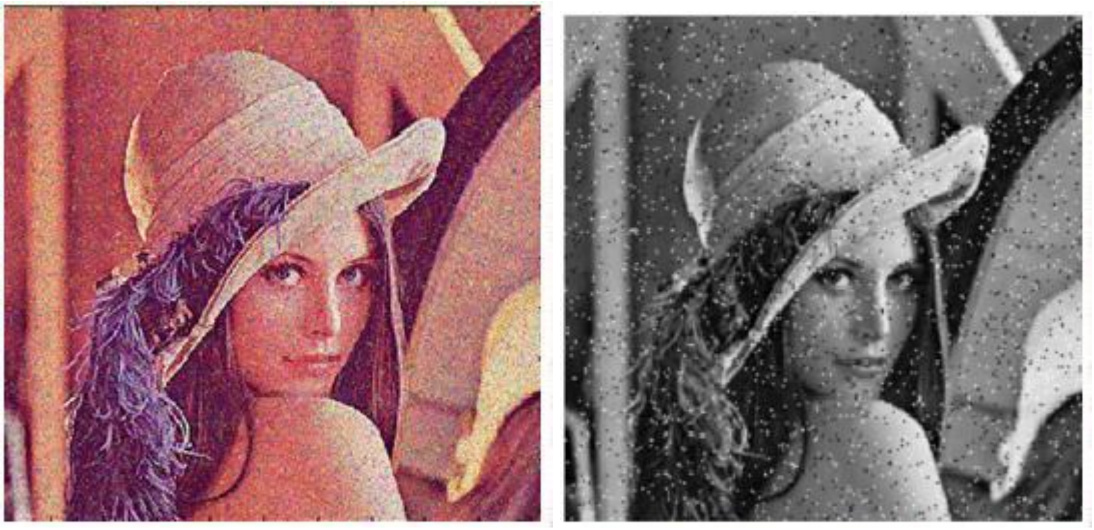


Fig. 4.17 Recuperación con Filtro Gaussiano (para ruido Salt & Pepper)

Fig. 4.18 Recuperación con Filtro Motion (para ruido Salt & Pepper)

Fig. 4.19 Recuperación con Filtro Unsharp (para ruido Salt & Pepper)

Fig. 4.20 Ruido Speckle aplicado a las dos Imágenes

Fig. 4.21 Recuperación con Filtro Average (para ruido Speckle)

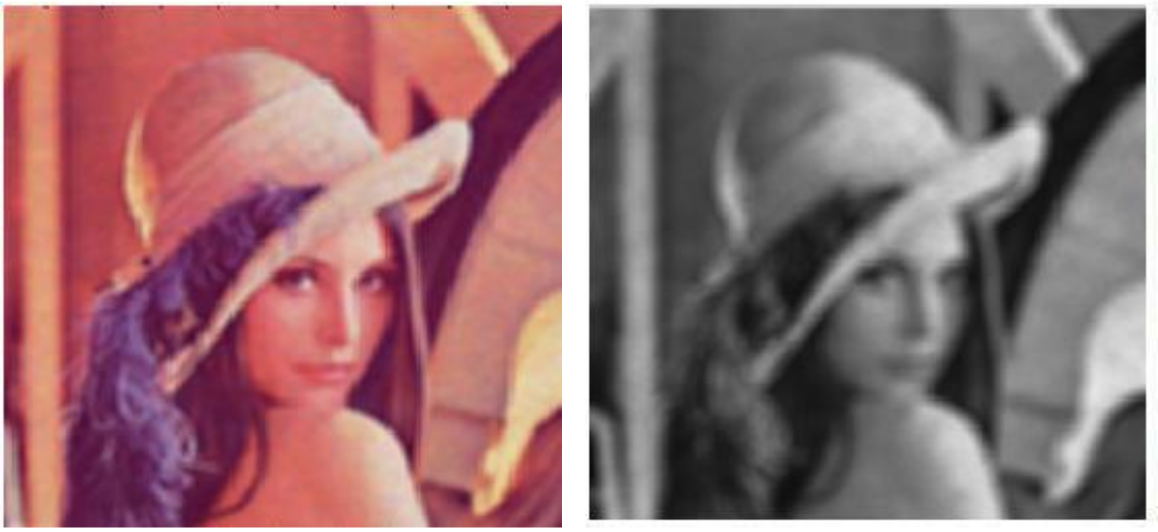


Fig. 4.22 Recuperación con Filtro Disk (para ruido Speckle)

Fig. 4.23 Recuperación con Filtro Gaussiano (para ruido Speckle)

Fig. 4.24 Recuperación con Filtro Motion (para ruido Speckle)

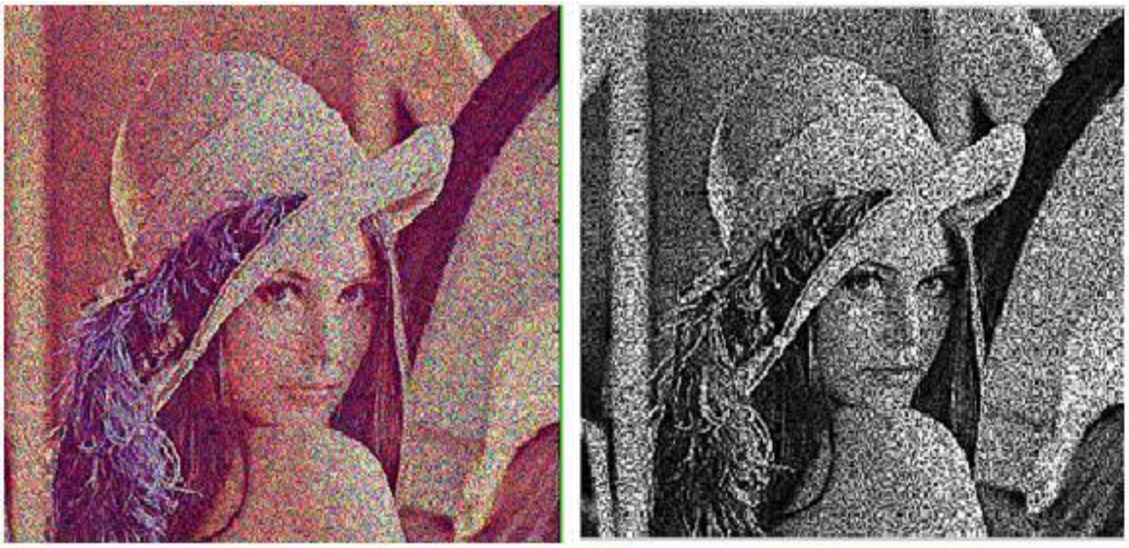


Fig. 4.25 Recuperación con Filtro Unsharp (para ruido Speckle)

Tabla II. Resultado visual de 50

		FILTROS DE RESTAURACION				
	Ruido Aplicado	Average	Disk	Gaussiano	Motion	Unsharp
baboon_b/n	Gaussiano	40	39	30	30	4
baboon		40	39	25	26	3
baboon_b/n	Poisson	36	30	44	20	16
baboon		36	28	42	15	12
baboon_b/n		30	25	40	20	18
						13
						15
						15
						5
						5
						16
						12
						16
						13
						10
						10
						2
						2
						16
						12
						18
						13
						18
						10

		FILTROS DE RESTAURACION				
	Ruido Aplicado	Average	Disk	Gaussiano	Motion	Unsharp
fruits_bn	Gaussiano	38	38	30	28	2
fruits		40	36	25	26	5
fruits_bn	Poisson	36	30	46	20	16
fruits		36	28	44	15	12
fruits_bn	Sal y Pimienta	45	24	40	20	18
fruits		40	25	35	14	13
fruits_bn	Speckle	43	30	40	22	12
fruits		41	28	36	15	10
peppers_bn	Gaussiano	40	32	30	24	2
peppers		40	32	25	24	5
peppers_bn	Poisson	36	30	44	20	16
peppers		36	28	44	15	12
peppers_bn	Sal y Pimienta	30	25	40	20	18
peppers		28	24	35	14	13
peppers_bn	Speckle	40	30	40	22	10
peppers		40	30	36	15	10
Lena_bn	Gaussiano	42	39	30	24	6
Lena		42	39	25	24	2
Lena_bn	Poisson	36	30	46	20	16
Lena		36	28	44	15	12
Lena_bn	Sal y Pimienta	30	22	40	20	18
Lena		28	22	35	14	13
Lena_bn	Speckle	42	35	40	22	10
Lena		40	34	36	15	10

personas encuestadas

CONCLUSIONES

- Al aumentar los parámetros de los ruidos aplicados, como la media y la varianza en el caso del ruido Gaussiano y el ruido Speckle, y la densidad en el ruido Salt & Pepper, se observa que las imágenes se degradan mas, haciéndose más difícil su recuperación.

- Según los resultados analíticos, para el ruido Gaussiano con una media de 0 y una varianza de 0.01, el filtro que mejor recupera las imágenes en blanco y negro es el filtro Average mientras que el filtro Disk recupera mejor las imágenes a color.

- Para el ruido tipo Poisson, se observó que el filtro Average recuperó de una mejor manera las imágenes, mientras que el filtro Unsharp es el filtro que menos recupera, esto se observó tanto visual como analíticamente al mirar que su MSE es el mayor de todos.

- Aplicando el ruido Salt & Pepper con una densidad de 0.05, se recuperó mejor con el filtro Gaussiano mientras que el filtro Unsharp nuevamente fue el filtro que menos recuperó ambas imágenes (color y blanco/negro para todas las imágenes).

- Por último, aplicando el ruido Speckle, con una media de 0 y una varianza de 0.04, el filtro que recuperó de una manera más eficiente las imágenes en blanco y negro fue el filtro Average, mientras que las imágenes a color fueron recuperadas de una mejor manera con el filtro Disk.

- Todas estas variaciones nos indica que un filtro es más eficiente que otro dependiendo de las características propias de las imágenes que usemos para realizar el análisis.

- Según los resultados de las encuestas realizadas a 50 personas se puede concluir que los resultados obtenidos distan de dos conceptos: el analítico y el visual que no necesariamente deben ser iguales ya que la percepción humana es muy diferente a un proceso experimental.

- Para los resultados de la encuesta, el 83% de las personas consultadas concluyó que para el ruido Gaussiano el mejor filtro que recupera la imagen es el tipo Average. Mientras para el ruido Poisson más del 90% concluyeron que los filtros Average y Gaussiano recuperan mejor la imagen.

- Resultados semejantes se obtuvieron para los ruidos Salt & Pepper y Speckle. Se puede concluir que el filtro Average y el Gaussiano son más eficientes al momento de recuperar imágenes en un entorno general.

RECOMENDACIONES

- El programa puede abrir cualquier tipo de imagen y tamaño sin importar su ubicación en la computadora en la que estemos ejecutando el programa.

- Este programa fue creado por motivos didácticos, para mostrar como varia la forma en que el ruido afecta a las imágenes, por tal motivo se recomienda que cuando se ejecute el programa por primera vez, se realice un análisis teórico de cada ruido aplicado en el programa, para tener un mejor entendimiento del tipo de degradación que realiza cada ruido.

- Se recomienda utilizar imágenes lo mas nítidas posibles, debido a que con esto se logra una mayor apreciación de la degradación aplicada a dicha imagen, y su posterior recuperación con los filtros.

- Se recomienda también realizar una investigación sobre cada uno de los filtros aplicados en el programa para su claro entendimiento y posterior uso.

REFERENCIAS

[1] “Digital Image Processing Using Matlab” - Gonzalez Woods & Eddins, Third Edition, 2003

[2] MATLAB, “Image Processing Toolbox User’s Guide”, Mathworks Inc., 2008.

[3] VASEGHI Sabed V. “Advanced Digital Signal Processing and Noise Reduction”, John Wiley & Sons Ltda., Second Edition, 2000.

[4] http://en.wikipedia.org/wiki/Gaussian_filter

[5] http://homepages.inf.ed.ac.uk/rbf/HIPR2/unsharp.htm

[6] Wikipedia, www.wikipedia.com

[7] http://en.wikipedia.org/wiki/Poisson_distribution

[8] http://www.isprs.org/commission1/ankara06/makaleler/Rajabi_Specle_Noise.pdf

[9] http://fi.uaemex.mx/h.a.montes/files/Giovanni%20Vega%20Velasquez%20-%20Reporte%20TI.pdf (ruido Salt & Pepper)

[10] http://www.um.es/geograf/sigmur/teledet/tema06.pdf (filtro Gaussiano)

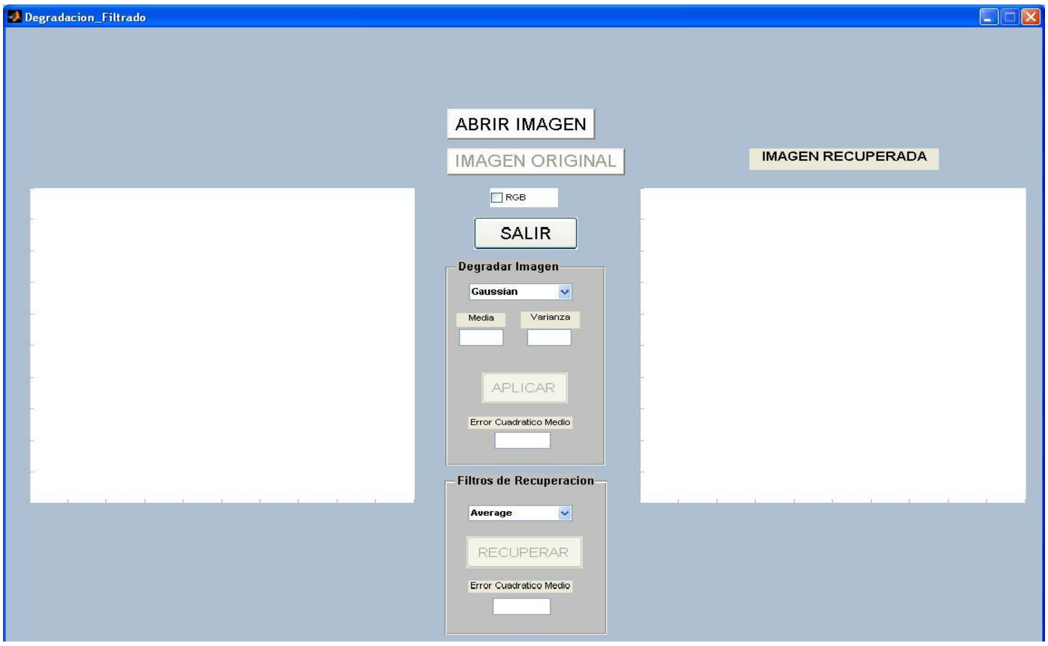
[11] Lena, baboon, airplane, dark_stream, fruits, peppers: Facilitada por Ing. Patricia Chávez.

ANEXO A

MANUAL DEL USUARIO

Este programa se basa en la aplicación de ruido a cualquier imagen en particular, y luego se la puede recuperar aplicándole 5 tipos distintos de filtro.

El programa le muestra al usuario cada etapa de la imagen procesada, esto es, se presenta primero la imagen original, luego de aplicarle cualquier ruido, se muestra la imagen degradada, al final, después de escoger un filtro, se muestra la imagen recuperada. En cada caso, se muestra el MSE para cada etapa del programa.

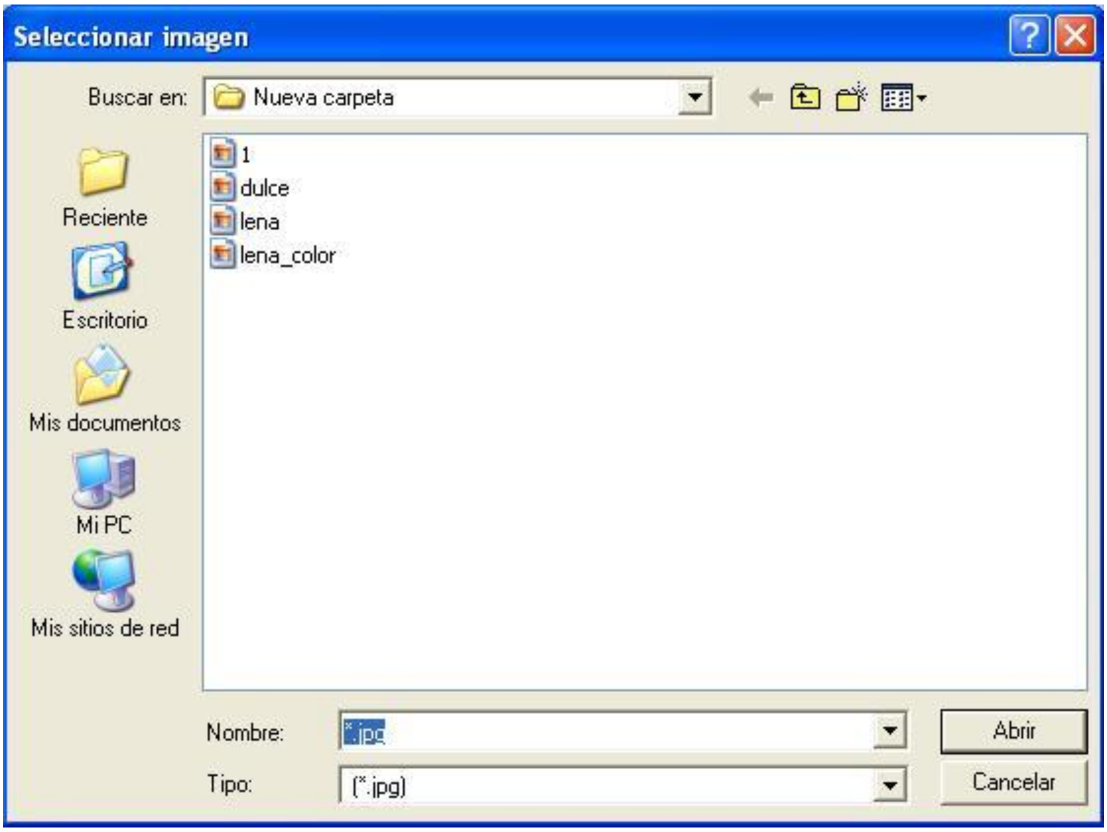


Degradar Imagen: Gaussian

Filtros de Recuperacion: Average

Fig. A.1 Menú principal del programa

Lo primero que debe realizar el usuario es abrir una imagen haciendo un clip sobre el botón “ABRIR IMAGEN”, y se abrirá el siguiente cuadro de dialogo:



Nombre: *.jpg

Tipo: (*.jpg)

Fig. A.2 Cuadro de dialogo para abrir la imagen

Luego de escoger la imagen, se procederá a escoger si es la imagen es a color o blanco y negro, después de esto, la imagen es presentada en pantalla.

Luego, el usuario procederá a escoger el tipo de ruido que desea aplicar a la imagen:

Fig. A.3 Escoger tipo de ruido

Luego de escoger el tipo de ruido, se pulsa sobre el botón "APLICAR", en ese instante se mostrará el Error Cuadrático Medio y se mostrará la imagen deteriorada.

Salt & Pepper

Densidad 0.05

0

Error Cuadratico Medio 6

Fig. A.4 Ruido aplicado con el MSE mostrado en pantalla

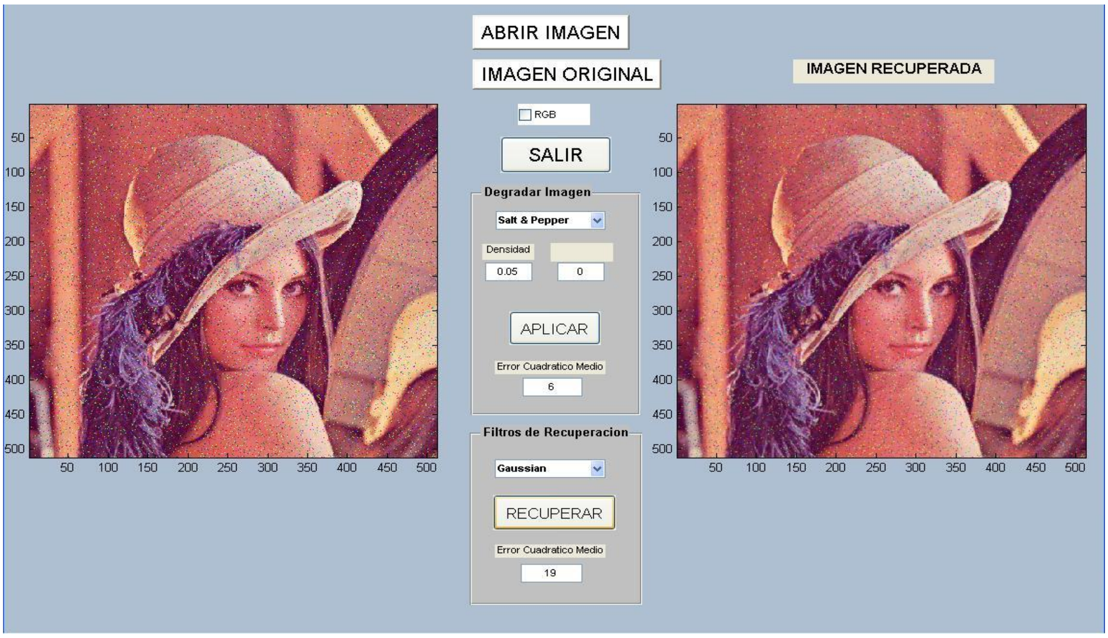
Después de distorsionar la imagen, el usuario escogerá el filtro con el que desea recuperar la imagen

Fig. A.5 Menú para escoger el tipo de filtro de recuperación

Al presionar el botón “APLICAR”, se mostrará el MSE y la imagen recuperada.

Si desea comparar la imagen original con la recuperada, debe presionar el botón “IMAGEN ORIGINAL” y así podrá comparar ambas imágenes.

Al final, el usuario escoge si desea salir del programa o continuar cargando otra imagen. Esto se realiza mediante el botón “SALIR”.



☐ RGB

Degradar Imagen

Salt & Pepper

Densidad 0.05 0

Error Cuadratico Medio 6

Filtros de Recuperacion

Gaussian

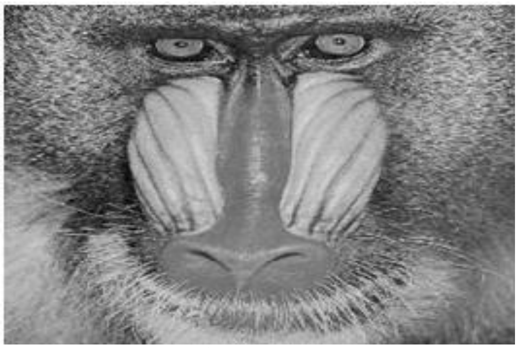
Error Cuadratico Medio 19

Fig. A.6 Aplicación del filtro escogido para recuperar la imagen

ANEXO B

IMÁGENES UTILIZADAS

A continuación, se presentan las imágenes a color y en blanco y negro que se utilizaron para las encuestas que se realizaron a 50 personas. Son seis en total y fueron proporcionadas por la Ing. Patricia Chávez:

 Imagen Original “baboon” a color	 Imagen Original “baboon” a b/n
 Imagen Original “airplane” a color	 Imagen Original “airplane” a b/n

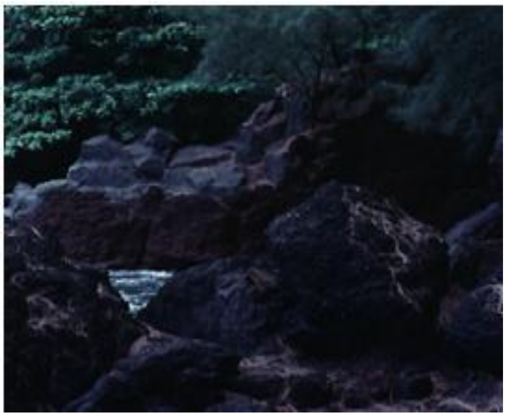
 Imagen Original “dark_stream” a color	 Imagen Original “dark_stream” a b/n
 Imagen Original “fruits” a color	 Imagen Original “fruits” a b/n
 Imagen Original “peppers” a color	 Imagen Original “peppers” a b/n

 Imagen Original "Lena" a color	 Imagen Original "Lena" a b/n