

UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA
FACULTAD DE INGENIERÍA, ARQUITECTURA Y DISEÑO
CAMPUS ENSENADA



Procesamiento digital de imágenes empleando filtros espaciales.

Tesis que para obtener el grado de
INGENIERO EN COMPUTACIÓN

Presenta:
REYNA CASTILLO MONROY

Ensenada, Baja California, México. Septiembre de 2012.

TESIS DEFENDIDA POR
REYNA CASTILLO MONROY
Y APROBADA POR EL SIGUIENTE COMITÉ



M.C. Juan Miguel Hernández Ontiveros
Codirector de Tesis



M.C. Everardo Inzunza González
Codirector de Tesis



M.C. Raúl Casillas Figueroa
Miembro del Comité

Septiembre de 2012

RESUMEN de la Tesis que presenta **Reyna Castillo Monroy**, para la obtención del grado de **Ingeniero en COMPUTACIÓN**. Ensenada, Baja California, México. Septiembre de 2012.

PROCESAMIENTO DIGITAL DE IMÁGENES EMPLEANDO FILTROS ESPACIALES.

Resumen aprobado por:



M.C. Juan Miguel Hernández Ontiveros

Codirector de Tesis



M.C. Everardo Inzunza González

Codirector de Tesis

En este trabajo de tesis, se presenta un estudio, diseño e implementación de filtros espaciales para el procesamiento de imágenes ruidosas. Para el desarrollo del software se empleó el MATLAB con el apoyo de la herramienta Guide, dando como resultado una interfaz gráfica la cual es amigable y fácil de utilizar por los usuarios, por lo cual se considera de tipo didáctica. Este trabajo toma en cuenta el uso de filtros con cajas adaptables y no-adaptables de tamaños de 3×3 , 5×5 y 7×7 . Los filtros que se implementaron con caja no-adaptable son: Pasabajas (filtro promedio), pasa altas, raíz cuadrada, cuadrático y sigma global. Con respecto a los filtros de caja adaptable, se implementaron los siguientes: Bit-Error y Sigma de caja local. Se hace un estudio experimental para saber cuáles filtros son más adecuados de utilizarse dependiendo del tipo de ruido. Para validar estos experimentos se utilizaron distintas imágenes y distintos tipos de ruido, tales como el ruido Gaussiano, Poisson y tipo Sal y Pimienta. Con la aplicación de los filtros desarrollados se lograron reducir considerablemente los niveles de ruido.

Palabras clave: Procesamiento digital de imágenes, Filtros espaciales, filtros digitales, algoritmos, software.

ABSTRACT of the Thesis presented by **Reyna Castillo Monroy**, to obtain the bachelor's degree in Computer Engineering. Ensenada, Baja California, Mexico. September 2012.

DIGITAL IMAGE PROCESSING USING SPATIAL FILTERS.

In this thesis work a study, a design, and an implementation of spatial filters for processing noisy images are presented. To develop the software, MATLAB was used supported by the Guide Tool, resulting in a graphic interface which is easy to use, therefore it is regarded as educative. This work takes into consideration the use of filters with adaptive and non-adaptive boxes and their sizes are 3x3, 5x5, and 7x7. The filters with a non-adaptive box that were implemented are: Low-pass filter (average), high-pass filter, square root, quadratic, and global sigma. For filters with an adaptive box, the following were implemented: Bit-Error and Sigma of a local box. An experimental study is made to know what filters are more suitable to use depending on the sort of noise. To validate these experiments different images and different types of noise were used, such as the Gaussian, Poisson, and Salt-and-Pepper noises. By applying the developed filters, a considerable reduction of the noise levels was achieved.

Keywords: Digital image processing, spatial filters, digital filters, algorithms, software.

DEDICATORIA

A Dios. Porque tú formaste mis entrañas, tú me hiciste en el vientre de mi madre, te alabaré porque formidables y maravillosas son tus obras y mi alma lo sabe muy bien.

A mi padre. Por ser el mejor ejemplo de vida que podré tener, por tus enseñanzas repetitivas que son las más sencillas de recordar. Gracias por enseñarme a valerme por mí misma, y el amor que debo poner en todo lo que haga. Por estar siempre para mí en cualquier momento y por darme más, mucho más de lo que merezco.

A mi madre. Tus consejos y motivaciones nunca tienen fin y siempre son bien atinados, compartir el tiempo contigo siempre es grato, divertido y edificante. Nunca tendré con que pagar el que hayas sacrificado tu vida profesional por mi bienestar. Gracias por tu ejemplo de trabajo incansable e incondicional.

A mi hermana. Gracias por ser mi mejor amiga, nuestras peleas y locuras hacen que cada día valga la pena, gracias por ser la parte fuerte de este dueto, y cantar conmigo la misma melodía desde que naciste.

AGRADECIMIENTOS ESPECIALES

Al M.C. Juan Miguel Hernández Ontiveros. Por su asesoramiento científico y estímulo para seguir creciendo intelectualmente. Gracias por nunca estar demasiado ocupado para resolver dudas, además de siempre tener las mejores sugerencias y aportaciones. Gracias porque sabes ser un excelente maestro y aún mejor un amigo. Esta tesis pudo ser realizada gracias a ti.

Al M.C. Everardo Inzunza González. Mil gracias por su valiosa colaboración y observaciones críticas en el desarrollo del trabajo, por ocupar horas extras para invertir en este proyecto de manera precisa y siempre con una buena disposición.

Al M.C. Raúl Casillas Figueroa. Por sus comentarios y colaboración desinteresada que fue muy útil para la culminación de esta tesis.

A la Universidad Autónoma de Baja California por el apoyo brindado.

CONTENIDO

CONTENIDO	I
ÍNDICE DE FIGURAS	III
ÍNDICE DE TABLAS.....	V
I. INTRODUCCIÓN	1
I.1. Antecedentes	1
I.1.1. Filtrado en el dominio de la frecuencia	3
I.1.2. Filtrado en el dominio del espacio.....	6
I.2. Planteamiento del problema.....	10
I.3. Objetivos.....	11
I.3.1. Objetivos generales.....	11
I.3.2. Objetivos específicos	11
I.4. Infraestructura.....	11
I.5. Estado del arte.....	12
II. TIPOS DE FILTROS	18
II.1. Introducción.....	18
II.2. Filtros digitales y analógicos	19
II.3. Aplicaciones de los filtros.....	22
II.4. Ventajas e inconvenientes de los filtros digitales	25
II.5. Filtros recursivos y no recursivos	29
II.6. Filtros digitales FIR	31
II.7. Filtros digitales IIR	34
II.8. Filtros espaciales.....	36
II.8.1. Filtros Bit Error de Caja Adaptable	38
II.8.2. Filtro Sigma de Caja Adaptable.....	41
II.9. Filtros Adaptativos.....	43
II.9.1. Ejemplos de sistemas adaptativos.....	44

II.9.2.	Características fundamentales de los sistemas adaptativos	45
II.10.	Transformada rápida de Fourier FFT.....	46
III.	CONCEPTOS BÁSICOS DE LAS IMÁGENES	49
III.1.	Introducción.....	49
III.2.	Conceptos básicos de las imágenes	50
III.2.1.	Lectura y escritura de imágenes a través de archivo	53
III.2.2.	Acceso a píxel y planos en las imágenes	56
III.2.3.	Sub-muestreo de imágenes	59
III.2.4.	Tipo de dato de los elementos de una imagen	60
IV.	DESARROLLO DEL PROYECTO	61
IV.1.	Desarrollo.....	61
IV.2.	Filtro Pasa-Bajas (Promedio).....	66
IV.3.	Filtro Sigma	68
IV.4.	Filtro Bit-Error.....	69
IV.5.	Filtro Sigma de Caja Adaptable.....	70
V.	RESULTADOS Y CONCLUSIONES.....	72
V.1.	Resultados.....	72
V.1.1.	Aplicación de los Filtros a Imágenes Con Ruido Tipo Sal y Pimienta.....	72
V.1.2.	Aplicación de filtros a imágenes con ruido tipo Gaussiano.....	82
V.1.3.	Aplicación de filtros a imágenes con ruido tipo Poisson.....	86
V.2.	Conclusiones.....	92
V.3.	Trabajo a futuro	93
REFERENCIAS	94	
REFERENCIAS (CONTINUACIÓN).....	95	

ÍNDICE DE FIGURAS

Figura 1. Etapas del procesamiento de imágenes en el dominio de la frecuencia.	4
Figura 2. Filtros en el dominio de la frecuencia y espacio.	4
Figura 3. Matriz de coeficientes en el uso del Kernel.	7
Figura 4. Ejemplo de imagen original sin ruido.	13
Figura 5. Ejemplo de imagen con ruido tipo “sal y pimienta”.	13
Figura 6. Ejemplo de imagen con ruido Gaussiano (aditivo).	14
Figura 7. Ejemplo de imagen con ruido multiplicativo (valor de pixel multiplicado por un ruido blanco).	14
Figura 8. Imagen obtenida con la sonda Vikingo en una región de Marte llamada Tihonia. A) Imagen cruda (original), B) Resultado de la aplicación de 3 filtros de ruido con caja adaptable.	16
Figura 9. Filtrado de una señal.....	19
Figura 10. Esquema básico del proceso de filtrado de una señal.	21
Figura 11. Filtrado de una señal analógica mediante un filtro digital.	28
Figura 12. Estructura de un filtro FIR.....	32
Figura 13. Estructura de un filtro IIR.	35
Figura 14. Representación de una imagen a escala de grises en MATLAB.....	51
Figura 15. Representación de una imagen a color en RGB en MATLAB.	52
Figura 16. Imagen mostrada al utilizar la función <i>imshow</i> en MATLAB.	55
Figura 17. Obtención del valor de un píxel de imagen1.	56
Figura 18. Planos de la imagen a) rojo, b) verde y c) azul.	58
Figura 19. Ejemplo de sub-muestreo.	60
Figura 20. Interfaz gráfica principal del software.....	62
Figura 21. Interfaz para abrir archivo de imágenes BMP ó JPG.	63
Figura 22. Ventana auxiliar para abrir archivo de imágenes BMP ó JPG.	64
Figura 23. Interfaz gráfica para seleccionar el tipo de filtro.....	65
Figura 24. Interfaz gráfica para seleccionar filtros No-adaptables.	65

Figura 25. Barra de progreso indicando que se está realizando la tarea de procesamiento o filtrado de la imagen.	67
Figura 26. Interfaz gráfica del filtro tipo Sigma con umbral global.	68
Figura 27. Interfaz gráfica para seleccionar filtros adaptables.	69
Figura 28. Interfaz gráfica del filtro adaptable tipo Bit-Error.	70
Figura 29. Interfaz gráfica del filtro adaptable tipo Sigma.	71
Figura 30. Imagen de monedas con ruido tipo sal y pimienta.	73
Figura 31. Efecto de aplicar el filtro promedio 3x3 a la imagen de monedas	74
Figura 32. Efecto de aplicar el filtro pasa-altas 3x3.	75
Figura 33. Efecto de aplicar el filtro cuadrático 3x3.	76
Figura 34. Efecto de aplicar el filtro sigma global 3x3.	77
Figura 35. Efecto de aplicar el filtro adaptable Bit-Error 3x3.	78
Figura 36. Efecto de aplicar el filtro adaptable Bit-Error 5x5.	79
Figura 37. Efecto de aplicar el filtro adaptable Bit-Error 7x7.	80
Figura 38. Efecto de aplicar el filtro sigma local 3x3.	81
Figura 39. Imagen de una casa con ruido tipo Gaussiano.	82
Figura 40. Resultado de aplicar el filtro tipo promedio con caja de 5x5.	83
Figura 41. Resultado de aplicar el filtro tipo Bit-Error con caja de 5x5.	84
Figura 42. Resultado de aplicar el filtro tipo sigma de caja adaptable 5x5.	85
Figura 43. Imagen de Lena con ruido tipo Poisson.	86
Figura 44. Resultado de la aplicación del filtro promedio con caja de 3x3 a la imagen de Lena con ruido de Poisson.	87
Figura 45. Resultado de aplicar el filtro tipo Bit-Error con caja 3x3 a la imagen de Lena con ruido Poisson.	88
Figura 46. Resultado de aplicar el filtro tipo sigma con caja adaptable 3x3 a la imagen de Lena con ruido Poisson.	89
Figura 47. Resultado de aplicar el filtro tipo sigma con caja adaptable 5x5 a la imagen de Lena con ruido Poisson.	90
Figura 48. Resultado de aplicar el filtro tipo sigma con caja adaptable 7x7 a la imagen de Lena con ruido Poisson.	91

ÍNDICE DE TABLAS

Tabla I. Máscara de tamaño de 3x3.....	37
Tabla II. Formatos y extensiones soportadas por MATLAB.....	53

I. INTRODUCCIÓN

I.1. Antecedentes

El procesamiento digital de imágenes es el conjunto de técnicas que se aplican a las imágenes digitales con el objetivo de mejorar la calidad o facilitar la búsqueda de información [Proakis y Manolakis, 1999].

El proceso de filtrado es el conjunto de técnicas englobadas dentro del pre-procesamiento de imágenes cuyo objetivo fundamental es obtener, a partir de una imagen origen, otra final cuyo resultado sea más adecuado para una aplicación específica mejorando ciertas características de la misma que posibilite efectuar operaciones del procesado sobre ella.

Los principales objetivos que se persiguen con la aplicación de filtros son:

- *Suavizar la imagen.* Reducir la cantidad de variaciones de intensidad entre pixeles vecinos.
- *Eliminar ruido.* Eliminar aquellos pixeles cuyo nivel de intensidad es muy diferente al de sus vecinos y cuyo origen puede estar tanto en el proceso de adquisición de la imagen como en el de transmisión.
- *Realzar bordes.* Destacar los bordes que se localizan en una imagen.
- *Detectar bordes.* Detectar los pixeles donde se produce un cambio brusco en la función intensidad.

Por tanto, se consideran los filtros como operaciones que se aplican a los pixeles de una imagen digital para optimizarla, enfatizar cierta información o conseguir un efecto especial en ella. El proceso de filtrado puede llevarse a cabo sobre los dominios de frecuencia y/o espacio.

I.1.1. Filtrado en el dominio de la frecuencia

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

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

El Teorema de la Convolución (frecuencia) se realiza mediante:

$$G(u, v) = F(u, v) * H(u, v) \quad (1)$$

Donde:

$F(u, v)$: transformada de Fourier de la imagen original

$H(u, v)$: Filtro atenuador de frecuencias

Como la multiplicación en el espacio de Fourier es idéntica a la convolución en el dominio espacial, todos los filtros podrían en teoría, ser implementados como un filtro espacial, las etapas del procesamiento de imágenes en el dominio de la frecuencia se ven en la figura 1.

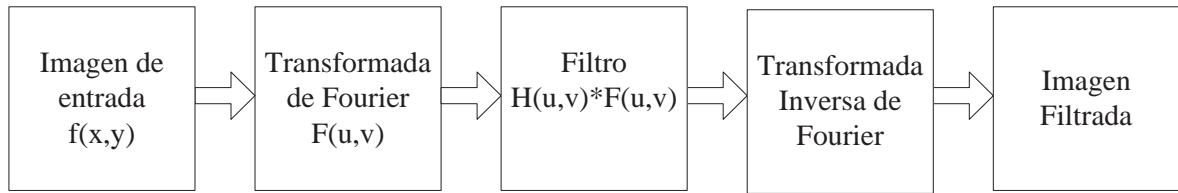


Figura 1. Etapas del procesamiento de imágenes en el dominio de la frecuencia.

Tipos de filtros para el dominio de la frecuencia.

Existen básicamente tres tipos distintos de filtros que pueden aplicarse los cuales se ven en la figura 2:

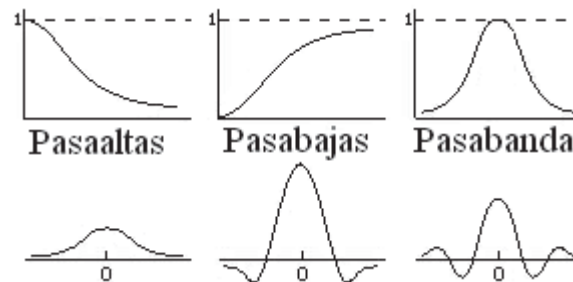


Figura 2. Filtros en el dominio de la frecuencia y espacio.

- *Filtro pasabajas.* Atenúa las frecuencias altas y mantiene sin variaciones las bajas. El resultado en el dominio espacial es equivalente al de un filtro suavizado, donde las altas frecuencias que son filtradas se corresponden con los cambios fuertes de intensidad. Consigue reducir el ruido suavizando las transiciones existentes.

- *Filtro pasaaltas.* Atenúa las frecuencias bajas manteniendo invariables las frecuencias altas. Puesto que las altas frecuencias corresponden en las imágenes a cambios bruscos de densidad, este tipo de filtros es usado porque entre otras ventajas, ofrece mejoras en la detección de bordes en el dominio espacial, ya que estos contienen gran cantidad de dichas frecuencias. Refuerza los contrastes que se encuentran en la imagen.
- *Filtro pasabanda.* Atenúa frecuencias muy altas o muy bajas manteniendo una banda de rango medio.

Ventajas:

- Método simple y sencillo de implementar.
- Fácil asociación del concepto de frecuencia con ciertas características de la imagen. Cambios de tonalidad suaves implican frecuencias bajas y cambios bruscos frecuencias altas.
- Proporciona flexibilidad en el diseño de soluciones de filtrado.
- Rapidez en el filtrado al utilizar el Teorema de la Convolución.

Desventajas:

- Se necesitan conocimientos en varios campos para desarrollar una aplicación para el procesamiento de imágenes
- El ruido no puede ser eliminado completamente.

I.1.2. Filtrado en el dominio del espacio

Las operaciones de filtrado se llevan a cabo directamente sobre los píxeles de la imagen. En este proceso se relaciona, para todos y cada uno de los puntos de la imagen, un conjunto de píxeles próximos al píxel objeto con la finalidad de obtener una información útil, dependiendo del tipo de filtro aplicado, que permita actuar sobre el píxel concreto en que se está llevando a cabo el proceso de filtrado para de este modo obtener mejoras sobre la imagen y/o datos que podrían ser utilizados en futuras acciones o procesos de trabajo sobre ella.

Tipos de filtros para el dominio del espacio.

- Filtros lineales (filtros basados en Kernels o máscaras de convolución).
- Filtros no lineales.

El concepto de Kernel se entiende como una matriz de coeficientes donde el entorno del punto (x,y) que se considera en la imagen para obtener $g(x,y)$ está determinado por el tamaño y forma del Kernel seleccionado. Aunque la forma y tamaño de esta matriz es variable y queda a elección de cada usuario, es común el uso de Kernels cuadrados $n \times n$. Dependiendo de la implementación, en los límites de la imagen se aplica un tratamiento especial (se asume un marco exterior de ceros o se repiten los valores del borde) o no se aplica ninguno. Es por ello, que el tipo de filtrado queda establecido por el contenido de dicho Kernel utilizado. Lo anterior se puede ver en la figura 3 [Lee, 1983].

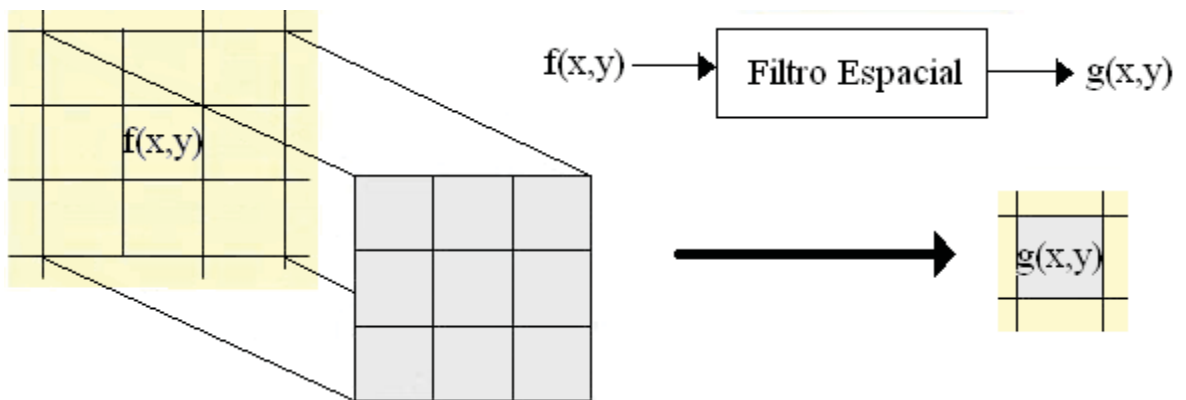


Figura 3. Matriz de coeficientes en el uso del Kernel.

Para realizar un filtrado en el dominio del espacio se realiza una convolución (barrido) del Kernel sobre la imagen. Para ello se sigue el Teorema de Convolución en el espacio:

$$g(x, y) = h(x, y) * f(x, y) \quad (2)$$

1. Cada pixel de la nueva imagen se obtiene mediante el sumatorio de la multiplicación del Kernel por los pixeles contiguos:

$$g(x, y) = \sum \sum f(i, j) w(i, j) \quad (3)$$

2. Generalmente se divide sobre cierto valor constante para normalizar que suele obtenerse de la suma de los valores del Kernel empleado.

Tipos.

- *Filtro pasabajas (suavizamiento)*. Utilizados para eliminar ruido o detalles pequeños de poco interés puesto que sólo afecta a zonas con muchos cambios. La frecuencia de corte se determina por el tamaño del Kernel y sus coeficientes. Se emplean diversos Kernels:
 - ✓ *Promedio*: promedio de pixeles vecinos (Kernel de unos).
 - ✓ *Pasabajas en frecuencia*.
 - ✓ *Media*: reemplaza cada pixel por el valor medio de sus contiguos.
 - ✓ *Mediana*: sustituye por el valor de la mediana de los pixeles vecinos (normalmente se comporta mejor que el de promedio).

- ✓ *Gaussiano*: aproximación a la distribución gassiana.
- *Filtro pasaaltas (atenuamiento)*. Intensifica los detalles, bordes y cambios de alta frecuencia, mientras que atenúa las zonas de tonalidad uniforme. Esto permite una mejor identificación posterior de los objetos que se encuentren en la imagen, puesto que el brillo se hace mayor en las zonas con frecuencias más altas, al mismo tiempo que se oscurecen las zonas de frecuencias bajas. Es común la aparición de ruido tras el proceso.
- *Realce de bordes por desplazamiento y diferencia*. Sustrae de la imagen original una copia desplazada de la misma, Así, es posible localizar y hacer resaltar los bordes existentes y que se quieran obtener según el modelo de Kernel aplicado:
 - ✓ Horizontal.
 - ✓ Vertical.
 - ✓ Diagonal.
- *Realce de bordes mediante Laplace*. Este tipo de filtros realza los bordes en todas direcciones. Este filtro trabaja con la segunda derivada, que permite obtener unos mejores resultados a pesar del aumento del ruido que se produce en la imagen.
- *Resalte de bordes con gradiente direccional*. Empleado para destacar y resaltar con mayor precisión los bordes que se localizan en una dirección

determinada, Trabaja con los cambios de intensidad existentes entre pixeles contiguos.

- *Detección de bordes y filtros de contorno (Prewitt y Sobel)*. Se centra en las diferencias de intensidad que se dan pixel a pixel. Son utilizados para obtener los contornos de objetos y de este modo clasificar las formas existentes dentro de una imagen. Este tipo de filtros requieren de un menor costo computacional.

I.2. Planteamiento del problema

El presente trabajo de tesis de licenciatura, parte de la necesidad de reducir el nivel de ruido en imágenes digitales, por lo general este ruido puede ser debido a la naturaleza del sensor o a partículas que se encuentran en el medio ambiente al momento de ser capturadas. Otro ejemplo de imágenes ruidosas son las que se capturan en el espacio sideral mediante el uso de satélites, tal como las imágenes de radar. Debido a la necesidad de investigadores del área de astronomía y meteorología de analizar imágenes ruidosas y realizar trabajos de investigación, surge la idea de desarrollar una herramienta sistematizada para reducir el ruido en este tipo de imágenes, de tal forma que se pueda interpretar de mejor manera la información importante.

I.3. Objetivos

I.3.1. Objetivos generales

Desarrollar un software de procesamiento de imágenes ruidosas mediante el uso de filtros espaciales bajo la plataforma de MATLAB.

I.3.2. Objetivos específicos

- Diseñar e implementar un software en MATLAB que sea amigable y fácil de utilizar por los usuarios.
- Reducir el nivel de ruido en las imágenes.
- Utilizar filtros espaciales adaptables y no adaptables.
- Evaluar la calidad de los filtros espaciales

I.4. Infraestructura

Para el diseño y la simulación que será presentado en este trabajo de tesis, se utilizará la siguiente infraestructura:

⇒ Laboratorio de Ingeniería y Negocios en UABC San Quintín.

⇒ Laboratorio de ingeniería en UABC Ensenada.

⇒ Biblioteca central UABC Ensenada y San Quintín.

⇒ Computadora con sistema operativo OS X.

⇒ Programas de aplicación tales como:

- ❖ MATLAB 7.0®
- ❖ Microsoft Office para MAC 2007®.
- ❖ Acceso a Internet.

I.5. Estado del arte

[Eliason y McEwn, 1990] presentan el desarrollo de algoritmos para eliminar errores aleatorios de bits (valores de pixeles que no tienen relación con la escena de la imagen) y para suavizar datos ruidosos (pixeles relacionados con la escena de la imagen pero con componentes de ruido aditivo o multiplicativo). Para ambos procedimientos se utiliza la desviación estándar (σ) de los pixeles dentro de una caja local circundante a cada pixel, por lo que se conocen como filtros adaptables.

Un punto de dato (pixel) es considerado para ser bit erróneo si éste varía con respecto al promedio de la caja por más de 1.0 a 2.0 σ , y este es reemplazado por el promedio de la caja (sin considerar el pixel erróneo). Por otro lado una caja (Box) del filtro es una matriz de dimensión de $(2K+1)$ por $(2L+1)$, donde K y L son enteros mayores que cero. El capítulo II aborda más sobre este tema.

El algoritmo usado por [Eliason y McEwn, 1990] tiene ventajas sobre otros debido a que preservan la resolución completa y detalles en los datos, además de ser sencillo y rápido. El objeto de este algoritmo es conservar los detalles de la escena entera, mientras que elimina o reduce el ruido. Este algoritmo elimina dos tipos de ruido aleatorio:

1. Pixel-único. Errores aleatorios de bits o ruido tipo “sal y pimienta”, cuyo valor del pixel no tiene relación con la escena de la imagen. Las figuras 4 y 5 muestran un ejemplo de este tipo de ruido.

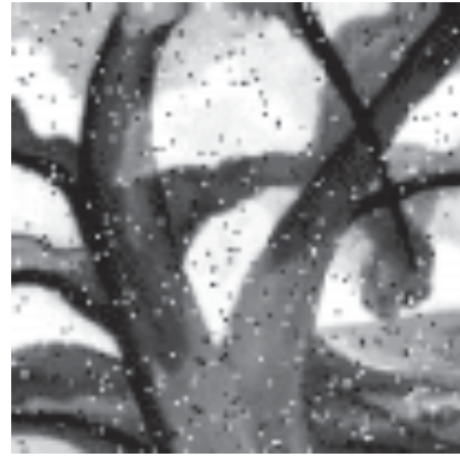


Figura 4. Ejemplo de imagen original sin ruido.

Figura 5. Ejemplo de imagen con ruido tipo “sal y pimienta”.

2. Datos ruidosos. Píxeles cuya intensidad está relacionada con la escena de la imagen pero tienen una componente de ruido aditivo o multiplicativo. Las figuras 6 y 7 muestran ejemplos de este tipo de ruido.

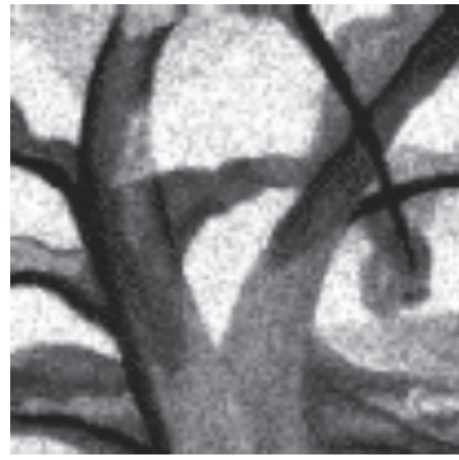


Figura 6. Ejemplo de imagen con ruido Gaussiano (aditivo).

Figura 7. Ejemplo de imagen con ruido multiplicativo (valor de pixel multiplicado por un ruido blanco).

En el 1er caso (pixel-único), el objetivo es eliminar los errores de bits reemplazando el valor del pixel por el valor promedio de los píxeles circundantes válidos (sin considerar el pixel o píxeles erróneos). En el segundo caso (datos ruidosos), el objetivo es suavizar los datos ruidosos. Este artículo se concentra en el 1er caso, pero también menciona una técnica para el 2do caso, la cual minimiza el borrono en los detalles de la escena.

La ocurrencia de errores de bits es muy frecuente cuando se adquieren imágenes con sondas espaciales, tal como Marinero y Vikingo, debido a la gran distancia que se tiene de telemetría. A mayor tasa de bits, mayor son los errores de bits. En aplicaciones donde se adquieren imágenes de muy alta resolución, es permitido aceptar una tasa alta de errores de bits (de hasta un 50%).

Cerca del 80% de las imágenes del Marinero 10 que tomaba imágenes del planeta Mercurio, fueron adquiridas a una tasa de bits de 117.6 Kbit/s. Más del 10% de los valores de estas imágenes contenía errores de bits.

En la sonda espacial Vikingo, primeramente las imágenes son grabadas a bordo (en cabina) en 7 pistas de cinta magnética, después son reproducidas y enviadas a la Tierra una pista a la vez. Las pérdidas de los datos de una pista durante la telemetría es aproximadamente en intervalos de 7 píxeles a través de las líneas de exploración de la imagen. Usualmente los datos perdidos son reemplazados por el promedio de los píxeles circundantes, pero si los píxeles circundantes contienen muchos bits erróneos, el promedio no será satisfactorio.

Las imágenes que obtiene la sonda Vikingo son mediante un Radar de Apertura Sintética. Algunas de las imágenes de la más alta resolución adquiridas por la

sonda espacial Vikingo sobre algunas regiones del planeta Marte tienen hasta un 50% de bits erróneos o datos perdidos, la imagen cruda tiene una apariencia inusual, la figura 8 muestra dicha apariencia. Con la técnica mostrada en este artículo se pueden producir imágenes que estén libres de manchas y sin perder la resolución de la imagen.

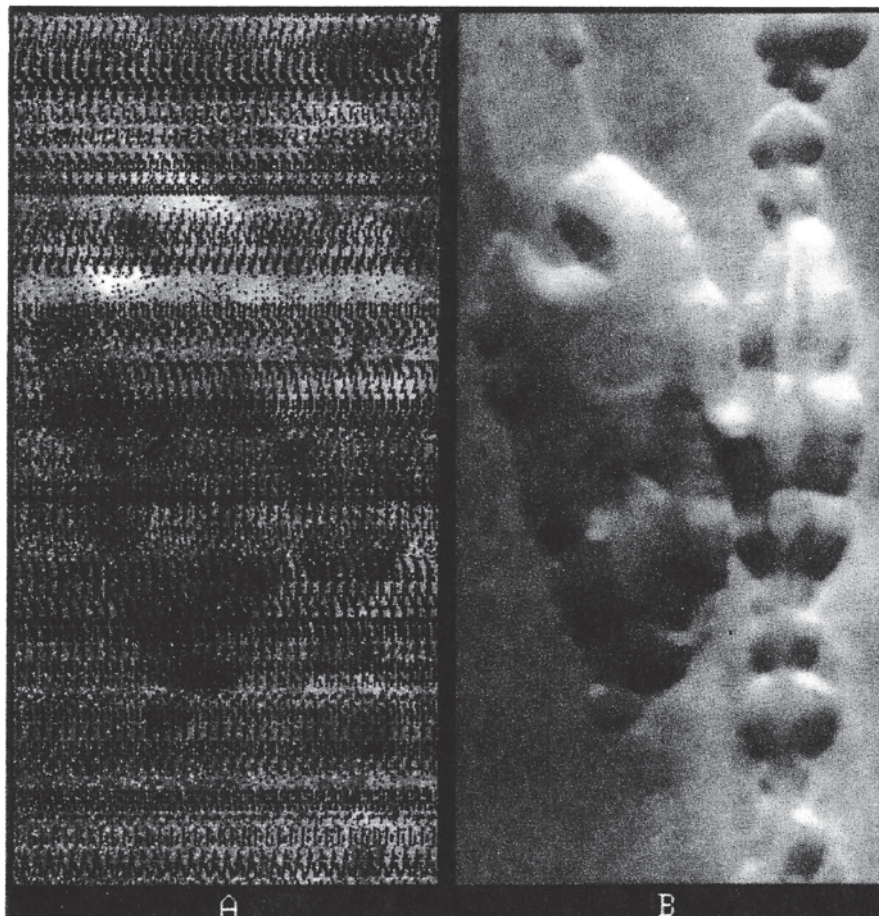


Figura 8. Imagen obtenida con la sonda Vikingo en una región de Marte llamada Tihonia. A) Imagen cruda (original), B) Resultado de la aplicación de 3 filtros de ruido con caja adaptable.

El algoritmo presentado por [Lee, 1983] es una modificación al filtro Sigma para suavizado de imágenes, una de las principales ventajas de los filtros de caja es su velocidad, dicha velocidad es independiente del tamaño de la caja. El filtrado con caja es aproximadamente 15 veces más rápido que la FFT para una imagen de 1024x1024, y no son necesarios los requerimientos de espacio en disco y memoria de la FFT.

El ruido de error de bit se debe a variaciones discretas de píxeles aislados que en una imagen tienen apariencia de “sal y pimienta” como ya se había mencionado antes. Un píxel con error de bit, a menudo tiene una intensidad bastante diferente al promedio de sus vecinos.

[Ramírez, 2004] también ha realizado investigación en el campo del diseño de filtros digitales, en su artículo se menciona que las señales transmitidas en el canal radio de un sistema de comunicaciones móviles celulares son susceptibles de interferencia co-canal, interferencia multiusuario y ruido aditivo. Se requieren técnicas avanzadas de procesamiento de arreglos de antenas inteligentes para extraer la señal deseada de los interferentes y el ruido. El principal reto es estimar la señal de interés con base a un filtraje espacial y temporal, con o sin el conocimiento a priori de la fuente de interés.

II. TIPOS DE FILTROS

II.1. Introducción

En este capítulo se mencionarán los tipos filtros digitales y analógicos, así como sus ventajas y desventajas, además se hablará sobre las aplicaciones más importantes de los filtros digitales, también se mencionan cuales son los filtros recursivos y no recursivos y que son los filtros digitales FIR e IIR. Finalmente se menciona sobre la Transformada Rápida de Fourier FFT.

II.2. Filtros digitales y analógicos

[Kamen y Heck, 2008] y [Proakis y Manolakis, 1999] mencionan que los filtros digitales se emplean en el procesamiento de señales para eliminar partes no deseadas de la misma, tales como ruido o sólo permitir el paso de un cierto rango de frecuencias, es decir, hacer un filtro divisor.

La figura 9 ilustra esta idea básica:



Figura 9. Filtrado de una señal.

Hay dos tipos principales de filtros: analógico y digital. Son bastantes diferentes tanto en su aspecto físico como en su modo de funcionamiento. Un filtro analógico emplea circuitos electrónicos con componentes discretos tales como resistencias, condensadores, amplificadores operacionales, entre otros, que sean requeridos para el filtrado deseado. Tales filtros son muy empleados

para reducción de ruido, mejora de señales de vídeo, ecualizadores gráficos y muchas otras áreas [Kamen y Heck, 2008].

Hay técnicas estándares bien asentadas para diseñar un filtro analógico con un requerimiento dado. En todas las diversas etapas la señal está siendo filtrada como un voltaje o corriente eléctrica, es decir, se involucra una magnitud física y real directamente de la señal analógica.

Un filtro digital emplea un procesador digital que efectúa operaciones matemáticas en valores muestreados de la señal. El procesador puede ser de propósito general, tal como cualquier computadora personal, un chip DSP (Procesador Digital de Señales) especializado o una FPGA programable.

La señal de entrada analógica debe ser muestreada y digitalizada usando un ADC (convertidor analógico-digital). El resultado son números binarios que representan los valores sucesivos muestreados. Estos son transferidos al procesador, el cual efectúa operaciones matemáticas en ellos. Las operaciones pueden ser desde filtros de promediado de la muestra actual con alguna de las anteriores hasta multiplicaciones por constantes de los valores de entrada o de instantes anteriores almacenados en memoria, para posteriormente sumar estos resultados de la multiplicación y dar una salida. Es decir, operaciones propias de teoría de sistemas lineales: convoluciones en el dominio temporal

(multiplicación en el dominio de la frecuencia) con otras señales prefijadas que consisten en una cadena de coeficientes. Para diseñar estos filtros suele usarse un impulso y desplazarlo sucesivas veces multiplicado por alguna constante, es decir, usando la transformada z .

Finalmente, si es necesario, los resultados de estos cálculos, que están representando valores muestreados de la señal filtrada, son enviados a través de un DAC (convertidor digital-analógico) para devolver la señal a una forma analógica. Por tanto, en un filtro digital la señal está siempre representada por una secuencia de números, en vez de un voltaje o una corriente [Kamen y Heck, 2008].

La figura 10 muestra el esquema básico de uno de estos sistemas:

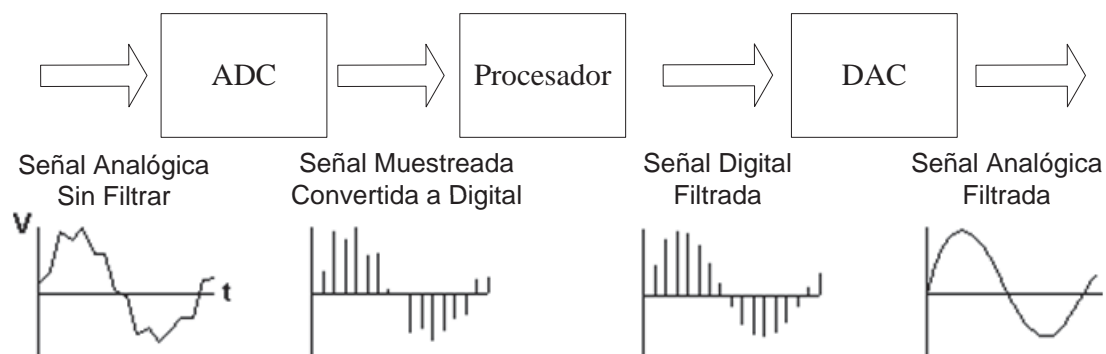


Figura 10. Esquema básico del proceso de filtrado de una señal.

II.3. Aplicaciones de los filtros

[Proakis y Manolakis, 1999] menciona que los filtros, sean estos digitales o analógicos, tienen un gran número de aplicaciones. En casi todas las aplicaciones subyace una de las dos posibles formas de considerar un filtro:

- ✓ En el dominio de la frecuencia. Como ya se sabe, los filtros actúan modificando la amplitud y fase de las componentes sinusoidales de una señal. Recordemos que cualquier señal puede considerarse como suma de sinusoides a través de la transformada de Fourier. Este hecho es usado en distintas aplicaciones como por ejemplo:
 - Eliminar determinadas componentes espectrales.
 - Enfatizar o atenuar algunas componentes espectrales.
 - Desfasar de forma distinta las diferentes componentes espectrales.

- ✓ En el dominio del tiempo. En este dominio, como es conocido, la relación entre la entrada y la salida de un filtro viene dada por la convolución. La convolución indica que la señal de salida, para un cierto instante, no es otra cosa que una suma (finita o infinita) de valores ponderados de la entrada en ciertos instantes (normalmente próximos). En función de los coeficientes de ponderación, que

dependen de la respuesta impulsional, se encuentran filtros que realzan un promedio en un determinado intervalo de tiempo, o que buscan las transiciones en el mismo.

A continuación se mencionan algunas de las posibles aplicaciones y se puede notar que corresponderán a una de las dos formas anteriores de considerar las señales.

Limitación del ancho de banda. En telecomunicaciones, existe muy a menudo la necesidad de limitar el ancho de banda de las señales. Algunos ejemplos de esta necesidad son:

- ✓ Multiplexado por división de frecuencia. Mediante esta técnica se comparte un mismo medio físico para transmitir diversas señales. Para ello se desplaza cada una de las *señales mensaje* mediante una portadora a una frecuencia diferente. A cada mensaje se le reserva un cierto ancho de banda. Normalmente, el ancho de banda de las señales mensaje es mayor que el asignado al mismo. Para evitar que los espectros de los distintos mensajes se traslapen se hace necesario que antes de mezclar los distintos mensajes se asegure que sus espectros no se traslaparán. El filtrado se puede realizar tanto en banda base como en

pasa-banda, pero en cualquier caso siempre antes de la mezcla de las señales.

- ✓ Multiplexado por división en tiempo. La idea es también transmitir por un mismo medio físico diversos mensajes, aunque en este caso la forma de realizarlo es mediante la transmisión cíclica de muestras de los distintos mensajes. Dado que se transmiten muestras, es necesario que las mismas se hayan obtenido de modo que no exista aliasing. Ello implica la necesidad de que el ancho de banda de las señales, antes de ser muestreadas, sea menor que la mitad de la frecuencia de muestreo.

Separación de señales multiplexadas en frecuencia. Tal y como se acaba de mencionar, cuando se multiplexan señales en frecuencia, cada una de ellas ocupa una banda de frecuencias diferente. Para extraer una señal es necesario el uso de un filtro que deje pasar la banda de frecuencias ocupada por la misma y atenúe el resto. Si se desean obtener todas las señales será necesario el uso de banco de filtros.

Controles de tono en audio. De todos son conocidos los controles de graves y agudos de los equipos de audio. Estos controles no son otra cosa que filtros cuyas características se pueden variar mediante un mando, de forma que se enfatice o atenúe a voluntad bien las frecuencias altas y bien las bajas.

Ecualizadores de audio. Permiten controlar de forma más flexible la respuesta en frecuencia. Consisten normalmente en agrupaciones en cascada o paralelo de filtros pasa-banda centrados en distintas frecuencias en los que es posible variar su ganancia. Dado que permiten variar individualmente la respuesta en frecuencia en un número más grande de frecuencias, permiten un control más exacto de la respuesta en frecuencia que los controles de tono.

Interpolación y diezmado de señales discretas. Esta es una aplicación típica de filtros digitales. La señal debe ser prefiltrada antes de ser diezmada, y cuando se interpola hay que filtrar la señal tras el relleno por ceros.

Conversión D/A. Para eliminar las frecuencias imagen, es necesario el uso de un filtro reconstructor tras la conversión D/A.

II.4. Ventajas e inconvenientes de los filtros digitales

A la hora de plantearse la realización de un filtro, hay que realizar una primera elección en cuanto a si el diseño se realizará de forma analógica o digital. Cada una de estas alternativas presenta ventajas e inconvenientes, siendo misión del ingeniero que realice el diseño decidir la mejor opción en cada caso [Proakis y Manolakis, 1999].

Se tiene principalmente las siguientes:

- ✓ Un filtro digital es programable, es decir, su funcionamiento está terminado por un programa almacenado en la memoria contigua al procesador. Esto significa que puede ser variado fácilmente sin afectar al hardware, mientras que la única manera de variar un filtro analógico es alterando el circuito.
- ✓ Los filtros digitales pueden ser fácilmente diseñados, probados e implementados en una computadora. Los analógicos pueden ser simulados, pero siempre hay que implementarlos a través de componentes discretos para ver su funcionamiento real.
- ✓ Las características de los filtros analógicos, particularmente los que contienen componentes activos, están sujetos a alteraciones y dependen de la temperatura. Los filtros digitales no sufren estos problemas y son extremadamente estables ante factores externos.
- ✓ A diferencia de los filtros analógicos, los digitales pueden manejar con mucha precisión las bajas frecuencias. Como la tecnología de los DSP va mejorando, el aumento de su velocidad permite que también sean

aplicados en el campo de la radio frecuencia (muy altas frecuencias), la cual en el pasado era exclusivamente dominio de la tecnología analógica.

- ✓ Los filtros digitales son mucho más versátiles a la hora de manipular la señal, que pueden llegar a variarla y tratarla radicalmente cambiando sus características.
- ✓ Un aspecto importante en la producción en serie es el ser capaz de realizar cientos o miles de filtros idénticos. En el caso de filtros digitales esto se logra sin más que ejecutar el mismo programa en cada DSP. Con filtros analógicos, y dado que todos los componentes tienen una cierta tolerancia, esto es difícil de lograr especialmente si el orden del filtro es alto.
- ✓ Los procesadores DSP más rápidos pueden manejar combinaciones complejas de filtros en paralelo o en serie (en cascada), haciendo los requerimientos de hardware relativamente simples y compactos en comparación con la circuitería analógica equivalente.

Los inconvenientes fundamentales de los filtros digitales son:

- ✓ Calidad del A/D y el D/A. Cuando un filtro digital se usa para filtrar señales analógicas como se muestra en la figura 11, es necesario un interfaz analógico. La calidad en cuanto a número de bits, linealidad, etc. de los convertidores va a condicionar mucho la calidad del filtrado analógico equivalente que se logre.

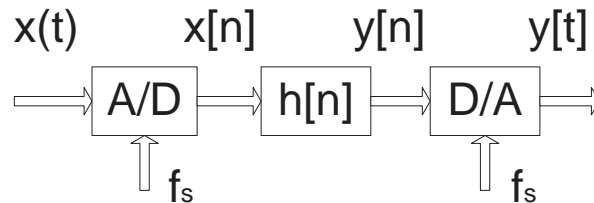


Figura 11. Filtrado de una señal analógica mediante un filtro digital.

- ✓ Velocidad del A/D y el D/A. Para poder filtrar señales analógicas usando filtros digitales es necesario un interfaz analógico. En la medida en que se disponga de convertidores suficientemente rápidos y precisos se podrá llevar a cabo el filtrado digital. Actualmente la tecnología ofrece convertidores de muy bajo costo y alta calidad en frecuencias de

audio ($f_s \approx 45Khz$), donde es posible encontrar convertidores de más de 20 bits. En el margen de frecuencias de video ($f_s \approx 13Mhz$) existen convertidores estándar de 8 bits, pero se pueden encontrar de 10 y 12. Existen también convertidores A/D y D/A para aplicaciones especiales capaces de trabajar a frecuencias muy elevadas (con pocos bits). De entre ellos se puede mencionar los que utilizan algunos osciloscopios digitales con frecuencias de muestreo de 2 Ghz, con 6 bits de resolución.

- ✓ Efectos de precisión finita. Se deben a redondeos en las operaciones que realizan los filtros digitales.
- ✓ Tiempo y costo del desarrollo del hardware. Un filtro digital implicará normalmente el desarrollo de un sistema microprocesador con interfaz analógico.

II.5. Filtros recursivos y no recursivos

Un filtro **no-recursivo** es aquel cuya salida está calculada exclusivamente a partir de valores de entrada ($Y_n = X_n + X_{n-1} + X_{n-2}...$), mientras que uno **recursivo** es aquel que además de los valores de entrada emplea valores previos de salida ($Y_{n-1}, Y_{n-2}...$), los cuales se almacenan en la memoria del

procesador. La palabra recursivo significa literalmente "volver hacia atrás" y se refiere al hecho de que valores de salida previamente calculados vuelven de nuevo para calcular los nuevos valores de salida.

Explicándolo así, puede parecer que los filtros recursivos requieren más cálculos para ser ejecutados. Pero la realidad es que un filtro recursivo generalmente requiere mucho menos coeficientes para que evalúe el procesador, es decir, que es de menor orden y es más corto, que un filtro no-recursivo que persiga una característica en frecuencia dada.

Hay quien prefiere una terminología alternativa, por lo que los filtros no-recursivos se conocen como filtro FIR (Respuesta al Impulso Finita) y los recursivos como filtros IIR (Respuesta al Impulso Infinita).

Estos términos se refieren a las diferentes respuestas al impulso de ambos tipos de filtros. La respuesta al impulso de un filtro digital es la secuencia de salida cuando se aplica un impulso unidad a su entrada (un impulso unidad es muy simple, tan sólo una secuencia consistente en un valor 1 en el instante de tiempo $t = 0$, seguido de ceros para todas las muestras siguientes), lo que se llama también una Delta de Kronecker.

Un filtro FIR es uno cuya respuesta es de una duración finita. Uno IIR es aquel cuya respuesta al impulso teóricamente continua para siempre debido a la recursividad con valores previos de salida que constantemente están siendo devueltos a la entrada. Pero realmente el término IIR no es muy afortunado dado que casi todos los filtros IIR reducen virtualmente su salida a cero a un tiempo dado, de hecho, antes que los FIR. De todas formas ambos acrónimos son muy coloquiales y de uso frecuente.

II.6. Filtros digitales FIR

Los filtros digitales FIR que provienen de sus siglas en inglés (Finite Impulse Response) Respuesta al Impulso Finita, son un tipo de filtro digital que si su entrada es un impulso (una delta de Kronecker) la salida será un número limitado de términos no nulos. Para obtener la salida sólo se emplean valores de la entrada actual y anterior. También se llaman filtros digitales no-recursivos. Su expresión en el dominio discreto es:

$$y_n = \sum_{k=0}^{N-1} b_k x(n-k) \quad (4)$$

El orden del filtro está dado por N , es decir, el número de coeficientes. También la salida puede ser expresada como la convolución de una señal de entrada $x[n]$ con un filtro $h[n]$:

$$y_n = \sum_{k=0}^{N-1} h_k x_{(n-k)} \quad (5)$$

La figura 12 muestra la estructura de un filtro FIR:

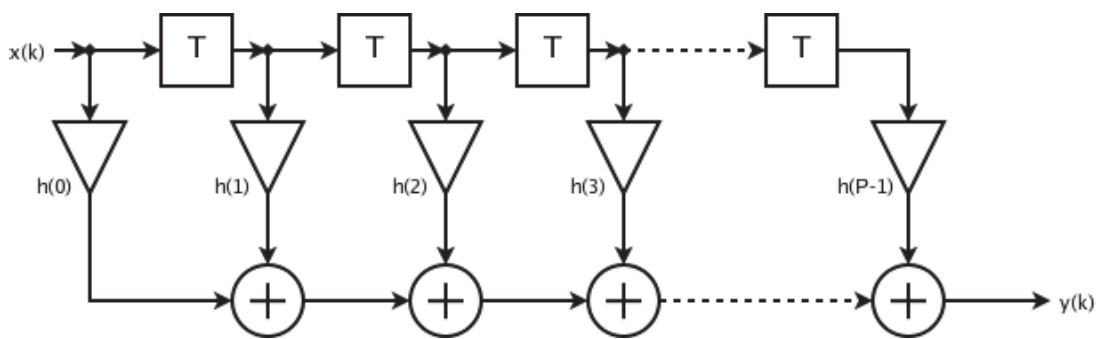


Figura 12. Estructura de un filtro FIR.

La cual puede verse reflejada en la aplicación de la transformada Z:

$$H(z) = \sum_{k=0}^{N-1} h_k z^{-k} = h_0 + h_1 z^{-1} + \dots + h_{N-1} z^{-(N-1)} \quad (6)$$

Se puede ver que es la misma entrada retardada cada vez más en el tiempo, multiplicada por diversos coeficientes y finalmente sumada al final. Hay muchas variaciones de esta estructura. Si se tiene una respuesta de frecuencia como objetivo, se consigue que la respuesta del filtro se asemeje más a ella cuanto más largo sea o número de coeficientes tenga.

Los filtros FIR son estables puesto que sólo tienen polos, es decir, elementos en el numerador en su función de transferencia. También tienen la ventaja que pueden diseñarse para ser de fase lineal, es decir, no introducen desfases en la señal, a diferencia de los IIR o los filtros analógicos. Por ese motivo tienen interés en audio.

Sin embargo, tienen el inconveniente de ser más largos al tener más coeficientes que los filtros IIR capaces de cumplir similares características. Esto requiere un mayor tiempo de cálculo que puede dar problemas en aplicaciones en tiempo real, como estudios de grabación o conciertos en directo.

II.7. Filtros digitales IIR

Los filtros digitales IIR que provienen de sus siglas en inglés (Infinite Impulse Response) Respuesta al Impulso Infinita, son un tipo de filtro digital que si su entrada es un impulso (una delta de Kronecker) la salida será un número ilimitado de términos no nulos, es decir, que nunca volverá a un estado de reposo. Para obtener la salida se emplean valores de la entrada actual y anteriores y, además, valores de salida anteriores que son almacenados en memoria y realimentados a la entrada. También se llaman filtros digitales recursivos. Su expresión en el dominio discreto es:

$$y[n] = \sum_{i=0}^P b_i x[n-i] - \sum_{j=1}^Q a_j y[n-j] \quad (7)$$

El orden del filtro está dado por el máximo entre P y Q. Una de las posibles estructuras para un filtro IIR se muestra en la figura 13:

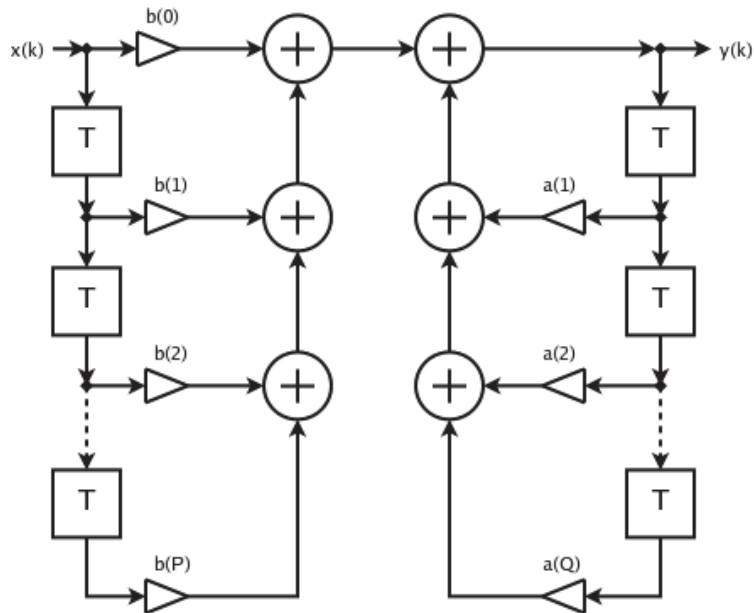


Figura 13. Estructura de un filtro IIR.

Se puede ver cómo la salida $y(k)$ es introducida de nuevo en el sistema. La transformada Z del mismo es:

$$H(z) = \frac{\sum_{i=0}^P b_i z^{-i}}{\sum_{j=0}^Q a_j z^{-j}} \quad (8)$$

Se puede observar que ahora se tiene un denominador, es decir, ceros además de polos, que son los causantes de las posibles inestabilidades que pueden comprometer la estabilidad y causalidad del sistema.

Las ventajas de los filtros IIR respecto a los FIR es que pueden conseguir una misma respuesta empleando un número de coeficientes en el filtro mucho menor, requiriendo un menor tiempo de cálculo. El inconveniente es, además de la inestabilidad ya comentada, la introducción de desfases en la señal, que pueden ser compensados pero a costa de añadir más coeficientes al filtro.

II.8. Filtros espaciales

El concepto de filtrado de una imagen está asociado a la representación de una imagen en el dominio de frecuencias. Cualquier filtro que se diseñe tendrá como objetivo modificar la contribución de determinados rangos de frecuencias a la formación de la imagen.

Aunque la terminología de la teoría del filtrado lineal es esencialmente del dominio de frecuencias, el teorema de convolución permite filtrar también en el dominio espacial aunque el sentido intuitivo de dichos filtros no sea tan claro como en el dominio de las frecuencias. Si $G(u,v)$ es la Transformada de Fourier de la imagen original y $H(u,v)$ es la Función de Transferencia del Filtro que se quiere aplicar, entonces:

$$F^{-1}[F(u,v) \cdot H(u,v)] = f(x,y) * h(x,y) \quad (9)$$

Por lo tanto se puede conocer que formas adoptarán las funciones $h(x,y)$ asociadas a determinados filtros de frecuencias. A estas funciones $h(x,y)$ se les denomina filtros espaciales.

Para diseñar un filtro espacial $h(x,y)$ en teoría se tendría que calcular los valores de la función sobre una retícula de igual tamaño que la definida por la imagen dada por $f(x,y)$, pero en la práctica puede considerarse que los valores de $h(x,y)$ decrecen muy rápidamente cuando se va alejando del origen. Así pues es práctica habitual discretizar dicha función para construir aproximaciones usando máscaras de distintos tamaños. Las más comunes serán las de tamaños 3×3 , 5×5 o 7×7 . Un ejemplo de máscara 3×3 se observa en la Tabla I.

Tabla I. Máscara de tamaño de 3×3 .

$W_{-1,-1}$	$W_{-1,0}$	$W_{1,0}$
$W_{0,-1}$	$W_{0,0}$	$W_{0,1}$
$W_{1,-1}$	$W_{1,0}$	$W_{1,1}$

Donde los $w_{i,j}$ son los valores discretizados en las posiciones de sus subíndices de la función $h(x,y)$. En el caso de funciones simétricas circularmente todos los valores de la máscara a igual distancia del origen tendrán el mismo valor.

Así pues el diseño de filtros espaciales se resume en calcular un conjunto de pesos $w_{i,j}$ que definan una máscara cuya Transformada de Fourier tenga el comportamiento de uno de los tres filtros básicos (pasaaltas, pasabajas o pasabandas).

II.8.1.Filtros Bit Error de Caja Adaptable

Sea $P(i,j)$ un pixel muestra de una imagen P , se tiene una caja de dimensión $(2K+1)$ por $(2L+1)$ centrada en $P(i,j)$, donde K y L son enteros > 0 . Para que un pixel sea válido en el arreglo de la caja debe estar entre el rango MIN y MAX .

Los parámetros MIN y MAX , permiten obtener la función Delta, D ,

$$D(i, j) = 0 \text{ si } P(i, j) < MIN, \text{ ó } P(i, j) > MAX \quad (10)$$

$$D(i, j) = 1 \text{ si } MIN \leq P(i, j) \leq MAX$$

La función Delta, D , se utiliza en las siguientes ecuaciones para discriminar entre los pixeles válidos y no válidos, se calcula:

$$S(i, j) = \sum_{m=i-K}^{i+K} \sum_{n=j-L}^{j+L} D(m, n)P(m, n) \quad (11)$$

$$N(i, j) = \sum_{m=i-K}^{i+K} \sum_{n=j-L}^{j+L} D(m, n) \quad (12)$$

$$SS(i, j) = \sum_{m=i-K}^{i+K} \sum_{n=j-L}^{j+L} (D(m, n)P(m, n))^2 \quad (13)$$

Donde:

$S(i, j)$ = Suma de los datos válidos

$N(i, j)$ = Cantidad de puntos válidos

$SS(i, j)$ = Suma de los cuadrados de los puntos válidos.

El filtro pasabajos $LPF(i, j)$, está definido como el promedio de la caja excluyendo los datos no válidos.

$$LPF(i, j) = \frac{S(i, j)}{N(i, j)} \quad (14)$$

El valor de reemplazo $R(i, j)$, está definido como el valor promedio de la caja excluyendo los valores inválidos y el pixel central.

$$R(i, j) = \frac{S(i, j) - D(i, j)P(i, j)}{N(i, j) - D(i, j)} \quad (15)$$

Cuando $P(i, j)$ está clasificado como un dato no válido, se utiliza $R(i, j)$ en lugar de $LPF(i, j)$ como el valor de reemplazo, debido a que el promedio no contiene los valores no validos de $P(i, j)$.

La varianza de la caja $V(i, j)$, está dada por la ecuación 16:

$$V(i, j) = \frac{SS(i, j)}{N(i, j)} - \frac{S(i, j)^2}{N(i, j)^2} \quad (16)$$

La desviación estándar, $\sigma(i, j)$ es la raíz cuadrada de la varianza.

Este algoritmo de eliminación de datos erróneos determina si $P(i, j)$ es un dato no válido comparándolo con la estadística local de la caja, por otro lado, si $P(i, j)$ se considera un dato válido, este permanece igual. Si $P(i, j)$ está marcado como un dato no válido, el algoritmo puede hacer cero el pixel o reemplazarlo por el valor de $R(i, j)$. La opción de reemplazar a cero el pixel es llamado

filtro STDZ. La opción STDZ permite múltiples aplicaciones de eliminación de píxeles no válidos para imágenes extremadamente ruidosas. $P(i, j)$ se clasifica como un dato no válido solo si cumple las condiciones de la ecuación siguiente:

$$|P(i, j) - LPF| > C\sigma(i, j) \quad \& \quad |P(i, j) - LPF| > TOL \quad (17)$$

Donde:

C = Constante (usualmente de 1 a 2).

TOL = Constante que es el valor del umbral mínimo que debe ser excedido (usualmente entre 2 y 10 para datos de 8 bits).

II.8.2. Filtro Sigma de Caja Adaptable

Se han desarrollado muchas técnicas para el suavizado de datos ruidos pero en general la desventaja principal de un filtro pasabajas sencillo es que las partes afiladas de la imagen y los detalles se reducen con la presencia del ruido. El uso de algoritmos con filtros de errores de bits en datos ruidosos es inapropiado debido a que se puede eliminar información.

[Lee, 1983] concluyó que un método muy sencillo que él le llamó Filtro Sigma, es muy efectivo y más rápido que otros métodos más elaborados. La

idea básica del Filtro Sigma es reemplazar cada pixel por el promedio de sus pixeles vecinos que tienen una intensidad dentro de un rango fijo σ del pixel central. Como resultado, solamente los pixeles que tienen los valores que pertenecen a la población son promediados. El Filtro Sigma es idéntico al filtro pasabajas, con un rango de intensidad válido y limitado (establecido por MIN y MAX), excepto que el rango válido de intensidad varía con la intensidad del pixel central.

El valor de Sigma se calcula par la imagen entera, el rango válido de intensidad está definido por el valor del pixel central más ó menos un múltiplo (usualmente 1 o 2) de Sigma (σ). El método propuesto por [Eliason y McEwn, 1990] es una modificación del Filtro Sigma a un filtro adaptable, simplemente se calcula un Sigma $\sigma(i, j)$ para cada caja, en vez de un Sigma (σ) para la imagen entera. Este filtro Sigma adaptable conserva detalles finos de la imagen, incluso en áreas de bajo contraste, pero no reduce el ruido de moteo significativamente.

Sea $X(i, j)$ la intensidad o nivel del gris del pixel (i, j)

$$\begin{aligned} \delta(K, l) &= 1, \text{ si } \left(\widehat{X}(i, j - \Delta) \right) \leq \widehat{X}(K, l) \leq \left(\widehat{X}(i, j + \Delta) \right) \\ \delta(K, l) &= 0, \text{ si estuviese fuera del rango anterior} \end{aligned} \quad (18)$$

Donde \hat{X} es el pixel suavizado, entonces:

$$\hat{X}(i, j) = \frac{\sum_{K=i-n}^{n+j} \sum_{l=j-m}^{m+j} \delta(K, l) \hat{X}(K, l)}{\sum_{k=i-n}^{n+j} \sum_{l=j-m}^{m+j} \delta(K, l)} \quad \Delta = C \cdot \sigma \quad (19)$$

O bien, para eliminar “spot” de ruido aislado, sin destruir las propiedades delgadas y detalles finos de la imagen, se considera:

$$\begin{aligned} \hat{X} &= \text{promedio}(2 - \sigma), \quad \text{si } M > K \\ \hat{X} &= \text{promedio de sus vecinos inmediatos}, \quad \text{si } M \leq K \end{aligned} \quad (20)$$

II.9. Filtros Adaptativos

[Eliason y McEwn, 1990] y [Lee, 1983] mencionan que un filtro adaptativo no es más que un filtro cuyos coeficientes varían con el tiempo. Lo que los hace especialmente interesantes es que la variación de los coeficientes es automática. En otras palabras, diseñar un filtro adaptativo consistirá en determinar la regla de variación de los coeficientes. El resto será automático.

Existen dos razones fundamentales para elegir un filtro adaptativo frente a un filtro de coeficientes fijos:

- Porque se sabe que filtro usar a priori.

- Porque el filtro óptimo que se necesita, debido a que la señal no sea estacionaria, deba variar con el tiempo.

II.9.1. Ejemplos de sistemas adaptativos

A continuación se presentan algunos sistemas adaptativos la comunidad en general se encuentra familiarizada:

- **CAG.** Es un circuito usado ampliamente que permite variar su ganancia en función de la amplitud de la señal de entrada. Cuando la señal de entrada es de poca potencia el factor de amplificación es grande y a la inversa, de modo que la potencia a la salida tiene una amplitud prácticamente independiente de la amplitud de entrada.
- **Pupila.** Sistema biológico que regula su apertura en función inversa a la luz incidente, de modo que el margen de variación de energía luminosa que reciben las células de la retina sea mucho menor que las variaciones de luminosidad que se pueden encontrar en la naturaleza.

Los ejemplos anteriores son ejemplos sencillos de sistemas que varían su comportamiento en función de las señales recibidas. En el caso de sistemas artificiales para el filtrado de señales, aunque conceptualmente sería posible

pensar en sistemas analógicos adaptativos, en la práctica únicamente se han desarrollado sistemas adaptativos discretos de cierta complejidad.

II.9.2. Características fundamentales de los sistemas adaptativos

Las características fundamentales de los sistemas adaptativos son:

- Automáticamente se adaptan (auto-optimizan) de acuerdo al entorno cambiante (no estacionario).
- Pueden ser entrenados: pueden aprender cómo funcionar de forma óptima.
- No hay métodos de síntesis de filtros adaptativos, se autodiseñan. Lo que existen son reglas de aprendizaje (algoritmos de adaptación).
- El estudio se hace considerando los sistemas no lineales variantes.
- Son más complicados y difíciles de analizar que los sistemas fijos. Su comportamiento, sin embargo, es mejor ante señales de características desconocidas o variantes. El análisis se suele hacer por simulación.

II.10. Transformada rápida de Fourier FFT

La Transformada Rápida de Fourier FFT es una técnica para realizar de una manera más eficiente y con un menor costo computacional el cálculo de la DFT (Transformada de Fourier Directa) y la IDFT (Transformada de Fourier Inversa).

Una Transformada de Fourier es el paso de una señal en el dominio del tiempo al dominio de la frecuencia o lo que es lo mismo la descomposición de una señal periódica en una suma de senos y cosenos de diferentes magnitudes, concretamente la serie de Fourier o su frecuencia fundamental y los diferentes armónicos. De este modo, se puede realizar un análisis espectral de la señal:

$$x(\omega) = \int_{-\infty}^{\infty} x(t) \cdot e^{-j\omega t} dt \quad (21)$$

La Transformada inversa de Fourier se calcula con:

$$x(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} x(\omega) \cdot e^{j\omega t} d\omega \quad (22)$$

Estas transformadas hacen referencia a las señales continuas reales, pero no es posible realizar una Transformada de Fourier en tiempo continuo en una computadora, ya que ésta trabaja con valores finitos y cuantificados. Por tanto, la señal ha de ser muestreada y de esa manera se realiza una Transformada de Fourier en tiempo discreto o DFT.

$$x[n] = \sum_{n=\langle N \rangle} a_k e^{jk\Omega t} \quad \text{con} \quad \Omega = \frac{2\pi}{N} \quad (23)$$

La serie de Fourier discreta es simplemente una modificación de la serie de Fourier tradicional, pero sustituyendo las integrales por sumatorios de las muestras, y el periodo ahora en vez de ser T (número real) será N, siendo N un número entero.

Sin embargo, esto es demasiado costoso de calcular. El número de operaciones crece de manera cuadrática con el número de muestras:

$$\text{N}^\circ \text{ operaciones} = (\text{N}^\circ \text{ de muestras})^2 \quad (24)$$

Sin embargo, mediante el algoritmo de Transformación Rápida de Fourier, este cálculo se reduce a una función de crecimiento log-lineal con el número de muestras:

$$\text{N}^\circ \text{ operaciones} = (\text{N}^\circ \text{ de muestras}) * \log_2(\text{N}^\circ \text{ de muestras}) \quad (25)$$

Para que el algoritmo de la FFT sea eficiente, el número de muestras tomadas deberá ser una potencia de 2. En caso de no ser así, se divide en partes el

número de muestras en sumas de potencias de 2, haciendo perder eficacia al algoritmo.

La descomposición espectral de una frecuencia permite programar compresores en tratamiento digital de imagen, filtrado digital mediante convolución con respuestas al impulso, ecualizadores digitales, líneas de transmisión, hasta campos como la estadística, economía, geología, etc.

Este trabajo de tesis utiliza la transformada de Fourier para la manipulación de imágenes en dos dimensiones por la facilidad que presenta para realizar este tipo de operaciones, además se estará programando sobre la plataforma de MATLAB el cual es un poderoso software matemático que ofrece un entorno de desarrollo integrado con un lenguaje de programación propio.

III. CONCEPTOS BÁSICOS DE LAS IMÁGENES

III.1. Introducción

[Cuevas y Zaldivar, 2007] menciona que la implementación de algoritmos en visión por computadora resulta muy costosa en tiempo ya que se requiere de la manipulación de punteros, gestión de memoria, etc. Hacerlo en lenguaje C++ (que por sus características compartidas de alto y bajo nivel lo hacen el más apropiado para la implementación de algoritmos de visión computacional) supondría la inversión de tiempo y sin la seguridad de que lo que se quiere implementar funcionará. Además utilizar C++ para el periodo de prueba exige un tiempo normal de corrección de errores debido al proceso de implementación del algoritmo, es decir errores programáticos efectuados por ejemplo al momento de multiplicar dos matrices, etc. Todos estos problemas pueden ser resueltos si la implementación de prueba es realizada en MATLAB

utilizando su toolbox de procesamiento de imágenes, con ello el tiempo de implementación se convierte en el mínimo con la confianza de utilizar algoritmos científicamente probados y robustos. El toolbox de procesamiento de imágenes contiene un conjunto de funciones de los algoritmos más conocidos para trabajar con imágenes binarias, transformaciones geométricas, morfología y manipulación de color que junto con las funciones ya integradas en MATLAB permite realizar análisis y transformaciones de imágenes en el dominio de la frecuencia (transformada de Fourier y Wavelets).

III.2. Conceptos básicos de las imágenes

En MATLAB una imagen a escala de grises es representada por medio de una matriz bidimensional de $m \times n$ elementos en donde n representa el número de píxeles de ancho y m el número de píxeles de largo. El elemento v_{11} corresponde al elemento de la esquina superior izquierda la cual se puede ver en la figura 14, donde cada elemento de la matriz de la imagen tiene un valor de 0 (negro) a 255 (blanco) [Cuevas y Zaldivar, 2007].

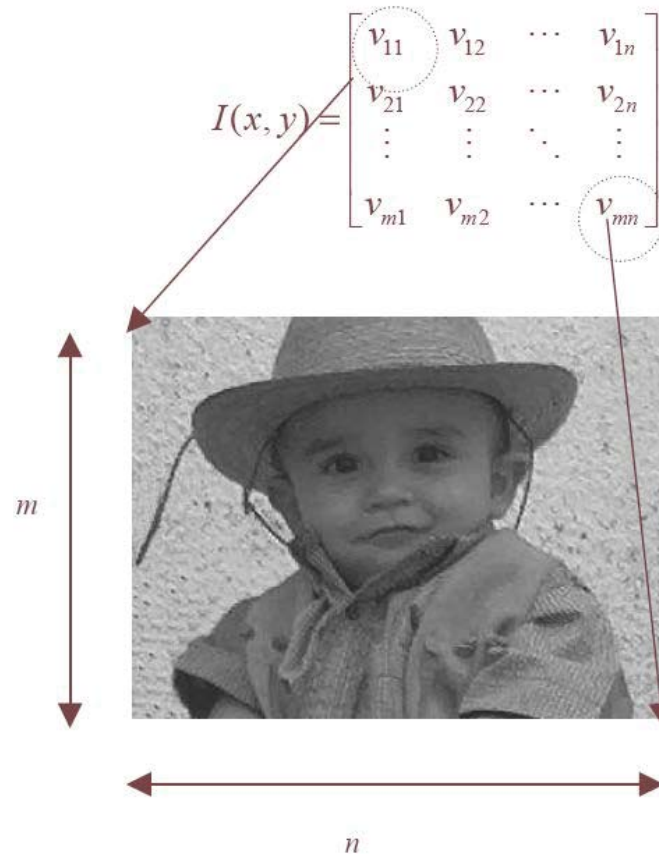


Figura 14. Representación de una imagen a escala de grises en MATLAB.

Por otro lado una imagen de color RGB (la más usada para la visión computacional, además de ser para MATLAB la opción default) es representada por una matriz tridimensional $m \times n \times p$, donde m y n tienen el mismo significado que para el caso de las imágenes de escala de grises mientras p representa el plano que para RGB, el cual puede ser 1 para el rojo, 2 para el verde y 3 para el azul. La figura 15 muestra detalles de estos conceptos.

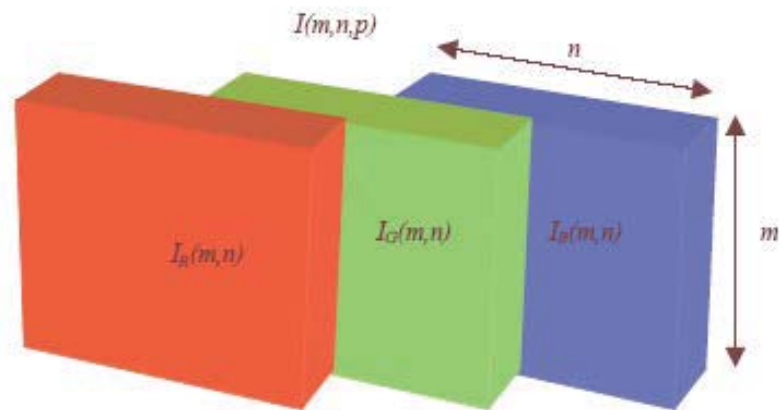
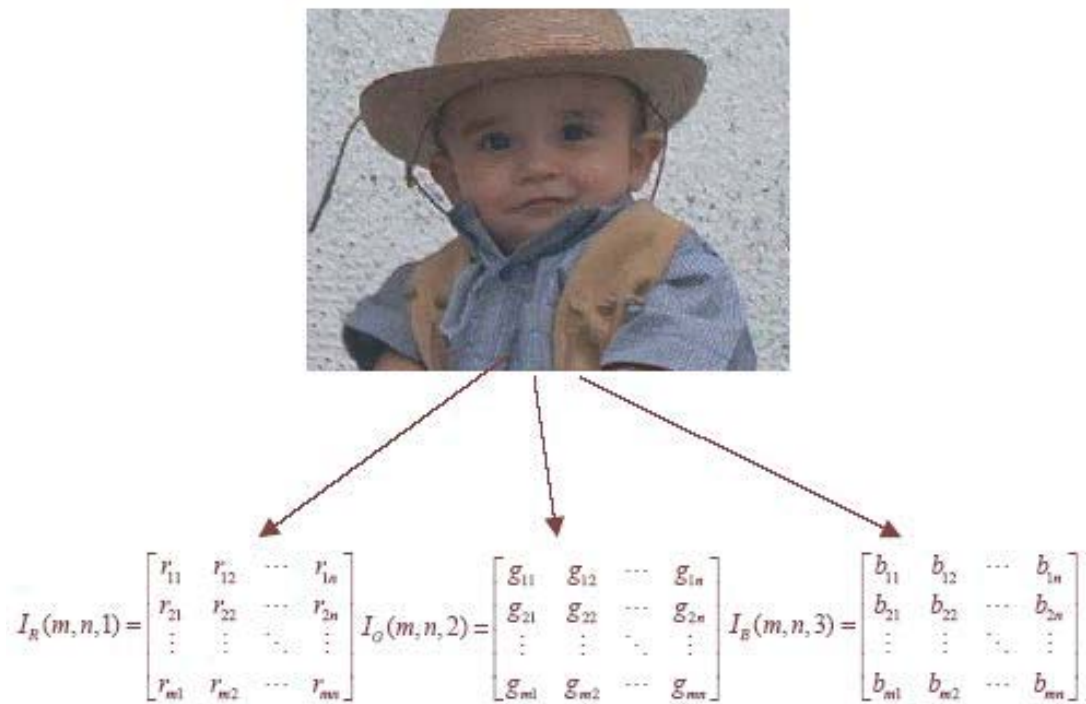


Figura 15. Representación de una imagen a color en RGB en MATLAB.

III.2.1. Lectura y escritura de imágenes a través de archivo

Para leer imágenes contenidas en un archivo al ambiente de MATLAB se utiliza la función `imread`, cuya sintaxis es:

`Imread('nombre del archivo')`

Donde el nombre del archivo es una cadena de caracteres conteniendo el nombre completo de la imagen con su respectiva extensión, los formatos de imágenes que soporta MATLAB son los mostrados en la tabla II.

Tabla II. Formatos y extensiones soportadas por MATLAB.

Formato	Extensión
TIFF	.tiff
JPEG	.jpg
GIF	.gif
BMP	.bmp
PNG	.png
XWD	.xwd

Para introducir una imagen guardada en un archivo con alguno de los formatos específicos en la tabla I solo tiene que usarse la función `imread` y asignar su resultado a una variable que representará a la imagen (de acuerdo a la

estructura de la figura 14 para escala de grises y figura 15 para RGB). De tal forma que si se quisiera introducir la imagen contenida en el archivo *uabc.jpg* a una variable para su procesamiento en MATLAB, entonces se tendría que escribir en línea de comandos:

```
>>imagen = imread('uabc.jpg');
```

Con ello la imagen contenida en el archivo *uabc.jpg* quedará contenida en la variable *imagen*.

Una vez que la imagen está contenida en una variable de MATLAB es posible utilizar las funciones para procesar la imagen. Por ejemplo, una función que permite encontrar el tamaño de la imagen es *size (variable)*:

```
>>[m, n] = size(imagen);
```

En donde *m* y *n* contendrán los valores de las dimensiones de la imagen. Para grabar el contenido de una imagen es un archivo se utiliza la función *imwrite(variable, 'nombre del archivo')*, en donde *variable* representa la variable que contiene a la imagen y *nombre del archivo* el nombre del archivo con su respectiva extensión de acuerdo a la tabla I. Suponiendo que la variable *imagen2* contiene la imagen que se necesita grabar en el archivo *uabc2.jpg* se tendría que escribir:

```
>>imagen2 = imwrite(imagen2, 'uabc2.jpg');
```

Después que se realiza un procesamiento con la imagen, es necesario desplegar el resultado obtenido, la función `imshow(variable)` permite desplegar la imagen en una ventana en el ambiente de trabajo de MATLAB. Si la variable a desplegar por ejemplo, es `cara` al escribir en la línea de comandos:

```
>>imshow(cara);
```

Se obtendría la imagen de la figura 16.

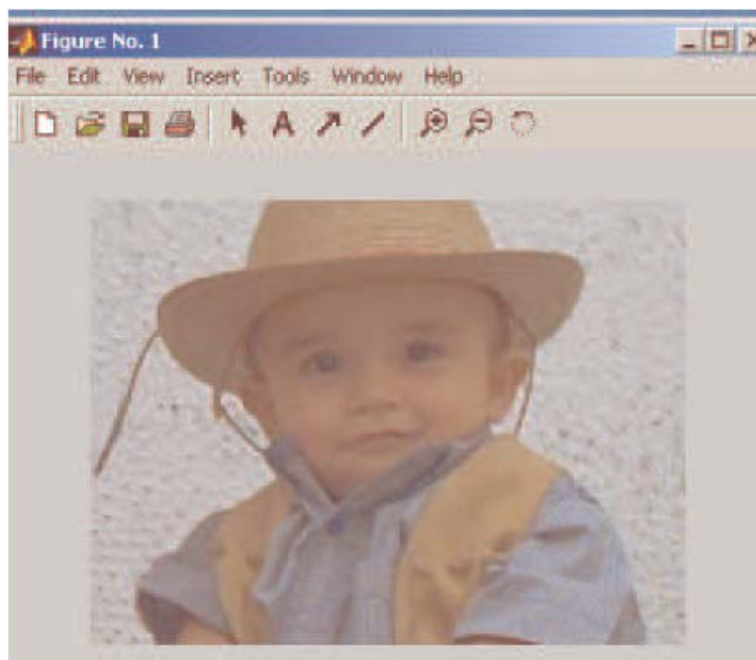


Figura 16. Imagen mostrada al utilizar la función `imshow` en MATLAB.

III.2.2. Acceso a píxel y planos en las imágenes

[Moore, 2007] y [Pérez, 2002] coinciden con que el acceso a píxel de una imagen es una de las operaciones más comunes en visión computacional y en MATLAB está sumamente simplificado; solo bastará con indexar el píxel de interés en la estructura de la imagen. Por ejemplo en la figura 17 se puede ver la consideración de tener una imagen llamada imagen1 en escala de grises y se desea obtener su valor de intensidad en el píxel especificado por $m=100$ y $n=100$, sólo se tendría que escribir:

```
>>imagen1(100, 100)
```

```
ans =
```

```
84
```

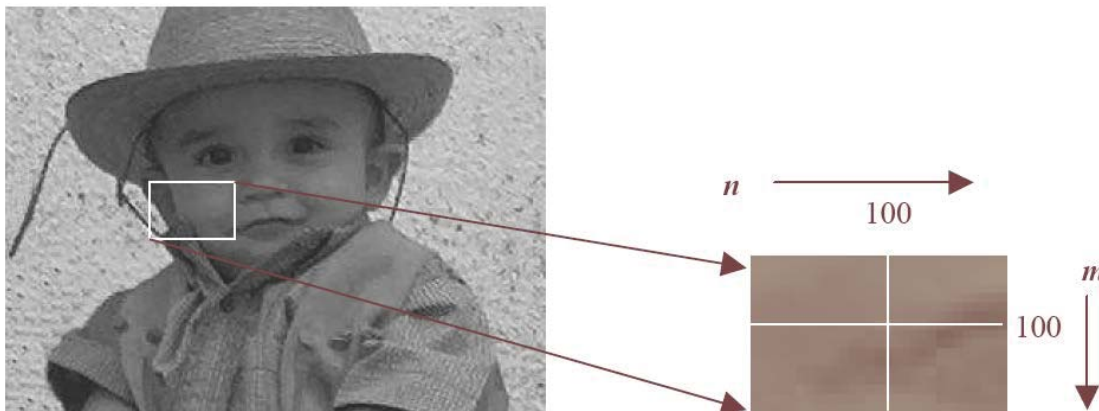


Figura 17. Obtención del valor de un píxel de imagen1.

De igual forma si se desea cambiar el valor de este píxel a negro, es decir asignarle el valor de 0 lo que tendría que escribirse en la línea de comandos es:

```
>>imagen1(100,100) = 0;
```

En el caso de imágenes a escala de grises estas solo tienen un plano, constituido por la matriz $m \times n$ que contiene los valores de intensidad para cada índice. Sin embargo las imágenes de color cuentan con más de un plano. En el caso de imágenes RGB, estas cuentan con 3 planos uno para cada color que representa, si se considera ahora que la imagen RGB contenida en la variable *imagen2* es la mostrada en la figura 16, y si se desea obtener cada uno de los planos que la comprenden, entonces se tendría que escribir:

```
>>planeR = imagen2(:, :, 1);
```

```
>>planeG = imagen2(:, :, 2);
```

```
>>planeB = imagen2(:, :, 3);
```

Los planos resultantes por los anteriores comandos se muestran en la figura 18.



Figura 18. Planos de la imagen a) rojo, b) verde y c) azul.

Si se deseara manipular un píxel de una imagen a color RGB este tendrá un valor para cada uno de sus planos correspondientes, por ejemplo si se tuviera la imagen RGB contenida en la variable `imagen2` y se desea obtener el valor del píxel $m = 100$ y $n = 100$ para cada uno de los diferentes planos R, G y B, se tendría que escribir:

```
>>valueR= imagen2(100,100,1);
```

```
>>valueG= imagen2(100,100,2);
```

```
>>valueB= imagen2(100,100,3);
```

Lo cual dará como resultado una tripleta de valores, [Cuevas y Zaldivar, 2007].

III.2.3. Sub-muestreo de imágenes

En ocasiones es necesario hacer cálculos que requieren procesar por completo la imagen, en estos casos hacerlo sobre la resolución original de la imagen sería muy costoso. Una alternativa más eficiente, resulta el sub-muestreo de la imagen. Sub-muestreo significa generar una imagen a partir de tomar muestras periódicas de la imagen original, de tal forma que esta quede más pequeña. Si se considera la imagen $I(m,n)$ definida como:

$$I(m,n) = \begin{bmatrix} v_{11} & v_{12} & v_{13} & \cdots & v_{1n} \\ v_{21} & v_{22} & v_{23} & \cdots & v_{2n} \\ v_{31} & v_{32} & v_{33} & \cdots & v_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ v_{m1} & v_{m2} & v_{m3} & \cdots & v_{mn} \end{bmatrix} \quad (26)$$

Y se desea sub-muestrear la imagen para obtener la mitad de su tamaño original, así la nueva imagen quedaría compuesta por los elementos tomando uno sí y otro no de la imagen original:

$$I_{s_2}(m,n) = \begin{bmatrix} v_{11} & v_{13} & v_{15} & \cdots & v_{1(n-2)} \\ v_{31} & v_{33} & v_{35} & \cdots & v_{3(n-2)} \\ v_{51} & v_{53} & v_{55} & \cdots & v_{5(n-2)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ v_{(m-2)1} & v_{(m-2)3} & v_{(m-2)5} & \cdots & v_{(m-2)(n-2)} \end{bmatrix} \quad (27)$$

El mismo concepto se explica gráficamente en la figura 19.

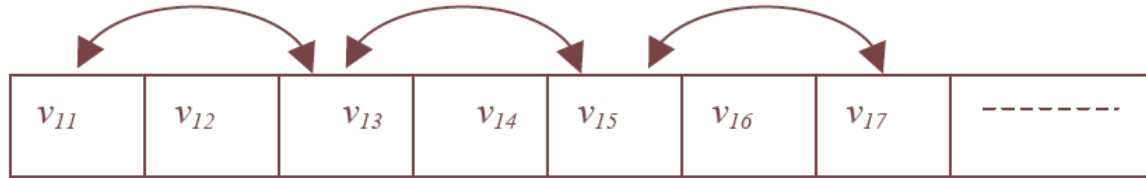


Figura 19. Ejemplo de sub-muestreo.

III.2.4. Tipo de dato de los elementos de una imagen

Los elementos que constituyen una imagen en MATLAB tienen el formato entero *uint8*, que es un tipo de dato que puede variar de 0 a 255, sin poder soportar decimales y valores que salgan fuera de ese rango. Lo anterior resulta una desventaja principalmente en aquellos casos donde se implementan algoritmos que trabajan con este tipo de datos para realizar operaciones de división o multiplicación por tipo de dato flotante. En estos casos es necesario transformar la imagen de tipo de dato *uint8* a *double*.

IV. DESARROLLO DEL PROYECTO

IV.1. Desarrollo

En este capítulo, se presenta el desarrollo e implementación de las interfaces gráficas de usuario correspondientes a los filtros espaciales para el procesamiento de imágenes ruidosas. Para el desarrollo de esta aplicación se empleó MATLAB con el apoyo de la herramienta GUIDE, dando como resultado una interfaz gráfica la cual es amigable y fácil de utilizar por los usuarios, por lo tanto se considera de tipo didáctica. Este trabajo toma en cuenta el uso de filtros con cajas adaptables y no-adaptables de tamaños de 3×3 , 5×5 y 7×7 . Los filtros que se implementaron con caja no-adaptable son: Pasabajas (filtro promedio), pasa altas, raíz cuadrada, cuadrático y sigma global. Con respecto a los filtros de caja adaptable, se implementaron los siguientes: Bit-Error y Sigma de caja local. Se presenta un estudio experimental para saber cuáles filtros son más adecuados de utilizarse dependiendo del tipo de ruido. Para validar estos experimentos se utilizaron

distintas imágenes y distintos tipos de ruido, tales como el ruido Gaussiano, Poisson y tipo Sal y Pimienta.

En la figura 20, se muestra la interfaz gráfica principal del software desarrollado, este programa se realizó bajo la plataforma de MATLAB, el programa desarrollado es interactivo y fácil de utilizar por los usuarios. A continuación se describirá brevemente como opera este programa y el tipo de procesamiento de imágenes que se puede realizar.



Figura 20. Interfaz gráfica principal del software.

Primeramente el usuario debe abrir la imagen que desea procesar o filtrar, tal como se observa en la figura 21. El software permite abrir archivos en format BMP o JPG. Por ejemplo, cuando se elige abrir una imagen tipo BMP,

aparece el recuadro que se presenta en la figura 22, para que el usuario pueda seleccionar el archivo de imagen que desea abrir. Luego se debe seleccionar el tamaño de la caja del filtro. El programa permite elegir los tamaños de caja más comunes, los cuales son: 3x3, 5x5 y 7x7.



Figura 21. Interfaz para abril archivo de imágenes BMP ó JPG.

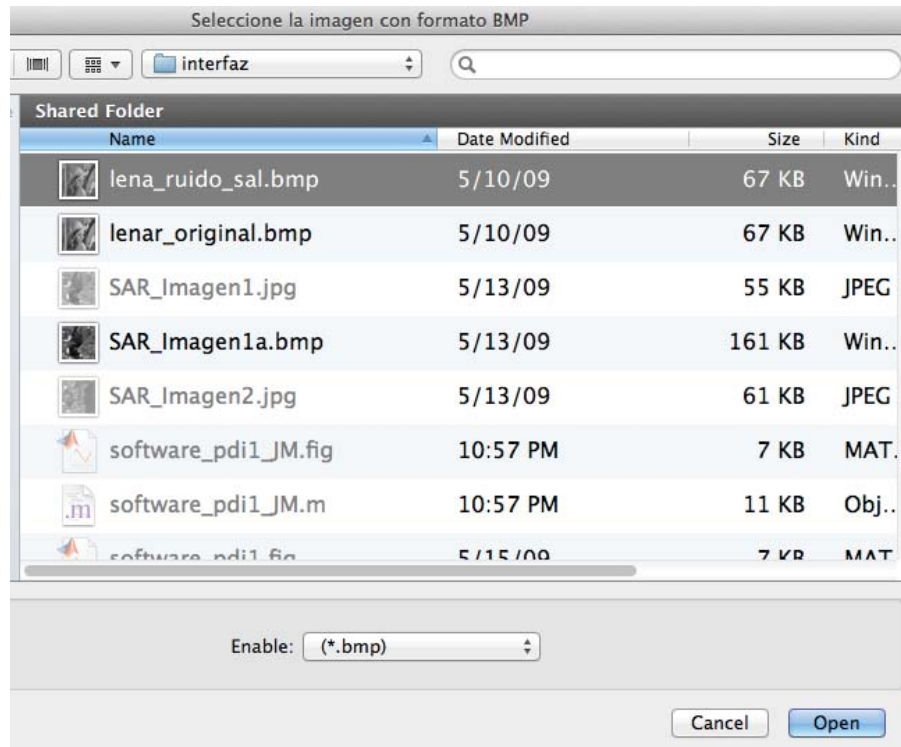


Figura 22. Ventana auxiliar para abrir archivo de imágenes BMP ó JPG.

Una vez abierta la imagen, ya es posible realizar diferentes tipos de procesamiento o filtrado. Por ejemplo, se pueden seleccionar filtros *Adaptables* ó *No-Adaptables*, tal como se muestra en la figura 23. Si el usuario selecciona la opción de filtros No-Adaptables, podrá escoger los siguientes filtros: Pasa-Bajas (Promedio), Pasa-Altas, Raíz cuadrada, Cuadrático y Sigma, tal como se puede apreciar en la figura 24.



Figura 23. Interfaz gráfica para seleccionar el tipo de filtro.



Figura 24. Interfaz gráfica para seleccionar filtros No-adaptables.

IV.2. Filtro Pasa-Bajas (Promedio)

Tal como se mencionó en el capítulo I.1.1, para realizar un filtro promedio en el dominio del espacio se realiza una convolución (barrido) del Kernel sobre la imagen a filtrar. Para ello se sigue el Teorema de Convolución en el espacio

$$g(x, y) = h(x, y) * f(x, y) \quad (2)$$

1. Cada pixel de la nueva imagen se obtiene mediante la sumatoria de la multiplicación del Kernel (w) por los pixeles contiguos:

$$g(x, y) = \sum \sum f(i, j) w(i, j) \quad (3)$$

donde $f(i, j)$ es la imagen a filtrar, $w(i, j)$ es el Kernel del filtro, en este caso es el filtro promedio y $g(x, y)$ es el nuevo pixel de la imagen filtrada.

El Kernel del filtro promedio para una caja de 3x3 es:

$$W = \begin{matrix} \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \\ \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \\ \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \end{matrix},$$

para una caja de 5x5 el Kernel del filtro promedio es:

$$W = \begin{matrix} \frac{1}{25} & \frac{1}{25} & \frac{1}{25} \\ \frac{1}{25} & \frac{1}{25} & \frac{1}{25} \\ \frac{1}{25} & \frac{1}{25} & \frac{1}{25} \\ \frac{1}{25} & \frac{1}{25} & \frac{1}{25} \\ \frac{1}{25} & \frac{1}{25} & \frac{1}{25} \end{matrix},$$

el Kernel del filtro promedio con caja de 7x7 es:

$$W = \begin{bmatrix} \frac{1}{49} & \frac{1}{49} & \frac{1}{49} \\ \frac{1}{49} & \frac{1}{49} & \frac{1}{49} \\ \frac{1}{49} & \frac{1}{49} & \frac{1}{49} \end{bmatrix}$$

Por ejemplo si el usuario selecciona el filtro Pasa-Bajas (Promedio), aparece un recuadro como el que se presenta en la figura 25. Está indicando que la computadora está realizando el proceso de cálculo de la Ecuación (2). Algo similar ocurre cuando se eligen los demás filtros: Pasa-Altas, Raíz cuadrada y Cuadrático, que en principio solamente requieren que se le proporcione la imagen a filtrar e indique el tamaño de la caja.

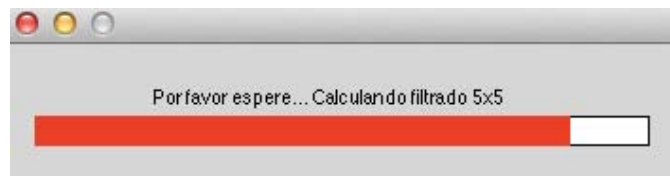


Figura 25. Barra de progreso indicando que se está realizando la tarea de procesamiento o filtrado de la imagen.

IV.3. Filtro Sigma

La metodología del cálculo de este filtro está reportada en [Lee, 1983]. Cuando el usuario selecciona el filtro tipo **Sigma**, aparece una pequeña interfaz gráfica, tal como se ilustra en la figura 26, esto con la finalidad de que el usuario puede introducir los parámetros de configuración de este filtro, los cuales son: Rango de Sigma (C), Mínima cantidad de pixeles dentro de rango de intensidad (K) y Tamaño de la caja del filtro. Luego cuando el usuario presiona el botón de Aceptar, se ejecuta el proceso de filtrado con el algoritmo Sigma, posteriormente el programa arroja la imagen filtrada, el tiempo que tardó todo el proceso, la cantidad de pixeles probables a corregir y la cantidad de pixeles corregidos.



Figura 26. Interfaz gráfica del filtro tipo Sigma con umbral global.

Por otra parte, cuando el usuario elige los filtros *Adaptables*, tal como se muestra en la figura 27. El usuario podrá elegir el filtro Bit-Error o el filtro Sigma adaptable.



Figura 27. Interfaz gráfica para seleccionar filtros adaptables.

IV.4. Filtro Bit-Error

Los detalles de la metodología de este filtro están reportados en [Eliason y McEwn, 1990]. Sin embargo, en la sección II.8.1 de este trabajo de tesis, se presenta una breve descripción de los pasos a seguir para implementar el filtro Bit-Error. Cuando el usuario elige el filtro **Bit-Error**, aparecerá la pantalla mostrada en la figura 28. Luego el usuario deberá introducir los principales parámetros que configuran a este filtro, los cuales son: Rango de Sigma (C), valor mínimo del pixel (MIN), valor máximo del pixel (MAX),

Tamaño de la caja del filtro, Tolerancia (TOL), después de que el usuario presiona el botón de Aceptar, inicia el proceso de filtrado, posteriormente el programa muestra la imagen filtrada, el tiempo que tardó el proceso y la cantidad de pixeles corregidos.



Figura 28. Interfaz gráfica del filtro adaptable tipo Bit-Error.

IV.5. Filtro Sigma de Caja Adaptable

Los detalles de la metodología de este filtro están reportados en [Eliason y McEwn, 1990]. Sin embargo, en la sección II.8.2 de este trabajo de tesis, se presenta una breve descripción de los pasos a seguir para implementar el

filtro Sigma de caja adaptable. Cuando el usuario elige el filtro adaptable tipo Sigma, aparece la ventana mostrada en la figura 29. De manera similar que en el caso anterior, se debe de introducir los parámetros que configuran a este filtro, estos parámetros son: Rango de Sigma (C), mínima cantidad de pixeles dentro de rango de intensidad (K), tamaño del filtro de caja. Cuando el usuario presiona el botón de Aceptar, inicia el proceso de filtrado, posteriormente el programa muestra la imagen filtrada, el tiempo que tardó el proceso de filtrado, la cantidad de pixeles probables a corregir y la cantidad de pixeles corregidos.

FILTRO SIGMA DE CAJA ADAPTABLE

Rango de Sigma (C)	Tiempo de proceso	0	Seg.
<input type="text" value="2"/>	Cantidad de pixeles probables a corregir	0	
Minima cantidad de pixeles dentro de rango de intensidad (K)	Cantidad de pixeles corregidos	0	
<input type="text" value="3"/> (N>K)			
Tamano de la Caja del Filtro			
<input type="text" value="3X3"/>			

Aceptar

Figura 29. Interfaz gráfica del filtro adaptable tipo Sigma.

V. RESULTADOS Y CONCLUSIONES

V.1. Resultados

V.1.1. Aplicación de los Filtros a Imágenes Con Ruido Tipo Sal y Pimienta

La figura 30 muestra la imagen de 4 monedas americanas con ruido tipo sal y pimienta, esta imagen se utilizó como base para probar el efecto de todos los filtros implementados en este trabajo de tesis.

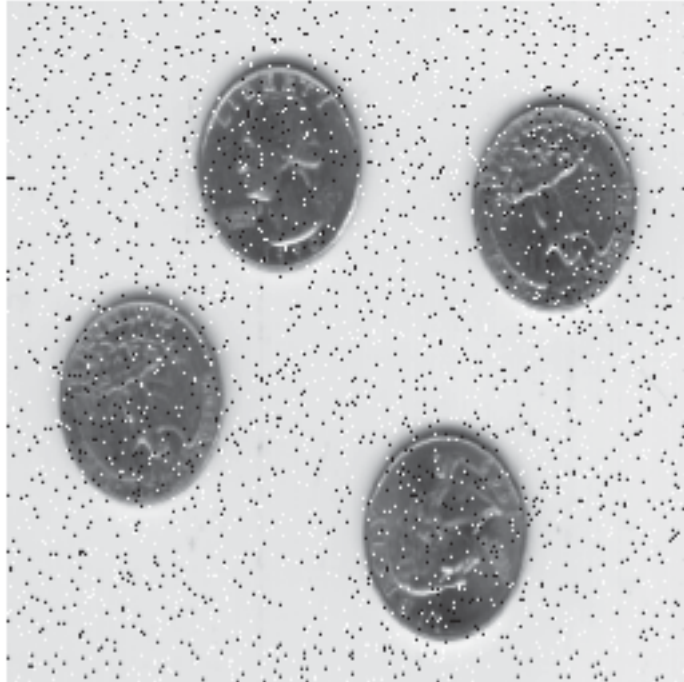


Figura 30. Imagen de monedas con ruido tipo sal y pimienta.

La figura 31 muestra el resultado de aplicar el filtro pasabajas (promedio) con caja 3 x 3 a la figura 30, se puede observar que la aplicación del filtro promedio a imágenes con ruido tipo sal y pimienta, no es muy efectivo, pues se observa que el resultado es una imagen muy borrosa.

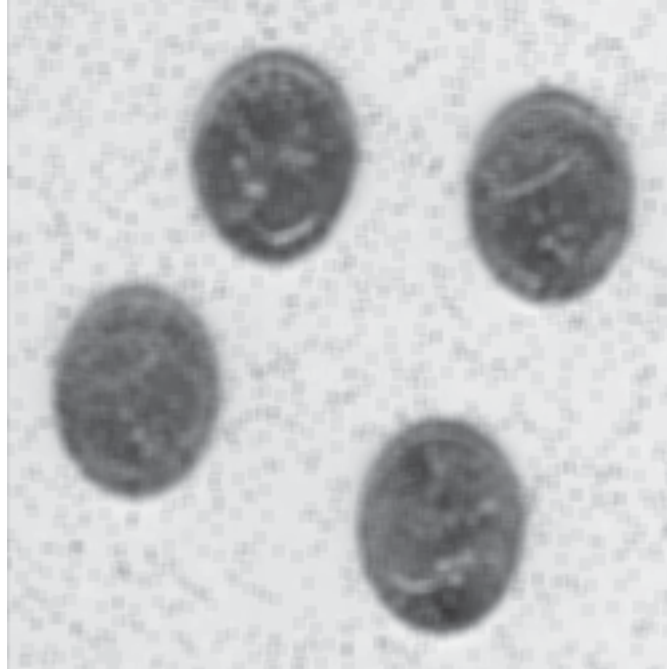


Figura 31. Efecto de aplicar el filtro promedio 3x3 a la imagen de monedas

En la figura 32 se muestra el resultado de aplicar un filtro pasa altas con caja de 3x3 a la imagen de monedas presentada en la figura 30. En este caso se observa que el filtro solamente permite resaltar las altas frecuencias de la imagen, en ese caso como la imagen contiene ruido tipo sal y pimienta, tampoco es recomendable aplicar este tipo de filtro a imágenes con este ruido.



Figura 32. Efecto de aplicar el filtro pasa-altas 3x3.

La figura 33 muestra el resultado de aplicar el filtro tipo cuadrático a la imagen de la figura 30, se observa que solamente recuperó el ruido de la imagen, por tal motivo no se recomienda aplicar este tipo de filtro a imágenes con ruido tipo sal y pimienta.

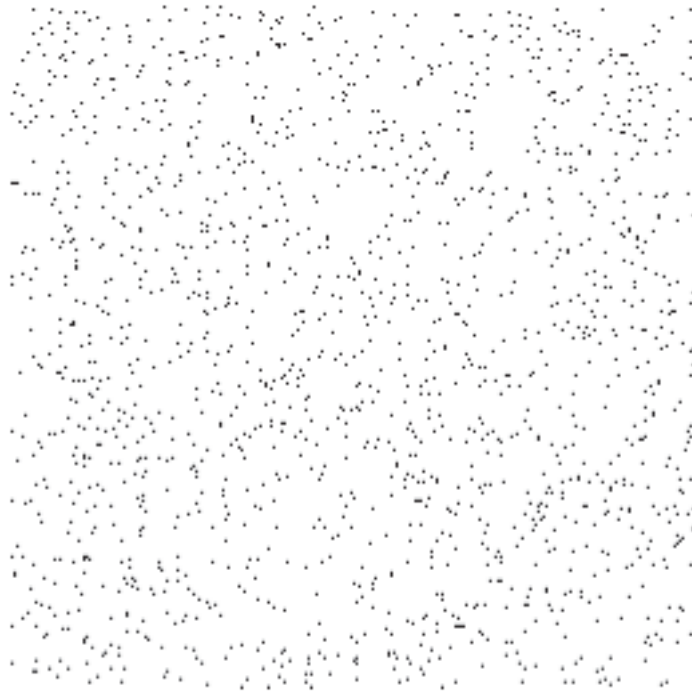


Figura 33. Efecto de aplicar el filtro cuadrático 3x3.

La figura 34 presenta el resultado de aplicar el filtro tipo sigma con umbral global, con las siguientes características: Rango de Sigma (C) = 2, mínima cantidad de pixeles dentro del rango de intensidad (K) = 3, tamaño de la caja del filtro es 3x3. En este caso se observa que se logró disminuir el ruido tipo sal y pimienta, sin embargo, se observan algunas manchas dentro de la imagen.

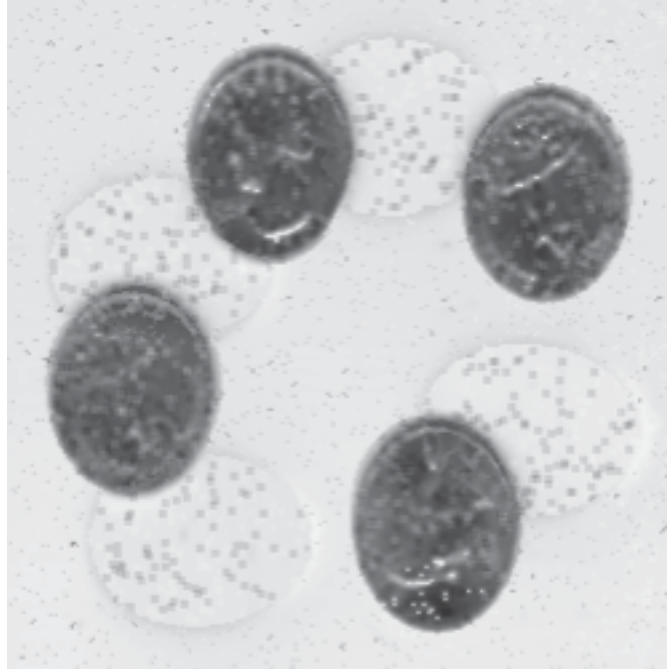


Figura 34. Efecto de aplicar el filtro sigma global 3x3.

En la figura 35, se ilustran el resultado de aplicar un filtro de caja adaptable de tipo Bit-Error, con las siguientes características: Rango de Sigma (C) = 1, valor mínimo de los pixeles 10, valor máximo de los pixeles 254, tamaño elegido para la caja del filtro es 3x3. Se observa que el resultado de aplicar este filtro es más favorable con respecto a los casos anteriores, el ruido residual en la imagen procesada es mínimo, se observan algunos pixeles de color blanco, los cuales son no deseados, por lo tanto, para imágenes con ruido tipo sal y pimienta, se recomienda el uso de los filtros de tipo Bit-Error.

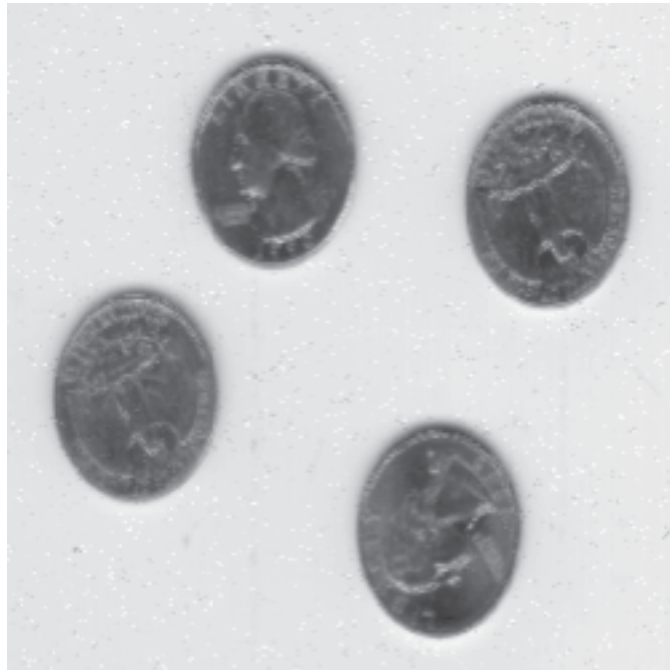


Figura 35. Efecto de aplicar el filtro adaptable Bit-Error 3x3.

La figura 36, muestra el resultado de aplicar el filtro tipo Bit – Error de caja adaptable de tamaño 5 x 5, en este caso también se observa un resultado favorable en la imagen resultante.



Figura 36. Efecto de aplicar el filtro adaptable Bit-Error 5x5.

La figura 37, muestra el resultado de aplicar el filtro tipo Bit-Error de caja adaptable de tamaño 7x7, en este caso también se observa un resultado favorable en la imagen resultante.

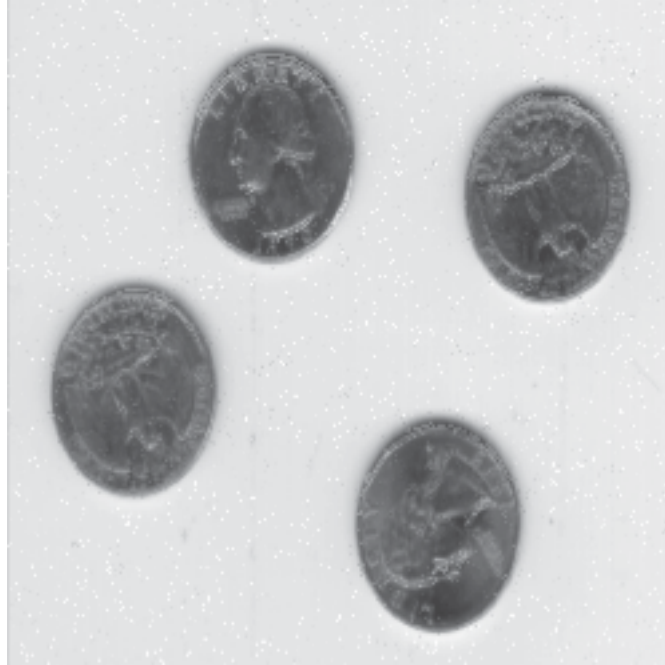


Figura 37. Efecto de aplicar el filtro adaptable Bit-Error 7x7.

La figura 38, muestra el resultado de aplicar un filtro tipo Sigma con caja adaptable y las siguientes características: Rango de Sigma (C) = 2, mínima cantidad de pixeles dentro del rango de intensidad (K) = 3, tamaño de caja 3x3. En este caso, el resultado de aplicar el filtro tipo sigma con caja adaptable no es favorable, por lo tanto tampoco se recomienda aplicar este tipo de filtro a imágenes con ruido tipo sal y pimienta.

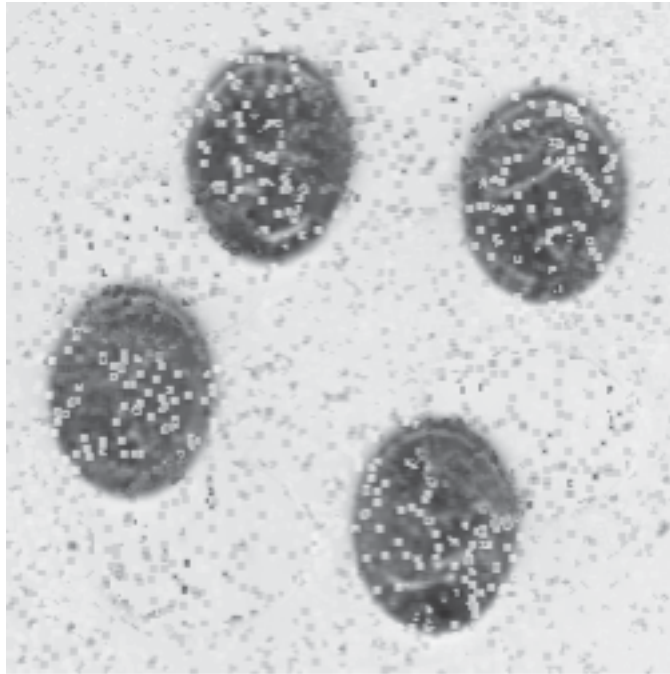


Figura 38. Efecto de aplicar el filtro sigma local 3x3.

V.1.2. Aplicación de filtros a imágenes con ruido tipo Gaussiano

La figura 39 muestra la imagen de una casa con ruido tipo Gaussiano, esta imagen se utilizó como base para probar el efecto de todos los filtros implementados en este trabajo de tesis.



Figura 39. Imagen de una casa con ruido tipo Gaussiano.

La figura 40 muestra el resultado de aplicar el filtro pasa-bajas o promedio con una caja tamaño 5x5 a la imagen de una casa con ruido tipo Gaussiano, se puede observar que el ruido Gaussiano se atenuó considerablemente, sin embargo nuevamente el resultado es una imagen borrosa.



Figura 40. Resultado de aplicar el filtro tipo promedio con caja de 5x5.

La figura 41 muestra el resultado de aplicar el filtro tipo Bit-Error con una caja de 5x5 a la imagen de una casa con ruido tipo Gaussiano, se observa que el ruido se atenuó, sin embargo sigue presente pero con una menor intensidad.



Figura 41. Resultado de aplicar el filtro tipo Bit-Error con caja de 5x5.

La figura 42, muestra el resultado de aplicar un filtro tipo sigma con caja adaptable de 5x5 con las siguientes características: Rango de Sigma (C) = 2, mínima cantidad de pixeles dentro del rango de intensidad (K) = 3, tamaño de caja 3x3. En este caso se observa que el ruido se logró atenuar, sin embargo se observan algunas manchas en la imagen.



Figura 42. Resultado de aplicar el filtro tipo sigma de caja adaptable 5x5.

V.1.3. Aplicación de filtros a imágenes con ruido tipo Poisson

La figura 43 muestra la imagen de Lena, la cual es la imagen más referenciada en el mundo en trabajos del área de procesamiento digital de imágenes, esta imagen de Lena posee ruido tipo Poisson, dicha imagen también se está utilizando para probar el efecto de algunos filtros implementados en este trabajo de tesis.



Figura 43. Imagen de Lena con ruido tipo Poisson.

La figura 44 muestra el resultado de aplicar el filtro pasa-bajas o promedio con una caja tamaño 3x3 a la imagen de Lena con ruido tipo Poisson, se puede observar que el ruido Poisson se atenuó considerablemente, sin embargo nuevamente el resultado es una imagen borrosa.



Figura 44. Resultado de la aplicación del filtro promedio con caja de 3x3 a la imagen de Lena con ruido de Poisson.

La figura 45 muestra el resultado de aplicar el filtro tipo Bit-Error con una caja de 3x3 a la imagen de Lena con ruido tipo Poisson, se observa que el ruido se atenuó, sin embargo sigue presente con una menor intensidad, pero con una mejora considerable en el contraste de la imagen.



Figura 45. Resultado de aplicar el filtro tipo Bit-Error con caja 3x3 a la imagen de Lena con ruido Poisson.

La figura 46, muestra el resultado de aplicar un filtro tipo sigma con caja adaptable de 3x3 con las siguientes características: Rango de Sigma (C) = 2, mínima cantidad de pixeles dentro del rango de intensidad (K) = 3, tamaño de caja 3x3. En este caso se observa que el ruido se logró atenuar, sin embargo se observan algunas manchas en la imagen y poco ruido de moteo.



Figura 46. Resultado de aplicar el filtro tipo sigma con caja adaptable 3x3 a la imagen de Lena con ruido Poisson.

La figura 47, muestra el resultado de aplicar un filtro tipo sigma con caja adaptable de 5x5 con las siguientes características: Rango de Sigma (C) = 2, mínima cantidad de pixeles dentro del rango de intensidad (K) = 3, tamaño de caja 5x5. En este caso se observa que el ruido se logró atenuar, sin embargo se observan algunas manchas en la imagen con mayor persistencia y poco ruido de moteo. Este es el primer indicio para comprobar que es mejor una caja pequeña que una grande.



Figura 47. Resultado de aplicar el filtro tipo sigma con caja adaptable 5x5 a la imagen de Lena con ruido Poisson.

La figura 48, muestra el resultado de aplicar un filtro tipo sigma con caja adaptable de 7x7 con las siguientes características: Rango de Sigma (C) = 2, mínima cantidad de pixeles dentro del rango de intensidad (K) = 3, tamaño de caja 7x7. En este caso se observa que el ruido se atenuó un poco, sin embargo se observan algunas manchas en la imagen con mayor persistencia y poco ruido de moteo. Además se observa que la imagen resultante está más borrosa y se pierde un poco el efecto de los bordes, en comparación con los resultados de emplear cajas de 5x5 y 3x3. Por lo tanto, es mejor utilizar cajas pequeñas

en los filtros para tener un mejor efecto de filtrado, sin embargo se requiere más carga computacional y por lo tanto tiempo de cómputo.



Figura 48. Resultado de aplicar el filtro tipo sigma con caja adaptable 7x7 a la imagen de Lena con ruido Poisson.

V.2. Conclusiones

En este trabajo de tesis se implementaron filtros digitales de cajas adaptables y no-adaptables, se desarrolló una interfaz gráfica bajo la plataforma de MATLAB, la cual es amigable y fácil de utilizar por los usuarios. Con el software desarrollado, se lograron reducir considerablemente los niveles de ruido, se hicieron pruebas con distintos tipos de imágenes y ruidos, tales como: Aditivo y multiplicativo, estas pruebas fueron de tipo Gaussiano, Poisson y el denominado Sal y Pimienta.

Los filtros implementados con caja No-Adaptable son el filtro promedio ó pasa bajas, pasa altas, cuadrático, raíz cuadrada y sigma global. Con respecto a los filtros con caja adaptable, se implementaron el filtro Bit-Error y sigma. Se puede decir que son mejores los filtros de caja adaptable y de tamaño pequeño, debido a que hacen una estadística local con los pixeles en cada una de las cajas. El filtro Bit-Error, es recomendable emplearse cuando las imágenes poseen ruido tipo multiplicativo, tal como el ruido sal y pimienta. Por otra parte, los filtros tipo sigma son más recomendables cuando las imágenes tienen ruido aditivos, tales como el Gaussiano y Poisson.

En conclusión, cuando los filtros emplean cajas pequeñas muestran mejores resultados, pero el tiempo de procesamiento es mayor, por el contrario las cajas más grandes ejecutan más rápido el procesamiento, pero los resultados no son tan buenos, tal como se mostró en el capítulo de resultados, por lo tanto existe un compromiso entre el tiempo de procesamiento y el resultado de la calidad final.

Finalmente, el software desarrollado, se puede emplear en universidades y centros de investigación que impartan cursos de procesamiento digital de imágenes, como una herramienta de apoyo didáctico.

V.3. Trabajo a futuro

- Considerar el uso de imágenes a color.
- Agregar más tipos de filtros digitales y que optimicen el tiempo de procesamiento.
- Desarrollar el software en una aplicación tipo Applet para que se pueda ejecutar desde una página de internet.
- Realizar combinaciones entre los distintos filtros, de tal forma que se aproveche lo mejor de cada filtro.

REFERENCIAS

Cuevas, E. y Zaldivar, D., 2007. **“Visión por Computadora utilizando MATLAB y el Toolbox de Procesamiento Digital de Imágenes”**.

Duiops, **“Transformada Rápida de Fourier”**, [en línea], 2012: Disponible en la Web: <http://www.duiops.net/hifi/enciclopedia/fft.htm>

Eliason, E. y McEwn A., 1990. **“Adaptive Box Filters for Removal of Random Noise from Digital Images”**. Photogrammetric Engineering and Remote Sensing. Vol. 56, No. 4. pp. 453-458.

Kamen, E. y Heck, B., 2008. **“Fundamentos de Señales y Sistemas usando la WEB y MATLAB”**. Pearson Prentice Hall. 3ra edición. ISBN: 978-970-26-1187-5. México. 672 p.

Lee J., 1983. **“Digital Image Smoothing and the Sigma Filter, Computer Vision, Graphics and Image Processing”**. Vol. 24., No. 2, pp. 255-269.

Moore, H., 2007. **“MATLAB para ingenieros”**. Pearson Prentice Hall. 1ra edición. ISBN: 978-970-26-1082-3. México. 624 p.

Pérez, C. 2002. **“MATLAB y sus aplicaciones en las Ciencias y la Ingeniería”**. Pearson Prentice Hall. 1ra edición. ISBN: 978-84-205-3537-1. España. 632 p.

REFERENCIAS (CONTINUACIÓN)

Proakis, J. y Manolakis, D., 1999. **“Tratamiento Digital de Señales”**. Prentice Hall. 3ra edición. ISBN: 84-8322-000-8. España. 1048 p.

Ramírez, A., 2004. Tesis de licenciatura con el nombre **“Diseño de Filtros Digitales Aplicados a Antenas Inteligentes”**. Universidad Autónoma de Baja California. 101 p.

Wikipedia, **“Procesamiento Digital de Imágenes”**, [en línea], 2012:
Disponibile en la Web:
http://es.wikipedia.org/wiki/Procesamiento_digital_de_im%C3%A1genes