

Análisis e Implementación de Algoritmos para Distorsionar Imágenes con Distintos Tipos de Ruido y Aplicación de Filtros en Dos Dimensiones para Restaurarlas

García Chilán Carlos César¹, Viteri Paredes Juan Salomón², Chávez Patricia³
Facultad de Ingeniería Eléctrica y Computación (FIEC)
Escuela Superior Politécnica del Litoral (ESPOL)
Campus Gustavo Galindo, Km 30.5 Vía Perimetral
Apartado 09-01-5863 Guayaquil, Ecuador
ccgarcia@fiec.espol.edu.ec¹, jviteri@fiec.espol.edu.ec², pchavez@espol.edu.ec³

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 más conveniente para la recuperación de las imágenes y se procederá a comparar ambos análisis.

Palabras Claves: *Distorsión de Imágenes, Filtrado de Imágenes.*

Abstract

This work analyses some algorithms for image degradation (noise aggregation) and image enhancement using predefined 2D filters. The filtering is a very useful tool in real situation. We will do an qualitative analysis using the Mean Square Error (MSE) that will allow us to determine which filter is more efficient under certain conditions; then we will do an quantitative analysis using human perception of randomly selected people that will determine which filter give the best result in image enhancement and distortion recovery. Finally we will compare the quantitative and the qualitative results.

Key words: *Images Distortion, Images Filtering.*

1. Introducción

En este trabajo presentaremos algunos 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 captura 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.

2. Tipos de Ruido

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:

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

De la ecuación (1), μ es la media σ es la desviación típica de la variable aleatoria

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.

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

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.

2.3. Ruido Salt & Pepper

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

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

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.

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.

1. Introducción

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 Average (o de la media móvil) es el filtro más común en el procesamiento digital de señales, en la mayoría de las veces es el filtro digital más fácil de entender y usar.

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:

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

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

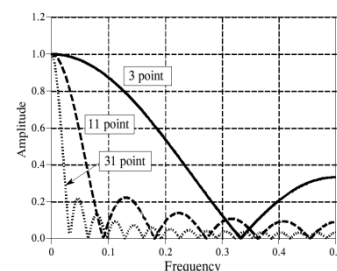


Figura 1. Variación del filtro Average según la media M

3.2. Filtro Disk

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.

¿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.

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.

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

$$g(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}} \quad (4)$$

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 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*, debido a que opera eliminando de la imagen original una parte de su correspondiente imagen paso bajo.

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 gráfico 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), 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 ABRIR, tal como indica la figura:

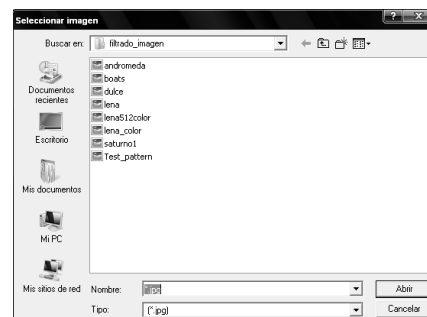


Figura 2. Cuadro de dialogo para abrir archivos

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

4.2. Funcionamiento del Programa

El programa está diseñado con una interfaz gráfica, en la cual el usuario puede abrir una imagen ubicada en cualquier directorio.

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

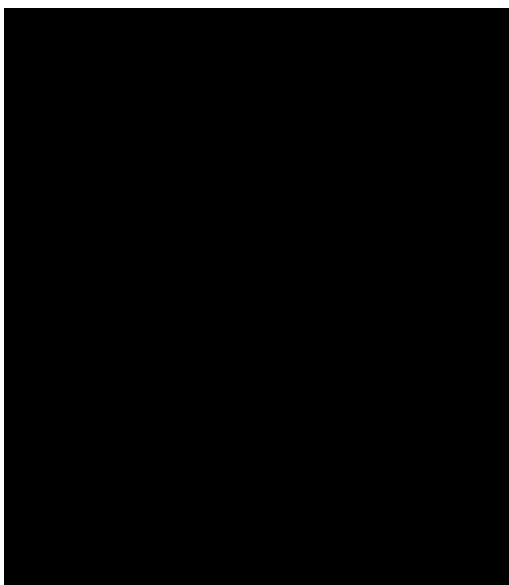
Realizaremos el análisis teórico utilizando el programa en seis imágenes (seis a color y sus respectivas en blanco y negro), pero por cuestiones de espacio, solo presentaremos el análisis de dos imágenes (una a color y otra a blanco y negro) de la imagen de Lena proporcionada por la Ing. Patricia Chávez.

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.

Tabla 1. Resultados analíticos para las imágenes a color y a blanco y negro



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



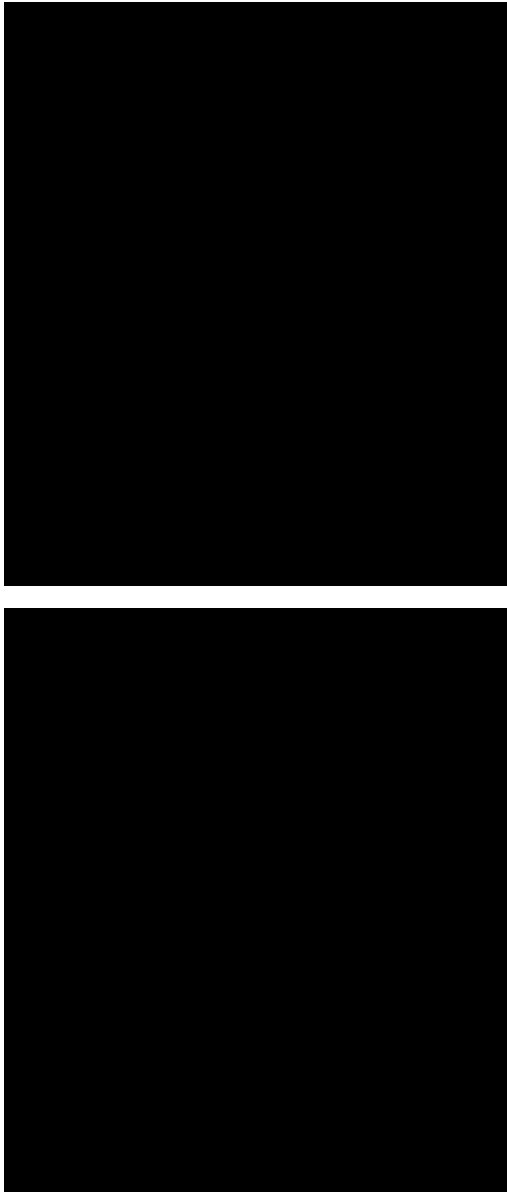
Figura 3. Imagen Original



(a)

(b)

Figura 4. Ruido Poisson aplicada en la imagen (a), y recuperada con el filtro Gaussiano en la imagen (b)

Tabla 2. Resultado visual de 50 personas encuestadas

5. 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 más, 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.

6. 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 varía 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 más 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.

7. Agradecimientos

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

8. 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, 1999.
- [4] Wikipedia, Filtro Gaussiano, http://en.wikipedia.org/wiki/Gaussian_filter, 2009
- [5] Homepages, Filtro Unsharp, <http://homepages.inf.ed.ac.uk/rbf/HIPR2/unsharp.htm>, 2009
- [6] Wikipedia, Distorsiones Lineales, www.wikipedia.com, 2009
- [7] Wikipedia, Distribución Poisson, http://en.wikipedia.org/wiki/Poisson_distribution, 2009
- [8]]Isprs, Ruido Speckle http://www.isprs.org/commission1/ankara06/makaleler/Rajabi_Speckle_Noise.pdf, 2009
- [9] Universidad de Murcia, Filtro Gaussiano, <http://www.um.es/geograf/sigmur/teledet/tema06.pdf>, 2009
- [10] Lena, baboon, airplane, dark_stream, fruits, peppers: Facilitada por Ing. Patricia Chávez.