



Un Acercamiento a la Deconvolución Ciega
Usando el Algoritmo de Lucy-Richardson

Tesis que presenta;
Eymard Hernández López

Para la obtención del grado académico de
Maestro en Ciencias Matemáticas
(Matemáticas Aplicadas e Industriales)

Marzo 2010

Asesor de Tesis;
Dr. Mario Gerardo Medina Valdez

Iztapalapa,, México D.F.

Reconocimientos

Mi más profundo reconocimiento a la universidad Autónoma Metropolitana en Iztapalapa por ser una institución ocupada en la formación de profesionales, por ser la casa de estudios que me ha abierto las puertas al mundo laboral.

Mis más sinceros reconocimientos a CONACYT por su programa de becas de posgrado nacional, gracias por apoyar a los estudiantes en su formación, siendo que estos serán los científicos y gobernantes de este país.

Gracias a mis profesores por su entrega en cada una de sus cátedras, Dr. Héctor Juárez, Dr. Nicolás Carrizosa, Dr. Andrey Novikov, Dr. Joaquín Delgado, Dr. Miguel Ángel Gutiérrez, Dr. John Goddard y Dr. Mario Medina (quien es mi profesor, asesor, una persona profesional, abierta al pensamiento y humana).

Un reconocimiento especial a la M. en C. Iseo por su paciencia y disposición para realizar los trámites administrativos.

A los sinodales Dr. Joaquín Delgado, M. en I.B. Oscar Yañez Suárez y al Dr. Jacobo Oliveros Oliveros, por su valiosa contribución para enriquecer este trabajo de tesis.

Índice

Resumen	i
Introducción	iii
1 Preliminares.	1
1.1 Transformada Discreta de Fourier.	7
1.1.1 Propiedades de la Transformada Discreta de Fourier.	11
1.2 Transformada Rápida de Fourier.	19
1.2.1 Algoritmos FFT.	20
1.2.2 Factorización de una matriz rápida de Fourier.	22
1.2.3 Notación exponencial.	24
1.2.4 Transformada discreta de Fourier en Matlab.	26
1.3 Representación de imágenes.	27
1.4 Problemas de Deconvolución.	28
1.4.1 Integral de Fredholm	30
1.4.2 Alisamiento e Inversión.	30
1.4.3 Regla de Discretización	31
1.5 Regularización.	36
1.5.1 Descomposición en Valores Singulares.	36
1.5.2 Análisis de Descomposición en Valores Singulares (DVS).	37
1.5.3 Método de Regularización Directa.	38
1.5.4 Regularización de Tikhonov.	39
2 Nociones de Probabilidad.	43
2.1 Probabilidad y Teorema de Bayes.	43
2.1.1 Conceptos básicos.	43
2.2 Correlación y Espectro de Potencia de un Proceso Estocástico.	51
2.2.1 Covarianza.	51
2.2.2 Correlación.	51
2.2.3 Independencia.	52
2.2.4 Matriz de Autocorrelación.	52
2.2.5 Espectro de Potencia.	53
2.3 Estimación de parámetros.	55
2.3.1 Estimación por Mínimos Cuadrados.	57

2.3.2	Estimador de Máxima Verosimilitud.	59
2.3.3	El Algoritmo EM (Maximización de la Esperanza).	60
3	Deconvolución de Imágenes.	67
3.1	Formulación del Problema.	67
3.2	Filtrado Inverso.	69
3.3	Filtro de Wiener.	71
3.3.1	Comentario; Regularización Determinista.	75
3.4	Algoritmo de Lucy-Richarson.	76
4	Implementación.	83
4.1	Código en Matlab.	83
4.1.1	Filtrado Inverso en Matlab.	83
4.1.2	Filtro de Wiener en Matlab.	85
4.1.3	Algoritmo de Lucy Richardson en Matlab.	86
4.1.4	GUI-Comparación.	87
4.2	Miscelánea.	112
5	Conclusiones.	123
A	Introducción a las imágenes digitales.	127
A.0.1	El proceso de digitalización.	127
A.0.2	Modelos en color e histogramas.	130
A.1	Representación de Imágenes digitales.	134
A.1.1	Formatos de imágenes digitales.	135
A.2	Filtros digitales.	135
A.2.1	Filtros digitales (en el dominio espacial).	136
A.2.2	Filtros espaciales de realce.	142
A.2.3	Filtros en dominio de frecuencia.	146
B	Mal condicionamiento	151
B.1	Norma inducida.	152
C	Discrepancia	155
C.1	Definición por el residual.	155
C.2	Escogiendo δ : Principio de discrepancia de Morozov.	158
C.3	Funcional de discrepancia.	158
D	Producto de Kronecker, Matrices Toeplitz y Circulantes .	161
D.1	Matrices Toeplitz.	161
D.2	Producto de Kronecker y Matrices circulantes.	162

Resumen

Se presenta un acercamiento a la deconvolución ciega, utilizando el algoritmo de *Lucy-Richardson EM*. Para este propósito se usa una herramienta estadística llamada Algoritmo EM *Maximización de la Esperanza*. El algoritmo de *Lucy-Richardson EM* se compara con las técnicas de restauración de imágenes clásicas, como lo es el *filtro inverso* y *filtro Wiener*. Estas técnicas son ampliamente utilizadas para restaurar imágenes, su desarrollo está tanto en el *dominio espacial* como en el *dominio frecuencia*. El problema de restauración de imágenes es abordado desde un caso más general, que es el campo de los *problemas inversos*, estos problemas están presentes en un amplio ámbito científico.

Si una imagen presenta borrado, el algoritmo de Lucy-Richardson puede reconstruir la imagen con una calidad cercana a la de las técnicas de *filtro inverso* y *filtro de Wiener*. Si una imagen tiene además ruido, el *filtro inverso* no es eficiente, comparado con la calidad del *filtro de Wiener* y *Lucy-Richardson EM*. Al final de este trabajo se compararán los resultados obtenidos con el *filtro inverso*, *filtro de Wiener* y *Lucy-Richardson EM*.

Proyecto de Investigación

**Un acercamiento a la deconvolución ciega
usando el algoritmo de Lucy-Richardson**
Universidad Autónoma Metropolitana-Iztapalapa
Asesor: **Dr. Mario Medina**
Por: Eymard Hernández López

2010

Introducción

Las imágenes son indispensables en las ciencias y en la vida diaria. Reflejan las capacidades de nuestro propio sistema visual, el sistema visual humano. Para nosotros es natural ver paisajes, personas, imágenes en la vida real. Imágenes que alcanzan muchas áreas, desde la fotografía en la astronomía, pasando por otras áreas como son imágenes médicas, microscopía. En cada caso estas imágenes son la base de lo que queremos observar, por ello nos interesa ver la representación que sea más próxima a la realidad de las escenas en cuestión, a la que llamaremos imagen real ó verdadera.

El proceso de observación no es perfecto. Esto es cierto a medida que tengamos fenómenos como borrado, ruido u otra degradación en el proceso de obtención y almacenamiento de las imágenes. En restauración digital de imágenes se pretende recobrar un estimador de la imagen original a partir de observaciones degradadas. La clave para ello esta en resolver un problema inverso donde es apropiado incorporar información a priori de la imagen a recuperar en el proceso de restauración.

Como se ha dicho en la restauración de imágenes clásica se busca una estimación de la imagen verdadera, donde se supone que la degradación es conocida. En cambio, la restauración ciega en imágenes, aborda el problema mucho más difícil, pero más realista; donde es desconocida la degradación. La degradación es generalmente no lineal; sin embargo, para la mayor parte de este trabajo, se supone que la imagen observada es la salida de un sistema lineal espacialmente invariante (ELI) al cual se le agrega ruido. Por lo tanto se convierte en un problema de Deconvolución Ciega, con la degradación desconocida representada como una función llamada Función de Dispersión del Punto (PSF, por sus siglas en ingles).

La restauración clásica ha madurado desde su inicio, desde la exploración espacial en los años 60 del siglo pasado, y las técnicas numerosas que se pueden encontrar en la literatura. Los primeros algoritmos que abordan el problema de deconvolución ciega aparecieron a mediados de los años setenta, intentado identificar patrones conocidos de la degradación; un esfuerzo pequeño pero dedicado a finales de los 80 ver [13], y un resurgimiento en los años 90. Desde entonces,

el área ha sido estudiada extensivamente principalmente por ingenieros y las comunidades del tratamiento de señales astronómicas, y en óptica. Muchos de los algoritmos de deconvolución ciega tienen sus orígenes en la teoría de álgebra lineal y en análisis numérico.

Una pregunta importante que aparece naturalmente es ¿porqué la deconvolución ciega es útil?, ¿Por qué no usar simplemente un procedimiento de captura más sofisticado? Quizás, pero siempre existen los límites físicos, tales como ruido, difracción, o un canal de la observación fuera de nuestro control, por tanto las imágenes se capturan a menudo en condiciones no óptimas. También hay imágenes existentes de los acontecimientos únicos que no se pueden volver a tomar, de los cuales quisiéramos recuperar (por ejemplo en medicina); además en estos casos es a menudo imposible medir las características del sistema de la proyección de la imagen directamente. Otra razón es el costo. El equipo de alta calidad de la óptica y de detección es costoso. Sin embargo, la capacidad de cálculo de hoy es creciente y abre la puerta en el uso de modelos cada vez más sofisticados.

Así, la deconvolución ciega representa una herramienta valiosa que se puede utilizar para mejorar la calidad de la imagen, sin necesidad de calibraciones complicadas del sistema en tiempo real en el proceso de detección de la imagen (es decir, en proyección de imagen médica, la video-comunicación, exploración espacial, proyección de imagen de radiografía, etc). El problema de deconvolución ciega se encuentra en muchas áreas, tales como; proyección de imágenes astronómicas, teledetección, microscopía, proyección de imágenes médicas, óptica, fotografía, usos de súper-resolución, y usos de seguimiento de movimiento, entre otros. Por ejemplo, la proyección de imagen astronómica es uno de los usos primarios de los algoritmos de deconvolución ciega.

Los sistemas terrestres de la obtención de imágenes astronómicas están propensos continuamente a degradación debido al índice de refracción cambiante de la atmósfera. Las observaciones extraterrestres de la tierra y de los planetas son degradadas por la falta de definición debido al movimiento de la tierra, ó como resultado de velocidades lentas del obturador de la cámara debido al movimiento rápido, por ejemplo, de la nave espacial. Se utiliza la deconvolución ciega para mejorar la calidad de la imagen de la película presente en radiografías borrosas (que poseen ruido de tipo poisson). En tales usos, la mayor parte del tiempo las degradaciones son inevitables porque los sistemas de obtención de la imagen médica limitan la intensidad de la radiación, para proteger la salud del paciente.

En óptica, se utiliza deconvolución ciega para restaurar la imagen original de la degradación introducida por un microscopio o cualquier otro instrumento óptico. Por ejemplo las imperfecciones principales del lente del telescopio del trasbordador espacial Hubble han proporcionado una cantidad de imágenes para su tratamiento digital. Los parámetros que describen éstos procesos son ge-

neralmente desconocidos; sin embargo, deconvolución ciega puede permitir la restauración de estas imágenes.

Como ejemplo final, en usos de seguimiento de objetos puede tener borrado debido a su propio movimiento, o el movimiento de la cámara. Consecuentemente, el uso de deconvolución ciega puede mejorar estas imágenes. En el capítulo 3, examinamos el campo de la deconvolución ciega de la imagen. Desarrollamos y presentamos una de las técnicas Bayesianas más importantes, el algoritmo de Lucy-Richardson EM. Puesto que el problema de deconvolución ciega es un problema inverso, numéricamente difícil de resolver, este es un campo de estudio muy actual y fértil.

Objetivos

- Aplicar la teoría de problemas inversos, para comprender la Deconvolución y posteriormente la Deconvolución Ciega.
- Entender los conceptos e ideas clave de la Deconvolución.
- Realizar recuperación de Imágenes primeramente usando Deconvolución en Matlab.
- Implementar pseudocódigo en Matlab y realizar un GUI de comparación del algoritmo de Lucy-Richardson con otros métodos.

Contenido

1	Preliminares.	9
1.1	Transformada Discreta de Fourier.	15
1.1.1	Propiedades de la Transformada Discreta de Fourier.	18
1.2	Transformada Rápida de Fourier.	25
1.2.1	Algoritmos FFT.	27
1.2.2	Factorización de una matriz rápida de Fourier.	28
1.2.3	Notacion exponencial.	31
1.2.4	Transformada discreta de Fourier en Matlab.	33
1.3	Representacion de imágenes.	34
1.4	Problemas de Deconvolución.	35
1.4.1	Integral de Fredholm	37
1.4.2	Alisamiento e Inversión.	37
1.4.3	Regla de Discretización	38
1.5	Regularización.	42
1.5.1	Descomposición en valores singulares.	42
1.5.2	Análisis de DVS.	44
1.5.3	Método de Regularización Directa.	44
1.5.4	Regularización de Tikhonov.	46
2	Nociones de Probabilidad.	51
2.1	Probabilidad y Teorema de Bayes.	51
2.1.1	Conceptos básicos.	51
2.2	Correlación y Espectro de Potencia de un Proceso Estocástico.	59
2.2.1	Correlación.	59
2.2.2	Covarianza.	60
2.2.3	Independencia.	60
2.2.4	Matriz de Autocorrelación.	61
2.2.5	Espectro de Potencia.	61
2.3	Estimación de parámetros.	63
2.3.1	Estimación por Mínimos Cuadrados.	65
2.3.2	Estimador de Máxima Verosimilitud.	67
2.3.3	El Algoritmo EM (Esperanza-Maximización).	69

3	Deconvolución de Imágenes.	75
3.1	Formulación del Problema.	75
3.2	Filtrado Inverso.	77
3.3	Filtro de Wiener.	80
3.3.1	Comentario; Regularización Determinista.	84
3.4	Algoritmo de Lucy-Richarson.	85
4	Implementación.	93
4.1	Código en Matlab.	93
4.1.1	Filtrado Inverso en Matlab.	93
4.1.2	Filtro de Wiener en Matlab.	95
4.1.3	Algoritmo de Lucy Richardson en Matlab.	96
4.1.4	GUI-Comparación.	98
4.2	Miscelánea.	98
5	Conclusiones.	113
A	Introducción a las imágenes digitales.	117
A.0.1	El proceso de digitalización.	117
A.0.2	Modelos en color e histogramas.	120
A.1	Representación de Imágenes digitales.	124
A.1.1	Formatos de imágenes digitales.	125
A.2	Filtros digitales.	125
A.2.1	Filtros digitales (en el dominio espacial).	126
A.2.2	Filtros espaciales de realce.	132
A.2.3	Filtros en dominio de frecuencia.	136
B	Mal condicionamiento	141
B.1	Norma inducida.	142
C	Discrepancia	145
C.1	Distribución Uniforme Módulo 1.	145
C.2	Discrepancia	146
D	Producto de Kronecker y Matrices Circulantes.	149

Capítulo 1

Preliminares.

El campo de la restauración de imágenes tiene una larga historia, comenzó en los años 50^s del siglo XX con el programa espacial de la NASA. Las primeras imágenes de la Tierra, Luna (por el lado opuesto de la tierra), y Marte tenían para ese tiempo una resolución inimaginable. Estas imágenes eran obtenidas con grandes dificultades técnicas, como vibración al momento de tomar las imágenes, movimientos bruscos, mal enfocamiento, etc. Estas dificultades resultan en cierta degradación de la imagen obtenida, que desde el punto de vista científico y económico puede ser devastador. La necesidad de recuperar la información perdida en las imágenes degradadas, era el objetivo principal. Inicialmente, se adaptaron algoritmos de procesamiento de señales en una dimensión para aplicarlos a las imágenes, creando un nuevo campo que hasta el día de hoy es ampliamente estudiado. Las técnicas de reconstrucción de imágenes tienen aplicación en muchas áreas como tomografía médica, resonancia magnética, criminología, por mencionar algunos.

Las técnicas de restauración de imágenes tienen una mejora muy notable desde sus inicios, el objetivo de este trabajo es presentar una técnica muy usada en restauración de imágenes, llamada *Algoritmo de Lucy-Richardson* ver [6] y [16]. Para una mejor comprensión del problema de restauración de imágenes, es necesario reconocer este como un caso particular de un campo más general que es el campo de los problemas inversos, los cuales surgen en muchas ciencias.

El problema de restauración de imágenes puede ser expresado en la forma de una ecuación integral de Fredholm del primer tipo (que veremos más adelante en este capítulo):

$$g(x) = \int H(x, y)f(y)dy,$$

donde $f(y)$ es la función de interés (imagen a recuperar), $g(x)$ es la función accesible (imagen observada), y $H(x, y)$ es el kernel de la ecuación integral (función de dispersión del punto). La ecuación anterior es la ecuación de la

imagen del problema de restauración. Para abordar el problema de recuperar $f(y)$ primero revisaremos algunos conceptos necesarios para el adecuado estudio de esta técnica de recuperación de imágenes (deconvolución). Empecemos con la Transformada de Fourier, que tiene muchas aplicaciones en la ingeniería.

Definición 1.0.1. Sea $f : \mathbb{R}^n \rightarrow \mathbb{R}$, la **Transformada de Fourier en \mathbb{R}^n** de f se define como

$$\mathcal{F}(w) = \int_{\mathbb{R}^n} f(x) e^{-iw^T x} dx \quad (1.1)$$

donde $w^T x = \sum_{i=1}^n w_i x_i$, con $w^T = (w_1, \dots, w_n)$ y $x = (x_1, \dots, x_n) \in \mathbb{R}^n$

La transformada de Fourier $\mathcal{F}(w)$ es una función que toma valores en los números complejos. El dominio de esta transformada le llamaremos el *dominio de frecuencia* (w en Hertz), al dominio de la función f se le conoce como *dominio temporal* (cuando la variable $x = t$ representa el tiempo) ó *dominio espacial* en procesamiento de imágenes. La transformada de Fourier para con $n = 1$ tiene la forma

$$\mathcal{F}(w) = \int_{\mathbb{R}} f(x) e^{-iwx} dx \quad (1.2)$$

donde los vectores w y x tienen solo una componente, esto es $w = w_1$ y $x = x_1$. Cuando se trabaja en procesamiento de señales, el dominio es temporal, por lo que $n = 1$. Cuando trabajemos en procesamiento de imágenes, el dominio le llamaremos dominio espacial, donde $n = 2$ y cada punto de este espacio será llamado píxel. El análogo de trabajar con $n = 3$ es procesamiento de imágenes tridimensionales, en dichas imágenes los puntos reciben el nombre de voxels. Esto último no lo abordaremos, solamente es para ilustrar la transformada de Fourier cuando $n = 3$. Durante este capítulo el desarrollo es unidimensional, la extensión al caso bidimensional es inmediata.

Para enunciar la siguiente proposición, el lector debe recordar el concepto de continuidad de una función: f es continua en x_0 si f está definida en algún intervalo abierto que contenga a x_0 , y si para cualquier $\varepsilon > 0$ existe una $\delta > 0$ tal que

$$\text{si } |x - x_0| < \delta \text{ entonces } |f(x) - f(x_0)| < \varepsilon.$$

El lector debe estar familiarizado con el concepto de continuidad de una función por partes: Decimos que una función es continua por partes en el intervalo $[-L, L]$ si cumple con los siguientes puntos

- i Teniendo $f(x)$ definida y continua en $[-L, L]$, excepto quizás en un número finito de puntos.
- ii Los límites laterales $\lim_{x \rightarrow +L} f(x)$ y $\lim_{x \rightarrow -L} f(x)$ existen y son finitos
- iii En cada punto $c \in (-L, L)$, donde $f(x)$ no es continua, $\lim_{x \rightarrow c} f(x)$ existe y es finito.

Decimos que una función $f(x)$ es suave por partes en el intervalo $[-L, L]$ si $f(x)$ y $f'(x)$ son funciones continuas por partes en $[-L, L]$.

Proposición 1.0.2. (Transformada de Fourier.)

Sea $f(x)$ una función continua, suave y absolutamente integrable en $(-\infty, \infty)$ (i.e. $\int_{-\infty}^{\infty} |f(x)| dx < \infty$) entonces

$$f(x) = \frac{1}{2\pi} \int_{-\infty}^{\infty} e^{iwx} \mathcal{F}(w) dw \quad (1.3)$$

donde $\mathcal{F}(w)$ esta dada por

$$\mathcal{F}(w) = \int_{-\infty}^{\infty} e^{-iwx} f(x) dx \quad (1.4)$$

Esta propocisión no es rigurosa, pues no hemos justificado muchos pasos, incluyendo la convergencia de la integral impropia. Una prueba rigurasa puede encontrarse en [17] pag 94. La hipótesis $\int_{-\infty}^{\infty} |f(t)| dt < \infty$ da una condición suficiente para que exista la transformada de Fourier;

$$\begin{aligned} |\mathcal{F}(w)| &\leq \int_{-\infty}^{\infty} |e^{-iwx} f(x)| dx \\ &= \int_{-\infty}^{\infty} |f(x)| dx \text{ puesto que } |e^{-iwx}| = 1 \\ &< \infty. \end{aligned}$$

Cabe mencionar que algunos autores anteponen el termino $1/\sqrt{2\pi}$, a la integral, así la inversa de la transformada de Fourier tendrá el mismo coeficiente, esto es, definen la transformada de Fourier de f como

$$\mathcal{F}(w) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} f(x) e^{-iwx} dx \quad (1.5)$$

De esta forma la transformada inversa de Fourier está dada por

$$f(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} \mathcal{F}(w) e^{iwx} dw. \quad (1.6)$$

La notación es algo que cambia en los distintos textos, por ejemplo, en lugar de usar el simbolo \mathcal{F} algunos textos usan $\mathcal{F}(f(x))$, otros usan $\tilde{f}(w)$ para representar a la tranformada de Fourier, pero en este texto se usara la primera notación. Veamos un ejemplo de transformada de fourier.

Ejemplo 1.0.3.

En nuestro primer ejemplo, queremos calcular la transformada de Fourier de la función

$$f(x) = \begin{cases} 1 & \text{si } x \in [-\pi, \pi] \\ 0 & \text{si } x \in \mathbb{R} - [-\pi, \pi] \end{cases}$$

Tenemos que $f(x)e^{-iw x} = f(x)(\cos wx - i \sin wx)$. Puesto que $f(x)$ es par y $f(x)\sin(wx)$ es impar y su integral en la recta real es cero. Por tanto la integral de la transformada de Fourier de $f(x)$ se reduce a

$$\begin{aligned} \mathcal{F}(w) &= \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} f(x) \cos(wx) dx \\ &= \frac{1}{\sqrt{2\pi}} \int_{-\pi}^{\pi} \cos(wx) dx \\ &= \frac{\sqrt{(2\pi)} \sin(w\pi)}{w\pi} \end{aligned}$$

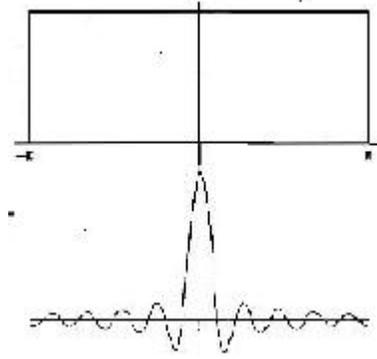


Figura 1.1: Transformada de Fourier.

□

La transformada de Fourier tiene la siguiente propocisión;

Proposición 1.0.4.

1. Linealidad. Si $h(x) = af(x) + bg(x)$, con $x \in \mathbb{R}^n$, $a, b \in \mathbb{R}$, entonces

$$\mathcal{H}(w) = a\mathcal{F}(w) + b\mathcal{G}(w)$$

2. Traslación. Si $h(x) = f(x - x_0)$, con $x, x_0 \in \mathbb{R}^n$, entonces

$$\mathcal{H}(w) = e^{-iw^T x_0} \mathcal{F}(w).$$

3. Modulación. Si $h(x) = e^{ix^T \xi_0} f(x)$, con $x \in \mathbb{R}^n$, $\xi_0 \in \mathbb{R}$, entonces

$$\mathcal{H}(w) = \mathcal{F}(w - \xi_0)$$

4. Escalamiento. Si $h(x) = f(ax)$, con $x \in \mathbb{R}^n$, $a \in \mathbb{R}$, entonces

$$\mathcal{H}(w) = \frac{1}{|a|} \mathcal{F}\left(\frac{w}{a}\right)$$

caso particular $a = -1$, entonces $h(x) = f(-x)$, entonces $\mathcal{H}(w) = \mathcal{F}(-w)$.

5. Conjugación. Si $h(x) = \overline{f(x)}$, con $x \in \mathbb{R}^n$, entonces

$$\mathcal{H}(w) = \mathcal{F}(-w)$$

Para ver la prueba de esta propocisión consultar [7]. Una última y muy importante propiedad es el llamado teorema de convolución. Esta propiedad tiene una gran relevancia en el presente texto. Antes de ver la siguiente definición, necesitamos conocer cuando una función es cuadrado integrable; una función $f(x)$ de una variable real con valores reales o complejos se dice de *cuadrado sumable* o también de *cuadrado integrable* sobre un determinado intervalo, si la integral del cuadrado de su módulo, definida en el intervalo de definición, converge, esto es

$$\int_{-\infty}^{\infty} |f(x)|^2 dx < \infty$$

Definición 1.0.5. Sean h y f dos funciones cuadrado integrables. La *convolución* de h y f , (denotada por $h * f$) se define como

$$(h * f)(t) = \int_{-\infty}^{\infty} h(t-x)f(x)dx. \quad (1.7)$$

En el siguiente teorema se enuncian propiedades de la convolución.

Teorema 1.0.6. Sean f , h y g funciones continuas en el intervalo $[0, +\infty]$, entonces

1. (ley conmutativa)

$$f * h = h * f$$

2. (ley distributiva)

$$f * (g + h) = f * g + f * h$$

3. (ley asociativa)

$$(f * g) * h = f * (g * h)$$

4. $f * 0 = 0 * f = 0$.

Se tiene el siguiente teorema en la transformada de Fourier de una convolución de dos funciones.

Teorema 1.0.7. Sean f y g dos funciones integrables. Entonces

$$\mathcal{F}[f * g] = \sqrt{2\pi}\mathcal{F} \cdot \mathcal{G}, \quad (1.8)$$

y también

$$\mathcal{F}^{-1}[\mathcal{F} \cdot \mathcal{G}] = \frac{1}{\sqrt{2\pi}}f * g.$$

Demostración: Para obtener la ecuación (1.9), usamos las definiciones de Transformada de Fourier y convolución.

$$\begin{aligned}\mathcal{F}[f * g](w) &= \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} [(f * g)(t)] e^{-iwt} dt \\ &= \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} \left[\int_{-\infty}^{\infty} f(t-x)g(x)dx \right] e^{-iwx} dt.\end{aligned}$$

Como escribimos $e^{-iwt} = e^{-iw(t-x)}e^{-iwx}$, cambiando el orden de integración obtenemos

$$\mathcal{F}[f * g](w) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(t-x)e^{-iw(t-x)}g(x)dt e^{-iwx} dx.$$

Con el cambio de variable $s = t - x$, obtenemos

$$\mathcal{F}[f * g](w) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(s)e^{-iw(s)}g(x)ds e^{-iwx} dx.$$

El lado derecho de la ecuación anterior, puede escribirse como

$$\sqrt{2\pi} \left(\frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} f(s)e^{-iws} ds \right) \left(\frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} g(x) e^{-iwx} dx \right),$$

el cual es $\sqrt{2\pi}\mathcal{F} \cdot \mathcal{G}$, lo deseado. Para la inversión se tiene que

$$\begin{aligned}\sqrt{2\pi}\mathcal{F}^{-1}[\mathcal{F} \cdot \mathcal{G}] &= \mathcal{F}^{-1}[\mathcal{F}(f * g)] \\ &= f * g\end{aligned}$$

□

La transformada de Fourier es una herramienta básica en el procesamiento y recuperación de imágenes (ver apéndice A). Para procesar una imagen (analizar, recuperar, detectar bordes o algo de interés, etc) usualmente se utilizan filtros, un filtro es una transformación T que transforma una señal, f , en otra señal \tilde{f} . Esta transformación, si satisface las propiedades de linealidad ($T[f + g] = T[f] + T[g]$, $T[cf] = cT[f]$, c constante), se le conoce como filtro lineal. Una señal es una función $f : \mathbb{R} \rightarrow \mathbb{C}$ que es usualmente continua por partes. con estos filtros se procesa la imagen, devolviendo una imagen (la imagen toma el papel de una señal en el filtro, la señal está en \mathbb{R}^1 y la imagen en \mathbb{R}^2 ó \mathbb{R}^3 en el caso de imágenes tridimensionales) de salida, que en algunos casos modifica la imagen de entrada. Un ejemplo de un filtro es un dispositivo que remueve ruido de una señal, como se verá más adelante. Por ello es necesario tener presente que para el desarrollo de este trabajo, la herramienta básica y muy relevante es la transformada de Fourier en forma discreta.

1.1 Transformada Discreta de Fourier.

La transformada de Fourier y las series de Fourier son técnicas muy usadas para analizar señales continuas. Sin embargo para muchas aplicaciones la señal es un conjunto de datos, muestras de la señal continua, por ejemplo un reproductor de música digital, usa formatos donde las melodías son señales discretizadas, esto es, son muestras que provienen de una señal continua (señal análoga). Asimismo en procesamiento de imágenes, una imagen digitalizada es un conjunto de datos que son muestras tomadas de una imagen real (por medio de un dispositivo de captura de imágenes). Una versión discreta de la transformada de Fourier es necesaria para el análisis de estas señales discretizadas.

Para motivar la idea detrás de la transformada de Fourier, aproximemos los coeficientes de una serie de Fourier ¹ para una función continua $f(t)$. Sea S_N un conjunto de sucesiones N -periódicas de números complejos. Cada elemento $y = \{y_j\}_{-\infty}^{\infty}$ en S_N , puede ser interpretado como una señal discreta donde y_j es el valor de la señal al tiempo $t = t_j$. La sucesión y_j es N -periódica, esto es cumple con la propiedad; $y_{k+N} = y_k$ para cualquier entero k . Usamos la regla trapezoidal para aproximar la integral $(2\pi)^{-1} \int_0^{2\pi} F(t)dt$, con tamaño de paso $h = 2\pi/N$, con N natural, esto es

$$\frac{1}{2\pi} \int_0^{2\pi} F(t)dt \approx \frac{1}{2\pi} \frac{2\pi}{N} \left[\frac{Y_0}{2} + Y_1 + \dots + Y_{N-1} + \frac{Y_N}{2} \right]$$

donde $Y_j := F(hj) = F(2\pi j/N)$, $j = 0, \dots, N-1$. Si $F(t)$ es n -periódica, entonces $Y_0 = Y_N$ y la ecuación anterior toma la forma

$$\frac{1}{2\pi} \int_0^{2\pi} F(t)dt \approx \frac{1}{N} \sum_{j=0}^{N-1} Y_j. \quad (1.9)$$

¹Serie de Fourier: Las series de Fourier tienen la forma:

$$f(t) \approx \frac{a_0}{2} + \sum_{n=1}^{\infty} [a_n \cos(nt) + b_n \sin(nt)]$$

donde

$$\begin{aligned} a_0 &= \frac{1}{L} \int_{-L}^L f(t)dt, \\ a_n &= \frac{1}{L} \int_{-L}^L f(t) \cos\left(\frac{n\pi}{L}t\right) dt \text{ y} \\ b_n &= \frac{1}{L} \int_{-L}^L f(t) \sin\left(\frac{n\pi}{L}t\right) dt. \end{aligned}$$

son los coeficientes de Fourier de la serie de Fourier de la función $f(t)$. La serie de Fourier de f puede expresarse también en forma compleja:

$$f(t) \approx \sum_{n=-\infty}^{\infty} c_n e^{\pi i n t / L} \text{ donde } c_n = \frac{1}{2L} \int_{-L}^L f(t) e^{-\pi i n t / L} dt.$$

Aplicando esta fórmula para el cálculo del k -ésimo coeficiente de Fourier complejo¹, se tiene que

$$c_k = \frac{1}{2\pi} \int_0^{2\pi} f(t)e^{-ikt} dt \approx \frac{1}{N} \sum_{j=0}^{N-1} f(2\pi j/N) e^{-2\pi ijk/N}. \quad (1.10)$$

Por tanto,

$$c_k \approx \frac{1}{N} \sum_{j=0}^{N-1} y_j \bar{w}^{jk},$$

donde $y_j = f(2\pi j/N)$ y $w = e^{2\pi i/N}$, $t = 2\pi j/N$, para $j = 0, \dots, N-1$.

Definición 1.1.1. Sea $y = \{y_j\}_{j=-\infty}^{\infty} \in S_N$. La transformada discreta de Fourier de y , es la sucesión $(\mathcal{F}\{y\})_k = \hat{y}_k$, donde

$$\hat{y}_k = \sum_{j=0}^{N-1} y_j w^{jk} \text{ con } w = e^{-2\pi i/N} \quad (1.11)$$

Es decir,

$$\hat{y}_k = \sum_{j=0}^{N-1} y_j e^{-2\pi ijk/N}.$$

La transformada de Fourier discreta es análoga a la fórmula para el k -ésimo coeficiente de Fourier. Si y_j surge de valores de una señal continua, f , definida en el intervalo $[0, 2\pi]$, entonces el k -ésimo coeficiente de Fourier, es aproximado por

$$c_k \approx \frac{1}{N} \hat{y}_k.$$

La ecuación (1.11) es equivalente a la expresión:

$$\hat{y} = \mathcal{F}_N(y) = \sum_{j=0}^{N-1} y_j W_{kj} \quad (1.12)$$

donde $y = (y_0, \dots, y_{N-1})^T$, $\hat{y} = (\hat{y}_0, \dots, \hat{y}_{N-1})$ y $W_{kj} = e^{-2\pi ijk/N}$. En consecuencia (1.12) puede escribirse matricialmente como

$$\hat{y} = \mathcal{F}_N(y) = Wy, \quad (1.13)$$

donde

$$W = \begin{pmatrix} W_0(0) & \cdot & \cdot & W_{N-1}(0) \\ \cdot & & & \cdot \\ \cdot & & & \cdot \\ W_0(N-1) & \cdot & \cdot & W_{N-1}(N-1) \end{pmatrix}$$

Con el fin de calcular los coeficientes \hat{y}_k , observemos que el término exponencial, $w = e^{-2i\pi/N}$ de la ecuación (1.11), aparece en todos los coeficientes de

la serie. Recordando la noción de clases de congruencias² $kj \equiv l \pmod{N}$, por ejemplo si $k = 1, 2, 3$, $N = 8$ y $j = 0, 1, \dots, N - 1$, entonces el término w^{kj} toma los siguientes valores

$$\begin{aligned} \{e^{-2\pi i(0)}, e^{-2\pi i(1)/8}, \dots, e^{-2\pi i(7)/8}\} &=_{k=1} \{1, w^1, w^2, \dots, w^7\} \\ \{e^{-2\pi i2(0)}, e^{-2\pi i2(1)/8}, \dots, e^{-2\pi i2(7)/8}\} &=_{k=2} \{1, w^2, w^4, \dots, w^6\} \\ \{e^{-2\pi i3(0)}, e^{-2\pi i3(1)/8}, \dots, e^{-2\pi i3(7)/8}\} &=_{k=3} \{1, w^3, w^6, \dots, w^5\}, \end{aligned}$$

cuando $k = 1$, $k = 2$ y $k = 3$ respectivamente. En la siguiente tabla se muestra la congruencia $kj \equiv l \pmod{8}$, para $k, j = 0, 1, \dots, 7$

$$\left[\begin{array}{c|cccccccc} k \backslash j & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 2 & 0 & 2 & 4 & 6 & 0 & 2 & 4 & 6 \\ 3 & 0 & 3 & 6 & 1 & 4 & 7 & 2 & 5 \\ 4 & 0 & 4 & 0 & 4 & 0 & 4 & 0 & 4 \\ 5 & 0 & 5 & 2 & 7 & 4 & 1 & 6 & 3 \\ 6 & 0 & 6 & 4 & 2 & 0 & 6 & 4 & 2 \\ 7 & 0 & 7 & 6 & 5 & 4 & 3 & 2 & 1 \end{array} \right]$$

La tabla anterior, representa a los exponentes de $w = e^{-2i\pi/N}$. Este arreglo representa una matriz simétrica E , dada por

$$E = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & \\ 0 & 2 & 4 & 6 & 0 & 2 & 4 & 6 & \\ 0 & 3 & 6 & 1 & 4 & 7 & 2 & 5 & \\ 0 & 4 & 0 & 4 & 0 & 4 & 0 & 4 & \\ 0 & 5 & 2 & 7 & 4 & 1 & 6 & 3 & \\ 0 & 6 & 4 & 2 & 0 & 6 & 4 & 2 & \\ 0 & 7 & 6 & 5 & 4 & 3 & 2 & 1 & \end{pmatrix},$$

En general para la congruencia $kj \equiv l \pmod{N}$, con los índices $k, j =$

²La aritmética modular es un sistema aritmético para clases de equivalencia de números enteros llamadas clases de congruencia. a y b se encuentran en la misma *clase de congruencia* módulo n , si ambos dejan el mismo residuo si los dividimos por n , o, equivalentemente, si $a - b$ es un múltiplo de n . Esta relación se expresa utilizando la notación; $a \equiv b \pmod{n}$, así se tiene por ejemplo que $63 \equiv 83 \pmod{10}$, ya que ambos, 63 y 83 dejan el mismo residuo (3) al dividir por 10, o, equivalentemente, $63 - 83$ es múltiplo de 10, se lee: 63 ES CONGRUENTE CON 83, MÓDULO 10. En este ejemplo $n = 10$.

$0, 1, \dots, N - 1$, se obtiene

$$E = \begin{pmatrix} 0 & 0 & 0 & \cdot & \cdot & \cdot & 0 & 0 \\ 0 & 1 & 2 & \cdot & \cdot & \cdot & \cdot & N - 1 \\ 0 & 2 & \cdot & & & & \cdot & \cdot \\ \cdot & \cdot & & & & & \cdot & \cdot \\ \cdot & \cdot & & & & & \cdot & \cdot \\ \cdot & \cdot & & & & & \cdot & \cdot \\ \cdot & \cdot & & & & & \cdot & \cdot \\ 0 & N - 1 & \cdot & \cdot & \cdot & \cdot & \cdot & 1 \end{pmatrix},$$

y así definimos la matriz de Fourier W del sistema (1.13) como la matriz

$$W^E = \begin{pmatrix} 1 & 1 & 1 & \cdot & \cdot & \cdot & 1 & 1 \\ 1 & W^1 & W^2 & \cdot & \cdot & \cdot & \cdot & W^{N-1} \\ 1 & W^2 & \cdot & & & & \cdot & \cdot \\ \cdot & \cdot & & & & & \cdot & \cdot \\ \cdot & \cdot & & & & & \cdot & \cdot \\ \cdot & \cdot & & & & & \cdot & \cdot \\ \cdot & \cdot & & & & & \cdot & \cdot \\ 1 & W^{N-1} & \cdot & \cdot & \cdot & \cdot & \cdot & W^1 \end{pmatrix}$$

1.1.1 Propiedades de la Transformada Discreta de Fourier.

La transformada de Fourier discreta de una señal en S_N produce una señal discreta que también esta en S_N , como lo muestra el siguiente resultado.

Lema 1.1.2. $\mathcal{F}_N(y)$ es un operador lineal de S_N a S_N .

Demostración: Para ver que $\mathcal{F}_N(y)$ pertenece a S_N , veremos que $\mathcal{F}_N(y)_k$ es N -periódico. Tenemos

$$\begin{aligned} \hat{y}_{k+N} &= \sum_{j=0}^{N-1} y_j w^{j(k+N)} \quad (\text{por la ecuación (1.11)}) \\ &= \sum_{j=0}^{N-1} y_j w^{j(k)} w^{jN} \\ &= \sum_{j=0}^{N-1} y_j w^{j(k)} \quad (\text{porque } w^N = e^{-2\pi i N/N} = 1) \\ &= \hat{y}_k. \end{aligned}$$

Por lo tanto, $\hat{y}_{k+N} = \hat{y}_k$ y esta sucesión es N -periódica. Finalmente, puesto que la multiplicación matricial por un vector, es un proceso lineal, $\mathcal{F}_N(y)$ es un operador lineal. □

La matriz W^E tiene propiedades muy interesantes, como veremos a continuación

Definición 1.1.3. Una matriz se dice que es unitaria si cumple lo siguiente

$$A^H A = A A^H = I,$$

donde A^H denota la matriz transpuesta compleja, la matriz transpuesta usual es expresada por A^T .

Un ejemplo de estas matrices es

$$A_\lambda = \begin{pmatrix} \cos(\lambda) & \sin(\lambda) \\ \sin(\lambda) & \cos(\lambda) \end{pmatrix},$$

con $\lambda \in \mathbb{R}$, A_λ es unitaria.

Calcular la inversa de la transformada de Fourier discreta, es equivalente a calcular la inversa de la matriz W^E . El siguiente resultado da una forma fácil para obtener la matriz inversa de W^E .

Teorema 1.1.4. Sea $y = \{y_k\}$ un elemento de S_N . Sea $\mathcal{F}_N(y) = \hat{y}$, entonces $y = \mathcal{F}_N^{-1}(\hat{y})$, esta dado por

$$y_j = \frac{1}{N} \sum_{k=0}^{N-1} \hat{y}_k w^{jk}. \quad (1.14)$$

Nota³

Demostración: Sea r_k el k -ésimo renglón de W^E y sea c_n la columna n -ésima de la matriz W^{-E} , entonces el producto escalar esta dado por:

$$\begin{aligned} r_k c_n &= \left(1, w^k, \dots, w^{(N-1)k}\right) \left(1, w^{-j}, \dots, w^{-(N-1)j}\right)^T \\ &= 1 + w^{k-j} + \dots + w^{(N-1)(k-j)} \end{aligned}$$

Como

$$\sum_{m=0}^{N-1} y^m = \frac{1 - y^N}{1 - y},$$

³Usando notación matricial, el Teorema 1.1.4 asegura que existe una matriz, tal que al multiplicarla por la matriz W^E , se puede recobrar y ; que es,

$$y = \mathcal{F}_N^{-1}(\hat{y}) = \left(W^E\right)^{-1} \hat{y}$$

Puesto que $\hat{y} = \mathcal{F}_N(y)$, encontrar una ecuación tal que $y = \frac{1}{N} W^{-E} \hat{y}$ es equivalente a

$$I_N = \left(\frac{W^E}{\sqrt{N}}\right) \left(\frac{(W^E)^{-1}}{\sqrt{N}}\right)$$

donde I_N es la matriz identidad $I_{N \times N}$, pues W^E es simétrica, la ecuación anterior nos dice que la matriz W^E/\sqrt{N} es unitaria.

entonces al tomar $y = w^{k-j}$ tenemos que

$$r_k c_j = \frac{1 - w^{N(k-j)}}{1 - w^{k-j}} = \begin{cases} N & \text{si } k = j \\ 0 & \text{si } k \neq j \end{cases}$$

Por lo tanto

$$\begin{aligned} \hat{y} &= \frac{1}{N} W^E y \\ &= \frac{1}{N} W^E [W^{-E} \hat{y}] \\ &= \left[\frac{1}{N} W^E W^{-E} \right] \end{aligned}$$

□

Cuando se trata de hacer un procesamiento digital de señal no tiene sentido hablar de convoluciones aplicando estrictamente la definición 1.0.5 ya que solo disponemos de valores en instantes discretos de tiempo. Es necesario, pues, una aproximación numérica. Para las funciones discretas se puede usar una forma discreta de la convolución. Esto es:

$$h(m) * f(m) = (h * f)(m) = \sum_n h(n) f(m - n).$$

De una forma más precisa, definimos tal operación en S_N , como:

Definición 1.1.5. Si $h, f \in S_N$, entonces la sucesión definida por

$$[h * f]_k := \sum_{j=0}^{N-1} h_j f_{k-j},$$

esta también en S_N . La sucesión $[h * f]$ es llamada la convolución de las sucesiones h y f .

Teorema 1.1.6. (de Convolución)

Si $h, f \in S_N$, entonces

$$\mathcal{F}[h * f]_k = \mathcal{F}[h]_k \mathcal{F}[f]_k.$$

Ejemplo 1.1.7.

Sea f una sucesión de números enteros y g una sucesión de números que depende de la sucesión f , un esquema de esto es tomar 5 elementos de la sucesión f y promediar estos para obtener la sucesión g , esto es

$$\begin{array}{ccccccccc} f & 4 & 3 & 9 & 8 & 2 & 1 & 6 & 5 & 4 & 2 & 3 \\ g & & & & & & g_6 = 4.4 & g_7 = 3.6 & g_8 = 3.6 & & & \end{array}$$

donde

$$g_6 = \frac{1}{5}(8 + 2 + 1 + 6 + 5) = 4.4$$

$$g_7 = \frac{1}{5}(2 + 1 + 6 + 5 + 4) = 3.6$$

$$g_8 = \frac{1}{5}(1 + 6 + 5 + 4 + 2) = 3.6$$

En general

$$\begin{aligned} g_n &= \frac{(f_{n-2} + f_{n-1} + f_n + f_{n+1} + f_{n+2})}{5} \\ &= \frac{1}{5} \sum_{m=n-2}^{n+2} f_m \end{aligned}$$

Para nuestro propósito aplicamos un peso a cada uno de los elementos de f , para ello, sea $h = (\dots, 0.3, 0.2, 0.1, 0.3, 0.1, \dots)$ una sucesión 5- periódica de tal forma que además $\sum_{j=1}^5 h_j = 1$, entonces

$$\begin{array}{cccccccccccc} f & 4 & 3 & 9 & 8 & 2 & 1 & 6 & 5 & 4 & 2 & 3 \\ h & & & & 0.3 & 0.2 & 0.1 & 0.3 & 0.1 & & & \end{array}$$

$$g_6 \qquad \qquad \qquad 5.2$$

En este ejemplo la suma de los pesos h es 1, con esto decimos que h esta *normalizada*. También cada elemento del arreglo g es un a combinación lineal de elementos de f , pues

$$\begin{aligned} g_6 &= 8(0.3) + 2(0.2) + 1(0.1) + 6(0.3) + 5(0.1) \\ &= 2.4 + 0.4 + 0.1 + 1.8 + 0.5 \\ &= 5.2 \end{aligned}$$

Generalizando,

$$\begin{array}{cccccccccccc} f_1 & f_2 & f_3 & f_4 & f_5 & f_6 & f_7 & f_8 & f_9 & f_{10} & f_{11} & f_{12} \\ & & & & h_{-2} & h_{-1} & h_0 & h_1 & h_2 & & & \\ g_1 & g_2 & g_3 & g_4 & g_5 & g_6 & g_7 & g_8 & g_9 & g_{10} & g_{11} & g_{12} \end{array}$$

entonces

$$g_7 = f_5 h_2 + f_6 h_1 + f_7 h_0 + f_8 h_{-1} + f_9 h_{-2}$$

En general

$$g_k = \sum_{j=k-2}^{k+2} f_j h_{k-j},$$

que es la convolución de la definición 1.1.5.

□

Si las dos señales f y h son N -periódicas, tales que los términos f_0, \dots, f_{N-1} y h_0, \dots, h_{N-1} son repeditos en las sucesiones f y h , entonces la convolución de estas señales producen la señal de salida g definida por

$$g_i = \sum_{j=0}^{N-1} f_j h_{i-j} \quad i = 0, 1, \dots, N-1$$

Si se considera que g es una imagen observada, la expresión anterior tiene la forma bidimensional siguiente

$$g_{i,k} = \sum_{j=0}^{N-1} \sum_{l=1}^{M-1} f_{i,k} h_{(i-j),(k-l)} \quad i = 0, 1, \dots, N-1, \quad k = 0, 1, \dots, M-1.$$

Muchas veces las imágenes presentan degradación de muchos tipos. Dicha degradación puede ser por ruido durante la transmisión o en otras partes del sistema de captura. Esto se ve a menudo en las imágenes convertidas a digital de una señal analógica. Usando técnicas para filtrar el ruido, el ruido puede ser suprimido y la imagen corrompida se puede restaurar a un nivel aceptable. Una imagen corrupta por ruido impulsivo tiene varios *píxeles* que tienen intensidades visiblemente incorrectas, en la imagen se ve un ruido como de nieve encima de la imagen, dichos píxeles tienen valores extremos en nuestra escala (en los casos más usuales la escala es de 0 a 255 y los valores extremos son parecidos a 0 y 255 para profundizar en este concepto ver el apéndice A y [23]).

Un método eficaz para eliminar el ruido impulsivo es el filtrado por medio de un filtro llamado *filtro de mediana* que usa la idea de atenuar dichos valores extremos. En el filtro de mediana, el nivel de gris de cada píxel se reemplaza por la mediana de los niveles de gris en un entorno (vecindad) de este píxel, en lugar de por la media. La *mediana* m de un conjunto de valores es tal que si ordenamos los valores de forma creciente, obtenemos una sucesión creciente, para el caso de imágenes, el tomar los vecinos de un píxel, incluyendo al píxel mismo, siempre obtendremos un número impar de elementos en nuestra sucesión creciente, para un mallado rectangular. Teniendo dicha sucesión, tomamos el valor que ocupa la posición central de la sucesión, para una ilustración ver Figura 1.2.

La serie de Fourier es álgebra lineal en dimensiones infinitas. Los vectores son funciones $f(x)$; éstas son proyectadas sobre senos y cosenos. Así se obtienen los llamados coeficientes de Fourier a_k y b_k . A partir de esta serie infinita de senos y cosenos, multiplicados por a_k y b_k , es posible reconstruir a $f(x)$. Aplicaciones reales se realizan por medio de la llamada *Transformada Discreta de Fourier* (TDF). Dicha transformada toma valores puntuales de la Transformada de Fourier continua. Estos valores son tomados como una muestra, dicha muestra es puntual, de tal manera que mientras mas puntos se tomen la aplicación en la cual estemos trabajando será de mejor precisión, pero se tendrá una mayor cantidad de datos y por tanto un mayor trabajo computacional. Si tenemos

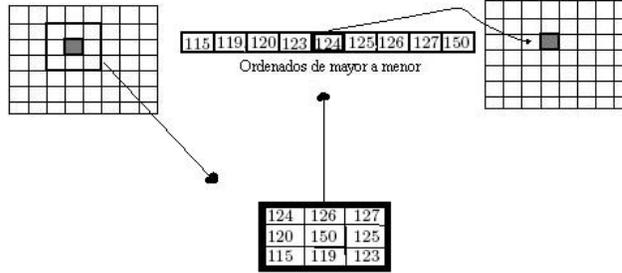


Figura 1.2: Mediana.

menos puntos, la aplicación en la cual estemos trabajando será de menor precisión, pero con menor costo computacional.

Aquí se trata de álgebra lineal pura, basada en la ortogonalidad. La señal de entrada es una sucesión de números $X(0), \dots, X(N-1)$. La señal de salida $x(0), \dots, x(N-1)$ tiene la misma longitud N . La relación entre X y x es lineal, de modo que debe estar expresada mediante una matriz. Esta es la llamada *matriz de Fourier* la cual denotamos por W^E . Toda la tecnología del procesamiento de señales digitales depende de ella. La matriz de Fourier W^E posee propiedades extraordinarias. Las señales se digitalizan, ya sea que provengan de una señal de sonido, de imágenes, de la acústica o de TV, (incluso en la explotación petrolera). Las señales son transformadas por una transformación T a la cual nos referimos al final de la sección anterior. Esta transformación o mejor dicho la matriz asociada a esta transformación es la matriz W^E .

Lo más importante es que W^E tiene una factorización que computacionalmente produce ventajas respecto a calcular la matriz W^E directamente en una computadora. $(W^E)^{-1}$ se ha conocido durante años, y es muy semejante a W^E . De hecho, W^E es simétrica y ortogonal (excepto por el factor constante \sqrt{N}), y sólo tiene un inconveniente: sus elementos son números complejos. Este es un precio bajo que hay que pagar, las dificultades son mínimas por el hecho de que todos los elementos de W^E y $(W^E)^{-1}$ son potencias de un solo número, $w = e^{2\pi i/N}$, tal que $w^N = 1$, es decir, potencias de la raíz N -ésima de la unidad.

Los números complejos $\cos \theta + i \sin \theta$ en la matriz de Fourier son extremadamente especiales. La parte real se traza sobre el eje x y la parte imaginaria sobre el eje y . Así, el número w se encuentra sobre la circunferencia unitaria; su distancia al origen es igual a 1. Forma un ángulo θ con la horizontal. El cuadrado de w puede encontrarse directamente (simplemente se duplica el ángulo):

$$w^2 = (\cos^2 \theta + i \sin^2 \theta)^2 = \cos^2 \theta - \sin^2 \theta + 2i \sin \theta \cos \theta.$$

Recordemos que la parte real $\cos^2 \theta - \sin^2 \theta$ es igual a $\cos 2\theta$, y la parte imaginaria $2 \sin \theta \cos \theta$ es igual a $\sin 2\theta$. Por tanto $w^2 = \cos 2\theta + i \sin 2\theta$, es decir

el cuadrado de W sigue estando en la circunferencia unitaria, pero al ángulo se duplica, esto es, 2θ . Esto hace sospechar que w^N está en el ángulo $N\theta$, y esta sospecha es correcta. Otra manera de representar w es

$$\cos \theta + i \sin \theta = e^{i\theta}$$

Con esta fórmula es posible resolver $w^N = 1$. Esto se convierte en $e^{iN\theta} = 1$, de modo que $N\theta$ debe llevarnos alrededor de la circunferencia unitaria y volver al número original. La solución es tomar $\theta = 2\pi/N$, y en consecuencia considerar la N -ésima raíz primitiva de la unidad

$$w_N = e^{2\pi i/N} = \cos \frac{2\pi}{N} + i \sin \frac{2\pi}{N}$$

Su N -ésima potencia es $e^{2i\pi}$, la cual es igual a 1. Para $N = 8$, esta raíz está dada por $(1+i)/\sqrt{2}$:

$$w_4 = \cos \frac{\pi}{2} + i \sin \frac{\pi}{2} = i$$

y

$$w_8 = \cos \frac{\pi}{4} + i \sin \frac{\pi}{4} = \frac{1+i}{\sqrt{2}}$$

La raíz cuarta está en $\theta = \pi/4$, que es $1/4(2\pi)$. Las otras raíces cuartas son las potencias $i^2 = -1$, $i^3 = -i$, e $i^4 = 1$. Las otras raíces octavas son las potencias $w_8^2, w_8^3, \dots, w_8^8$. Las raíces son equidistantes sobre la circunferencia unitaria, a intervalos de $2\pi/n$. Obsérvese nuevamente que el cuadrado de w_8 es w_4 , lo cual es esencial en la transformada de Fourier rápida. La suma de las raíces cuartas de la unidad es cero; $1 + i - 1 - i = 0$, es decir

$$1 + w_8 + w_8^2 + \dots + w_8^7 = 0$$

Considerando la fórmula de D'Moivre tenemos que $z^N = (re^{i\theta})^N$. Es decir todo número complejo distinto de cero tiene exactamente N raíces N -ésimas complejas, dichas raíces tienen el mismo módulo, y sus argumentos se encuentran igualmente espaciados. Geométricamente, las raíces N -ésimas de un número complejo, distinto de cero, son los vértices de un polígono regular de N lados (ver Figura 1.3).

Para $N = 5$, las soluciones z_k con $k = 0, 1, \dots, 4$ de la ecuación $z^N = 1$ están dados por $z_k = (e^{i\theta_k})$, con $\theta_k = 2\pi k/5$, las raíces N -ésimas se muestran en la Figura 1.3. El siguiente resultado generaliza un poco lo dicho anteriormente.

Proposición 1.1.8. *Las raíces N -ésimas de la unidad (las soluciones de $z^N = 1$), están dadas por $w = \cos(\frac{2\pi}{N}) + i \sin(\frac{2\pi}{N})$. Si w es una raíz distinta de uno, entonces las raíces pueden expresarse como $1, w, \dots, w^{N-1}$, y si $w \neq 0$ es solución de la ecuación:*

$$0 = z^N - 1 = (z - 1)(z^{N-1} + z^{N-2} + \dots + z^2 + z + 1),$$

tenemos que w es solución de la ecuación

$$z^{N-1} + z^{N-2} + \dots + z^2 + z + 1 = 0.$$

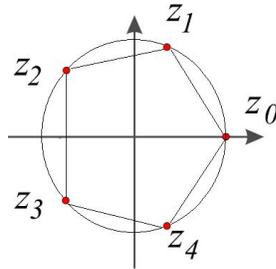


Figura 1.3: Raíces quintas de la unidad.

1.2 Transformada Rápida de Fourier.

En esta sección queremos estudiar el problema de calcular los componentes de la Transformada de Fourier Discreta de un vector $y \in S_N$. Recordemos que esta expresión la podemos escribir en una forma matricial como en la ecuación (1.13).

Veamos un algoritmo para representar la Transformada de Discreta de Fourier para una matriz cuadrada. Sea $W \in \mathbb{C}^{N \times N}$ una matriz de Fourier y sea $y \in S_N$, podemos evaluar $\hat{y} = Wy$ mediante el algoritmo siguiente

```

for k=0,1,...,N-1
   $\hat{y}=0$ 
  for n=0,1,...,N-1
     $\hat{y} = \hat{y} + W_{kn}y$ 
  end
end,

```

donde $W_{kn} = e^{k(2\pi i/n)}$, este algoritmo transforma el vector y en el vector \hat{y} . El costo de este cálculo es aproximadamente N^2 operaciones complejas cuando definimos la operación interna del bucle, esto es cuando se calcula: $\hat{y} = \hat{y} + W_{kn}y$. De hecho, cuando se realiza una operación compleja se requiere más memoria que al realizar un cálculo con números reales. Usando la descomposición real-imaginaria

$$\begin{aligned} \hat{y}_R + i\hat{y}_I &= \hat{y}_R + i\hat{y}_I + (W_R + iW_I)(y_R + iy_I) \\ &= \{(\hat{y}_R + W_R y_R) - y_R W_I\} + i\{(\hat{y}_I + W_I y_I) + W_I y_I\} \end{aligned}$$

Con esto se verifica que para realizar una operación compleja, se requiere de cuatro operaciones reales. Por esta razón vemos que la matriz de Fourier $W \in \mathbb{C}^{N \times N}$ que se puede construir requiere aproximadamente N^2 operaciones complejas ó $4N^2$ operaciones reales.

Uno no aprecia la revolución científica que la Transformada Rápida de Fourier aporta al trabajo computacional, hasta que se comprende que para realizar una operación compleja, se deben realizar 4 operaciones reales, esto es, uno debe de pagar el precio de $4N^2$ operaciones reales para conseguir realizar el cálculo de la Transformada Discreta de Fourier de un vector de tamaño N . Hablando de calcular operaciones, no debería de haber dificultad, para realizar el cálculo real triple del algoritmo anterior, esto es, realizar manualmente la operación $\mathbf{X}=\mathbf{X}+\theta_{kn}\mathbf{x}(\mathbf{n})$ que esta dentro del bucle a mano, digamos en 10^2 segundos o un minuto con 40 segundos (Esto permite tener un tiempo necesario para checar que el resultado sea correcto.). Si uno puede mantener esta velocidad de cálculo, se puede generar un vector de tamaño 8 aplicando la Transformada Discreta de Fourier en aproximadamente

$$4 \cdot 12^2 \text{operaciones} \cdot \frac{10^2 \text{seg}}{\text{operación}} \cdot \frac{1 \text{min}}{60 \text{seg}} \cdot \frac{1 \text{hr}}{60 \text{min}} = 16 \text{hr}.$$

Nos preguntamos ¿Que motiva el llevar a cabo este cálculo?, (en el sentido computacional) la respuesta es análoga a la pregunta de calcular la inversa de una matriz con el método de Gauss. Durante la vida de las computadoras digitales, hemos visto que continuamente cambia el costo unitario de una operación, pero siempre para efectuar la TDF se mantiene el orden de $4N^2$ operaciones. En los 50s una computadora tenía la capacidad de realizar 10^3 operaciones por segundo, un cálculo de un vector de tamaño 100 TDF en aproximadamente

$$4 \cdot 100^2 \text{operaciones} \cdot \frac{1 \text{seg}}{10^3 \text{operaciones}} = 40 \text{seg},$$

pero un vector TDF de tamaño 1000 en $4000 \text{ seg} = 1.1 \text{ hr}$. En el 2000 una PC tenía la capacidad de realizar 10^7 operaciones por segundo, se podía realizar un vector TDF en aproximadamente

$$4 \cdot 1000^2 \text{operaciones} \cdot \frac{1 \text{seg}}{10^7 \text{operaciones}} = 0.4 \text{seg}.$$

Esto parece rápido, pero si un algoritmo TDF calcula cada cuadro de una película de tamaño 1000 deberíamos esperar $400 \text{ seg} = 6.7 \text{ min}$ para ver el resultado. Sería desesperante el estar viendo un video en la PC con ese tipo de desfase. En 1965, James W. Cooley y Jhon W. Tukey publicaron un algoritmo nuevo y sustancialmente rápido para calcular un N -vector TDF en una computadora digital. Ellos se dan cuenta que por medio de la factorización

$$N = 2^L \quad \text{ó} \quad N = P_1 P_2 \cdots P_m,$$

donde $2 \leq P_i, L \in \mathbb{Z}$, para $i = \{1, \dots, m\}$, es posible reducir el costo computacional de un N -vector TDF de la siguiente manera

$$N^2 = N \cdot \{P_1 \cdots P_m\} \quad \text{a} \quad N \cdot \{(P_1 - 1) + (P_2 - 1) + \cdots + (P_m - 1)\},$$

operaciones complejas. La reducción del costo es muy dramático en los casos donde $P_1 = P_2 = \dots = P_m = 2$ cuando pasamos de

$$N^2 = 2^{2m} \quad a \quad 2^m \cdot m = N \log_2 N$$

operaciones complejas. Por ejemplo, cuando $N = 1024 = 2^{10}$ se reduce el costo de $N^2 = 1,048,576$ a $N \log_2 N = 10,240$ operaciones complejas. El nuevo algoritmo reduce el costo de un vector de tamaño 1024 TDF por un factor de 100. Tal reducción en el costo computacional hace que la Transformada de Fourier Discreta sea práctica al ser implementada como algoritmo en una computadora.

1.2.1 Algoritmos FFT.

El algoritmo FFT lo único que busca es resolver de la manera más eficiente posible la ecuación (1.11), donde, como ya vimos, la evaluación directa de esta suma implica N^2 operaciones. Es necesario hacer una serie de ordenaciones para conseguir una versión rápida (computacionalmente hablando) de (1.11) llamada transformada rápida de Fourier ó por sus siglas en inglés FFT (Fast Fourier Transform). Primero se deben separar las muestras pares e impares:

Sea $N = 2^L$, $L \in \mathbb{Z}$

$$\begin{aligned} \hat{y} &= \frac{1}{N} \sum_{j=0}^{N-1} y_j W^{kj} \\ &= \frac{1}{N} \sum_{j=0}^{N/2-1} \left[y_j W^{kj} + y_{j+N/2} W^{k(j+N/2)} \right] \\ &= \frac{1}{N} \sum_{j=0}^{N/2-1} \left[y_j + y_{j+N/2} W^{kN/2} \right] W^{kj}, \end{aligned}$$

esto es

$$\hat{y} = \frac{1}{N} \sum_{j=0}^{N/2-1} \left[y_j + y_{j+N/2} W^{kN/2} \right] W^{kj} \quad (1.15)$$

- Si $k = 2l$, $l = 0, 1, 2, \dots, N/2 - 1$

$$g_j = y_j + y_{j+N/2} W^{kN/2}$$

pues

$$\begin{aligned} W^{kN/2} &= e^{-\frac{j2\pi}{N} \frac{kN}{2}} \\ &= e^{-j\pi} \\ &= \cos k\pi - j \sin k\pi = (-1)^k \end{aligned}$$

y entonces (1.13) cambia a

$$\hat{y}_{2l} = \frac{1}{N/2} \frac{1}{2} \sum_{j=0}^{N/2-1} g_n (W^2)^{kj} \quad (1.16)$$

La última expresión reduce el número de entradas por 2, usando $N/2$ puntos para la sucesión $g(n)$.

- Si $k = 2l + 1$, $0, \dots, N/2 - 1$ y haciendo

$$h_j = y_j - y_{j+N/2}$$

la serie (1.13) se transforma en

$$\begin{aligned} \hat{y}_{2l+1} &= \frac{1}{N} \sum_{j=0}^{N/2-1} h_j W^{(2l+1)j} \\ &= \frac{1}{2} \frac{1}{N/2} \sum_{j=0}^{N/2-1} \check{h}_j (W^2)^{lj} \end{aligned}$$

Por tanto

$$\hat{y}_{2l+1} = \frac{1}{2} \frac{1}{N/2} \sum_{j=0}^{N/2-1} \check{h}_j (W^2)^{lj} \quad (1.17)$$

donde $\check{h}_j = h_j W^j$.

Para el cálculo eficiente de TDF, se usa por lo regular TRF (FFT por sus siglas en inglés), esto es, se usa el algoritmo 2-FFT, con ecuaciones del tipo (1.16) y (1.17), las cuales reducen de manera significativa el número de operaciones y por ello el costo computacional. Existen algoritmos TRF que calculan eficientemente su inversa (IFFT, la inversa FFT) hasta con un orden de $\mathcal{O}(N \log_2 N)$ operaciones. Esto se debe a las técnicas basadas en el algoritmo propuesto por Cooley-Tukey, como cuando la matriz W^E se factoriza de un manera muy *conveniente*, por ejemplo; $W^{E_L \dagger E_{L-1} \dots \dagger E_1} = W^{E_{L-1}} \dots W^{E_1}$, donde el producto *conveniente* de estas matrices reduce el número de operaciones.

1.2.2 Factorización de una matriz rápida de Fourier.

En esta sección queremos producir una factorización de W^E de la forma

$$W^E = W^{E_{L-1}} \dots W^{E_1} B,$$

cuando $N = 2^L$. Denotemos; $y = f$ y $W^E = \mathcal{F}$. Quisiéramos que cada una de las matrices $B, W^{E_{L-1}}, \dots, W^{E_1}$ sean matrices que tengan pocos elementos distintos de cero en cada renglón, tal que con el mínimo esfuerzo podamos calcular

$$f_0 = Bf, \quad f_1 = W^{E_1} f_0, \quad f_2 = W^{E_2} f_1, \dots, f_L = W^{E_L} f_{L-1},$$

de esta forma

$$f_L = W^{E_L} W^{E_{L-1}} \dots W^{E_1} Bf = \mathcal{F}f.$$

Podemos llevar a cabo este proceso con el siguiente algoritmo

```
f:=Bf
for μ=1,2,...,L
f:=WμEf
end
```

que sucesivamente reescribe el vector original f con f_0, f_1, \dots, f_L . Las matrices $B, W^{E_1}, W^{E_2}, \dots, W^{E_L}$ tienen estructuras simples para facilitar la lectura del código.

Para aclarar esta factorización tomemos el caso $N = 2^3 = 8$ y sea $w = e^{-2\pi i/8}$. Recordemos que la estructura de $\mathcal{F} = W^E$ es

$$\mathcal{F} = W^E = \begin{pmatrix} 1 & 1 & 1 & \dots & 1 & 1 \\ 1 & w^1 & w^2 & \dots & \dots & w^{N-1} \\ 1 & w^2 & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 1 & w^{N-1} & \dots & \dots & \dots & w^1 \end{pmatrix},$$

Para $N = 8$

$$\mathcal{F}_8 = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & w^1 & w^2 & w^3 & w^4 & w^5 & w^6 & w^7 \\ 1 & w^2 & w^4 & w^6 & w^0 & w^2 & w^4 & w^6 \\ 1 & w^3 & w^6 & w^1 & w^4 & w^7 & w^2 & w^5 \\ 1 & w^4 & w^0 & w^4 & w^0 & w^4 & w^0 & w^4 \\ 1 & w^5 & w^2 & w^7 & w^4 & w^1 & w^6 & w^3 \\ 1 & w^6 & w^4 & w^2 & w^0 & w^6 & w^4 & w^2 \\ 1 & w^7 & w^6 & w^5 & w^4 & w^3 & w^2 & w^1 \end{pmatrix}$$

Podemos ver fácilmente que

$$\mathcal{F}_8 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & w^2 & w^4 & w^6 & w^1 & w^3 & w^5 & w^7 \\ 1 & w^4 & w^8 & w^{12} & w^2 & w^6 & w^{10} & w^{14} \\ 1 & w^6 & w^{12} & w^{18} & w^3 & w^9 & w^{15} & w^{21} \\ 1 & w^8 & w^{16} & w^{24} & w^4 & w^{12} & w^{20} & w^{28} \\ 1 & w^{10} & w^{20} & w^{30} & w^5 & w^{15} & w^{25} & w^{35} \\ 1 & w^{12} & w^{24} & w^{36} & w^6 & w^{18} & w^{30} & w^{42} \\ 1 & w^{14} & w^{28} & w^{42} & w^7 & w^{21} & w^{35} & w^{49} \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix},$$

donde usamos las identidades $w^2 = e^{-2\pi i/4}$, $w^4 = -1$, $w^8 = 1$ para expresar cada bloque 4×4 de la matriz en terminos de \mathcal{F}_4

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ w & w^3 & w^5 & w^7 \\ w^2 & w^6 & w^{10} & w^{14} \\ w^3 & w^9 & w^{15} & w^{21} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & w & 0 & 0 \\ 0 & 0 & w^2 & 0 \\ 0 & 0 & 0 & w^3 \end{bmatrix} \mathcal{F}_4,$$

de forma análoga

$$\begin{bmatrix} w^4 & w^{12} & w^{20} & w^{28} \\ w^5 & w^{15} & w^{25} & w^{35} \\ w^6 & w^{18} & w^{30} & w^{42} \\ w^7 & w^{21} & w^{35} & w^{49} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -w & 0 & 0 \\ 0 & 0 & -w^2 & 0 \\ 0 & 0 & 0 & -w^3 \end{bmatrix} \mathcal{F}_4,$$

donde

$$\mathcal{F}_4 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & w^2 & w^4 & w^6 \\ 1 & w^4 & w^8 & w^{12} \\ 1 & w^6 & w^{12} & w^{18} \end{bmatrix}.$$

En este sentido obtenemos la factorización

$$\mathcal{F}_8 = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & w & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & w^2 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & w^3 \\ 1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & -w & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & -w^2 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & -w^3 \end{bmatrix} \begin{bmatrix} \mathcal{F}_4 & \mathbf{0}_4 \\ \mathbf{0}_4 & \mathcal{F}_4 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Un argumento semejante da la siguiente identidad

$$\mathcal{F}_{2M} = Q_{2M} \begin{bmatrix} \mathcal{F}_M & \mathbf{0}_M \\ \mathbf{0}_M & \mathcal{F}_M \end{bmatrix} S_{2M}, \quad M = 1, 2, \dots \quad (1.18)$$

donde $Q_{2M} \in \mathbb{C}^{M' \times M'}$ y $M' = 2M$, entonces

$$Q_{2M} = \begin{bmatrix} 1 & 0 & 0 & 0 & w^0 & 0 & 0 & 0 \\ 0 & \cdot & 0 & 0 & 0 & \cdot & 0 & 0 \\ 0 & 0 & \cdot & 0 & 0 & 0 & \cdot & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & w^{M-1} \\ 1 & 0 & 0 & -w^0 & 0 & 0 & 0 & 0 \\ 0 & \cdot & 0 & 0 & 0 & \cdot & 0 & 0 \\ 0 & 0 & \cdot & 0 & 0 & 0 & \cdot & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & -w^{M-1} \end{bmatrix}$$

y donde obtenemos la matriz permutación S_{2M} , tal que

$$S_{2M} = [\delta_0, \delta_M, \delta_{M+1}, \delta_2, \delta_{M+2}, \dots, \delta_{M-1}, \delta_{2M-1}] \quad (1.19)$$

cada δ_k para $k = \{0, \dots, 2M - 1\}$, es el k -ésimo vector columna de la matriz identidad. Esto quiere decir que la matriz S_{2M} es una matriz de permutación que es consecuencia de permutar los vectores columna de la matriz identidad $I \in \mathbb{C}^{M' \times M'}$ de tal manera que la identidad se obtiene con

$$I_{2M} = [\delta_0, \delta_1, \dots, \delta_{M-1}, \delta_M, \delta_{M+1}, \dots, \delta_{2M-1}]$$

1.2.3 Notacion exponencial.

Dada una matriz $A \in \mathbb{C}^{M \times M}$ y la matriz de ceros $\mathbf{0} \in \mathbb{C}^{M \times M}$ definamos

$$A^{(1)} := A, \quad A^{(2)} := \begin{bmatrix} A & \mathbf{0} \\ \mathbf{0} & A \end{bmatrix}, \quad A^{(3)} := \begin{bmatrix} A & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & A & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & A \end{bmatrix}, \dots$$

$$\dots, A^{(k)} := \begin{bmatrix} A & & \\ & \ddots & \\ & & A \end{bmatrix},$$

donde $A \in \mathbb{C}^{M \times M}$, $A^2 \in \mathbb{C}^{2M \times 2M}$, $A^3 \in \mathbb{C}^{3M \times 3M}$, ..., $A^k \in \mathbb{C}^{kM \times kM}$. Así producimos las matrices diagonales con índices $p, q \in \mathbb{N}$ que son copias de A a lo largo de la diagonal. Fácilmente se verifican la siguientes propiedades

1. $[A^{(p)}]^q = A^{(pq)}$, con $p, q \in \mathbb{N}$
2. $[AB]^{(p)} = A^{(p)}B^{(p)}$, con $p \in \mathbb{N}$ y $B \in \mathbb{C}^{M \times M}$
3. $[\alpha A]^{(p)} = \alpha A^{(p)}$, con $p \in \mathbb{N}$ y $\alpha \in \mathbb{C}$
4. $[A^T]^{(p)} = [A^{(p)}]^T$, con $p \in \mathbb{N}$ donde A^T es la matriz transpuesta de A .
5. $[A^{-1}]^{(p)} = [A^{(p)}]^{-1}$, con $p \in \mathbb{N}$ donde A es una matriz no singular.

Con la notación establecida y estas propiedades obtenemos la identidad (1.19)

$$\mathcal{F}_{2M} = Q_{2M} \mathcal{F}^{(2)} S_{2M} \quad M = 1, 2, \dots \quad (1.20)$$

Esta identidad nos permite factorizar \mathcal{F}_N cuando $N = 2^L$ con $L = 0, 1, 2, \dots$ Por ejemplo podemos descomprimir \mathcal{F}_{16} usando (1.20) y las propiedades ante-

riores

$$\begin{aligned}
\mathcal{F}_{16} &= Q_{16} \mathcal{F}_8^{(2)} S_{16} \\
&= Q_{16} \left[Q_8 \mathcal{F}_4^{(2)} S_8 \right]^{(2)} S_{16} \\
&= Q_{16} Q_8^{(2)} \mathcal{F}_4^{(4)} S_8^{(2)} S_{16} \\
&= Q_{16} Q_8^{(2)} \left[Q_4 \mathcal{F}_2^{(2)} S_4 \right]^{(4)} S_8^{(2)} S_{16} \\
&= Q_{16} Q_8^{(2)} Q_4^{(4)} \mathcal{F}_2^{(8)} S_4^{(4)} S_8^{(2)} S_{16} \\
&= Q_{16} Q_8^{(2)} Q_4^{(4)} \left[Q_2 \mathcal{F}_1^{(2)} S_2 \right]^{(8)} S_4^{(4)} S_8^{(2)} S_{16} \\
&= Q_{16} Q_8^{(2)} Q_4^{(4)} Q_2^8 \mathcal{F}_1^{(16)} S_2^{(8)} S_4^{(4)} S_8^{(2)} S_{16} \\
&= Q_{16} Q_8^{(2)} Q_4^{(4)} Q_2^8 B_{16},
\end{aligned}$$

donde $B_{16} := S_2^{(8)} S_4^{(4)} S_8^{(2)} S_{16}$, de tal manera que $\mathcal{F}_1^{(16)} = [1]^{(16)} = \mathbf{I}_{16}$ es la identidad de tamaño 16×16 . De forma análoga

$$\mathcal{F}_{2^L} = Q_{2^L} Q_{2^{L-1}}^{(2)} Q_{2^{L-2}}^{(4)} \cdots Q_2^{(2^{L-1})} B_{2^L}, \quad L = 0, 1, 2, \dots \quad (1.21)$$

donde $B_{2^L} := S_2^{(2^{L-1})} S_4^{(2^{L-2})} \cdots S_{2^{L-1}}^{(2)} S_{2^L}$. La matriz B_{2^L} arrastra literalmente una permutación puesto que esta formada por matrices de permutación S_{2^M} para $M = 1, 2, \dots$. De modo que B_8 transforma a $f = (f_0, f_1, \dots, f_7)^T$ en

$$B_8 f = (f_0, f_4, f_2, f_6, f_1, f_5, f_3, f_7)^T,$$

i.e. B_8 es la permutación para el vector f de tamaño 2^3 a través de la ecuación (1.20).

La factorización (1.21) corresponde a un algoritmo rápido para calcular la TDF de cualquier vector de tamaño $N = 2^L$. Si tomamos en cuenta que la matriz \mathcal{F}_{2^L} es simétrica podemos usar (1.21) y la propiedad (4) de exponentes matriciales de la página anterior para escribir

$$\begin{aligned}
\mathcal{F}_{2^L} &= \mathcal{F}_{2^L}^T \\
&= \left[Q_{2^L} Q_{2^{L-1}}^{(2)} Q_{2^{L-2}}^{(4)} \cdots Q_2^{(2^{L-1})} B_{2^L} \right]^T \\
&= B_{2^L}^T \left[Q_2^{(2^{L-1})} \right]^T \cdots \left[Q_{2^{L-1}}^{(2)} \right]^T \left[Q_{2^L} \right]^T \\
&= B_{2^L} \left[Q_2^T \right]^{(2^{L-1})} \cdots \left[Q_{2^{L-1}}^T \right]^{(2)} \left[Q_{2^L}^T \right]
\end{aligned}$$

donde $B_{2L}^T = B_{2L}$, B_{2L} es unitaria, pues cumple con la Definición 1.1.3, entonces $B_{2L}^T B_{2L} = \mathbf{I}_{2L}$. Por tanto

$$\mathcal{F}_{2^L} = B_{2^L} [Q_2^T]^{(2^L-1)} \cdots [Q_{2^{L-1}}^T]^{(2)} [Q_{2^L}^T], \quad (1.22)$$

es también el corazón de un algoritmo rápido para calcular la TDF. Hasta este momento comprendemos el algoritmo Transformada Rápida de Fourier (FFT por sus siglas en inglés) cuando $N = 2^L$. El esquema central de la Transformada Rápida de Fourier es el ya analizado en esta sección (Las ecuaciones (1.20) y (1.22) pueden ser escritas teóricamente en términos del producto de Kronecker, ver apéndice **D**).

1.2.4 Transformada discreta de Fourier en Matlab.

Aunque Matlab tiene implementada la Transformada Discreta de Fourier por medio de un algoritmo relativamente rápido, la implementación en un lenguaje de alto nivel es aún más rápida. (En el apéndice E se muestra un código donde esta implementada esta Transformada Rápida de Fourier en Fortran.)

Usando la notación clásica y que Matlab retoma, para la convolución $g = f * h$, al aplicar la TDF, usando el teorema de convolución $G = FH$, donde, en matlab $G = \text{fft}(g)$, $F = \text{fft}(f)$ y $H = \text{fft}(h)$, la siguiente relación se cumple

$$\text{fft}(g) = \text{fft}(f) .* \text{fft}(h) \quad (1.23)$$

i.e. $\text{fft}(g)$ es el producto de los vectores $\text{fft}(f)$ y $\text{fft}(h)$. Como consecuencia obtenemos la inversa de la TDF en Matlab

$$g = \text{ifft}(\text{fft}(f) .* \text{fft}(h))$$

Definición 1.2.1. La *deconvolución* es el proceso para calcular la señal original f dadas las señales g y h , a partir de la ecuación (1.7).

Para señales discretas y periódicas, la expresión simple para calcular f considerado como un vector de tamaño N es

$$f = \text{ifft}(\text{fft}(g) ./ \text{fft}(h)). \quad (1.24)$$

La TRF (FFT) se puede usar para realizar estos cálculos eficientemente. Por lo general para señales discretas la transformada de Fourier de la señal f deconvolucionada es formalmente dada por $F = G/H$, pero no hay esquema de cómputo similar para calcular f , aunque los algoritmos de TRF (FFT) y la ecuación (1.12) sean de uso frecuente, son mal empleados, pues el problema es mal condicionado ver apéndice *B*.

1.3 Representación de imágenes.

Es necesario conocer la representación de una imagen para poder manipularla en una computadora, esto para poder procesar las imágenes. Una de los objetivos del procesamiento de imágenes es su mejoramiento en su calidad. Los sistemas de adquisición de imágenes no ofrecen imágenes con calidad perfecta, hay degradación por distintos motivos. Por esta razón la imagen necesita ser restaurada. Para recuperar una imagen, se usan técnicas en modelos deterministas y modelos aleatorios ó probabilistas (pseudo-aleatorios, puesto que se usa una computadora). La degradación en estos modelos son basados en el concepto de función de dispersión del punto, veamos un modelo espacial (dominio espacial) lineal invariante (no cambia con el tiempo);

$$g(r, c) = \sum_{j=0}^{N-1} \sum_{i=0}^{M-1} f(i, j)h(r - i, c - j) + \eta(r, c),$$

para $r = 0, 1, \dots, M - 1$ y $c = 0, 1, \dots, N - 1$. En notación matricial tenemos

$$\mathbf{g} = \mathbf{H}\mathbf{f} + \eta,$$

donde \mathbf{g} , \mathbf{f} y η son vectores MN -dimensionales y \mathbf{H} es una matriz bloque circulante (Una matriz circulante está totalmente determinado por un vector, c , que aparece como la primera columna de la C . El resto de columnas de C son permutaciones cíclicas del vector c con desplazamiento igual al índice de la columna. La última fila de C es el vector c en orden inverso, y las filas restantes son permutaciones cíclicas de la última fila. ver [17] y [18]), h ó \mathbf{H} representan procesos de degradación integrados en el proceso de formación de imágenes. Hay varios tipos de degradaciones como; degradación puntual, espacial, temporal, cromática y combinaciones de estas. Consideremos solo la degradación espacial. Algunos ejemplos de esta degradación son; (i) turbulencia atmosférica, (ii) movimiento de borrado y (iii) enfocamiento defectuoso. La función de dispersión del punto para estas degradaciones esán descritas a continuación:

(i) Turbulencia atmosférica.

$$H(\omega_x, \omega_y) = \exp \left\{ -\frac{(\omega_x^2 + \omega_y^2)}{2\sigma^2} \right\}$$

(ii) Movimiento uniforme.

$$H(\omega_x, \omega_y) = \frac{\sin a(\omega_x \cos \theta + \omega_y \sin \theta)}{\omega_x \cos \theta + \omega_y \sin \theta}$$

(iii) Mal enfocamiento debido a la lente.

$$H(\omega_x, \omega_y) = \frac{J_1(a\rho)}{a\rho}, \quad \rho = \sqrt{(\omega_x^2 + \omega_y^2)},$$

donde

$$J_1(x) = \sum_{n=0}^{\infty} \frac{(-1)^n x^{2n+1}}{n!(n+1)!2^{2n+1}},$$

es la función de Bessel de primer orden.

$H(\omega_x, \omega_y)$ es la representación de la función de dispersión del punto en el dominio de frecuencias continuo, en el dominio discreto se representa como $H(u, v)$. Otra fuente de degradación que se presenta en las imágenes es el ruido η .

El problema de recuperar la imagen, puede ser visto también como el obtener una aproximación de f , denotada como \hat{f} , dada la imagen observada g , la matriz de degradación H ó llamada función de dispersión del punto y el ruido η . Queremos entonces encontrar un operador inverso Φ , el cual nos de la imagen restaurada,

$$\Phi(g) \rightarrow \hat{f}.$$

En el dominio de Fourier (dominio de frecuencias), Φ es conocido como la función de filtrado.

1.4 Problemas de Deconvolución.

Dadas dos funciones f y h , la operación de convolución (Definición 1.0.5) tiene la forma

$$g(s) = \int_0^1 h(s-t)f(t)dt, \quad 0 \leq s \leq 1$$

Aunque los problemas de convolución pueden estar definidas en el intervalo $[0, \infty]$, para los problemas de deconvolución consideraremos el intervalo $[0, 1]$. Matemáticamente f y h juegan papeles similares (ver Teorema 1.0.4 primer punto). La definición de la convolución (definición 1.0.5) proporciona un problema, donde las funciones h y f son conocidas y se requiere determinar la función g . Pero si no fuera el caso anterior, entonces h ó f serían desconocidas, en este caso llegaríamos a un problema de *deconvolución*. La deconvolución es el proceso de obtener ó recuperar la función f a partiendo de las funciones conocidas h y g .

Los problemas de deconvolución están clasificados en tres grupos.

I Problema de deconvolución: En este caso queremos recuperar la función original f de la definición de convolución (definición 1.0.5), con las funciones h y g son conocidas.

II Problema de deconvolución miope: Es este caso queremos recuperar la función original f de la definición 1.0.5, cuando g es observada ó conocida y se tiene información parcial de la función h .

III Problema de deconvolución Ciega: Este es el caso de la deconvolución donde se involucran situaciones reales y más complicados de resolver. En este caso queremos recuperar la función original f desconociendo por completo a las funciones f y h , solamente se conoce la función g .

Ejemplo 1.4.1.

- En Geomagnetismo

$$g(s) = \int_0^1 \frac{d}{(d^2 + (s-t)^2)^{3/2}} f(t) dt,$$

$$\text{con } h(s-t) = \frac{d}{(d^2 + (s-t)^2)^{3/2}}.$$

- Astrometría (paralajes estelares)

Si f denota la distribución real del paralaje, entonces f y g están relacionados por la ecuación.

$$g(s) = \int_0^1 \frac{\exp\left(-\frac{1}{2}\left(\frac{s-t}{\sigma}\right)^2\right)}{\sigma\sqrt{2\pi}} f(t) dt$$

donde σ es un parámetro de exactitud, en este caso

$$h(s-t) = \frac{\exp\left(-\frac{1}{2}\left(\frac{s-t}{\sigma}\right)^2\right)}{\sigma\sqrt{2\pi}}.$$

- Ecuación inversa del calor. Otra vez un problema inverso puede ser representado por una ecuación integral de primer tipo, para el caso unidimensional acotado (i.e. $\Omega = \mathbb{R}$), la temperatura final $u(\cdot, T)$ está relacionada con la temperatura inicial $u(\cdot, 0)$, (ver referencia [1] pag 14)

$$u(x, T) = \frac{1}{2\sqrt{\pi T}} \int_{-\infty}^{\infty} u(s, 0) \exp\left(-\frac{(x-s)^2}{4T}\right) ds,$$

la cual es una *ecuación de convolución* con kernel

$$h(s-t) = \frac{1}{2\sqrt{\pi T}} \exp\left(-\frac{(x-s)^2}{4T}\right)$$

de suavizado.

Ya que hemos visto algunos ejemplos de problemas de deconvolución, podemos decir que estos problemas están contenidos en un grupo más extenso en matemáticas llamado los *problemas inversos*, los cuales son problemas matemáticos, mal planteados de acuerdo con la definición de **Hadamard**. Un problema bien planteado debe satisfacer las tres propiedades expuestas en la siguiente definición.

Definición 1.4.2. *Un problema matemático está bien planteado si cumple con las siguientes propiedades:*

1. *Para todos los datos posibles, existe una solución.*
2. *Para todos los datos posibles, la solución es única.*
3. *La solución depende continuamente de los datos.*

Un problema está mal planteado si no es bien planteado, i.e. no cumple con alguna de las propiedades anteriores. Para hacerla precisa en una situación concreta, se debe especificar; la noción de solución, los datos considerados posibles y cual es la topología usada para la medición de la continuidad. Al igual que los problemas del Ejemplo 1.4.1, la deconvolución es un problema inverso mal planteado, para su estudio se requiere introducir algunos conceptos más.

1.4.1 Integral de Fredholm

La forma genérica de la ecuación integral de Fredholm es:

$$g(s) = \int_0^1 k(s, t)f(t)dt \quad 0 \leq s \leq 1 \quad (1.25)$$

donde k es el kernel ó núcleo, g es una función conocida y f es una función por determinar. Notemos que los problemas de deconvolución son casos especiales de la ecuación (1.25), donde $k(s, t) = h(s - t)$. Con el kernel descrito de esta forma se obtiene otra ecuación famosa, llamada la ecuación de *Volterra*, cuya expresión es

$$g(s) = \int_0^1 h(s - t)f(t)dt \quad 0 \leq s \leq 1 \quad (1.26)$$

el kernel ó núcleo de la función es cero cuando $t > s$.

1.4.2 Alisamiento e Inversión.

La ventaja de trabajar con (1.25) es que la teoría que se deriva está bien desarrollada. Como ya comentamos, la operación de deconvolución de f con k es una operación de alisamiento o amortiguamiento, donde g es una función alisada. El proceso inverso, esto es calcular f de g , se puede decir que es un proceso de desalisar. Si tuviéramos precisión infinita, podríamos calcular f de manera perfecta. Desafortunadamente esta situación no pasa en la vida real ni en la práctica, ya que en los cálculos numéricos existen redondeos no-insignificantes. Podemos ilustrar lo antes mencionado con un ejemplo de procesamiento de señales.

Ejemplo 1.4.3.

Si h corresponde a la operación de alizamiento, sea $\tilde{g} = g + e$, donde g es la información exacta, e es el ruido, con una distribución de probabilidad uniforme en el intervalo $(0, 1)$. Entonces la TDF (DFT) de \tilde{g} esta dada por

$$\begin{aligned}\mathbf{fft}(\tilde{g}) &= \mathbf{fft}(g) + \mathbf{fft}(e) \\ &= \mathbf{fft}(g) + w\end{aligned}$$

donde $w = \mathbf{fft}(e)$, que es un vector, cuyos elementos tienen la misma probabilidad. Entonces la expresión para TDF, para la solución con ruido f es

$$f = \mathbf{ifft}(\mathbf{fft}(g) ./ \mathbf{fft}(h)).$$

Por tanto

$$\begin{aligned}\mathbf{fft}(\tilde{f}) &= \mathbf{fft}(\tilde{g}) ./ \mathbf{fft}(h) \\ &= (\mathbf{fft}(g) + w) ./ \mathbf{fft}(h) \\ &= \mathbf{fft}(g) ./ \mathbf{fft}(h) + w ./ \mathbf{fft}(h) \\ &= \mathbf{fft}(f) + w ./ \mathbf{fft}(h).\end{aligned}$$

Donde el termino $w ./ \mathbf{fft}(h)$ es castigado cuando los elementos de $\mathbf{fft}(h)$ son pequeños.

1.4.3 Regla de Discretización

Una integral definida puede ser aproximada mediante una expresión de la forma

$$\int_0^1 \varphi(t) dt \simeq \sum_{j=1}^n w_j \varphi(t_j)$$

donde $t_j \in [0, 1]$ y w_j representan un *peso* para cada valor $\varphi(t_j)$.

Ejemplo 1.4.4.

- Regla del punto medio, donde el *peso* está dado por

$$\bar{t}_j = \frac{t_{j-1} + t_j}{2}, \quad w_j = 1/n, \quad j = 1, \dots, n$$

Por lo que

$$\int_0^1 \varphi(t) dt \simeq \sum_{j=1}^n w_j \varphi(\bar{t}_j)$$

- Si la aproximación se realiza mediante la Regla de Simpson

$$\begin{aligned}\int_0^1 \varphi(t) dt &\simeq \sum_{j=1}^n w_j \varphi(t_j) \\ &= \frac{1}{3n} (\varphi(t_0) + 4\varphi(t_1) + 2\varphi(t_2) + \dots + 2\varphi(t_{n-2}) + 4\varphi(t_{n-1}) + \varphi(t_n))\end{aligned}$$

donde n par, con $j = 1, \dots, n$.

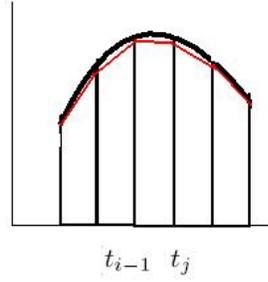


Figura 1.4: Regla del trapecio.

- Regla del trapecio

Si $\varphi(t) \geq 0$, el área del trapecio sobre el j -ésimo subintervalo (ver figura 1.4) es

$$w_j \left(\frac{\varphi(t_{j-1}) + \varphi(t_j)}{2} \right) = \frac{w_j}{2} (\varphi(t_{j-1}) + \varphi(t_j))$$

con $w_j = (b - a)/(2n)$, $t_j = \frac{j}{n}$, $j=1, \dots, n$, entonces

$$\begin{aligned} \int_0^1 \varphi(t) dt &\simeq \sum_{j=1}^n w_j \varphi(t_j) \\ &= \frac{1}{2n} (\varphi(t_0) + 2\varphi(t_1) + 2\varphi(t_2) + \dots + 2\varphi(t_{n-2}) + 2\varphi(t_{n-1}) + \varphi(t_n)) \end{aligned}$$

Considerando la ecuación integral de Fredholm (1.25), esta puede aproximarse mediante.

$$\int_0^1 k(s, t) f(t) dt \approx \sum_{j=1}^n w_j k(s, t_j) \tilde{f}(t_j) = \psi(s)$$

donde $\tilde{f} = f + e$, donde e es un término de error; si $e = 0$, recuperamos la función f original. Para muestras $s_i \in [0, 1]$

$$\psi(s_i) = g(s_i), \quad i = 1, \dots, m$$

Si considerando el caso particular $n = m$ tenemos

$$\int_0^1 k(s, t) f(t) dt \approx \sum_{i=1}^n w_i k(s_i, t_i) \tilde{f}(t_i)$$

donde $i, j = 1, \dots, n$. Matricialmente $Ax = b$, donde $A = a_{i,j}$, $i, j = 1, \dots, n$, $x = (x_1, x_2, \dots, x_n)^T$, con $a_{i,j} = w_j k(s_i, t_j)$, $x_j = \tilde{f}(t_j)$ y $b_i = g(s_i)$. Para obtener $\tilde{f}(t_i)$ a partir de las otras

$$Ax = b \Rightarrow x = A^{-1}b$$

pero este es un problema numérico si A es una matriz mal condicionada. Para ver esto más claramente veamos el siguiente ejemplo.

Ejemplo 1.4.5.

Sea f una función real definida en el intervalo $[0, \infty]$. La *Transformada de Laplace* $\mathfrak{L}(f)$ de f esta definida por la integral

$$\mathfrak{L}(f) = \int_0^{\infty} e^{-st} f(t) dt,$$

siempre y cuando la integral sea convergente similar a la Transformada de Fourier. Dados los valores de la transformada de Laplace en los puntos s_j , $0 < s_1 < s_2 < \dots < s_n < \infty$ es posible estimar la función f . Primero aproximamos la integral que define a la transformada de Laplace por medio de una suma finita

$$\int_0^{\infty} e^{-s_j t} f(t) dt \approx \sum w_k e^{-s_j t_k} f(t_k) = g(s_j),$$

donde los w_k son los pesos y los t_k s son los nodos de la regla de cuadratura trapezoidal. Sean $x_k = f(t_k)$, $g_j = \mathfrak{L}(f(s_j))$ y $a_{jk} = w_k e^{-s_j t_k}$, obteniendo así la matriz.

$$\begin{bmatrix} \mathfrak{L}(s_0) \\ \mathfrak{L}(s_1) \\ \vdots \\ \mathfrak{L}(s_n) \end{bmatrix} = \frac{b-a}{2n} \begin{bmatrix} e^{-s_0 t_0} & 2e^{-s_0 t_1} & \dots & 2e^{-s_0 t_{n-1}} & e^{-s_0 t_n} \\ e^{-s_1 t_0} & 2e^{-s_1 t_1} & \dots & 2e^{-s_1 t_{n-1}} & e^{-s_1 t_n} \\ \vdots & \vdots & & \vdots & \vdots \\ e^{-s_n t_0} & 2e^{-s_n t_1} & \dots & 2e^{-s_n t_{n-1}} & e^{-s_n t_n} \end{bmatrix} \begin{bmatrix} f(t_0) \\ f(t_1) \\ \vdots \\ f(t_n) \end{bmatrix}$$

Escribimos la aproximación numérica de la transformada de Laplace por $Ax = b$, donde A es una matriz cuadrada. Tomamos los datos distribuidos logarítmicamente,

$$\log(s_j) = \left(-1 + \frac{j-1}{20}\right) \log 10, \quad j = 1, 2, \dots, 40$$

$$s_j = 10^{-1 + \frac{j-1}{20}},$$

con el fin de garantizar un muestreo cerca del origen. Usamos la regla de de cuadratura trapezoidal con 40 nodos t_k en el intervalo $[0, 5]$. Por lo que $A \in \mathbb{R}^{40 \times 40}$.

Consideramos la función f dada por

$$f(t) = \begin{cases} t, & \text{si } t \in [0, 1); \\ \frac{3}{2} - \frac{t}{2}, & \text{si } t \in [1, 3); \\ 0, & \text{si } t \geq 3 \end{cases}$$

Calculamos la Transformada de Laplace de $f(t)$ como sigue

$$\begin{aligned}\mathfrak{L}(f) &= \int_0^{\infty} e^{-st} f(t) dt \\ &= \int_0^1 t e^{-st} dt + \int_1^3 \left(\frac{3}{2} - \frac{t}{2} \right) e^{-st} dt + \int_3^{\infty} (0) e^{-st} dt \\ &= \int_0^1 t e^{-st} dt + \int_1^3 \frac{3}{2} e^{-st} dt - \int_1^3 \frac{t}{2} e^{-st} dt,\end{aligned}$$

calculamos cada una de las integrales

$$\begin{aligned}\int_0^1 t e^{-st} dt &= -\frac{t}{s} e^{-st} \Big|_0^1 + \int \frac{1}{s} e^{-st} dt \\ &= -\frac{t}{s} e^{-st} \Big|_0^1 - \frac{t}{s^2} e^{-st} \Big|_0^1 \\ &= -\frac{1}{s} e^{-s} - \frac{1}{s^2} e^{-s} + \frac{1}{s^2}\end{aligned}$$

$$\begin{aligned}\int_1^3 \frac{3}{2} e^{-st} dt &= -\frac{3}{2s} e^{-st} \Big|_1^3 \\ &= -\frac{3}{2s} e^{-3s} + \frac{3}{2s} e^{-s}\end{aligned}$$

$$\begin{aligned}\frac{1}{2} \int_1^3 t e^{-st} dt &= \frac{1}{2} \left(-\frac{t}{s} e^{-st} \Big|_1^3 - \frac{1}{s^2} e^{-st} \Big|_1^3 \right) \\ &= \frac{1}{2} \left(-\frac{3}{s} e^{-3s} + \frac{1}{s} e^{-s} - \frac{1}{s^2} e^{-3s} + \frac{1}{s^2} e^{-s} \right) \\ &= -\frac{3}{2s} e^{-3s} + \frac{1}{2s} e^{-s} - \frac{1}{2s^2} e^{-3s} + \frac{1}{2s^2} e^{-s}\end{aligned}$$

por lo tanto

$$\begin{aligned}\mathfrak{L}(f) &= -\frac{1}{s} e^{-s} - \frac{1}{s^2} e^{-s} + \frac{1}{s^2} - \frac{3}{2s} e^{-3s} + \frac{3}{2s} e^{-s} \\ &+ \frac{3}{2s} e^{-3s} - \frac{1}{s} e^{-s} + \frac{1}{2s^2} e^{-3s} - \frac{1}{2s^2} e^{-s} \\ &= \frac{1}{s^2} + \frac{3}{2s^2} e^{-s} + \frac{1}{2s^2} e^{-3s} \\ &= \frac{1}{2s^2} (2 - 3e^{-s} + e^{-3s})\end{aligned}$$

Estimamos los valores de $x_i = f(t_j)$, resolviendo directamente la ecuación $Ax = b$ en Matlab, obtenemos.

```

>>n=40;
>>a=0;
>>b=10;
>>wk=(b-a)/(n-1);
>>t(1)=a;
>>t(n)=b;
>>t=[1/40:5/40:5]
>>f1=t(1:4);
>>f2=3/2-t(5:12)/2;
>>f3=0*(t(13:40));
>>f=[f1 f2 f3]';
>>for j=1:n
s=10^{-1+(j-1)/20};
A(j,i)=exp(-s*a)*(b-a)/(2*n);
for k=1:n-2
t(1+k)=k*wk;
A(j,k+1)=exp(-s*t(1+k))*(b-a)/n;
end
A(j,n)=exp(-s*b)*(b-a)/(2*n);
L(j,1)=(2-3*exp(-s)+exp(-3*s))/(2*s^2);
end
x=A/L
>>
x = 1.0e + 005*
    0.0000, 0.0000, -0.0000, 0.0002, 0.0096, -0.0422, 0.1411, -0.3499, 0.6080,
    -0.6237, 0.0669, 0.7319, -0.5101, 3.3329, 2.7765, 0.3068, 1.5348, -2.7905,
    4.7254, -5.2853, 2.4783, 0.3305, -2.9519, 1.4057, 2.5496, -2.3716, -2.1814,
    3.6846, -0.9115, 1.0268, -4.9210, -2.4466, -0.4965, 0.8791, -0.3225, 0.0827

```

ver figura 1.5.

Este resultado puede cambiar, aumentando o disminuyendo las dimensiones, si las dimensiones aumentan, teóricamente la solución tendría más nodos, habría posibilidades mayores de obtener una solución mejor. Lamentablemente la matriz A es una matriz mal condicionada (ver apéndice B) y este concepto afecta la solución deseada, los errores se amplifican al calcular la inversa de la matriz A . Para subsanar hasta cierto punto este fenómeno, es necesario aplicar ciertas técnicas que nos permitan obtener una solución *acceptable*.

1.5 Regularización.

1.5.1 Descomposición en valores singulares.

Una herramienta de las más flexibles en algebra lineal es la descomposición en valores singulares (DVS) de una matriz. La DVS esta definida para matrices rectangulares de tamaño $m \times n$. La DVS tiene la forma

$$A = U\Sigma V^T$$

```

1 %Construcción de la matriz A(j,k)=wk*Exp(-Sj.Tk)
2 n=40;
3 a=0;
4 b=10;
5 wk=(b-a)/(n-1);
6 t(1)=a;
7 t(n)=b;
8 t=[1/40:5/40:5];
9 f1=t(1:4);
10 f2=3/2-t(5:12)/2;
11 f3=0*(t(13:40));
12 f=[f1 f2 f3]';
13 for j=1:n
14     s=10^(-1+(j-1)/20);
15     A(j,1)=exp(-s*a)*(b-a)/(2*n);
16     for k=1:n-2
17         t(1+k)=k*wk;
18         A(j,k+1)=exp(-s*t(1+k))*(b-a)/n;
19     end
20     A(j,n)=exp(-s*b)*(b-a)/(2*n);
21     L(j,1)=(2-3*exp(-s)+exp(-3*s))/(2*s^2);
22 end
23 %Solución aproximada por mínimos cuadrados
24 x=A\L;
25 error=x-f

```

Figura 1.5: Transformada de Laplace discreta en Matlab

Las dos matrices U y V consisten de los vectores singulares, esto es

$$U = (u_1, \dots, u_m), \quad V = (v_1, \dots, v_n),$$

ambas matrices son ortogonales, i.e. $U^T U = V^T V = I$. Esto implica que los vectores singulares son ortonormales, $u_i^T u_j = v_i^T v_j = \delta_{ij}$. La matriz $\Sigma = \text{diag}(\sigma_1, \dots, \sigma_n)$ es una matriz diagonal, que consta de elementos σ_i , los cuales son los valores singulares de A , que son no negativos y ordenados decrecientemente, i.e.

$$\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_p \geq 0,$$

donde $p = \min(m, n)$ y

$$A = \sum_{i=1}^r \sigma_i u_i v_i^T$$

si r es el número de valores singulares diferentes de cero, entonces r es igual al rango de la matriz A , el número de condición de A en la norma-2 (ver apéndice B) esta definido en terminos de DVS como

$$\text{cond}(A) = \|A\|_2 \|A^{-1}\|_2 = \frac{\sigma_1}{\sigma_n}.$$

Si A es singular, i.e $\text{rank}(A) < n$, entonces $\sigma_n = 0$ y $\text{cond}(A) = \infty$. Si A es matriz simétrica, $A = A^T$, entonces la DVS de A se relaciona como la

descomposición de eigenvalores $A = W\Lambda W^T$, con $W = (w_1, \dots, w_n)$ y $\Lambda = \text{diag}(\lambda_1, \dots, \lambda_n)$ como

$$(u_i, \sigma_i, v_i) = \begin{cases} (w_i, \lambda_i, w_i) & \text{para } \lambda_i \geq 0 \\ (w_i, -\lambda_i, -w_i) & \text{para } \lambda_i < 0 \end{cases}$$

esta relación es usada para simplificar el cálculo de la DVS para matrices simétricas. Notemos que podemos escribir a b y x , en terminos de los valores singulares u_i, v_i , respectivamente como

$$b = \sum_{i=1}^n (u_i^T b) u_i, \quad x = \sum_{i=1}^n (v_i^T x) v_i, \quad Ax = \sum_{i=1}^n \sigma_i (v_i^T x) v_i,$$

igualando $b = Ax$, llegamos a que $(u_i^T b) = \sigma_i (v_i^T x)$, entonces $v_i^T x = u_i^T b / \sigma_i$, para $i = 1, \dots, n$. Entonces la solución *naive* a $Ax = b$ se escribe como

$$x_{naive} = \sum_{i=1}^n \frac{u_i^T b}{\sigma_i} v_i$$

1.5.2 Analisis de DVS.

Para matrices que presentan la discretización de la ecuación integral de Fredholm (1.27) podemos decir lo siguiente en base al Ejemplo 1.4.2.

1. Los valores singulares de A decrecen gradualmente a cero (**eps** de la máquina, si existiera precisión infinita decrece hasta el cero)
2. El número de condición $\text{cond}(A) = \sigma_1 / \sigma_n$ es aproximadamente el recíproco de la precisión de maquina $1/\text{eps}$.
3. Típicamente los valores singulares siguen una progresión, como pueden ser
 - Progresión armónica $\sigma_i \approx 1/i^a$, $a > 0$
 - Progresión geométrica $\sigma_i \approx e^{-ai}$, $a > 0$ con $i = 1, \dots, n$

Para ilustrar esto, calculemos la DVS de A del Ejemplo 1.4.2, con $n = 40$. Los valores singulares de A se muestran en la Figura 1.6, vemos ahí que el número de valores singulares mayores a la precisión de la maquina es 25, los restantes son cero para la maquina (**eps**).

1.5.3 Método de Regularización Directa.

La solución numérica de un sistema de ecuaciones lineales de la forma

$$Ax = b \tag{1.27}$$

puede ser analizada en términos de la descomposición en valores singulares de la matriz A .

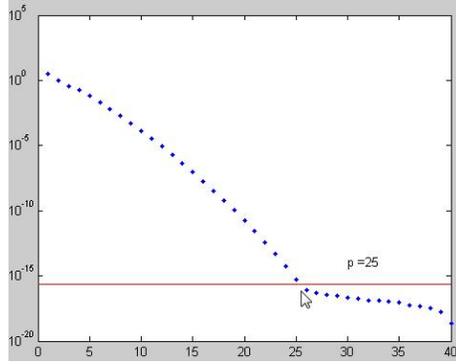


Figura 1.6: DVS para el Ejemplo 1.4.2 con $n=40$

Dada $A \in \mathbb{R}^{m \times n}$, $n \leq m$, existen $U \in \mathbb{R}^{m \times m}$ y $V \in \mathbb{R}^{n \times n}$ ortogonales, tales que

$$\begin{aligned} A &= U \Sigma V^T \\ &= \sum_{i=1}^n \sigma_i \vec{u}_i \vec{v}_i^T \end{aligned}$$

donde $U = [\vec{u}_1, \dots, \vec{u}_m]$, $V = [\vec{v}_1, \dots, \vec{v}_n]$, $\Sigma = \text{diag}[\sigma_1, \dots, \sigma_n]$ tales que $U^T U = I_m$ y $V^T V = I_n$, además $\sigma_1 \geq \dots \geq \sigma_r > \sigma_{r+1} = \dots = \sigma_n = 0$, ó también

$$\Sigma^\dagger = \left(\frac{1}{\sigma_1}, \frac{1}{\sigma_2}, \dots, \frac{1}{\sigma_r}, \dots, 0, \dots, 0 \right)$$

además recordemos que el producto exterior $\vec{u}_i \vec{v}_i^T$ está dado por

$$\begin{aligned} \vec{u}_1 &= \begin{pmatrix} u_{11} \\ \vdots \\ u_{m1} \end{pmatrix} \in \mathbb{R}^n, \\ \vec{v}_1 &= \begin{pmatrix} v_{11} \\ \vdots \\ v_{n1} \end{pmatrix} \in \mathbb{R}^m, \\ \vec{u}_i \vec{v}_i^T &= \begin{pmatrix} u_{11} \\ \vdots \\ u_{m1} \end{pmatrix} \begin{pmatrix} v_{11} & \dots & v_{n1} \end{pmatrix} \\ &= \begin{pmatrix} u_{11}v_{11} & \dots & u_{11}v_{n1} \\ \vdots & \ddots & \vdots \\ u_{m1}v_{11} & \dots & u_{m1}v_{n1} \end{pmatrix} \end{aligned}$$

El producto exterior de los vectores $\vec{u}_i \vec{v}_i^T$ genera matrices de tamaño $m \times n$. Vamos a usar estas matrices para representar a la matriz A , además de la matriz $\Sigma^\dagger = \text{diag}[1/\sigma_1, \dots, 1/\sigma_r, 0, \dots, 0]$ la cual se conoce como la inversa generalizada de Moore-Penrose de Σ . Con esto se define la inversa generalizada de A mediante:

$$\begin{aligned} A &= V \Sigma^\dagger U^T \\ &= \sum_{i=1}^r \frac{1}{\sigma_i} \vec{v}_i \vec{u}_i^T \end{aligned}$$

Hasta ahora podemos tener en cuenta dos sistemas; la ecuación (1.27) y el sistema

$$A^\dagger b = x^\dagger = \sum_{i=1}^r \frac{\langle \vec{u}_i, b \rangle}{\sigma_i} \vec{v}_i$$

que es la solución generalizada Moore-Penrose, también puede ser expresada mediante la ecuación

$$x^\dagger = A^\dagger b = \sum_{i=1}^r \left(\frac{\vec{u}_i^T b}{\sigma_i} \right) \vec{v}_i \quad (1.28)$$

Si la matriz es mal condicionada, entonces σ_i decrece rápidamente a cero conforme crece i ; a partir de (1.28) es fácil ver que a pequeñas perturbaciones de b pueden generar grandes variaciones en la solución x . El método de regularización directa consiste en truncar la expresión (1.28) en k términos, con $k \leq r$, $r = \text{rango}(A)$ conocido como *Truncamiento de la Descomposición de Valores Singulares*, esto es, se considera:

$$x^r = \sum_{i=1}^k \left(\frac{u_i^T b}{\sigma_i} \right) v_i,$$

notemos que x^r es muy parecida a la x_{naive} de la sección anterior, la diferencia es que le faltan los $r - k$ términos de x_{naive} .

1.5.4 Regularización de Tikhonov.

Un método, el cuál ha tenido una gran aceptación en los últimos años, es el método *Regularización de Tikhonov*. Esto se debe a su mayor generalidad en comparación con la regularización directa. Esta regularización nos permite hacer que x^r sea más estable.

Para hacer x^r más estable, la reemplazamos por

$$x_\alpha^\delta := \sum_{i=1}^r \frac{\sigma_i}{\sigma_i^2 + \alpha} \langle b_\delta, u_i \rangle v_i \quad (1.29)$$

con b_δ la información perturbada por algún $\delta > 0$ y $\alpha > 0$. Este es el famoso método de regularización de Tikhonov. Sea x_α^δ , entonces por la propiedad de

ortonormalidad de la v_i tenemos para toda $i \in \{1, \dots, r\}$:

$$\langle x_\alpha^\delta, v_j \rangle = \frac{\sigma_j}{\sigma_j^2 + \alpha} \langle b_\delta, u_j \rangle,$$

para toda $j \in \{1, \dots, r\}$, entonces

$$\sigma_j^2 \langle x_\alpha^\delta, v_j \rangle + \alpha \langle x_\alpha^\delta, v_j \rangle = \sigma_j \langle b_\delta, u_j \rangle.$$

Notemos que

$$\begin{aligned} A\vec{v}_i &= \left(\sum_{i=1}^n \sigma_i \vec{u}_i \vec{v}_i^T \right) \vec{v}_i \\ &= \sigma_i \vec{u}_i \end{aligned}$$

por la ortogonalidad de \vec{v}_i , de forma análoga tenemos que

$$\begin{aligned} A^T \vec{u}_i &= \left(\sum_{i=1}^n \sigma_i \vec{v}_i \vec{u}_i^T \right) \vec{u}_i \\ &= \sigma_i \vec{v}_i \end{aligned}$$

Con estas observaciones podemos notar que

$$\begin{aligned} Ax &= \sum_{i=1}^r \langle Ax, \vec{u}_i \rangle \vec{u}_i \\ &= \sum_{i=1}^r \langle x, A^T \vec{u}_i \rangle \vec{u}_i \\ &= \sum_{i=1}^r \langle x, \sigma_i \vec{v}_i \rangle \vec{u}_i \\ &= \sum_{i=1}^r \sigma_i \langle x, \vec{v}_i \rangle \vec{u}_i \end{aligned}$$

por tanto

$$Ax = \sum_{i=1}^r \sigma_i \langle x, \vec{v}_i \rangle \vec{u}_i$$

usando la misma idea para el sistema $A^T b_\delta$, donde $\delta > 0$, se tiene que

$$\begin{aligned} A^T b_\delta &= \sum_{i=1}^r \langle A^T b_\delta, \vec{v}_i \rangle \vec{v}_i \\ &= \sum_{i=1}^r \langle b_\delta, A\vec{v}_i \rangle \vec{v}_i \\ &= \sum_{i=1}^r \sigma_i \langle b_\delta, \vec{u}_i \rangle \vec{v}_i \end{aligned}$$

por tanto

$$A^T b_\delta = \sum_{i=1}^r \sigma_i \langle b_\delta, \vec{u}_i \rangle \vec{v}_i$$

entonces

$$\begin{aligned} \sum_{i=1}^r [\sigma_i^2 \langle x_\alpha^\delta, \vec{v}_i \rangle + \alpha \langle x_\alpha^\delta, \vec{v}_i \rangle] \vec{v}_i &= \sum_{i=1}^r [\langle x_\alpha^\delta, A^T A \vec{v}_i \rangle + \alpha \langle x_\alpha^\delta, \vec{v}_i \rangle] \vec{v}_i \\ &= A^T A x_\alpha^\delta + \alpha I x_\alpha^\delta \end{aligned}$$

Por lo tanto

$$A^T b_\delta = (A^T A + \alpha I) x_\alpha^\delta \quad (1.30)$$

y

$$\begin{aligned} A^T A x_\alpha^\delta + \alpha I x_\alpha^\delta &= A^T b_\delta \\ \Leftrightarrow (A^T A + \alpha I) x_\alpha^\delta &= A^T b_\delta \\ \Leftrightarrow x_\alpha^\delta &= (A^T A + \alpha I)^{-1} A^T b_\delta \end{aligned}$$

Que es la solución obtenida método de regularización de Tikhonov.

De esta última expresión podemos encontrar otra caracterización, llamada el funcional de Tikhonov.

$$x \rightarrow \|Ax - b_\delta\|^2 + \alpha \|x\|^2, \quad (1.31)$$

la cual, si derivamos e igualamos a cero este funcional obtenemos la solución ya vista del método de regularización de Tikhonov.

Sea

$$J(x) = \|Ax - b_\delta\|^2 + \alpha \|x\|^2$$

el funcional de Tikhonov, o bien

$$J(x) = (Ax - b_\delta)^T (Ax - b_\delta) + \alpha x^T x.$$

Derivando este funcional respecto a x

$$\begin{aligned} \frac{\partial J(x)}{\partial x} &= A^T (Ax - b_\delta) + \alpha I x = 0 \\ \Leftrightarrow A^T A x - A^T b_\delta + \alpha I x &= 0 \\ \Leftrightarrow A^T A x + \alpha I x &= A^T b_\delta \\ \Leftrightarrow (A^T A + \alpha I) x &= A^T b_\delta \\ \Leftrightarrow x &= (A^T A + \alpha I)^{-1} A^T b_\delta. \end{aligned}$$

Esta es la solución ya vista del método de regularización de Tikhonov, por medio de una minimización de (1.31). Por otro lado, si $\alpha = 0$ en (1.31) obtenemos que $x := A^\dagger b$ que es la solución de norma mínima de la ecuación normal

$$A^T A x = A^T b.$$

En el capítulo tres abordaremos de lleno el problema mal planteado (en el sentido de Hadamard) llamado **Deconvolución Ciega**. Donde toda, ó la mayor parte de la teoría vista en este primer capítulo, es utilizada.

Capítulo 2

Nociones de Probabilidad.

2.1 Probabilidad y Teorema de Bayes.

Las nociones fundamentales de probabilidad que usaremos son básicas; variables aleatorias y distribución de probabilidad. En este trabajo no profundizaremos mucho en los axiomas de la teoría de probabilidad, sin embargo para su aplicación se necesita cierta comprensión de estos conceptos (se recomienda ver las referencias [26] y [27]).

2.1.1 Conceptos básicos.

Definición 2.1.1. Sean Ω un espacio abstracto (al que llamaremos espacio muestral) y \mathcal{F} una colección de los subconjuntos de Ω . Diremos que \mathcal{F} es una σ -álgebra en Ω si cumple con las siguientes condiciones

1. $\Omega \in \mathcal{F}$
2. Si $A_n \in \mathcal{F}$, $n \in \mathbb{N}$, entonces $\bigcup_{n=1}^{\infty} A_n \in \mathcal{F}$.
3. Si $A \in \mathcal{F}$, entonces $A^C \in \mathcal{F}$.

Ejemplo 2.1.2. Si X es cualquier conjunto, la familia $\{\emptyset, X\}$ es una σ -álgebra sobre X , se conoce como σ -álgebra trivial por obvias razones.

Definición 2.1.3. Sea \mathcal{F} una σ -álgebra sobre un espacio muestral Ω . Una función $\mu : \mathcal{F} \rightarrow \mathbb{R}$ se le llama medida si cumple con las siguientes condiciones:

1. $\mu(A) \geq 0 \forall A \in \mathcal{F}$.
2. $\mu(\emptyset) = 0$.
3. Si los conjuntos $A_i \in \mathcal{F}$ son disjuntos, $i \in \mathbb{N}$, (i.e., $A_i \cap A_j = \emptyset$, siempre que $i \neq j$), entonces

$$\mu\left(\bigcup_{i=1}^{\infty} A_i\right) = \sum_{i=1}^{\infty} \mu(A_i). \quad (\sigma\text{-aditividad})$$

Definición 2.1.4. Sea \mathcal{F} una σ -álgebra y $\mu : \mathcal{F} \rightarrow \mathbb{R}$ una medida, diremos que \mathcal{F} es llamada completa si contiene todos los conjuntos de medida cero, i.e. si $A \subset B$, y $B \in \mathcal{F}$ con $\mu(B) = 0$, entonces $A \in \mathcal{F}$.

Definición 2.1.5. Una medida μ se le llama finita si $\mu(\Omega) < \infty$. La medida es llamada σ -finita, si existe una colección de conjuntos $A_n \in \mathcal{F}$, $n \in \mathbb{N}$, tales que

1. $\Omega = \bigcup_{n=1}^{\infty} A_n$,
2. $\mu(A_n) < \infty$ para todo n .

Definición 2.1.6. A una medida que cumple con la propiedad de que $\mu(\Omega) = 1$ se le llama medida de probabilidad.

Notación: Una medida de probabilidad se denotará por P .

Definición 2.1.7. La terna (Ω, \mathcal{F}, P) es llamada espacio de probabilidad. El espacio abstracto Ω es llamado espacio muestral, mientras que la σ -álgebra \mathcal{F} la nombraremos el conjunto de eventos, y $P(A)$ es la probabilidad de un evento $A \in \mathcal{F}$.

Un concepto central en la teoría de probabilidad es el de independencia.

Definición 2.1.8. Sea (Ω, \mathcal{F}, P) un espacio de probabilidad. Diremos que dos eventos $A, B \in \mathcal{F}$ son independientes si

$$P(A \cap B) = P(A)P(B).$$

En general, una familia de eventos $\{A_i \mid i \in T\} \subset \mathcal{F}$ (con T un conjunto finito de índices) es independiente si

$$P\left(\bigcap_{i=1}^m A_i\right) = \prod_{i=1}^m P(A_i).$$

Definición 2.1.9. Sea (Ω, \mathcal{F}, P) un espacio de probabilidad. Una variable aleatoria en \mathbb{R}^n es una función

$$X : \Omega \rightarrow \mathbb{R}^n,$$

tal que para cada abierto $B \subset \mathbb{R}^n$, $X^{-1} \in \mathcal{F}$, observemos que X es una función medible i.e., $X^{-1}(B) \in \mathcal{F}$ para todo abierto B en la topología⁴ \mathbb{R}^n .

Ejemplo 2.1.10. Para ilustrar la noción de variable aleatoria, considérese el lanzamiento de una moneda. El espacio muestral Ω está constituido por dos posibles resultados de lanzar una moneda, sol y águila. Sea $X(\text{sol}) = 0$ y $X(\text{águila}) = 1$ de esta forma se han transformado los dos posibles resultados del espacio muestral en puntos sobre la recta de los reales.

⁴Sea X un conjunto cualquiera y $P(X)$ el conjunto de sus partes. Una topología sobre X es un conjunto $T \subset P(X)$ que cumpla que $X \in T$, $\emptyset \in T$; si $A, B \in T$ entonces $A \cap B \in T$, y que si $S \subset T$ entonces $\bigcup_{G \in S} G \in T$. A los elementos de T se les denomina conjuntos abiertos. Un espacio topológico es un par ordenado (X, T) donde X es el conjunto y T es una colección de subconjuntos de X tal que cumpla con las propiedades ya mencionadas.

Definición 2.1.11. Una variable aleatoria, la cual denotaremos simplemente por $X \in \mathbb{R}^n$, genera una medida de probabilidad μ_X en \mathbb{R}^n (equipado con la σ -álgebra de borel⁵ \mathcal{B}), donde μ_X se define como

$$\mu_X(B) = P(X^{-1}(B)), \quad B \in \mathcal{B}.$$

Definición 2.1.12. Sea (Ω, \mathcal{F}, P) un espacio de probabilidad y X una variable aleatoria. La medida P_X sobre $(\mathbb{R}, \mathcal{B})$ definida por $P_X(B) := P(X \in B)$ con $B \in \mathcal{B}$ se llama distribución de la variable aleatoria X .

Definición 2.1.13. La función de distribución de una variable aleatoria X es la función $F : \mathbb{R} \rightarrow [0, 1]$ dada por $F(x) = P(X \leq x)$.

Definición 2.1.14. Un proceso estocástico, es una familia de variables aleatorias $X = \{X_t, t \in T\}$ definidas en un espacio de probabilidad (Ω, \mathcal{F}, P) . El conjunto T es llamado dominio de definición del proceso.

Si $T = \{t_0, t_1, \dots\}$, el proceso estocástico se llama en tiempo discreto, y si $t \in [0, T] \subseteq \mathbb{R}$, se denomina proceso estocástico en tiempo continuo. Para aclarar un poco más este concepto consideremos la función

$$X : \mathbb{R} \times \Omega \rightarrow \mathbb{R},$$

formamos con esto una familia de funciones en el tiempo, una función $X(t)$ para cada $\omega \in \Omega$. Un proceso estocástico puede ser visto como una función de dos variables $t \in T$ (números reales) y $\omega \in \Omega$ (espacio muestral). Para ω_0 con $\omega_0 \in \Omega$ fijo, la expresión $X(t, \omega_0)$ se convierte en una función que depende del tiempo. Para un tiempo específico $t_0 \in T$ fijo, la expresión $X(t_0, \omega)$ es una cantidad que depende de ω , la cual es una variable aleatoria. Finalmente $X(t_0, \omega_0)$ es un número real. En el presente trabajo usaremos la notación

$$X_t,$$

para representar un proceso estocástico, omitiendo usualmente su independencia con ω . De lo anterior se tiene que son cuatro los objetos que representa X_t :

1. Una familia de funciones en el tiempo (t y ω variables).
2. Una sola función en el tiempo (t variable, ω fija).
3. Una variable aleatoria (t fija, ω variable).
4. Un escalar (t fija, ω fija).

Exigiremos que para cualquier t , X_t sea una variable aleatoria, sólo entonces la familia de funciones es llamada un proceso estocástico. Si fijamos ω_0 los procesos son tales que

$$X_t = Y_t \quad \Leftrightarrow \quad X(t, \omega_0) = Y(t, \omega_0),$$

similarmente se definen las operaciones $X_t + Y_t$, $X_t Y_t$ o cualquier otra operación (por ejemplo la derivada), que involucran uno o más procesos.

⁵la σ -álgebra de Borel, o boreliana, sobre un espacio topológico es la σ -álgebra generada por el conjunto de conjuntos abiertos (o equivalentemente, el conjunto de conjuntos cerrados).

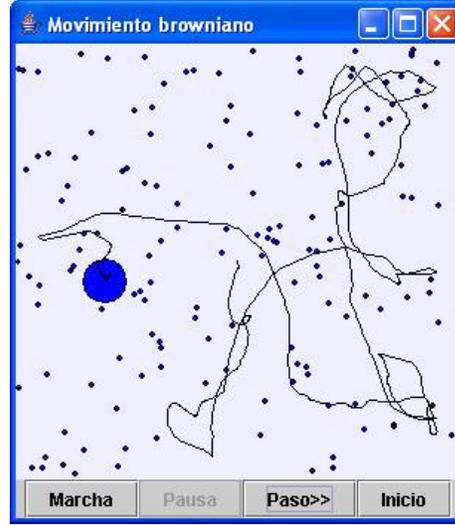


Figura 2.1: Movimiento Browniano.

Definición 2.1.15. Sean $X = \{X_t, t \in T\}$ e $Y = \{Y_t, t \in T\}$ dos procesos estocásticos definidos sobre el mismo espacio de probabilidad (Ω, \mathcal{F}, P) , decimos:

1. X e Y son iguales si $X_t(\omega) = Y_t(\omega)$ para todos $t \in T$ y $\omega \in \Omega$.
2. X e Y son equivalentes si $P(X_t = Y_t) = 1$ para todo $t \in T$.
3. X e Y se dicen indistinguibles si casi todas sus trayectorias coinciden, esto es $P(X_t = Y_t, t \in T) = 1$.

En general las funciones de un proceso estocástico son complicadas. Supongamos, por ejemplo, que X_t representa el movimiento de una partícula que está sometida a un gran número de choques con otras partículas (movimiento Browniano). Tomando una de estas partículas y denotando por X_t su comportamiento, este se vería como en la Figura 2.1, irregular y no puede ser descrita por una fórmula o expresión determinista. Más aun, si tomamos una partícula X_t y conocemos su comportamiento al tiempo $t < t_0$, con $t_0 \in T$ fijo, no podemos predecir su valor futuro al tiempo $t > t_0$. Sin embargo no todos los procesos tienen esas propiedades. Mencionemos un ejemplo simple: definamos un experimento X_t tal que

$$X_t = \sin(t) \quad \text{si} \quad \omega = \omega_0 \quad X_t = 2t \quad \text{si} \quad \omega = \omega_1.$$

Entonces X_t consiste de dos curvas muy regulares. Esto es, sin embargo, un proceso estocástico.

Un proceso estocástico puede ser estacionario o no estacionario. Un proceso estacionario (o proceso estrictamente estacionario) es un proceso estocástico

cuya distribución de probabilidad en un instante de tiempo fijo o una posición fija es la misma para todos los instantes de tiempo o posiciones. En consecuencia, parámetros tales como la media y la varianza, si existen, no varían a lo largo del tiempo o la posición.

Definición 2.1.16. (*Proceso estacionario*) Un proceso estocástico X_t se dice estacionario si su función de distribución (definición 2.1.13) para cada t fijo no es afectado por el tiempo, i.e. que los dos procesos X_t y $X_{t+\varepsilon}$ tienen la misma función distribución para cualquier ε .

Ejemplo 2.1.17.

Por ejemplo, el *ruido blanco* es estacionario este tipo de ruido lo veremos con más detalle más adelante. Sin embargo, el sonido de un golpe de platillos no es estacionario, pues la energía acústica del golpe (y por lo tanto su varianza) disminuye con el tiempo.

Definición 2.1.18. Un vector aleatorio $\mathbf{X} = (X_1, \dots, X_n)$ es una arreglo de las variables aleatorias X_1, \dots, X_n .

Definición 2.1.19. La función de distribución conjunta $F: \mathbb{R}^n \rightarrow [0, 1]$ de un vector aleatorio $\mathbf{X} = (X_1, \dots, X_n) \in \mathbb{R}^n$ está definida como una función de la forma

$$F(\mathbf{x}) = P(X_1 \leq x_1, \dots, X_n \leq x_n), \quad \mathbf{x} = (x_1, \dots, x_n).$$

Definición 2.1.20. La esperanza de una variable aleatoria X está definida como

$$E\{X\} = \int_{\Omega} X(\omega) dP(\omega) = \int_{\mathbb{R}^n} x d\mu_X(x),$$

siempre que las integrales converjan. Si X es variable aleatoria discreta, entonces

$$E\{X\} = \sum_n x_n P\{X = x_n\} = \sum_n x_n p_n,$$

Definición 2.1.21. (*Distribución de Poisson.*)

Una variable aleatoria X , sigue una distribución de Poisson con parámetro $\lambda > 0$ y distribución de probabilidad

$$p(x; \lambda) = P\{X = x\} = \frac{\lambda^x e^{-\lambda}}{x!}, \quad x = 0, 1, 2, \dots$$

En general la descripción de un proceso de Poisson no es necesariamente sencilla, en [2] puede encontrarse una discusión simplificada.

Ejemplo 2.1.22. Dado que $p(x; \lambda) \geq 0$ y recurriendo a la expresión

$$e^x = \sum_{i=0}^{\infty} \frac{x^i}{i!},$$

se obtiene que

$$\begin{aligned}\sum_{i=0}^{\infty} p(i; \lambda) &= \sum_{i=0}^{\infty} \frac{\lambda^i e^{-\lambda}}{i!} \\ &= e^{-\lambda} \sum_{i=0}^{\infty} \frac{\lambda^i}{i!} \\ &= e^{-\lambda} e^{\lambda} = 1,\end{aligned}$$

con la cual $p(x; \lambda) = P[X = x]$ cumple con las propiedades para ser una distribución de probabilidad (Definición 2.1.11). Para una distribución de Poisson se tiene que la esperanza está dada por

$$\begin{aligned}E\{X\} &= \sum_{i=0}^{\infty} iP(X = i) \\ &= \sum_{i=0}^{\infty} ip(i, \lambda) \\ &= \sum_{i=0}^{\infty} i \frac{\lambda^i}{i!} e^{-\lambda} \\ &= e^{-\lambda} \sum_{i=1}^{\infty} \frac{\lambda^i}{(i-1)!} \\ &= e^{-\lambda} \sum_{i=0}^{\infty} \frac{\lambda^{i+1}}{i!} \\ &= e^{-\lambda} \lambda \sum_{i=0}^{\infty} \frac{\lambda^i}{i!} \\ &= e^{-\lambda} \lambda e^{\lambda} \\ &= \lambda,\end{aligned}$$

por tanto $E\{X\} = \lambda$. □

Definición 2.1.23. Sea (Ω, \mathcal{F}, P) un espacio de probabilidad y sean $A, B \in \mathcal{F}$ dos eventos, $P(B) > 0$. Definimos la probabilidad condicional de A dado B por

$$P(A | B) = \frac{P(A \cap B)}{P(B)} \quad (2.1)$$

Suponiendo que el evento B ocurre.

Observación: Claramente si A y B son disjuntos (i.e. $A \cap B = \emptyset$), entonces $P(A|B) = 0$ pues $P(\emptyset) = 0$. Por otro lado si $A \subset B$ tenemos que $A \cap B = A$ y por la definición anterior; $P(A|B) = \frac{P(A)}{P(B)} \geq P(A)$. Análogamente si $B \subset A$, entonces $A \cap B = B$ y $P(A|B) = \frac{P(B)}{P(B)} = 1$.

Ejemplo 2.1.24. Sean $B = \{x_2, x_4, x_6\}$, $A = \{x_2\}$, es claro que $A \cap B = \{x_2\}$, $A \subset B$, la probabilidad de tomar un elemento de $\Omega = \{x_1, x_2, \dots, x_6\}$ (probabilidad a priori) $P(x_i) = 1/6$ para todo $i = 1, \dots, 6$. Aplicando la definición anterior

$$\begin{aligned} P(A | B) &= \frac{P(A \cap B)}{P(B)} \\ &= \frac{P(A)}{P(B)} \\ &= \frac{P(A)}{P(\{x_2\} \cup \{x_4\} \cup \{x_6\})} \\ &= \frac{P(\{x_2\})}{P(\{x_2\}) + P(\{x_4\}) + P(\{x_6\})} \\ &= \frac{1/6}{3/6} = 1/3. \end{aligned}$$

Lema 2.1.25. Sean (Ω, \mathcal{F}, P) un espacio de probabilidad, $A_i \in \mathcal{F}$ una familia de eventos mutuamente ajenos y supongamos que $\bigcup_{i=1}^n A_i = S$. Si $B \subset S$ entonces

$$P(B) = \sum_{i=1}^n P(B | A_i)P(A_i) \quad (2.2)$$

Demostración:

Como $B \subset S \Rightarrow B \cap S = B$, $\Rightarrow B \cap (\bigcup_{i=1}^n A_i) = \bigcup_{i=1}^n (B \cap A_i) = B$ (ver [3]), entonces

$$P(B) = P\left(\bigcup_{i=1}^n (B \cap A_i)\right) = \sum_{i=1}^n P(B \cap A_i), \quad (2.3)$$

esto se sigue de la definición 2.1.3. Ahora de (2.1) tenemos que $P(B \cap A_i) = P(B | A_i)P(A_i)$, aplicando esta expresión en la ecuación (2.3) se tiene que

$$\begin{aligned} P(B) &= \sum_{i=1}^n P(B \cap A_i) \\ &= \sum_{i=1}^n P(B | A_i)P(A_i), \end{aligned}$$

por lo tanto (2.2) se satisface. □

A continuación demostraremos un resultado que será fundamental en el desarrollo de este trabajo, puesto que es la base del algoritmo *EM* (Esperanza-Maximización).

Teorema 2.1.26. (Bayes)

Sean (Ω, \mathcal{F}, P) un espacio de probabilidad, $A_i \in \mathcal{F}$ $i = 1, \dots, n$ mutuamente ajenos, $S = \bigcup_{i=1}^n A_i$ y $B \subset S$, entonces

$$P(A_i | B) = \frac{P(B | A_i)P(A_i)}{\sum_{i=1}^n P(B | A_i)P(A_i)} \quad (2.4)$$

Demostración:

Observemos que $P(A_i \cap B) = P(B \cap A_i)$, por (2.1) tenemos que $P(A_i \cap B) = P(A_i | B)P(B)$ y $P(B \cap A_i) = P(B | A_i)P(A_i)$, por tanto $P(B | A_i)P(A_i) = P(A_i | B)P(B)$, entonces

$$\begin{aligned} P(A_i | B) &= \frac{P(B | A_i)P(A_i)}{P(B)} \\ &= \frac{P(B | A_i)P(A_i)}{\sum_{i=1}^n P(B | A_i)P(A_i)}, \end{aligned}$$

la segunda igualdad se sigue de (2.2).

□

Ejemplo 2.1.27.

Una caja contiene 2000 aparatos, de los cuales el 100 están defectuosos, Una segunda caja contiene 500 aparatos, de los cuales 200 están defectuosos. Otras dos cajas tienen 1000 aparatos de los cuales el diez por ciento de cada caja esta defectuoso. Si se selecciona aleatoriamente una caja y sacamos aleatoriamente un aparato.

a) ¿Cual es la probabilidad de que saquemos una caja defectuosa?. Sean A_i la i -ésima caja, D el evento de los 500 aparatos defectuosos. Queremos encontrar $P(D)$.

Claramente $\bigcup_{i=1}^4 A_i = S$, donde A_i son mutuamente ajenos para todo $i = 1, \dots, 4$. La probabilidad de seleccionar una caja esta dada por $P(A_1) = P(A_2) = P(A_3) = P(A_4) = 1/4$. La probabilidad de seleccionar un aparato defectuoso en cada caja viene dado por

$$\begin{aligned} P(D | A_1) &= \frac{100}{2000} = 0.05 \\ P(D | A_2) &= \frac{200}{500} = 0.4 \\ P(D | A_3) &= \frac{100}{1000} = 0.1 \\ P(D | A_4) &= \frac{100}{1000} = 0.1 \end{aligned}$$

Por (2) $P(D) = [0.05 + 0.4 + 0.2]1/4 = 0.1625$

- b) En base en el análisis queremos determinar la probabilidad de que el aparato defectuoso venga de la caja 2, esto es, queremos calcular la probabilidad condicional $P(A_2 | D)$ como $P(D) = 0.1625$, $P(D | A_2) = 0.4$ y $P(A_2) = 1/4$. Entonces aplicando el teorema de Bayes

$$P(A_2 | D) = \frac{P(D | A_2)P(A_2)}{P(D)} = \frac{(0.4)(0.25)}{0.1625} \approx 0.615.$$

□

2.2 Correlación y Espectro de Potencia de un Proceso Estocástico.

En esta sección discutiremos las propiedades de la correlación $R(\tau)$ y el espectro de potencia $S(\omega)$ de un proceso estocástico. El espectro está definido como la Transformada de Fourier de $R(\tau)$. En esta sección usaremos los conceptos de sistemas lineales y transformadas.

2.2.1 Correlación.

Definición 2.2.1. Sea X_t un proceso estocástico, se define su media como

$$\eta = E[X_t] = \eta_X,$$

la versión discreta de la correlación se define como

$$\eta_X = \lim_{N \rightarrow \infty} \left\{ \frac{1}{N} \sum_{i=1}^N X_{i,t} \right\},$$

donde N es el tamaño de la muestra, i.e. $X_{i,t}$ valor muestral con índice i al tiempo t .

Definición 2.2.2. El segundo momento conjunto de dos procesos X_t y Y_t , se define como

$$R_{XY}(\tau) = E[X_{t+\tau}Y_t^*],$$

es su relación cruzada.

Si $X_t = Y_t$, la correlación es conocida como la autocorrelación. Y la autocorrelación tiene la siguiente ecuación

$$\begin{aligned} R(\tau) &= E[X_{t+\tau}X_t^*] \\ &= R(\tau)_{XX} \\ &= R(\tau)_{\ddot{X}} \end{aligned}$$

2.2.2 Covarianza.

Definición 2.2.3. La relación de covarianza esta definida para un proceso estocástico $\{X_t\}$ mediante

$$\begin{aligned} C(\tau) &= E[(X_{t+\tau} - \eta)(X_t^* - \eta^*)] \\ &= R(\tau) - |\eta|^2 \end{aligned}$$

y la covarianza cruzada para los procesos estocástica $\{X_t\}$ y $\{Y_t\}$ como

$$\begin{aligned} C(\tau)_{XY} &= E[(X_{t+\tau} - \eta_X)(Y_t^* - \eta_Y^*)] \\ &= R_{XY}(\tau) - \eta_X \eta_Y^*. \end{aligned}$$

La varianza es obtenida a partir de la definición de covarianza

$$C(\tau) = \sigma_X^2 = E\{(X_{t+\tau} - \eta_X)^2\} = E\{X_{t+\tau}X_t^*\} - \eta_X^2.$$

Si el valor η_X es cero, entonces la varianza y la correlación son idénticas,

$$C(\tau) = \sigma_X^2 = E\{X_{t+\tau}X_t^*\} = R(\tau)$$

2.2.3 Independencia.

Si la función de distribución de dos variables aleatorias puede separarse como el producto de dos funciones de distribución, ie. $P(X, Y) = P(X)P(Y)$ (ver definición 2.1.8), entonces las variables aleatorias son independientes. En consecuencia,

$$E\{X_{t+\tau}X_t\} = E\{X_{t+\tau}\}E\{X_t\} = \eta_{X_{t+\tau}}\eta_{X_t}$$

La condición anterior es necesaria y suficiente para que las dos variables aleatorias $X_{t+\tau}$ y X_t no esten correlacionadas. Sin embargo la condición inversa no necesariamente es cierta. Si el valor medio de cualesquiera dos variables aleatorias no correlacionadas es cero, entonces las variables son llamadas *ortogonales*. En general, dos variables aleatorias son llamadas *ortogonales* si su correlación es cero.

La correlación del producto $X_t Y_t$ no puede ser expresada en forma general en términos de momentos de segundo orden del proceso dado. Sin embargo, si X_t y Y_t son independientes, entonces los procesos $X_{t+\tau}$ y X_t son independientes de $Y_{t+\tau}$ y Y_t ; por lo tanto

$$E[(X_{t+\tau}Y_{t+\tau})(X_t^*Y_t^*)] = E[X_{t+\tau}X_t^*]E[Y_{t+\tau}Y_t^*].$$

En consecuencia, si W_t es el producto de dos procesos X_t y Y_t , ie $W_t = X_t Y_t$ es un proceso, el cual es expresado como

$$R_{WW}(\tau) = R_{XX}(\tau)R_{YY}(\tau).$$

2.2.4 Matriz de Autocorrelación.

Definición 2.2.4. Si $\mathbf{X} = [X(0), X(1), \dots, X(N-1)]^T$ es la representación vectorial de una sucesión aleatoria finita, entonces la matriz de autocorrelación $\mathbf{R}_{\ddot{X}} = E[\mathbf{X}\mathbf{X}^T]$ esta dada por

$$\begin{bmatrix} E[X(0)X(0)] & E[X(0)X(1)] & \cdot & \cdot & \cdot & E[X(0)X(N-1)] \\ E[X(1)X(0)] & E[X(1)X(1)] & \cdot & \cdot & \cdot & E[X(1)X(N-1)] \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ E[X(N-1)X(0)] & E[X(N-1)X(1)] & \cdot & \cdot & \cdot & E[X(N-1)X(N-1)] \end{bmatrix}$$

2.2.5 Espectro de Potencia.

Definición 2.2.5. El espectro de potencia $S(\omega)$ (o densidad espectral también denotado por $S_{\ddot{X}}(\omega)$ o por $S_{X_X}(\omega)$) de un proceso X_t es la Transformada de Fourier (1.1) de su autocorrelación

$$S(\omega) = \int_{-\infty}^{\infty} e^{-j\omega\tau} R(\tau) d\tau.$$

Mediante la Transformada Inversa de Fourier (1.6) se sigue que $R(\tau)$ puede ser expresado en terminos de $S(\omega)$:

$$R(\tau) = \frac{1}{2\pi} \int_{-\infty}^{\infty} S(\omega) e^{j\omega\tau} d\omega$$

Si $\tau = 0$, tenemos que

$$\frac{1}{2\pi} \int_{-\infty}^{\infty} S(\omega) d\omega = R(0),$$

donde $R(0) = E[|X_t|^2] \geq 0$. Entonces, interpretando la integral como el área bajo la curva, tenemos que $S(\omega)/2\pi$ es el área no negativa e igual al promedio de potencia del proceso X_t .

Definición 2.2.6. El espectro de potencia cruzado $S_{XY}(\omega)$ de los procesos X_t y Y_t es la Transformada de Fourier de su Correlación Cruzada; es decir,

$$S_{XY}(\omega) = \int_{-\infty}^{\infty} R_{XY}(\tau) e^{-j\omega\tau} d\tau$$

La Inversa de la Transformada de Fourier da

$$R_{XY}(\tau) = \frac{1}{2\pi} \int_{-\infty}^{\infty} S_{XY}(\omega) e^{j\omega\tau} d\omega$$

Si $\tau = 0$,

$$\frac{1}{2\pi} \int_{-\infty}^{\infty} S_{XY}(\omega) d\omega = R_{XY}(0) = E[X_t Y_t^*].$$

Si los procesos X_t y Y_t son ortogonales (i.e $R_{XY}(\tau) = 0$), entonces $S_{XY}(\omega) = 0$. En este caso,

$$R_{X+Y}(\tau) = R_X(\tau) + R_Y(\tau) \quad \text{y} \quad S_{X+Y}(\omega) = S_X(\omega) + S_Y(\omega).$$

Sea Y_t una señal de salida, tal que

$$Y_t = \int_{-\infty}^{\infty} X_{t-\alpha} h(\alpha) d\alpha = \int_{-\infty}^{\infty} Y_\alpha h(t-\alpha) d\alpha \quad (2.5)$$

donde X_t es un proceso. Para determinar la correlación de la salida Y_t , primero necesitamos obtener la correlación cruzada entre X_t y Y_t multiplicando ambos lados de la última ecuación por $X_{t-\tau}^*$, por lo que tenemos

$$Y_t X_{t-\tau}^* = \int_{-\infty}^{\infty} X_{t-\alpha} X_{t-\tau}^* h(\alpha) d\alpha. \quad (2.6)$$

Pero

$$E[X_{t-\alpha} X_{t-\tau}^*] = R_{XX}[(t-\alpha) - (t-\tau)] = R_{XX}(\tau - \alpha).$$

Entonces, tomando el valor esperado de la ecuación (2.6) se obtiene que

$$E[Y_t X_{t-\tau}^*] = \int_{-\infty}^{\infty} R_{XX}(\tau - \alpha) h(\alpha) d\alpha.$$

Esta integral es independiente de t , y es igual a la convolución de $R_{XX}(\tau)$ con $h(\tau)$. El lado izquierdo de esta ecuación es también independiente de t , y puesto que es igual a la correlación cruzada de X_t con Y_t , concluimos que

$$R_{YX}(\tau) = R_{XX}(\tau) * h(\tau). \quad (2.7)$$

Multiplicando el conjugado de ambos lados de (2.5) por $Y_{t+\tau}$ obtenemos

$$Y_{t+\tau} Y_t^* = \int_{-\infty}^{\infty} Y_{t+\tau} X_{t-\alpha}^* h^*(\alpha) d\alpha.$$

Entonces

$$R_{YY}(\tau) = \int_{-\infty}^{\infty} R_{YX}(\tau + \alpha) h^*(\alpha) d\alpha = R_{YX}(\tau) * h^*(-\tau) \quad (2.8)$$

Un argumento semejante muestra que:

$$R_{XY}(\tau) = R_{XX}(\tau) * h^*(-\tau) \quad (2.9)$$

$$R_{YY}(\tau) = R_{XY}(\tau) * h(\tau) \quad (2.10)$$

$$\Rightarrow R_{YY}(\tau) = R_{XX}(\tau) * h^*(-\tau) * h(\tau) \quad (2.11)$$

Consideremos la Transformada de Fourier (1.1) para una función $h(t)$.

$$H(\omega) = \int_{-\infty}^{\infty} h(t)e^{-j\omega t} dt \quad (2.12)$$

Del teorema (1.0.5) y la definición del espectro de potencia (definición 2.1.28), obtenemos de (2.9) y (2.10) que

$$S_{XY}(\omega) = S_{XX}(\omega)H^*(\omega) \quad \text{y} \quad S_{XY}(\omega) = S_{XX}(\omega)H(\omega) \quad (2.13)$$

Combinando estos resultados obtenemos el siguiente teorema.

Teorema 2.2.7. *El espectro de potencia $S_{YY}(\omega)$ de la salida del sistema lineal (ecuación (2.5)) está dado por*

$$S_{YY}(\omega) = S_{XX}(\omega)|H(\omega)|^2, \quad (2.14)$$

donde $S_{XX}(\omega)$ es el espectro de potencia de la entrada X_t .

Para el lector interesado en profundizar en este tema, le sugerimos ver las referencias [4] y [7].

2.3 Estimación de parámetros.

Sea θ un parámetro cuyo valor se desea conocer a partir de una muestra. Sea X_1, \dots, X_n una muestra aleatoria y sea $\hat{\theta} = T(X_1, \dots, X_n)$ una función de la muestra, la cual utilizaremos para estimar el valor de θ . Obsérvese que el valor θ es una función que depende de la muestra y se nombra *estimador*. Realmente el valor que toma $\hat{\theta}$ en una muestra determinada es la estimación de θ .

Hay dos tipos básicos de estimación:

I Estimación puntual.

II Estimación por intervalo (intervalo de confianza).

En el presente trabajo revisaremos solamente la estimación puntual, dado que no usaremos la estimación por intervalo, el lector interesado puede consultar la referencia [10].

Métodos de estimación puntual.

1. Método de mínimos cuadrados
2. Métodos de máxima verosimilitud
3. Método de los momentos, entre otros.

En el presente trabajo hacemos uso de los métodos de mínimos cuadrados y máxima verosimilitud. Por ahora los otros métodos de estimación puntual no los usaremos.

Propiedades deseables de un estimador.

En estadística la esperanza matemática (también llamada esperanza, valor esperado, media poblacional o media) de una variable aleatoria X , es $E(X)$ de la definición 2.1.20 que formaliza la idea de valor medio de un fenómeno aleatorio.

1. Ausencia de sesgo (insesgado).

Definición 2.3.1. Si $\hat{\theta}$ es un estimador de θ , se llama sesgo a la cantidad $E(\hat{\theta}) - \theta$.

Definición 2.3.2. Se dice que un estimador puntual $\hat{\theta}$ es un estimador insesgado de θ si $E(\hat{\theta}) = \theta$.

En otras palabras, un estimador insesgado es aquel para el cual la media de su distribución muestral es el parámetro estimado. O también, el estimador cuyo sesgo es cero. En general, se prefiere un estimador insesgado a uno que no lo sea.

2. Consistencia.

Muchos estimadores no tienen buenas propiedades para muestras pequeñas, pero cuando el tamaño muestral aumenta, muchas de las propiedades deseables pueden cumplirse. En esta situación se habla de propiedades asintóticas de los estimadores. Como el estimador va a depender del tamaño de la muestra vamos a expresarlo utilizando el símbolo $\hat{\theta}_n$. Por ejemplo, el sesgo puede depender del tamaño de la muestra. Si el sesgo tiende a cero cuando el tamaño de la muestra crece hasta infinito decimos que el estimador es *asintóticamente insesgado*. Específicamente

Definición 2.3.3. Un estimador $\hat{\theta}_n$ se dice que es *asintóticamente insesgado* si

$$\lim_{n \rightarrow \infty} E(\hat{\theta}_n) = \theta,$$

ó equivalentemente

$$\lim_{n \rightarrow \infty} [E(\hat{\theta}_n) - \theta] = 0.$$

Definición 2.3.4. Se dice que un estimador $\hat{\theta}_n$ es **consistente** con θ si se cumple que

$$\lim_{n \rightarrow \infty} P(|\hat{\theta}_n - \theta| > \epsilon) = 0,$$

esto $\forall \epsilon > 0$. Ó lo que es lo mismo, si $\lim_{n \rightarrow \infty} P(|\hat{\theta}_n - \theta| \leq \epsilon) = 1$, esto $\forall \epsilon > 0$.

Es decir, a medida que se incrementa el tamaño muestral, el estimador se acerca más y más al valor del parámetro. La *consistencia* es una propiedad asintótica. Dicho de otro modo, que $\hat{\theta}$ sea un estimador consistente de θ significa que para muestras grandes es poco probable que $\hat{\theta}$ se aleje de θ .

Proposición 2.3.5. *Si $Var(\hat{\theta}_n) \rightarrow 0$ y $E(\hat{\theta}_n)$ entonces $\hat{\theta}_n$ es consistente.*

La varianza muestral $s_n^2 := \frac{1}{n} \sum_{i=1}^n (X_i - \bar{X})^2$ es también un estimador consistente de la varianza poblacional σ^2 , dado que a medida que el tamaño muestral se incrementa, el sesgo disminuye.

3. Eficiencia.

Para decidir cuál estimador seleccionar entre varios estimadores insesgados se puede utilizar las varianzas de los mismos. La varianza $var(X)$ de una variable aleatoria X mide la dispersión alrededor de la media $E(X)$. Menor varianza para una variable aleatoria significa que, en promedio, sus valores fluctúan poco alrededor de la media comparados con los valores de otra variable aleatoria con la misma media y mayor varianza. Menor varianza implica mayor precisión y entonces el estimador que tenga menor varianza es claramente más deseable porque, en promedio, está más cerca del verdadero valor de θ .

Definición 2.3.6. *Si θ es un estimador insesgado de θ , se dice que $\hat{\theta}$ es un estimador insesgado eficiente o de varianza mínima para θ , si para cualquier otro estimador insesgado de θ , digamos $\tilde{\theta}$, se verifica que*

$$Var(\hat{\theta}) \leq Var(\tilde{\theta}).$$

En otras palabras si consideramos a los estimadores $\hat{\theta}_1$ y $\hat{\theta}_2$ del parámetro θ , con varianzas $Var(\hat{\theta}_1)$ y $Var(\hat{\theta}_2)$ respectivamente tales que $Var(\hat{\theta}_1) < Var(\hat{\theta}_2)$, decimos que $\hat{\theta}_1$ es más eficiente que $\hat{\theta}_2$.

2.3.1 Estimación por Mínimos Cuadrados.

El antiguo uso de los mínimos cuadráticos ha sido regresión lineal, la cuál corresponde al problema de encontrar una línea ó curva que mejor describa el comportamiento del conjunto de datos. En la formulación estándar para este problema, se considera a un conjunto de n pares de observaciones $\{Y_i, X_i\}$ es usado para encontrar una función dado el valor de la variable dependiente Y , de los valores independientes X . Con una variable y una función lineal, la predicción esta representada por la siguiente ecuación:

$$\hat{Y} = a + bX.$$

Esta ecuación involucra dos parámetros libres a y b . El método de mínimos cuadrados definen el estimador de estos parámetros como los valores los cuales

mínimizan la suma de cuadrados (de aquí el nombre *mínimos cuadrados*) entre las medidas y el modelo (i.e. los valores estimados). Específicamente el problema es un problema de optimización siguiente:

min

$$\varepsilon = \sum_i (Y_i - \hat{Y}_i)^2 = \sum_i [Y_i - (a + bX_i)]^2,$$

donde ε expresa el error, el cual es la cantidad a minimizar. La forma de resolver este problema es usando técnicas estándar del cálculo. Tomando la derivada de ε con respecto a los parámetros a y b e igualando a cero, obtenemos el siguiente conjunto de ecuaciones llamadas; *ecuaciones normales*:

$$\begin{aligned} \frac{\partial \varepsilon}{\partial a} &= 2Na + 2b \sum X_i - 2 \sum Y_i = 0 \\ \frac{\partial \varepsilon}{\partial b} &= 2b \sum X_i^2 + 2a \sum X_i - 2 \sum Y_i X_i = 0 \end{aligned}$$

Resolviendo estas dos ecuaciones obtenemos el estimador de mínimos cuadrados para a y b :

$$\begin{aligned} a &= M_Y - bM_X \\ b &= \frac{\sum (Y_i - M_Y)(X_i - M_X)}{\sum (X_i - M_X)^2}, \end{aligned}$$

donde M_X y M_Y denotan la media de X y Y . Extendiendo esta idea a más de una variable independiente y funciones no lineales hacemos uso del álgebra lineal.

La mejor aproximación deberá tender a interpolar la función de la que proviene el conjunto de pares $\{X_i, Y_i\}$, esto es, deberá tender a pasar exactamente por todos los puntos. Eso supone que se debería cumplir que $f(X_i) = Y_i$, con $i = 1, \dots, n$. Sustituyendo $f(X)$ por su expresión, tendremos $\sum_{j=1}^m c_j f_j(X_i) = Y_i$, con $i = 1, \dots, n$. Esto es, se tendría que verificar exactamente un sistema de n ecuaciones y m incógnitas, pero como en general $n > m$, dicho sistema está sobredeterminado, no tiene solución general. De ahí surge la necesidad de aproximarlos. Dicho sistema podría expresarse en forma matricial como:

$$Ac = b$$

La aproximación trata de hallar el vector c , que mejor aproxime el sistema $Ac = b$. Con dicho vector c , es posible definir el vector residuo como $r = b - Ac$. El problema de Mínimos cuadrados es minimizar

$$\|r\| = \|b - Ac\|$$

El problema de aproximación será hallar aquella combinación lineal de columnas de A lo más cercano posible al vector b . Se comprueba que el conjunto de las columnas de A generan un espacio vectorial $span(A_1, A_2, \dots, A_m)$, al que el vector b no tiene por qué pertenecer (si lo hiciera, el sistema $Ac = b$ tendría solución). De todos los vectores del espacio vectorial $span(A_1, A_2, \dots, A_m)$ que son combinación lineales de los vectores de la base, se tratará de hallar el más cercano al vector b . De entre todos ellos, el que cumple esto con respecto a la norma euclídeana es la proyección ortogonal del vector b sobre el espacio vectorial $span(A_1, A_2, \dots, A_m)$. La condición de minimización del residuo será:

$$\begin{aligned} r &\perp span(A_1, \dots, A_m) \\ \Leftrightarrow r &\perp A_j \\ \Leftrightarrow A^T r &= 0 \end{aligned}$$

La última igualdad porque cada una de las m condiciones de perpendicularidad se puede agrupar en una sola para $j = 1, \dots, m$. Sustituyendo el residuo r , obtenemos

$$A^T(b - Ac) = 0 \quad \text{ó} \quad A^T Ac = A^T b$$

Por tanto la mejor aproximación lineal por mínimos cuadrados, para un conjunto de números discretos, se obtiene al resolver $A^T Ac = A^T b$. La solución generalizada Moore-Penrose, está dada por la ecuación (1.28) del primer capítulo.

2.3.2 Estimador de Máxima Verosimilitud.

Consideremos de la definición 2.1.13 a la función

$$F_X(x) = P\{X \leq x\}, \quad x \in \mathbb{R},$$

con X variable aleatoria, de manera que se satisfacen las siguientes propiedades

1. Es una función no decreciente: $\forall x_0 \leq x_1, F_X(x_0) \leq F_X(x_1)$.
2. Se satisfacen los siguientes límites

$$\lim_{x \rightarrow -\infty} F_X(x) = 0, \quad \lim_{x \rightarrow \infty} F_X(x) = 1$$

3. Es continua por la derecha i.e $F(x^+) = F(x)$, esto es

$$F(x^+) = \lim_{\epsilon \rightarrow 0} F(x + \epsilon).$$

Definición 2.3.7. Una variable aleatoria X se dice discreta si existen; conjuntos $\{x_i\} \subset \mathbb{R}$ y $\{p_i\} \subset \mathbb{R}_+$ tales que

$$p_i = P\{X = x_i\} > 0$$

y

$$\sum_i p_i = 1.$$

Definición 2.3.8. Una función de densidad de probabilidad para la variable aleatoria X es una función $p_X : \mathbb{R} \rightarrow \mathbb{R}$ tal que

$$p_X(x) = \begin{cases} p_i & \text{si } x = x_i \\ 0 & \text{si } x \neq x_i \end{cases}$$

Definición 2.3.9. Dadas X, Y variables aleatorias.

a Definamos la función de distribución conjunta para (X, Y) como

$$p_{X,Y}(x, y) = P\{X = x, Y = y\},$$

donde x, y son observaciones de las variables aleatorias X y Y , respectivamente.

b La función de distribución marginal de X está definida como

$$p_X(x) = \sum_{P\{Y=y\}>0} p_{X,Y}(x, y).$$

c La probabilidad condicional para Y dado $X = x$ está dada por

$$p_{Y|X}(y | x) = \frac{p_{X,Y}(x, y)}{p_X(x)},$$

siempre que $p_X(x)$ sea diferente de cero.

Existen diferentes procedimientos para estimar los parámetros de una distribución de probabilidad. De entre esos procedimientos probablemente el más versátil, ya que se puede aplicar en gran cantidad de situaciones, y por ello uno de los más empleados se conoce con el nombre de "método de máxima verosimilitud" (en inglés "method of maximum likelihood"). Este método es la base para desarrollar el algoritmo de Lucy-Richardson, el cual es el objetivo central en el presente trabajo. Para ilustrar este método de máxima verosimilitud veremos un ejemplo en la sección 2.3.3.

Sea \mathbf{X} un vector aleatorio con función de densidad $p_X(x; \theta)$, donde θ es un parámetro desconocido, y sea $\mathbf{d} = (d_1, \dots, d_n)$ un vector de datos el cual es resultado de un experimento; $\mathbf{d} = X(s)$, donde s es elemento del espacio muestral Ω . \mathbf{d} simplemente es un vector; $\mathbf{d} = (X_1, \dots, X_n)$.

Definición 2.3.10. Un estimador de máxima verosimilitud (EMV) de θ para \mathbf{d} es un parámetro $\hat{\theta}$ que maximiza la función de verosimilitud

$$L(\theta) := p_X(\mathbf{d}; \theta). \quad (2.15)$$

Esta definición nos permite encontrar un estimador óptimo para nuestro problema. Como veremos más adelante este estimador posee las propiedades deseables en un estimador (insesgado, consistente y eficiente).

El logaritmo al ser una transformación monótona creciente preserva máximos, este hecho nos permite simplificar los cálculos para el EMV. Así un EMV es un máximo de la llamada función de log-verosimilitud,

$$\ell(\theta) = \log p_X(\mathbf{d}; \theta). \quad (2.16)$$

2.3.3 El Algoritmo EM (Esperanza-Maximización).

Sea \mathbf{Y} un vector aleatorio con distribución de probabilidad $p_Y(y; \theta)$ que depende de un parámetro θ . El algoritmo *EM* es un proceso iterativo para el cual, dado \mathbf{Y} (datos incompletos), se obtiene una sucesión de aproximaciones de un estimador de máxima verosimilitud para el parámetro θ . El algoritmo emplea un vector aleatorio auxiliar \mathbf{X} , el cual corresponde a los datos no observados, completos. Para mayor claridad supongamos que X y Y son variables aleatorias discretas con la misma distribución, y cuya distribución conjunta es $p_{X,Y}(x, y; \theta)$, donde θ denota el parámetro de interés. Entonces la probabilidad condicional (Definición 2.1.33) para X dado Y está dada por

$$p_{X|Y}(x | y; \theta) = \frac{p_{X,Y}(x, y; \theta)}{p_Y(y; \theta)}, \quad (2.17)$$

donde el denominador $p_Y(y; \theta)$ es la probabilidad marginal de Y , esto es

$$p_Y(y; \theta) = \sum_{\{x | P\{X=x\} > 0\}} p_{X,Y}(x, y; \theta). \quad (2.18)$$

De acuerdo a la ecuación (2.16), la función de verosimilitud para Y dadas las observaciones y y usando las ecuaciones (2.17) y (2.18), tenemos

$$\begin{aligned} l_Y(\theta; y) &= \log p_Y(y; \theta) \\ &= \log p_{X,Y}(x, y; \theta) - \log p_{X|Y}(x | y; \theta) \\ &= l_{X,Y}(\theta; x, y) - l_{X|Y}(\theta; x | y), \end{aligned}$$

por tanto

$$l_Y(\theta; y) = l_{X,Y}(\theta; x, y) - l_{X|Y}(\theta; x | y).$$

También para cualquier parámetro fijo θ_ν ,

$$\begin{aligned} l_Y(\theta; y) &= l_Y(\theta; y) \left[\sum \frac{p_{X,Y}(x, y; \theta_\nu)}{p_Y(y; \theta_\nu)} \right] \\ &= l_Y(\theta; y) \sum_x p_{X|X}(x | y; \theta_\nu) \\ &= \sum_x l_Y(\theta; y) p_{X|Y}(x | y; \theta_\nu) \\ &= \sum_x l_{X,Y}(\theta; x, y) p_{X|Y}(x | y; \theta_\nu) \\ &\quad - \sum_x l_{X|Y}(\theta; x | y) p_{X|Y}(x | y; \theta_\nu) \end{aligned}$$

Si el primer término de la última igualdad lo denotamos por $Q(\theta | y; \theta_\nu)$ y el segundo por $H(\theta | \mathbf{y}; \theta_\nu)$ entonces tenemos que

$$l_Y(\theta; y) = Q(\theta | y; \theta_\nu) - H(\theta | y; \theta_\nu) \quad (2.19)$$

Proposición 2.3.11. *Si*

$$Q(\theta_{\nu+1} | y; \theta_\nu) \geq Q(\theta_\nu | y; \theta_\nu)$$

entonces

$$l_Y(\theta_{\nu+1}; y) \geq l_Y(\theta_\nu; y).$$

Demostración

Sea $\theta_{\nu+1}$ un parámetro cualquiera, por (2.19) se tiene que

$$\begin{aligned} l_Y(\theta_{\nu+1}; y) - l_Y(\theta_\nu; y) &= [Q(\theta_{\nu+1} | y; \theta_\nu) - Q(\theta_\nu | y; \theta_\nu)] \\ &\quad + [H(\theta_\nu | y; \theta_\nu) - H(\theta_{\nu+1} | y; \theta_\nu)] \end{aligned}$$

Para demostrar nuestra afirmación es suficiente verificar que el término contenido en el segundo corchete es no negativo. Para esto observamos que

$$\begin{aligned} H(\theta_\nu | y; \theta_\nu) - H(\theta_{\nu+1} | y; \theta_\nu) &= - \sum_x [l_{X|Y}(\theta_{\nu+1}; x | y) - l_{X|Y}(\theta_\nu; x | y)] \cdot \\ &\quad \cdot p_{X|Y}(x | y; \theta_\nu) \\ &= - \sum_x \log \left(\frac{p_{X|Y}(x | y; \theta_{\nu+1})}{p_{X|Y}(x | y; \theta_\nu)} \right) p_{X|Y}(x | y; \theta_\nu) \\ &\geq - \log \sum_x p_{X|Y}(x | y; \theta_\nu) \\ &= - \log(1) \\ &= 0 \end{aligned}$$

La desigualdad se cumple por la convexidad de la función $-\log$, mientras que las primeras igualdades se siguen de (2.17) y (2.18).

□

Este resultado motiva el siguiente procedimiento iterativo para maximizar la función de verosimilitud $l_Y(\theta; y)$, dada una observación y de un vector aleatorio \mathbf{Y} , con valor inicial θ_0 para el parámetro θ , el cual llamaremos Algoritmo EM.

Algoritmo EM (Esperanza Maximización).

Dado θ_0 .

Para $\nu = 1, \dots, N - 1$.

1. (**Esperanza**) Consiste en calcular $Q(\theta | \mathbf{y}; \theta_\nu)$, la esperanza condicional de la función verosimilitud para la observación \mathbf{y} y la aproximación del Estimador de Máxima Verosimilitud θ_ν . En el caso discreto, tiene la forma

$$Q(\theta | \mathbf{y}; \theta_\nu) = \sum_{\mathbf{x}} l(\mathbf{x}, \mathbf{y})(\theta; \mathbf{x}, \mathbf{y}) p_{\mathbf{x}|\mathbf{y}}(\mathbf{x} | \mathbf{y}; \theta_\nu), \quad (2.20)$$

esto de la ecuación (2.19a)

2. (**Maximización**) Consiste en calcular un máximo $\theta_{\nu+1}$ de $Q(\theta | \mathbf{y}; \theta_\nu)$.

Termina.

Ejemplo 2.3.12. (Para ilustrar el algoritmo EM.)

Supongamos que \mathbf{y} representa el conjunto de datos observados (También llamado el conjunto incompleto). Supongamos que \mathbf{x} representa el conjunto de datos completos, el cual no es observado directamente, si no a través de \mathbf{y} . Es decir, existe una función $h : \mathcal{X} \rightarrow \mathcal{Y}$, donde \mathcal{X} , \mathcal{Y} representan el conjunto de vectores de datos completos e incompletos respectivamente. Rao en [9], [10] presenta datos de 197 animales distribuidos multinomialmente en 4 categorías, esto es

$$\mathbf{y} = (y_1, y_2, y_3, y_4)^T = (125, 18, 2, 34)$$

El modelo genético para esta población tiene probabilidad multinomial de aparición

$$\begin{aligned} \mathbf{p} &= \left(\frac{1}{4} + \frac{1}{4}\theta, \frac{1}{4}(1-\theta), \frac{1}{4}(1-\theta), \frac{1}{4}\theta \right) \\ &= (p_1, p_2, p_3, p_4) \quad 0 \leq \theta \leq 1. \end{aligned}$$

Entonces

$$g(\mathbf{y} | \theta) = \frac{(y_1 + y_2 + y_3 + y_4)!}{y_1! y_2! y_3! y_4!} p_1^{y_1} p_2^{y_2} p_3^{y_3} p_4^{y_4}$$

Los datos completos están representados por el vector $\mathbf{x} = (x_1, x_2, x_3, x_4, x_5)$ que le llamaremos de quinta categoría, con probabilidad

$$\begin{aligned} \mathbf{q} &= \left(\frac{1}{4}, \frac{1}{4}\theta, \frac{1}{4}(1-\theta), \frac{1}{4}(1-\theta), \frac{1}{4}\theta \right) \\ &= (q_1, q_2, q_3, q_4, q_5) \end{aligned}$$

La idea consiste en bi-partir la primera categoría de los datos incompletos \mathbf{y} en dos categorías, esto es dado $\mathbf{x} = (x_1, x_2, x_3, x_4, x_5)$ elegimos \mathbf{y} tal que $y_1 = x_1 + x_2$, $y_2 = x_4$, $y_3 = x_3$, $y_4 = x_5$, de tal manera que

$$f(\mathbf{x} | \theta) = \frac{(x_1 + x_2 + x_3 + x_4 + x_5)!}{x_1! x_2! x_3! x_4! x_5!} q_1^{x_1} q_2^{x_2} q_3^{x_3} q_4^{x_4} q_5^{x_5}$$

(Notemos que $q_3 = p_2$, $q_4 = p_3$, $q_5 = p_4$).

Calcular $Q(\theta | \mathbf{y}, \theta_\nu)$, consideremos para ello a la función de distribución multinomial de datos completos, esto es

$$f(\mathbf{x} | \theta) = \frac{(x_1 + x_2 + x_3 + x_4 + x_5)!}{x_1!x_2!x_3!x_4!x_5!} \left(\frac{1}{2}\right)^{x_1} \left(\frac{\theta}{4}\right)^{x_2} \left(\frac{1}{4}(1-\theta)\right)^{x_3} \left(\frac{1}{4}(1-\theta)\right)^{x_4} \left(\frac{\theta}{4}\right)^{x_5}$$

Luego, el logaritmo de esta función será

$$\log f(\mathbf{x} | \theta) = x_2 \log(\theta) + (x_3 + x_4) \log(1 - \theta) + x_5 \log(\theta) + O(const),$$

donde $O(const)$ son terminos constantes, que no dependen de θ .

Aplicando el Paso 1 (**Esperanza**) tenemos como consecuencia

$$\begin{aligned} Q(\theta | \mathbf{y}; \theta_\nu) &= E[\log f(\mathbf{x} | \theta) | \mathbf{y}, \theta_\nu] \\ &= E[x_2 \log(\theta) + (x_3 + x_4) \log(1 - \theta) + x_5 \log(\theta) | \mathbf{y}, \theta_\nu] \\ &= \log(\theta) E[x_2 | \mathbf{y}, \theta_\nu] + (x_3 + x_4) \log(1 - \theta) + x_5 \log(\theta) \\ &= \log(\theta) E[x_2 | \mathbf{y}, \theta_\nu] + (y_2 + y_3) \log(1 - \theta) + y_4 \log(\theta) \\ &= y_1 \frac{\theta_\nu}{2 + \theta_\nu} \log(\theta) + (x_3 + x_4) \log(1 - \theta) + x_5 \log(\theta), \end{aligned}$$

donde $E[x_2 | x_1 + x_2 = y_1] = y_1 \frac{p_2}{p_1 + p_2}$ y también $E[x_1 | x_1 + x_2 = y_1] = y_1 \frac{p_1}{p_1 + p_2}$ con $p_1 = 1/2$, $p_2 = \theta/4$.

Paso 2 (**Maximización**)

Para maximizar Q , con respecto a θ , usamos la misma técnica de cálculo diferencial, derivamos e igualamos a cero.

$$\begin{aligned} Q(\theta | \mathbf{y}; \theta_\nu) &= y_1 \frac{\theta_\nu}{2 + \theta_\nu} \log(\theta) + (y_2 + y_3) \log(1 - \theta) + y_4 \log(\theta) \\ &= \left(\frac{y_1 \theta_\nu}{2 + \theta_\nu} + y_4 \right) \log(\theta) + (y_2 + y_3) \log(1 - \theta). \end{aligned}$$

Entonces

$$\frac{\partial Q(\theta | \mathbf{y}; \theta_\nu)}{\partial \theta} = \left(\frac{y_1 \theta_\nu}{2 + \theta_\nu} + y_4 \right) \frac{1}{\theta} - (y_2 + y_3) \frac{1}{1 - \theta} = 0$$

$$\begin{aligned}
&\Leftrightarrow \left(\frac{y_1 \theta_\nu}{2 + \theta_\nu} + y_4 \right) (1 - \theta) = (y_2 + y_3) \theta \\
&\Leftrightarrow \left(\frac{y_1 \theta_\nu}{2 + \theta_\nu} + y_4 \right) = \left(\frac{y_1 \theta_\nu}{2 + \theta_\nu} + y_4 \right) \theta + (y_2 + y_3) \theta \\
&\Leftrightarrow \left(\frac{y_1 \theta_\nu}{2 + \theta_\nu} + y_4 \right) = \left(\frac{y_1 \theta_\nu}{2 + \theta_\nu} + y_4 + y_2 + y_3 \right) \theta \\
&\Leftrightarrow \theta = \frac{x_2^\nu + y_4}{x_2^\nu + y_4 + y_2 + y_3} = \theta^{\nu+1},
\end{aligned}$$

donde

$$x_2^\nu = \frac{y_1 \theta_\nu}{2 + \theta_\nu}$$

Teniendo además que la segunda derivada de Q respecto de θ es

$$\frac{\partial^2 Q(\theta | \mathbf{y}; \theta_\nu)}{\partial \theta^2} = -(x_2^\nu + y_4 + y_2 + y_3)^2 \left[\frac{1}{x_2^\nu + y_4} + \frac{y_3 + y_4}{y_2 + y_3} \right] < 0,$$

esto nos garantiza el tener un máximo local. Implementando este algoritmo en matlab, resulta que el parámetro aproximado es $\theta^{\nu+1} = 0.6268$, con 8 iteraciones.

Veamos el contraste con el método de máxima verosimilitud tomamos en cuenta que

$$g(\mathbf{y} | \theta) = \frac{(y_1 + y_2 + y_3 + y_4)!}{y_1! y_2! y_3! y_4!} p_1^{y_1} p_2^{y_2} p_3^{y_3} p_4^{y_4},$$

entonces

$$\begin{aligned}
\log g(\mathbf{y} | \theta) &= \log \left(\frac{(y_1 + y_2 + y_3 + y_4)!}{y_1! y_2! y_3! y_4!} p_1^{y_1} p_2^{y_2} p_3^{y_3} p_4^{y_4} \right) \\
&\cdot \\
&\cdot \\
&\cdot \\
&= y_1 \log(2 + \theta) + (y_2 + y_3) \log(1 - \theta) + y_4 \log(\theta) + O(const) \\
&= L(\theta | \mathbf{y})
\end{aligned}$$

derivando $L(\theta | \mathbf{y})$ e igualando a cero, obtenemos

$$A\theta^2 + (2(y_2 + y_3) + y_4 - y_1)\theta - 2y_4 = 0,$$

resolviendo para θ , obtenemos el estimador

$$\hat{\theta} = \frac{-((2(y_2 + y_3) + y_4 - y_1)) \pm \sqrt{(2(y_2 + y_3) + y_4 - y_1)^2 + 8Ay_4}}{2A}$$

Ahora como

$$\frac{\partial^2 L(\theta | \mathbf{y})}{\partial \theta^2} = -\frac{y_1}{(2 + \theta)^2} - \frac{y_2 + y_3}{(1 - \theta)^2} - \frac{y_4}{\theta^2} \leq 0$$

Por tanto $\hat{\theta}$ es el estimador de máxima verosimilitud cuyo valor es

$$\hat{\theta} = \frac{15 + 231.968}{394} \approx 0.626822$$

Observación: La sucesión $\{\theta_\nu\}$ converge eventualmente a un estimador de máxima verosimilitud bajo ciertas condiciones. Este algoritmo es aplicado al proceso de encontrar un estimador de una imagen, dicha aplicación de este algoritmo se conoce como algoritmo de Lucy- Richardson EM, que es el tema principal de este trabajo. Abordaremos este tema en el siguiente capítulo.

Capítulo 3

Deconvolución de Imágenes.

3.1 Formulación del Problema.

En la sección 1.4.1 vimos la forma genérica de la ecuación integral de Fredholm de primer tipo

$$g(s) = \int_0^1 h(s, t)f(t)dt \quad 0 \leq s \leq 1,$$

donde h es el núcleo, g es una función conocida y f es una función desconocida. Observemos que en esta ecuación integral se hace uso de la definición 1.05; la convolución de h con f da como resultado a la función con la que contamos g . Este es un problema inverso complicado de resolver, y aún más difícil cuando además de f , h es también desconocida. A este problema le llamamos *deconvolución ciega*, pues no conocemos nada respecto de h , cuando se conoce parcialmente al núcleo h el problema se conoce como *deconvolución miope*.

El problema de resolver la ecuación integral de Fredholm es un problema complicado [26], por ende es un campo de estudio fértil en matemáticas hoy día (problemas inversos). La ecuación integral de Fredholm tiene otra versión más complicada de resolver, llamada *ecuación integral de Fredholm de segundo tipo* ver [26], dicha ecuación tiene la forma

$$g(s) = \lambda \int_a^b h(s, t)f(t)dt + \eta(s) \quad a \leq s \leq b,$$

donde h es el núcleo, η es factor independiente de f ; la cual es desconocida, g función conocida y λ un escalar. También esta ecuación hace uso de la convolución, multiplicada por una constante λ y agregando un término η que en forma general es independiente de f (función que queremos recuperar), i.e. no hay relación entre f y η .

En procesamiento digital de imágenes, el modelo general para una degradación lineal causada por un proceso de borrado h , con ruido aditivo n , está dado en forma discreta por

$$g(x) = \sum_s h(x, s)f(s) + n(x), \quad (3.1)$$

donde $f(s)$, $g(x)$, $h(x, s)$ y $n(x)$ representan la imagen original, la imagen observada, la función de dispersión del punto ó núcleo y el ruido observado, respectivamente. Observemos que este modelo es muy parecido a la *ecuación integral de Fredholm de segundo tipo*, solo que en forma discreta, con convolución discreta (definición 1.1.5), $\lambda = 1$ y n , que correspondería a η .

El ruido $n(x)$ se pudo originar durante la adquisición de la imagen, al momento de procesar la imagen o al transmitirla. Comúnmente los tipos de ruido son electrónicos, fotoelectricos ó ruido de cuantización (cuando se convierte una señal analógica en digital, no siempre se trata de un ruido en sentido estricto: en ocasiones se debe entender como una distorsión.), entre otros. Es común asumir $n(x)$ como ruido Gaussiano, el cual no está relacionado con la imagen original.

El objetivo del problema de deconvolución ciega es estimar tanto f como h a partir de g . La dificultad en resolver este problema con un movimiento de borrado espacialmente variable, motiva a un modelo de movimiento más sencillo, como el modelo más sencillo. Esto nos conduce a la siguiente expresión para la degradación del sistema:

$$g(x) = (h * f)(x) + n(x) = \sum_s h(x - s)f(s) + n(x), \quad (3.2)$$

donde el operador $*$ denota la convolución. Los modelos de las ecuaciones (3.1) y (3.2) se pueden expresar matricialmente como

$$\mathbf{g} = \mathbf{H}\mathbf{f} + \mathbf{n}, \quad (3.3)$$

donde $\mathbf{g} \in \mathbb{R}^{MN}$ representa a la imagen observada, $\mathbf{H} \in \mathbb{R}^{MN \times MN}$ el núcleo y $\mathbf{n} \in \mathbb{R}^{MN}$ el ruido. Como en la sección 1.3 (representación de imágenes) \mathbf{H} es una matriz de bloque circulante (en el apéndice D se da una revisión de este tipo de matrices).

La **deconvolución** es una operación matemática. Se usa en restauración de imágenes para recuperar datos que han sido degradados por cualquier proceso físico que pueda describirse mediante la operación inversa, una *convolución*.

Si la convolución implica reemplazar cada fuente luminosa puntual original por su correspondiente función de dispersión del punto (PSF) para producir una imagen borrosa, el proceso de restauración sigue el camino inverso, recolectando toda la luz dispersa y poniéndola en su sitio de nuevo. Esto produce una mejor

representación del objeto real.

En este trabajo el problema de Deconvolución Ciega se refiere a encontrar estimadores $\hat{f}(x)$ y $\hat{h}(x)$ para $f(x)$ y $h(x)$ basado en $g(x)$ y conocimiento a priori de $f(x)$, $h(x)$, y $n(x)$.

La Deconvolución Ciega es un problema mal planteado (en el sentido de Hadamard, Definición 1.4.2): la solución de la ecuación (3.2) puede no depender continuamente de la información, no ser única, o puede no existir. Con un enfoque práctico, una aproximación a la solución es una *estimación* de la solución del problema, de modo que la existencia de la solución puede no ser tomada en cuenta, pues al momento de estimar tenemos estimaciones malas y buenas, la estimación no es única; sin embargo, la no unicidad (puesto que tenemos que escoger la mejor estimación de entre muchas) y la sensibilidad de la solución con respecto del ruido siguen siendo problemas graves.

3.2 Filtrado Inverso.

Si queremos restaurar (deconvolucionar) una imagen, esto es, dada una imagen observada recobrar la imagen que ha sido degradada por distintas causas. Esta degradación, como se ha comentado, surge por ruido, una imagen tomada fuera de foco, ruido debido a imperfecciones de la lente. Necesitamos aplicar un modelo que nos permita recobrar la imagen original. Tal modelo está descrito por medio de las ecuaciones (3.2) y (3.3), donde f es la imagen original, g es la imagen que vemos (imagen observada), h es la función de dispersión del punto ó núcleo y n es ruido aditivo (en general no es aditivo). Normalmente se asume que es ruido blanco Gaussiano (Ver Apéndice A).

El *filtrado Inverso* es un enfoque clásico y directo para resolver la ecuación matricial (3.3), con el fin de encontrar un estimador \hat{f} el cual minimiza la diferencia entre la imagen observada g y la imagen estimada re-degradada $H\hat{f}$. Esto es, minimiza el funcional

$$J(\hat{f}) = \|g - H\hat{f}\|^2, \quad (3.4)$$

esta ecuación provee de un problema de *mínimos cuadrados*, tema visto en el capítulo anterior. Buscamos \hat{f} tal que $J(\hat{f})$ sea mínimo. Observemos que

$$\begin{aligned} \|g - H\hat{f}\|^2 &= (g - H\hat{f})^T (g - H\hat{f}) \\ &= g^T g - g^T H\hat{f} - \hat{f}^T H^T g + \hat{f}^T H^T H\hat{f}. \end{aligned}$$

Para hallar el mínimo de J , derivamos e igualamos a cero

$$\begin{aligned}
 \frac{\partial J(\hat{f})}{\partial \hat{f}} &= \frac{\partial \left[(g - H\hat{f})^T (g - H\hat{f}) \right]}{\partial \hat{f}} \\
 &= \frac{\partial (g^T g - g^T H\hat{f} - \hat{f}^T H^T g + \hat{f}^T H^T H\hat{f})}{\partial \hat{f}} \\
 &= \frac{\partial (g^T g)}{\partial \hat{f}} - \frac{\partial (g^T H\hat{f})}{\partial \hat{f}} - \frac{\partial (\hat{f}^T H^T g)}{\partial \hat{f}} + \frac{\partial (\hat{f}^T H^T H\hat{f})}{\partial \hat{f}} \\
 &= -2H^T g + 2H^T H\hat{f}
 \end{aligned}$$

Es decir

$$H^T g = H^T H\hat{f}.$$

Entonces, el mínimo está dado por

$$\hat{f} = (H^T H)^{-1} H^T g = H^{-1} g,$$

siempre que H^{-1} exista y entonces la solución por mínimos cuadrados esta dada por la solución de la ecuación

$$(H^T H) \hat{f} = H^T g, \quad (3.5)$$

a estas ecuaciones las llamaremos ecuaciones normales (ver secciones 1.5.3 y 2.3.1). El problema que surge de estas ecuaciones normales es el de la amplificación del ruido. En el *dominio de Frecuencia* (al aplicar la transformada de Fourier Discreta ver apéndice A), obtenemos que la ecuación (3.5) está dada por

$$(\bar{H}^* H) \hat{F} = HG,$$

por tanto

$$\begin{aligned}
 \hat{F}(u, v) &= \frac{G(u, v)}{H(u, v)} \\
 &= \frac{H(u, v)F(u, v) + N(u, v)}{H(u, v)} \\
 &= F(u, v) + \frac{N(u, v)}{H(u, v)}.
 \end{aligned}$$

En consecuencia

$$\hat{F}(u, v) = F(u, v) + \frac{N(u, v)}{H(u, v)}, \quad (3.6)$$

donde (u, v) son las coordenadas que corresponden al *dominio de Frecuencia*.

Como hemos ya mencionado una de las desventajas de este filtrado, es que la degradación debida al ruido es amplificada, esto ocurre cuando el término

$H(u, v)$ es cercano al *epsilon de máquina*, para algún $H(u, v)$, donde (u, v) una cordenada en el dominio de frecuencia, ó en el peor de los casos éste se anule. En este caso la estimación \hat{F} de la imagen original es muy mala. En el caotulo 4 se hará patente este problema al implementar este filtro inverso e intentar recuperar una imagen degradada que presente ruido. Más aún en presencia de ruido, el término $1/H(u, v)$ amplifica dicho ruido cuando $H(u, v)$ toma valores muy pequeños en la región de frecuencia. Para estabilizar esta variación se propone

$$\frac{1}{H(u, v)} = \begin{cases} \frac{1}{H(u, v)} & \text{si } |H(u, v)| > \varepsilon \\ 0 & \text{en otro caso} \end{cases}$$

Esto regulariza de alguna manera la solución, de la misma forma que el truncamiento en la descomposición de valores singulares visto en la sección 1.5.1. regulariza la solución obtenida mediante la pseudoinversa.

Para recuperar la imagen, aplicamos la inversa de la Tansformada de Fourier a la estimación en el dominio de frecuencia de la imagen original

$$\text{ifft}(\hat{F}) = \hat{\mathbf{f}} = \text{ifft}\left(\frac{G}{H}\right),$$

donde G es la tranformada de Fourier de g y H es la transformada de Fourier de la función de dispersión del punto h . Una implementación del filtrado inverso en matlab se presenta en el capítulo 4 como se ha comentado.

Las limitaciones del filtrado inverso y pseudoinverso son por la sensibilidad con respecto al ruido. En la restauración de imágenes se usa el *Filtro de Wiener* con el fin de disminuir el fenómeno de amplificación de ruido, este filtro se verá en la siguiente sección.

En términos matemáticos, el problema inverso representado en la ecuación (3.3) es mal planteado en el sentido de Hadamard (definición 1.4.2). En este caso hablamos de la ecuación integral de Fredholm del primer tipo. La naturaleza del mal planteamiento de este problema implica que a pequeños cambios en los datos de entrada, nos devuelve resultados no controlados en la solución. Con respecto al problema discretizado de la ecuación (3.3), el mal planteamiento del problema continuo implica que la matriz \mathbf{H} sea mal condicionada (ver apéndice B). Es interesante notar que a mayor finura en la discretización de \mathbf{H} obtenemos una matriz más mal condicionada, como se vió en el Ejemplo 1.4.5 del capítulo 1. La teoría de regularización puede ser usada para resolver problemas mal planteados. El propósito de la regularización es proveer un análisis numérico de un problema mal planteado a través del análisis de **un problema asociado a este**, que nos arroje aproximaciones significativas del problema mal planteado. Posteriormente, en este capítulo, revisaremos algunas aproximaciones recursivas, directas e iterativas para un problema mal planteado.

Para resolver la ecuación (3.3) por medio de regularización (además del filtrado inverso), en la literatura se usan aproximaciones ya sean estocásticas ó deterministas, para recuperar la imagen original f . En ambos casos, el modelo (3.3) representa información a priori acerca de la solución, la cual puede ser usada para obtener un problema "bien planteado". Una técnica, que usa regularización estocástica es el Filtro de Wiener.

3.3 Filtro de Wiener.

En esta sección desarrollamos una clase de filtro lineal, el cual se conoce como *Filtro de Wiener*. El *filtro de Wiener* es un método de restauración de imágenes (también lo es de señales) en presencia de borrado \mathbf{H} y ruido \mathbf{n} . Este filtro es un mínimo, en el sentido de un problema de optimización, aplicado a una función de error. Dicha función se conoce como función de costo. La función que es comúnmente usada en optimización de filtros, es la **Media Cuadrática del Error** (MCE). Minimizar la función de MCE involucra los conceptos vistos en la sección 2.2.

El problema, recordemos, es recobrar una imagen \mathbf{f} , dada una imagen observada \mathbf{g} ecuación (3.3) con ruido \mathbf{n} ; esto es, a partir de una imagen observada $\mathbf{g}=\mathbf{H}\mathbf{f}+\mathbf{n}$, donde \mathbf{f} y \mathbf{n} son procesos estocásticos independientes.

El problema puede ser planteado como sigue:

Diseñar un filtro que produzca un estimador $\hat{\mathbf{f}}$ de la imagen \mathbf{f} , usando a la imagen disponible \mathbf{g} , tal que $\hat{\mathbf{f}}$ minimice la función de costo de la MCE dada por

$$J = E \left[(\mathbf{f} - \hat{\mathbf{f}})^T (\mathbf{f} - \hat{\mathbf{f}}) \right] = E \left[\mathbf{e}^T \mathbf{e} \right] \quad (3.7)$$

Dependiendo de la forma en que estén relacionadas las imágenes \mathbf{g} y \mathbf{f} , el problema tiene su respectiva dificultad (Procesos es/-ta/-cio/-na/-rios y no estacionarios ver Definición 2.1.13.). Sea $\mathbf{Y} \in \mathbb{R}^{MN \times MN}$ una matriz muestra del filtro deseado, queremos que sea un proceso que pueda estimar \mathbf{f} a través de la imagen observada \mathbf{g} , mediante

$$\hat{\mathbf{f}} = \mathbf{Y}\mathbf{g}, \quad (3.8)$$

La operación de filtrado puede realizarse tanto en el dominio temporal o considerando la transformada de Fourier, en el dominio de frecuencia o dominio de Fourier:

$$\hat{F} = YG, \quad (3.9)$$

y, posteriormente realizar la transformada inversa de Fourier de \hat{F} para obtener la estimación $\hat{\mathbf{f}}$. Es de observar que para el caso de imágenes, los argumentos considerados son bidimensionales. Como se ha dicho, deseamos obtener una

estimación de la imagen original que minimice el error $\mathbf{e} = \mathbf{f} - \hat{\mathbf{f}}$ mediante la función de costo de la MCE,

$$J = E [\mathbf{e}^T \mathbf{e}],$$

donde E denota la esperanza (ver Definición 2.1.17). Basandonos en [44] el problema de optimización anterior es equivalente a este otro

min

$$J(\hat{\mathbf{f}}) = E \left[\text{Tr} \{ (\mathbf{f} - \hat{\mathbf{f}})(\mathbf{f} - \hat{\mathbf{f}})^T \} \right],$$

donde E y Tr representan los operadores lineal esperanza y traza respectivamente ver[45].

Si sustituimos el término $\hat{\mathbf{f}}$ por $\mathbf{Y}\mathbf{g}$ de la ecuación (3.8), y \mathbf{g} por $\mathbf{H}\mathbf{f} + \mathbf{n}$ se tendrá:

$$\begin{aligned} J(\mathbf{Y}) &= E \left[\text{Tr} \{ (\mathbf{f} - \hat{\mathbf{f}})(\mathbf{f} - \hat{\mathbf{f}})^T \} \right] \\ &= E \left[\text{Tr} \{ (\mathbf{f} - (\mathbf{Y}\mathbf{g}))(\mathbf{f} - (\mathbf{Y}\mathbf{g}))^T \} \right] \\ &= E \left[\text{Tr} \{ (\mathbf{f} - (\mathbf{Y}(\mathbf{H}\mathbf{f} + \mathbf{n}))) (\mathbf{f} - (\mathbf{Y}(\mathbf{H}\mathbf{f} + \mathbf{n})))^T \} \right] \\ &= E \left[\text{Tr} \{ (\mathbf{f} - (\mathbf{Y}\mathbf{H}\mathbf{f} + \mathbf{Y}\mathbf{n})) (\mathbf{f} - (\mathbf{Y}\mathbf{H}\mathbf{f} + \mathbf{Y}\mathbf{n}))^T \} \right] \\ &= E \left[\text{Tr} \{ (\mathbf{f} - \mathbf{Y}\mathbf{H}\mathbf{f} - \mathbf{Y}\mathbf{n}) (\mathbf{f} - \mathbf{Y}\mathbf{H}\mathbf{f} - \mathbf{Y}\mathbf{n})^T \} \right] \\ &= E \left[\text{Tr} \left\{ (\mathbf{f} - \mathbf{Y}\mathbf{H}\mathbf{f} - \mathbf{Y}\mathbf{n})(\mathbf{f}^T - (\mathbf{Y}\mathbf{H}\mathbf{f})^T - (\mathbf{Y}\mathbf{n})^T) \right\} \right] \end{aligned}$$

Puesto que los operadores E y Tr son lineales, se pueden intercambiar. Además $E(\mathbf{n}\mathbf{f}^T) = E(\mathbf{f}\mathbf{n}^T) = 0$, pues hemos supuesto que \mathbf{f} y \mathbf{n} son independientes. Recordemos que $\text{Tr}(A) = \text{Tr}(A^T)$.

Al expandir $J(\mathbf{Y})$ obtenemos

$$J(\mathbf{Y}) = \text{Tr} \left(\mathbf{R}_f - 2\mathbf{Y}\mathbf{H}\mathbf{R}_f + \mathbf{Y}\mathbf{H}\mathbf{R}_f\mathbf{H}^T\mathbf{Y}^T + \mathbf{Y}\mathbf{R}_n\mathbf{Y}^T \right), \quad (3.10)$$

donde \mathbf{R}_f y \mathbf{R}_n son matrices de autocorrelación (definición 2.2.4). Derivando $J(\mathbf{Y})$ respecto a \mathbf{Y} e igualando al vector cero obtenemos

$$-2\mathbf{R}_f\mathbf{H}^T + 2\mathbf{Y}\mathbf{H}\mathbf{R}_f\mathbf{H}^T + 2\mathbf{Y}\mathbf{R}_n = \mathbf{0}$$

Entonces, la matriz \mathbf{Y} esta dada por

$$\mathbf{Y} = \mathbf{R}_f\mathbf{H}^T \left(\mathbf{H}\mathbf{R}_f\mathbf{H}^T + \mathbf{R}_n \right)^{-1} \quad (3.11)$$

Las matrices de autocorrelación $\mathbf{R}_f, \mathbf{R}_n \in \mathbb{R}^{MN \times MN}$ son matrices de Toeplitz [46]. Sin embargo, para un tratamiento matemático suponemos a \mathbf{R}_f en forma

circulante la cual puede ser diagonalizada por una transformada de Fourier bidimensional. En el caso de restauración de imágenes aplicamos esta matriz \mathbf{Y} a (3.8), por lo que

$$\begin{aligned}\hat{\mathbf{f}} &= \mathbf{Y}\mathbf{g} \\ &= \mathbf{R}_f\mathbf{H}^T \left(\mathbf{H}\mathbf{R}_f\mathbf{H}^T + \mathbf{R}_n \right)^{-1} \mathbf{g}\end{aligned}$$

Por lo tanto en el dominio de frecuencia (aplicando la transformada de Fourier), se tiene

$$\hat{F}(u, v) = Y(u, v)G(u, v) = \left(\frac{H^*(u, v)}{|H(u, v)|^2 + \frac{R_n(u, v)}{R_f(u, v)}} \right) G(u, v), \quad (3.12)$$

para $u \in \{0, 1, \dots, M-1\}$ y $v \in \{0, 1, \dots, N-1\}$. $Y(u, v)$, $H(u, v)$, $R_f(u, v)$ y $R_n(u, v)$ son la transformada de Fourier de \mathbf{Y} , \mathbf{H} , \mathbf{R}_f y \mathbf{R}_n respectivamente, $H^*(u, v)$ es el complejo conjugado de $H(u, v)$. Este filtro es conocido como el *filtro de Wiener*.

El cociente $R_n(u, v)/R_f(u, v)$ es la razón entre la señal y el ruido que denotaremos $SNR(f) = R_n(u, v)/R_f(u, v)$. En el caso de que el ruido se anule (i.e. la razón entre señal y ruido se anula), el *filtro de Wiener* es precisamente el filtro inverso. En esta expresión, $1/H(u, v)$ es el mismo que obtuvimos en el filtrado inverso.

Sin embargo, cuando el ruido a cierta frecuencia aumenta entonces la razón entre la señal y el ruido tiende a aumentar. Esto quiere decir que el filtro de Wiener atenúa las frecuencias que dependen de la razón entre la señal y el ruido. Si no conocemos bien los parámetros del ruido, se puede introducir un término K que regulariza el filtro.

$$\hat{F}(u, v) = \left(\frac{H^*(u, v)}{|H(u, v)|^2 + K} \right) G(u, v)$$

Este se considera como un parámetro libre que se puede ajustar para obtener un mejor resultado, aunque se recomienda estimar $SNR(f)$.

La ecuación (3.12) anterior para el filtro de Wiener necesita que conozcamos $H(u, v)$, así como del ruido $R_n(u, v)$. Usualmente, no se tiene acceso a estas cantidades exactas, pero quizá sí podemos obtener una buena estimación de los mismos. Por ejemplo, en el caso de fotografías, la imagen original posee de manera típica frecuencias bajas fuertes y frecuencias altas débiles y en muchos casos el ruido tendrá frecuencias relativamente planas. Una implementación está en el Capítulo 4, otra en el apéndice E y una comparación entre filtrado inverso y filtro Wiener se muestra en la figura 3.1.



Figura 3.1: Métodos.

3.3.1 Comentario; Regularización Determinista.

El uso de información a priori determinista acerca de la imagen puede ser usada para regularizar el problema mal planteado. Por ejemplo el problema de restauración usando mínimos cuadrados puede ser formulado de tal manera que tomamos a \hat{f} como un mínimo del Lagrangiano

$$\hat{f} = \arg \min \left[\|g - H\hat{f}\|^2 + \alpha \|C\hat{f}\|^2 \right], \quad (3.13)$$

donde el término $C\hat{f}$ representa un filtro paso alto (ver apéndice A). Este es esencialmente una restricción de suavizado el cual sugiere que la imagen tenga un límite de alta frecuencia, y entonces esta aproximación minimice la cantidad de energía de paso alto en la imagen restaurada. El uso del operador C provee de un camino para reducir los efectos de valores singulares de H pequeños, que ocurren en las altas frecuencias. El típico operador C que se usa en el Lagrangiano anterior [13], está dado por

$$C = \begin{pmatrix} 0.00 & -0.25 & 0.00 \\ -0.25 & -1.00 & -0.25 \\ 0.00 & -0.25 & 0.00 \end{pmatrix}.$$

En la ecuación del Lagrangiano el término α representa un multiplicador de Lagrange, el cual es el parámetro de regularización en la ecuación (1.30), que controla la fidelidad y el suavizado de la imagen. En la sección 1.5.4, consideremos la ecuación (1.30), la cual nos da la expresión para la solución del problema mal planteado (3.3) y que está dada por

$$\hat{f} = (H^T H + \alpha C^T C)^{-1} H^T g.$$

La elección del parámetro α forma parte de otra investigación en la cual se hace necesario buscar técnicas óptimas para encontrar α , dada información a priori del ruido y borrado. Con este método, una restauración de la imagen f estaría definida a través de la intersección de dos elipsoides definidos por

$$\begin{aligned} Q_{f|g} &= \{f \mid \|g - Hf\|^2 \leq \varepsilon^2\}, \\ Q_f &= \{f \mid \|Cf\|^2 \leq \epsilon^2\}. \end{aligned}$$

En este caso se usa el parámetro $\alpha = (\varepsilon/\epsilon)^2$. La misma solución se obtiene con la llamada regularización de *Miller* en [13]. Otras aproximaciones estocásticas tienen que ver con el uso de estimadores de máxima verosimilitud para resolver el problema inverso. Algunos de estos trabajos incluyen formulaciones como el algoritmo Esperanza Maximización (EM) para identificación del ruido y deconvolución de la imagen. Tal técnica es efectiva cuando el operador de borrado y el ruido son parcialmente conocidos, construyendo un proceso de identificación iterativamente.

3.4 Algoritmo de Lucy-Richarson.

Analizaremos en esta sección el problema de deconvolución de imágenes mediante el *algoritmo de Lucy-Richardson*. Usaremos en esencia el algoritmo EM (visto en la sección 2.3.3), para encontrar una estimación de la imagen ideal asociada a la imagen borrosa. El procedimiento que seguiremos será el visto en la sección 2.3.3.

Uno de los métodos más usados en restauración de imágenes en astronomía es el *Algoritmo de Lucy-Richardson* (L-R). Este algoritmo inicialmente es obtenido del teorema de Bayes. Posteriormente se hace uso del *algoritmo Esperanza Maximización* (EM), ver sección 2.3.3 de este trabajo o ver [14]. El siguiente desarrollo es tomado de [41]. Consideremos un sistema lineal con parámetros positivos, sujetos a restricciones positivas en la solución, tales problemas con restricciones positivas (LININPOS por sus siglas en inglés) pueden ser interpretadas como problemas de estimación estadística de información incompleta basada en un número grande de muestras, y que la estimación de máxima verosimilitud, aunada al algoritmo EM proveen un método sencillo para resolver el problema de deconvolución.

En aplicaciones de reconstrucción de imágenes este algoritmo es útil para encontrar una estimación de la imagen \mathbf{f} al sistema

$$H\mathbf{f} = \mathbf{g}, \quad (3.14)$$

donde $H \in \mathbb{R}^{m \times n}$ y el vector observado $\mathbf{g} \in \mathbb{R}^m$ tienen componentes no negativos (para un tratamiento matricial, la notación manejada hasta ahora la extendemos a una matriz H no cuadrada). Por supuesto tal estimación no necesita ser exacta, así que intentamos reducir al mínimo una cierta medida de la **discrepancia** (Ver apéndice C) entre los datos \mathbf{g} y el modelo $H\mathbf{f}$. Lo anterior lo podemos hacer con el funcional generalizado de Tikhonov (Ver Capítulo 1) para (3.10). $(T_\alpha(f; g) = \rho(H(f), g) + \alpha J(f))$, donde $\alpha > 0$ es el parámetro de regularización, J funcional de penalización ó funcional de regularización y ρ es llamada el funcional de discrepancia). El propósito del funcional de discrepancia es minimizar la diferencia de $H\mathbf{f}$ con \mathbf{g} . Aquí usamos la discrepancia de Kullback-Leibler (Ver apéndice C y [8]). El objetivo es minimizar dicha discrepancia, esto es minimizar la diferencia entre dos medidas del mismo tipo, en este caso minimizar la diferencia entre el vector $H\mathbf{f}$ y el vector \mathbf{g} . Este problema se reduce a un problema de optimización, el cual se puede escribir de la siguiente forma

min

$$\rho_{HL}(\mathbf{g}, H\mathbf{f}) = \langle \mathbf{g}, \log(g/H\mathbf{f}) \rangle = \sum_{i=1}^m g_i (\log g_i - \log [H\mathbf{f}]_i), \quad (3.15)$$

sujeto a

$$[H\mathbf{f}]_i, g_i, h_{ij}, f_j \geq 0, \quad j = 1, \dots, n \quad (3.16)$$

$$\sum_{i=1}^m [H\mathbf{f}]_i, \sum_{i=1}^m g_i, \sum_{i=1}^m h_{ij}, \sum_{j=1}^n f_j = 1, \quad i = 1, \dots, m, \quad (3.17)$$

esto es $\sum_{i=1}^m [H\mathbf{f}]_i = 1$, $\sum_{i=1}^m g_i = 1$, $\sum_{i=1}^m h_{ij} = 1$ y $\sum_{j=1}^n f_j = 1$.

Observemos que el problema de minimizar (3.15) con las condiciones (3.16) y (3.17) es equivalente al siguiente problema de maximización (por propiedades de logaritmos)

max

$$J(\mathbf{f}) = \sum_{i=1}^m g_i \log [H\mathbf{f}]_i \quad (3.18)$$

sujeto a

$$\begin{aligned} [H\mathbf{f}]_i, g_i, h_{ij}, f_j &\geq 0, & j = 1, \dots, n \\ \sum_{i=1}^m [H\mathbf{f}]_i, \sum_{i=1}^m g_i, \sum_{i=1}^m h_{ij}, \sum_{j=1}^n f_j &= 1, & i = 1, \dots, m \end{aligned}$$

Como vimos en la sección 2.3.3 el algoritmo EM hace uso de dos variables aleatorias discretas, una que representa a los datos observados incompletos (conocimiento parcial de estos) y otra variable aleatoria que representa a los datos no observados completos. También el algoritmo hace uso de una función de distribución conjunta. Sean X y Y variables aleatorias que representan a los datos completos e incompletos con índices $j = 1, \dots, n$, $i = 1, \dots, m$ respectivamente. Considerando las condiciones (3.16) y (3.17) para $[H\mathbf{f}]_i$ se tiene que

$$P\{X = j, Y = i\} = p_{X,Y}(j, i; \mathbf{f}) = h_{ij} f_j, \quad (3.19)$$

es una función de distribución conjunta, donde el parámetro de interés es estimar \mathbf{f} (ver Definición 2.3.10). Las restricciones (3.16) y (3.17) garantizan que $p_{X,Y}(j, i; \mathbf{f})$ es una función de distribución ([14]), por lo que la función de probabilidad marginal para la variable aleatoria Y esta dada de acuerdo a la definición 2.3.10.b. por

$$\begin{aligned} p_Y(i, \mathbf{f}) &= \sum_{j=1}^n P_{X,Y}(j, i; \mathbf{f}) \\ &= \sum_{j=1}^n h_{ij} f_j \\ &= [H\mathbf{f}]_i. \end{aligned}$$

En resumen

$$p_Y(i; \mathbf{f}) = [H\mathbf{f}]_i \quad (3.20)$$

Sea $\mathbf{g} = (g_1, \dots, g_m) \in \mathbb{R}^m$ un vector que contiene información, sujeto a las restricciones (3.16) y (3.17). Si suponemos que cada g_i es un número racional (Ver [12]), entonces existe $r \in \mathbb{N}$ tal que

$$N_i = rg_i, \quad (3.21)$$

donde N_i es un entero para cada i . Tomamos r copias independientes de la variable aleatoria Y , idénticamente distribuidas, para formar el vector aleatorio \mathbf{Y} . Sea $\mathbf{y} = (y_1, \dots, y_r)$ un vector muestra para el cual el número N_i nos da el número de índices $k \in \{1, 2, \dots, r\}$, que cumple con la igualdad $y_k = i$ (N_i es el número de veces que y_k es igual a k). Esto es, consideremos la *delta de Kronecker*

$$\delta_{y_k, i} = \begin{cases} 1 & \text{si } y_k = i \\ 0 & \text{si } y_k \neq i, \end{cases}$$

de tal manera que $\sum_i \delta_{y_k, i} = 1$ (para el índice k fijo). Entonces la función de verosimilitud para Y dada la información \mathbf{y} y el parámetro \mathbf{f} (ver ecuación (2.16) y la sección 2.3.3) está dada por

$$\begin{aligned} l_Y(\mathbf{f}; \mathbf{y}) &= \sum_{k=1}^r \log p_Y(y_k; \mathbf{f}) \\ &= \sum_{k=1}^r (1) \log p_Y(y_k; \mathbf{f}) \\ &= \sum_{k=1}^r \left(\sum_{i=1}^m \delta_{y_k, i} \right) \log p_Y(y_k; \mathbf{f}), \end{aligned}$$

dado que son sumas finitas se pueden intercambiar

$$\begin{aligned} \sum_{k=1}^r \left(\sum_{i=1}^m \delta_{y_k, i} \right) \log p_Y(y_k; \mathbf{f}) &= \sum_{i=1}^m \sum_{k=1}^r \delta_{y_k, i} \log p_Y(y_k; \mathbf{f}) \\ &= \sum_{i=1}^m N_i \log p_Y(y_k; \mathbf{f}) \quad y_k \text{ aleatorio} \\ &= r \sum_{i=1}^m g_i \log ([H\mathbf{f}]_i), \end{aligned}$$

la segunda igualdad ocurre porque N_i nos da el número de índices $k \in \{1, 2, \dots, r\}$, que cumple con la igualdad $y_k = i$. En consecuencia

$$l_Y(\mathbf{f}; \mathbf{y}) = r \sum_{i=1}^m g_i \log ([H\mathbf{f}]_i).$$

De una forma similar, podemos construir r copias de X y obtener un vector aleatorio \mathbf{X} . Sea (\mathbf{x}, \mathbf{y}) una observación con (x_k, y_k) , $k = 1, \dots, r$, de los vectores aleatorios \mathbf{X} y \mathbf{Y} para los cuales $N_{ij}(\mathbf{x}, \mathbf{y})$ denota el número de índices k tal que $y_k = i$ y $x_k = j$. Notemos que para cada i ,

$$\sum_{j=1}^n N_{ij}(\mathbf{x}, \mathbf{y}) = N_i = rg_i, \quad (3.22)$$

el número N_i de veces que $y_k = i$. Usando una idea análoga podemos obtener la función de verosimilitud para los datos completos \mathbf{X} e \mathbf{Y} , la cual está dada por

$$l_{\mathbf{X}, \mathbf{Y}}(\mathbf{f}; \mathbf{x}, \mathbf{y}) = \sum_{i=1}^m \sum_{j=1}^n N_{ij}(\mathbf{x}, \mathbf{y}) [\log h_{ij} + \log f_j];$$

es decir,

$$l_{\mathbf{X}, \mathbf{Y}}(\mathbf{f}; \mathbf{x}, \mathbf{y}) = \sum_{i=1}^m \sum_{j=1}^n N_{ij}(\mathbf{x}, \mathbf{y}) [\log h_{ij} f_j].$$

de (2.17), (2.18) y (3.19) tenemos que la probabilidad condicional para X dada Y está dada por

$$p_{X|Y}(j|i; \mathbf{f}^\nu) = \frac{h_{ij} f_j^\nu}{\sum_{l=1}^n h_{il} f_l^\nu} := \hat{p}_{ij}^\nu, \quad (3.23)$$

donde \mathbf{f}_j^ν denota el j -ésimo componente del vector \mathbf{f}^ν el ν -ésimo iterado obtenido. Usando (2.20), para estimar \mathbf{f}^ν llegamos al primer paso del algoritmo *EM*.

Primer Paso: Esperanza.

$$\begin{aligned} Q(\mathbf{f}; \mathbf{y}; \mathbf{f}_\nu) &= \sum_{\mathbf{x}} [l_{\mathbf{X}, \mathbf{Y}}(\mathbf{f}; \mathbf{x}, \mathbf{y})] p_{X|Y}(j|i; \mathbf{f}^\nu) \\ &= \sum_{\mathbf{x}} \left[\sum_{i=1}^m \sum_{j=1}^n N_{ij}(\mathbf{x}, \mathbf{y}) [\log h_{ij} + \log f_j^\nu] \right] \hat{p}_{ij}^\nu, \end{aligned}$$

por la ecuación (2.22) tenemos que

$$\begin{aligned} \sum_{\mathbf{x}} \left[\sum_{i=1}^m \sum_{j=1}^n N_{ij}(\mathbf{x}, \mathbf{y}) [\log h_{ij} + \log f_j^\nu] \right] \hat{p}_{ij}^\nu &= \sum_{j=1}^n \left[\sum_{i=1}^m rg_i [\log h_{ij} + \log f_j^\nu] \right] \hat{p}_{ij}^\nu \\ &= \sum_{i=1}^m \sum_{j=1}^n rg_i [\log h_{ij} + \log f_j^\nu] \hat{p}_{ij}^\nu \\ &= \sum_{i=1}^m \sum_{j=1}^n rg_i [\log h_{ij} f_j^\nu] \hat{p}_{ij}^\nu. \end{aligned}$$

Esto completa el primer paso; Esperanza, del algoritmo *EM*:

$$Q(\mathbf{f}|y; \mathbf{f}_\nu) = \sum_{i=1}^m \sum_{j=1}^n r g_i [\log h_{ij} f_j^\nu] \hat{p}_{ij}^\nu.$$

Segundo Paso: Maximización.

En este paso, maximizaremos a $Q(\mathbf{f}|y; \mathbf{f}_\nu)$ con el objetivo de encontrar un buen estimador para la imagen desconocida \mathbf{f} , sujeto a las condiciones (3.16) y (3.17).

Tomemos la función $Q(\mathbf{f}|y; \mathbf{f}_\nu)$, derivándola con respecto de f_l e igualando a cero tenemos

$$\frac{\partial}{\partial f_l} Q(\mathbf{f}|y; \mathbf{f}_\nu) = \frac{\partial}{\partial f_l} \sum_{i=1}^m \sum_{j=1}^n r g_i [\log h_{ij} + \log f_j^\nu] \hat{p}_{ij}^\nu = 0. \quad (3.24)$$

Notemos que, usando la restricción sobre f_j que imponen las ecuaciones (3.16) y (3.17), esto es $\sum_{j=1}^n f_j = 1$ se tiene que la ecuación (3.24) es equivalente a la ecuación siguiente

$$\frac{\partial}{\partial f_l} \left[Q(\mathbf{f}|y; \mathbf{f}_\nu) - \lambda \left(\sum_{j=1}^n f_j - 1 \right) \right] = 0, \quad (3.25)$$

donde $\lambda \in \mathbb{R}$ distinto de cero. De esta forma

$$\begin{aligned} \frac{\partial}{\partial f_l} Q(\mathbf{f}|y; \mathbf{f}_\nu) &= \frac{\partial}{\partial f_l} \sum_{i=1}^m \sum_{j=1}^n r g_i [\log h_{ij} + \log f_j^\nu] \hat{p}_{ij}^\nu \\ &= \sum_{i=1}^m r g_i \left[\frac{1}{f_l^\nu} \right] \hat{p}_{il}^\nu, \end{aligned}$$

sustituyendo en (3.25)

$$\begin{aligned} \frac{\partial}{\partial f_l} \left[Q(\mathbf{f}|y; \mathbf{f}_\nu) - \lambda \left(\sum_{j=1}^n f_j - 1 \right) \right] &= \left[\frac{\partial}{\partial f_l} Q(\mathbf{f}|y; \mathbf{f}_\nu) - \lambda \left(\sum_{j=1}^n \frac{\partial}{\partial f_l} f_j - \frac{\partial}{\partial f_l} 1 \right) \right] \\ &= \sum_{i=1}^m r g_i \left[\frac{1}{f_l^\nu} \right] \hat{p}_{il}^\nu - \lambda \\ &= 0, \end{aligned}$$

por tanto

$$f_l = \frac{r}{\lambda} \sum_{i=1}^m g_i \hat{p}_{il}^\nu. \quad (3.26)$$

Al considerar nuevamente la restricción (3.17) para f_j , tenemos que

$$\begin{aligned} 1 &= \sum_{j=1}^n f_j \\ &= \sum_{j=1}^n \frac{r}{\lambda} \sum_{i=1}^m g_i \hat{p}_{ij}^\nu, \end{aligned}$$

en la segunda igualdad se usa la ecuación (3.26). Reacomodando algunos términos como sigue

$$\begin{aligned} \sum_{j=1}^n \frac{r}{\lambda} \sum_{i=1}^m g_i \hat{p}_{ij}^\nu &= \frac{r}{\lambda} \sum_{j=1}^n \sum_{i=1}^m g_i \hat{p}_{ij}^\nu \\ &= \frac{r}{\lambda} \sum_{i=1}^m \sum_{j=1}^n g_i \hat{p}_{ij}^\nu \\ &= \frac{r}{\lambda} \sum_{i=1}^m g_i \left(\sum_{j=1}^n \hat{p}_{ij}^\nu \right), \end{aligned}$$

y usando (3.23)

$$\begin{aligned} \frac{r}{\lambda} \sum_{i=1}^m g_i \left(\sum_{j=1}^n \hat{p}_{ij}^\nu \right) &= \frac{r}{\lambda} \sum_{i=1}^m g_i \left(\sum_{j=1}^n \frac{h_{ij} f_j^\nu}{\sum_{l=1}^n h_{il} f_l^\nu} \right) \\ &= \frac{r}{\lambda} \sum_{i=1}^m g_i \left(\frac{\sum_{j=1}^n h_{ij} f_j^\nu}{\sum_{l=1}^n h_{il} f_l^\nu} \right) \\ &= \frac{r}{\lambda} \sum_{i=1}^m g_i \\ &= \frac{r}{\lambda} \end{aligned}$$

obtenemos que

$$1 = \frac{r}{\lambda}, \quad \lambda \neq 0.$$

Por lo tanto $r = \lambda$, entonces $\lambda \in \mathbb{N}$ y de las ecuaciones (3.26) y (3.23)

$$\begin{aligned} f_j &= \sum_{i=1}^m g_i \hat{p}_{ij}^\nu \\ &= \sum_{i=1}^m g_i \frac{h_{ij} f_j^\nu}{\sum_{l=1}^n h_{il} f_l^\nu} \\ &= f_j^\nu \sum_{i=1}^m h_{ij} \left(\frac{g_i}{\sum_{l=1}^n h_{il} f_l^\nu} \right). \end{aligned}$$

Obteniendo así el estimador $\hat{\mathbf{f}}$ de \mathbf{f} , por el algoritmo EM , que produce la $(\nu + 1)$ -ésima iteración dada por:

$$f_j^{\nu+1} = f_j^\nu \sum_{i=1}^m h_{ij} \left(\frac{g_i}{\sum_{l=1}^n h_{il} f_l^\nu} \right), \quad j = 1, \dots, n. \quad (3.27)$$

Podemos ver de la ecuación (3.26) que la restricción $f_j^\nu \geq 0$, para la ecuación (3.14) se cumple, para cada ν , lo que nos garantiza una solución no negativa.

Capítulo 4

Implementación.

Ya hemos visto tres métodos para la deconvolución de imágenes (recuperación de imágenes), debido a un borrado y ruido, este problema, como vimos es un problema mal planteado. En el capítulo anterior hemos obtenido teóricamente tres métodos para la deconvolución de imágenes, en este capítulo veremos la utilidad de todo el desarrollo hecho hasta ahora.

Primero veremos los códigos implementados en matlab con los cuales compararemos los tres métodos.

4.1 Código en Matlab.

El *código fuente* de un programa informático (o software) es un conjunto de líneas de texto que son las instrucciones que debe seguir la computadora para ejecutar dicho programa. Por tanto, en el código fuente de un programa está descrito por completo su funcionamiento.

Se escogió Matlab debido a su eficiencia y capacidad para manipular imágenes, como el agregar borrado y ruido, de forma vectorial y con un lenguaje no tan complicado en forma estructurada. Matlab es un entorno que nos permite incluir nuestros programas en un entorno gráfico llamado GUI (Graphical User Interface) de forma simple. Primero veremos la implementación de los métodos en Matlab programación estructurada, para posteriormente incluir estas funciones en el GUI; `Comparacion.m` y `Comparacion.fig`.

4.1.1 Filtrado Inverso en Matlab.

En el siguiente programa en matlab los argumentos de entrada de la función (`FiltroInverso`) son; la imagen degradada (`im`), el tamaño de borrado de la imagen (`Tam`), el ángulo de borrado (`Ang`). Lo que obtenemos al ejecutar este

programa es una imagen restaurada (*imres*).

```
function imres=FiltroInverso(im, Tam, Ang)
%Funcion para restaurar una imagen usando el filtrado inverso
%Argumentos de entrada: im, Tam, Ang.
%Argumentos de salida: imres.
%
%im: Imagen de entrada.
%Ang: Angulo de borrado. El angulo en el cual es borrada la imagen.
%Tam: Tama\~no de borrado. El tamao es el numero
%      de pixeles por el cual la imagen es borrada.
%imres: Imagen restaurada.

%Convirtiendo a dominio de frecuencia.
f=fft2(im);

%Creando la funcion de dispersion del punto PSF usando una funcion
de matlab.
PSF = fspecial('motion',Tam, Ang);

%Convirtiendo la PSF al tama\~no deseado usando la funcion de
%dispersion del punto.
OTF = psf2otf(PSF, size(f));
H=OTF;

%Para evitar division por cero se usa la regularizaci\~on
%por truncamiento siguiente.
for i = 1:size(H, 1)
    for j = 1:size(H, 2)
        if H(i, j) == 0
            H(i, j) = 0.000001;
        end
    end
end
end
%Restaurando la imagen usando filtrado inverso.
fd = f./H;

%Convirtiendo a dominio espacial.
imres = ifft2(fd);
```

4.1.2 Filtro de Wiener en Matlab.

En el siguiente programa en matlab los argumentos de entrada de la función (`FiltroWiener`) son; la imagen degradada (`im`), el tamaño de borrado de la imagen (`Tam`), el ángulo de borrado (`Ang`) y la razón señal-ruido (`SNR`). Lo que obtenemos al ejecutar este programa es una imagen restaurada (`imres`).

```
function imres=FiltroWiener(im, Tam, Ang, SNR)
%Funcion para restaurar la imagen usando el filtrado de wiener.
%Argumentos de entrada: im, Tam, Ang, SNR.
%Argumentos de salida: imres.
%
%im: Imagen de entrada.
%Ang: Angulo de borrado. El angulo en el cual es borrada la imagen.
%Tam: Tamao de borrado. El tamao es el numero
%      de pixeles por el cual la imagen es borrada.
%SNR: Se trata de la relacin seal / ruido de la imagen
%imres: Imagen restaurada.

%Preprocesamiento
%Usamos filtro de media de matlab
im=medfilt2(abs(im));

%Convirtiendo al dominio de frecuencia.
f=fft2(im);

%Crea la funcion de dispersion del punto.
PSF=fspecial('motion',Tam, Ang);

%Convirtiendo la PSF al tamaño deseado usando la funcion
%de dispersion del punto.
OTF=psf2otf(PSF,size(f));
H=OTF;
HC=conj(H);
modH=H.*HC;

%Para evitar division por cero se usa la regularizaci'on
%por truncamiento siguiente.
for i=1:size(H, 1)
    for j=1:size(H, 2)
        if H(i, j)==0
            H(i, j)=0.000001;
        end
    end
end
```

```

end

%Formula de recuperacion de imagen usando Filtro de Wiener.
Filtro=((modH./(modH+SNR))./(H)).*im;\ Filtro=ifft2(Filtro);
%Imagen Restaurada.
imres=Filtro;

```

4.1.3 Algoritmo de Lucy Richardson en Matlab.

En el siguiente programa en matlab los argumentos de entrada de la función (LucyRichardson) son; la imagen degradada (*im*), el tamaño de borrado de la imagen (*Tam*), el ángulo de borrado (*Ang*) y el número de iteraciones (*iteraciones*). Lo que obtenemos al ejecutar este programa es una imagen restaurada (*imres*).

```

function imres=LucyRichardson(im, Tam, Ang, iteraciones)
%Funcion para deconvolucionar una imagen usando Lucy-Richardson
%Entrada: im, Tam, Ang, iteraciones.
%Salida: imres
%
%im: imagen de entrada.
%Ang: Angulo de borrado. El angulo en el cual la imagen es borrada.
%Tam: Tama\~no de borrado. El tama\~no es el numero
%      de pixeles por el cual la imagen es borrada.
%iteraciones: El numero de iteraciones.
%imres: La imagen restaurada.
%
%Ejemplo:
%      imres=Lucy(imagen, Tam, Ang, iteraciones);
%      Esta funcion toma una imagen, tama\~no de borrado Tam,
%      angulo de borrado Ang y numero de iteraciones
%      como argumento de entrada y imres es la imagen restaurada

%Preprocesamiento
%Usando Filtro de media, aunque podriamos usar filtro de Wiener
im=medfilt2(abs(im));

%Imagen inicial fo en el algoritmo de Lucy Richarson
est=im;

%Funcion de dispersion del punto
PSF=fspecial('motion',Tam, Ang);

```

```
%Convirtiendo la PSF a tamaño deseado en dominio de frecuencia
OTF=psf2otf(PSF,size(im));
H=OTF;

%Para evitar la división por cero
for i=1:size(H, 1)
    for j=1:size(H, 2)
        if H(i, j)==0
            H(i, j)=0.000001;
        end
    end
end
end

i=1; while i<=iteraciones
    %Convirtiendo a dominio de frecuencia mediante fft en matlab
    fest=fft2(est);

    %Multiplicando H con la imagen en el dominio de frecuencia
    fblest=H.*fest;

    %Convirtiendo a dominio espacial
    ifblest=ifft2(fblest);

    %Calculando la razón de la imagen inicial y la imagen borrada
    raz=im./ifblest;

    %Convirtiendo la razón en dominio de frecuencia
    razf=fft2(raz);

    %Correlación
    corrvec=H.*razf;

    %Convirtiendo la correlación a dominio espacial
    correspa=ifft2(corrvec);

    %Multiplicando la correlación y la imagen inicial para
    %encontrar la siguiente estimación
    correspa=est.*correspa;
    est=correspa;

    i=i+1;
end
```

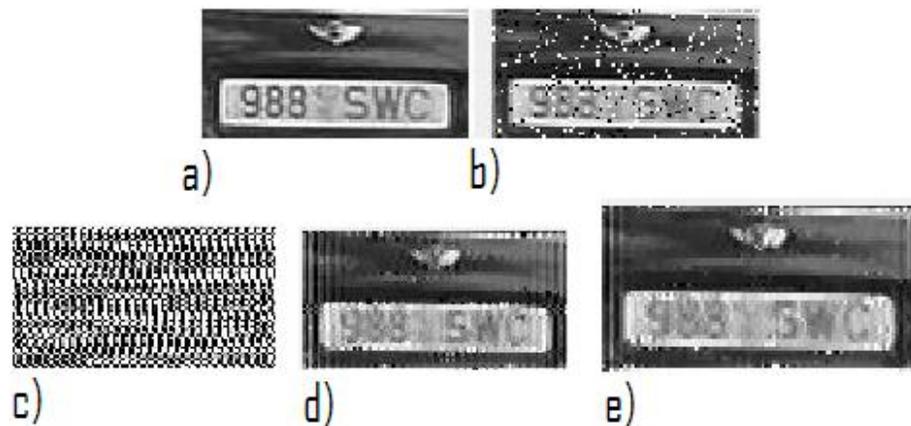


Figura 4.1: Comparación, imagen a. original, b. degradada (ángulo=0, tamaño=7,ruido: sal y pimienta=0.1), c. filtrado inverso, d. filtro Wiener y e. Lucy-Richardson 30 iteraciones.

```
%Imagen restaurada
imres=abs(est);
```

4.1.4 GUI-Comparación.

En la figura 4.11 se muestra una tabla que indica la comparación de los errores numéricos de los métodos vistos en el presente trabajo. La primer columna de la tabla indica el tamaño de borrado, en la segunda el ángulo de borrado, en la tercera la saturación de ruido sal y pimienta (usamos la función del toolbox de matlab), la cuarta y quinta indican la varianza y media del ruido tipo gaussiano (usamos la función del toolbox de matlab), en la sexta se muestra el error de la imagen real con la degradada mediante la norma de Frobenius $e(\hat{\mathbf{f}}) = \|\mathbf{f} - \hat{\mathbf{f}}\|_{\mathbf{F}}$, en la séptima se muestra el error correspondiente al filtro inverso, la octava el error correspondiente al filtro de Wiener y por último en la novena se muestra el error del algoritmo de Lucy-Richardson.

4.2 Miscelánea.

Los siguientes programas, son tomados de trabajos hechos por otros autores, por ejemplo de [46] y algunas páginas de internet más, su aportación a este trabajo fue fundamental para comprender algunos aspectos en la programación de los métodos expuestos.

```
Algoritmo EM. em2=function(y1,y2,y3,y4,tol,start)
{#Algoritmo EM par un modelo multinomial usado en genetica
#y1,y2,y3,y4 son las frecuencias observadas
```

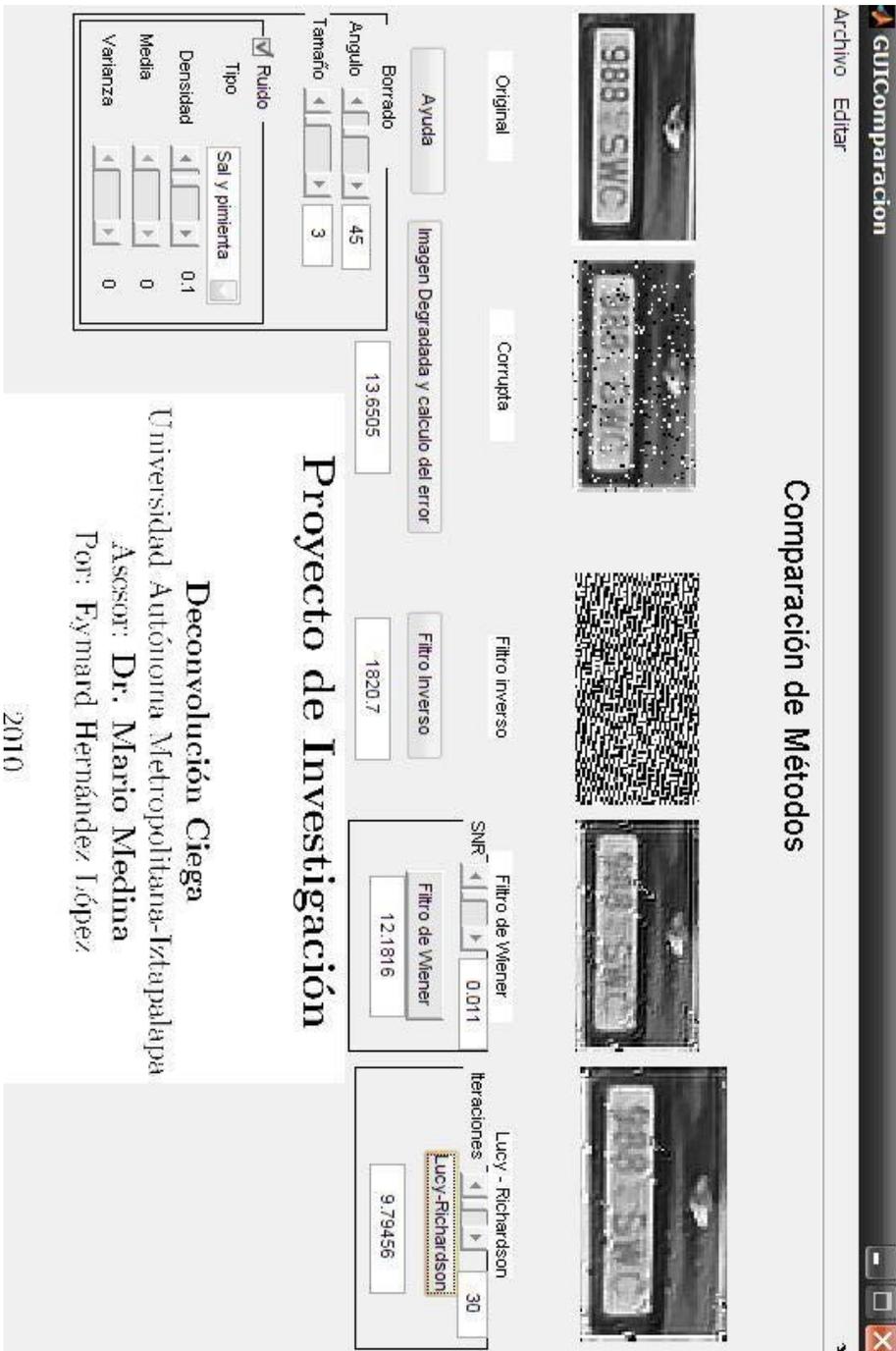


Figura 4.2: Comparación

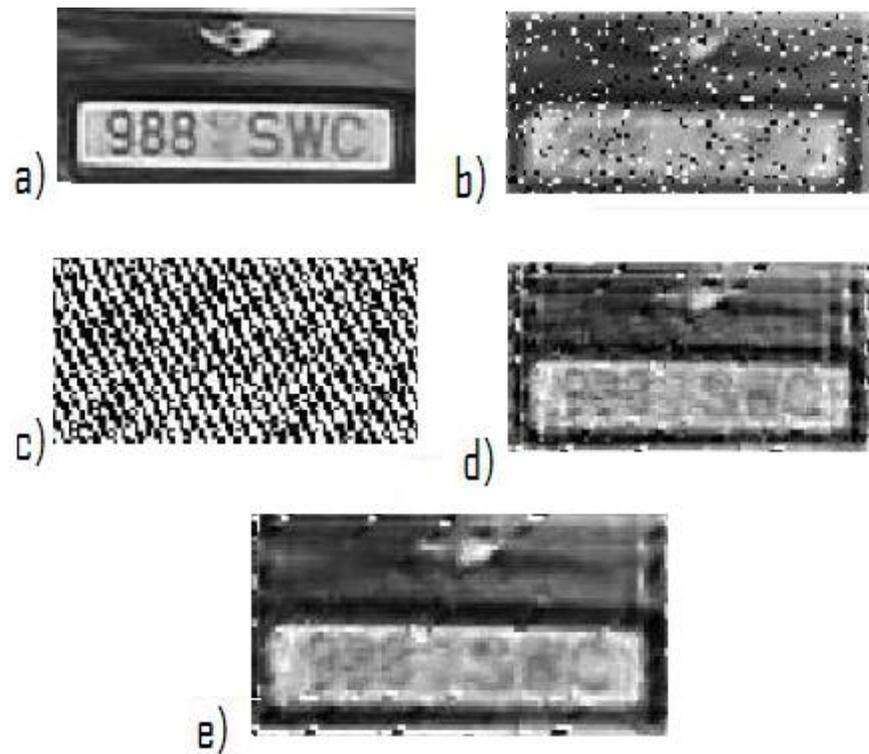


Figura 4.3: Comparación, imagen a. original, b. degradada (ángulo=45, tamaño=7,ruido: sal y pimienta=0.12), c. filtrado inverso, d. filtro Wiener y e. Lucy-Richardson 30 iteraciones.

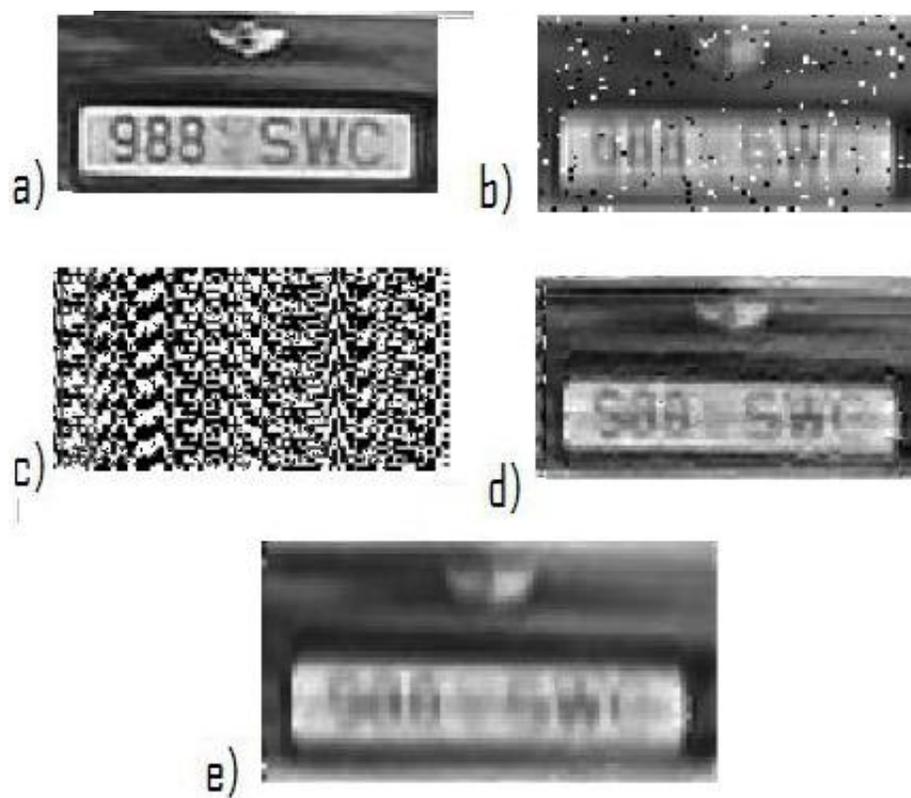


Figura 4.4: Comparación, imagen a. original, b. degradada (ángulo=45, tamaño=7, ruido: sal y pimienta=0.12), c. filtrado inverso, d. filtro Wiener y e. Lucy-Richardson 30 iteraciones.

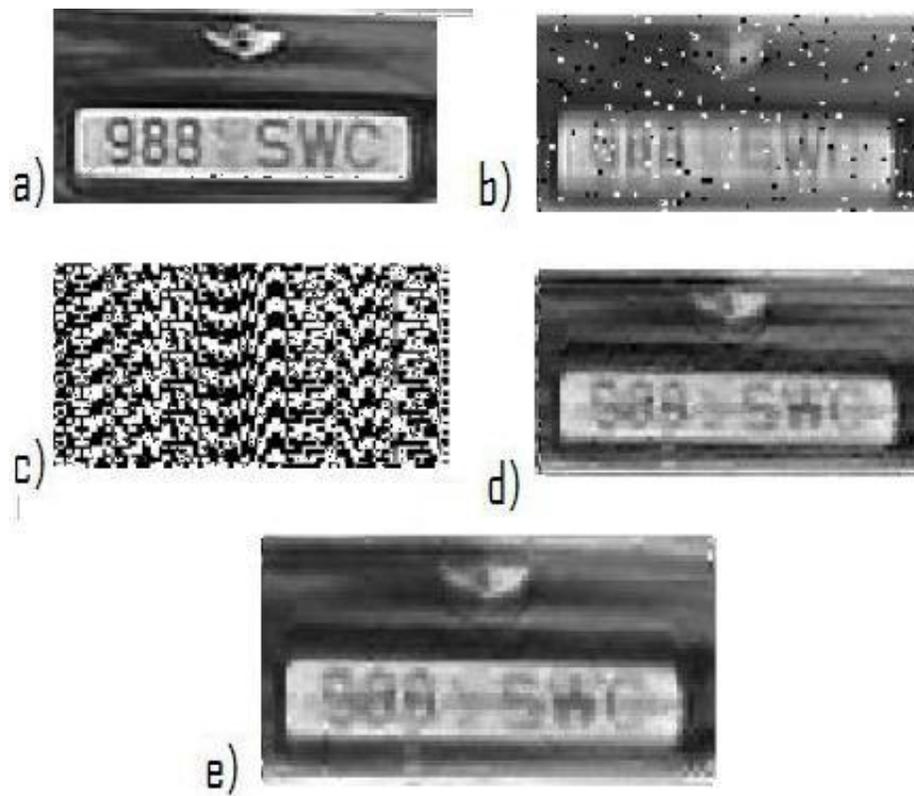


Figura 4.5: Comparación, imagen a. original, b. degradada (ángulo=90, tamaño=7,ruido: sal y pimienta=0.05), c. filtrado inverso, d. filtro Wiener y e. Lucy-Richardson 30 iteraciones.

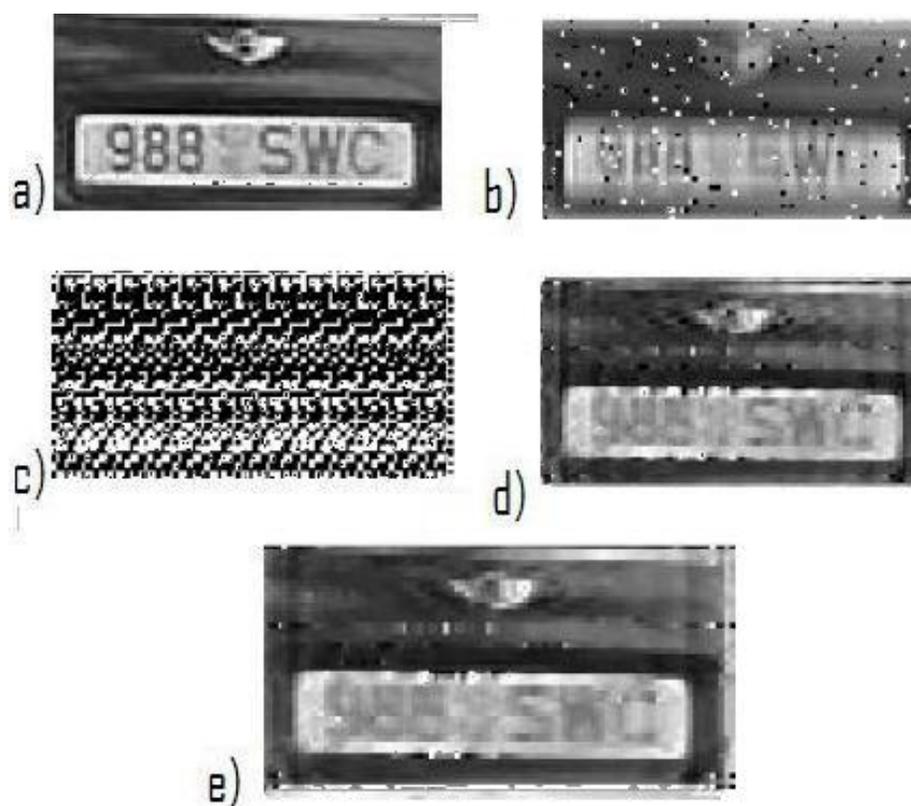


Figura 4.6: Comparación, imagen a. original, b. degradada (ángulo=180, tamaño=7, ruido: sal y pimienta=0.05), c. filtrado inverso, d. filtro Wiener y e. Lucy-Richardson 30 iteraciones.

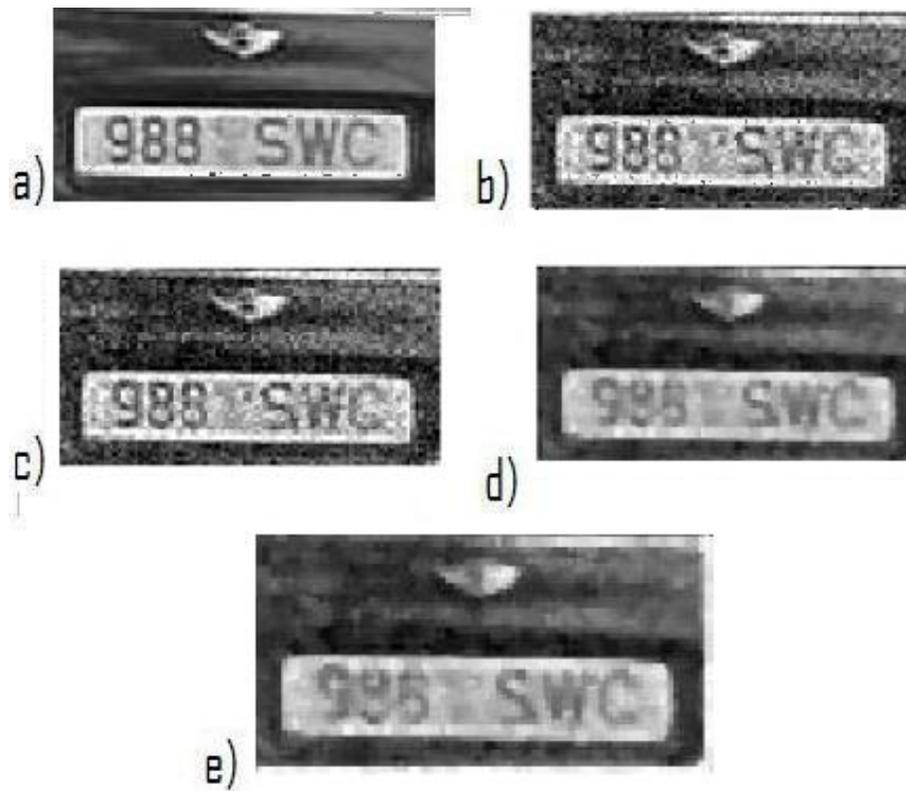


Figura 4.7: Comparación, imagen a. original, b. degradada (ángulo=0, tamaño=1, ruido: gaussiano;varianza=0.01,media=0), c. filtrado inverso, d. filtro Wiener y e. Lucy-Richardson 30 iteraciones.

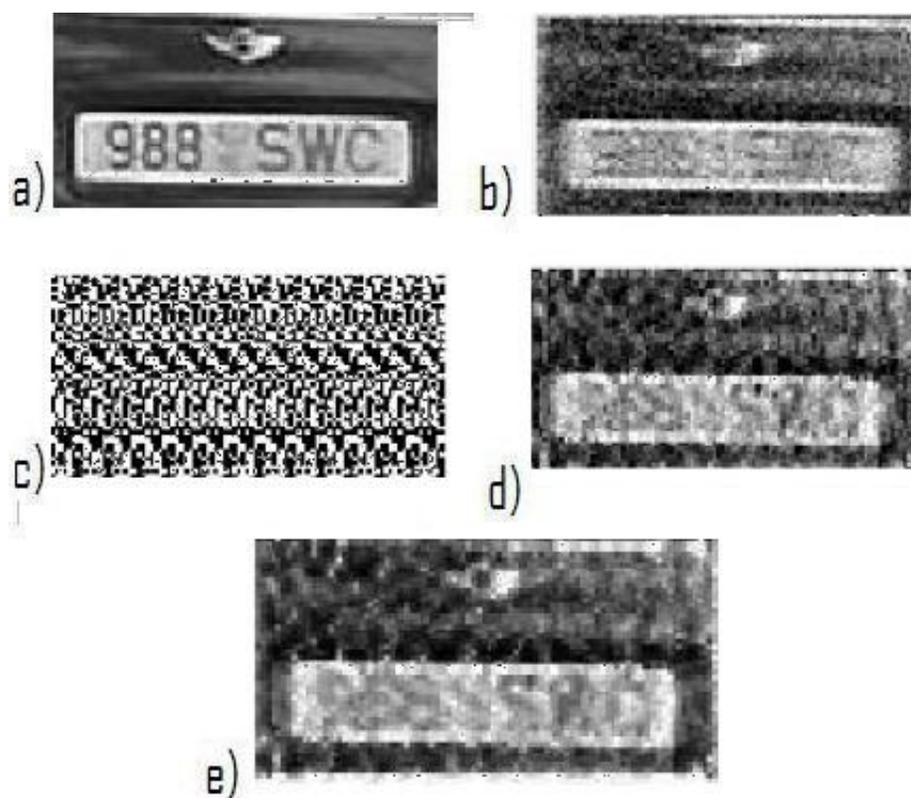


Figura 4.8: Comparación, imagen a. original, b. degradada (ángulo=0, tamaño=7, ruido: gaussiano;varianza=0.01,media=0), c. filtrado inverso, d. filtro Wiener y e. Lucy-Richardson 30 iteraciones.

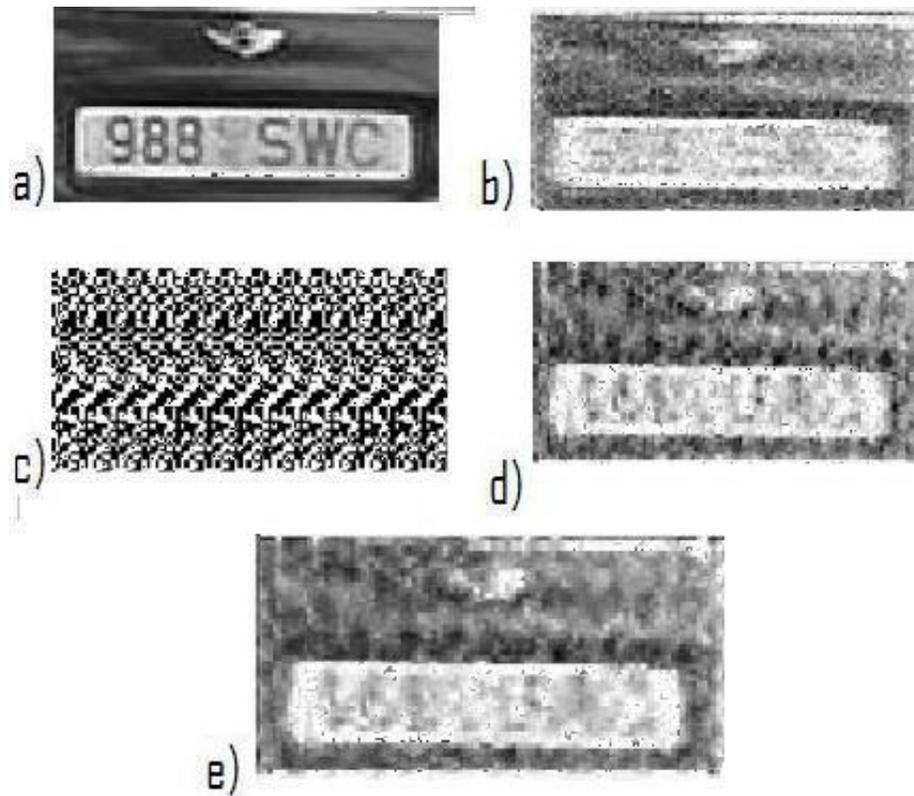


Figura 4.9: Comparación, imagen a. original, b. degradada (ángulo=0, tamaño=7, ruido: gaussiano;varianza=0.01,media=0.18), c. filtrado inverso, d. filtro Wiener y e. Lucy-Richardson 30 iteraciones.

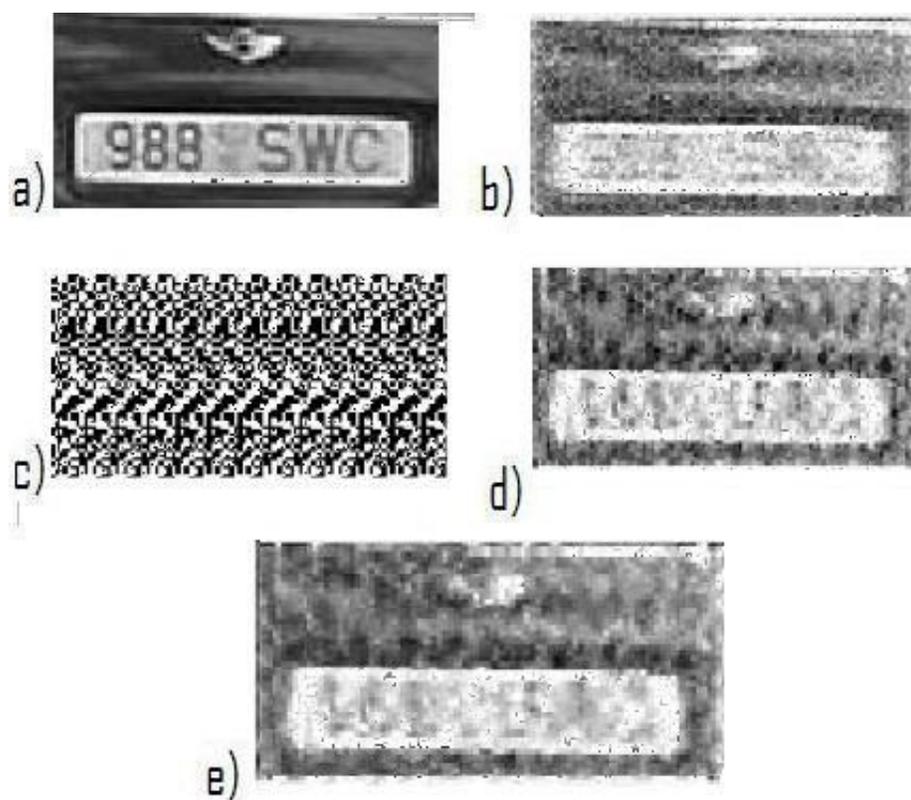


Figura 4.10: Comparación, imagen a. original, b. degradada (ángulo=0, tamaño=7, ruido: gaussiano;varianza=0.10,media=0.18), c. filtrado inverso, d. filtro Wiener y e. Lucy-Richardson 30 iteraciones.

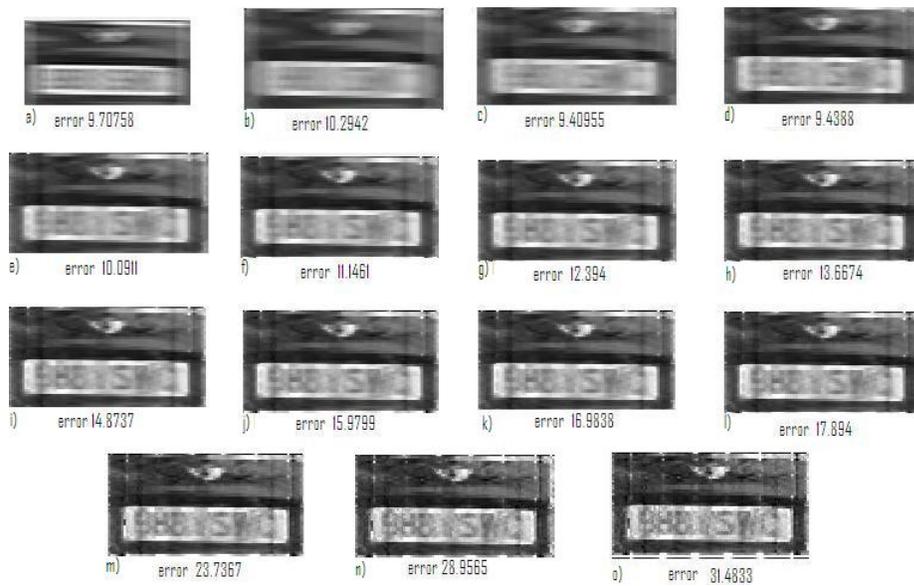


Figura 4.11: Iteraciones, imágenes; a. corrupta, b. L.R. con una iteración, c. L.R. con diez iteraciones, d. L.R. con 20 iteraciones, e. L.R. con 30 iteraciones, f. L.R. con 40 iteraciones, g. L.R. con 50 iteraciones, h. L.R. con 60 iteraciones, i. L.R. con 70 iteraciones, j. L.R. con 80 iteraciones, k. L.R. con 90 iteraciones, l. L.R. con 100 iteraciones, m. L.R. con 200 iteraciones, n. L.R. con 400 iteraciones y o. L.R. con 650 iteraciones.

Borrado		Ruido	Gaussiano		Error(Frobenius)			
Tamaño	Ángulo	Sal y pimienta	Varianza	Media	Imagen degradada	Filtrado inverso	Filtrode Wiener	Lucy-Richardson
3	0	0.1	X	X	7.572	160.606	10.633	10.115
3	45	0.1	X	X	9.539	1820.7	12.181	9.794
7	45	0.12	X	X	17.526	1.946.650	16.164	14.852
7	90	0.05	X	X	15.286	2.109xE+16	11.307	11.134
7	180	0.05	X	X	12.005	2.260xE+06	11.005	10.609
1	0	X	0.01	0	6.324	6.300	6.569	6.503
7	0	X	0.01	0	10.362	1.603xE+06	12.834	12.198
7	0	X	0.01	0.18	14.834	1.598xE+06	16.860	16.792
7	0	X	0.10	0.18	21.306	4246xE+06	29.949	25.756
1	0	0.11	X	X	12.764	12.562	6.798	NaN
1	0	0.17	X	X	16.105	15.728	7.766	NaN
4	0	0.11	X	X	15.677	5.772.930	12.168	7.964
4	0	0.17	X	X	16.513	7.439.650	16.741	9.109
5	0	0.17	X	X	16.905	285.567	17.019	9.992
6	0	0.17	X	X	17.530	7.403.730	17.715	10.32
14	0	0.17	X	X	17.995	1.350xE+17	17.972	17.021
14	45	0.17	X	X	21.889	10.554.507	153.272	154.767
14	100	0.17	X	X	21.583	7.620.123	16.113	15.763
14	100	X	X	X	16.288	2.714xE+12	12.001	11.957
3	0	X	X	X	4.283	9.122xE14	8.522	8.617
7	0	X	X	X	7.766	1.979	8.525	10.472
10	0	X	X	X	8.65	3.947xE-12	9.511	10.767
14	0	X	X	X	9.592	131.353	9.701	10.727
50	45	X	X	X	18.755	1.357xE-11	14.825	15.204
50	45	0.1	X	X	21.643	27.841.000	15.314	15.656
50	45	X	0.01	0	201.272	11757.3	17.436	17.212

Figura 4.12: Tabla de comparación de los métodos.

Otro Filtrado inverso.

```

function ex = Filtroinverso(y,h,mu);
% ex = Filtroinverso(y,h,mu);
% El filtro inverso usa parametro mu ;
% inv_g(H) = mu*abs(fft(h))/fft(h), si abs(fft(h)) <= 1/mu
% inv_g(H) = inv(H),          en otro caso
N = size(y,1);
Yf = fft2(y); %Transformada de Fourier de la convolucion
Hf = fft2(h,N,N); %Transformada de Fourier del kernel
% Caso singular
sHf = Hf.*(abs(Hf)>0)+1/mu*(abs(Hf)==0); iHf = 1./sHf;
% Hf inverso usando umbral mu
iHf =iHf.*(abs(Hf)*mu>1)+mu*abs(sHf).*iHf.*(abs(sHf)*mu<=1);
ex=real(ifft2(iHf.*Yf)); return

```

Otro Filtro de Wiener.

```

function ex = Filtrowiener(y,h,sigma,gamma,alpha);
% ex = Filtrowiener(y,h,sigma,gamma,alpha);
% Filtro de Wiener generalizado
% Si alpha = 1,(potencia del ruido ) Devuelve el filtro de wiener.
% En otro caso es el filtro inverso regularizado.
SIZE = size(y,1);
Yf = fft2(y); %Transformada de fourier de la convolucion y
Hf = fft2(h,SIZE,SIZE); % Transformada de fourier del kernel
Pyf = abs(Yf).^2/SIZE^2;
%Transformada de fourier del kernel + su inversa
sHf = Hf.*(abs(Hf)>0)+1/gamma*(abs(Hf)==0);
iHf = 1./sHf; %inversa
%Arreglo para estabilizar las altas y bajas frecuencias

```

```
iHf=iHf.*(abs(Hf)*gamma>1)+gamma*abs(sHf).*iHf.*(abs(sHf)*gamma<=1);
%iHf espectro de potencia frecuencial
%PyfEspectro de potencia del ruido, estabiliza las altas y bajas frecuencias del ruido
Pyf = Pyf.*(Pyf>sigma^2)+sigma^2*(Pyf<=sigma^2);
%Estimacion en transformada de Fourier de la imagen
Gf = iHf.*(Pyf-sigma^2)./(Pyf-(1-alpha)*sigma^2);
% Imagen restaurada
eXf = Gf.*Yf;
%La inversa de la transformada de fourier de la imagen restaurada
ex = real(ifft2(eXf)); return
```

Capítulo 5

Conclusiones.

El problema inverso de restauración de imágenes expresado en la forma de una ecuación integral de Fredholm de primer tipo

$$g = \int Hf,$$

es un problema mal planteado, resolverlo numéricamente de mediana o gran escala (512×512 - $2,048 \times 2,048$) tiene características muy importantes; por un lado, el tamaño de la matriz obliga a tener un método que optimice los recursos de la máquina y por el otro el mal comportamiento de la matriz, por tanto se requieren métodos que sean estables a pequeñas perturbaciones.

La ventaja de trabajar con la ecuación integral de Fredholm de primer tipo es que la teoría que se deriva esta bien desarrollada. Este tema fue tratado en el capítulo 1 donde se analizó la descomposición en valores singulares. Numéricamente este problema es muy costoso computacionalmente, además de que da resultados no son muy favorables, por lo que se necesitó introducir otras área de las matemáticas; probabilidad, por lo que para obtener mejores resultados, es necesario hablar de un problema multidisciplinario (matemáticas, ingeniería y programación).

La principal conclusión al respecto en el presente trabajo es haber logrado una comparación gráfica-numérica en el término del error, elaborando código diferente al implementado en el toolbox de matlab.

Otra aportación fue en la integración de la información en un solo texto, tales como la deconvolución de imágenes, recuperación de imágenes, problemas inversos, procesamiento de señales, procesamiento de imágenes, análisis numérico, álgebra lineal, probabilidad, estadística, procesos estocásticos, programación, entre otros. Logrando con esto un texto autocontenido del tema. Además como aportación, se elaboró un GUI (Graphical User Interface) con el

cual el comparar los métodos es muy práctico.

La ejecución del GUI, nos permitió ver de forma amplia lo que ocurre con los métodos expuestos. El cambiar parámetros de una forma sencilla facilitó la comparación con los métodos, así como la comparación del error en la norma de Frobenius para las imágenes; una norma matricial no natural (no *inducida*, apéndice B). La norma matricial no *inducida* más conocida es la denominada norma de Hilbert-Schmidt ó norma de Frobenius definida por

$$\|A\|_F = \left(\sum_{i=1}^n \sum_{j=1}^n |a_{ij}|^2 \right)^{\frac{1}{2}}$$

así el error fue calculado mediante

$$\mathbf{e}(\hat{\mathbf{f}}) = \left\| \mathbf{f} - \hat{\mathbf{f}} \right\|_F,$$

donde el subíndice F denota la norma de Frobenius.

En esta comparación se arrojaron resultados que dependen mucho de los parámetros propios de los métodos, del borrado y del ruido. Para el filtrado inverso no hay parámetros, sin embargo en el filtro de Wiener se usa el parámetro SNR, que es la razón señal-ruido, con el que podemos manipular para obtener mejores resultados. Para el algoritmo de Lucy-Richardson se usa el número de iteración como parámetro.

El borrado se usó en el GUI de manera que pudieramos cambiar valores, como el tamaño y ángulo de borrado. Para el ruido aditivo, se uso el ruido sal y pimienta, el cual podemos cambiar la cantidad fácilmente en matlab. Otro tipo de ruido que usamos fue el ruido de tipo Gaussiano, este ruido tiene como parámetros a la media y la varianza.

Después de realizar un número considerable de simulaciones y comparaciones con los distintos métodos, podemos concluir que los tres métodos tienen sus ventajas, que depende del tipo de degradación que presenta la imagen.

El desarrollo del algoritmo de Lucy-Richardson es diferente al clásico el cual se supone una distribución de poisson, hay versiones que tienen en cuenta ruido de poisson + ruido gaussiano; ver [42]. Nuestro desarrollo esta basado en [39] y [41]. Donde hay restricciones de positividad.

El algoritmo de Lucy-Richardson es un método de *deconvolución no lineal*, el filtro de Wiener e inverso son métodos de *deconvolución lineales*. La no linealidad del algoritmo Lucy Richardson es por el término $\sum_l h_{lj} f_l^n$ en (3.26). Sin embargo Lucy-Richardson usa un *inicializador* para iterar, dicho inicializador le llamamos el primer estimador. Para encontrar este primer estimador se usa

un filtro ó un método de deconvolución lineal, el llamado filtro de media (que usamos en la implementación) ó también puede usarse el llamado filtro de Wiener respectivamente.

Resultados positivos:

Dado que los datos observacionales g_i son positivos en (3.14) y (3.15) ([39], [41] y [42]), la forma del algoritmo garantiza que la solución es siempre positiva (o cero) en cada píxel, por lo que este método es más eficiente que los métodos de deconvolución lineales donde se pueden presentar valores negativos.

Aspectos negativos;

Esta técnica (máxima verosimilitud, algoritmo EM) sufre del problema de amplificación de ruido cuando iteramos muchas veces, esto es, luego de gran cantidad de iteraciones las diferencias obtenidas son muy pequeñas (en términos del error y visual), muy probablemente debidas al efecto de amplificación del ruido. Las iteraciones se detienen cuando se alcanza un resultado *acceptable*, dependiendo de lo que queremos, esto es visual o numérico. Aunque existe una variedad de criterios para definir exactamente cuando se considera *acceptable* un resultado; ver [43].

Trabajo a futuro; El trabajo a futuro en esta línea es el de estimar el número de iteraciones para obtener una calidad *acceptable* en la imagen, puesto que el algoritmo como tal no tiene un criterio de paro que optimice los resultados. Así como el buscar regularizar \mathbf{H} de manera más efectiva que la que se usó en la sección 4.1.3 o bien buscar una regularización para $\sum_l h_{il} f_l^n$ en (3.25). Otro trabajo sería el estimar o modelar \mathbf{H} en (3.12), de manera más precisa usando solamente la imagen observada g , donde la intención es obtener un método de deconvolución ciega.

Apéndice A

Introducción a las imágenes digitales.

A.0.1 El proceso de digitalización.

Una imagen de tipo analógica es una imágenes capturada regularmente con una cámara, un telescopio, un microscopio o cualquier otro tipo de instrumento óptico, estas imágenes presentan una variación de sombras y tonos. Para que una imagen analógica, en blanco y negro, en escala de grises o a color, pueda ser *manipulada* usando una computadora, primero debe convertirse a un formato adecuado. Este formato es la imagen digital correspondiente.

La transformación de una imagen analógica a otra discreta se llama digitalización y es el primer paso para aplicar cualquier tratamiento de procesamiento de imágenes digitales. El proceso de digitalización se muestra en la Figura A.1.

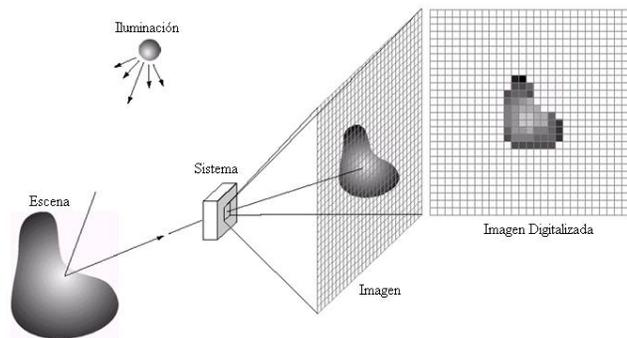


Figura A.1: Proceso digital

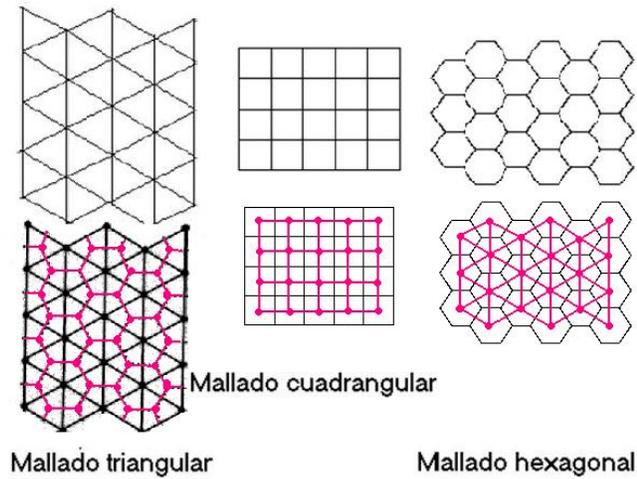


Figura A.2: Tipos de Mallado

En el modelo matemático de una imagen; un píxel se identifica con su centro, representando como puntos (x, y) del plano, donde (x, y) son las típicas coordenadas cartesianas. Dependiendo de los distintos tipos de mallado, la distribución de los píxeles es distinta ver la Figura A.2. Los bordes de las regiones están pintados en negro. Los píxeles están representados por puntos en color rojo. Hemos conectado dos píxeles si las regiones correspondientes comparten un lado común.

Es necesario determinar que tipo de discretización tenemos, esto es de que tipo de imagen estamos tratando, puede ser una imagen en color o escala de grises. Para el caso de escala de grises, solamente tenemos un valor (matricial), para imagen en color usamos un valor (matricial) con tres valores por polígono (RGB), que corresponden con la intensidad de color rojo (R), verde (G), y azul (B). La escala de colores tiene un rango discreto (por ejemplo, de 8 bits, equivale a 256 colores). Las imágenes que cuentan con sólo 2 colores; blanco y negro (0 y 1, respectivamente), se les conoce como imágenes binarias. Para nuestros fines nos centraremos en píxeles de forma cuadrada o rectangulares, esto es cada par ordenado (x, y) representa un píxel cuadrado ó rectangular. La coordenada x indica la fila donde está localizado el píxel, mientras que la coordenada y indica la columna. Por convención se usa en procesamiento de imágenes, que el píxel $(0, 0)$ se ubique en en la esquina superior izquierda de la imagen.

Una imagen digital de m por n píxeles se puede interpretar matemáticamente como una matriz. De esta forma denotaremos regularmente en este apéndice a $I \in R^{m \times n}$ como una imagen rectangular de tamaño $m \times n$.

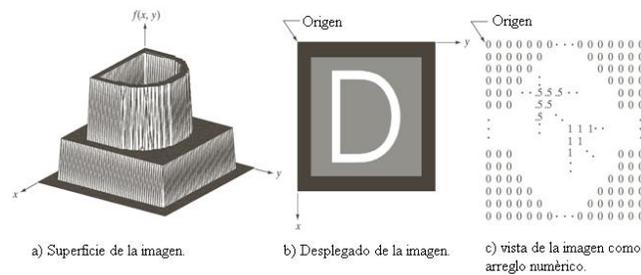


Figura A.3: Representación de la imagen.

Una imagen digital de tamaño $m \times n$ en escala de grises también se puede ver como una función, esto es $f : [0, m - 1] \times [0, n - 1] \rightarrow [0, L - 1]$, donde L representa el nivel de gris, tal que a cada punto (pixel) (x, y) , se le asigna un valor (nivel de gris) i.e. $f(x, y) = L$ con $L \in D$, donde D es el dominio de grises la imagen, en muchos casos D toma los valores de 0 a 255.

De tal manera que esta función en el espacio representa una nube de puntos, al unir estos puntos se forma un mallado, obteniendo una superficie (ver Figura A.3). Con esta superficie en el espacio, podemos obtener mucha información acerca de la imagen por medio de un estudio analítico. Para ver esto más claramente consideremos el siguiente ejemplo.

Ejemplo A.0.1.

Como ya comentamos, el mallado puede ser rectangular, pero en este caso. Usamos un mallado cuadrangular de tamaño 9×9 y una paleta de niveles de gris de 256 niveles, donde 0 indica el color negro y 255 el color blanco (ver Figura A.4). Como vemos en la Figura A.4, si partimos de una misma imagen digital y dependiendo del mallado que escojamos, la imagen obtenida es diferente. Llegados a este punto, es lógico preguntarse ¿Qué muestreo y cuántos niveles de gris son necesarios para una buena aproximación?

La resolución (el grado de detalle) de una imagen depende estrechamente de estos dos parámetros. Cuanto más se incrementan, más se aproxima la imagen digitalizada a la original. La cantidad de niveles de gris (resolución de intensidad) y la finura del mallado (resolución espacial) que escojamos, deben producir una imagen digital aceptable, en el sentido de que no sea perceptible al ojo humano el paso de un color a otro, entre dos píxeles consecutivos.

Hemos de tener en cuenta que si la imagen I tiene un tamaño $m \times n$ o sea $I \in \mathbb{R}^{m \times n}$ consiste de un mallado de m por n píxeles y el número de niveles de gris permitido es $L = 2^k$ para $k \in \mathbb{N}$, entonces el número de bits necesarios para almacenar una imagen digitalizada es $m \times n \times k$ ver [23]. Por ejemplo, una imagen de tamaño $m \times n = 128 \times 128$ con $L = 64$ niveles de gris necesita 98.304

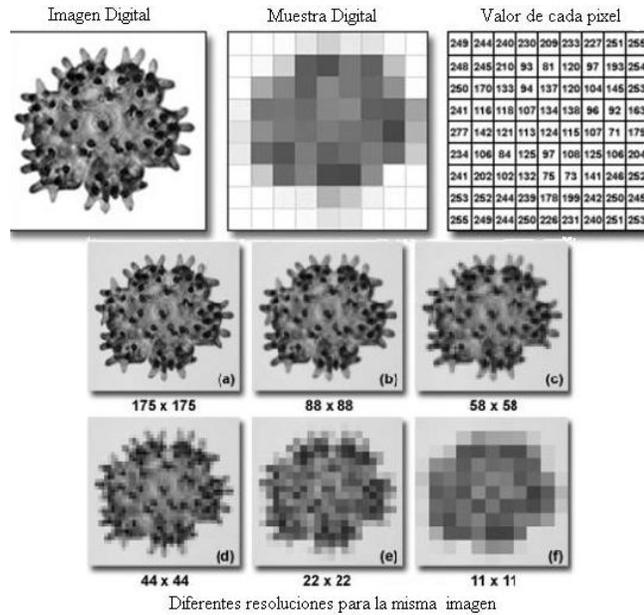


Figura A.4: Resolución.

bits = 12 KB de memoria, pues $k = \log(64)/\log(2) = 6$. Una de 256×256 con $L = 132$ niveles de gris necesita 458.752 bits = 56 KB. Y una de 1024×1024 con $L = 256$ niveles de gris necesita 8.388.608 bits = 1024 KB = 1 MB.

Las imágenes digitales a color están gobernadas por los mismos conceptos que las imágenes en escala de grises. Sin embargo, en lugar de un único valor de intensidad que expresa el nivel de gris, los píxeles de las imágenes a color usan tres componentes independientes, uno por cada color (vector tridimensional RGB = rojo, verde y azul). Combinando distintas intensidades de estos tres colores, podemos obtener todos los colores visibles.

A.0.2 Modelos en color e histogramas.

El modelo CMY.

En algunos casos, son más apropiados modelos diferentes del RGB para algoritmos y aplicaciones específicas. De cualquier manera, cualquier otro modelo sólo requiere una conversión matemática simple para obtener el modelo RGB. Para imprimir una imagen digital, es necesario convertir la imagen RGB al modelo CMY (cian-magenta-amarillo). El modelo es el siguiente

$$\begin{pmatrix} R \\ G \\ B \end{pmatrix} = \begin{pmatrix} L \\ L \\ L \end{pmatrix} - \begin{pmatrix} C \\ M \\ Y \end{pmatrix},$$

siendo $L + 1$ la cantidad de niveles de color de la imagen.

El modelo YIQ.

El modelo YIQ se usa en las televisiones comerciales. Básicamente, YIQ es una recodificación de RGB para mantener la compatibilidad con las televisiones en blanco y negro. De hecho, la componente Y (luminancia) provee toda la información requerida para una televisión en blanco y negro. La conversión de RGB a YIQ se obtiene mediante la ecuación

$$\begin{pmatrix} Y \\ I \\ Q \end{pmatrix} = \begin{pmatrix} 0.299 & 0.587 & 0.114 \\ 0.596 & -0.275 & -0.321 \\ 0.212 & -0.523 & 0.311 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix}.$$

Si sólo tenemos en cuenta la componente Y de la imagen, lo que obtenemos es una imagen en escala de grises. Así pues, la forma de obtener una imagen en escala de grises a partir de una en RGB es aplicando al valor RGB de cada pixel:

$$Y = 0.299R + 0.587G + 0.114B.$$

El modelo HSI.

Otro modelo muy utilizado es el HSI que representa el color de una manera intuitiva (es decir, de la forma en los humanos percibimos el color). La componente I se corresponde con la intensidad, H con el color y S con la saturación. Este modelo es muy utilizado en algoritmos de procesamiento de imágenes basados en propiedades del sistema de visión humano. La conversión de RGB a HSI es más complicada. Pero la componente I (intensidad) es fácil de calcular (ver [23] o [44]):

$$I = 1/3(R + G + B).$$

Histogramas.

Supongamos dada una imagen en niveles de grises, siendo el rango de 256 colores (de 0 a 255). El histograma de la imagen consiste en una gráfica donde se muestra el número de pixeles de cada nivel de gris que aparecen en la imagen. En las Figuras A.5, A.6 y A.7 podemos ver tres imágenes con sus correspondientes histogramas.

El análisis estadístico derivado del histograma puede servir para comparar contrastes e intensidades entre imágenes. El histograma podría ser alterado para producir cambios en la imagen. Por ejemplo, el histograma es utilizado para binarizar una imagen digital, es decir, convertirla en una imagen en blanco y negro, de tal manera que se preserven las propiedades *esenciales* de la imagen. La forma usual de binarizar una imagen es eligiendo un valor adecuado u (valor umbral) dentro de los niveles de grises, tal que el histograma forme un "valle" en ese nivel. Todos los niveles de grises menores que u se convierten en 0 (negro), y los mayores que u se convierten en 255 (blanco). El histograma de una imagen a color RGB consiste en tres gráficas siendo cada una el histograma de cada color primario:

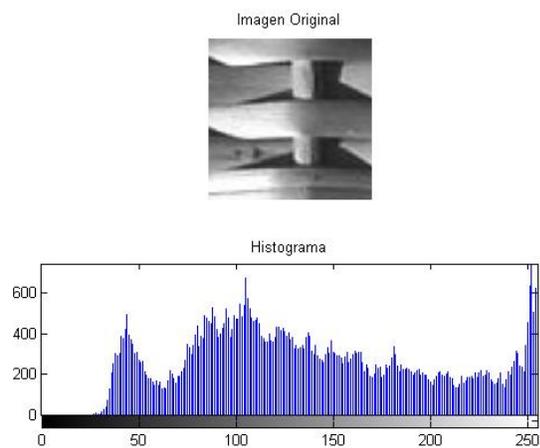


Figura A.5: Contraste Normal.

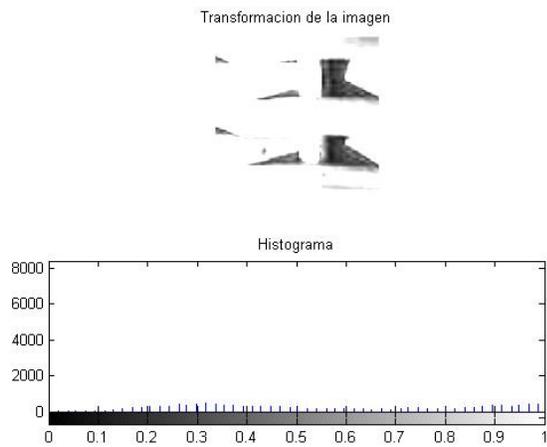


Figura A.6: Contraste Alto.

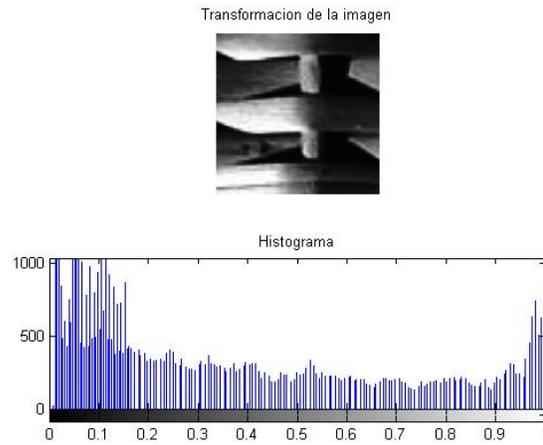


Figura A.7: Contraste Bajo.

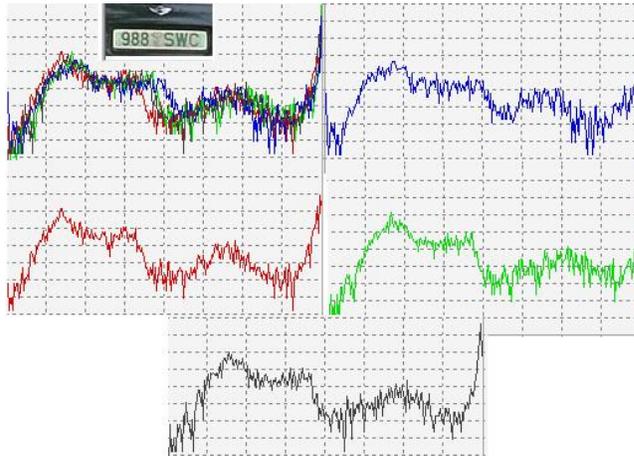


Figura A.8: Histograma en color.

A.1 Representación de Imágenes digitales.

Representar una imagen de $n \times n$ pixeles mediante una matriz requiere n^2 bits si la imagen es binaria, $n^2 \log(L)$ bits si la imagen es en niveles de gris, con L niveles y $3n^2 \log(L)$ bits si la imagen es a color (RGB) con L niveles. En esta sección estudiaremos representaciones de imágenes que pueden ser más económicas que la representación matricial en determinados casos, ya que se elimina información redundante.

Definición A.1.1. Sean n_p y n_q el número de bits necesario para almacenar dos representaciones, P y Q , distintas de la misma imagen. La **redundancia relativa** de P (con respecto de Q) se define por

$$R_D = 1 - \frac{1}{C_R},$$

donde $C_R = n_P/n_Q$ es el radio de compresión.

Ejemplo A.1.2. Si $C_R = 10$, significa que la representación P necesita 10 bits por cada bit de la representación Q . Así, $R_D = 0.9$ y por tanto, el 90 por ciento en los datos de la representación P es redundante.

Observemos que si $n_Q = n_P$, $C_R = 1$ y $R_D = 0$, por lo que P no es redundante. Ahora veamos lo siguiente

Si $n_Q \ll n_P$, entonces $C_R \rightarrow +\infty$, y $R_D \rightarrow 1$.

Si $n_Q \gg n_P$, entonces $C_R \rightarrow 0$, y $R_D \rightarrow -\infty$.

En una imagen digital hay tres tipos de redundancia:

Redundancia en el código.

Donde el código es un sistema de símbolos usado para representar la información. A veces, la longitud de las palabras usadas en el código es mayor de lo necesario.

Redundancia entre pixeles.

Debida a la correlación espacial entre un pixel y sus vecinos.

Redundancia psicovisual. Parte de la información es ignorada por el ojo humano.

La compresión de imágenes consiste en eliminar una o más de estas redundancias. Se pueden clasificar las distintas formas de compresión de imágenes en: aquellas que no producen errores (o pérdida de información) en la imagen; aquellas que sí lo producen. Los procesos de compresión que eliminan la redundancia de código y/o entre pixeles no producen errores. Aquellos que eliminan la redundancia psicovisual, sí.

A.1.1 Formatos de imágenes digitales.

Un formato de fichero de imagen es una forma estándar de organizar y almacenar los datos que representan la imagen. A un formato se le llama contenedor cuando puede manejar distintos tipos de datos. Un estándar de compresión define una serie de procedimientos para comprimir y descomprimir imágenes.

Una compresión sin pérdidas devuelve la imagen descomprimida exactamente igual a la original. Por el contrario, la compresión con pérdidas acepta alguna degradación en la imagen en beneficio de una mayor compresión. Las imágenes más simples contienen sólo dos colores: blanco y negro, y sólo se necesita de un bit para representar cada pixel. La mayoría de las tarjetas de video en las PC más antiguas soportaban sólo 16 colores pre-fijados. En la actualidad admiten 2^{24} ó bien, 16 millones de colores.

A.2 Filtros digitales.

Los filtros digitales 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 espacial, trabajando directamente sobre los píxeles de la imagen, o en el dominio de la frecuencia, donde las operaciones se llevan a cabo sobre la Transformada de Fourier de la imagen. Como ya se mencionó, los filtros se usan para suavizar las imágenes (reducir la cantidad de variaciones de intensidad entre píxeles vecinos), reducir el ruido (eliminar aquellos píxeles cuyo nivel de intensidad es muy diferente al de sus vecinos), realzar bordes, detectar bordes (detectar píxeles donde se produce un cambio brusco en la función intensidad). Podemos decir intuitivamente que el ruido en una imagen es aquel que hace que la imagen se vea menos nitida, menos clara, donde los detalles están "sucios" o "borrosos". Para ver esto matemáticamente tenemos la siguiente definición

Definición A.2.1. *El ruido es la información no deseada que contamina la imagen, la cual denotamos por*

$$n(x, y), \quad (\text{A.1})$$

si además denotamos a la imagen original por

$$f(x, y), \quad (\text{A.2})$$

tenemos que la imagen observada $g(x, y)$ es una imagen que está compuesta por la imagen original más el ruido, lo cual queda expresado de la siguiente manera

$$g(x, y) = f(x, y) + n(x, y), \quad (\text{A.3})$$

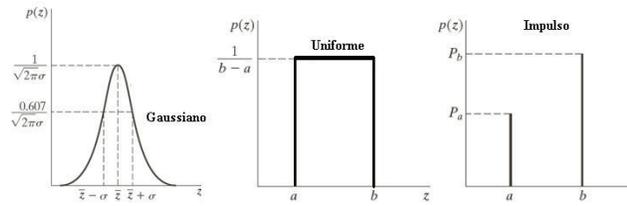


Figura A.9: Ruido1.

El origen de este ruido puede estar tanto en el proceso de adquisición de la imagen (errores en los sensores), como en el de transmisión (debido a interferencias en el canal de transmisión). Existen distintos modelos de ruido, según las funciones de densidad de probabilidad que sigan sus intensidades $n(x, y)$; ruido gaussiano, ruido uniforme, ruido impulsivo (sal y pimienta), etc.

Ruido Gaussiano (o normal).

Modela el ruido producido por los circuitos electrónicos o ruido de los sensores por falta de iluminación y/o altas temperaturas.

Ruido uniforme.

Toma valores en un determinado intervalo de forma equiprobable. Se da en un menor número de situaciones reales.

Ruido impulsivo ó sal y pimienta.

Se produce normalmente en la cuantificación que se realiza en el proceso de digitalización.

A.2.1 Filtros digitales (en el dominio espacial).

Los filtros se pueden clasificar en dos tipos, filtros en el dominio espacial y filtros en el dominio de la frecuencia.

Filtros en el dominio espacial:

Las operaciones espaciales (o de vecindad) se definen en un entorno E_N (vecindad) del punto a transformar (m_0, n_0) , ver Figura A.10. Los filtros en el dominio del espacio pueden clasificarse en; filtros lineales (filtros basados en

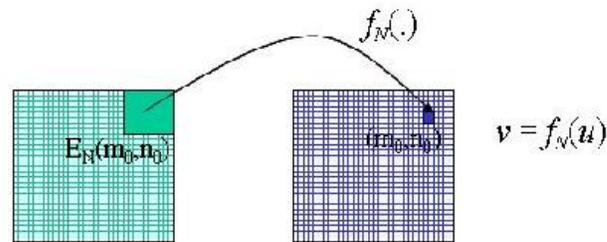


Figura A.10: Filtro espacial.

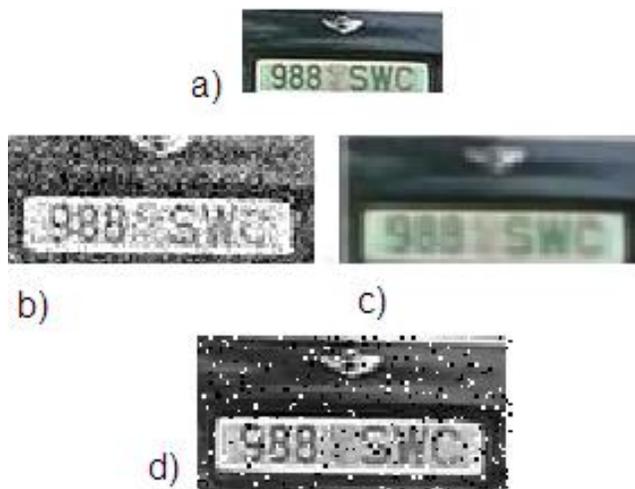


Figura A.11: a. Imagen original, b. Ruido de Gauss, c. Ruido uniforme, d. Ruido sal y pimienta.

máscaras de convolución), filtros no lineales (por ejemplo, el filtro de Wiener).

Dada una imagen $f(x, y)$ y una máscara $w(x, y)$, la imagen resultante $g(x, y)$ se obtiene al realizar la operación

$$g(x, y) = \sum_{s=-a}^a \sum_{t=-b}^b w(s, t) f(x + s, y + t), \quad (\text{A.4})$$

donde la máscara es una matriz de coeficientes, donde se considera el entorno del punto (x, y) para obtener la imagen $g(x, y)$, que esta determinada por el tamaño y forma de la máscara $w(x, y)$, así que el filtrado está determinado por el contenido de la máscara, ver Figura A.13. Puede aplicarse la máscara extendiendo la imagen con un marco de ceros de anchura adecuada. Esto puede tener efectos no deseados, pero, en general, poco significativos si la máscara es pequeña en relación con el tamaño de la imagen.

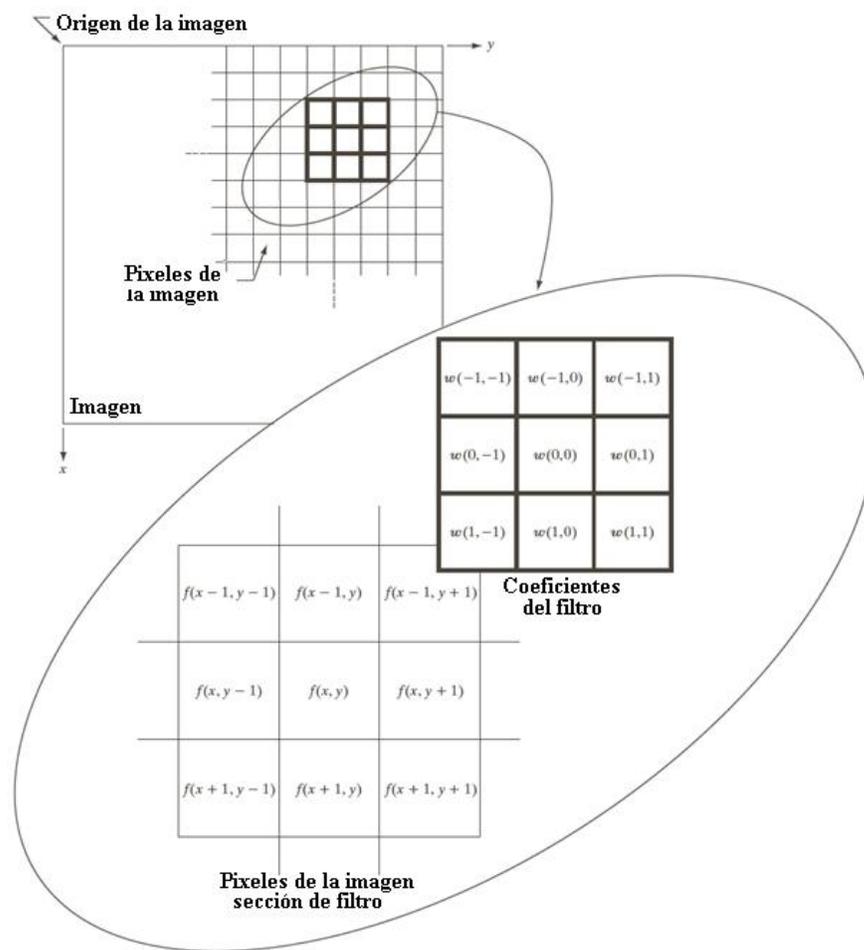


Figura A.12: Filtro lineal.

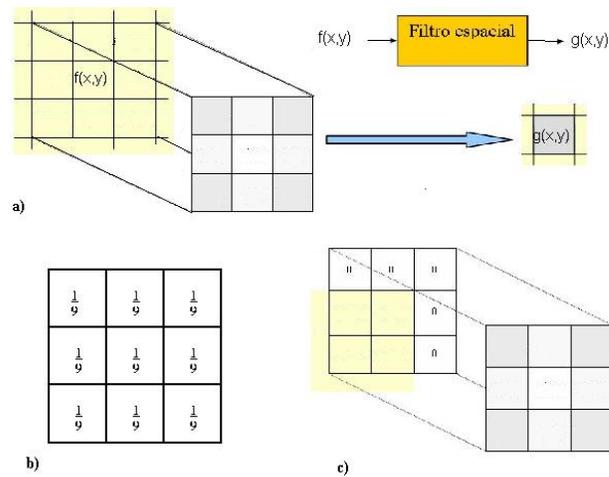


Figura A.13: a. Filtro lineal, b. Ejemplo de mascara y c. Un marco de ceros.

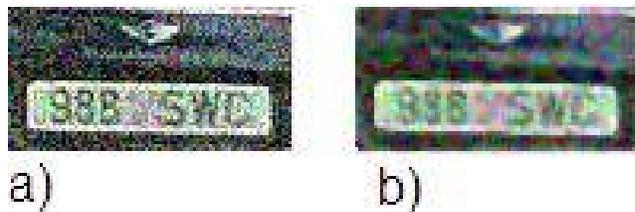


Figura A.14: a. Imagen con ruido, b. Filtro de media.

Filtro de media.

El filtro de la media es el más simple, intuitivo y fácil de implementar con el fin de suavizar imágenes, comparado con otros filtros espaciales, es decir, se busca reducir la cantidad de variaciones de intensidad entre píxeles vecinos. ¿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, ver un ejemplo en la Figura A.13.b. El filtro de la media ofrece ciertas desventajas; es más sensible a cambios locales que el de la mediana (un filtro no lineal), puede crear nuevas intensidades de grises (ó colores) que no aparecían en la imagen, ver Figura A.14.

Filtro gaussiano.

El filtro gaussiano se usa para recuperar imágenes y eliminar ruido. Es simi-

lar al filtro de media pero se usa una máscara diferente, modelando la función gaussiana:

$$G(x, y) = \frac{1}{Z} e^{-\frac{x^2+y^2}{2\sigma^2}}, \quad (\text{A.5})$$

donde σ^2 representa la varianza (nivel de suavizado).

Ejemplo A.2.2.

Máscara de 5×5 para el filtro gaussiano con $\sigma = 1$.

$$\frac{1}{273} \begin{bmatrix} 1 & 4 & 7 & 4 & 1 \\ 4 & 16 & 26 & 16 & 4 \\ 7 & 26 & 41 & 26 & 7 \\ 4 & 16 & 26 & 16 & 4 \\ 1 & 4 & 7 & 4 & 1 \end{bmatrix}$$

Las desventajas del filtrado gaussiano son principalmente tres (ver Figura A.15.)

1. Disminución de nitidez.
2. Aumento de borrosidad.
3. Pérdida de detalles.

Las ventajas del filtro gaussiano frente al filtro de media son; es separable, es decir, en lugar de realizar una convolución bidimensional, podemos realizar dos convoluciones unidimensionales. Una en sentido horizontal y otra en sentido vertical, puesto que

$$e^{-\frac{x^2+y^2}{2\sigma^2}} = e^{-\frac{x^2}{2\sigma^2}} e^{-\frac{y^2}{2\sigma^2}}$$

Otra observación es que el filtro gaussiano produce un suavizado más uniforme que el filtro de media.

Filtros no lineales.

En los filtros no lineales se realiza un tipo de operación distinta a la convolución, una operación no lineal. Esto es, si $h(i, j)$ es la máscara y $f(i, j)$ la imagen original, entonces la imagen final resulta de realizar una operación del tipo:

$$g(i, j) = O_{m,n}[h(m, n)f(i - m, j - n)], \quad (\text{A.6})$$

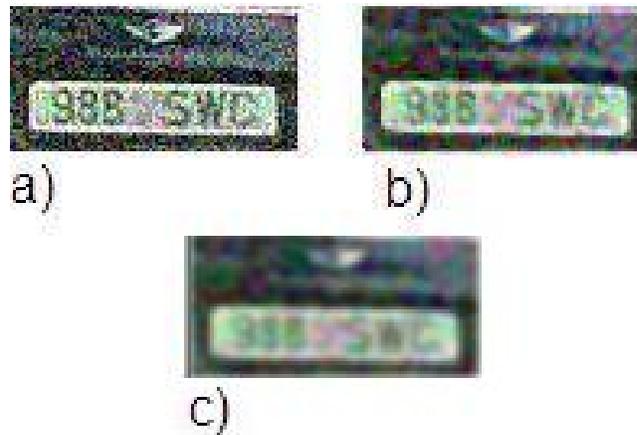


Figura A.15: a. Imagen con ruido, b. Filtro de media, c. Filtro de gauss 5 por 5

donde $O_{m,n}$ denota un operador no lineal, ver [45].

Filtro de la mediana.

Este filtro se suele usar para eliminar ruido en la imagen. ¿Cómo funciona? Se toma cada píxel de la imagen y se reemplaza por la mediana de los píxeles vecinos. La mediana se calcula ordenando los valores de los píxeles vecinos en orden y seleccionando el que queda en medio. Esto es, si tenemos la siguiente imagen en forma matricial

$$\begin{pmatrix} 123 & 125 & 126 & 130 & 140 \\ 122 & 124 & 126 & 127 & 135 \\ 118 & 120 & 150 & 125 & 134 \\ 119 & 115 & 119 & 123 & 133 \\ 111 & 116 & 110 & 120 & 130 \end{pmatrix}$$

Los valores de los píxeles vecinos del píxel cuyo valor es 150 (vecindad en imagenología) son 115, 119, 120, 123, 124, 125, 126, 127 y 150 cuya mediana es 124. Una de las desventajas de este filtro, es que es no lineal, dadas dos imágenes A y B ,

$$\text{Mediana}[A + B] \neq \text{Mediana}[A] + \text{Mediana}[B].$$

Una de las ventajas es que da muy buenos resultados en caso de ruido *sal y pimienta* (salt and pepper) Figura A.16.

Otros filtros.

Otros filtros son parecidos, como el filtro de mediana en entorno de vecindad, punto medio en entorno de vecindad (usa $g = f_{\max} + f_{\min}/2$), alpha-media del

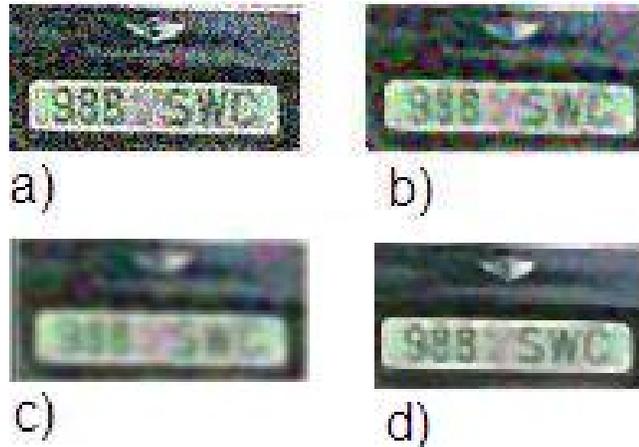


Figura A.16: a. Imagen con ruido, b. Filtro de media, c. Filtro de gauss 5 por 5, d. Imagen resultante tras realizar un filtro de mediana de tamaño 7×7 .

entorno de vecindad, media geométrica del entorno de vecindad (usa $M_{geo} = \prod [f]^{1/N^2}$), filtros adaptativos como el Filtro Mínimo Error Cuadrático MMSE (usa $MMSE = f_{i,j} - (\sigma_n^2 / \sigma_1^2) [f_{i,j} - m_{i,j}]$, donde σ_n es la varianza del ruido, σ_1 es la varianza local y m es la media local), etc.

A.2.2 Filtros espaciales de realce.

El objetivo de estos filtros es realzar los detalles de una imagen que hayan podido quedar degradados. Estos filtros están asociados, por tanto, con la detección de lados ó bordes.

La idea que subyace en la mayor parte de las técnicas de detección de lados es el cálculo de un operador local de derivación ya que un píxel pertenece a un borde si se produce un cambio brusco entre niveles de grises con sus vecinos. Mientras más brusco sea el cambio, más fácil es detectar el borde.

El primer problema que surge usando esta definición es debido a la digitalización. El segundo problema es debido al ruido. La derivada de una función digital se define en términos de variaciones entre píxeles adyacentes. Existen varias formas de definir estas diferencias, pero deben cumplir que la primera derivada debe ser cero en zonas de intensidad constante y distinta de cero en zonas de variaciones (escalones o rampas) y la segunda derivada debe ser cero en zonas de intensidad constante y a lo largo de rampas con pendiente constante y debe ser distinta de cero en escalones y en comienzo y fin de rampas. Así, se pueden definir derivadas de primer y segundo orden de una función unidimensional $f(x)$,

por

$$\frac{\partial f}{\partial x} = f(x+1) - f(x)$$

y

$$\frac{\partial^2}{\partial x^2} = f(x+1) + f(x-1) - 2f(x),$$

respectivamente

En general, la segunda derivada será más sensible que la primera ante cambios bruscos en la imagen, por lo que también detectará más sutilmente el ruido. Una aproximación del gradiente de una imagen sería

$$\nabla_1 f(x, y) = (\nabla_{x1} f(x, y), \nabla_{y1} f(x, y)),$$

donde

$$\begin{aligned} \nabla_{x1} f(x, y) &= f(x, y) - f(x-1, y) \\ \nabla_{y1} f(x, y) &= f(x, y) - f(x, y-1) \end{aligned}$$

Estas operaciones pueden expresarse en forma de convolución usando las siguientes máscaras, respectivamente:

$$\begin{bmatrix} 0 & -1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad \text{y} \quad \begin{bmatrix} 0 & 0 & 0 \\ -1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}.$$

A continuación, calculamos el módulo del gradiente obtenido en cada píxel de la imagen. Los valores grandes corresponden a píxeles del borde o a ruido. Un problema de esta aproximación es que no calcula el gradiente en el punto (x, y) sino en el punto $(x-1/2, y-1/2)$. Una mejor aproximación podría ser

$$\nabla_2 f(x, y) = (\nabla_{x2} f(x, y), \nabla_{y2} f(x, y)),$$

donde

$$\begin{aligned} \nabla_x f(x, y) &= f(x+1, y) - f(x-1, y) \\ \nabla_y f(x, y) &= f(x, y+1) - f(x, y-1) \end{aligned}$$

Este operador es simétrico respecto al píxel (x, y) . Las máscaras que se aplicarían en este caso serían, respectivamente:

$$\begin{bmatrix} 0 & -1 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \quad \text{y} \quad \begin{bmatrix} 0 & 0 & 0 \\ -1 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}.$$

El problema es que no se tiene en cuenta el valor de la imagen en dicho píxel, ni de los píxeles que se encuentran en la diagonal. Una vez calculado el gradiente $\nabla f = (f_x, f_y)$, calculamos el módulo del mismo $|\nabla f| = \sqrt{f_x^2 + f_y^2}$, en cada píxel de la imagen, sin embargo, se suele calcular la suma del valor absoluto de

sus coordenadas $|\nabla f| \approx |f_x| + |f_y|$ en lugar del módulo (por ser menos costoso). Computacionalmente los valores grandes corresponden a píxeles del borde o a ruido.

Operador cruzado de Roberts.

En este caso, se suelen usar dos máscaras para modelizar el gradiente

$$F_x = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}, \quad F_y = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}$$

La ventaja de este operador es que es fácil y rápido de calcular. Sólo está implicado un entorno de vecindad de 4 píxeles y sólo se usan sumas y restas en los cálculos. La principal desventaja es que, si lo que se quiere es determinar bordes, es muy sensible al ruido y tiene una respuesta débil a los verdaderos bordes, a menos que sean muy pronunciados. Para este propósito funciona mejor el operador de Sobel ver Figura A.17.

Operador de Sobel.

En este caso, se suelen usar dos máscaras para modelizar el gradiente, llamadas operadores de Sobel:

$$S_x = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}, \quad S_y = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -3 & 0 & 1 \end{bmatrix}$$

Observemos en la figura A.17 que en las máscaras de Sobel, tienen más peso los píxeles situados en posición vertical y horizontal respecto el píxel estudiado que los situados en la diagonal. Como observación decimos que este operador es menos sensible al ruido, esto se puede ver en la Figura A.17, donde se compara con el operador cruzado de Roberts, en este ejemplo puede apreciarse la diferencia al aplicar a la imagen el operador cruzado de Roberts y el operador Sobel.

El laplaciano.

El laplaciano de una función bidimensional f es un operador de derivación isotrópico (independiente de la dirección de la discontinuidad en la imagen) definido por:

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

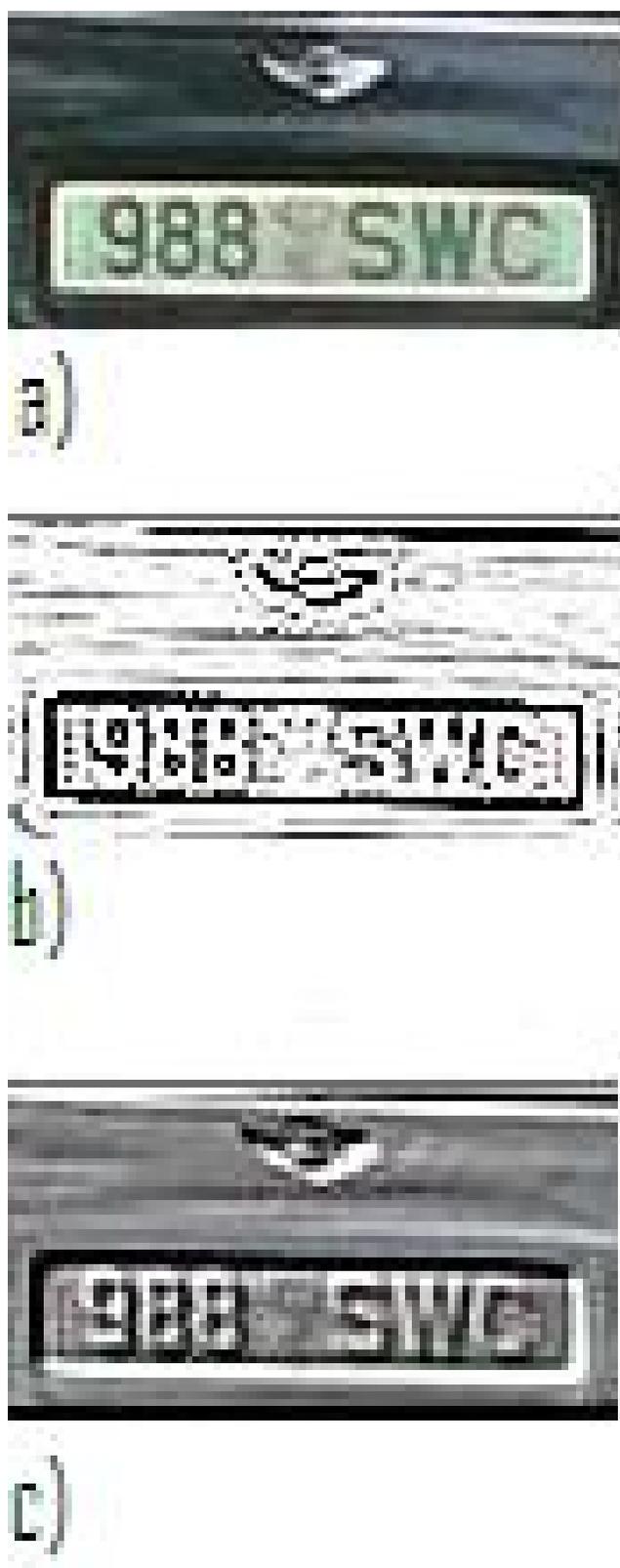


Figura A.17: Filtros de realce. a.-Original, b.- Roberts y c.- Sobel.

Como en el caso del gradiente, la ecuación del laplaciano puede implementarse en forma digital de varias maneras. La más frecuente en la práctica es aplicar la máscara:

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix};$$

es decir,

$$\nabla^2 f(x, y) = [f(x + 1, y) + f(x - 1, y) + f(x, y + 1) + f(x, y - 1)] - 4f(x, y).$$

Existen varios modelos para implementar el Laplaciano digital, una de las características de la implementación es que la suma de los coeficientes de la máscara debe ser cero, lo que es coherente en el caso de que el punto en cuestión y sus vecinos tengan el mismo valor. Los píxeles del borde darán como respuesta un número negativo grande. El Laplaciano no se suele usar directamente en la práctica por ser muy sensible al ruido, por lo que se suele usar sumado o restado (según la máscara usada) con la imagen original para realzar los contornos.

Por la misma razón, también a veces se usa primero un filtro gaussiano para eliminar ruido, lo que da lugar al filtro llamado Laplaciano del Gaussiano ($L \circ G$), cuyo núcleo puede calcularse componiendo ambos:

$$(L \circ G)(x, y) = -\frac{1}{\pi\sigma^4} \left[1 - \frac{x^2 + y^2}{s\sigma^2} \right] e^{-\frac{x^2 + y^2}{2\sigma^2}}.$$

A.2.3 Filtros en dominio de frecuencia.

Una imagen, como hemos dicho, se puede filtrar ya sea en el dominio de frecuencia o en el dominio espacial. Esto es, dada la convolución

$$g(x, y) = (h * f)(x, y),$$

donde h representa la función de borrado de la imagen ideal f y g es la imagen borrada obtenida. Aplicando la ecuación (1.9) del Teorema de convolución 1.0.4, pasamos al dominio de frecuencia

$$G(u, v) = H(u, v)F(u, v).$$

Los filtros en el dominio de la frecuencia se usan, principalmente, para eliminar altas o bajas frecuencias de la imagen, lo que se traduce en suavizar la imagen, o bien, realzar o detectar bordes. Básicamente, hay tres tipos diferentes de filtros de frecuencia; filtros de paso bajo (lowpass filter), filtros de paso alto (highpass filter) y filtros de banda (bandpass filter).

1. **Filtro de paso bajo:** Deja inalterables las bajas frecuencias y atenúa o elimina las altas frecuencias. El resultado en el dominio espacial consiste

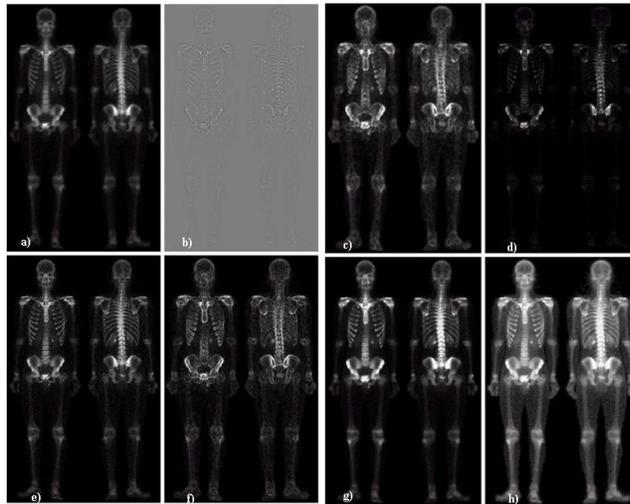


Figura A.18: a. Exploración del cuerpo entero, b. El Laplaciano, c. Imagen obtenida afilada, d. Sobel e. Sobel suavizado con filtro medio de cinco por cinco, f. imagen formada por el producto de c y e, g. Imagen obtenida por la suma de a y f, h. Resultado de aplicar ley de potencia a g.

en un suavizado; eliminar pequeños detalles y ruidos de la imagen. El más sencillo es el Filtro ideal de paso bajo, en el que

$$H(u, v) = \begin{cases} 1 & \text{si } D(u, v) \leq D_0 \\ 0 & \text{si } D(u, v) > D_0 \end{cases}$$

donde $D(u, v)$ es la distancia euclidiana de (u, v) al origen del plano de frecuencias. Este filtro suprime las altas frecuencias mayores que un cierto valor D_0 , que se denomina frecuencia de corte, y deja las demás tal como están.

Otro ejemplo es el Filtro Butterworth de paso bajo de orden n , está dado por

$$H(u, v) = \frac{1}{1 + \left(\frac{D(u, v)}{D_0}\right)^{2n}}$$

Conforme aumenta n , la función H se parece cada vez más a la correspondiente al filtro ideal, ver imagen A.21, donde $D(u, v)$ es la distancia euclidiana de (u, v) al origen del plano de frecuencias y D_0 es la llamada frecuencia de corte.

Otro filtro de paso bajo es el Filtro paso bajo Gaussiano

$$H(u, v) = e^{-\frac{D^2(u, v)}{2D_0^2}}$$

Imagen original (superior izquierda) , junto con los resultados de pasar un filtro ideal de paso bajo con frecuencias de corte de 5, 15, 30, 80 y 230. Nótese el efecto de ondas o anillos alrededor de los bordes.

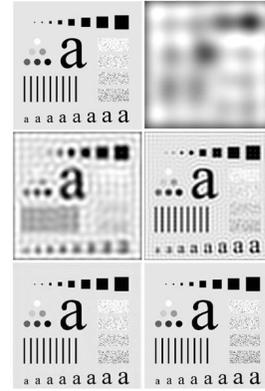


Figura A.19: Filtro paso bajo.

Con este filtro se obtienen resultados comparables a los del filtro Butterworth (sin anillos). Mediante la transformada inversa de Fourier de $H(u, v)$ se obtiene un filtro espacial Gaussiano.

2. **Filtro de paso alto:** Deja inalterables las altas frecuencias y atenúa o elimina las bajas frecuencias. El resultado en el dominio del espacio consiste en un realzado de los cambios bruscos de niveles de grises. De ahí que se use para detectar bordes. Las áreas de niveles de gris constantes o con poca variación se corresponden con las bajas frecuencias, que se suprimen.

El más sencillo es el Filtro ideal de paso alto, en el que

$$H(u, v) = \begin{cases} 0 & \text{si } D(u, v) \leq D_0 \\ 1 & \text{si } D(u, v) > D_0 \end{cases}$$

donde $D(u, v)$ es la distancia euclídeana de (u, v) al origen del plano de frecuencias. Este filtro suprime las frecuencias menores o iguales que un cierto valor D_0 , que se denomina frecuencia de corte. Al igual que para el caso de Filtro de pasabajo, también para el Filtro de paso bajo hay Filtro de paso alto de Butterworth de orden n y Filtro de paso alto Gaussiano que están dados por

$$H(u, v) = \frac{1}{1 + \left(\frac{D_0}{D(u, v)}\right)^{2n}}.$$

y

$$H(u, v) = 1 - e^{-\frac{D^2(u, v)}{2D_0^2}},$$

respectivamente

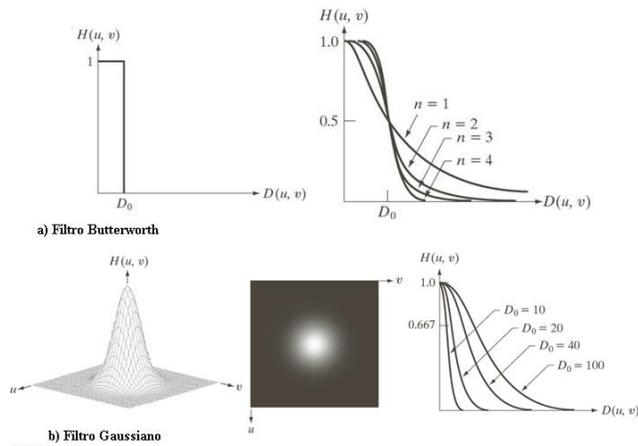


Figura A.20: Filtro Butterworth y Gaussiano.

3. Filtros de paso de banda.

Un filtro de banda atenúa las altas y bajas frecuencias, pero mantiene intactas las frecuencias que se encuentren en una banda determinada. En el caso del filtro ideal, sólo deja pasar las frecuencias que están entre dos frecuencias de corte. Se puede obtener un filtro de banda multiplicando uno de paso bajo por uno de paso alto, en el que la frecuencia de corte del de paso bajo sea superior a la del de paso alto. El opuesto al filtro de paso de banda será de rechazo de banda (bandreject o band-stop), en el que se atenúan las frecuencias de la banda, pero se mantienen las frecuencias fuera de ella. Para un estudio más profundo ver [23].

Imagen original (superior izquierda) ,
 junto con los resultados de pasar un fil-
 tro Butterworth de paso bajo con fre-
 cuencias de corte de 5, 15, 30, 80 y 230.
 Nótese que ahora no aparecen los efec-
 tos de ondas o anillos alrededor de los
 bordes.

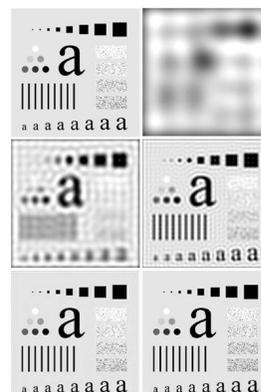


Figura A.21: Filtro paso bajo.

Apéndice B

Mal condicionamiento

Ejemplo B.0.3.

El problema del cálculo de los valores propios de una matriz no simétrica es frecuentemente mal condicionado. Un ejemplo sencillo para darse cuenta de esto es considerar la matriz

$$A = \begin{pmatrix} 1 & 1000 \\ 0 & 1 \end{pmatrix}$$

cuyos valores propios son $\lambda_1 = 1$ y $\lambda_2 = 1$. Si perturbamos uno de los coeficientes fuera de la diagonal por 0.001 (o $\varepsilon > 0$), por ejemplo el coeficiente 0, obtenemos la matriz

$$A' = \begin{pmatrix} 1 & 1000 \\ 0.001 & 1 \end{pmatrix}$$

cuyos valores propios son ahora $\lambda'_1 = 0$ y $\lambda'_2 = 2$, obteniendo un cambio drástico en los valores propios. De hecho, esto mismo sucede con la matriz

$$\begin{pmatrix} 1 & 10^n \\ 0 & 1 \end{pmatrix} \quad n \in \mathbb{N}$$

cuando perturbamos por

$$\begin{pmatrix} 1 & 10^n \\ 10^{-n} & 1 \end{pmatrix} \quad n \in \mathbb{N}$$

Algunos ejemplos de matrices mal condicionadas pueden ser las matrices de Hilbert, de Vandermonde, etc.. por ejemplo ver Figura B.1 en matlab.

Para definir el mal condicionamiento de una matriz, primero necesitamos enunciar algunos conceptos.

```

>> A=[1 10. ^3; 10. ^-3 1]

A =

1.0e+003 *

    0.0010    1.0000
    0.0000    0.0010

>> inv(A)
Warning: Matrix is singular to working precision.

ans =

    Inf    Inf
    Inf    Inf

```

Figura B.1: Intento de inversion en matlab. Matriz A, mal condicionada.

B.1 Norma inducida.

Sea V un espacio vectorial real. Una **norma** en V es una función no negativa $\|\cdot\| : V \rightarrow \mathbb{R}$ que satisface:

1. $\|x\| \geq 0$, $\forall x \in \mathbb{R}$, y $\|x\| = 0$ si y solo si $x = 0$
2. $\|\alpha x\| = |\alpha| \|x\|$, $\forall \alpha \in \mathbb{R}$, $\forall x \in V$
3. $\|x + y\| \leq \|x\| + \|y\|$, $\forall x, y \in V$ (desigualdad del triángulo)

Existen varias normas, denominadas **normas-p**.

Sea $x = (x_1, \dots, x_n)^T \in \mathbb{R}^n$, con $V = \mathbb{R}^n$, se definen

1. norma-1: $\|x\|_1 = \sum_{i=1}^n |x_i|$
2. norma-2: $\|x\|_2 = (\sum_{i=1}^n |x_i|^2)^{1/2}$
3. norma-p: $\|x\|_p = (\sum_{i=1}^n |x_i|^p)^{1/p}$ ($p \geq 1$)
4. norma- ∞ : $\|x\|_\infty = \max_{1 \leq i \leq n} |x_i|$

Cuando se hace análisis numérico de problemas donde intervienen matrices, es útil tener el concepto de norma de una matriz y, en consecuencia, de la distancia entre matrices. Existe una forma geométrica (y aparentemente natural) que permite definir la norma de una matriz. Esta forma toma en cuenta el comportamiento de la matriz como un operador:

Definición B.1.1. Si $A \in \mathbb{R}^{n \times n}$ y $\|\cdot\|$ una norma vectorial en \mathbb{R}^n , entonces la norma inducida de A se define por:

$$\|A\| := \sup_{x \neq 0} \frac{\|Ax\|}{\|x\|} = \sup_{\|x\|=1} \|Ax\|$$

Hay normas que no son inducidas (no se pueden obtener a partir de la definición B.1.1), tal es el caso de la norma de Frobenius. A continuación daremos fórmulas para calcular las normas matriciales inducidas por las normas 1, 2, ∞ .

1. $\|A\|_1 = \max_{1 \leq j \leq n} \|\vec{a}_j\| = \max_{1 \leq i \leq n} \sum_{i=1}^n |a_{ij}|$, máxima suma por columna.

2. Si $A \in \mathbb{R}^{n \times n}$, entonces

$$\|A\| = \sqrt{\rho(A^T A)}.$$

3. $\|A\|_\infty = \max_{1 \leq j \leq n} \sum_{j=1}^n |a_{ij}|$ máxima suma por renglón.

donde $\rho(A)$ es el **Radio espectral de una matriz**.

Dada una matriz $A \in \mathbb{R}^{n \times n}$, el radio espectral de A se define como el máximo de las magnitudes de sus valores propios, y se denota por $\rho(A)$; es decir,

$$\rho(A) = \max \{|\lambda| : \lambda \text{ es valor propio de } A\}.$$

Definición B.1.2. Para dos matrices $A, B \in \mathbb{R}^{n \times n}$ y una norma inducida $\|\cdot\|$, se satisface

1. $\|Ax\| \leq \|A\|\|x\|, \forall x \in \mathbb{R}^n$.

2. $\rho(A) \leq \|A\|$.

3. $\|AB\| \leq \|A\|\|B\|$

4. $\|A + B\| \leq \|A\| + \|B\|$

En análisis numérico existen 2 conceptos muy importantes

Condicionamiento: Se refiere al comportamiento bajo perturbaciones de un problema expresado matemáticamente.

Estabilidad: Se refiere al comportamiento bajo perturbaciones de un algoritmo utilizado para resolver un problema en una computadora.

Un problema *bién condicionado* es aquel en el que cualquier perturbación pequeña de $x \in X$ ocasiona un pequeño cambio en $f(x)$. En un problema *mal condicionado* pequeñas perturbaciones de $x \in X$ ocasiona grandes cambios en los resultados $f(x)$.

Definición B.1.3. (Número de condición absoluto) Sea δx una pequeña perturbación del dato $x \in X$, y denotemos por $\delta f = f(x+\delta x) - f(x)$, la correspondiente perturbación en el resultado. El número de condición absoluto del problema se define por

$$\hat{K} = \sup_{\delta x} \frac{\|\delta f\|_Y}{\|\delta x\|_X}$$

en donde $\|\cdot\|_X$ y $\|\cdot\|_Y$ representan normas vectoriales en los espacios normados X y Y respectivamente.

Definición B.1.4. (Número de condición relativo) Cuando estamos interesados en cambios relativos, necesitamos la noción de condición relativa. Por ejemplo, en el caso simple en el que $X = Y$ con norma vectorial $\|\cdot\|$, el número de condición relativo $K = K(x)$ del problema se define por

$$K = \sup_{\delta x} \left(\frac{\|\delta f\|/\|\delta x\|}{\|f(x)\|/\|x\|} \right) = \sup_{\delta x} \left(\frac{\|\delta f\|/\|f(x)\|}{\|\delta x\|/\|x\|} \right)$$

Observación: Si $A \in \mathbb{R}^n$ es una matriz simétrica no singular, $\|A\|_2 = \rho(A) = \lambda_{\max}(A)$, y en la norma-2:

$$K(A) = \text{cond}(A) = \frac{\lambda_{\max}(A)}{\lambda_{\min}(A)}$$

donde $\lambda_{\max}(A)$ es el máximo valor propio de A , y $\lambda_{\min}(A)$ es el mínimo valor propio de A . Para un estudio más profundo sobre el tema ver [46].

Apéndice C

Discrepancia

C.1 Distribución Uniforme Módulo 1.

Las discrepancia es la diferencia de dos mediadas del mismo tipo. Empezaremos en esta sección definiendo la discrepancia en el sentido de Lebesgue, y por último enunciaremos la discrepancia de *Kullback Leibler* ver [8]. Otra perspectiva de discrepancia usada en problemas inversos es el *principio de discrepancia de morozov* (para un estudio más profundo ver [48]) que no abordaremos en esta sección.

Definición C.1.1. *La sucesión $P = \{x_n\}$ $n = 1, 2, \dots$ de números reales se dice que es de **distribución uniforme modulo 1** (se abrevia como **d.u.mod 1**), si para cualquier par de números reales a, b tales que $0 \leq a < b \leq 1$ tenemos*

$$\lim_{N \rightarrow \infty} \frac{A([a, b]; P)}{N} = \lambda_1([a, b]) = b - a \quad (\text{C.1})$$

Donde λ_1 es la medida de Lebesgue de dimensión uno (en general se dice que es una medida de Lebesgue de dimensión d de un cierto conjunto B si; $B \subseteq I^d$ entonces $\lambda_d(B)$). Para $d = 1$, $d = 2$, $d = 3$, es la longitud, área y volumen respectivamente). La función

$$A([a, b]; P) = \sum_{n=1}^N \chi_{[a, b]}(x_n),$$

es llamada *la función contadora*, que utiliza a otra función, *la función característica* que se define como:

$$\chi_{[a, b]}(x_n) = \begin{cases} 1 & \text{si } x_n \in [a, b) \\ 0 & \text{en otro caso} \end{cases}$$

Notemos que tomando $\chi_{[a,b]}(x_n)$ con $[a,b] \subseteq I$, la ecuación (C.1) toma la forma:

$$\lim_{N \rightarrow \infty} \frac{1}{N} \sum_{n=1}^N \chi_{[a,b]}(x_n) = \int_0^1 \chi_{[a,b]}(x) dx \quad (\text{C.2})$$

Esto es porque la integral de la función característica es el tamaño del intervalo donde esta definida; es decir,

$$\begin{aligned} \int_0^1 \chi_{[a,b]} dx &= b - a \\ &= \lambda_1([a,b]) \end{aligned}$$

Nota: Por el momento trabajaremos en $d = 1$ posteriormente se verá la generalización.

C.2 Discrepancia

Nociones de discrepancia. La idea común de discrepancia es la diferencia entre dos valores medidos del mismo tipo. A continuación se enuncia la definición.

Definición C.2.1. Sea $P = \{x_1, \dots, x_N\}$ una sucesión finita de números reales. El número

$$D_N(x_1, \dots, x_N) = \sup_{0 \leq \alpha < \beta \leq 1} \left| \frac{A([\alpha, \beta]; P)}{N} - \mu_1([\alpha, \beta]) \right|, \quad (\text{C.3})$$

le llamaremos **discrepancia extrema** de la sucesión P . Donde μ_1 es la medida de Lebesgue de dimensión $d = 1$ (en este caso su medida es $(\beta - \alpha)$). Por convención a veces se utiliza $D_N(P)$ donde $P = \{x_1, \dots, x_N\}$ en lugar de $D_N(x_1, \dots, x_N)$.

Definición C.2.2. Sea $P = \{x_1, \dots, x_N\}$ una sucesión finita de números reales. El número

$$D_N^*(x_1, \dots, x_N) = \sup_{0 \leq \alpha < 1} \left| \frac{A([0, \alpha]; P)}{N} - \mu_1([0, \alpha]) \right| \quad (\text{C.4})$$

le llamaremos **discrepancia estrella** de la sucesión P . Donde μ_1 es la medida de Lebesgue de dimensión $d = 1$ (en este caso su medida es $(\alpha - 0) = \alpha$). A veces se escribe $D_N^*(P)$ en lugar de $D_N^*(x_1, \dots, x_N)$.

Definición C.2.3. Sea $P = \{x_1, \dots, x_N\}$ una sucesión finita en \mathbb{R}^d . Las discrepancias D_N y D_N^* para la dimensión d , se definen como:

$$D_N(P) = \sup_J \left| \frac{A(J; P)}{N} - \mu_d(J) \right| \quad (\text{C.5})$$

$$D_N^*(P) = \sup_{J^*} \left| \frac{A(J^*; P)}{N} - \mu_d(J^*) \right|, \quad (\text{C.6})$$

donde J representa a subintervalos de I^d de la forma $\prod_{i=1}^d [\alpha_i, \beta_i)$ y J^* representa a subintervalos de I^d de la forma $\prod_{j=1}^d [0, \alpha_j)$. Donde μ_d denota la medida de Lebesgue **d-dimensional**.

Teorema C.2.4. *Las discrepancias están relacionadas por la desigualdad siguiente*

$$D_N^*(P) \leq D_N(P) \leq 2^d D_N^*(P) \quad (\text{C.7})$$

Definición C.2.5. *(Discrepancia de Kullback Leibler.)*

Para distribuciones de probabilidad P, Q de una variable aleatoria discreta X (ver capítulo 2), la discrepancia de Kullback Leibler de Q dada P esta definida como

$$D_{KL}(P||Q) = \sum_i P(i) \log \frac{P(i)}{Q(i)}.$$

Una propiedad de esta discrepancia es la positividad.

$$D_{KL}(P||Q) \geq 0,$$

además esta discrepancia no es simétrica, esto es $D_{KL}(P||Q) \neq D_{KL}(Q||P)$ y no cumple con la desigualdad del triángulo. También tiene una propiedad muy interesante, concretamente, si $\{P_1, \dots, P_n\}$ es una sucesión de distribuciones de probabilidad, la discrepancia de Kullback-Leibler cumple

$$\lim_{n \rightarrow \infty} D_{KL}(P_n||Q) = 0,$$

en este caso escribimos $P_n \rightarrow^D P$.

Apéndice D

Producto de Kronecker y Matrices Circulantes.

Sea $A \in \mathbb{C}^{K \times L}$, $B \in \mathbb{C}^{M \times N}$ matrices con entradas dadas por tales que

$$A[k, l], \quad k = 0, 1, \dots, K - 1, \quad l = 0, 1, \dots, L - 1,$$

y

$$B[m, n], \quad m = 0, 1, \dots, M - 1, \quad n = 0, 1, \dots, N - 1.$$

respectivamente.

Se define el *producto matricial de Kronecker* $A \otimes B \in \mathbb{C}^{KM \times LN}$ como

$$A \otimes B = \begin{pmatrix} A[0, 0]B & A[0, 1]B & \cdots & A[0, L - 1]B \\ A[1, 0]B & A[1, 1]B & \cdots & A[1, L - 1]B \\ \vdots & \vdots & & \vdots \\ \vdots & \vdots & & \vdots \\ A[0, 0]B & A[0, 1]B & \cdots & A[0, L - 1]B \end{pmatrix} \quad (\text{D.1})$$

De modo que cada bloque de $A \otimes B$ es un múltiplo escalar de B , y los elementos de A funcionan como factores escalares, como se ve en el siguiente ejemplo

Ejemplo D.0.6.

$$\begin{aligned} \begin{bmatrix} 1 \\ 2 \end{bmatrix} \otimes \begin{bmatrix} 3 & 5 \end{bmatrix} &= \begin{bmatrix} 1 \begin{bmatrix} 3 & 5 \end{bmatrix} \\ 2 \begin{bmatrix} 3 & 5 \end{bmatrix} \end{bmatrix} = \begin{bmatrix} 3 & 5 \\ 6 & 10 \end{bmatrix}, \\ \begin{bmatrix} 3 & 5 \end{bmatrix} \otimes \begin{bmatrix} 1 \\ 2 \end{bmatrix} &= \begin{bmatrix} 3 \begin{bmatrix} 1 \\ 2 \end{bmatrix} & 5 \begin{bmatrix} 1 \\ 2 \end{bmatrix} \end{bmatrix} = \begin{bmatrix} 3 & 5 \\ 6 & 10 \end{bmatrix}, \end{aligned}$$

$$\begin{bmatrix} 1 & 2 & 3 \\ 2 & 5 & 6 \end{bmatrix} \otimes \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 2 & 0 & 3 & 0 \\ 0 & 1 & 0 & 2 & 0 & 3 \\ 4 & 0 & 5 & 0 & 6 & 0 \\ 0 & 4 & 0 & 5 & 0 & 6 \end{bmatrix},$$

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \otimes \begin{bmatrix} 1 & 2 & 3 \\ 2 & 5 & 6 \end{bmatrix} = \begin{bmatrix} 1 & 2 & 3 & 0 & 0 & 0 \\ 4 & 5 & 6 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 2 & 3 \\ 0 & 0 & 0 & 4 & 5 & 6 \end{bmatrix},$$

los últimos dos ejemplos nos muestran que $A \otimes B \neq B \otimes A$, además todos estos ejemplos nos ilustran la propiedad de exponentes matriciales vistos en la sección de Transformada Rápida de Fourier. Tal propiedad podemos reescribirla de la siguiente forma

$$A^{(p)} = I_p \otimes A, \quad (\text{D.2})$$

de forma análoga se pueden definir las siguientes propiedades (equivalentes a las propiedades de exponentes matriciales) del producto de Kronecker.

$$I_Q \otimes (I_P \otimes A) = I_{PQ} \otimes A, \quad (\text{D.3})$$

$$I_P \otimes (AB) = (I_P \otimes A)(I_P \otimes B) \text{ si } AB \text{ esta definido,} \quad (\text{D.4})$$

$$I_P \otimes (\alpha A) = \alpha(I_P \otimes A) \text{ cuando } \alpha \text{ escalar,} \quad (\text{D.5})$$

$$(I_P \otimes A)^T = I_P \otimes A^T, \quad (\text{D.6})$$

Es bastante fácil verificar estas propiedades, al igual que las siguientes a partir de la definición C.8

$$A \otimes (B \otimes C) = (A \otimes B) \otimes C, \quad (\text{D.7})$$

$$(A + B) \otimes C = (A \otimes C) + (B \otimes C) \text{ } A + B \text{ definida,} \quad (\text{D.8})$$

$$C \otimes (A + B) = (C \otimes A) + (C \otimes B) \text{ } A + B \text{ definida,} \quad (\text{D.9})$$

$$(\alpha A) \otimes B = \alpha(A \otimes B) = A \otimes (\alpha B), \quad (\text{D.10})$$

$$(A \otimes B)(C \otimes D) = (AC) \otimes (BD) \text{ } AC, BD \text{ definidas,} \quad (\text{D.11})$$

$$(A \otimes B)^T = A^T \otimes B^T, \quad (\text{D.12})$$

$$(A \otimes B)^{-1} = A^{-1} \otimes B^{-1} \text{ } A^{-1}, B^{-1} \text{ definidas,} \quad (\text{D.13})$$

$$I_P \otimes I_Q = I_{PQ} \text{ cuando } P, Q = 1, 2, \dots \quad (\text{D.14})$$

$$S_{P,Q}(A \otimes B) = (B \otimes A)S_{P,Q}, \quad (\text{D.15})$$

$$(\text{D.16})$$

donde $S_{P,Q}$ es llamada matriz de conmutación que tiene propiedades interesantes. Usando esta matriz de conmutación podemos reescribir la relación (1.22) de una forma alternativa

$$\mathcal{F}_{2M} = Q_{2M} S_{M,2} (F_M \otimes I_2).$$

Esta relación es una factorización de Transformada Rápida de Fourier (FFT, por sus siglas en ingles).

Matrices Circulantes.

Una matriz *circulante* es un tipo especial de matriz de *Toeplitz* ver [24] en el que cada vector fila se hace girar un elemento a la derecha con respecto al vector fila anterior. En análisis numérico las matrices circulantes son importantes porque son diagonalizadas por una transformada discreta de Fourier, y por lo tanto, las ecuaciones lineales que los contienen pueden resolverse rápidamente mediante una transformación rápida de Fourier. Puede ser interpretado analíticamente como el núcleo esencial de un operador de convolución en el *grupo cíclico* $\mathbb{Z}/n\mathbb{Z}$.

Definición D.0.7.

Dados $n \in \mathbb{Z}_+$ y un vector $c = (c_1, \dots, c_n)$, la matriz C de tamaño $n \times n$ está definida como

$$C = \text{circ}(c) = \begin{pmatrix} c_0 & c_{n-1} & c_{n-2} & \dots & c_1 \\ c_1 & c_0 & c_{n-1} & \dots & c_2 \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ c_{n-2} & \dots & c_1 & c_0 & c_{n-1} \\ c_{n-1} & \dots & c_2 & c_1 & c_0 \end{pmatrix}$$

es llamada matriz circulante de orden n .

El conjunto de matrices circulantes n - dimensionales pueden interpretarse como el espacio de funciones en el *grupo cíclico*⁴ Por ejemplo, si $G = (G_0, G_1, G_2, G_3, G_4, G_5)$ es un grupo, entonces $g_6 = g_0$, y G es cíclico. De hecho, G es esencialmente el mismo (es decir, *isomorfo* a) el conjunto $(0, 1, 2, 3, 4, 5)$ con módulo 6. Por ejemplo, $1 + 2 = 3 \pmod{6}$ corresponde a $G_1 G_2 = G_3$, y $2 + 5 = 1 \pmod{6}$ corresponde a $g_1 g_2 = g_3$ $g_7 g_1 = 1$, y así sucesivamente. Uno puede utilizar el isomorfismo $(gi) = i$ de orden n , $\mathbb{Z}/n\mathbb{Z}$, o equivalente, el *anillo de grupo* ver [25].

Los vectores propios de una matriz circulante de un tamaño determinado son las columnas de la matriz de Fourier del mismo tamaño

$$C = W_n^{-1} \text{diag}(W_n c) W_n,$$

donde c es la primer columna de la matriz C , $W_n = e^{-2\pi i k j / n}$. Así los valores propios de C están dados por el producto $W_n c$ que se puede calcular con una transformada rápida de Fourier (FFT).

⁴Un grupo G se llama cíclico si existe un elemento g en G tal que $G = \langle g \rangle = (g \mid n)$ es un entero. Puesto que cualquier grupo generado por un elemento en un grupo es un subgrupo de ese grupo, demostrando que el subgrupo de un grupo G que contiene G en sí mismo, es suficiente para demostrar que G es cíclico ver [25].

Solución de ecuaciones lineales con matrices circulantes.

Dado un sistema de ecuaciones en forma matricial

$$Cx = b$$

dónde C es una matriz circulante de tamaño n podemos escribir el sistema de ecuaciones como la convolución discreta

$$c * x = b,$$

dónde c es la primera columna de la matriz C . Usando el teorema de convolución (teorema 1.07) y usando la transformada discreta de Fourier (definición 1.11) tenemos

$$\mathcal{F}_n(c * x) = \mathcal{F}_n(c)\mathcal{F}_n(x) = \mathcal{F}_n(b),$$

de modo que

$$x = \mathcal{F}_n^{-1} \left[\left(\frac{(\mathcal{F}_n(b))_l}{(\mathcal{F}_n(c))_l} \right)_{l \in \mathbb{Z}} \right]$$

Este algoritmo es más rápido que la eliminación de Gauss estándar, sobre todo cuando se utiliza un algoritmo rápido de Fourier.

Definición D.0.8.

Sean A_1, \dots, A_m matrices cuadradas, cada una de orden n . Por *bloque circulante* de tipo (m, n) y orden mn entendemos la matriz de tamaño $mn \times mn$ dada por

$$\text{circ}(A_1, \dots, A_m) = \begin{pmatrix} A_1 & A_2 & A_3 & \dots & A_m \\ A_m & A_1 & A_2 & \dots & A_{m-1} \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ A_2 & A_3 & A_4 & \dots & A_1 \end{pmatrix}$$

Es decir, son matrices circulantes donde cada elemento es una matriz. Para un estudio más profundo de estas matrices ver [17] y [18].

Bibliografía

- [1] Heinz W. Engl. *Inverse Problems*. Sociedad Matemática Mexicana 1995.
- [2] Devore J. *Probabilidad y Estadística para Ingeniería y Ciencias*. 4a ed. International Thomson Editores, México, 1998.
- [3] Bartle, R. G. *The Elements of Real Analysis*. New York: John Wiley and Sons, Inc., 1964.
- [4] Papoulis. *Probability, Random Variables and Stochastic Processes*. Mc Graw-Hill 1965.
- [5] Umberto Neri. *Singular Integrals*. Lecture Notes in Mathematics. Springer-Verlag. 1971
- [6] Patrizio Campisi, Karen Egiazarian. *Blind Image Deconvolution*. Taylor and Francis Group, LLC 2007.
- [7] Alexander D. P, Zayed M. R. *Adaptive Filtering Primer with Matlab*. Taylor and Francis Group, LLC 2006.
- [8] Curtis R. Vogel. *Computational Methods for Inverse Problems*. Frontiers in applied mathematics Siam. 2002.
- [9] A.P. Dempster, N. M. Laird, D.B. Rubin. *Maximum Likelihood from Incomplete Data via the EM Algorithm*. Journal of the Royal Statistical Society. Series B, Vol. 39, No.1. 1976.
- [10] Martin A. Taner. *Tools for Statistical Inference*. Springer-Verlag 1996.
- [11] Joseph J. K., D. W. Fitzgerald. *Numerical Bayesian Methods Applied to Signal Processing*. Springer-Verlag 1995.
- [12] Ludmig F. Gerard T. *Multivariate S. Modelling Based on Generalized Linear Models*. Springer-Verlag 2001.
- [13] Mark R. A. K. Katsangelos *Digital Image Restoration*. IEEE Signal Processing Magazine 1997.
- [14] Geoffrey J.M. Thriyambakam K. *The EM Algorithm and Extensions*. John Wiley and Sons, inc. 1997.

- [15] Y.Vardi and Lee *The EM Algoritim and Extensions*. John Wiley and Sons, inc. 1997.
- [16] D.A Fish, A. M. Brinicombe, and E. R. Pike *Blind deconvolution by means of the Richardson-Lucy algoritm*. John Wiley and Sons, inc. 1997.
- [17] Bellman, R. *Introduction to Matrix Analysis*, 2nd ed., McGraw-Hill, New York, 1970.
- [18] Davis, P. J. *Circulant Matrices*, John Wiley and sons, New York, 1979.
- [19] Rafael Molina, Jorge Nuñez *A Bayesian Perspective*, IEEE signal processing magazine, 2001.
- [20] Radford M. Neal, Geoffrey E. Hinton *A view of the Algorithm that justifies incremental, sparse, and other variants*, University of Toronto, Ontario, Canada.
- [21] Deepa Kundur and Dimitrios Hatzinakos *Blind Image Deconvolution*, IEEE signal processing magazine, 1996.
- [22] L.B. Lucy *An iterative technique for the rectification of observed distributions*, Departments of physics and Astronomy, The University of Pittsburgh, Pennsylvania, 1974.
- [23] Rafael C. Gonzalez, Richard E. Woods *Digital Image Processing*, 2008 by Pearson Education, Inc.
- [24] Robert M. Gray *Toeplitz and Circulant Matrices: A review*, Department of Electrical Engineering Stanford University Stanford 94305, USA
- [25] Herstein, *Abstract algebra*, Prentice Hall 3ra ed. 1996.
- [26] Patrick Billingsley *Probability and Measure*, Wiley-Interscience; 3 edition (April 17, 1995).
- [27] S. Ross *Introduction to Probability Models*, Publisher: Academic Press 1997.
- [28] G.Milton Wing, John D. Zahrt. *A primer on Integral Equations of the First Kind*, Society for Industrial and Applied Mathematics © 1991.
- [29] Joaquin Delgado, Mario Medina. *Ploblemas inversos en procesamiento de im ágenes*, primer coloquio del Departamento de Matemáticas UAM-I.
- [30] Qi Shan, Jiaya Jia, Aseem A.. *High-quality Motion Deblurring from a Single Image*, Departament of Computer Science and Engineering, The Chinese University of Hong Kong.
- [31] William Hadley Richardson. *Bayesian-Based Iterative Method of Image Restoration*, Visibility Laboratory, University of California, San Diego, San Diego, California 1970.

- [32] Todd K. Moon *The Expectation-Maximization Algorithm*, IEEE Signal Processing Magazine 1996
- [33] Frank Dellaert *The Expectation-Maximization Algorithm*", College of computing, Georgia Institute of Technology 2002
- [34] Sean Borman *The Expectation-Maximization Algorithm; A short tutorial*, July 2004 last upted january 09, 2009
- [35] Cheng Xiang Zhai *A note on the Expectation-Maximization Algorithm*, Departmente of computer science, University of Illinois at Urbana-Champaing, 2004
- [36] Richard A. Levine and George Casella *Implementations of the Monte Carlo EM Algorithm*, American Statical Association, Institute of Mathematical Statics and Interface Foundation of North America, 2001
- [37] Nir Friedman *The Bayesian Structural EM Algorithm*, Computer Science Division, 3087 soda Hall, University of California, Berkeley
- [38] L.A. Shepp and Y. Vardi *Maximum Likelihood Reconstruction for Emission Tomography*,IEEE 1982
- [39] E. Somersalo *Computational Methods in inverse problems*,IEEE 1982
- [40] Ming Jiang and Jing Wang *Thechniques in blind deblurring of spiral computed tomography images and their aplications*,LMAM, School of Mathematical Sciencies, Peking University
- [41] Y. Vardi and D. Lee, *From image deblurring to optimal investments: Maximum likelihood solutions for positive linear inverse problems*, Journal of the Royal Statistical 1992.
- [42] Richard L. White, *Restoration of images and spectra from the hobble space telescope*, 1990.
- [43] Bi and Borner, A and AS 108, 409. 1994.
- [44] B. Chanda and D. Dutta Mujumder, *Digital Image Processing and Analysis.*, Prentice Hall 2000.
- [45] Albrecht Böttcher, Bernd Silbermann, *Analysis of Toeplitz Operators.*, © Springer-Verlag Berlin Heidelberg 2006.
- [46] L. Hctor Jurez V, *Análisis Numérico.*, Departamento de Matemáticas, Universidad Autónoma Metropolitana. 2007.
- [47] E. Isaacson, H. B. Keller, *Analysis of Numerical Methods*, Wiley 1966.
- [48] Andreas Kirsch, *An Introduction to the mathematical Theory of Inverse Problems* , Springer-Verlag 1996.

Links

[49] <http://www.rippon.com/code/fft842.html>