



UNIVERSIDAD NACIONAL DE COLOMBIA

Restauración de Imágenes Borrosas Usando Programación Lineal

José Exequiel Fuentes Gil

Universidad Nacional de Colombia
Facultad de Ciencias, Departamento de Matemáticas
Bogotá D.C., Colombia
2020

Restauración de Imágenes Borrosas Usando Programación Lineal

José Exequiel Fuentes Gil

Tesis de maestría presentada como requisito para optar al título de:
Magister en Matemática Aplicada

Director:
Dr. Jorge Mauricio Ruiz Vera

Línea de Investigación:
Matemática aplicada, procesamiento de imágenes

Universidad Nacional de Colombia
Facultad de Ciencias, Departamento de Matemáticas
Bogotá D.C., Colombia
2020

A mi familia,
a mis amigos,
a ti, que estás leyendo esto.

Agradecimientos

Al profesor Jorge Mauricio Ruiz Vera, por ser el guía de este trabajo y apoyarlo cuando fue necesario, a Alvaro Andrés Sandino, Diego Fernando Bravo Higuera, Jorge Daniel Jara Hurtado, Leidy Tatiana Molano Muñoz, Miguel Ángel Beltrán Velásquez, Nicolás Múnica Garzón, al profesor Edgar Eduardo Romero Castro y a los integrantes grupo de investigación CIM@LAB por su apoyo en los con las ideas, la realización y la divulgación de producción académica relacionada con imágenes médicas realizada en este trabajo.

Resumen

En el proceso de captura de imágenes es común tratar con imágenes deterioradas. Estas aparecen en diversos ámbitos como lo son: astronomía, medicina, entre otros. En este trabajo se desarrolla un método para la restauración de imágenes borrosas basado en el planteamiento de ecuaciones integrales mal propuestas. La solución de la ecuación integral es vista como el mínimo de un problema de optimización considerado en la norma $\|\cdot\|_{L_1}$. De esta forma es expresado como un problema de programación lineal. También, se encontró que el problema debe ser adaptado al caso particular de la restauración de imágenes agregando términos extra al modelo originalmente propuesto. Además de evaluar su eficiencia y eficacia, se muestra que este método es competitivo con respecto a otros propuestos inicialmente y que puede ser usado en diferentes ámbitos mostrando resultados satisfactorios.

Palabras clave: Imágenes borrosas, problemas inversos, problemas mal propuestos, programación lineal, regularización.

Abstract

In the process of capturing images, it is common to deal with deteriorated images. These appear in various fields such as: astronomy, medicine, among others. In this work, a method is developed for the restoration of blurred images based on the approach of poorly proposed integral equations. The solution of the integral equation is seen as the minimum of a limited optimization problem in the norm $\|\cdot\|_{L_1}$. In this way, it is expressed as a linear programming problem. Also, it was found that the problem needs to be adapted to the particular image restoration problem by adding additional terms to the originally proposed model. In addition, evaluating its efficiency and effectiveness, this method is shown to be competitive with respect to other ones, and it can be used in different environments, showing satisfactory results.

Keywords: Blurry images, ill posed problems, inverse problems, linear programming, regularization.

Lista de Figuras

1-1	Imagen tomada por el telescopio Hubble. Izquierda: imagen degradada, Derecha: imagen restaurada. Tomado de [17]	2
1-2	Imagen tomada por una cámara de seguridad en un incidente ocurrido en abril del 2019 en el centro comercial Andino ubicado en Bogotá. Tomado de [6] . .	2
1-3	Imágenes de ultrasonido. (a) Imagen original, (b) Imagen restaurada, (c) Diferencia entre la imagen real y su restauración. Tomado de [1]	3
2-1	Comparación entre las normas L_1 y L_2 optimizando funciones lineales.	10
2-2	Ilustración de los vértices óptimos. a) Curva de nivel paralela, b) Curva de nivel no paralela	16
2-3	Construcción de la tabla asociada al método simplex.	17
2-4	Selección de la variable básica entrante.	18
2-5	Selección de la variable básica saliente. Aquí $x_i = x_i$	18
2-6	Eliminación de Gauss-Jordan aplicada a la columna escogida. La barra sobre los elementos indica que posiblemente han cambiado sus valores debido a la eliminación.	19
2-7	Ilustración del algoritmo simplex sobre el conjunto factible.	19
2-8	Ilustración del algoritmo de punto interior sobre el conjunto factible.	21
3-1	Imágenes degradadas con la ecuación diferencial (3-5). (a) Imagen real, (b) Imagen borrosa. Tomado de [20].	27
3-2	Imágenes degradadas con la ecuación diferencial (3-5). (a) Imagen real, (b) Imagen borrosa. Tomado de [11].	27
3-3	Ejemplo una PSF exponencial de desenfoque simétrica con varianza $\sigma = 10$ y de tamaño 75×75 . (a) Imagen real. Tomado de [19], (b) Imagen degradada con la PSF, (c) Función que representa la PSF, (d) Matriz de convolución que representa a la función.	29
3-4	Ejemplo una PSF exponencial de desenfoque simétrica con un parámetro $q = 1,2$ y de tamaño 75×75 . (a) Función que representa la PSF, (b) Matriz de convolución que representa a la función.	30
3-5	Ejemplo una PSF de movimiento horizontal $\theta = 0$ y de tamaño 75×75 . (a) Imagen real, (b) Imagen degradada con la PSF, (c) Función que representa la PSF, (d) Matriz de convolución que representa a la función.	31

3-6	Ejemplo una PSF Moffat de varianza $\sigma = 8$, $\beta = 3$ y de tamaño 75×75 . (a) Imagen real, (b) Imagen degradada con la PSF, (c) Función que representa la PSF, (d) Matriz de convolución que representa a la función.	33
3-7	Restauración usando la matriz inversa de la ecuación (3-4). (a) Imagen original. Tomado de [43], (b) Imagen borrosa con 1% de ruido, (c) Imagen al aplicar la matriz inversa.	36
4-1	Imagen degradada con 0,05% de ruido y su restauración. (a) Imagen original. Tomado de [41], (b) Imagen borrosa, (c) Imagen restaurada.	41
4-2	Imagen degradada con 0,1% de ruido y su restauración. (a) Imagen original, (b) Imagen borrosa, (c) Imagen restaurada.	42
4-3	Imagen degradada con 2% de ruido y su restauración. (a) Imagen original, (b) Imagen borrosa, (c) Imagen restaurada.	43
4-4	Imagen degradada con 0,5% de ruido y su restauración. (a) Imagen original, (b) Imagen borrosa, (c) Imagen restaurada.	43
4-5	Imagen degradada con 1% de ruido y su restauración. (a) Imagen original, (b) Imagen borrosa, (c) Imagen restaurada.	44
4-6	Imagen degradada con 5% de ruido y su restauración. (a) Imagen original, (b) Imagen borrosa, (c) Imagen restaurada.	45
4-7	Imagen degradada con 0,5% de ruido y sus restauraciones. (a) Imagen original. Tomado de [39], (b) Imagen borrosa, (c) Imagen restaurada con el método simplex, (d) Imagen restaurada con el método de punto interior.	47
4-8	Imagen degradada con 0,5% de ruido y sus restauraciones. (a) Imagen original, (b) Imagen borrosa, (c) Imagen restaurada con el método simplex, (d) Imagen restaurada con el método de punto interior.	48
4-9	Ejemplos de funciones suaves y no suaves cuyas normas $\ \cdot\ _1$ son iguales. (a) Función suave cuya derivada tiene norma pequeña, (b) Función no suave cuya derivada tiene norma más grande que la función (a).	51
4-10	Núcleos de convolución para calcular las derivadas parciales de la imagen. (a) Derivada parcial en la dirección horizontal, (b) Derivada parcial en la dirección vertical.	52
4-11	Ejemplos de matrices A_c y A_r para una PSF exponencial de una imagen de tamaño 85×91 . (a) Matriz A_c , (b) Matriz A_r . Los puntos azules indican elementos no nulos en la matriz, n_z indica la cantidad de elementos no nulos en la matriz.	55
4-12	Matrices A_c y A_r truncadas de la figura 4-11. (a) Matriz A_c , (b) Matriz A_r . Los puntos azules indican elementos no nulos en la matriz, n_z indica la cantidad de elementos no nulos en la matriz.	56

4-13 Matrices A construidas directamente de las matrices de las figuras 4-11 y 4-12 . (a) Matriz A sin truncar, (b) Matriz A truncada. Los puntos azules indican elementos no nulos en la matriz, nz indica la cantidad de elementos no nulos en la matriz.	56
4-14 Imagen degradada con 5% de ruido y sus restauraciones usando el algoritmo de punto interior. (a) Imagen original, (b) Imagen borrosa, (c) Imagen restaurada con el modelo sin regularizar, (d) Imagen restaurada con el modelo regularizado.	60
4-15 Imagen degradada con 0,5% de ruido y sus restauraciones usando el algoritmo de punto interior. Tomado de [36]. (a) Imagen original, (b) Imagen borrosa, (c) Imagen restaurada con el modelo sin regularizar, (d) Imagen restaurada con el modelo regularizado.	61
4-16 Comparación de los esquemas de regularización. (a) Imagen original, (b) Imagen borrosa con 0,5% de ruido, (c) Imagen restaurada regularizando sólo la norma de la solución ($\alpha = 2 \times 10^{-2}$, $\gamma = 0$), (d) Imagen restaurada regularizando sólo la norma de la derivada ($\alpha = 0$, $\gamma = 1,5 \times 10^{-2}$), (e) Imagen restaurada regularizando ambas normas ($\alpha = 2 \times 10^{-2}$, $\gamma = 1,5 \times 10^{-2}$).	63
4-17 Imagen degradada con 3% de ruido y sus restauraciones. (a) Imagen original, (b) Imagen borrosa, (c) Imagen restaurada con método de Richardson-Lucy, (d) Imagen restaurada con método de deconvolución de Wiener, (e) Imagen restaurada con el método blind deconvolution, (f) Imagen restaurada con método BDNSM, (g) Imagen restaurada con método de regularización de Tikhonov, (h) Imagen restaurada con método propuesto.	67
4-18 Imagen degradada con 0,5% de ruido y sus restauraciones. (a) Imagen original, (b) Imagen borrosa, (c) Imagen restaurada con método de Richardson-Lucy, (d) Imagen restaurada con método de deconvolución de Wiener, (e) Imagen restaurada con método blind deconvolution, (f) Imagen restaurada con método BDNSM, (g) Imagen restaurada con método de regularización de Tikhonov, (h) Imagen restaurada con método propuesto.	69

Lista de Tablas

4-1	Resultados de la restauración de la imagen de la Figura 4-1	42
4-2	Resultados de la restauración de la imagen de la Figura 4-2	42
4-3	Resultados de la restauración de la imagen de la Figura 4-3	43
4-4	Resultados de la restauración de la imagen de la Figura 4-4	44
4-5	Resultados de la restauración de la imagen de la Figura 4-5	45
4-6	Resultados de la restauración de la imagen de la Figura 4-6	45
4-7	Resultados de las restauraciones de la imagen de la Figura 4-7	47
4-8	Resultados de las restauraciones de la imagen de la Figura 4-8	48
4-9	Resultados de las restauraciones de la imagen de la Figura 4-14	60
4-10	Resultados de las restauraciones de la imagen de la Figura 4-15	62
4-11	Resultados de las restauraciones de la imagen de la Figura 4-16 . (c), (d) y (e) se refieren a las imágenes de la Figura 4-16	64
4-12	Resultados de las restauraciones de la imagen de la Figura 4-17	67
4-13	Resultados de las restauraciones de la imagen de la Figura 4-18	69

Contenido

Agradecimientos	vii
Resumen	ix
Lista de figuras	xi
Lista de tablas	xv
Lista de símbolos	xviii
1 Introducción	1
1.1 Objetivos	5
2 Preliminares	7
2.1 Problemas inversos	7
2.2 Conjuntos, funciones convexas y problemas de optimización	11
2.2.1 Conjuntos y funciones convexas	11
2.2.2 Algoritmos de optimización	15
3 El modelo en la restauración de imágenes borrosas	25
3.1 Descripción del problema	25
3.2 Funciones de dispersión puntual (PSF)	26
3.2.1 PSF de desenfoque	28
3.2.2 PSF de movimiento	30
3.2.3 PSF degradado atmosférico	32
3.3 Limitaciones y dificultades	34
3.3.1 Tamaño de los sistemas de ecuaciones	34
3.3.2 Sensibilidad a los errores	34
3.4 El modelo desde la programación lineal	37
4 Restaurando imágenes	39
4.1 Medidas de desempeño	39
4.2 Restaurando imágenes de prueba	41
4.2.1 Borrado exponencial	41
4.2.2 Borrado de movimiento	43

4.2.3	Borrado atmosférico (Moffat)	44
4.3	El algoritmo simplex versus el algoritmo de punto interior	46
4.4	Regularización del problema	49
4.4.1	Regularización de la norma $\ \cdot\ _1$ de la solución	49
4.4.2	Regularización de la derivada $\ \cdot\ _1$ de la solución	50
4.5	Abordaje y dificultades del modelo regularizado	51
4.5.1	Abordaje	52
4.5.2	Dificultades	54
4.6	Modelo regularizado y modelo sin regularizar	59
4.6.1	Comparación del modelo regularizado y modelo sin regularizar	59
4.6.2	Comparación entre esquemas de regularización del modelo regularizado	62
4.7	Comparación contra otros métodos de deconvolución o deblurring	65
5	Aplicaciones y Productos	71
5.1	International Symposium on Medical Information Processing and Analysis .	71
6	Conclusiones y recomendaciones	82
6.1	Conclusiones	82
6.2	Recomendaciones	83
	Bibliografía	84

Lista de símbolos

Símbolos con letras latinas

Símbolo	Término
A	Matriz de tamaño $n \times m$
b	Imagen borrosa
x^*	Solución aproximada
$\ \cdot\ _{L_p}$	Norma p en el espacio de Banach L_p
$\ \cdot\ _p$	Norma p en el espacio vectorial \mathbb{R}^n

Símbolos con letras griegas

Símbolo	Término
α	Coefficiente de regularización para la norma de la solución
γ	Coefficiente de regularización para la norma de la derivada de la solución
ΔA	Perturbación de la matriz A
Δx	Dirección a seguir al modificar al vector x

Abreviaturas

Símbolo	Término
PSF	Función de propagación puntual
SVD	Descomposición en valores singulares
PSNR	Proporción máxima de señal a ruido
SSIM	Índice de similaridad estructural
ECM	Error cuadrático medio
Err	Error relativo en la norma L_1
KKT	Condiciones de optimalidad de Karush-Kuhn-Tucker

1 Introducción

Cuando se dispone a tomar una fotografía lo ideal es capturar perfectamente la escena a retratar, es decir, que con la fotografía sea posible recrear con precisión la escena que captura. Esto es imposible en primera medida por la interpretación del objeto que captura la imagen (fotografía), es decir, de aquello que quiere capturar. En esencia se busca responder a *¿qué* es una imagen? y *¿cómo* se interpreta?. Una vez se acuerda que la fotografía es sólo *una* interpretación de lo que está sucediendo, se considera que ésta interpretación sufre de problemas (limitaciones). Algunos de éstos ocurren al momento de la captura y el procesamiento subsiguiente y pueden deberse a:

- Factores ambientales (nubosidad, humedad).
- La escena en sí misma, es decir, el movimiento (si lo hay) de la misma provoca que no se pueda capturar a la perfección.
- Imperfecciones del dispositivo de captura (por ejemplo el lente de una cámara o telescopio).

En vista de lo anterior, el objetivo será, recuperar la imagen que representa la escena, a partir del conocimiento de la imagen alterada y de las posibles causas de la alteración.

La solución de éste problema ha sido de gran interés en diversas áreas. Por ejemplo, cuando se lanzó el telescopio Hubble el 26 de abril de 1990 las primeras imágenes que enviaba eran borrosas debido a una falla en el espejo principal. Este tipo de situaciones en las que es muy difícil, muy costoso o no se puede tomar de nuevo la fotografía hacen que se obligue a trabajar con la fotografía dañada de forma que se intente recuperar la información que originalmente se pretendía capturar.



Figura 1-1: Imagen tomada por el telescopio Hubble. Izquierda: imagen degradada, Derecha: imagen restaurada. Tomado de [17]

También ha ganado popularidad en los sistemas de cámaras de seguridad donde en muchos casos las imágenes tomadas de baja calidad y borrosas, que a primera vista no parecen ser útiles en la investigación de un crimen.



Figura 1-2: Imagen tomada por una cámara de seguridad en un incidente ocurrido en abril del 2019 en el centro comercial Andino ubicado en Bogotá. Tomado de [6]

Otra área de aplicación es en la medicina, por ejemplo las imágenes de ultrasonido, éstas imágenes contienen componentes borrosas y ruidosas debido al movimiento de las estructuras que apuntan a capturar y a los componentes que están en su interior. Es importante entonces, usar una técnica que permita distinguir de mejor forma las estructuras que se desean identificar, por ejemplo, hallar anomalías en el funcionamiento de estas estructuras.

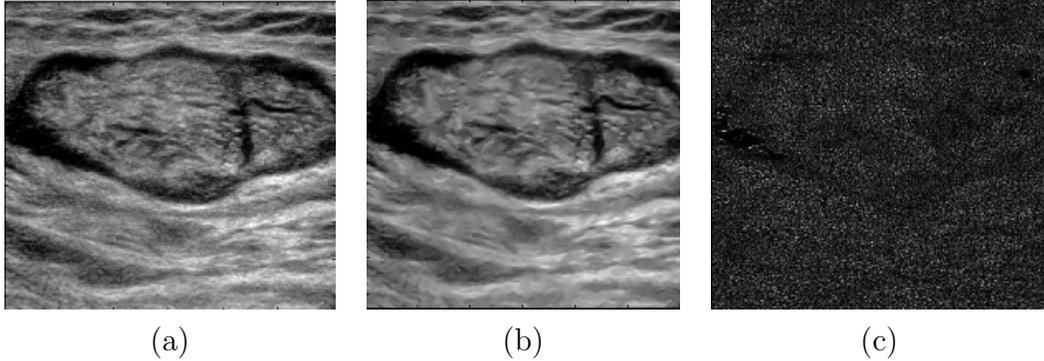


Figura 1-3: Imágenes de ultrasonido. (a) Imagen original, (b) Imagen restaurada, (c) Diferencia entre la imagen real y su restauración. Tomado de [1]

En este trabajo se estudia e implementa un método que permita restaurar imágenes borrosas usando técnicas de programación lineal. Resolver un problema en la norma $\|\cdot\|_{L_1}$, en comparación con la norma $\|\cdot\|_{L_2}$ tiene la ventaja que la solución es menos sensible a valores atípicos u observaciones anormales que ocurran en el proceso de borrado de una imagen. La solución de este problema será la imagen restaurada, la cual tiene propiedades deseables propias de esta norma, tales como que la solución posee bastantes ceros en sus componentes y la menor sensibilidad a los datos atípicos, que no suelen considerar los métodos convencionales al minimizar otras funciones. También se busca estudiar su efectividad, ventajas y desventajas con respecto a otros métodos ya existentes.

En particular el proceso de degradación de una imagen se modela de la forma

$$\psi(x) = \int_G K(x,t)\phi(t)dt, \quad (1-1)$$

donde $\psi(x)$ (fotografía tomada) y $\phi(x)$ (imagen real) son funciones escalares continuas cuyo dominio es un subconjunto acotado del plano \mathbb{R}^2 que llamamos G y $K(x,t)$ es una función escalar continua cuyo dominio es $G \times G$, que representa el proceso de degradación y registro de la imagen real. Con esta ecuación se busca encontrar $\phi(x)$ teniendo a $\psi(x)$ y una estimación de $K(x,t)$.

Atendiendo a las dificultades teóricas y prácticas que presentan este tipo de ecuaciones debido a las pequeñas perturbaciones, se busca discretizar este problema para abordarlo usando técnicas pertenecientes al análisis numérico. Es así, que para resolver el problema de restauración de imágenes se han propuesto diversos métodos a tener en cuenta. Por ejemplo, el método de deconvolución de Richardson-Lucy [37] consiste en un método iterativo para reconstruir imágenes borrosas con un núcleo K predefinido. Este método se apoya en el teorema de Bayes al crear un método iterativo que va refinando la imagen hasta parar

cuando el ruido se haga muy evidente. Este método se describe como

$$X_i^{t+1} = X_i^t \sum_{j=1}^N \frac{B_j}{C_j} K_{ij}, \quad C_j = \sum_{l=1}^N X_l^t K_{jl}. \quad (1-2)$$

Donde X_i^t es la estimación de la imagen dada el la iteración t y el pixel i , la matriz cuyas componentes son K_{ij} es la matriz que representa al núcleo de convolución y B_i es el pixel i de la imagen borrosa. Este método converge empíricamente a la solución de máxima verosimilitud. Posee ventajas con respecto al cómputo ya que es posible efectuar los pasos rápidamente. Sin embargo, es un método muy sensible a altos niveles de ruido. También, se ha propuesto usar la transformada de Fourier para multiplicar por un núcleo inverso en el dominio de la frecuencia [5]. La idea es considerar el hecho la imagen borrosa está descrita por la convolución $B = X * K$ y usa el teorema de convolución de la transformada de Fourier en el cual

$$\mathcal{F}\{B\} = \mathcal{F}\{X * K\} = \mathcal{F}\{X\}\mathcal{F}\{K\}. \quad (1-3)$$

Donde $\mathcal{F}\{\cdot\}$ indica la transformada de Fourier. Así, es posible encontrar la imagen original haciendo $G = 1/\mathcal{F}\{K\}$ y $X = \mathcal{F}^{-1}\{\mathcal{F}\{B\}G\}$. Esto hace que el método sea bastante eficiente al tener que realizar una multiplicación y dos transformadas de Fourier. No obstante, en presencia de ruido en la imagen borrosa suelen aparecer ondas alrededor de los bordes de la imagen restaurada. Esto indica que hay ruido en las frecuencias altas en la imagen que resulta magnificado. El filtro de Winner [13] funciona en cierta medida de forma similar al método de de Richardson-Lucy, en este método se asume que la imagen borrosa es producto de una convolución con un núcleo K añadiendo ruido aditivo. En este método se busca minimizar el error cuadrático medio en términos de la esperanza $\mathbb{E}[(X - B)^2]$. Esto significa que tanto la imagen real X , la imagen borrosa B y el ruido E son considerados como variables aleatorias. La solución a este problema está dada en el dominio de la frecuencia dado por la transformada de Fourier

$$\mathcal{F}\{\hat{X}\} = \mathcal{F}\{B\} \left[\left(\frac{1}{\mathcal{F}\{K\}} \right) \left(\frac{1}{1 + 1/(|\mathcal{F}\{K\}|^2 SNR)} \right) \right], \quad SNR = \frac{\mathbb{E}[|\mathcal{F}\{X\}|^2]}{\mathbb{E}[|\mathcal{F}\{E\}|^2]}. \quad (1-4)$$

Donde \hat{X} es la estimación de la imagen real. Al considerar el factor SNR se observa que se necesita información a priori acerca del nivel de ruido de la imagen y la imagen en si misma. En [34] y [16] se usa el método de regularización de Tikhonov planteando un problema de optimización al minimizar el error cuadrático de la norma de la solución X con una constante $\alpha > 0$, llamada parámetro de regularización junto con el error cuadrático de del modelo de degradado de la imagen dado por la convolución de X con el núcleo K .

$$f(X) = \|K * X - B\|_{L_2}^2 + \alpha \|X\|_{L_2}^2. \quad (1-5)$$

La ventaja de este método es su eficiencia, al igual que los anteriores se apoya en el uso de la norma euclídea, lo que lo hace sencillo de tratar. En [28] se efectúa un proceso de regularización sobre la solución similar al realizado en [34], salvo que el núcleo de convolución K y el cociente $\frac{\|X\|_{L_1}}{\|X\|_{L_2}}$ son los términos regularizados usando constantes $\lambda, \psi > 0$ en el funcional

$$f(X, K) = \lambda \|K * X - B\|_{L_2}^2 + \frac{\|X\|_{L_1}}{\|X\|_{L_2}} + \psi \|K\|_{L_1}^2. \quad (1-6)$$

Algo que comparten estos métodos es que en general poseen ventaja respecto al costo computacional que tienen, esto se debe a que minimizan funciones muy relacionadas con la norma $\|\cdot\|_{L_2}$. Esto permite tomar ventaja de propiedades como que esta norma proviene de un producto interior o que es posible usar la ortogonalidad entre vectores. Sin embargo, esta norma puede ser muy sensible a aquellas observaciones anormales de grandes magnitudes y esto influye en el proceso de minimización. Por último, en [29] se presenta el método de restauración de imágenes usando un pre-procesamiento previo al usar la descomposición en valores singulares del operador

$$H = U\Lambda V^\top \quad (1-7)$$

que representa la convolución con la imagen, para luego usar programación lineal en la restauración de la imagen. En este sentido, este trabajo está orientado en la misma dirección este artículo. Aquí se exploran diferentes modelos de núcleos que simulan la degradación de las imágenes, además de estudiar el costo que estos tienen al usar varios algoritmos en el proceso de optimización. También, se hicieron mejoras para disminuir el costo computacional en el proceso de optimización y se añadieron términos de regularización que muestran mejoras sobre los resultados tomando como base algunos resultados del modelo originalmente propuesto. El código fuente usado en este trabajo puede ser consultado en la siguiente dirección URL:

<https://github.com/Xioeng/Restauracion-de-imagenes>

1.1. Objetivos

El objetivo central de este trabajo es restaurar imágenes borrosas usando técnicas de programación lineal. Para ello se llevan a cabo los siguientes objetivos específicos:

- Construir modelos de los diferentes tipos de borrado.
- Estudiar e implementar algoritmos que permitan resolver los problemas de optimización mencionados.

- Resolver los diferentes problemas de optimización derivados de los modelos de borrado e imágenes borrosas con diferentes algoritmos de optimización.
- Mejorar el rendimiento de los algoritmos de programación lineal y mejorar el modelo de programación lineal al agregar términos de regularización.
- Comparar resultados con otros métodos que buscan resolver problemas iguales o similares.

El texto se divide de la siguiente forma: el Capítulo 2 presenta algunos conceptos y resultados que son usados en el desarrollo del trabajo, el Capítulo 3 presenta modelo con la que se aborda este problema, el Capítulo 4 presenta algunos resultados comparando diferentes algoritmos y una mejora al modelo original, el Capítulo 5 presenta los productos elaborados durante el proyecto y el Capítulo 6 muestra las conclusiones acerca del trabajo.

2 Preliminares

A continuación se presentan algunos de los conceptos más importantes que son usados en el trabajo. De estos se destacan, problemas inversos, problemas de optimización y problemas de programación lineal.

2.1. Problemas inversos

Con el desarrollo del procesamiento digital de imágenes, los problemas inversos han cobrado importancia recientemente. En particular, estas áreas están relacionadas con la recuperación de información sobre un objeto a partir de observaciones indirectas, incompletas o ruidosas. Desde el punto de vista matemático dichos problemas, como en [34] se pueden ver de la siguiente manera:

Definición 2.1.1 (Problema directo e inverso). *Sean V e W espacios vectoriales normados, $A : V \rightarrow W$ una función (también llamada operador) y $\phi \in V$, $\psi \in W$ tales que $A\phi = \psi$. El problema directo consiste en calcular la salida ψ dada la entrada ϕ y el problema inverso consiste en calcular la entrada ϕ dada la salida ψ .*

En este caso nos interesa encontrar soluciones al problema inverso, entonces, surgen varias preguntas como aquellas sobre la existencia, la naturaleza y la cantidad de soluciones dada una salida fija ψ . Para esto se definen los problemas bien propuestos [34]. Un problema es bien propuesto, en el sentido de Hammadard, si:

1. Existe su solución y esta es única
2. La solución es continuamente dependiente de los datos iniciales.

Al llevar esto a los espacios vectoriales, se obtiene una definición más precisa:

Definición 2.1.2 (Problema bien propuesto). *Sean V e W espacios vectoriales normados, $U \subseteq V$, $T \subseteq W$ y $A : U \rightarrow T$ un operador. La ecuación*

$$A\phi = \psi \tag{2-1}$$

se dice que está bien propuesta si el operador $A^{-1} : T \rightarrow U$ existe y es continuo.

Particularmente en este proyecto estamos interesados en aquellos operadores que son producto de ecuaciones integrales, en particular en las que son de primer tipo.

Definición 2.1.3 (Ecuación integral de Fredholm de primer orden). Sean V un espacio vectorial normado contenido en el espacio vectorial de las funciones $F = \{f | f : G \rightarrow \mathbb{R}\}$ donde $G \subseteq \mathbb{R}^n$, $\psi, \phi \in F$ y $K : G \times G \rightarrow \mathbb{R}$ una función denominada kernel o núcleo. La ecuación integral de Fredholm de primer orden se define como

$$\psi(x) = \int_G K(x, t)\phi(t)dt \quad (2-2)$$

Tomando en cuenta la definición de problema inverso, es de interés dado ψ encontrar ϕ . Sin embargo, si consideramos el operador A como el definido en (2.1.3), esto es

$$\begin{aligned} A : V \subseteq F &\longrightarrow A(V) \\ \phi(x) &\longrightarrow A\phi(x) = \int_G K(x, t)\phi(t)dt. \end{aligned} \quad (2-3)$$

Este operador resulta ser un operador compacto:

Teorema 2.1.1 (Compacidad del operador integral). Sea A el operador definido como en la definición 2.1.3 y en (2-3) con $V = C(G) = \{f : f \text{ es continua en } G\}$, $A(V) \subseteq C(G)$ y $K(x, t) \in C(G \times G)$, entonces, el operador A es compacto.

Demostración. La demostración se encuentra en [26].

No obstante, al invertir estos operadores (si es que tal inversa existe), éstos resultan ser sensibles a las perturbaciones, también consideradas como errores de los datos:

Teorema 2.1.2 (Inversa de un operador compacto). Sea V un espacio vectorial normado, A un operador compacto tal que $A : V \rightarrow V$. Si A^{-1} existe, entonces, A^{-1} es continuo si y sólo si V es de dimensión finita.

Demostración. La demostración se encuentra en [26].

Con estos resultados, se tiene que la ecuación integral de Fredholm es un problema mal propuesto para el caso en el que V es de dimensión infinita. Para poder trabajar estos operadores de dimensión infinita estos deben ser aproximados por operadores de dimensión finita. Estos operadores de dimensión finita se construyen discretizando la ecuación integral en un conjunto de puntos del dominio de integración y dado que el operador integral es lineal, esta discretización se ve reflejada en una matriz cuadrada $A_d \in \mathcal{M}_{n \times n}(\mathbb{R})$. Aunque los operadores en dimensión finita siempre son continuos, las consecuencias del teorema 2.1.2 se ven reflejadas sobre la matriz A_d en su número de condición $cond(A_d) = \|A_d\| \|A_d^{-1}\|$, con $\|\cdot\|$ una norma matricial sub-multiplicativa. Intuitivamente $cond(A_d)$ cuantifica la sensibilidad a los errores de la solución x teniendo errores en el vector y y la matriz A_d del sistema de ecuaciones

$$A_d x = y, \text{ con } y \in \mathbb{R}^n \text{ dado.} \quad (2-4)$$

Esto gracias al resultado

Teorema 2.1.3 (Número de condición y sensibilidad a los errores). Sean $A_d \in \mathcal{M}_{n \times n}(\mathbb{R})$ con inversa A_d^{-1} ,

$$\hat{A}_d = A_d + \Delta A_d \quad (2-5)$$

una aproximación de la matriz A_d tal que $\|A_d^{-1}\| \|A_d^{-1} - A_d\| \leq 1$, y $\hat{y} = y + \Delta y$ soluciones a las ecuaciones

$$\hat{A}_d \hat{x} = \hat{y}, \quad A_d x = y, \quad (2-6)$$

de forma que $\hat{x} = x + \Delta x$ y $\|\cdot\|$ una norma matricial submultiplicativa. Entonces,

$$\frac{\|\Delta x\|}{\|x\|} \leq \frac{\text{cond}(A_d)}{1 - \text{cond}(A_d) \frac{\|\Delta A_d\|}{\|A_d\|}} \left(\frac{\|\Delta y\|}{\|y\|} + \frac{\|\Delta A_d\|}{\|A_d\|} \right). \quad (2-7)$$

Demostración. La demostración se encuentra en [31].

El número de condición de una matriz satisface $\text{cond}(A_d) \geq 1$. Cuando $\text{cond}(A_d) \approx 1$ significa que la solución del sistema (2-4) es poco sensible a los errores de los datos iniciales. En este sentido suele decirse que aquellas matrices con este número de condición son aproximadamente la matriz identidad. Por otra parte, el hecho que $\text{cond}(A_d) \gg 1$ implica que la solución al sistema de ecuaciones (2-4) resulta ser muy sensible a los errores, pues los errores relativos se propagan proporcionalmente al número de condición de la matriz A_d . De forma análoga suele interpretarse que estas matrices están “cerca” de ser matrices singulares.

Por lo tanto, para los problemas en los que se requiere encontrar x en $Ax = b$ con b dado y A un operador compacto se requieren acercamientos desde otra perspectiva, particularmente se proponen métodos de optimización, para poder encontrar una solución “razonable”, por ejemplo en [34] se propone emplear el método de regularización de Tikhonov. Este método consiste en minimizar el funcional

$$f(x) = \|Ax - b\|_{L_2}^2 + \alpha \|x\|_{L_2}^2. \quad (2-8)$$

Con este funcional se busca lidiar con la no continuidad del operador (en caso de existir) A^{-1} , evitando soluciones de normas muy grandes sin sacrificar el hecho de satisfacer la ecuación $Ax = b$. Esto implica que hay un grado de compromiso entre el error de aproximación del operador que se está dispuesto a asumir (primer término) y la magnitud de la solución a este problema (segundo término). La constante α mide este grado de compromiso entre uno u otro error. Por lo general se suele hacer la aproximación en la norma $\|\cdot\|_{L_2}$ puesto que esta norma proviene de un producto interior. Este hecho permite definir y explotar el concepto de ortogonalidad para resolver el problema de minimización (2-8).

Por otra parte, en este proyecto se abordan estos problemas usando la norma L_1 . Por lo cual trataremos con problemas que minimizan

$$f(x) = \|Ax - b\|_{L_1} \quad (2-9)$$

añadiendo restricciones en las que las variables de la solución son no negativas.

Esta norma no es tan usada como la norma euclidiana, pues, su falta de suavidad en los puntos óptimos hace que varias herramientas basadas en la derivada no puedan usarse en este caso. Sin embargo, esta norma posee ventajas con respecto a la norma euclidiana que son usadas a lo largo del trabajo, al optimizar funciones cuyas curvas de nivel son rectas, planos o hiperplanos.

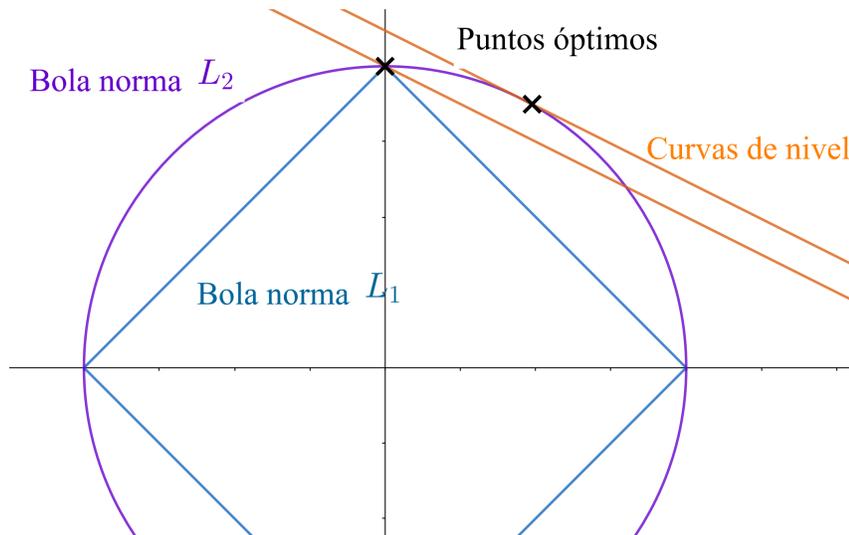


Figura 2-1: Comparación entre las normas L_1 y L_2 optimizando funciones lineales.

Una de las ventajas de la norma L_1 es encontrar los óptimos sobre los ejes, esto implica que las soluciones tienen bastantes ceros, como puede apreciarse en la Figura 2-1. Luego, es posible describir el objeto que se desea recuperar con pocas componentes. Por otra parte la norma euclídea tiende a darle más importancia a las grandes componentes (aquellas cuyo valor absoluto es más grande que uno) que a las pequeñas componentes (aquellas cuyo valor absoluto es menor uno) debido a que se consideran sumas de elementos elevados al cuadrado, a diferencia de la norma L_1 que tiende a considerar a todos los errores por igual, debido a que son sumas de valores absolutos.

Para abordar de esta forma el problema en este trabajo necesitamos algunas definiciones y resultados sobre funciones convexas, mostrados a continuación.

2.2. Conjuntos, funciones convexas y problemas de optimización

En el transcurso del trabajo estamos interesados en los problemas de optimización donde la función objetivo, las restricciones son funciones convexas y el dominio del problema es convexo.

2.2.1. Conjuntos y funciones convexas

Definición 2.2.1 (Función convexa). *Una función escalar $f : \mathbb{R}^n \rightarrow \mathbb{R}$ es convexa si para cualesquiera $x, y \in \text{dom}(f)$ y $\theta \in [0, 1]$ se cumple que*

$$f(\theta x + (1 - \theta)y) \leq \theta f(x) + (1 - \theta)f(y).$$

Esta definición asume que el dominio de la función f es convexo, es decir, el segmento que une a cualesquiera dos puntos en el conjunto está contenido en el conjunto. Esto para garantizar que $\theta x + (1 - \theta)y \in \text{dom}(f)$ con $0 \leq \theta \leq 1$.

Ejemplos de estas funciones son: la función norma $\|x\|_{L_1} = \|x\|_1 = \sum_{i=1}^n |x_i|$, para $x = (x_1, \dots, x_n)^\top \in \mathbb{R}^n$, pues, si tomamos $x, y \in \mathbb{R}^n$ y $\theta \in [0, 1]$, entonces,

$$\begin{aligned} \|\theta x + (1 - \theta)y\|_1 &= \sum_{i=1}^n |\theta x_i + (1 - \theta)y_i| \\ &\leq \sum_{i=1}^n |\theta x_i| + |(1 - \theta)y_i| \\ &= \sum_{i=1}^n \theta |x_i| + (1 - \theta)|y_i| \\ &= \sum_{i=1}^n \theta |x_i| + \sum_{i=1}^n (1 - \theta)|y_i| \\ &= \theta \|x\|_1 + (1 - \theta)\|y\|_1. \end{aligned} \tag{2-10}$$

Y las transformaciones lineales.

Para definir las restricciones y el dominio del problema se hacen las siguientes definiciones:

Definición 2.2.2 (Orden). *Dados dos vectores $x = (x_1, \dots, x_n)^\top$, $y = (y_1, \dots, y_n)^\top \in \mathbb{R}^n$ se dice que*

$$x \leq y \quad \text{si y sólo si} \quad x_i \leq y_i \quad \text{para cada } i=1, 2, \dots, n. \tag{2-11}$$

Esto define una relación de orden parcial sobre \mathbb{R}^n .

Definición 2.2.3 (Semiespacios y rectas). *Dados dos vectores $b \in \mathbb{R}^n$, $c \in \mathbb{R}^n$ y dos números reales d , e los conjuntos*

$$S(b, d) = \{x \in \mathbb{R}^n : b^\top x \leq d\}, \quad R(c, e) = \{x \in \mathbb{R}^n : c^\top x = e\} \quad (2-12)$$

Se llaman el semiespacio definido por b y d y la recta definida por c y e respectivamente.

Estos conjuntos tienen la propiedad de ser convexos.

Definición 2.2.4 (Poliedro). *Dadas dos matrices $D \in \mathcal{M}_{m \times n}(\mathbb{R})$, $E \in \mathcal{M}_{k \times n}(\mathbb{R})$ y dos vectores $d \in \mathbb{R}^m$, $e \in \mathbb{R}^k$ el conjunto*

$$X = \{x \in \mathbb{R}^n : Dx = d, Ex \leq e\}. \quad (2-13)$$

se llama poliedro.

Esta clase de conjuntos se pueden escribir como la intersección de semiespacios y rectas, por lo que estos conjuntos también son convexos. Estos conjuntos son los dominios sobre los que se trabaja en los problemas de programación lineal.

Dentro de todos los posibles problemas de optimización con restricciones, son de gran interés los problemas donde la función a optimizar y las restricciones son lineales.

Definición 2.2.5 (Problema de programación lineal). *Un problema de programación lineal es un problema del tipo*

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & c^\top x \\ \text{Sujeto a:} \quad & Dx = d \\ & Ex \leq e \\ & x \geq 0 \end{aligned} \quad (2-14)$$

Con x , c , d , e , D y E como en la definición anterior, este problema se dice primal. El conjunto

$$\{x \in \mathbb{R}^n : Dx = d, Ex \leq e, x \geq 0\}$$

se llama conjunto factible y cada elemento del conjunto se llama solución factible.

Definición 2.2.6 (Incorporación de variables de holgura). *Sea*

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & c^\top x \\ \text{Sujeto a:} \quad & Dx = d \\ & Ex \leq e \\ & x \geq 0 \end{aligned} \quad (2-15)$$

un problema de programación lineal. Se define $s = (s_1, \dots, s_k)^\top = e - Ex$ como el vector de variables de holgura. El problema anterior se puede reescribir como

$$\begin{aligned} \min_{(x,s) \in \mathbb{R}^{n+k}} \quad & c^\top x \\ \text{Sujeto a:} \quad & Dx = d \\ & Ex + s = e \\ & x \geq 0, \quad s \geq 0 \end{aligned} \tag{2-16}$$

y se dice que el problema ya ha incorporado sus variables de holgura.

Esto permite eliminar las desigualdades asociadas a la matriz E , solamente deja las desigualdades que garantizan que las variables son no negativas.

Definición 2.2.7 (Forma canónica). *Se dirá que un problema de programación lineal está en la forma canónica (también llamada clásica o estándar) si se puede formular como*

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & c^\top x \\ \text{Sujeto a:} \quad & Ax = b \\ & x \geq 0 \end{aligned} \tag{2-17}$$

Donde $c \in \mathbb{R}^n$, $b \in \mathbb{R}^k$ y $A \in \mathcal{M}_{k \times n}(\mathbb{R})$.

Normalmente los algoritmos que solucionan problemas de programación lineal parten desde esta formulación. Con la incorporación de las variables de holgura cualquier problema de programación lineal puede ser llevado a su forma canónica, la cual es equivalente y de dimension más grande, donde la matriz A se obtiene poniendo en bloques las matrices que tiene el problema de programación lineal originalmente.

Por último se presenta el problema dual.

Definición 2.2.8 (Problema dual). *El problema dual asociado al problema primal (2-17) es*

$$\begin{aligned} \max_{y \in \mathbb{R}^k} \quad & b^\top y \\ \text{Sujeto a:} \quad & A^\top y \leq c \\ & y \in \mathbb{R}^k \end{aligned} \tag{2-18}$$

En este caso también deben agregarse variables de holgura al problema dual. Es bastante útil trabajar con los problemas primal y dual en forma paralela. Esto a causa de los siguientes resultados:

Teorema 2.2.1 (Dualidad débil). *Considérese los problemas (2-17) y (2-18) en forma canónica, con s el vector de variables de holgura en el problema (2-18), entonces,*

$$c^\top x - b^\top y = x^\top s \geq 0 \tag{2-19}$$

Demostración. *Se tiene que*

$$b^\top = x^\top A^\top \quad y \quad s = c - A^\top y, \quad (2-20)$$

entonces,

$$c^\top x - b^\top y = c^\top x - x^\top A^\top y = x^\top c - x^\top A^\top y = x^\top (c - A^\top y) = x^\top s \quad (2-21)$$

□

Esto significa que la solución del problema primal (minimización) siempre es mayor o igual que la solución de un problema dual asociado.

Teorema 2.2.2 (Condiciones de de optimalidad Karush-Kuhn-Tucker (KKT)). *Considere los problemas (2-17) y (2-18) en su forma canónica, con s el vector de variables de holgura en el problema (2-18). Entonces, los vectores x^* y (y^*, s^*) son soluciones óptimas de sus problemas respectivos si y sólo si*

$$\blacksquare Ax^* = b \quad (2-22)$$

$$\blacksquare A^\top y^* + s^* = c \quad (2-23)$$

$$\blacksquare s^* \geq 0, \quad x^* \geq 0 \quad (2-24)$$

$$\blacksquare x_i^* s_i^* = 0 \text{ para cada } i = 1, 2, \dots, n. \quad (2-25)$$

Demostración. *La demostración se encuentra en [2] y en [4].*

Las ecuaciones (2-22), (2-23) y (2-24) especifican que los vectores deben ser soluciones factibles, las condiciones expresadas en la familia de ecuaciones (2-25) son llamadas *condiciones de holgura complementaria*, estas al usarse en conjunto con el teorema de dualidad débil generan el *teorema de fundamental de dualidad* o *dualidad fuerte*.

Teorema 2.2.3 (Teorema fundamental de dualidad). *Considere los problemas (2-17) y (2-18) en forma canónica, con s el vector de variables de holgura en el problema (2-18), entonces, una y sólo una de las siguientes sucede:*

- *Ambos tienen soluciones óptimas x^* y (y^*, s^*) . Las condiciones de optimalidad implican que $x^{*\top} s^* = 0$, añadiendo el teorema de dualidad débil se tiene que $c^\top x^* = b^\top y^*$. Esto último es a lo que se le llama dualidad fuerte.*
- *La función objetivo de uno de los problemas no está acotada en el conjunto factible y en el otro problema el conjunto factible es vacío.*
- *Ambos problemas tienen sus conjuntos factibles vacíos.*

Demostración. *La demostración se encuentra en [2].*

Teniendo en cuenta estos resultados se puede apreciar la importancia del problema dual al ser un complemento del problema primal. Este problema es usado en los algoritmos de programación lineal en conjunto con el problema primal ya que las variables de holgura del problema dual pueden verse como los multiplicadores de Lagrange en el problema primal gracias a la condición (2-25). También, estos algoritmos hacen uso de el teorema de dualidad fuerte en el cálculo de los pasos siguientes en la ejecución de estos algoritmos.

Otra ventaja es que el problema dual puede ser interpretado como el mismo problema planteado desde otro punto de vista. Un ejemplo de esto es: una persona que desea comprar alimentos que satisfagan cierta dieta nutricional, minimizando el costo que paga por ellos, entonces, este problema es el primal. Al mismo tiempo tenemos una compañía que fabrica suplementos vitaminados (éstos pueden reemplazar la comida si es necesario), la compañía desea maximizar su ingreso al poder fijar los precios a voluntad de forma que sea capaz de suplir la necesidad de la persona y de hacer que ésta le compre sus suplementos en vez de adquirir alimento. El teorema de dualidad muestra que en los puntos óptimos son aquellos donde lo que la persona gasta en la dieta es *precisamente* lo que la compañía percibe como ingreso, en cualquier otro par de puntos factibles para los problemas primal y dual la persona estará pagando más de lo que la compañía percibe como ingreso, siendo no óptimo para ambos.

2.2.2. Algoritmos de optimización

Una vez se tienen condiciones con las cuales se garantiza la existencia de óptimos, se puede observar que la función alcanza el óptimo en la frontera. Esto ocurre ya que si la función es no constante, esto es, si $c \neq 0$ el gradiente no es nulo en el interior del conjunto factible. Ya que si $f(x) = c^\top x$, entonces, $\nabla f = c^\top \neq 0$. Esto implica que no hay puntos críticos en el interior del conjunto, obligando a los óptimos a estar sobre la frontera, más específicamente sobre los lados del poliedro que conforma el conjunto factible. También, teniendo en cuenta que las curvas de nivel de la función objetivo son hiper-planos, los puntos óptimos se pueden escoger de forma que estén sobre los vértices del poliedro, como se ilustra en la Figura 2-2.

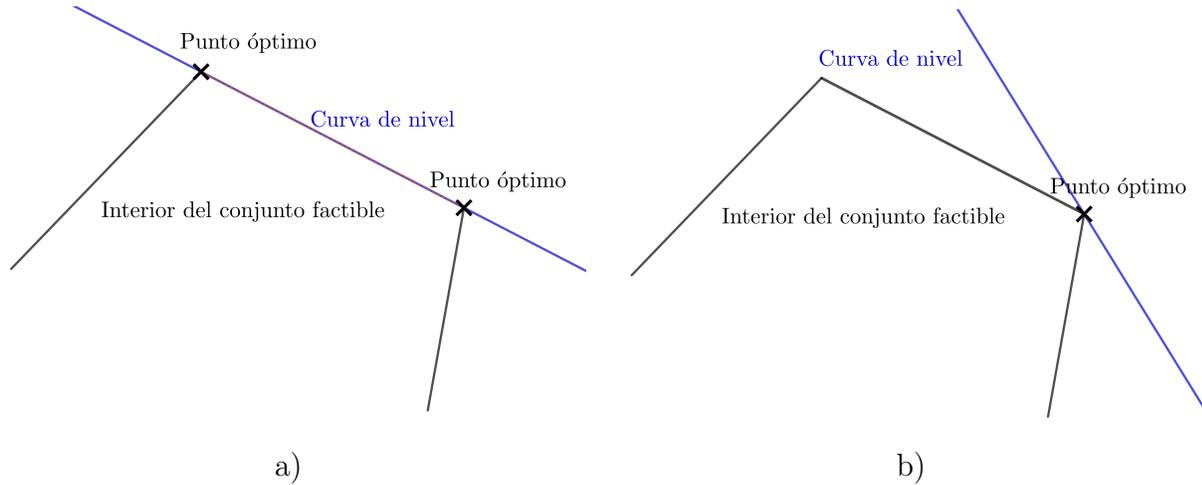


Figura 2-2: Ilustración de los vértices óptimos. a) Curva de nivel paralela, b) Curva de nivel no paralela

Con este último resultado en mente el método simplex propuesto por Dantzig en 1947 busca encontrar el óptimo analizando vértices adyacentes del poliedro que forma el conjunto factible. Sin embargo, primero se debe dar una definición más precisa:

Definición 2.2.9 (Solución básica). *Considere el problema dado por la ecuación (2-17). Suponga que la matriz $A \in \mathcal{M}_{k \times n}(\mathbb{R})$, que su rango es k y que además es el mismo que el de la matriz extendida $[A, b]$. Con esto, es posible re-ordenar las columnas de la matriz A de tal forma que $A=[B, N]$, donde $B \in \mathcal{M}_{k \times k}(\mathbb{R})$ es una matriz invertible y N es una matriz que contiene las columnas restantes de A . Entonces, una solución x de la ecuación $Ax = b$, donde*

$$x = \begin{bmatrix} x_B \\ x_N \end{bmatrix}, \quad x_B = B^{-1}b \text{ y } x_N = 0$$

es llamada solución básica. x_B es llamado el vector de variables básicas y x_N es llamado el vector de variables no básicas.

A primera vista las soluciones básicas no parecen ser relevantes dentro del contexto del problema y su interpretación geométrica. No obstante, el siguiente resultado muestra que son, de hecho, los vértices del poliedro. Así se garantiza que estos dos conceptos son equivalentes.

Teorema 2.2.4 (Soluciones básicas y vértices). *Si el conjunto factible del problema (2-17) es no vacío, entonces, existen soluciones básicas, además, el conjunto de soluciones básicas es el mismo que el conjunto de vértices del conjunto factible.*

Demostración. *La demostración se encuentra en [2].*

La existencia de puntos óptimos garantiza la existencia de soluciones básicas óptimas, complementando la Figura 2-2.

Teorema 2.2.5 (Existencia de soluciones óptimas básicas). *Si una solución óptima para el problema (2-17) existe, entonces, existe una solución óptima que es una solución básica.*

Demostración. *La demostración se encuentra en [2].*

Algoritmo Simplex

El algoritmo simplex consiste en ir buscando a través de soluciones básicas soluciones óptimas básicas las cuales son óptimas en el problema original. Esto se hace construyendo una tabla, donde se van encontrando soluciones básicas. Dada una solución básica, se construye la siguiente tabla cambiando una variable básica por una no básica, continuando este proceso se puede llegar a una solución básica óptima.

	x_1	x_2	\cdots	x_{n-1}	x_n	$c^\top x$
z	$-c_1$	$-c_2$	\cdots	$-c_{n-1}$	$-c_n$	$c_B^\top b$
x_{i_1}	$A_{1,1}$	$A_{1,2}$	\cdots	$A_{1,n-1}$	$A_{1,n}$	b_1
x_{i_2}	$A_{2,1}$	$A_{2,2}$	\cdots	$A_{2,n-1}$	$A_{2,n}$	b_2
\vdots	\vdots	\vdots	\ddots	\vdots	\vdots	\vdots
$x_{i_{k-1}}$	$A_{k-1,1}$	$A_{k-1,2}$	\cdots	$A_{k-1,n-1}$	$A_{k-1,n}$	b_{k-1}
x_{i_k}	$A_{k,1}$	$A_{k,2}$	\cdots	$A_{k,n-1}$	$A_{k,n}$	b_k

Figura 2-3: Construcción de la tabla asociada al método simplex.

Donde:

- z indica la fila asociada a los elementos del vector de costo c
- c_B^\top es el vector de costo c , con la excepción que $c_i = 0$ para aquellos i para los cuales x_i es una variable no básica
- x_{i_j} con $j = 1, 2, \dots, k$ es una enumeración de las variables básicas
- $A_{i,j}$ corresponde al elemento que está en la fila i y en la columna j de la matriz A .

Para escoger la variable que entra, esto es, aquella variable que dejará de ser variable no básica para ser una variable básica se escoge aquella variable x_j de de la primera fila para la cual el coeficiente asociado en la segunda fila satisface es positivo y el mayor de los elementos de la segunda fila, asociada a z .

$$-c_j = \max_{\substack{i=1,\dots,n \\ -c_i > 0}} \{-c_i\}.$$

	x_1	\cdots	x_j	\cdots	x_n	$c^\top x$
z	$-c_1$	\cdots	$-c_j$	\cdots	$-c_n$	$c_B^\top b$
x_{i_1}	$A_{1,1}$	\cdots	$A_{1,j}$	\cdots	$A_{1,n}$	b_1
x_{i_2}	$A_{2,1}$	\cdots	$A_{2,j}$	\cdots	$A_{2,n}$	b_2
\vdots	\vdots	\ddots	\vdots	\ddots	\vdots	\vdots
$x_{i_{k-1}}$	$A_{k-1,1}$	\cdots	$A_{k-1,j}$	\cdots	$A_{k-1,n}$	b_{k-1}
x_{i_k}	$A_{k,1}$	\cdots	$A_{k,j}$	\cdots	$A_{k,n}$	b_k

Figura 2-4: Selección de la variable básica entrante.

Una vez escogida la variable que entra se toman los cocientes de la columna asociada a la variable y la última columna, de forma que los coeficientes de la última columna están en el denominador. Una vez efectuados estos cocientes se escoge la fila en la que esté la variable x_i que satisfaga:

$$\frac{b_i}{A_{i,j}} = \min_{\substack{l=0,\dots,k \\ A_{l,j}>0}} \left\{ \frac{b_l}{A_{l,j}} \right\}.$$

	x_1	\cdots	x_j	\cdots	x_n	$c^\top x$
z	$-c_1$	\cdots	$-c_i$	\cdots	$-c_n$	$c_B^\top b$
x_{i_1}	$A_{1,1}$	\cdots	$A_{1,j}$	\cdots	$A_{1,n}$	b_1
\vdots	\vdots	\ddots	\vdots	\ddots	\vdots	\vdots
x_{i_l}	$A_{l,1}$	\cdots	$A_{l,j}$	\cdots	$A_{l,n}$	b_l
\vdots	\vdots	\ddots	\vdots	\ddots	\vdots	\vdots
x_{i_k}	$A_{k,1}$	\cdots	$A_{k,j}$	\cdots	$A_{k,n}$	b_k

Figura 2-5: Selección de la variable básica saliente. Aquí $x_{i_l} = x_i$.

Una vez se tienen seleccionadas la variable que sale y la variable que entra se procede a hacer eliminación de Gauss-Jordan sobre la columna seleccionada, dejando un uno en la fila seleccionada y lo demás en ceros. Además cambiar la variable básica que sale por la variable no básica que entra en la primera columna.

	x_1	\dots	x_j	\dots	x_n	$c^\top x$
z	\bar{c}_1	\dots	0	\dots	\bar{c}_n	$c_B^\top \bar{b}$
x_{i_1}	$\bar{A}_{1,1}$	\dots	0	\dots	$\bar{A}_{1,n}$	\bar{b}_1
\vdots	\vdots	\dots	\vdots	\ddots	\vdots	\vdots
x_j	$\bar{A}_{l,1}$	\dots	1	\dots	$\bar{A}_{l,n}$	\bar{b}_l
\vdots	\vdots	\ddots	\vdots	\ddots	\vdots	\vdots
x_{i_k}	$\bar{A}_{k,1}$	\dots	0	\dots	$\bar{A}_{l,n}$	\bar{b}_k

Figura 2-6: Eliminación de Gauss-Jordan aplicada a la columna escogida. La barra sobre los elementos indica que posiblemente han cambiado sus valores debido a la eliminación.

Una vez realizado este proceso se regresa al paso donde se escogen las variables que entran y salen (Figura 2-4). Este proceso continúa hasta que no se encuentren elementos de la segunda fila que no sean positivos para las variables no básicas. De esta forma se garantiza que la solución básica que se tiene en el momento de parar sea solución básica óptima. En [2] se muestra cómo este proceso está orientado a satisfacer las condiciones de optimalidad KKT a medida que se llevan a cabo las iteraciones necesarias.

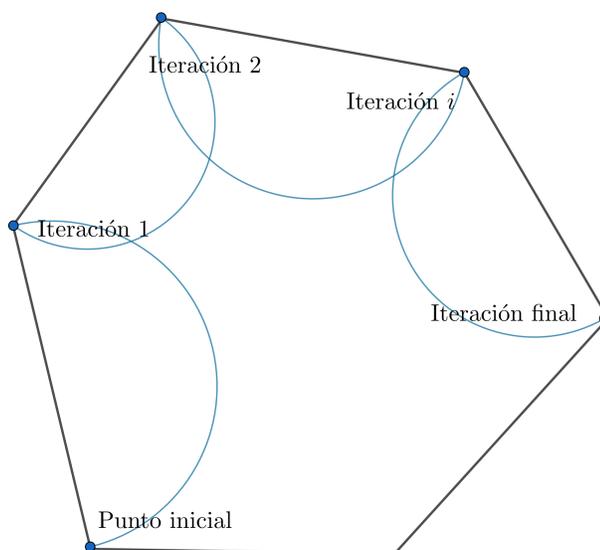


Figura 2-7: Ilustración del algoritmo simplex sobre el conjunto factible.

De hecho, cuando se encuentra el punto óptimo esta tabla presenta información importante: principalmente en la segunda fila (aquella asociada a los coeficientes \bar{c}_i , $i = 1, \dots, n$) cuyos

valores al final de las iteraciones corresponden a los multiplicadores de Lagrange del problema de optimización (2-17) en el punto óptimo. Con esto podemos observar que variables son más propensas a cambiar si se permite un cambio en las restricciones. También, en la tabla se puede apreciar el valor de la función objetivo, esto es, el valor asociado a $c_B^\top b$. Además la solución óptima está compuesta por variables no básicas cuyos valores son cero y una parte compuesta por variables básicas cuyos valores se encuentran en la última columna de la tabla. Con estas observaciones se puede ver la cantidad de información que provee la tabla del método simplex acerca del problema de optimización.

Dado que el método simplex consiste en recorrer los vértices del poliedro asociado al conjunto factible, si se cuentan la cantidad de vértices que se tienen, se puede observar que hay que seleccionar k de las n variables en el vector para que sean no nulas. Esto lleva a que la cantidad sea

$$\binom{n}{k} > \left(\frac{n}{k}\right)^k$$

vértices, por lo que es razonable pensar que este algoritmo tiene comportamiento exponencial. De hecho, en 1971 Victor Klee y George Minty [24] presentaron un problema en el que se tienen que recorrer todos los vértices del poliedro y es de complejidad exponencial. Sin embargo, estos casos parecen ser atípicos, se ha registrado [2] y [4] que la cantidad de iteraciones usualmente está comprendida entre $2k$ y $6k$. En cada iteración se requieren $\mathcal{O}(kn)$ operaciones, dando como resultado que el caso promedio se comporta de forma $\mathcal{O}(k^2n)$, a diferencia del caso exponencial, tiene un comportamiento polinomial.

Métodos de punto interior

El método simplex ha sido muy popular debido a que tiene un bajo costo computacional¹ en comparación de lo que se esperaba. Sin embargo, problemas como soluciones degeneradas y soluciones cíclicas² hacen que este método tenga limitaciones una vez que se abordan problemas de grandes dimensiones.

Teniendo esto en cuenta se han propuesto técnicas para hallar un óptimo que no consideren recorrer los vértices del conjunto factible. Para ello se han elaborado algoritmos que buscan hallar puntos óptimos para los problemas (2-17) y (2-18), estos generalmente están orientados en hallar soluciones que satisfagan las condiciones de optimalidad KKT.

A continuación se presenta un algoritmo de punto interior basado en el método de Newton para solucionar ecuaciones el tipo $f(x) = 0$ con $x, 0 \in \mathbb{R}^n$, presentado en el texto de [12].

¹En este texto se entenderá como costo computacional la cantidad de operaciones de punto flotante que se necesitan para llevar a cabo un algoritmo.

²Son aquellas soluciones básicas que se repiten durante el proceso, dando lugar a ciclos infinitos en los que no se converge a una solución óptima.

Este algoritmo está basado en solucionar varios de los siguientes problemas de una familia de problemas indexada por una constante $\theta > 0$. En estos problemas se introduce una barrera logarítmica para incluir implícitamente las restricciones de no negatividad y forzar que las variables sean positivas. Esto hace que las aproximaciones sucesivas a la solución estén en el interior del conjunto factible y no sobre su frontera, como se muestra en la ecuación (2-26) y en la Figura 2-8:

$$\begin{aligned}
 P(\theta) : \min_{y \in \mathbb{R}^n} \quad & c^\top y - \theta \sum_{i=1}^n \ln(y_i) \\
 \text{Sujeto a:} \quad & Ay = b \\
 & y > 0
 \end{aligned}
 \tag{2-26}$$

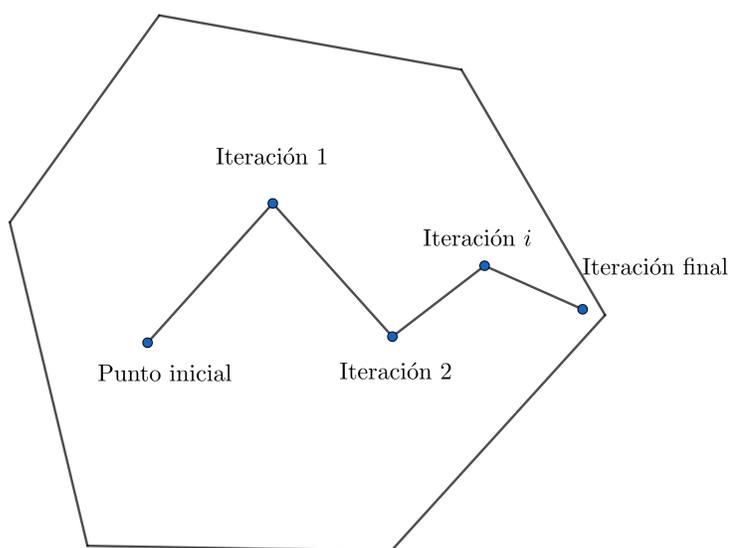


Figura 2-8: Ilustración del algoritmo de punto interior sobre el conjunto factible.

Para presentar el algoritmo se hace uso la siguiente notación:

- Se consideran los problemas primal y dual en las formas (2-17) y (2-18) respectivamente en su forma canónica, donde s denota el vector de variables de holgura en el problema dual.
- Si $y = (y_1, \dots, y_n)^\top$ entonces Y se define como la matriz diagonal cuya diagonal está compuesta por los elementos del vector y , es decir:

$$Y = \begin{pmatrix} y_1 & 0 & \cdots & 0 \\ 0 & y_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & \cdots & y_n \end{pmatrix}. \quad (2-27)$$

Además se define y^k como el vector y en el paso k , Δy como el paso a seguir en la siguiente iteración, esto es, $y^{k+1} = y^k + \Delta y$, θ^k como el parámetro de relajación en el paso k (realmente esto indica cuál problema de la familia indexada por θ se está solucionando) y $\|\cdot\|$ una norma definida en \mathbb{R}^n , usualmente $\|\cdot\| = \|\cdot\|_2$.

Algoritmo 2.2.1 (Algoritmo del método de punto interior). *Dados $(y^0, z^0, s^0) \in \mathbb{R}^n$ con $y > 0, y > 0, \theta^0 > 0, r$ tal que $0 < r < 1$, una tolerancia $\epsilon > 0$ y $k = 0$. Haga:*

1. *Verificar el criterio de parada:*

$$\begin{aligned} \|Ay^k - b\| &\leq \epsilon \\ \|A^\top z^k + s^k - c\| &\leq \epsilon \\ s^k \cdot y^k &\leq \epsilon \end{aligned} \quad (2-28)$$

Si se cumplen, pare, si no:

2. *Haga $\theta^k = \frac{(s^k)^\top y^k}{10n}$, donde n es la longitud de los vectores s^k e y^k .*

Definiendo θ de esta forma se busca que en cada paso se haga cada vez más pequeño, esto convierte la familia de problemas (2-26) al problema (2-17).

3. *Resuelva el sistema de ecuaciones al encontrar $\Delta y, \Delta s$ y Δz :*

$$\begin{aligned} A\Delta y &= b - Ay^k \\ A^\top \Delta z + \Delta s &= c - A^\top z^k - s^k \\ S_k \Delta y + Y_k \Delta s &= \theta \mathbf{1} + S_k Y_k \mathbf{1} \end{aligned} \quad (2-29)$$

donde $\mathbf{1}$ es el vector de unos de dimensión n . En este caso Y_k y S_k con las matrices diagonales asociadas a los vectores y^k y s^k respectivamente.

Este sistema aparece al considerar las condiciones KKT del problema (2-26). Se observa que el gradiente de la función objetivo en la ecuación (2-26) es $c - \theta Y^{-1} \mathbf{1}$. Si definimos $s = \theta Y^{-1} \mathbf{1}$ (notemos que $s = S \mathbf{1}$) y z el vector que como los multiplicadores para las

restricciones de igualdad, se pueden escribir las condiciones KKT como

$$\begin{aligned} Ay &= b \\ y &> 0 \\ A^\top z + s &= c \\ \frac{1}{\theta} Y S \mathbf{1} - \mathbf{1} &= 0. \end{aligned} \quad (2-30)$$

Así, y es factible en el problema (2-17), el par (z, s) es factible en el problema (2-18) y además

$$c^\top y - b^\top z = y^\top s = \mathbf{1}^\top Y S e \mathbf{1} = \theta \mathbf{1}^\top \mathbf{1} = n\theta. \quad (2-31)$$

El sistema (2-30) posee términos no lineales, lo que hace difícil resolverlo. En este caso se optará por linealizarlo. Suponga que en el paso k , y^k es factible (quizás con un cierto error) en el problema (2-17) y el par (z^k, s^k) es factible (quizás con un cierto error) en el problema (2-18), esto es,

$$\begin{aligned} Ay^k &\approx b, \quad y^k > 0 \\ A^\top z^k + s^k &\approx c, \quad s^k > 0, \end{aligned} \quad (2-32)$$

queremos que $y^{k+1} = y^k + \Delta y$ sea **factible** en el problema (2-17) y satisfaga las condiciones KKT (2-30), además que el par (z^{k+1}, s^{k+1}) con $z^{k+1} = z^k + \Delta z$ y $s^{k+1} = s^k + \Delta s$ sea **factible** en el problema (2-18). Al reemplazar y^{k+1} , z^{k+1} y s^{k+1} en (2-30) se obtiene

$$\begin{aligned} A\Delta y &= b - Ay^k \\ A^\top \Delta z + \Delta s &= c - A^\top z^k - s^k \\ S_k \Delta y + Y_k \Delta s &= \theta \mathbf{1} + S_k Y_k \mathbf{1} + (\Delta Y \Delta S)e \end{aligned} \quad (2-33)$$

Para aproximar el sistema (2-30) por uno lineal, se elimina el término $(\Delta Y \Delta S)e$ de la tercera ecuación de (2-33). Esto nos deja con el sistema de ecuaciones (2-29).

Como se observa, lo que se busca es aproximar iterativamente soluciones para las condiciones de optimalidad KKT. A su vez, al hacer $\theta \rightarrow 0$ se encuentra la solución al problema de programación lineal inicialmente propuesto. También, se puede apreciar (al igual que el método simplex) que las variables duales resultan ser los multiplicadores de Lagrange en el problema primal.

4. *Determinar coeficientes para los pasos (éstos impiden que los nuevos vectores tengan componentes no positivas):*

$$\rho_p = \min \left\{ 1, r \min_{\Delta y_i < 0} \left\{ \frac{y_i^k}{-\Delta y_i} \right\} \right\}, \quad \rho_d = \min \left\{ 1, r \min_{\Delta s_i < 0} \left\{ \frac{s_i^k}{-\Delta s_i} \right\} \right\}. \quad (2-34)$$

5. *Calcular el paso siguiente:*

$$(y^{k+1}, z^{k+1}, s^{k+1}) = (x^k, y^k, s^k) + (\rho_p \Delta y, \rho_d \Delta z, \rho_d \Delta s). \quad (2-35)$$

6. *Establecer $k + 1 \leftarrow k$ y regresar al paso 1.*

La eficiencia de este algoritmo está directamente relacionada con lo rápido que se es capaz de solucionar sistemas de ecuaciones (paso 4 del algoritmo) y esto depende de la estructura de la matriz $A \in \mathcal{M}_{n \times n}(\mathbb{R})$. Se ha observado que este algoritmo, en el caso promedio, requiere de 25 a 80 iteraciones [2], [4]. Así que el costo es proporcional a la cantidad de operaciones que se requieren para solucionar sistemas de ecuaciones asociados a la matriz AA^\top , que en el caso de métodos directos como la factorización de Cholesky y algunos indirectos como el método del gradiente conjugado tienen orden $\mathcal{O}(n^3)$. Luego, este método de punto interior tiene un costo computacional del mismo orden. También, es importante señalar que existen mejores aproximaciones, en [9] se muestra un algoritmo de orden $\mathcal{O}(n^\omega \ln(n))$ donde $\mathcal{O}(n^\omega)$ es el costo computacional de multiplicar dos matrices.

3 El modelo en la restauración de imágenes borrosas

Una imagen es considerada como una función escalar $X(t)$ donde $t \in G \subseteq \mathbb{R}^2$ y $X(t)$ es el valor de intensidad en ese punto¹. Dada la naturaleza que tienen las imágenes, esto es, su forma cuadrada, el dominio de la función puede ser considerado como un rectángulo, de modo que, $G = [0, n] \times [0, m]$ para ciertos n y m que determinan las dimensiones de la imagen a tratar.

3.1. Descripción del problema

Para tener una descripción matemática del difuminado de una imagen, se considera que $X(t)$ describe la imagen verdadera. Entonces la imagen observada $X(t)$ y la imagen borrosa y ruidosa $B(s)$ están relacionadas por

$$B(s) = \int_G K(s, t)X(t)dt + E(s), \quad (3-1)$$

donde $s \in \mathbb{R}^2$, $E(s)$ representa ruido² aditivo y G al ser un rectángulo cerrado resulta ser un conjunto compacto de \mathbb{R}^2 . La función $K(s, t)$ núcleo del operador integral especifica como los puntos de la imagen se difuminan y es llamada función puntual de propagación o PSF por sus siglas en inglés (point spread function). El problema de restauración de la imagen borrosa es: dados K y B encontrar una aproximación de X .

En problemas reales, las imágenes son capturadas solamente en puntos discretos dispuestos en un arreglo rectangular, cada uno de éstos es denominado pixel. Por lo tanto en la práctica se trabaja con la semidiscretización de (3-1)

$$B(s_i) = \int_G K(s_i, t)X(t)dt + E(s_i), \quad i = 1, \dots, N, \quad (3-2)$$

¹Usualmente se usa la escala que varía de 0 a 255, aquí 0 indica negro y 255 indica blanco. Los valores intermedios son diferentes tipos de grises.

²Es considerado ruido aquellas variaciones de los niveles de intensidad que no corresponden con lo que en realidad debe ser capturado. Suele modelarse de forma que tiene un comportamiento aleatorio.

donde N es el número de píxeles en la imagen observada. Sin embargo, la imagen que se pretende encontrar ($X(t)$) se asume de las mismas dimensiones y discretizada de la misma forma que la imagen borrosa. En particular, ambas poseen la misma cantidad de píxeles. Esto sugiere que para discretizar las integrales en (3-2) se usa la regla de cuadratura del punto medio en cada uno de los píxeles, como sigue

$$B(s_i) = \sum_{j=1}^N K(s_i, t_j) X(t_j) \Delta t_j + E(s_i), \quad i = 1, \dots, N. \quad (3-3)$$

Es posible considerar que la distancia tanto horizontal como vertical entre los píxeles es de 1, por lo que $\Delta t_j = 1$ para cada $j = 1, \dots, N$. Ahora bien, si consideramos la matriz A cuyas entradas están dadas por $A_{ij} = K(s_i, t_j)$, los vectores $x = (X(t_1), \dots, X(t_N))^T$, $b = (B(s_1), \dots, B(s_N))^T$ y $e = (E(s_1), \dots, E(s_N))^T$ la ecuación integral (3-3) se convierte en el sistema de ecuaciones

$$b = Ax + e. \quad (3-4)$$

El proceso de captura, degradación y difuminado de una imagen para la ecuación (3-4) suele ser modelado como una convolución [16], [7, pág 417]. Esto da una estructura particular a la función núcleo. Esto es, $K(s, t) = k(s-t)$, donde $k(x)$ es una función escalar cuyo dominio es el conjunto G . Así mismo veremos que la matriz A también tiene una estructura particular debido a este hecho.

3.2. Funciones de dispersión puntual (PSF)

En la descripción de las funciones de dispersión puntual se ha propuesto como modelo para generar imágenes borrosas el uso de ecuaciones diferenciales parciales [22] y [18]. Particularmente se propone usar la ecuación del calor en dos dimensiones con condiciones de Neumann sobre el conjunto G .

Entonces, en la ecuación del calor el objetivo es encontrar $u : G \times [0, T] \rightarrow \mathbb{R}$ tal que

$$\begin{aligned} \frac{\partial u}{\partial t} - \nabla \cdot (d(x) \nabla u) &= 0, \\ u(x, 0) &= I(x), \\ \frac{\partial u}{\partial \eta}(x) &= 0 \text{ para } x \in \partial G. \end{aligned} \quad (3-5)$$

Donde:

- $u(x, t)$ es la imagen degradándose en el transcurso del tiempo.
- $d(x)$ es el coeficiente de difusión, controla intensidad y dirección de la degradación.

- $I(x)$ es la imagen original, sin degradar.
- $\frac{\partial u}{\partial \eta}(x)$ es el flujo, aquí, indica que la suma de los valores de intensidad de la imagen no cambia. En este punto se asume que se están redistribuyendo las intensidades de los pixeles de la imagen.
- G es el rectángulo donde están los valores de intensidad de la imagen.

Algunos ejemplos de imágenes degradadas al usar la ecuación del calor con coeficientes constantes son:



Figura 3-1: Imágenes degradadas con la ecuación diferencial (3-5). (a) Imagen real, (b) Imagen borrosa. Tomado de [20].



Figura 3-2: Imágenes degradadas con la ecuación diferencial (3-5). (a) Imagen real, (b) Imagen borrosa. Tomado de [11].

Si el coeficiente $d(x)$ es constante y la condición inicial $I(x)$ es suficientemente suave, la solución de (3-5) tiene la forma (ver [22])

$$u(x, t) = \frac{1}{\sqrt{4\pi t}} \int_{\mathbb{R}^2} \exp\left(-\frac{\|\xi\|^2}{4t}\right) I(x - \xi) d\xi. \quad (3-6)$$

Esto resulta en una convolución con un núcleo exponencial, lo que justifica que en la ecuación (3-1) se convierta el operador integral en una convolución con un núcleo.

Las PSF $k(x)$ mostradas a continuación se construyen teniendo en cuenta tres propiedades:

- El soporte de la función está contenido en el conjunto G . De hecho, el tamaño del soporte suele ser bastante pequeño comparado al conjunto G .
- $k(x) \geq 0$ para todo $x \in G$.
- Tomando en cuenta la hipótesis en la que el efecto de degradación consiste en que el objeto que captura la imagen dispersa la luz recibida. Se tiene que

$$\int_{\mathbb{R}^2} k(x) dx = \int_G k(x) dx = 1. \quad (3-7)$$

3.2.1. PSF de desenfoque

Esta PSF es una generalización de aquella que aparece en la solución de (3-5) como núcleo en la ecuación (3-6), y tiene la forma

$$k(x, y) = \frac{1}{M} \exp\left(-\frac{1}{2} \begin{bmatrix} x - x_0 \\ y - y_0 \end{bmatrix}^\top \begin{bmatrix} \sigma_1^2 & \sigma_2 \\ \sigma_2 & \sigma_3^2 \end{bmatrix}^{-1} \begin{bmatrix} x - x_0 \\ y - y_0 \end{bmatrix}\right). \quad (3-8)$$

M es una constante de forma que se cumple la condición (3-7) y $(x_0, y_0)^\top$ es el centro de la PSF. La matriz conformada por σ_i , $i = 1, 2, 3$ controla la intensidad y dirección del difuminado a través de sus valores y vectores propios, en este caso exigiremos que esta matriz sea simétrica definida positiva.

Para el caso en el que $\sigma_2 = 0$ y $\sigma_1 = \sigma_3 = \sigma$ se llega al caso simétrico y a el cual se suele referir como una PSF exponencial con varianza σ^2 .

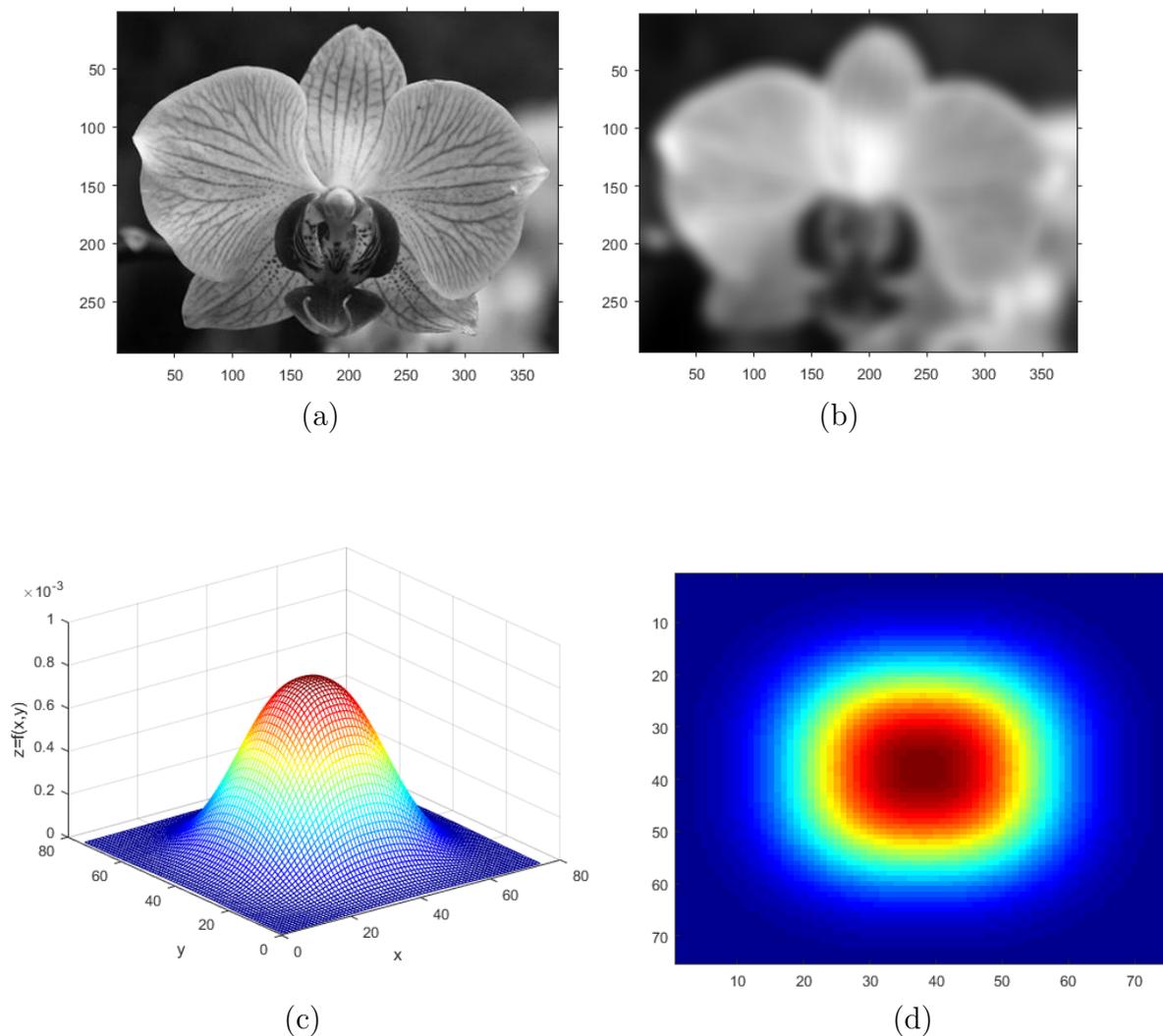


Figura 3-3: Ejemplo una PSF exponencial de desenfoque simétrica con varianza $\sigma = 10$ y de tamaño 75×75 . (a) Imagen real. Tomado de [19], (b) Imagen degradada con la PSF, (c) Función que representa la PSF, (d) Matriz de convolución que representa a la función.

En este punto proponemos extender el caso simétrico al siguiente modelo exponencial

$$k(x, y) = \frac{1}{M} \exp \left(-\frac{1}{2} \left(\frac{|x - x_0|}{\sigma} \right)^q + \left(\frac{|y - y_0|}{\sigma} \right)^q \right). \quad (3-9)$$

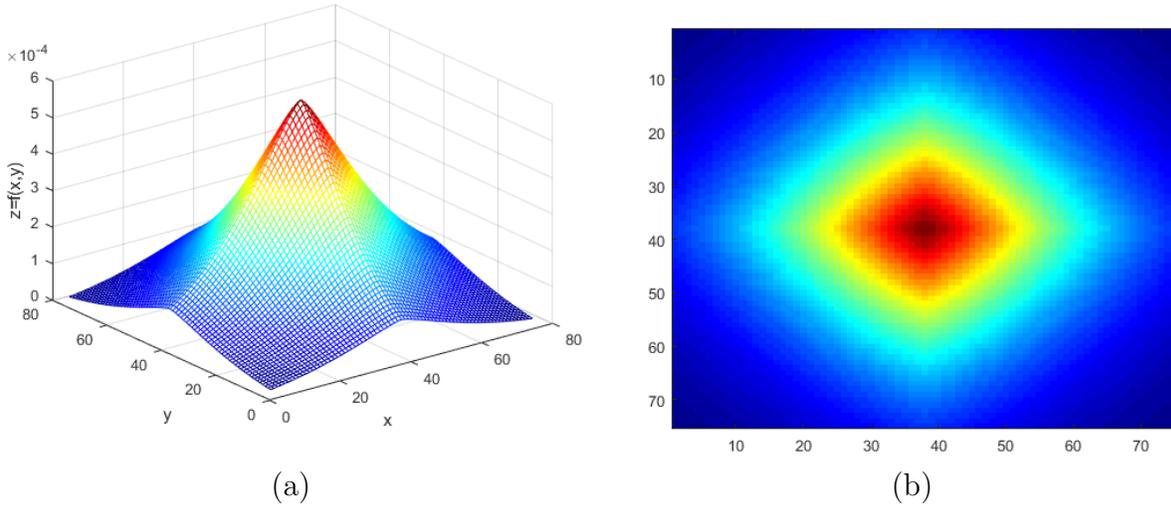


Figura 3-4: Ejemplo una PSF exponencial de desenfoque simétrica con un parámetro $q = 1,2$ y de tamaño 75×75 . (a) Función que representa la PSF, (b) Matriz de convolución que representa a la función.

Nótese que en el caso $q = 2$ se tiene el caso simétrico con varianza σ^2 . La ventaja de esta función exponencial simétrica extendida de varianza σ^2 y coeficiente q es que al discretizarla sobre su soporte la matriz de convolución resultante (véase la Figura 3-4) tiene rango uno. Eso hace que la matriz A en la ecuación (3-4) se pueda factorizar como

$$A = A_c \otimes A_r, \quad (3-10)$$

para ciertas matrices A_c y A_r construidas a partir de las columnas de la matriz de convolución de la PSF exponencial (3-9). El símbolo \otimes hace referencia al producto tensorial entre las matrices A_c y A_r .

3.2.2. PSF de movimiento

Para simular el efecto de borrado producido por un elemento que se mueve en la imagen se considera una función que toma valores a lo largo de una sola dirección la cual es aquella en la que se da el efecto de borrado.

La expresión para esta PSF está dada por

$$k(x, y) = \begin{cases} 1/M, & \text{si } (x, y) \in D \\ 0, & \text{si } (x, y) \notin D \end{cases} \quad (3-11)$$

Donde D es un conjunto que tiene forma de línea recta. Este conjunto comprende desde el centro de la PSF hasta la frontera del soporte de la función formando un ángulo $\theta \in [0, 2\pi)$

con la parte positiva del eje x . M , de nuevo, una constante para que se cumpla la condición de la ecuación (3-7). Se observa que en los casos $\theta = 0, \pi$ corresponden a movimientos horizontales y que los casos $\theta = \frac{\pi}{2}, \frac{3\pi}{2}$ corresponden a movimientos verticales.

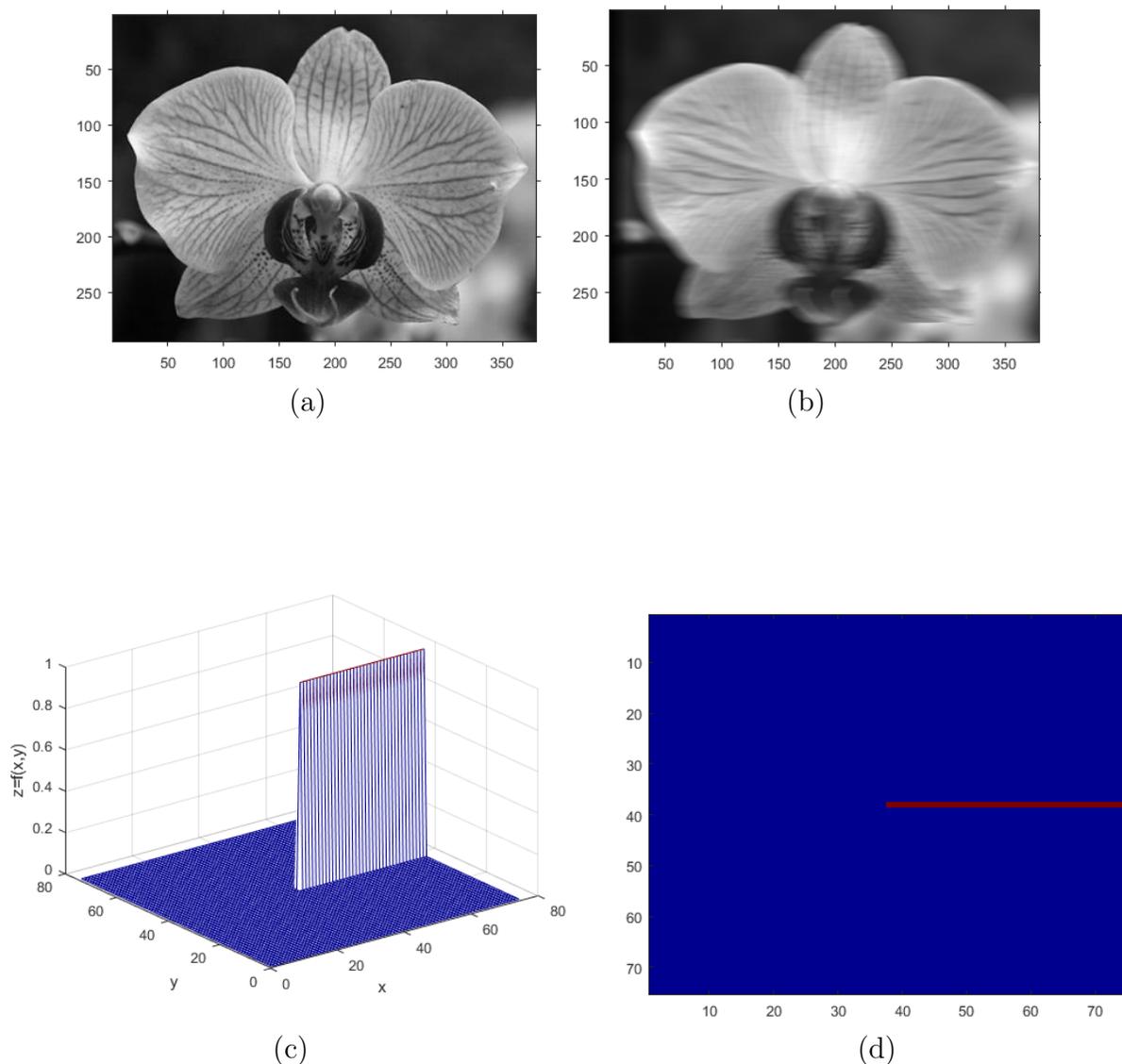


Figura 3-5: Ejemplo una PSF de movimiento horizontal $\theta = 0$ y de tamaño 75×75 . (a) Imagen real, (b) Imagen degradada con la PSF, (c) Función que representa la PSF, (d) Matriz de convolución que representa a la función.

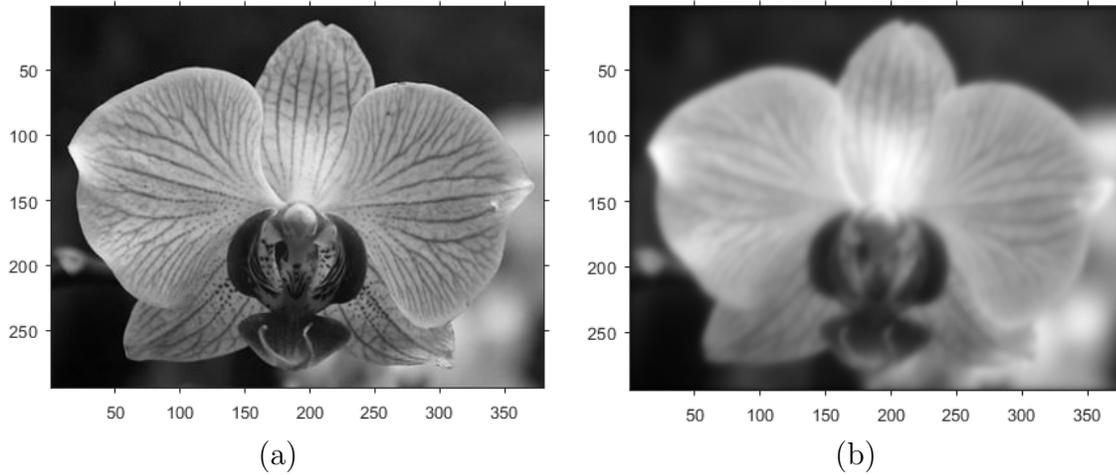
Cuando se trata de movimiento horizontal o vertical, la matriz de convolución también tiene rango uno por lo que la matriz A del modelo (3-4) se puede factorizar también como (3-10).

3.2.3. PSF degradado atmosférico

Esta PSF es usada para modelar turbulencias atmosféricas y turbulencias captadas por telescopios. También es llamada función Moffat [33]. Esta función a diferencia de las dos anteriores no posee la propiedad en la que su matriz de convolución tiene rango uno. Esta función se describe como

$$k(x, y) = \frac{1}{M} \left(1 + \begin{bmatrix} x - x_0 \\ y - y_0 \end{bmatrix}^\top \begin{bmatrix} \sigma_1^2 & \sigma_2 \\ \sigma_2 & \sigma_3^2 \end{bmatrix}^{-1} \begin{bmatrix} x - x_0 \\ y - y_0 \end{bmatrix} \right)^{-\beta} \quad (3-12)$$

Al igual que la PSF de desenfoque, esta función requiere una matriz conformada por σ_i con $i = 1, 2, 3$, añadiendo un parámetro $\beta > 0$ que controla qué tan rápido decae la función y M , de nuevo, una constante para que se cumpla la condición de la ecuación (3-7). Nótese la diferencia entre las funciones de las Figuras 3-3 y 3-6.



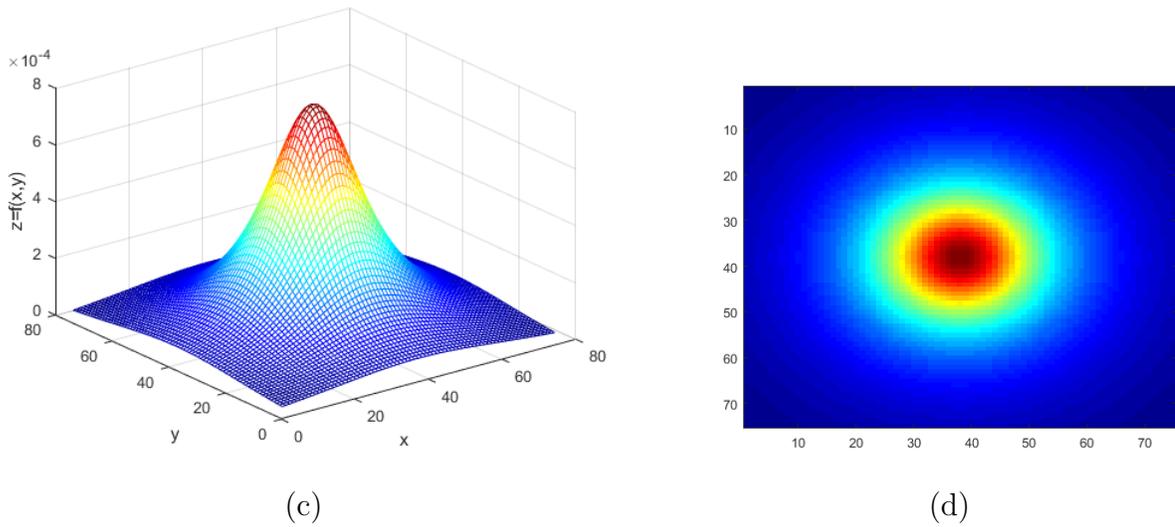


Figura 3-6: Ejemplo una PSF Moffat de varianza $\sigma = 8$, $\beta = 3$ y de tamaño 75×75 . (a) Imagen real, (b) Imagen degradada con la PSF, (c) Función que representa la PSF, (d) Matriz de convolución que representa a la función.

3.3. Limitaciones y dificultades

Las limitaciones del problema de restauración de imágenes borrosas vienen principalmente de dos fuentes: el tamaño de los sistemas de ecuaciones asociados al proceso de discretización y del mal condicionamiento de la matriz asociada al sistema (3-4).

3.3.1. Tamaño de los sistemas de ecuaciones

Como se dijo en las ecuaciones (3-2) y (3-3), el sistema (3-4) tiene los vectores $b, x, e \in \mathbb{R}^N$ donde $N = nm$ es la cantidad de píxeles de la imagen teniendo en cuenta la estructura del conjunto G ; la matriz $A \in \mathcal{M}_{N \times N}(\mathbb{R})$ tiene N^2 entradas.

Para ilustrar la capacidad computacional que se requiere, se usará de ejemplo una imagen X de tamaño 500×500 píxeles, es decir que la imagen tiene 250.000 píxeles, lo que implicaría que A es una matriz de tamaño 250.000×250.000 , de $6,25 \times 10^{10}$ elementos. Si las componentes la matriz se escriben en formato de precisión doble la matriz completa ocuparía aproximadamente 470 Gb de memoria. Esto hace que se requieran aproximaciones o factorizaciones de la matriz A para realizar cálculos eficientes usando la estructura de A o usando los factores que tenga A si la descomposición es posible, de forma que no sea necesario tener toda la matriz de forma explícita para realizar productos Ax donde x es un vector de la dimensión adecuada.

3.3.2. Sensibilidad a los errores

Como consecuencia de los teoremas 2.1.1 y 2.1.2 el operador que describe cómo la imagen se degrada no posee inversa continua. Esto implica que a pequeñas perturbaciones el resultado de invertir este operador puede generar grandes cambios en el resultado. Por lo que es de esperar que si hay presencia de ruido en las imágenes al calcular la aproximación derivada de la ecuación (3-4)

$$x^* = A^{-1}b \quad (3-13)$$

sea una solución muy diferente a la imagen real $x = A^{-1}b - A^{-1}e$, a pesar de que la matriz A y su inversa sean operadores en espacios vectoriales de dimensión finita y por lo tanto continuos.

Sin embargo, esta no continuidad del operador en el espacio vectorial

$$C(G) = \{f : f \text{ es continua en } G\} \quad (3-14)$$

se ve reflejado en el número de condición de la matriz A , el cual como se dijo antes, refleja que tanto son afectadas las soluciones a perturbaciones en los datos de entrada.

Teorema 3.3.1 (Norma del operador integral). Sean $\psi(x)$, $\phi(x)$, $K(x, t)$ y A con las hipótesis del teorema 2.1.1 y $G = [0, n] \times [0, m]$, entonces,

$$\|A\|_{L_\infty} = \max_{x \in G} \int_G |K(x, t)| dt. \quad (3-15)$$

Demostración. La demostración se encuentra en [25].

Para conectar lo que sucede en dimensión infinita con la discretización en dimensión finita se considera el operador A definido en (2-3)

$$(A\phi)(x) = \int_G K(x, t)\phi(t)dt, \quad \phi(t) \in C(G) \quad (3-16)$$

y una regla de cuadratura con l puntos que aproxima a (3-16)

$$(A_l\phi)(x) = \sum_{i=1}^l a_i K(x, t_i)\phi(t_i), \quad \phi(t) \in C(G), \quad (3-17)$$

entonces se tiene:

Teorema 3.3.2 (Norma del operador en dimensión finita). Considere el operador definido en (3-17), entonces,

$$\|A_l\|_\infty = \max_{x \in G} \left\{ \sum_{i=1}^l |a_i K(x, t_i)| \right\}. \quad (3-18)$$

Demostración. La demostración se encuentra en [25].

Podemos notar que las normas de los operadores conservan similaridad entre el operador en dimensión finita y el de dimensión infinita. También, que la norma está directamente relacionada con la función núcleo K . Entonces, es de esperar que la norma del operador inverso en dimensión finita sea grande debido a que el operador inverso en dimensión infinita es discontinuo. Con estos dos hechos es posible concluir que el número de condición de la matriz A debe ser grande.

Para ilustrar cuán problemático es este hecho, se creó una imagen de prueba difuminada con una PSF exponencial simétrica con varianza $\sigma^2 = 4$ de tamaño 25×25 . Esta imagen fue contaminada con 1 %³ de ruido⁴, después se usó la matriz inversa de la ecuación (3-4) para encontrar la aproximación de la solución.

³Este porcentaje es calculado de forma que si la imagen está dada por el vector x y el error por el vector e , entonces se tenga que $\frac{\|e\|_1}{\|x\|_1} = 0,01$. Salvo que se diga lo contrario y en lo que sigue del texto los porcentajes de ruido se calculan de esta forma.

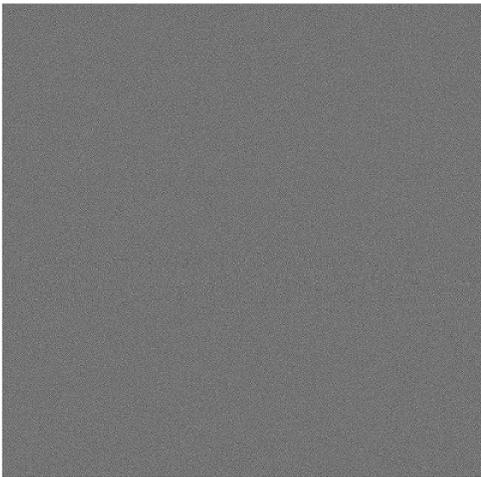
⁴También, salvo que se diga lo contrario y en lo que sigue del texto, este ruido se modela para cada componente de forma independiente como una variable aleatoria Z , donde $Z \sim \mathcal{N}(0, \sigma^2)$.

$E_0 = m m m m m m m m m m m m \dots$
 $E_1 = w w w w w w w w w w w w \dots$
 $E_2 = m w m w m w m w m w m w \dots$
 $E_3 = w m w m w m w m w m m w \dots$
 $E_4 = w m m w w m m w m w m w \dots$
 $E_5 = m w m w w m w m w m w m \dots$
 $E_6 = m w m w w m w w m w m w \dots$
 $E_7 = w m m w m w m w m w m w \dots$
 $E_8 = m m w m w m w m w m w m \dots$
 $E_9 = w m w m m w w m w w m w \dots$
 $E_{10} = w w m w m w m w m m w m \dots$
 $E_{11} = m w m w w m w m m w m m \dots$
 $\vdots \quad \vdots \quad \vdots$
 $E_u \approx w m w w m w m m m m m w \dots$

(a)

$E_0 = m m m m m m m m m m m m \dots$
 $E_1 = w w w w w w w w w w w w \dots$
 $E_2 = m w m w m w m w m w m w \dots$
 $E_3 = w m w m w m w m w m m w \dots$
 $E_4 = w m m w w m m w m w m w \dots$
 $E_5 = m w m w w m w m w m w m \dots$
 $E_6 = m w m w w m w w m w m w \dots$
 $E_7 = w m m w m w m w m w m w \dots$
 $E_8 = m m w m w m w m w m w m \dots$
 $E_9 = w m w m m w w m w w m w \dots$
 $E_{10} = w w m w m w m w m m w m \dots$
 $E_{11} = m w m w w m w m m w m m \dots$
 $\vdots \quad \vdots \quad \vdots$
 $E_u \approx w m w w m w m m m m m w \dots$

(b)



(c)

Figura 3-7: Restauración usando la matriz inversa de la ecuación (3-4). (a) Imagen original. Tomado de [43], (b) Imagen borrosa con 1% de ruido, (c) Imagen al aplicar la matriz inversa.

Este ejemplo muestra que al usar la matriz A^{-1} se obtienen resultados inaceptables y esto se debe principalmente al número de condición de la matriz A en este caso tiene un valor aproximado $Cond(A) = 3,88 \times 10^8$ (en la norma espectral) y a la influencia del término $A^{-1}e$ sobre la solución x^* en la ecuación (3-13). De esta forma se hace necesario usar otros

enfoques que implican permitir cierta diferencia entre el modelo y lo que realmente se tiene como imagen borrosa.

3.4. El modelo desde la programación lineal

Con las soluciones provenientes de aplicar la matriz inversa a la imagen borrosa se hace imperante el uso de un enfoque diferente al ya presentado. Para ello, en vez de resolver el sistema de ecuaciones (3-4), buscamos encontrar $x \in \mathbb{R}^n$ tal que distancia entre el modelo Ax y la imagen borrosa observada b sea lo más pequeña posible.

De la ecuación (3-4) se observa que la diferencia entre el modelo y la imagen borrosa está dada por

$$b - Ax = e. \quad (3-19)$$

Esto muestra que la forma para acercarse lo más posible a la imagen borrosa a partir del modelo de degradado que sintetiza A se logra minimizando el ruido que ésta tenga. Estamos interesados en considerar variables positivas y con esto construir un modelo para el cual sea posible usar programación lineal. Por esto se divide el vector error e en dos partes como sigue

$$e = e^+ - e^-. \quad (3-20)$$

Donde los superíndices “+” y “-” indican la parte positiva y negativa de e respectivamente. Estos son definidos como

$$e_i^+ = \max\{0, e_i\} \quad y \quad e_i^- = -\min\{0, e_i\}, \quad \text{con } i = 1, \dots, N. \quad (3-21)$$

Para esta definición si se toma el ejemplo en el que $N = 4$ y $e = (5, -3, 7, -2)^\top$ entonces

$$e^+ = (5, 0, 7, 0)^\top, \quad e^- = (0, 3, 0, 2)^\top. \quad (3-22)$$

Este ejemplo ilustra tres cosas:

- Que el vector e siempre puede ser descompuesto como la resta de dos partes no negativas.
- Para cada componente del vector error e_i se pueden encontrar dos números e_i^+ y e_i^- tales que al menos uno de los dos es cero.
- Permite calcular la norma del vector e deshaciéndose de los valores absolutos, luego,

$$\|e^+\|_1 = \sum_{i=1}^N e_i^+, \quad \|e^-\|_1 = \sum_{i=1}^N e_i^-, \quad y \quad \|e\|_1 = \|e^+\|_1 + \|e^-\|_1.$$

Por supuesto, las componentes de la foto las cuales son la intensidad de los colores (si se ve en formato RGB) de x son no negativas también. Con esto se permite plantear el problema minimización con el que se usará programación lineal

$$\begin{aligned} \min_{x \in \mathbb{R}^N} \quad & \|b - Ax\|_1 = \|e\|_1 = e_1^+ + e_2^+ + \cdots + e_N^+ + e_1^- + e_2^- + \cdots + e_N^- \\ \text{Sujeto a:} \quad & b - Ax = e \\ & e^+ \geq 0, e^- \geq 0, x \geq 0. \end{aligned} \quad (3-23)$$

Para aclarar la notación e identificar cuales son las variables, los coeficientes y hacer notoria la cantidad de restricciones se hacen las siguientes definiciones:

- $N = nm$, donde la imagen es una matriz de n filas y m columnas.
- $f \in \mathbb{R}^{3N}$ se define como

$$f = (x_1, \dots, x_N, e_1^+, \dots, e_N^+, e_1^-, \dots, e_N^-)^\top. \quad (3-24)$$

Este vector es la concatenación de la imagen, la parte positiva del error y la parte negativa del error.

- $H \in \mathcal{M}_{N \times 3N}(\mathbb{R})$ es la matriz por bloques definida así

$$H = \begin{bmatrix} A_{N \times N} & I_{N \times N} \\ I_{N \times N} & -I_{N \times N} \end{bmatrix}. \quad (3-25)$$

Donde $I_{N \times N}$ es la matriz identidad.

- $c \in \mathbb{R}^{3N}$ es el vector de “costos” y se define como

$$c = (\underbrace{0, \dots, 0}_{N \text{ ceros}}, \underbrace{1, \dots, 1}_{2N \text{ unos}})^\top. \quad (3-26)$$

Con esta nueva notación puede escribirse el problema primal (3-23) en la forma canónica, que se necesita para usar técnicas de programación lineal

$$\begin{aligned} \min_{f \in \mathbb{R}^{3N}} \quad & c^\top f \\ \text{Sujeto a:} \quad & Hf = b \\ & f \geq 0. \end{aligned} \quad (3-27)$$

Con esta notación se observa que no es necesario agregar variables de holgura para escribir (3-27) en la forma canónica, también se puede pensar que ya fueron agregadas a través de el vector de error e .

4 Restaurando imágenes

En este capítulo estudiaremos el desempeño y la efectividad de los métodos de solución de problemas de programación lineal. Comparamos el método simplex propuesto en [29] y el método de punto interior usado en este trabajo al solucionar el problema (3-27) con diferentes tipos de borrado y diferentes cantidades de ruido aditivo.

Con los resultados de la sección 4.2 podemos evaluar el funcionamiento del modelo. Teniendo en cuenta esto, en este capítulo se presentan los mayores aportes en este trabajo. En la sección 4.4 proponemos mejoras a este imponiendo términos de regularización a la norma de la solución así como a la derivada de la solución. También, en la sección 4.5 se hace una mejora al costo de cómputo del problema de programación lineal.

4.1. Medidas de desempeño

Para evaluar la calidad de las restauraciones se usan tres medidas, dos de ellas son muy usadas en la literatura correspondiente a imágenes. Sean Im y Ref dos matrices que representan a las imágenes en sus arreglos como pixeles y donde se busca medir que tanto se parece la imagen Im a una imagen de referencia Ref :

- Error relativo: en este caso se define

$$\|A\|_{F_1} := \sum_{i=1}^n \sum_{j=1}^m |a_{ij}| \quad (4-1)$$

donde A es una matriz de $n \times m$ y sus entradas están dadas por los números a_{ij} con $i = 1, \dots, n$ y $j = 1, \dots, m$. Entonces se define el error relativo como

$$Err(Im, Ref) = 100 \frac{\|Im - Ref\|_{F_1}}{\|Ref\|_{F_1}} \%. \quad (4-2)$$

La idea de proponer esta medida es cuantificar los errores relativos que están relacionados con el número de condición de la matriz A de la ecuación (3-23).

- Proporción Máxima de Señal a Ruido: también llamado PSNR por sus siglas en inglés (Peak Signal to Noise Ratio). Para definir esta medida se define el error cuadrático

medio (ECM), calculado como

$$ECM = \frac{1}{nm} \sum_{i=1}^n \sum_{j=1}^m (Im_{ij} - Ref_{ij})^2, \quad (4-3)$$

también se tiene en cuenta el mayor valor de intensidad que puede tomar la imagen, usualmente es denotado como Max y su valor, si se tienen b bits para representar la imagen es $Max = 2^b - 1$, entonces, el PSNR es calculado como

$$PSNR(Im, Ref) = 10 \log_{10} \left(\frac{Max^2}{ECM} \right) \text{dB}. \quad (4-4)$$

En este caso la calidad de la restauración es mejor a medida que $PSNR(Im, Ref)$ se hace cada vez más grande.

- Índice de similaridad estructural: También llamado SSIM por sus siglas en inglés (Structural Similarity Index), con este índice se busca evaluar la similaridad entre dos imágenes, con este índice que varía de cero a uno. Donde uno es que las imágenes son la misma y cero es que las imágenes no tienen estructuras comunes. Aquí se busca evaluar que tanto las estructuras de las imágenes se parecen entre si. Este índice es calculado como

$$SSIM(Im, Ref) = \frac{(2\mu_{Im}\mu_{Ref} + c_1)(2\sigma_{Im Ref} + c_2)}{(\mu_{Im}^2 + \mu_{Ref}^2 + c_1)(\sigma_{Im}^2 + \sigma_{Ref}^2 + c_2)}. \quad (4-5)$$

Donde:

- μ_{Im} y μ_{Ref} son los promedios de las intensidades de Im y Ref respectivamente.
- σ_{Im}^2 y σ_{Ref}^2 son las varianzas de las intensidades de Im y Ref respectivamente.
- $\sigma_{Im Ref}$ es la covarianza de las intensidades de Im y Ref .
- c_1 y c_2 son constantes para evitar dividir por números muy pequeños. No obstante, están relacionadas con la cantidad de bits, de forma que $c_1 = (k_1(2^b - 1))^2$ y $c_2 = (k_2(2^b - 1))^2$, con $k_1 = 0,01$ y $k_2 = 0,03$ son parámetros prefijados y b la cantidad de bits.

Más detalles acerca de esta medida se pueden consultar en [42].

De forma análoga que en el PSNR, μ_{Im} , σ_{Im}^2 y $\sigma_{Im Ref}$ son calculados como

$$\mu_{Im} = \frac{1}{nm} \sum_{i=1}^n \sum_{j=1}^m Im_{ij} \quad (4-6)$$

$$\sigma_{Im}^2 = \sum_{i=1}^n \sum_{j=1}^m (Im_{ij} - \mu_{Im})^2 \quad (4-7)$$

$$\sigma_{Im Ref} = \sum_{i=1}^n \sum_{j=1}^m (Im_{ij} - \mu_{Im})(Ref_{ij} - \mu_{Ref}). \quad (4-8)$$

En el caso en el que no haya confusión con las imágenes a tratar, se usarán las notaciones Err , $PSNR$ y $SSIM$ en lugar de $Err(Im, Ref)$, $PSNR(Im, Ref)$ y $SSIM(Im, Ref)$.

4.2. Restaurando imágenes de prueba

En esta sección se presentan algunos ejemplos de imágenes degradadas con diferentes funciones núcleo representadas por sus PSF que consideran diferentes tipos de degradado con diferentes niveles de ruido usando un algoritmo de punto interior para problemas de programación lineal. Los experimentos presentados en este capítulo fueron ejecutados en un computador con un procesador Intel® Core™ i5 8250u 1.60-3.40 Ghz con una memoria RAM de 4 Gb.

4.2.1. Borrado exponencial

En las imágenes de las Figuras 4-1 y 4-2 se ha degradado una imagen cuyas dimensiones son 236×197 usando con una PSF exponencial que tiene un tamaño de 15×15 , una varianza de 16 y un coeficiente q de 1,5:

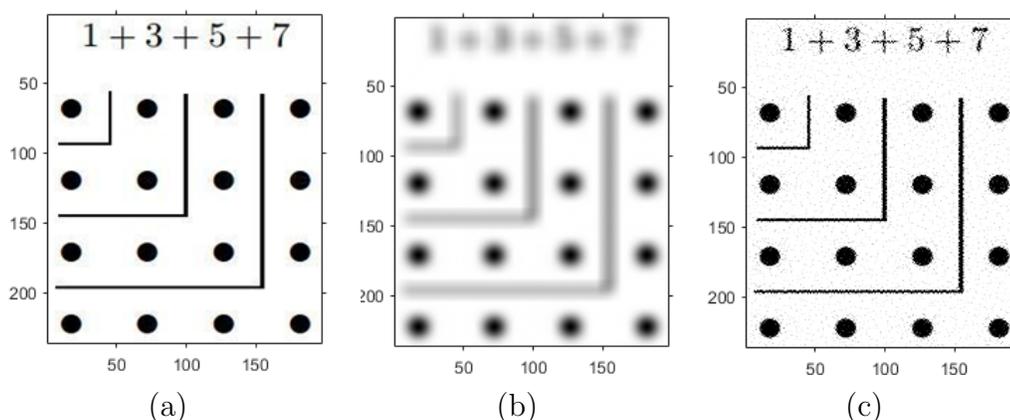


Figura 4-1: Imagen degradada con 0,05 % de ruido y su restauración. (a) Imagen original. Tomado de [41], (b) Imagen borrosa, (c) Imagen restaurada.

Con los resultados¹ :

¹En ambos casos las medidas Err , $PSNR$ y $SSIM$ se tiene como referencia la imagen real, se calcula con la imagen borrosa en la segunda columna y con la imagen restaurada en la tercera

	Imagen borrosa	Imagen restaurada
Err	9,45 %	4,62 %
PSNR	15,76 dB	18,46 dB
SSIM	0,606	0,522

Tabla 4-1: Resultados de la restauración de la imagen de la Figura 4-1.

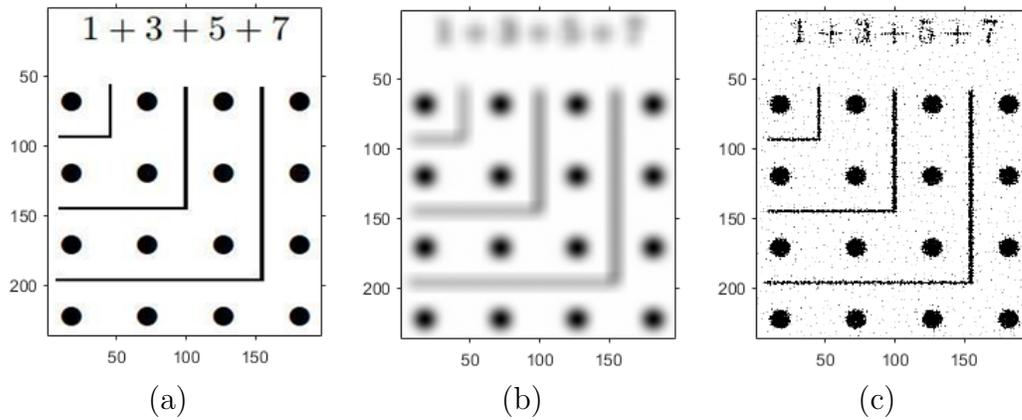


Figura 4-2: Imagen degradada con 0,1 % de ruido y su restauración. (a) Imagen original, (b) Imagen borrosa, (c) Imagen restaurada.

Con los resultados:

	Imagen borrosa	Imagen restaurada
Err	9,45 %	4,62 %
PSNR	15,76 dB	18,46 dB
SSIM	0,606	0,522

Tabla 4-2: Resultados de la restauración de la imagen de la Figura 4-2.

Estas restauraciones han tomado en promedio 2800 seg, la norma $\|\cdot\|_1$ de la imagen real, la restaurada y la borrosa es $1,07 \times 10^7$. También, para estos caso es posible calcular el número de condición (en la norma espectral) de la matriz A del modelo de programación lineal, el cual es $Cond(A) = 3,66 \times 10^7$, por lo que se puede observar que este modelo es muy propenso a los errores.

4.2.2. Borrado de movimiento

Usando la misma imagen de antes, en las Figuras 4-3 y 4-4, de nuevo es degradada usando con una PSF de movimiento horizontal que tiene un tamaño de 21×21 :

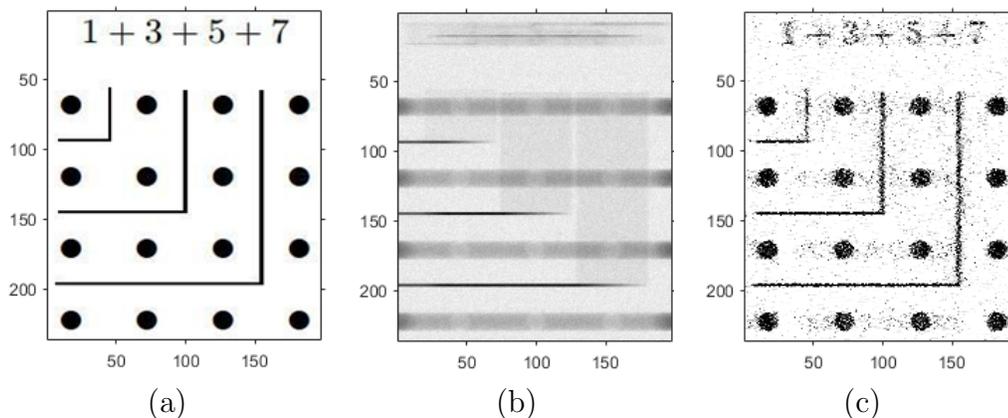


Figura 4-3: Imagen degradada con 2% de ruido y su restauración. (a) Imagen original, (b) Imagen borrosa, (c) Imagen restaurada.

Con los resultados:

	Imagen borrosa	Imagen restaurada
Err	13,54 %	6,66 %
PSNR	13,65 dB	16,46 dB
SSIM	0,445	0,444

Tabla 4-3: Resultados de la restauración de la imagen de la Figura 4-3.

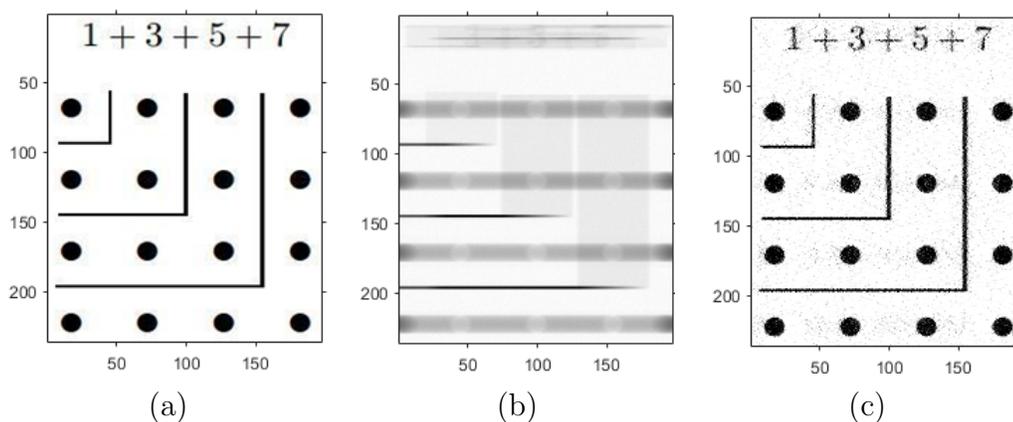


Figura 4-4: Imagen degradada con 0,5% de ruido y su restauración. (a) Imagen original, (b) Imagen borrosa, (c) Imagen restaurada.

Con los resultados:

	Imagen borrosa	Imagen restaurada
Err	12,85 %	3,48 %
PSNR	13,05 dB	23,27 dB
SSIM	0,590	0,664

Tabla 4-4: Resultados de la restauración de la imagen de la Figura 4-4.

Para este caso también fue posible calcular el número de condición de la matriz A , correspondiente a $Cond(A) = 6,77 \times 10^3$, en promedio las restauraciones tomaron 7 seg.

4.2.3. Borrado atmosférico (Moffat)

Usando la misma imagen de referencia, ésta ahora es degradada en las Figuras 4-5 y 4-6 mediante una PSF Moffat de tamaño 13×13 con una matriz $\begin{bmatrix} \sigma_1^2 & \sigma_2 \\ \sigma_2 & \sigma_3^2 \end{bmatrix} = \begin{bmatrix} 10 & 6 \\ 6 & 10 \end{bmatrix}$, un coeficiente $\beta = 0,8$:

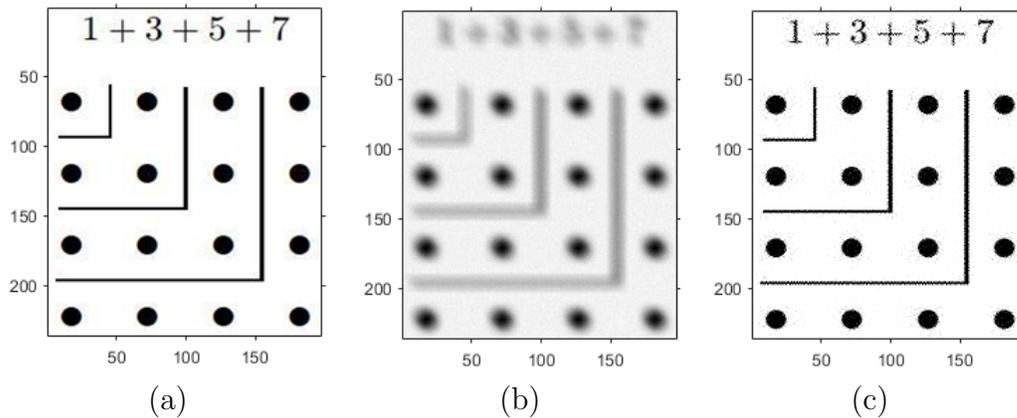


Figura 4-5: Imagen degradada con 1 % de ruido y su restauración. (a) Imagen original, (b) Imagen borrosa, (c) Imagen restaurada.

Con los resultados:

	Imagen borrosa	Imagen restaurada
Err	8,5 %	1,7 %
PSNR	17,19 dB	25,97 dB
SSIM	0,645	0,918

Tabla 4-5: Resultados de la restauración de la imagen de la Figura 4-5.

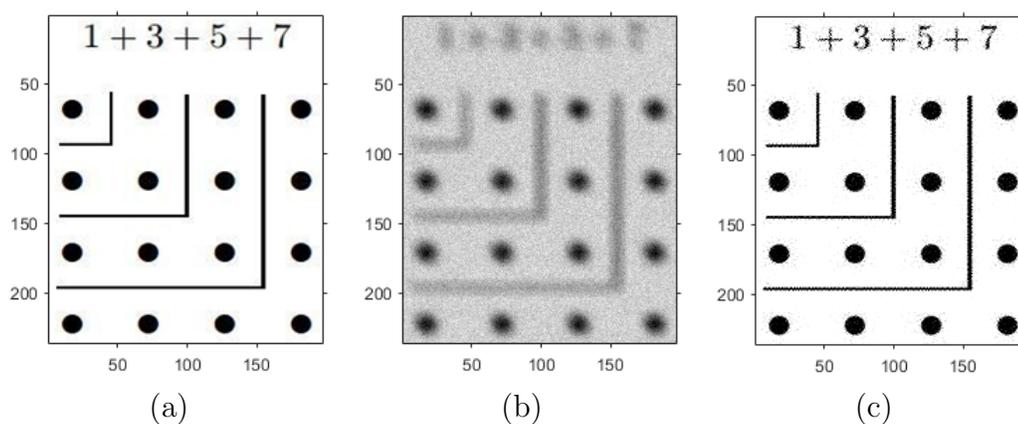


Figura 4-6: Imagen degradada con 5 % de ruido y su restauración. (a) Imagen original, (b) Imagen borrosa, (c) Imagen restaurada.

Con los resultados:

	Imagen borrosa	Imagen restaurada
Err	11,5 %	1,7 %
PSNR	16 dB	25,97 dB
SSIM	0,247	0,918

Tabla 4-6: Resultados de la restauración de la imagen de la Figura 4-6.

Las restauraciones producidas con esta PSF tomaron en promedio 2020 seg, para este caso el número de condición de la matriz A corresponde a $Cond(A) = 1,84 \times 10^8$. Lo interesante de esta PSF es que a pesar de los niveles de ruido inducidos la calidad de las restauraciones es muy similar. También, se observa que esta PSF no es rango uno, a diferencia de las PSF de los dos anteriores experimentos.

Observando las diferentes restauraciones, notamos que la simplicidad de la PSF es importante en las restauraciones, ya que reduce el número de condición de la matriz A y la cantidad de

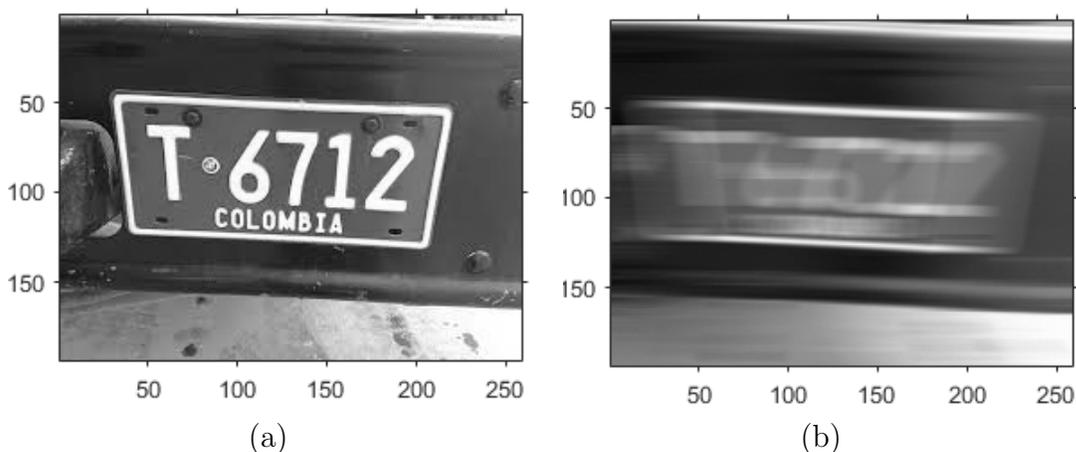
recursos empleados para la restauración, tales como: memoria RAM, procesador, lectura y escritura en el disco duro, y tiempo. Esto también permite tratar con niveles de ruido más altos. En la mayoría de los casos se obtienen resultados visualmente aceptables comparados con las imágenes degradadas originales. No obstante, no debe esperarse que los errores al final de la restauración sean pequeños, debido de nuevo, a los grandes números de condición que pueden presentarse en estas matrices.

4.3. El algoritmo simplex versus el algoritmo de punto interior

Al principio de este texto se presentaron dos alternativas para solucionar problemas de programación lineal que difieren bastante en la forma de alcanzar el óptimo. Por una parte, tenemos el algoritmo simplex que busca encontrar el punto óptimo en los vértices del conjunto factible iterando a través de ellos. Por otra parte, tenemos algoritmos de punto interior que buscan hallar el óptimo cruzando por el interior del conjunto factible aproximándose a la frontera de éste.

El propósito de esta sección es evaluar y contrastar estos dos algoritmos en el proceso de optimización, además, comparar las soluciones que presentan con el fin de dar una justificación al elegir el algoritmo que se usará en restauraciones futuras.

En la siguiente imagen de prueba mostrada en la Figura 4-7 que tiene dimensiones 194×259 se usó una PSF de movimiento horizontal que tiene tamaño de 51×51 a la cual se le añadió 0,5 % de ruido:



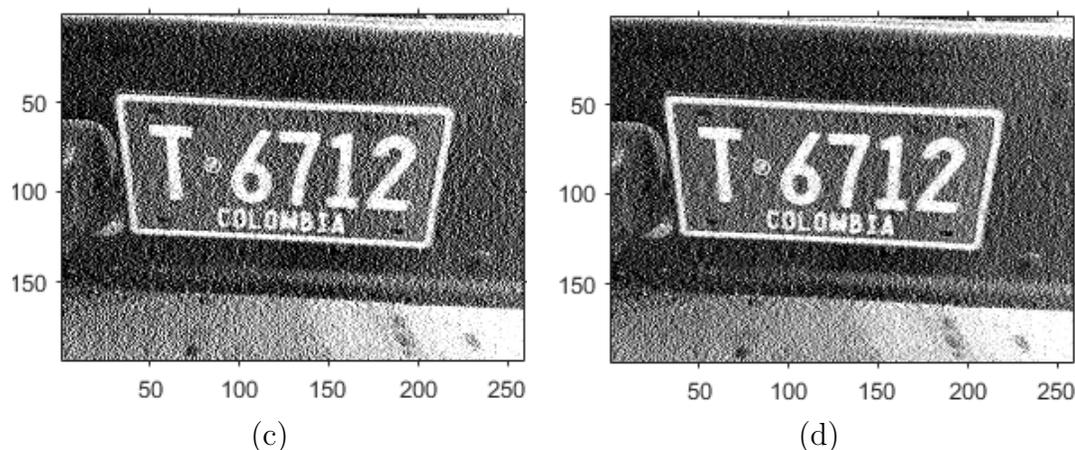


Figura 4-7: Imagen degradada con 0,5 % de ruido y sus restauraciones. (a) Imagen original. Tomado de [39], (b) Imagen borrosa, (c) Imagen restaurada con el método simplex, (d) Imagen restaurada con el método de punto interior.

	Imagen borrosa	Simplex	Punto interior
Tiempo	—	66,6 Seg.	8,7 Seg.
Número de iteraciones	—	67618	18
Norma $\ \cdot \ _1$	$5,619 \times 10^6$	$5,619 \times 10^6$	$5,619 \times 10^6$
Componentes no nulas	50246	44719	50246
Err	16 %	37,99 %	36,79 %
PSNR	17,99 dB	15,25 dB	13,53 dB
SSIM	0,673	0,278	0,239

Tabla 4-7: Resultados de las restauraciones de la imagen de la Figura 4-7.

Se puede apreciar que en la imagen restaurada con el método de punto interior se ve menos ruidosa encima de la placa que en la restaurada por el método simplex. Esta zona corresponde a una región homogénea y esto se ve reflejado en la medida Err. No obstante, los errores en cada uno de los píxeles son más grandes, como se observa en las medidas PSNR y SSIM.

También se encontró que las soluciones obtenidas por los métodos de punto interior y simplex son notoriamente diferentes, empezando por la cantidad de componentes no nulas; la menor cantidad es para el método simplex, esto era de esperarse pues en este algoritmo las soluciones son soluciones básicas que constan de bastantes ceros. Cuantificando esta diferencia, se tiene que la norma $\| \cdot \|_1$ de la diferencia de las soluciones es $1,08 \times 10^6$ lo que es equivalente al 20 % de la norma $\| \cdot \|_1$ de la imagen original, luego, estas soluciones aunque se parecen visualmente, son bastante lejanas.

En la Figura 4-8 se muestra una imagen de prueba que tiene dimensiones 89×114 fue degradada por una PSF exponencial de dimensión 5×5 con una varianza de 8, un coeficiente $q = 3$ y contaminada con 0,5 % de ruido:

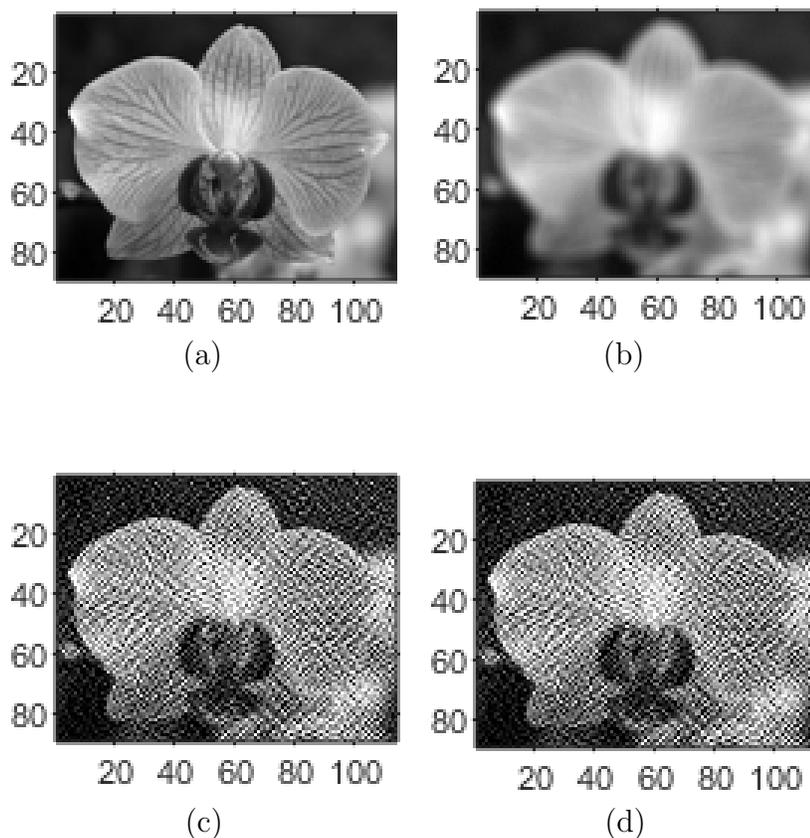


Figura 4-8: Imagen degradada con 0,5 % de ruido y sus restauraciones. (a) Imagen original, (b) Imagen borrosa, (c) Imagen restaurada con el método simplex, (d) Imagen restaurada con el método de punto interior.

	Imagen borrosa	Simplex	Punto interior
Tiempo	—	4709 Seg.	44 Seg.
Número de iteraciones	—	149824	17
Norma $\ \cdot \ _1$	$1,08 \times 10^6$	$1,08 \times 10^6$	$1,08 \times 10^6$
Componentes no nulas	10146	10146	10117
Err	8,45 %	14,01 %	14,01 %
PSNR	24,80 dB	14,01 dB	14,01 dB
SSIM	0,630	0,230	0,230

Tabla 4-8: Resultados de las restauraciones de la imagen de la Figura 4-8.

A diferencia del primer ejemplo, se puede observar que las imágenes y los resultados asociados a los dos algoritmos son similares y esto se debe a que la norma $\|\cdot\|_1$ de la diferencia de las soluciones es bastante pequeña y corresponde a $4,08 \times 10^{-5}$ que es aproximadamente el $4 \times 10^{-9} \%$ de la norma $\|\cdot\|_1$ de la imagen original. También, en la cantidad de tiempo empleada por los algoritmos es significativamente diferente dando un rendimiento mucho mejor para el algoritmo de punto interior el cual es aproximadamente 100 veces más rápido que el método simplex.

Teniendo en cuenta los resultados asociados a las Figuras 4-7 y 4-8 en términos de costo computacional y tiempo. Los algoritmos de punto interior son mucho mejores que el uso del método simplex, además que este último método tiende a estancarse en soluciones cíclicas. El uso de estos algoritmos de punto interior sacrifica un poco el hecho que las soluciones tengan muchos ceros y esto puede ser apreciado en las Tablas 4-7 y 4-8.

Por otra parte se puede ver que las restauraciones pueden tener resultados visualmente aceptables como en las Figuras 4-1, 4-4 y 4-7. En éstos se aprecian claramente en los caracteres que han sido degradados por el efecto borroso. Sin embargo, en todas las figuras se observa que las restauraciones tienen ruido, de forma que aparecen muchos puntos blancos donde se supone hay regiones homogéneas. Estos pueden afectar las restauraciones, de hecho, empeoran los resultados, como se ve en las Figuras 4-2, 4-4 y 4-8. Esto obliga a añadir elementos al modelo inicialmente descrito para evitar estos efectos aparezcan.

4.4. Regularización del problema

Con las dificultades encontradas en las soluciones de los ejemplos de las anteriores secciones nos proponemos ajustar el modelo de programación lineal anteriormente propuesto para evitar estos percances.

4.4.1. Regularización de la norma $\|\cdot\|_1$ de la solución

Es importante notar que existen PSF para las cuales es posible encontrar soluciones a la ecuación (3-4) de forma que $Ax + e = b$ y que tienen normas muy grandes. Por ejemplo, la imagen original de la Figura 3-7 tiene norma $\|\cdot\|_1$ igual a $2,1 \times 10^5$. Mientras, la norma $\|\cdot\|_1$ de la imagen resultante al invertir la matriz es igual a $2,5 \times 10^9$. Esto hace que las soluciones adquieran comportamientos que no son deseables e incluso que sean restauraciones que no sean aceptables.

Para lidiar con este problema decidimos penalizar (regularizar) la norma $\|x\|_1$ de la solución, esto para subsanar el problema de la aparición de pixeles con grandes valores que entorpezcan el proceso de restauración y controlar también la cantidad de ceros. También, aprovechando el hecho que las imágenes poseen cierto rango de valores $[0, 255]$ en el caso usual, aunque también, suele verse el rango $[0, 1]$. Decidimos convertir las restricciones de no negatividad en restricciones de caja, esto es, decidimos cambiar la desigualdad $0 \leq x$ por la desigualdad $0 \leq x \leq u$ donde $0 \leq u \in \mathbb{R}^N$ es un vector de las mismas dimensiones que x y actúa como cota superior de la solución óptima.

Al agregar el término $\alpha\|x\|_1$ a la función objetivo y las restricciones de caja al modelo descrito por la ecuación (3-23) queda

$$\begin{aligned} \min_{x \in \mathbb{R}^N} \quad & \|e\|_1 + \alpha\|x\|_1 \\ \text{Sujeto a:} \quad & b - Ax = e \\ & 0 \leq e, \quad 0 \leq x \leq u. \end{aligned} \tag{4-9}$$

Donde el coeficiente α determina que tanto es penalizada la norma de la solución con respecto al error del modelo $\|Ax - b\|_1$ y el vector u , salvo que se tenga información acerca de la imagen real, es un vector constante, usualmente se toma $u = 255 \cdot \mathbf{1}$, con $\mathbf{1}$ definido como el vector de unos de dimensión N .

4.4.2. Regularización de la derivada $\|\cdot\|_1$ de la solución

Para tratar con los casos en los que hay muchas variaciones de intensidad en ciertas regiones de la imagen, se decide contar también con la derivada de la solución para escoger de aquellas que posean la misma norma y que satisfagan de igual manera el modelo descrito por la matriz A .

En este caso usamos la derivada como una forma de medir las variaciones en una imagen. Cuando hay presencia de ruido se presentan muchos cambios en la imagen y esto se ve reflejado en la norma de la derivada, haciéndose más grande en aquellas imágenes que tienen muchos cambios de intensidad que en aquellas que tienen regiones homogéneas. Este hecho se ilustra en la Figura 4-9.

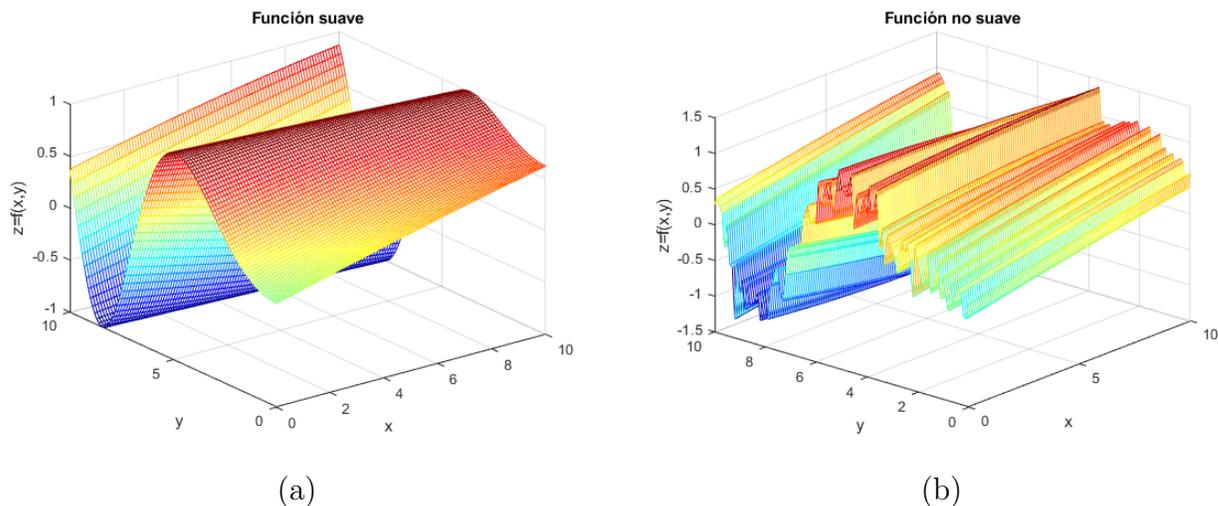


Figura 4-9: Ejemplos de funciones suaves y no suaves cuyas normas $\|\cdot\|_1$ son iguales. (a) Función suave cuya derivada tiene norma pequeña, (b) Función no suave cuya derivada tiene norma más grande que la función (a).

Normalmente estamos interesados en aquellas imágenes que no tienen demasiados cambios de intensidades, además que nos permite diferenciar de aquellas imágenes que son ruidosas de aquellas que no lo son.

Para incluir el término correspondiente a la derivada en el modelo se decide agregar el término $\gamma\|\nabla x\|_1$ a la función objetivo de la ecuación (4-9)

$$\begin{aligned}
 \underset{x \in \mathbb{R}^N}{\text{mín}} \quad & \|e\|_1 + \alpha\|x\|_1 + \gamma\|\nabla x\|_1 \\
 \text{Sujeto a:} \quad & b - Ax = e \\
 & 0 \leq e, \quad 0 \leq x \leq u.
 \end{aligned} \tag{4-10}$$

Donde al igual que la constante $\alpha > 0$, la constante $\gamma > 0$ determina la proporción en la que es penalizada la derivada de la función con respecto al error en el modelo y a la norma de la solución.

4.5. Abordaje y dificultades del modelo regularizado

Con el modelo que tiene elementos agregados expresado en la ecuación (4-10) se busca, de nuevo, escribirlo de forma que sea posible continuar usando programación lineal para resolverlo.

4.5.1. Abordaje

Para este propósito usamos los siguientes hechos:

1. Se puede observar que el vector $x \in \mathbb{R}^N$ tiene componentes no negativas, entonces,

$\|x\|_1 = \sum_{i=0}^N |x_i| = \sum_{i=0}^N x_i$, es decir, podemos obviar los valores absolutos y obtener la suma de las componentes del vector x .

2. Al igual que en la ecuación (3-23) escribimos $\|b - Ax\|_1 = e^+ - e^-$ con $e^+ \geq 0$ y $e^- \geq 0$.

3. Cuando se considera $\|\nabla x\|_1$ se hace referencia al gradiente de la imagen vista como función $\mathbb{R}^2 \rightarrow \mathbb{R}$. Teniendo en cuenta esto, se puede notar que $\|\nabla x\|_1 = \left\| \left(\frac{\partial x}{\partial x_1}, \frac{\partial x}{\partial x_2} \right) \right\|_1 =$

$\left\| \frac{\partial x}{\partial x_1} \right\|_1 + \left\| \frac{\partial x}{\partial x_2} \right\|_1$. Es decir se pueden separar las derivadas en la dirección horizontal y vertical de la imagen. De hecho, pueden ser calculadas realizando una convolución de la imagen con las matrices de tamaño 3×3 :

0	0	0	0	1	0
0	-1	1	0	-1	0
0	0	0	0	0	0
(a)			(b)		

Figura 4-10: Núcleos de convolución para calcular las derivadas parciales de la imagen. (a) Derivada parcial en la dirección horizontal, (b) Derivada parcial en la dirección vertical.

De igual forma como se calculó la matriz A . Las derivadas parciales se pueden calcular con matrices D_h y D_v de forma que $\frac{\partial x}{\partial x_1} = D_h x$ y que $\frac{\partial x}{\partial x_2} = D_v x$. Esto nos deja con el hecho que $\|\nabla x\|_1 = \|D_h x\|_1 + \|D_v x\|_1$.

4. Al igual que como se hizo en el segundo ítem, se separan los términos de las derivadas parciales en los vectores $h^+ \geq 0$, $h^- \geq 0$, $v^+ \geq 0$ y $v^- \geq 0$, de forma que $D_h x = h^+ - h^-$ y $D_v x = v^+ - v^-$.
5. Al limitar los posibles valores de la norma de la solución a un cierto rango obliga a que los posibles valores de los vectores error e^+ , e^- , h^+ , h^- , v^+ y v^- estén en el mismo rango. Esto se debe a que si a , b y $c > 0$ son tres números reales con $a, b \in [0, c]$, entonces, $|a - b| \in [0, c]$.

Con todas estas observaciones se puede reescribir el problema de optimización de la ecuación (4-10) en términos de los problemas de optimización asociados a las constantes α y γ como sigue:

$$\begin{aligned}
P(\alpha, \gamma) : \min_{x \in \mathbb{R}^N} & (e_1^+ + \cdots + e_N^+ + e_1^- + \cdots + e_N^-) + \alpha(x_1 + \cdots + x_N) + \\
& \gamma(h_1^+ + \cdots + h_N^+ + h_1^- + \cdots + h_N^- + v_1^+ + \cdots + v_N^+ + v_1^- + \cdots + v_N^-) \\
\text{Sujeto a:} & \quad b - Ax = e^+ - e^- \\
& \quad D_h x = h^+ - h^- \\
& \quad D_v x = v^+ - v^- \\
& \quad 0 \leq x, e^+, e^-, h^+, h^-, v^+, v^- \leq u.
\end{aligned} \tag{4-11}$$

De nuevo, al igual que en la ecuación (3-23), para establecer el tamaño del problema y escribirlo claramente como un problema de programación lineal, se hacen las siguientes definiciones:

- El vector de variables $f \in \mathbb{R}^{7N}$ se define como

$$\begin{aligned}
f = (x_1, \dots, x_N, e_1^+, \dots, e_N^+, e_1^-, \dots, e_N^-, \\
h_1^+, \dots, h_N^+, h_1^-, \dots, h_N^-, v_1^+, \dots, v_N^+, v_1^-, \dots, v_N^-)^\top. \tag{4-12}
\end{aligned}$$

Este vector es la concatenación de la imagen, la parte positiva del error y la parte negativa del error del modelo de borrado, la parte positiva y la parte negativa de las derivadas parciales.

- $H \in \mathcal{M}_{3N \times 7N}(\mathbb{R})$ es una matriz por bloques definida así

$$H = \begin{bmatrix} A_{N \times N} & I_{N \times N} & -I_{N \times N} & 0_{N \times N} & 0_{N \times N} & 0_{N \times N} & 0_{N \times N} \\ D_{h_{N \times N}} & 0_{N \times N} & 0_{N \times N} & -I_{N \times N} & I_{N \times N} & 0_{N \times N} & 0_{N \times N} \\ D_{v_{N \times N}} & 0_{N \times N} & 0_{N \times N} & 0_{N \times N} & 0_{N \times N} & -I_{N \times N} & I_{N \times N} \end{bmatrix}. \tag{4-13}$$

- $c \in \mathbb{R}^{7N}$ es el vector de “costos” y se define como:

$$c = (\underbrace{\alpha, \dots, \alpha}_{N \text{ alphas}}, \underbrace{1, \dots, 1}_{2N \text{ unos}}, \underbrace{\gamma, \dots, \gamma}_{4N \text{ gammas}})^\top. \tag{4-14}$$

- Se define el vector $d \in \mathbb{R}^{3N}$ de elementos constantes como:

$$d = (b^\top, \underbrace{0, \dots, 0}_{2N \text{ ceros}})^\top, \tag{4-15}$$

con b el vector que contiene en sus componentes las intensidades de los pixeles de la imagen borrosa.

- El vector $u \in \mathbb{R}^{7N}$ es extendido en cuanto a su tamaño, de forma que $u = 255 \cdot \mathbf{1}$, con $\mathbf{1}$ el vector de unos en \mathbb{R}^{7N} .

Así, es posible reescribir la familia de problemas descritos en la ecuación (4-11) de forma que se puede usar programación lineal de forma similar a la forma descrita en la ecuación (3-27) del modelo sin regularizar:

$$\begin{aligned}
 P(\alpha, \gamma) : \quad & \min_{f \in \mathbb{R}^{7N}} \quad c^\top f \\
 \text{Sujeto a:} \quad & Hf = d \\
 & 0 \leq f \leq u.
 \end{aligned} \tag{4-16}$$

Nótese que en el caso que $\alpha = \gamma = 0$ es el caso del modelo sin regularizar con restricciones de caja impuestas a la solución.

4.5.2. Dificultades

El principal problema al abordar este modelo regularizado es el tamaño de la matriz H descrita en la ecuación (4-13). Esta matriz tiene siete veces el tamaño que de la matriz H descrita en la ecuación (3-27). Sin tomar en cuenta la estructura de estas matrices, el número de elementos de esta matriz pasaría de $3N^2$ a $21N^2$ y esto se vuelve peor si analizamos el costo computacional de solucionar sistemas de ecuaciones asociados a estas matrices. Esto sucede ya que el método simplex pasa de tener orden $\mathcal{O}(3N^3)$ a tener orden $\mathcal{O}(63N^3)$ y el método de punto interior pasa de tener orden $\mathcal{O}(3N^3)$ a tener orden $\mathcal{O}(27N^3)$.

Pese a ello, la ventaja que tenemos es que la matriz H del modelo regularizado está compuesta por doce bloques de matrices nulas, seguido de seis bloques correspondientes a matrices identidad y dos bloques correspondientes a las matrices D_h y D_v . Debido a la simplicidad de las matrices de convolución que las generan (Figura 4-10) estas matrices tienen sólo dos elementos no nulos por fila, lo que hace que estas matrices tengan estructura muy similar a la matriz identidad. En total, éstas matrices conforman veinte de los veintiún bloques de la matriz H . Además, son matrices dispersas, es decir, son matrices que en su gran mayoría están conformadas por ceros y que no generan mucho costo computacional adicional al efectuar cálculos con ellas.

Dicho esto, al igual que en la matriz H del modelo sin regularizar, el costo computacional y las dificultades giran en torno a la matriz del modelo de degradación A . Por otra parte, para tratar con las condiciones de caja $0 \leq x \leq u$ hay algoritmos que trabajan con estas

condiciones de forma implícita, es decir, no es necesario involucrar variables de holgura la ecuación (4-16) para eliminar la desigualdad $x \leq u$.

Para aliviar el costo computacional relacionado con la matriz A en el caso que la PSF tiene rango uno, A admite la factorización descrita en la ecuación (3-10). Como la PSF k (la matriz de convolución) tiene rango uno es posible escribirla en su factorización en valores singulares como

$$k = U\Sigma V^\top = \sigma uv^\top = \hat{u}v^\top. \quad (4-17)$$

Donde σ es el único valor singular no nulo de la PSF, u y v son los primeros vectores columna de las matrices U y V respectivamente y $\hat{u} = \sigma u$. Las matrices Ac y Ar son construidas a partir la ecuación (3-10) y de estos vectores \hat{u} y v . De hecho, en [16] se muestra que la matriz $Ac = \text{toeplitz}(\hat{u}) + \text{hankel}(\hat{u})$ y $Ar = \text{toeplitz}(v) + \text{hankel}(v)$. Las matrices $\text{toeplitz}(u)$ y $\text{hankel}(u)$ son las matrices de Toeplitz y Hankel construidas para el vector u . Para conocer más acerca de las matrices de estructura Toeplitz, Hankel y circulante puede consultarse en [40].

No obstante, debido a la forma de calcular u y v existen elementos que deberían ser cero pero que en estos vectores se manifiestan como entradas muy pequeñas, a su vez se ve reflejado en las matrices Ac y Ar en la Figura 4-11. Debido a esto, se decidió truncar los elementos de las matrices Ac y Ar de forma que los elementos cuyos valores absolutos son menores a 10^{-12} son reemplazados con ceros (véase Figura 4-12).

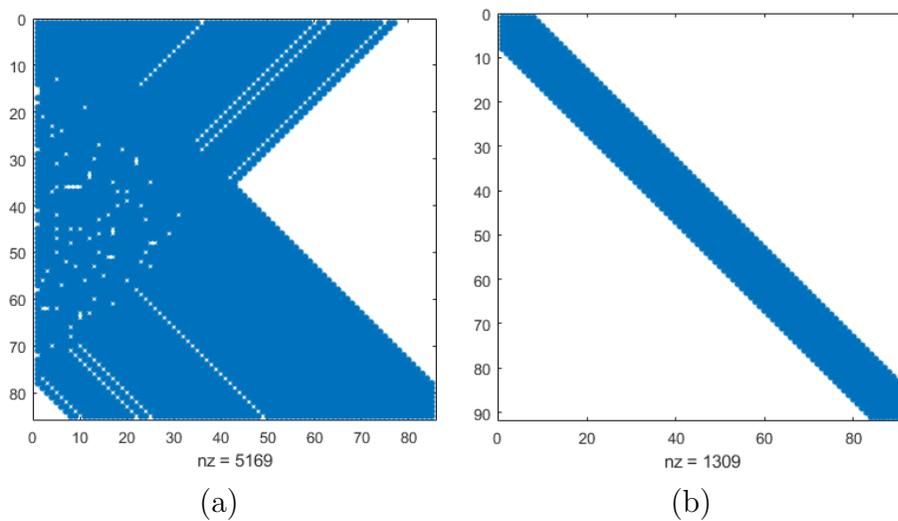


Figura 4-11: Ejemplos de matrices Ac y Ar para una PSF exponencial de una imagen de tamaño 85×91 . (a) Matriz Ac , (b) Matriz Ar . Los puntos azules indican elementos no nulos en la matriz, nz indica la cantidad de elementos no nulos en la matriz.

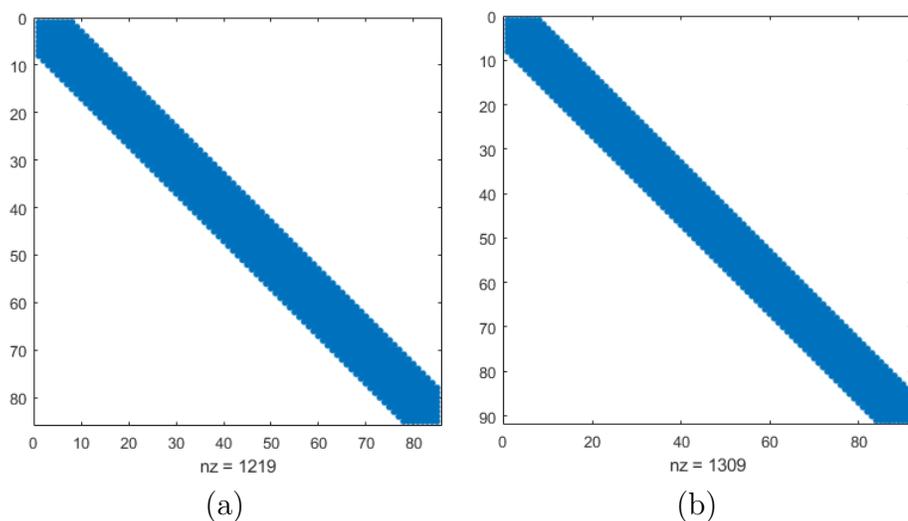


Figura 4-12: Matrices A_c y A_r truncadas de la figura 4-11. (a) Matriz A_c , (b) Matriz A_r . Los puntos azules indican elementos no nulos en la matriz, nz indica la cantidad de elementos no nulos en la matriz.

También esto tiene consecuencias sobre la matriz A , ya que le da una estructura en forma de banda en la que se tienen bastantes ceros con respecto a la cantidad de elementos no nulos.

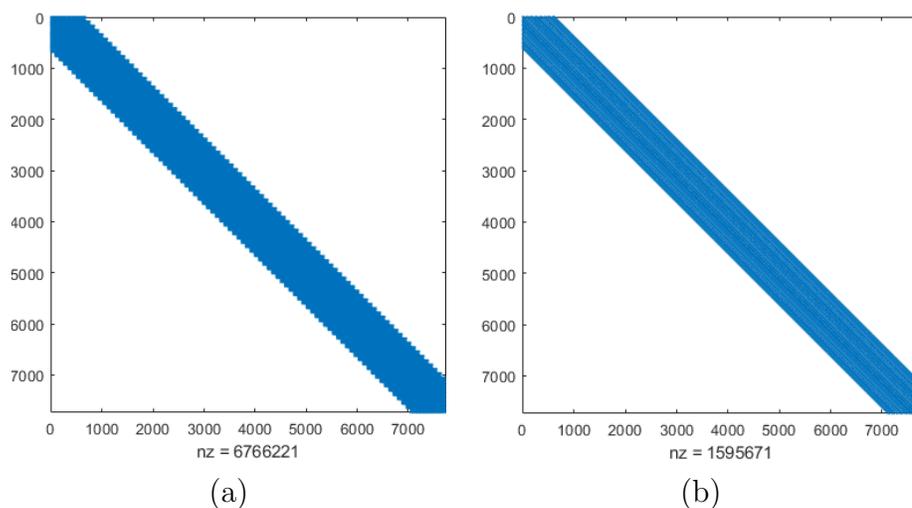


Figura 4-13: Matrices A construidas directamente de las matrices de las figuras 4-11 y 4-12. (a) Matriz A sin truncar, (b) Matriz A truncada. Los puntos azules indican elementos no nulos en la matriz, nz indica la cantidad de elementos no nulos en la matriz.

Por otra parte, para cuantificar los errores producidos por esta aproximación sobre la solución al problema de optimización, notamos que el óptimo encontrado por el algoritmo de punto interior 2.2.1 depende de soluciones de sistemas de ecuaciones asociadas a la matriz

A. Para cuantificarlos tenemos la siguiente proposición.

Proposición 4.5.1 (Perturbación de la matriz A). *Sea A la matriz construida a partir de una PSF que admite la factorización (3-10) $A = Ac \otimes Ar$. Si consideramos $\hat{A} = Ac_\delta \otimes Ar_\delta$ como la matriz factorizada de la misma manera que A , con la excepción que los elementos de las matrices Ac_δ y Ar_δ son los elementos de las matrices Ac y Ar salvo aquellos tales que su valor absoluto no es mayor que δ , con $0 < \delta \ll 1$ y en cuyo caso son reemplazados con ceros. Además $\Delta A = A - \hat{A}$, entonces,*

$$\frac{\|\Delta A\|_1}{\|A\|_1} \leq N\delta, \quad (4-18)$$

donde N es el número de píxeles en la imagen.

Demostración. *Para probar esto usaremos el teorema 2.1.3 y la ecuación (2-7) para estimar los errores provocados por esta aproximación. Para ello usamos los siguientes hechos:*

$$\|A\|_1 = \sup_{\|x\|_1=1} \|Ax\|_1 = \max_{1 \leq j \leq N} \left\{ \sum_{i=1}^N |a_{ij}| \right\}, \quad (4-19)$$

esto es, la norma $\|A\|_1$ es el máximo de la suma por columnas de la matriz A . Además,

$$Ae_i = c_i, \quad i = 1, \dots, N, \quad (4-20)$$

donde e_i es el i -ésimo vector de la base canónica y c_i es la i -ésima columna de la matriz A . Esto es, la convolución con una imagen que en un píxel vale uno y en los demás cero, que es equivalente a encontrar las columnas de la matriz A .

Tomando la notación de la ecuación (3-1) donde la imagen que en el píxel i vale uno y en los demás cero está dada por $X_i(t)$, el núcleo de convolución es $k(t)$ y sus soportes están contenidos en G . Observamos que la regla de cuadratura usada nos permite aproximar la integral

$$\begin{aligned} \int_{\mathbb{R}^2} \int_{\mathbb{R}^2} k(t-s)X_i(t)dt ds &= \int_G \int_G k(t-s)X_i(t)dt ds \\ &\approx \sum_{j=1}^N (Ae_i)_j = \sum_{j=1}^N (c_i)_j \\ &= \sum_{j=1}^N |(c_i)_j|, \quad i = 1, \dots, N. \end{aligned} \quad (4-21)$$

Es posible entonces, poner o quitar los valores absolutos ya que los elementos de la matriz A y la imagen son no negativos. Observando la segunda integral de la ecuación (4-21), notamos que

$$\begin{aligned}
\int_G \int_G k(t-s)X_i(t)dt ds &= \int_G \int_G k(t-s)X_i(t)ds dt \\
&= \int_{\mathbb{R}^2} X_i(t) \left(\underbrace{\int_{\mathbb{R}^2} k(t-s)ds}_{=1. \text{ Véase (3-7)}} \right) dt \\
&= \int_{\mathbb{R}^2} X_i(t)dt \\
&= 1, \quad i = 1, \dots, N.
\end{aligned} \tag{4-22}$$

Esto muestra que la suma por columnas de la matriz es uno para cada una. Luego, es posible concluir que $\|A\|_1 = 1$. Así, $\frac{\|\Delta A\|_1}{\|A\|_1} = \|\Delta A\|_1$.

Ahora, fijamos nuestra atención en la matriz ΔA , esta matriz está conformada por los elementos de la matriz A que fueron llevados a cero mediante el truncamiento mencionado anteriormente. Como la matriz A proviene de un producto tensorial, entonces, para cada componente a_{ij} de la matriz A , existen componentes b_{lk} y c_{rs} de las matrices Ac y Ar respectivamente, tales que

$$a_{ij} = b_{lk}c_{rs}, \quad \text{para ciertos } l, k, r, s \text{ que dependen de } i, j. \tag{4-23}$$

De la misma forma, como \hat{A} proviene de un producto tensorial, entonces, para cada componente \hat{a}_{ij} de la matriz \hat{A} , existen componentes \hat{b}_{lk} y \hat{c}_{rs} de las matrices Ac_δ y Ar_δ respectivamente, tales que

$$\hat{a}_{ij} = \hat{b}_{lk}\hat{c}_{rs}, \quad \text{para ciertos } l, k, r, s \text{ que dependen de } i, j. \tag{4-24}$$

Entonces, los elementos Δa_{ij} de la matriz ΔA tienen la forma

$$\Delta a_{ij} = a_{ij} - \hat{a}_{ij} = b_{lk}c_{rs} - \hat{b}_{lk}\hat{c}_{rs}. \tag{4-25}$$

Luego, si los números b_{lk} y c_{rs} satisfacen que $|b_{lk}| > \delta$ y $|c_{rs}| > \delta$, entonces, estos elementos no son truncados. Así, $\hat{b}_{lk} = b_{lk}$ y $\hat{c}_{rs} = c_{rs}$, por lo tanto $|\Delta a_{ij}| = 0 \leq \delta$.

Por otra parte, se sabe que las matrices Ac y Ar son construidas a partir del núcleo k de convolución, sus elementos son números entre cero y uno (de acuerdo a las propiedades de k), esto quiere decir que $|b_{lk}| \leq 1$ y $|c_{rs}| \leq 1$ para todas las componentes de las matrices Ac y Ar respectivamente. Además, si alguno de los números b_{lk} y c_{rs} satisface que $|b_{lk}| \leq \delta$ o bien $|c_{rs}| \leq \delta$ (supongamos que $|b_{lk}| \leq \delta$ sin pérdida de generalidad), entonces, se tiene que $\hat{a}_{ij} = 0$ y por lo tanto $\Delta a_{ij} = a_{ij}$. De esto se observa que

$$|\Delta a_{ij}| \leq \delta \max_{1 \leq r, s \leq m} \{ |c_{rs}| \} = \delta 1 = \delta. \tag{4-26}$$

Luego, al calcular $\|\Delta A\|_1$ se obtiene que

$$\|\Delta A\|_1 = \max_{1 \leq j \leq N} \left\{ \sum_{i=1}^N |\Delta a_{ij}| \right\} \leq \max_{1 \leq j \leq N} \left\{ \sum_{i=1}^N \delta \right\} = N\delta. \quad (4-27)$$

□

Este número $N\delta$ puede ser muy pequeño, puesto que la cantidad de pixeles en las imágenes trabajadas no supera el millón. Por ejemplo, al escoger $\delta = 10^{-12}$ se tiene que $\|\Delta A\|_1 = N10^{-12} \leq 10^{-6}$. Al considerar el cociente $\frac{\|\Delta A\|_1}{\|A\|_1} \leq \frac{10^{-6}}{1} = 10^{-6}$ se observa que es pequeño. Incluso, es pequeño comparado al factor $\frac{\|\Delta Y\|_1}{\|Y\|}$ pues este último suele ser del orden de $10^{-4} = 0,01\%$.

En resumen, es conveniente truncar estos valores, ya que permiten trabajar con imágenes más grandes debido al ahorro de memoria (en la Figura 4-13 se puede notar una disminución de aproximadamente 70% en la cantidad de elementos no nulos). También, mostramos que el error relativo de realizar este truncamiento es apenas comparable con el error relativo que proporciona el ruido de la imagen, por lo que tratamos con un modelo de características muy similares, pero, con un costo computacional menor.

4.6. Modelo regularizado y modelo sin regularizar

Al igual que en las secciones 4.2 y 4.3 presentaremos en el mismo esquema los resultados de regularizar el problema de acuerdo a (4-16) comparándolas con el problema sin regularizar descrito en (3-23). En esta sección usamos el algoritmo de punto interior para resolver cada uno de los problemas.

4.6.1. Comparación del modelo regularizado y modelo sin regularizar

En la siguiente imagen se aplicó una PSF de movimiento horizontal de tamaño 21×21 y a la imagen borrosa se le añadió 5% de ruido. Se usaron parámetros de regularización $\alpha = 2 \times 10^{-2}$ y $\gamma = 8 \times 10^{-2}$:

²De hecho, son los niveles de ruido que se han considerado en las imágenes tratadas.

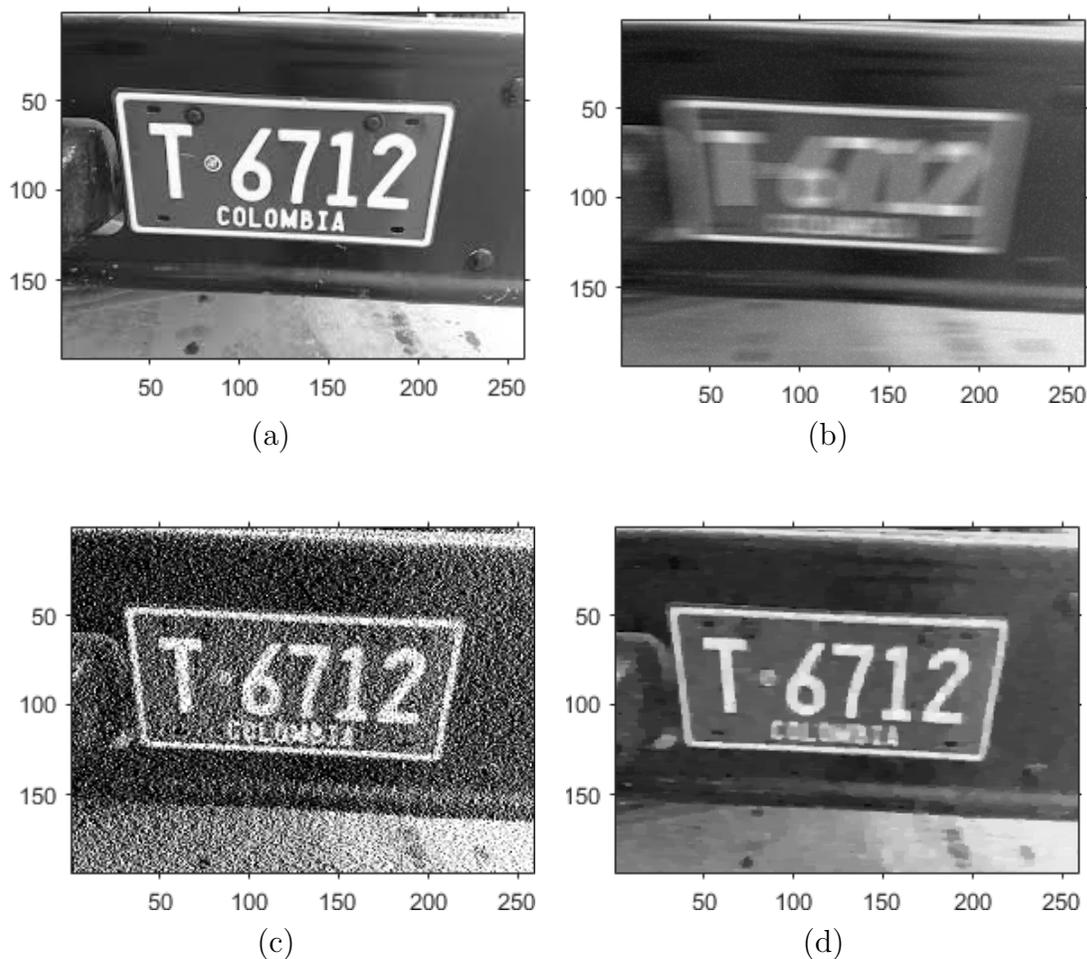


Figura 4-14: Imagen degradada con 5% de ruido y sus restauraciones usando el algoritmo de punto interior. (a) Imagen original, (b) Imagen borrosa, (c) Imagen restaurada con el modelo sin regularizar, (d) Imagen restaurada con el modelo regularizado.

	Imagen borrosa	Regularizado	Sin regularizar
Err	15 %	7,9 %	49,14 %
PSNR	19,77 dB	26,37 dB	12,43 dB
SSIM	0,640	0,840	0,170

Tabla 4-9: Resultados de las restauraciones de la imagen de la Figura 4-14.

En esta imagen podemos notar que varias de las zonas que son homogéneas en la imagen original son contaminadas con ruido en la imagen no regularizada. Sin embargo, en la imagen regularizada podemos volverlas homogéneas de nuevo, también se observa que no se regularizó demasiado la derivada como para perder ciertos detalles de la imagen (como lo son los

tornillos y pernos en la placa y cerca de ella). Por otra parte, hay que tener cuidado de no regularizar demasiado la derivada ya que ciertos detalles importantes como lo puede ser la palabra “Colombia” pueden verse afectados por este efecto.

En la siguiente imagen cuyo tamaño es de 255×271 se usó una PSF exponencial con una varianza de 16 con un coeficiente $q = 3$ y de tamaño 11×11 . Esta imagen fue contaminada con 0,5% de ruido. Se usaron parámetros de regularización $\alpha = 1 \times 10^{-2}$ y $\gamma = 2,5 \times 10^{-2}$:

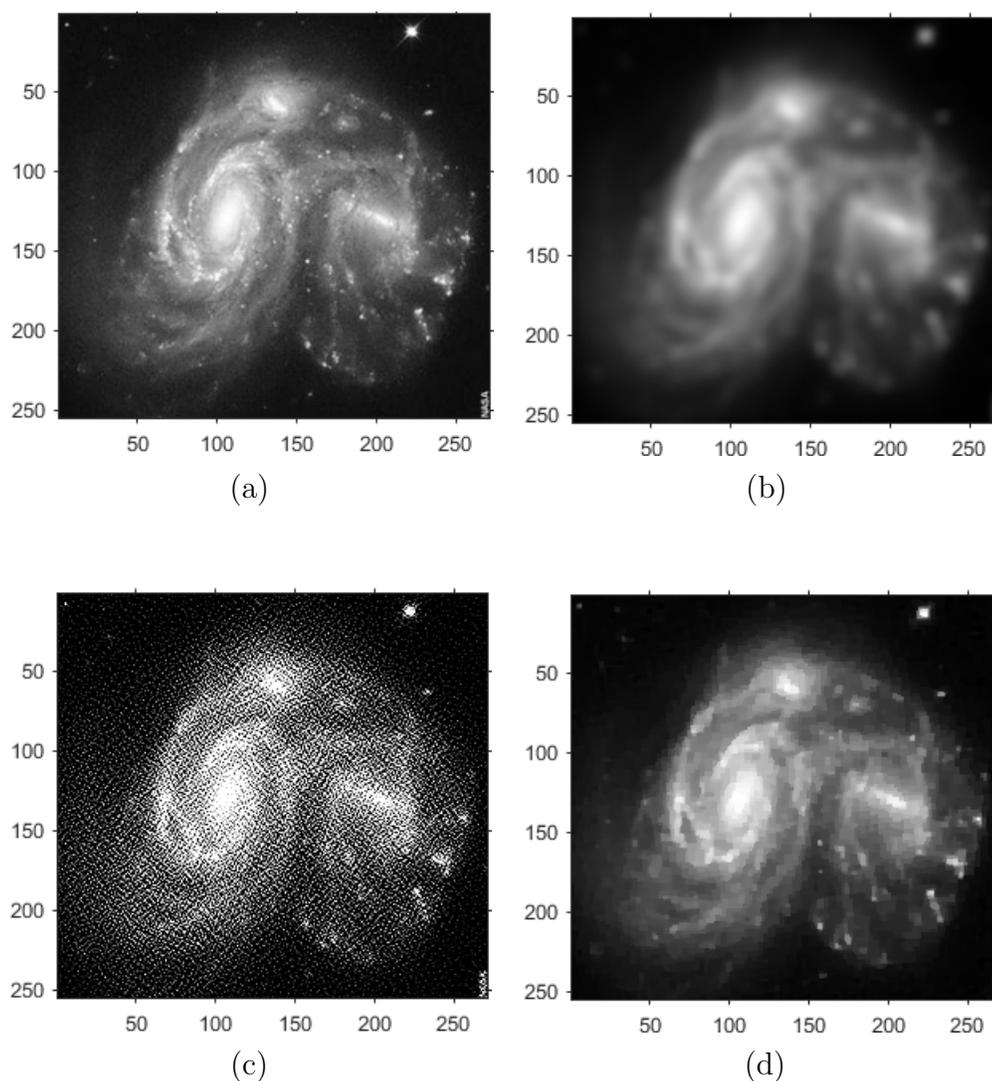


Figura 4-15: Imagen degradada con 0,5% de ruido y sus restauraciones usando el algoritmo de punto interior. Tomado de [36]. (a) Imagen original, (b) Imagen borrosa, (c) Imagen restaurada con el modelo sin regularizar, (d) Imagen restaurada con el modelo regularizado.

	Imagen borrosa	Regularizado	Sin regularizar
Err	9,39 %	7,50 %	81,43 %
PSNR	27,58 dB	31,06 dB	11,51 dB
SSIM	0,738	0,813	0,079

Tabla 4-10: Resultados de las restauraciones de la imagen de la Figura 4-15.

En esta imagen también se puede apreciar que al regularizar la norma y su derivada se puede evitar la aparición del ruido artificial que se observa en la imagen que no fue regularizada, permitiendo recuperar zonas que en verdad son homogéneas, al igual que el caso anterior. No obstante, debido a la naturaleza de la imagen en la que se presentan varios detalles que pueden ser interpretados como ruido (los puntos blancos que son polvo estelar o estrellas) se necesita regularizar menos la derivada de la solución. Esto permite que estos detalles no sean eliminados al ser confundidos con ruido. En este ejemplo se puede apreciar que se necesita un compromiso entre la cantidad de ruido admisible y la cantidad de detalles permitidos que puedan parecer ruido y esto debería estar reflejado en las constantes α y γ .

4.6.2. Comparación entre esquemas de regularización del modelo regularizado

Hemos mostrado que el modelo regularizado obtiene mejores resultados, tanto cualitativos como cuantitativos que el modelo sin regularizar. Ahora, estamos interesados en observar que tanto contribuye el regularizar la norma de la solución como el regularizar la derivada de la solución y compararlos. Para ello se realizó un experimento en el que se tomó la imagen (a) de la figura 4-16 que tiene dimensiones 221×285 y se degradó usando una PSF Moffat de tamaño 13×13 con una matriz $\begin{bmatrix} \sigma_1^2 & \sigma_2 \\ \sigma_2 & \sigma_3^2 \end{bmatrix} = \begin{bmatrix} 10 & -6 \\ -6 & 10 \end{bmatrix}$, un coeficiente $\beta = 0,6$ a la cual se le añadió 0,5 % de ruido. Luego, se restauró usando el modelo regularizado de tres esquemas:

- Regularizando solamente la norma de la solución. Esto significa que $\gamma = 0$.
- Regularizando solamente la norma de la derivada de la solución. Esto significa que $\alpha = 0$.
- Regularizando ambas normas.

En este ejemplo se usaron factores de regularización $\alpha = 2 \times 10^{-2}$ y $\gamma = 1,5 \times 10^{-2}$. Obteniendo las imágenes:

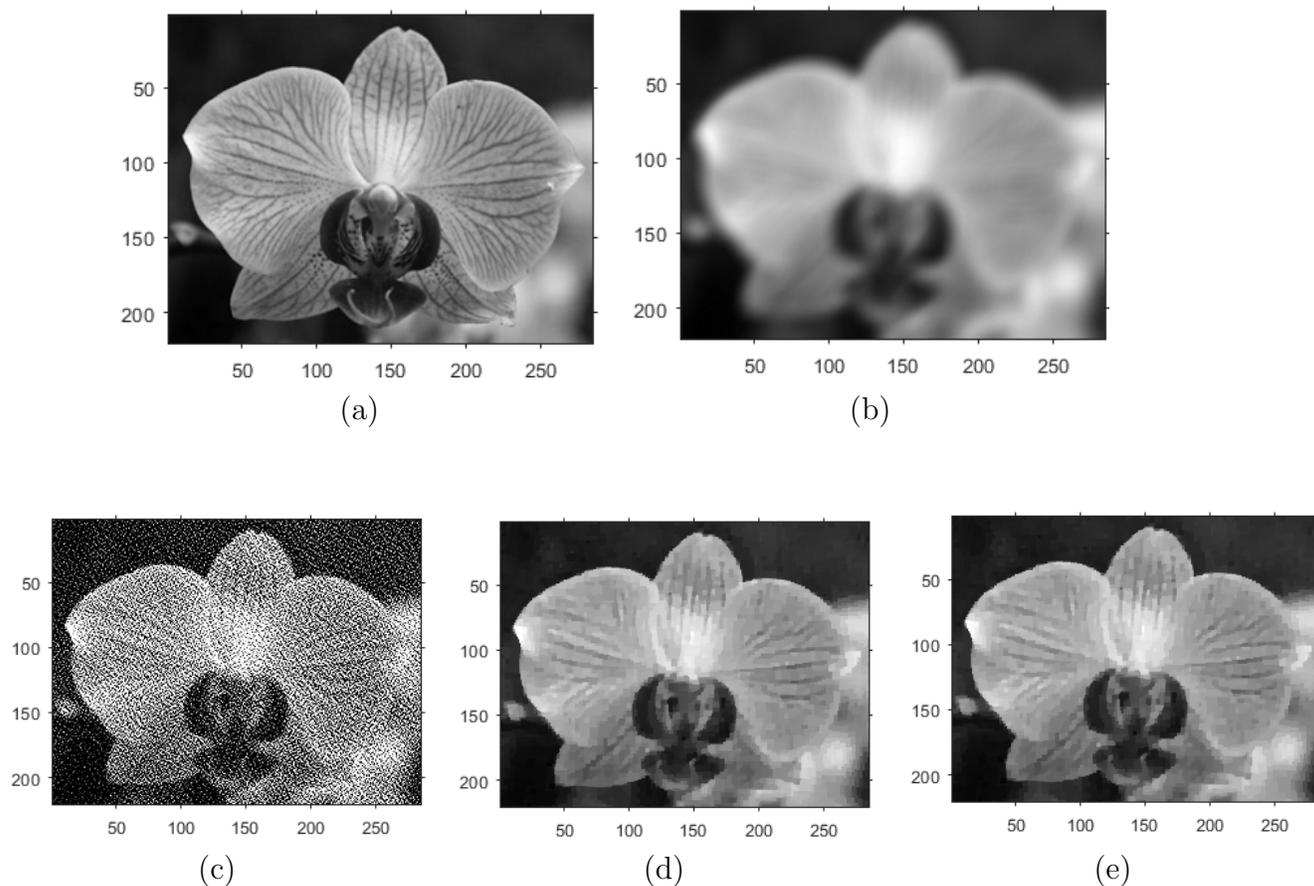


Figura 4-16: Comparación de los esquemas de regularización. (a) Imagen original, (b) Imagen borrosa con 0,5% de ruido, (c) Imagen restaurada regularizando sólo la norma de la solución ($\alpha = 2 \times 10^{-2}$, $\gamma = 0$), (d) Imagen restaurada regularizando sólo la norma de la derivada ($\alpha = 0$, $\gamma = 1,5 \times 10^{-2}$), (e) Imagen restaurada regularizando ambas normas ($\alpha = 2 \times 10^{-2}$, $\gamma = 1,5 \times 10^{-2}$).

y los resultados:

	Imagen borrosa	Esquema de regularización		
		$\ x\ _1$, (c)	$\ \nabla x\ _1$, (d)	$\ x\ _1$ y $\ \nabla x\ _1$, (e)
Err	10,24 %	67,78 %	6,40 %	6,41 %
PSNR	22,21 dB	9,37 dB	27,32 dB	27,37 dB
SSIM	0,544	0,055	0,722	0,722

Tabla 4-11: Resultados de las restauraciones de la imagen de la Figura 4-16. (c), (d) y (e) se refieren a las imágenes de la Figura 4-16

Esto muestra que la mayoría de la mejoría en la restauración de la imagen proviene de regularizar la norma de la derivada de la solución. Sin embargo, al regularizar la norma de la solución podemos imponer la condición en la que la solución tiene bastantes ceros. Aunque, como se puede observar esto es a cambio de un leve daño en la imagen restaurada. Esto se aprecia comparando el esquema de regularización en el que sólo se incluye la derivada con el esquema en el que se incluyen ambos.

4.7. Comparación contra otros métodos de deconvolución o deblurring

En este apartado observaremos el desempeño de nuestra propuesta con respecto a otros algoritmos ya propuestos por otros autores para ello usamos los siguientes algoritmos:

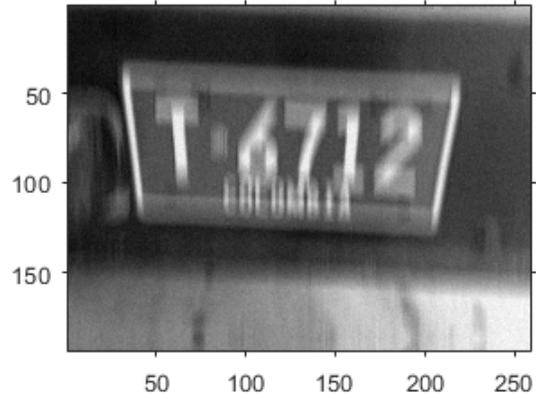
1. Método de deconvolución de Richardson-Lucy propuesto en [37].
2. Método de deconvolución de Wiener propuesto en [13].
3. Método blind deconvolution desarrollado en [15] y [3].
4. Algoritmo propuesto en [28] llamado “Blind Deconvolution Using a Normalized Sparsity Measure” (BDNSM).
5. Método de regularización de Tikhonov desarrollado en [34] y [16].
6. Método propuesto en el que usamos programación lineal.

Sobre las implementaciones de estos métodos, varios de estos algoritmos como el algoritmo de Richardson-Lucy, el algoritmo de Wiener y el algoritmo de blind deconvolution son funciones implementadas por defecto en MATLAB, la implementación del algoritmo (BDNSM) fue tomada de la web del autor [27] y la implementación del método de regularización de Tikhonov fue tomada del sitio web de los autores [8]. En los experimentos que siguen todos estos algoritmos fueron provistos con la misma PSF y fueron usados con los parámetros que tienen por defecto. Para los casos del método de regularización de Tikhonov y para el propuesto se usó el mismo α para ambos.

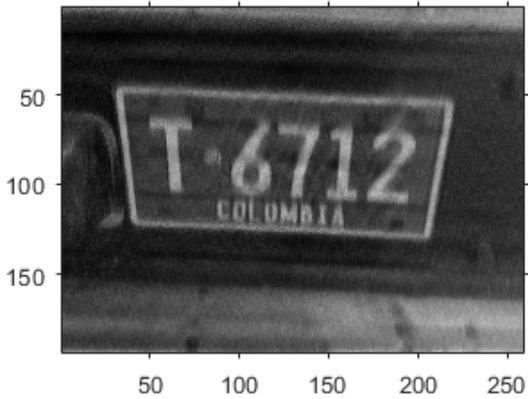
En el siguiente experimento, para degradar la imagen se usó una PSF de movimiento vertical de tamaño 31×31 y la imagen degradada fue contaminada con 3% de ruido. Se usaron parámetros $\alpha = 1 \times 10^{-2}$ y $\gamma = 7 \times 10^{-2}$. Con los cuales se obtienen las imágenes:



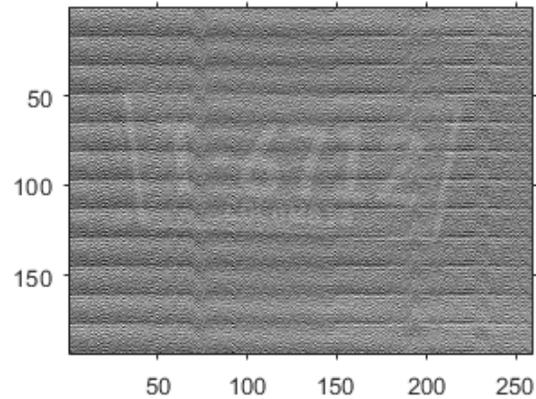
(a)



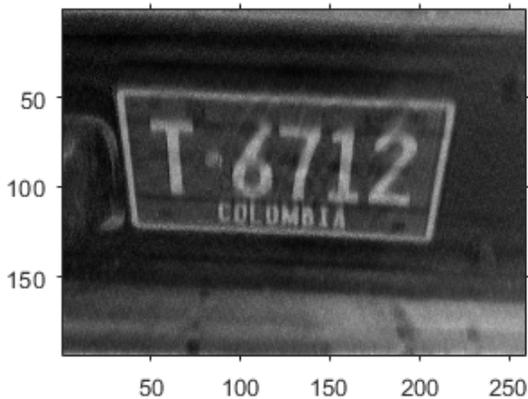
(b)



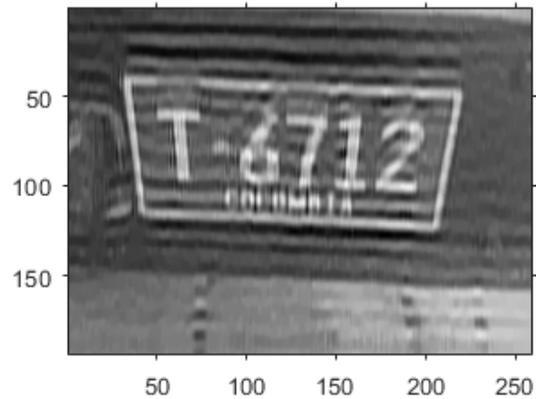
(c)



(d)



(e)



(f)

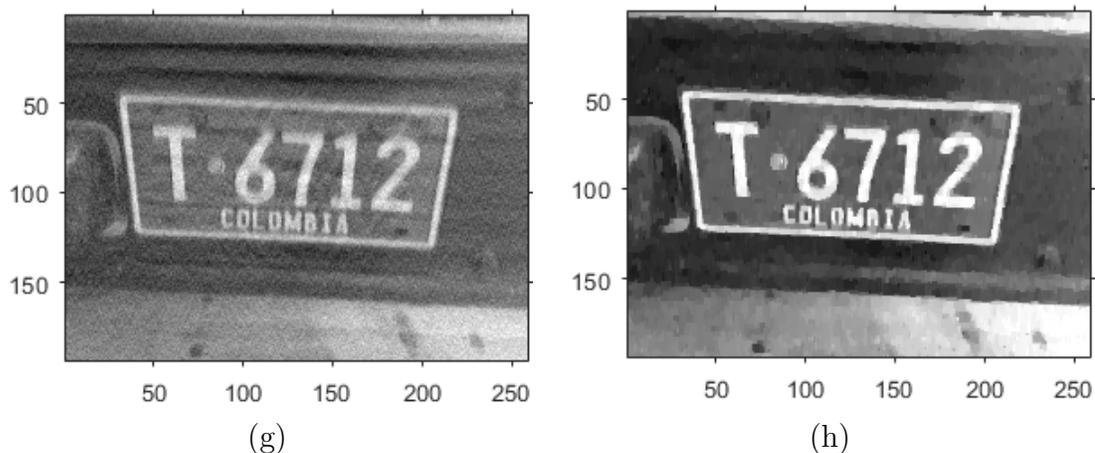


Figura 4-17: Imagen degradada con 3% de ruido y sus restauraciones. (a) Imagen original, (b) Imagen borrosa, (c) Imagen restaurada con método de Richardson-Lucy, (d) Imagen restaurada con método de deconvolución de Wiener, (e) Imagen restaurada con el método blind deconvolution, (f) Imagen restaurada con método BDNSM, (g) Imagen restaurada con método de regularización de Tikhonov, (h) Imagen restaurada con método propuesto.

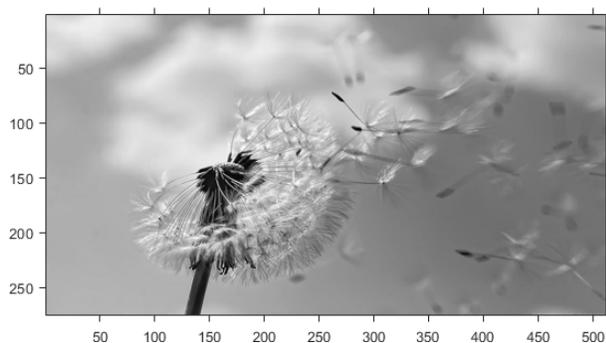
y los resultados:

Método\Medida	Err	SSIM	PSNR
Borrosa	26,46 %	0,425	14,99 dB
Richardson-Lucy	15,44 %	0,555	24,16 dB
Wiener	221,65 %	0,014	12,68 dB
Blind	15,63 %	0,546	24,36 dB
BDNSM	40,12 %	0,162	14,23 dB
Tikhonov	18,43 %	0,421	20,77 dB
Propuesto	7,97 %	0,787	25,06 dB

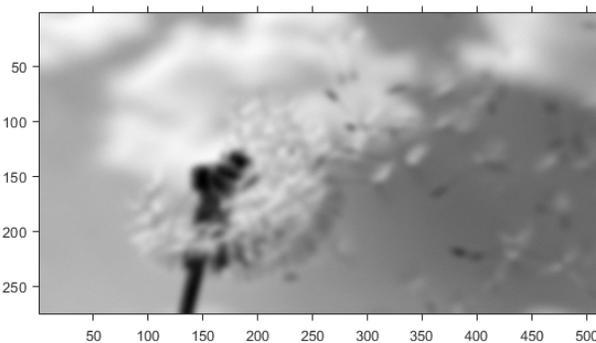
Tabla 4-12: Resultados de las restauraciones de la imagen de la Figura 4-17.

En estos resultados se aprecia la efectividad del modelo propuesto al restaurar imágenes con regiones homogéneas, como se ha visto antes. En los otros métodos tienden a aparecer líneas horizontales que originalmente no estaban en la imagen, esto puede ser visto como ruido adicional en la imagen. Particularmente en los métodos de Wiener (d) y regularización de Tikhonov (g) aparecen sus restauraciones con muchos puntos blancos (el ruido que comentamos que aparece en nuestro modelo sin regularizar). Esto puede deberse tanto la ausencia de la regularización en la derivada como la norma usada en optimización que se realiza en estas dos técnicas.

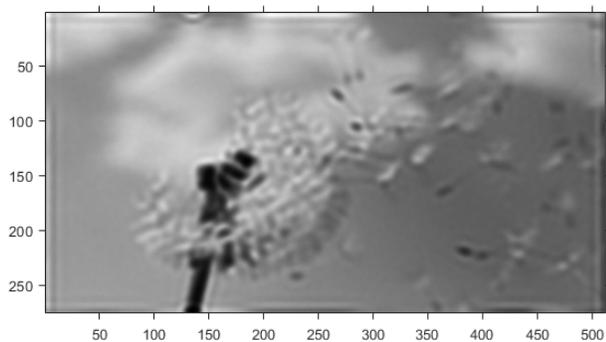
En el siguiente experimento se tomó una imagen de tamaño 275×512 la cual se degradó con una PSF Moffat de tamaño 17×17 con una matriz $\begin{bmatrix} \sigma_1^2 & \sigma_2 \\ \sigma_2 & \sigma_3^2 \end{bmatrix} = \begin{bmatrix} \frac{40}{3} & \frac{-20}{3} \\ \frac{-20}{3} & \frac{40}{3} \end{bmatrix}$, un coeficiente $\beta = 0,3$ a la cual se le añadió 0,5% de ruido. Con los cuales se obtienen las imágenes:



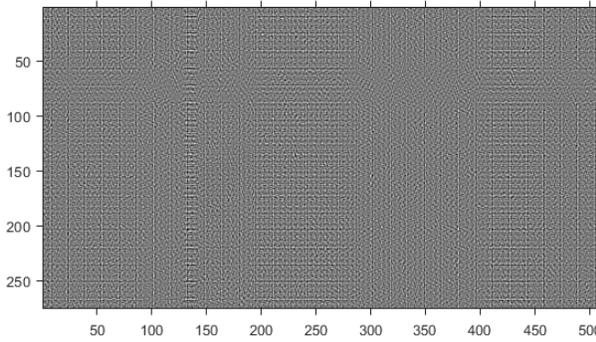
(a)



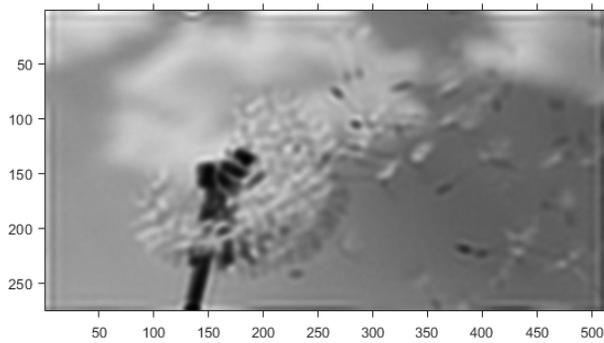
(b)



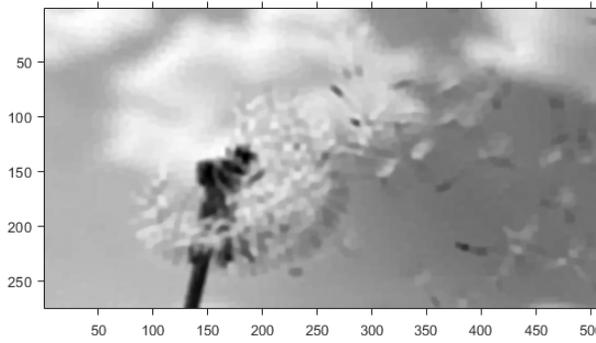
(c)



(d)



(e)



(f)

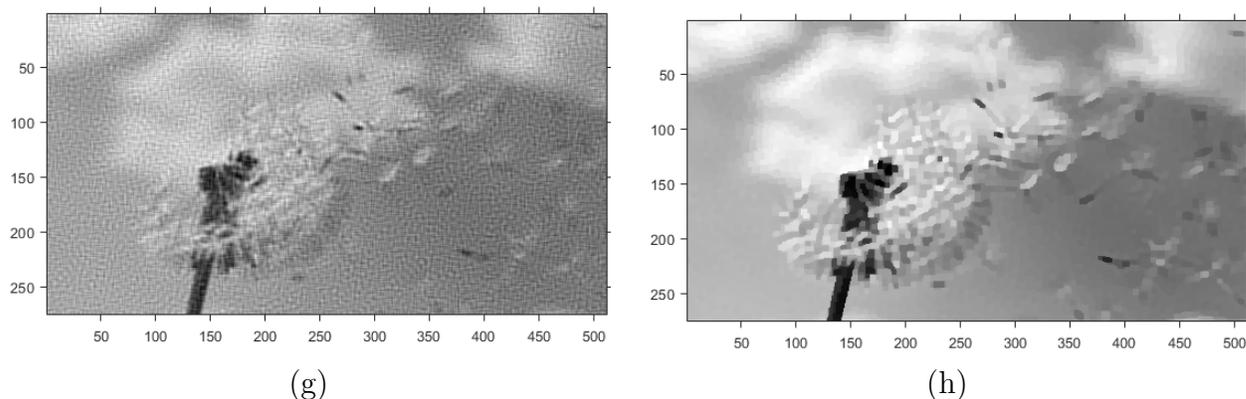


Figura 4-18: Imagen degradada con 0,5 % de ruido y sus restauraciones. (a) Imagen original, (b) Imagen borrosa, (c) Imagen restaurada con método de Richardson-Lucy, (d) Imagen restaurada con método de deconvolución de Wiener, (e) Imagen restaurada con método blind deconvolution, (f) Imagen restaurada con método BDNSM, (g) Imagen restaurada con método de regularización de Tikhonov, (h) Imagen restaurada con método propuesto.

y los resultados:

Método\Medida	Err	SSIM	PSNR
Borrosa	3,15 %	0,830	26,41 dB
Richardson-Lucy	3,47 %	0,834	28,13 dB
Wiener	148,41 %	$2,34 \times 10^{-5}$	13,64 dB
Blind	3,47 %	0,834	28,14 dB
BDNSM	2,73 %	0,857	25,57 dB
Tikhonov	9,07 %	0,2878	24,25 dB
Propuesto	2,77 %	0,857	27,94 dB

Tabla 4-13: Resultados de las restauraciones de la imagen de la Figura 4-18.

Aquí a diferencia del experimento anterior se puede apreciar que las partes más difíciles de recobrar son aquellas que están más cerca a la flor, en los métodos iterativos (Richardson-Lucy (c) y blind deconvolution (e)) se pueden apreciar ciertas líneas que se van formando cerca de las fronteras de la imagen, además, estas imágenes son más opacas que el resto. El método de deconvolución de Wiener (d) provee resultados parecidos a los que se obtenían en la Figura 3-7 por lo que se puede notar que este método es muy sensible al ruido introducido a la imagen. Por otra parte el método BDSNM (f) aprovecha el hecho que el fondo de la imagen es borroso en la imagen original, también evita el hecho que la imagen se vea opaca como en los otros métodos. El método de regularización de Tikhonov (g) posee esa estructura granular en parte debido a que aproxima la PSF por una que tiene rango uno, esto para

realizar los cálculos necesarios bajo esa hipótesis. Por último el método propuesto (h) logra evitar la opacidad de la imagen y la estructura granular de los demás métodos. Sin embargo, los coeficientes de regularización deben ser escogidos con más cuidado ya que si se presta atención a los detalles hay zonas homogéneas que no lo son por lo que se debe ser más laxo con la norma de la derivada en este caso.

5 Aplicaciones y Productos

Finalmente, queremos destacar que durante el desarrollo de este estudio, avances del trabajo fueron presentados en eventos académicos nacionales e internacionales, de los cuales se resaltan:

- XXII CCMN: Vigésimo segundo congreso colombiano de matemáticas. Universidad del Cauca, Junio 10-14, (2019), Popayán, Colombia. Charla contribuida sobre el modelo de restauración de imágenes y su solución con el método simplex.
- Workshop on Optimization Methods and Applied Mathematics (WOMAP). Universidad Nacional de Colombia, Octubre 4-5, (2019) Bogotá, Colombia.
- 15th International Symposium on Medical Information Processing and Analysis (SI-PAIM 2019), Noviembre 6-8, (2019), Medellín, Colombia. Sobre la aplicación del modelo de restauración en imágenes médicas.
- Revista Ingeniería, Universidad Distrital sometido en Mayo (2020). “Blurred Image Restoration Using a Regularized Linear Programming Model” en este artículo se mencionan los avances del modelo regularizado y sus comparaciones con otros métodos similares (Secciones 4.4-4.7).

5.1. International Symposium on Medical Information Processing and Analysis

En el proceso de sobre-muestreo, esto es, aumentar las dimensiones de una imagen, pueden producirse efectos como aliasing (solapamiento), desenfoco o adición de ruido que afectan principalmente a los bordes de las imágenes. Por esas razones es necesario elegir un método que conserve la calidad de las imágenes de forma que minimice estos problemas. En este artículo presentado al International Symposium on Medical Information Processing and Analysis (SI-PAIM) celebrado en noviembre del 2019 en la ciudad de Medellín, Colombia¹ y adjunto a continuación, se presenta un método alternativo para restaurar imágenes borrosas provenientes de métodos de interpolación usados para hacer sobre-muestreo de imágenes. Esto usando programación lineal para resolver un problema de minimización establecido en

¹Puede consultarse en el link: <https://www.spiedigitallibrary.org/conference-proceedings-of-spie/11330.toc>

la norma L_1 . El modelo requiere la imagen borrosa y algunos conocimientos previos sobre el tipo de función de desenfoque (función de dispersión puntual). En el método propuesto obtenemos un PSNR de 30 dB superando un método bi-lineal clásico por 4 dB en un conjunto de treinta imágenes de un conjunto de datos de imágenes de resonancia magnética cardíaca.

A Deblurring Model for Super-Resolution MRI Interpolated Images

José Fuentes^a and Jorge Mauricio Ruiz V^a

^a Departamento de Matemáticas, Universidad Nacional de Colombia

ABSTRACT

In the up-sampling process may occur effects like aliasing, blurring or noise addition which mainly affect the edges of the images. For those reasons is necessary to choose a method that preserves images quality so that these problems are minimized. In this paper, we present an alternative method to restore blurred images using linear programming to solve a minimization problem stated in the L_1 norm. The model requires the blurred image and some prior knowledge about the blurring function type (Point spread function). In the proposed method we obtain a PSNR of 30 dB overcoming a classic bi-linear method by 4 dB in a set of thirty images from a cardiac MRI data set.

Keywords: Blurred images, linear programming, interpolation, super-resolution.

1. INTRODUCTION

Image interpolation handles the problem of using a low resolution (LR) image to estimate a high resolution (HR) image assuming there is a relationship between them. This is helpful in several cases to improve the details of the HR image that can not be seen in the LR imag. In the medical imaging case this procedure can be combined with some other strategy to improve the image quality. For example, Leng et. al.¹⁹ present an image registration process consisting of two stages: registration and interpolation.

The interpolation problem⁷ consists that for given observations $f(x_i)$, $i = 1, \dots, m$, and basis functions U_k , $k = 1, \dots, n$, the coefficients a_k must be chosen such that if $g(x)$ is defined by

$$g(x) = \sum_{k=1}^n a_k U_k(x)$$

the differences $|g(x_i) - f(x_i)|$ are small for $i = 1, \dots, m$.

The interpolation methods can be divided²⁰ adaptive and the non-adaptive. Some classical methods e.g., bi-cubic and nearest neighbor are non-adaptive methods and are popular because they are fast and simple. The main goal of adaptive methods are reconstruct the edges of the HR image from the LR image. Usually, these methods have a higher computational cost than the non-adaptive methods. In the following, some adaptive and non-adaptive interpolation methods are briefly presented.

The non-adaptive methods are the classic methods, e.g., bi-linear, nearest neighbor, bi-cubic,²¹ Lanczos, among others. These also use the information around the pixel to be interpolated. All of them use a kernel function that indicates how neighboring pixels are used to perform interpolation. The disadvantage of these methods are the blurred or degraded edges appearance.

Unlike the classical methods, the adaptive methods use the LR image edges to estimate the edges of the HR image, usually they are combined with the non-adaptive. A method to find edges is proposed by Xi and Orchard^{18,16} called New Edge-Directed Interpolation (INEDI), in this method the co-variance in the LR image is calculated and used to estimate it in the HR image and It is combined with a bi-linear interpolation. Asuni and Giachetti¹⁵ present a strategy to obtain smooth iso-level curves, it is called Iterative Curvature Based Interpolation (ICBI). Zhou and Shen¹⁷ extend a cubic²¹ interpolation so that it is a directed interpolation to estimate and preserve the edges of the (HR) image, obtaining a cubic-directed interpolation (DCC).

José Fuentes: E-mail: jefuentesg@unal.edu.co

Most of the super-resolution methods, and therefore, the interpolation methods have the problem of increasing the resolution of an image while preserving the high-resolution image quality²⁰¹⁸¹⁷. The high-resolution images appear blurred. So, this problem can also be seen as restoring a blurred image. Therefore, a method of deblurring is required to be used together with an interpolation method. For Montenegro⁹ and Hansen et. al⁵ this deblurring process is done by solving a minimization problem stated in the L_2 norm. The process is included in a family of problems called “inverse problems” and this method uses a family of operators, each of them indexed by a constant $\alpha > 0$. This results in a Tychonoff regularization. The method uses an estimation of the blur model also called Point spread function (PSF); this function models the blur effect on the image.

A main contribution of this work is a novel super-resolution method which has been assessed with MRI blurred images. Unlike most super-resolution methods, which model local relationships, this method uses linear programming and solves a global minimization problem in terms of the L_1 norm, under which the distance between the model and the observations is computed. The proposed method needs some knowledge about the PSF that affects the image and the blurred image to estimate the real image.

The paper is organized as follows: Section 2 describes the mathematical model, Section 3 presents the methodology, Section 4 presents the experimental results, and Sections 5 and 6 makes comments and conclusions about the proposed method.

2. DEBLURRING MODEL

The goal of image deblurring is to recover the sharp contents from a blurry image. From the discrete point of view, the image degradation is caused by blurring and noise, and it can be described as follows:

$$B(s_i) = \sum_{j=1}^N f(s_i, s_j)X(s_j) + E(s_i), \quad i = 1, 2, \dots, N, \quad j = 1, 2, \dots, N. \quad (1)$$

Where $B(x)$ is the blurred image, $X(x)$ is the original image, $f(x, \xi)$ is the blur kernel (or point spread function, PSF), $E(x)$ is an additive noise and N is the number of pixels. This image degradation model can be written in terms of a matrix-vector formulation, that is,

$$b = Ax + e \quad (2)$$

Where b is the blurred image, x the original image, e the additive noise, all of them organized in a vector of N entries and A is the matrix model with $a_{ij} = f(s_i, s_j)$.

It turns out, that equation (2) is an ill conditioned problem, then its solution x is very sensitive to noise e . This characteristic of the problem comes from the fact of the problem (2) is the discretization of an integral equation⁹¹³. Therefore, a “good” solution would be that one which minimizes the noise. Keeping this in mind the following minimization problem is proposed:

$$\begin{aligned} & \underset{e \in \mathbb{R}^N}{\text{minimize}} && \|e\|_1 \\ & \text{Subject to:} && Ax - b = e \\ & && x \geq 0. \end{aligned} \quad (3)$$

Where $\|\cdot\|_1$ indicates the L_1 norm and $x \geq 0$ for $x \in \mathbb{R}^N$ indicates that each component x_i of x is non-negative for $i = 1, 2, \dots, N$. The choice of L_1 norm is done for two reasons:

1. When an optimization problem with restrictions is solved the optimal points are in the edges of the restrictions set and these edges are usually in the axes. So the optimal points are sparse solutions.
2. In the L_2 optimization problem the L_2 norm ($\|\cdot\|_2$) is not optimized, instead of it, the squared norm is optimized ($\|\cdot\|_2^2$) this approach becomes that the small errors (errors less than one) are harmed to reduce large errors (errors greater than one). The L_1 norm avoid this problem handling errors equally (see Figure 1).

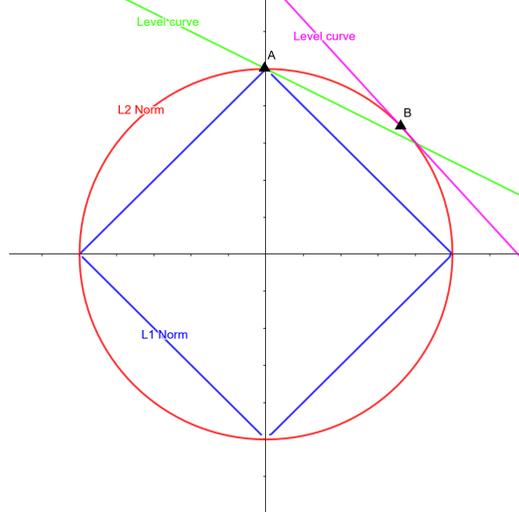


Figure 1: Optimization in L_1 and L_2 norms

This establishes several differences with the classical methods, especially finding sparse solutions.

The objective function in problem (3) is piece-wise linear function. Using this property, this problem can be written as a linear programming²² problem as follows:

$$\begin{aligned}
 & \underset{f \in \mathbb{R}^{3N}}{\text{minimize}} && c^t f \\
 & \text{Subject to:} && Hf = b \\
 & && f \geq 0.
 \end{aligned} \tag{4}$$

Where:

- N is the number of the pixels i.e. if the image has $n \times m$ size, then $N = nm$,
- the vector error e is decomposed into its positive e^+ and negative e^- parts, i.e $e = e^+ - e^-$,
- $f = (x_1, \dots, x_N, e^+_1, \dots, e^+_N, e^-_1, \dots, e^-_N)^t$, then, $f \in \mathbb{R}^{3N}$,
- $H \in \mathbb{R}^{N \times 3N}$ is a block matrix defined as

$$H = \left[\begin{array}{c|c} A_{N \times N} & I_{N \times N} \\ \hline & -I_{N \times N} \end{array} \right],$$

- $c \in \mathbb{R}^{3N}$ is the "vector cost" $c = \underbrace{(0, \dots, 0)}_{N \text{ zeros}}, \underbrace{(1, \dots, 1)}_{2N \text{ ones}}^t$.

3. METHODOLOGY

The proposed methodology is shown with the following flow chart, it is composed of four steps: down-sampling, up-sampling, restoring, and evaluation (section 4):

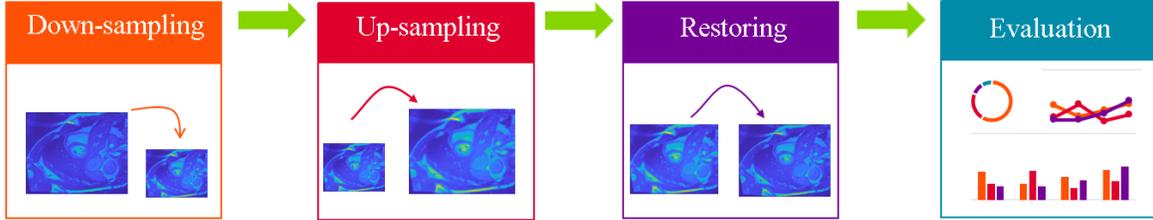


Figure 2: Pipeline for the proposed methodology

3.1 Data set

The images for the experiments were taken from Automated Cardiac Diagnosis Challenge (ACDC) a MICCAI challenge from 2017,^{23*} acquired at the University Hospital of Dijon. The dataset is composed of 150 exams (all from different patients), the method was tested with 30 cases, randomly extracted.

3.2 Down-sampling

The chosen images were down-sampled at a quarter of their original size, that is, the real images have 240×256 pixels then the sub-sampled images have 120×128 pixels. This is done with a bi-linear algorithm. The real images are used for two things: to compare the performance of the methods, and to create the images to be up-sampled, this, in summary to create the test images for the different methods.

3.3 Up-sampling

The down-sampled images are up-sampled to their original resolution; this is done for each algorithm, namely: bi-linear interpolation, bi-cubic interpolation, nearest-neighbor interpolation, Lanczos interpolation, DCC interpolation, ICBI interpolation, and INEDI interpolation. This generates a blur effect in the images as can be seen in the images in Figure 5 and motivates the usage of a deblurring method to improve the performed interpolations. At this point we have the first images to compare, these are the images that interpolation methods manage to produce.

3.4 Restoration (method application)

For this experiment, we use the up-sampled images provided by the bi-linear interpolation algorithm. The kernel is empirically estimated, that is, the PSF. In this case, it was observed that a good PSF turned out to be a Gaussian kernel with a variance of 0.87 (see Figure 3). With this, the proposed method is applied to each of the images produced by the bi-linear algorithm. To solve the minimization problem (4) an interior point algorithm is chosen⁵³. This algorithm is an iterative method proposed by Mehrotra¹⁴ and is used to handle large linear programming problems.

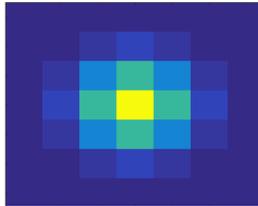


Figure 3: Gray level-scale PSF estimate

*<https://www.creatis.insa-lyon.fr/Challenge/acdc/databases.html>

4. RESULTS AND EVALUATION

Below is shown an image processed with the proposed methodology. The restorations with our method in average took 20 minutes, running in a computer with RAM of 8.0 GB and a AMD Ryzen processor:

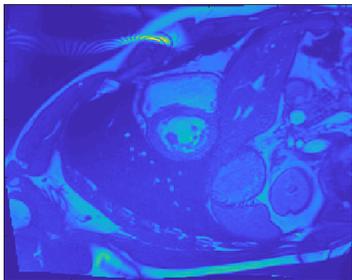


Figure 4: Real image

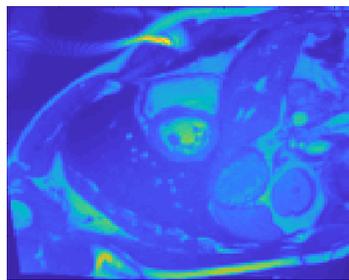


Figure 5: Down-sampled image

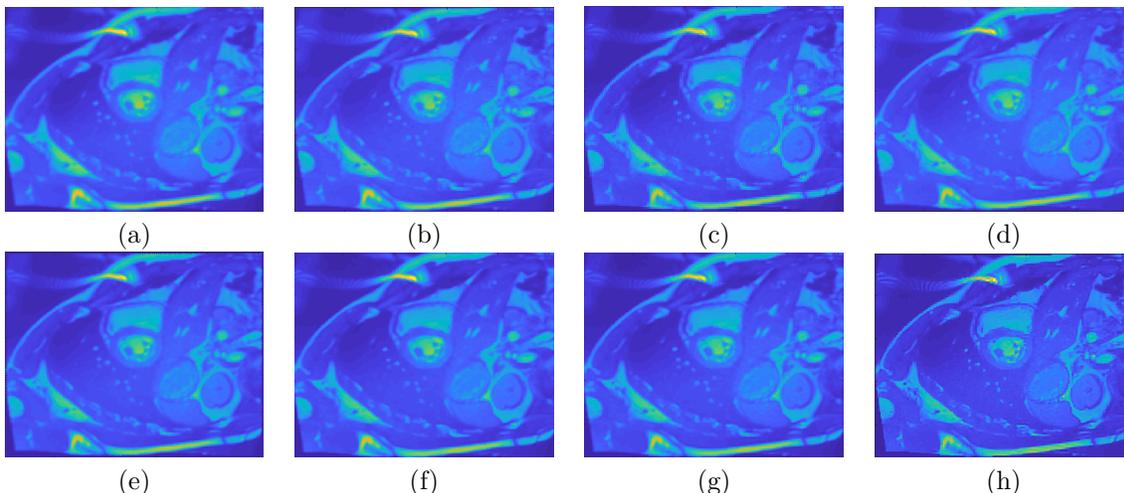


Figure 6: Results of different methods: a) Bi-linear, b) Bi-cubic, c) Nearest, d) Lanczos, e) DCC,¹⁷ f) INEDI,¹⁸ g) ICBI,¹⁵ h) Proposed

The images obtained from different algorithms and the restored images are compared with the original image to quantify the improvement of the method. This is done using three measures: PSNR, SSIM, and L_1 error.

The images are normalized to $[0, 1]$ values and the restorations are evaluated using a relative error in the matrix norm (like a ‘‘Frobenius’’ norm⁷). Let A a matrix with a_{ij} entries, then, $\hat{L}_1(A) = \sum_{i=1}^n \sum_{j=1}^m |a_{ij}|$. When A is an image that will be compared to the real image R (ground truth) the relative error is defined by $Err(A, R) = 100 \frac{\hat{L}_1(A-R)}{\hat{L}_1(R)} \%$; this error allows to measure all errors handled equally across the whole image. In addition the images are compared using the PSNR (Peak Signal-to-Noise Ratio) measure defined by $PSNR(A, B) = 10 \cdot \log_{10} \left(\frac{Max_B}{MSE} \right) dB$ where $Max_B = \max_{1 \leq i, j \leq n, m} b_{ij}$ and $MSE = \frac{1}{nm} \sum_{i=1}^n \sum_{j=1}^m \|a_{ij} - b_{ij}\|^2$, this measure captures how reliable is the image restoration. Also is used the Structural Similarity index (SSIM) to measure the structural information, in this case we are measuring how much the structure of the image is preserved.

To make the graphical presentation of the results easily readable, two groups are formed by the aforementioned interpolation methods. The first group consists of: the bi-cubic, bi-linear, nearest-neighbor, Lanczos algorithms, and our proposed method. The second group consists of the DDC, ICBI, INEDI algorithms, and our proposed method. Finally, Figures 7, 8, 9, 10, 11, 12 and Tables 1 and 2 present the results of the thirty experiments.

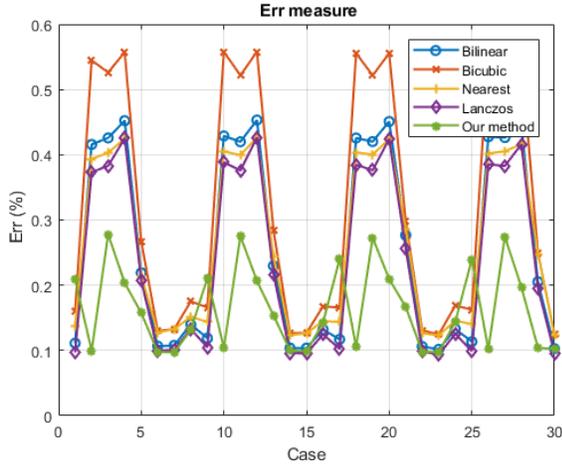


Figure 7: ERR measure comparison group 1.

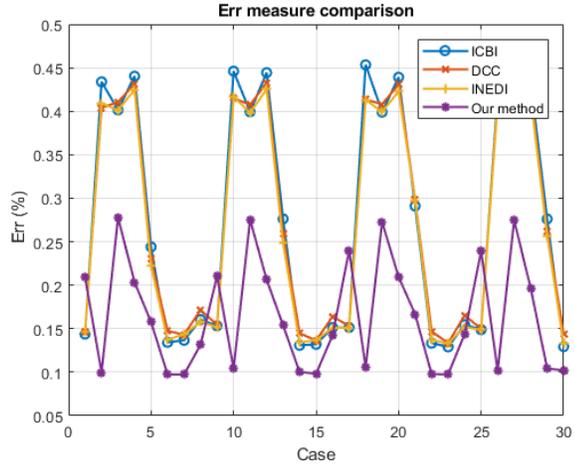


Figure 8: ERR measure comparison group 2.

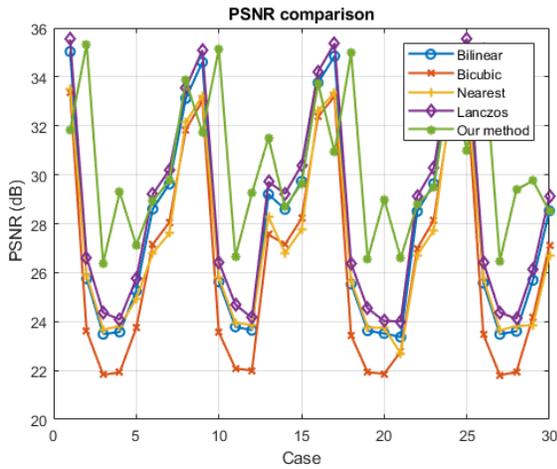


Figure 9: PSNR comparison group 1.

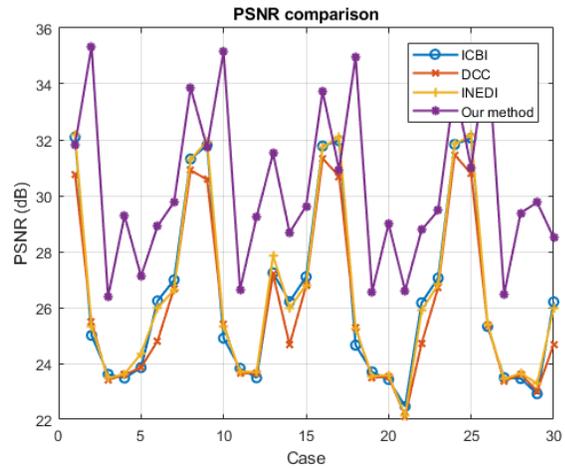


Figure 10: PSNR comparison group 2.

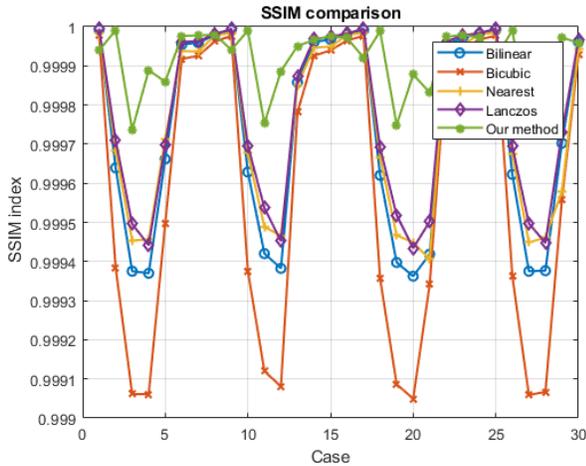


Figure 11: SSIM index comparison group 1.

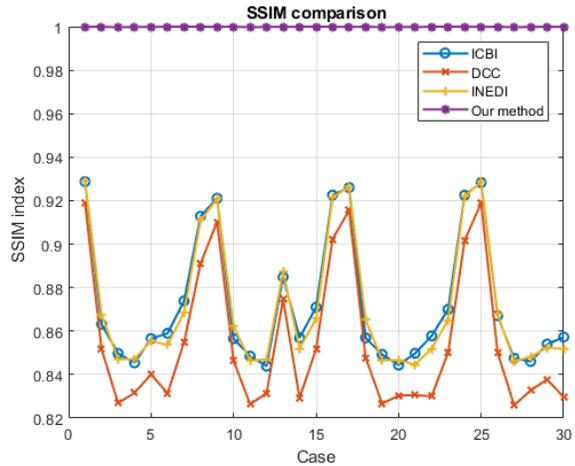


Figure 12: SSIM index comparison group 2.

Method	Measure		
	PSNR (dB)	SSIM (SSIM index)	Err (%)
Bi-linear	26.34 (4.2)	0.9996 (3.8×10^{-4})	32.30 (18.9)
Bi-cubic	27.94 (4.2)	0.9997 (2.6×10^{-4})	25.70 (15.1)
Nearest	27.14 (3.7)	0.9997 (2.2×10^{-4})	25.90 (12.9)
Lanczos	28.56 (4.1)	0.9998 (2.2×10^{-4})	23.60 (13.8)
DCC¹⁷	26.05 (2.9)	0.8549 (3.2×10^{-3})	27.18 (12.6)
ICBI¹⁵	26.45 (3.3)	0.8724 (3.0×10^{-3})	27.30 (13.5)
INEDI¹⁸	26.50 (3.3)	0.8715 (3.4×10^{-3})	26.50 (12.7)
Proposed	30.31 (2.8)	0.9999 (8.1×10^{-5})	16.39 (6.4)

Table 1: Mean and standard deviation of the experiments

Method	P-value		
	PSNR	SSIM	Err
Bi-linear	7.7×10^{-5}	9.5×10^{-4}	5.4×10^{-5}
Bi-cubic	1.0×10^{-2}	4.0×10^{-2}	3.0×10^{-3}
Nearest	3.9×10^{-4}	1.0×10^{-2}	6.4×10^{-4}
Lanczos	6.0×10^{-2}	7.0×10^{-2}	1.0×10^{-2}
DCC¹⁷	5.2×10^{-4}	3.0×10^{-4}	1.1×10^{-4}
ICBI¹⁵	5.1×10^{-4}	3.0×10^{-4}	1.8×10^{-4}
INEDI¹⁸	1.1×10^{-4}	3.0×10^{-4}	2.4×10^{-4}

Table 2: Statistical relevance of the results. Bold numbers indicate that there is no statistical difference with p value of 0.05

5. DISCUSSION

The results show important improvements (approximately 70% of the cases) in all measures (*Err*, *SSIM* and *PSNR*) comparing with up-sampled images and the restored images (our method applied to the images) with the real images. The proposed method can recover the edges of the images. This is because the L_1 norm allows larger errors than the L_2 norm, that is, it can highlight the blurred details in the interpolated image. However, there is the need to be careful with this propriety in order of not over highlighting the edges.

As said before the image restoration is very sensitive to the noise, the following example is obtained using an image from the data set. This image has the same conditions with the proposed methodology, only that noise which follows a normal distribution was added to this image in each entry; 5%[†] noise was added.

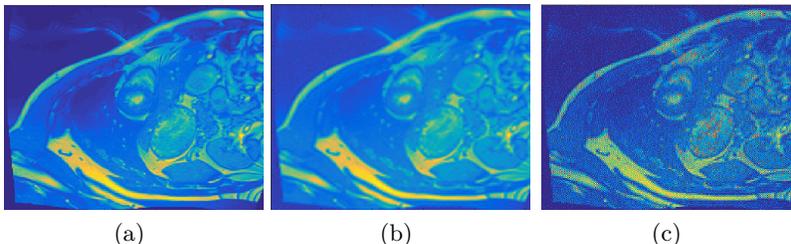


Figure 13: Noisy restoration. a) Real image, b) Blurred image with noise (re-sized image), c) Restored image

This is caused by the ill-conditioned matrix A from (2) and this effect can be avoided by using regularization in the optimization problem. Also, it is necessary to be careful when estimating the PSF because it produces

[†]In *ERR* measure

similar results to those obtained when noise is added, actually, in certain sense this can be considered as noise too.

This method has the problem of high computational cost that grows quite fast, even though it has polynomial behavior. For example, for these images H have a size of 61.140×184.320 that is 1.644.063 nonzero elements. If the images were four times larger, that is, 480×512 then H would have a size of 245.760×737.280 and 6.576.252 nonzero elements. The optimization problem requires considerable resources. It is noted that the major demand comes from the usage of RAM (approximately 2.5GB-2.7GB per image) this is due to the size of the systems of equations performed by the interior-point algorithm.

6. CONCLUSIONS

In this paper, an alternative method for restoring images is proposed. The method takes advantage of L_1 norm properties to handle errors. The proposed method is a way of dealing with PSF that generates ill-conditioned matrices that propagate and magnify errors. The results in improved images can be visually and numerically observed. This is a joint method composed of an interpolation method followed by a restoration method.

It is desirable to improve the performance of the method, particularly the Err measure since it plays an important role in the optimization. This can be done by getting more knowledge about the noise affecting these kind of images or by improving the PSF estimation.

Future work will be focused in reducing the computational cost at the optimization step to handle larger images. On the other hand, we will study how to estimate the PSF automatically and improve it by using only the blurred image, getting rid of previous knowledge from the user. Finally we will cover other interpolation methods with different images sizes.

7. ACKNOWLEDGMENTS

The authors would like to thank to Angélica Atehortúa, Jorge Jara and Eduardo Romero for provide us the data set and to give us ideas for the experiment.

REFERENCES

- [1] BAZARAA, M.S., J.J. JARVIS y H.D. SHERALI, *Programación lineal y flujo en redes*, segunda edición, Limusa, México, DF, 2004.
- [2] BUADES, BARTOMEU y MICHEL, M., J. , *On image denoising methods*, Marzo del 2019.
- [3] FREUND, R., *Primal-Dual Interior-Point Methods for Linear Programming based on Newton's Method*, Massachusetts Institute of Technology, Marzo del 2004.
- [4] GOLUB , G., H. y VAN LOAN, C., F. , *Matrix Computations*, edición 4, The Johns Hopkins University Press, Baltimore, 2013.
- [5] HANSEN, C.,NAGY, J., G. y O'LEARY, D., P. , *Deblurring Images, Matrices, Spectra and Filtering*, Society for Industrial and Applied Mathematics, Philadelphia, 2006.
- [6] KE CHEN, *Matrix Preconditioning Techniques and Applications*, Cambridge University Press, 2005.
- [7] KINCAID, D. y CHENEY, W., *Numerical Analysis Mathematics of scientific computing*, edición 3, American Mathematical Society, 2002.
- [8] MAMMONI, R., *Image recovery: Theory and application*, Academic Press, Nueva York, 1987, pp. 127-155.
- [9] MONTENERGO, F., *Regularización de problemas inversos e imágenes borrosas*, Universidad Nacional de Colombia, Mayo del 2010.
- [10] MOORE, H., *MATLAB para ingenieros*, Pearson Prentice Hall.
- [11] SAEED KETABCHI, HOSSEIN MOOSAEI, HOSSEIN SAHLEH y MOHAMMAD HEDAYATI, *New Methods for Solving Large Scale Linear Programming Problems in the Windows and Linux computer operating systems*, volumen 7, Appl. Math. Inf. Sci., Octubre del 2018.
- [12] SOTELO, J., C., *Álgebra lineal para estudiantes de Ingeniería y ciencias*, Mc Graw Hill, Ciudad de México, 2011.

- [13] KRESS, R., MAZ'YA, V., & KOZLOV, V., *Linear integral equations*, 1989.(Vol. 82, pp. xii+-299). Berlin: Springer.
- [14] MEHROTRA, S., *On the implementation of a primal-dual interior point method. SIAM Journal on optimization*, 1992, SIAM, 2(4), 575-601.
- [15] A. GIACHETTI AND N. ASUNI, *Fast Artifacts-Free Image Interpolation*, Br. Mach. Vis. Conf. Leeds UK, pp. 123–132, 2008.
- [16] N. ASUNI AND A. GIACHETTI, *Accuracy improvements and artifacts removal in edge based image interpolation*, Proc 3rd Int Conf Comput Vis Theory Appl. VISAPP08, 2008.
- [17] D. ZHOU, X. SHEN, AND W. DONG, *Image zooming using directional cubic convolution interpolation*, IET Image Process., vol. 6, no. 6, pp. 627 – 634, 2012.
- [18] X. LI AND M. T. ORCHARD, *New edge-directed interpolation*, IEEE Trans. Image Process., vol. 10, no. 10, pp. 1521 – 1527, 2001.
- [19] JUELIN LENG, GUOLIANG XU, AND ,YONGJIE ZHANG, *Medical image interpolation based on multi-resolution registration* Computers & Mathematics with Applications, 66(1), 1-18. 2013.
- [20] OUSGUINE, S., ESSANNOUNI, F., ESSANNOUNI, L., AND ABOUTAJDINE, D., *A new image interpolation using gradient-orientation and cubic spline interpolation*. International Journal of Innovation and Applied Studies, 5(3), 215. 2014.
- [21] KEYS, ROBERT, *Cubic convolution interpolation for digital image processing*. IEEE transactions on acoustics, speech, and signal processing 29.6 (1981): 1153-1160.
- [22] BOYD, STEPHEN, AND LIEVEN VANDENBERGHE, *Convex optimization*. Cambridge university press, 2004.
- [23] O. BERNARD, A. LALANDE, C. ZOTTI, F. CERVENANSKY, ET AL. *Deep Learning Techniques for Automatic MRI Cardiac Multi-structures Segmentation and Diagnosis: Is the Problem Solved ?* in IEEE Transactions on Medical Imaging, vol. 37, no. 11, pp. 2514-2525, Nov. 2018

6 Conclusiones y recomendaciones

En este último capítulo presentamos las conclusiones del trabajo realizado, así como algunas posibles recomendaciones para el trabajo y una futura extensión de éste.

6.1. Conclusiones

- En este trabajo se exploró y se probó un método alternativo para la restauración de imágenes borrosas, este método pudo ser planteado y solucionado como un problema de programación lineal tomando en cuenta las propiedades particulares de la norma $\|\cdot\|_1$. Esto como una forma de lidiar con aquellas matrices mal condicionadas que provienen de la ecuación integral (3-1).
- Fue posible constatar que los algoritmos de punto interior son mejores que el método simplex para la resolución de estos problemas, mostrando que consumen menos recursos (memoria RAM, procesador, etc.) y son más manejables que el método clásico dado por este algoritmo.
- Se adaptó el modelo original (3-27), en el que aparecía ruido artificial en las soluciones, de acuerdo a los resultados de los experimentos realizados al introducir términos de regularización con diferentes imágenes de forma que se lograron mejoras cualitativas y cuantitativas. Particularmente con estos términos es posible eliminar ese ruido artificialmente creado. También fue posible adaptar el problema al caso particular de restauración de imágenes reduciendo el consumo de recursos de memoria y mejorando la velocidad de cómputo.
- Se comprobó que el método propuesto es una alternativa viable y eficaz con respecto a otras propuestas hechas por otros autores en el problema de restauración de imágenes. También, que es posible su uso y aplicación en diferentes campos. Un ejemplo de esto son los resultados mostrados en problemas de imágenes médicas.

6.2. Recomendaciones

- Es necesario profundizar en técnicas que permitan resolver grandes sistemas lineales de forma implícita, dado que estas soluciones juegan un papel crucial en la solución del problema de optimización. Esto incluye buscar preconditionadores para esta clase de matrices para ser usados en conjunto con estas técnicas.
- El hecho que el método propuesto se desmarca de las aproximaciones usuales hace que el problema de optimización sea más costoso en términos de tiempo principalmente. Se debe buscar cómo adaptar el método de punto interior para esta tarea específica de restauración de imágenes de forma que sea un método más eficiente.
- Algunos parámetros como las constantes α y γ así como la PSF k son escogidos en principio de acuerdo a información previa. Para escogerlos de forma automática y eficaz es necesario profundizar en métodos como la validación cruzada generalizada (expuesta en [34] y [16]) para la búsqueda de las constantes. Así mismo, es necesario desarrollar una herramienta que permita escoger de buena manera esta PSF al realizar un proceso de deconvolución ciega (blind deconvolution) donde la PSF es desconocida.

Bibliografía

- [1] *Restauración de ultrasonido*. <https://google.com.com/ultrasonido/>, 2019.
- [2] M. S. Bazaraa, J. J. Jarvis, and H. D. Sherali. *Linear programming and network flows*. John Wiley & Sons, 2011.
- [3] D. S. Biggs and M. Andrews. Acceleration of iterative image restoration algorithms. *Applied optics*, 36(8):1766–1775, 1997.
- [4] S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge university press, 2004.
- [5] A. Buades, B. Coll, and J. M. Morel. On image denoising methods. *CMLA Preprint*, 5, 2004.
- [6] Canal1.com. *Video de cámara de seguridad*. <https://canal1.com.co/entretenimiento/virales/video-camara-de-seguridad-agresion-pareja-gay-andino/>, 2019.
- [7] K. Chen. *Matrix preconditioning techniques and applications*, volume 19. Cambridge University Press, 2005.
- [8] J. G. N. Christian Hansen and D. P. O’Leary. <http://www.imm.dtu.dk/pcha/hno/>, 2011.
- [9] M. B. Cohen, Y. T. Lee, and Z. Song. Solving linear programs in the current matrix multiplication time. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, pages 938–942. ACM, 2019.
- [10] J. C. del Valle Sotelo. *Álgebra lineal para estudiantes de Ingeniería y ciencias*. Mc Graw Hill, 2011.
- [11] flickr.com. *Placa de auto*. from <https://www.flickr.com/photos/pattoncito/2471478157>, 2019.
- [12] R. M. Freund. *Primal-Dual Interior-Point Methods for Linear Programming based on Newton’s Method*. Massachusetts Institute of Technology, 2004.
- [13] R. Gonzalez, R. Woods, and S. Eddins. *Digital Image Processing Using MATLAB*. McGraw Hill Education, 2013.

-
- [14] W. Hackbusch. *Iterative solution of large sparse systems of equations*, volume 95. Springer, 1994.
- [15] R. J. Hanisch, R. L. White, and R. L. Gilliland. Deconvolution of hubbles space telescope images and spectra. In *Deconvolution of images and spectra (2nd ed.)*, pages 310–360. 1996.
- [16] P. C. Hansen, J. G. Nagy, and D. P. O’leary. *Deblurring images: matrices, spectra, and filtering*, volume 3. Siam, 2006.
- [17] W. Harwood. *Fixing Hubble’s blurry vision*. <https://spaceflightnow.com/2015/04/23/fixing-hubbles-blurry-vision/>, 2015.
- [18] R. A. Hummel, B. Kimia, and S. W. Zucker. Deblurring gaussian blur. *Computer Vision, Graphics, and Image Processing*, 38(1):66–80, 1987.
- [19] infoagro.com. *El cultivo de la Orquídea*. <https://www.infoagro.com/documentos/el-cultivo-orquidea.asp>.
- [20] iredbubble.com. *Diosa Aqua*. <https://www.redbubble.com/3790348-chibi-llorando-aqua?p=sticker>, 2019.
- [21] M. Jin, S. Roth, and P. Favaro. Normalized blind deconvolution. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 668–684, 2018.
- [22] S. L. Kala. Deblurring images via partial differential equations. *Proceedings of the Louisiana-Mississippi Section of the Mathematical Association of America*, 2004.
- [23] D. Kincaid and W. Cheney. *Análisis numérico las matemáticas del cálculo científico*. Adison-Wesley, 1994.
- [24] V. Klee and G. J. Minty. How good is the simplex algorithm? Technical report, Washington university Seattle departament of Mathematics, 1970.
- [25] R. Kress. Numerical analysis. *Academic Edition*, 1, 1998.
- [26] R. Kress, V. Maz’ya, and V. Kozlov. *Linear integral equations*, volume 82. Springer, 1989.
- [27] D. Krishnan. <https://dilipkay.wordpress.com/blind-deconvolution/>, 2011.
- [28] D. Krishnan, T. Tay, and R. Fergus. Blind deconvolution using a normalized sparsity measure. In *CVPR 2011*, pages 233–240. IEEE, 2011.
- [29] R. Mammone and G. Eichmann. Superresolving image restoration using linear programming. *Applied optics*, 21(3):496–501, 1982.

-
- [30] R. Mammoni. *Image recovery: Theory and application*. Academic Press, 1987.
- [31] I. Mantilla. *Análisis Numérico*. Universidad Nacional de Colombia. Bogotá, 2004.
- [32] S. Mehrotra. On the implementation of a primal-dual interior point method. *SIAM Journal on optimization*, 2(4):575–601, 1992.
- [33] A. Moffat. A theoretical investigation of focal stellar images in the photographic emulsion and application to photographic photometry. *Astronomy and Astrophysics*, 3:455, 1969.
- [34] A. F. Montenegro. Regularización de problemas inversos e imágenes borrosas. Master's thesis, Universidad Nacional de Colombia, 2010.
- [35] H. Moore, V. C. Olguín, and R. M. Nuño. *Matlab para ingenieros*. Number 620.0013 M66 2007. Pearson Educación, 2007.
- [36] NASA, ESA, the Hubble Heritage Team (STScI/AURA)-ESA/Hubble Collaboration and K. Noll (STScI). *NGC 6050*. <https://www.spacetelescope.org/images/heic0810ap/>, 2008.
- [37] W. H. Richardson. Bayesian-based iterative method of image restoration. *JoSA*, 62(1):55–59, 1972.
- [38] H. S. Saeed Ketabchi, Hossein Moosaei and M. Hedayati. New methods for solving large scale linear programming problems in the windows and linux computer operating systems. *Appl. Math. Inf. Sci.*, 7, Octubre 2018.
- [39] Tucarro.com. *Placa de automóvil*. <https://tucarro.com/>, 2018.
- [40] C. F. Van Loan and G. H. Golub. *Matrix computations*. Johns Hopkins University Press, 1983.
- [41] Vicmat.com. *La Conjetura de Nicomaco*. <https://vicmat.com/la-conjetura-nicomaco-cubo-diferencia-dos-cuadrados/>, 2017.
- [42] Z. Wang, A. C. Bovik, H. R. Sheikh, E. P. Simoncelli, et al. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.
- [43] Wikipedia.org. *Argumento diagonal de Cantor*. <https://es.wikipedia.org/wiki/Argumento-de-la-diagonal-de-Cantor>, 2019.
- [44] M. Zaharescu and C. A. Boiangiu. Image deblurring: challenges and solutions. *The 12th International Conference on Circuits, Systems, Electronics, Control & Signal Processing (CSECS '13)*, 9:10, 2013.