



PONTIFICIA UNIVERSIDAD JAVERIANA
Bogotá, Colombia

TRABAJO DE GRADO

RECONSTRUCCIÓN DE CORTES AXIALES CON MUESTREO COMPRIMIDO

Autor:
Johan Sebastián Peña Campos

Director:
Cesar L. Niño, Ph.D.

Jun 19, 2015

Artículo 23 de la resolución No. 13 de junio de 1946

“La universidad no se hace responsable por los conceptos emitidos por sus alumnos en sus trabajos de tesis. Solo velará porque no se publique nada contrario al dogma y la moral católica y porque las tesis no contengan ataques personales contra persona alguna, antes bien se vea en ellas el anhelo por buscar verdad y justicia”

CONTENTS

Introducción	4
1 RECONSTRUCCIÓN POR MEDIO DEL MÉTODO ALGEBRAICO	5
Método Algebraico	5
Geometría en paralelo	7
2 RECONSTRUCCIÓN USANDO EL MÉTODO DE MUESTREO COMPRIMIDO	15
Muestreo Comprimido	15
Algoritmo de reconstrucción Iteraciones Recursivas de Mínimos Cuadrados con norma p	15
3 RESULTADOS	18
3.1 Vector Aleatorio	18
3.2 Imagen sintética binaria	19
3.3 Imagen sintética	20
3.4 Probabilidad de Reconstrucción	21
4 CONCLUSIONES	24
Lista de Figuras	25
Lista de Tablas	26
A IRLS PARA NORMA P	28
B DESARROLLO E IMPLEMENTACIÓN DE SOFTWARE	30

INTRODUCCIÓN

La tomografía computarizada (en inglés Computer Tomography, CT) es la técnica más usada en la adquisición de imágenes médicas, esta técnica está fundamentada en la toma de radiografías. Las radiografías siguen el principio físico donde el rayo, generado por emisor de rayos X, atraviesa el cuerpo, o el objeto irradiado, y al llegar al receptor, los haces de rayos X tienen distintas intensidades debido a las densidades de las estructuras traspasadas [1].

La adquisición de imágenes médicas es una de las áreas que más adelantos ha tenido dentro de la medicina, lo que ha desencadenado, entre otras, un diagnóstico temprano de enfermedades. Una de las condiciones indispensables para lograr dichos adelantos ha sido la interdisciplinariedad a través de la investigación por áreas del conocimiento ajenas a la medicina. Los procesos de investigación siempre han buscado mejorar la calidad de vida del paciente y están fundamentados bajo uno de los axiomas más importantes: desarrollar imágenes médicas a través de métodos no invasivos [2].

Por otro lado, dicha técnica ha permitido que el campo de investigación sobre la misma se amplíe considerablemente, teniendo como objetivo eliminar el mayor problema de la radiografía convencional: la superposición de estructuras internas. Es por esto que se reconstruyen cortes axiales, ya que se puede analizar desde distintas vistas; en la vista frontal, por ejemplo, se tendría el mismo inconveniente que en la radiografía convencional, en cambio en la vista lateral se puede apreciar distintas estructuras que son atravesadas por el rayo.

Aunque la técnica ha tenido un auge importante y cada vez mayor con el paso del tiempo, estudios han demostrado que someter a los pacientes a esta radiación es un factor de riesgo considerable debido a que la radiación recibida por medio de un escaneo CT es de 100 a 1000 veces mayor que los rayos X convencionales, lo que conlleva a un inminente riesgo de cáncer dada la cantidad de células comprometidas durante el proceso [3]. En la presente implementación se integrará el uso del muestreo comprimido usando el código de optimización Iteración Recursiva de Mínimos Cuadrados de norma p (en inglés Iterative Recursive Least Square, IRLS) con el objetivo de reconstruir un corte axial con buena resolución exponiendo al paciente a una menor radiación reduciendo el riesgo anteriormente mencionado [4] y obteniendo una alta probabilidad de reconstrucción perfecta con un número muy bajo de iteraciones.

 RECONSTRUCCIÓN POR MEDIO DEL MÉTODO ALGEBRAICO

MÉTODO ALGEBRAICO

El método algebraico para la reconstrucción de imágenes en CT representa el problema de reconstrucción como un problema de ecuaciones lineales, ya que discretiza el problema de las proyecciones basado en el modelo discreto del arreglo emisor y de los píxeles dentro del campo de visión. Es decir, es un arreglo discreto de variables desconocidas llamadas coeficientes de atenuación f_j donde $j = \{1, \dots, N\}$. Dichos coeficientes deben ser calculados por medio de la relación física entre la intensidad del rayo que llega a los receptores y la atenuación de la intensidad del rayo generada por estos coeficientes, como se ilustra en la Figura 1. El sistema de ecuaciones que se debe plantear se ilustrará en la Ecuación 1.

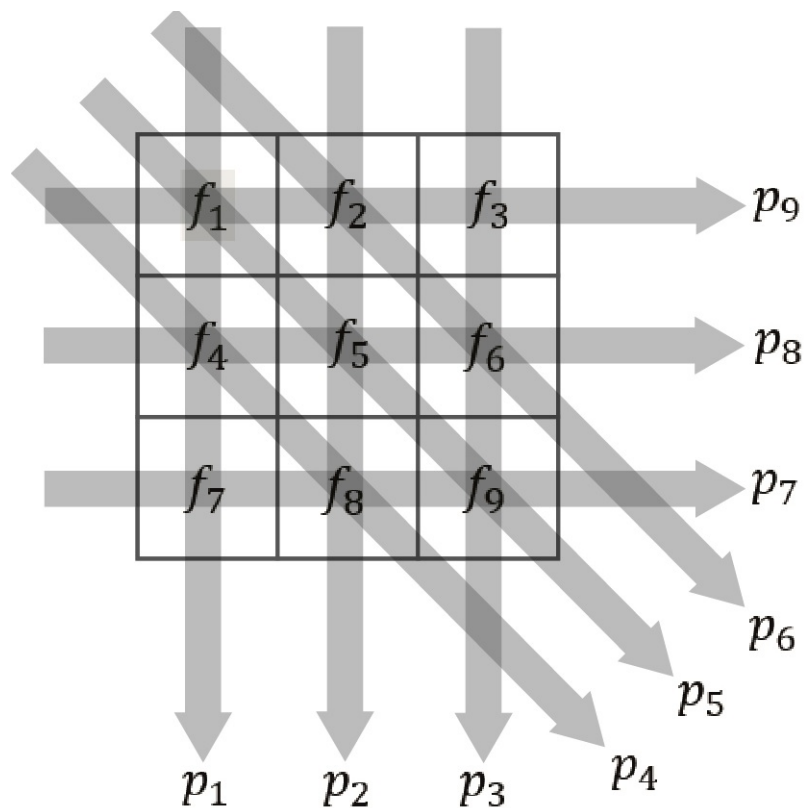


Figure 1.: Reconstrucción de imagen por medio del método algebraico: Las ecuaciones lineales usadas para reconstruir están dadas por nueve proyecciones compuestas por tres ángulos y tres rayos, de esta manera se obtiene un sistema de ecuaciones de 9×9

$$\begin{aligned}
 f_1 + f_4 + f_7 &= p_1 \\
 f_2 + f_5 + f_8 &= p_2 \\
 &\vdots \\
 f_1 + f_5 + f_9 &= p_5 \\
 &\vdots \\
 f_1 + f_2 + f_3 &= p_9
 \end{aligned} \tag{1}$$

Para completar el modelo es necesario tener en cuenta el ancho del rayo, lo que genera una variable adicional. Esta variable representa cuánto del píxel que se esta reconstruyendo es atravesado por el rayo, es decir cuanta área ilumina el rayo dentro del píxel.

$$a_{ij} = \frac{A_r \text{ del píxel } j \text{ por el rayo } i}{A_T \text{ del píxel } j} \tag{2}$$

Donde a_{ij} esta dentro del intervalo $[0, 1]$ y representa cual es el área radiada del píxel j por un rayo i que tiene un ancho $\Delta\xi$. Para este caso en particular el área total del píxel esta dada por b^2 como se ilustra en la Figura 2. De esta manera obtenemos la Ecuación general 3 para el sistema de ecuaciones, en la cual N representa

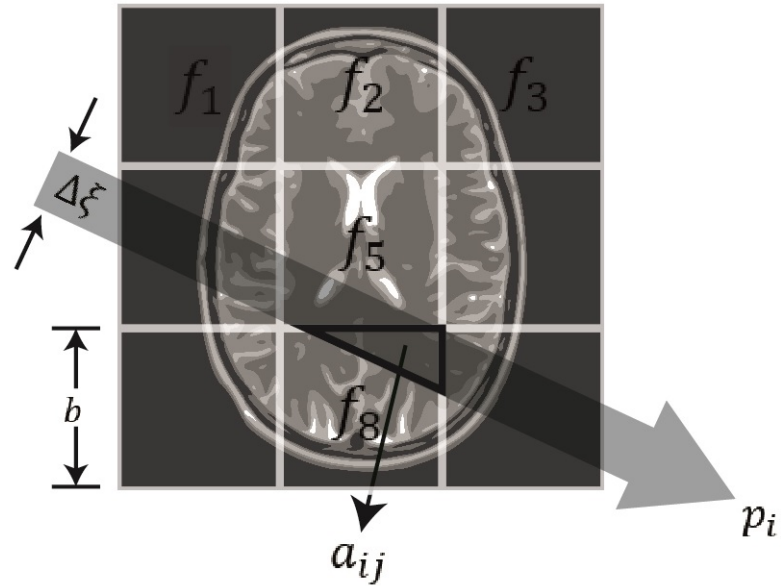


Figure 2.: El rayo X de ancho $\Delta\xi$ no atraviesa toda el área del píxel (b^2). La sección de área atravesada por dicho rayo es a_{ij} para el píxel f_8

el número de píxeles que deben ser reconstruidos e $i = \{1, \dots, M\}$ es el índice de las proyecciones, siendo $M = N - Theta \times N - Sen$ donde $N - Sen$ representa el número de detectores por proyección y $N - 2Theta$ el número de proyecciones.

$$\left[\sum_{j=1}^N a_{ij} f_j = p_i \right]_{i=1}^M \tag{3}$$

Escribiendo todas las proyecciones $\mathbf{p} = (p_1, \dots, p_M)^T$ y los coeficientes de atenuación que serán reconstruidos $\mathbf{f} = (f_1, \dots, f_N)^T$, y representando la matriz de área como una matriz $M \times N$ [5].

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1N} \\ a_{21} & \ddots & & a_{2N} \\ \vdots & & \ddots & \vdots \\ a_{M1} & a_{M2} & \cdots & a_{MN} \end{bmatrix}$$

(4)

El sistema de ecuaciones se representará en la Ecuación 5.

$$\mathbf{p} = \mathbf{A}\mathbf{f}$$

(5)

Geometría en paralelo

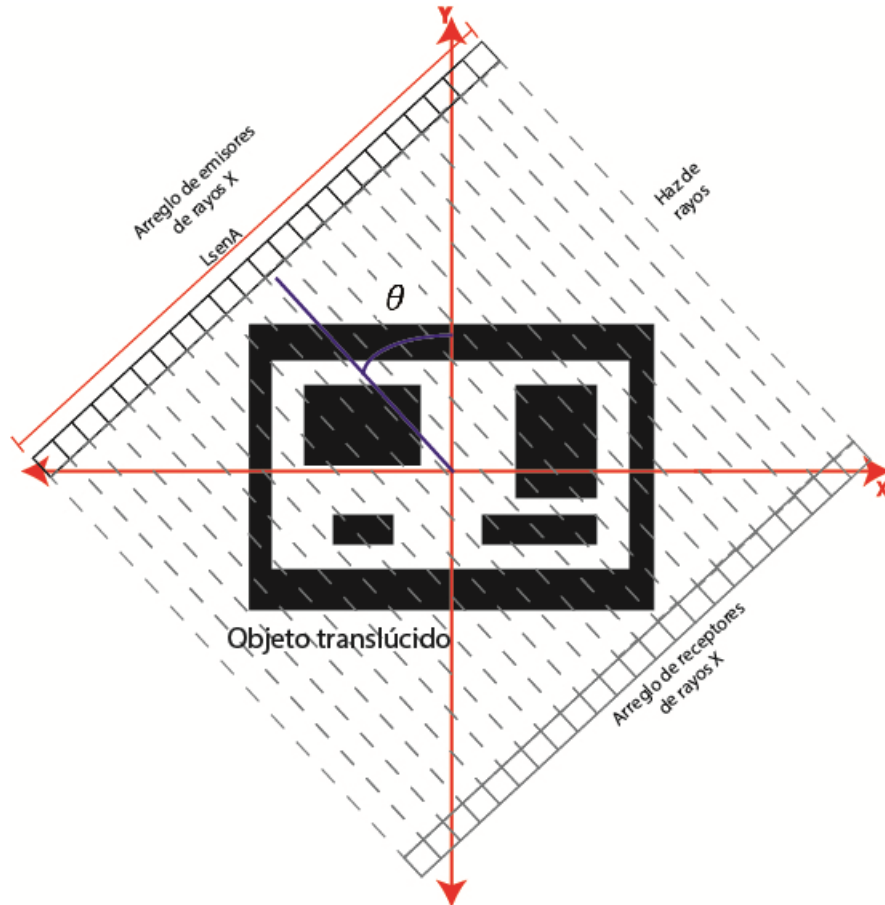


Figure 3.: Geometría paralela de un cuerpo translúcido a θ , se especifican los parámetros de entrada L_{senA} , N_{Sen} y N_{Theta} los cuales hacen referencia a la longitud de los arreglos emisor y receptor, la cantidad de sensores por arreglo y la cantidad de ángulos sobre los que se quiere proyectar

En la DLL se generan las proyecciones del objeto translúcido por medio de una radiación en paralelo usando el método algebraico. Esta geometría implica que el arreglo de emisores y el arreglo de receptores

se encuentran ubicados en paralelo, y rotan a lo largo de una circunferencia que tiene su centro en la mitad del objeto. Dicha rotación está dada por un ángulo θ medido desde el eje y , como se muestra en la Figura 3, dicho ángulo aumenta positivamente en sentido horario [6].

Dada la geometría y las condiciones anteriormente descritas, las áreas generadas entre los haces y las celdas se evaluaron por medio de 17 casos. Dichos casos se clasifican según el ángulo de incidencia de los rayos y según los límites de incidencia de los mismos en la celda.

Para este análisis se modelan los rayos como líneas rectas, dichas rectas tienen ecuaciones definidas por los siguientes parámetros:

$$m = \frac{\Delta y}{\Delta x} = \frac{\frac{L}{2} \cos(\theta)}{\frac{L}{2} \sin(\theta)} = \frac{\cos(\theta)}{\sin(\theta)} = \cot(\theta) \quad (6)$$

$$b \rightarrow \cos(\alpha) = \frac{L_{dif}}{h} \rightarrow b = \frac{L_{dif}}{\cos(\alpha)} = \frac{L_{dif}}{\cos(90 - \theta)} \quad (7)$$

Donde m es la pendiente del haz modelado como una línea recta y b es el DC específico para calcular el área.

$$f(x) = \cot(\theta)x + N_i b \quad (8)$$

Donde N_i es el índice correspondiente al haz que se está evaluando.

$$\begin{aligned} \int_{x_a}^{x_b} f(x) dx &= \int_{x_a}^{x_b} (\cot(\theta)x + N_i b) dx \\ &= \cot(\theta) \left[\frac{x^2}{2} \right]_{x_a}^{x_b} + N_i b [x]_{x_a}^{x_b} \end{aligned} \quad (9)$$

El resultado es el área bajo la línea hasta el nivel de DC establecido, desde el punto x_a hasta el punto x_b .

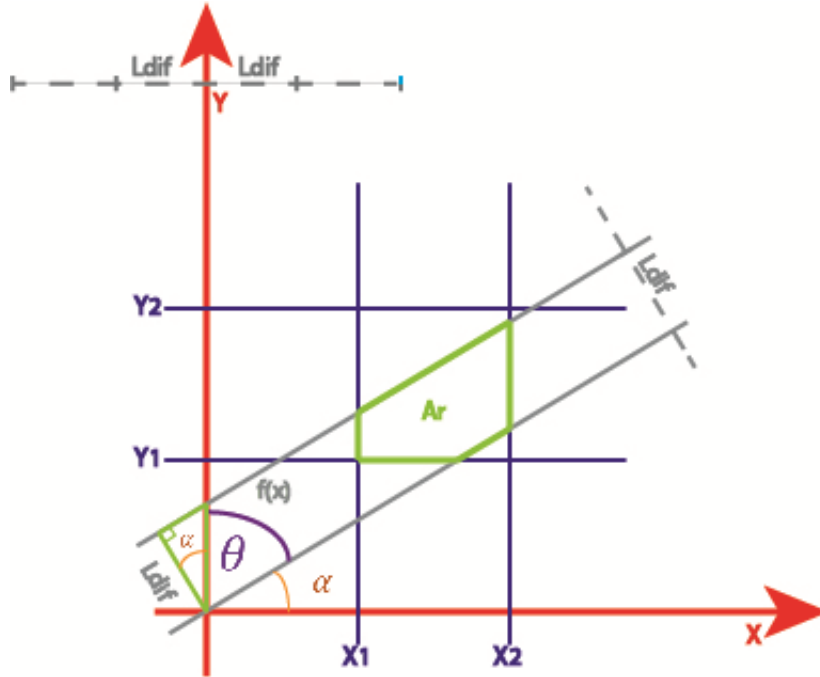


Figure 4.: Se modela el par de haces como líneas rectas perpendiculares donde se genera un área limitada dentro de la celda, los haces inciden a un ángulo θ , las esquinas de la celda son x_1, x_2, y_1, y_2 , la distancia entre rayos es L_{dif}

1. Si $\theta = -90$:

Es un caso particular en la DLL ya que las áreas siempre describen un rectángulo que esta dado por los extremos verticales de la celda y la altura depende de $Ldif = \frac{L \cdot \text{sen} \theta}{N}$. Se evalúa dichas áreas según el recorrido de los dos rayos al pasar por la celda evaluada como se ilustra en la Figura 5.

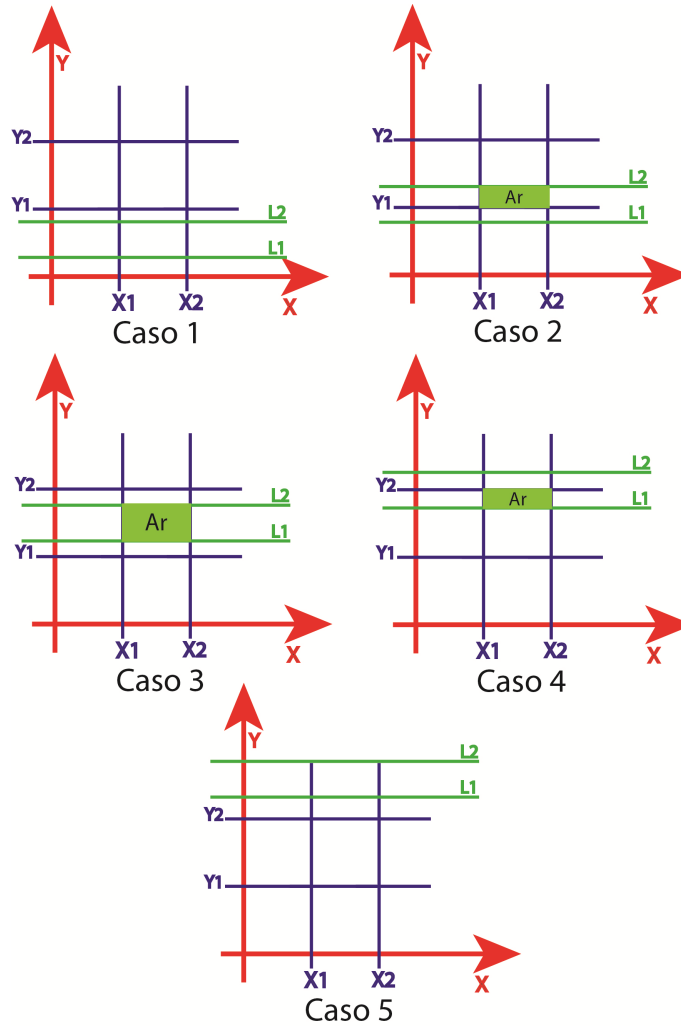


Figure 5.: Para un ángulo $\theta = -90$ se generan 5 casos

- Caso 1: Ninguno de los dos rayos atraviesa la celda evaluada ya que se encuentran por debajo de ella, en este caso el $A_r = 0$.
- Caso 2: L_2 atraviesa la celda evaluada pero L_1 no, en este caso el área resultante es un rectángulo donde su base está determinada por los extremos de la celda y la altura depende de la incidencia de L_2

$$A_r = (x_2 - x_1) * ((Ldif * n_b) - y_1)$$
- Caso 3: L_1 y L_2 atraviesan la celda evaluada, en este caso el área resultante es un rectángulo donde su base está determinada por los extremos de la celda y la altura depende de la incidencia de L_2 y L_1

$$A_r = (x_2 - x_1) * (Ldif * (n_b - n_a))$$
- Caso 4: L_1 atraviesa la celda evaluada pero L_2 no, en este caso el área resultante es un rectángulo donde su base está determinada por los extremos de la celda y la altura depende de la incidencia

de L_1

$$A_r = (x_2 - x_1) * (y_2 - (Ldif * n_a))$$

- e) Caso 5: Ninguno de los dos rayos atraviesa la celda evaluada ya que se encuentran por encima de ella, en este caso el $A_r = 0$.

2. Si $\theta > -90$ && $\theta < 0$:

Para este caso el algoritmo de la DLL calcula ocho puntos de intersección, cuatro puntos definidos para el primer haz de rayos y cuatro para el segundo haz:

$$\begin{aligned} P_{1-L1}(x_1, ya_1), & P_{2-L1}(xa_1, y_1), \\ P_{3-L1}(x_2, ya_2), & P_{4-L1}(xa_2, y_2), \\ P_{5-L2}(x_1, yb_1), & P_{6-L2}(xb_1, y_1), \\ P_{7-L2}(x_2, yb_2), & P_{8-L2}(xb_2, y_2) \end{aligned} \quad (10)$$

Como se muestra en la Figura 6, estos puntos se

usan para definir si los haces ($L1 \& L2$) intersecan la celda evaluada, para lo cual se tienen tres casos específicos:

- a) $L1$ atraviesa la celda pero $L2$ no:

Por lo menos dos de los puntos definidos para el rayo $L1$ en la Ecuación 10 se encuentran dentro de la celda; en este caso podrían suceder cuatro variantes que determinan los limites a evaluar en la integral como se puede apreciar en la Figura 7

- i. Caso 1: Se calcula la integral de $L1$ entre x_1 y x_2 , y se le resta el área total de la celda.
- ii. Caso 2: Se calcula la integral de $L1$ entre x_1 y xa_1 , y se le resta el área total de la celda.
- iii. Caso 3: Se calcula la integral de $L1$ entre x_1 y xa_1 , y la integral de $L1$ entre x_1 y xa_2 . A la primera integral calculada se le resta la segunda y se le resta el área total de la celda.
- iv. Caso 4: Se calcula la integral de $L1$ entre x_1 y x_2 , y la integral de $L1$ entre x_1 y xa_2 . A la primera integral calculada se le resta la segunda y se le resta el área total de la celda.

- b) $L2$ atraviesa la celda pero $L1$ no:

Por lo menos dos de los puntos definidos para el rayo $L2$ en la Ecuación 10 se encuentran dentro de la celda; en términos generales se tienen los mismos cuatro casos que en el caso anterior como se muestra en la Figura 8.

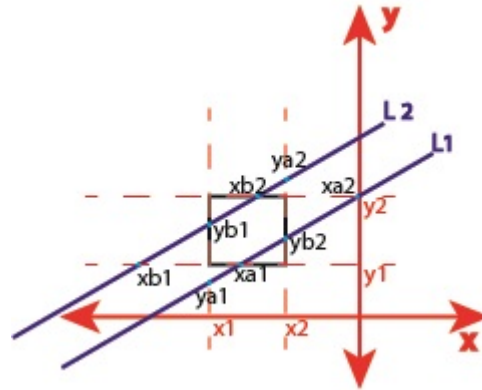


Figure 6.: Puntos de intersección de los haces del rayo con la celda, se definen 4 puntos para $L1$ y 4 puntos para $L2$

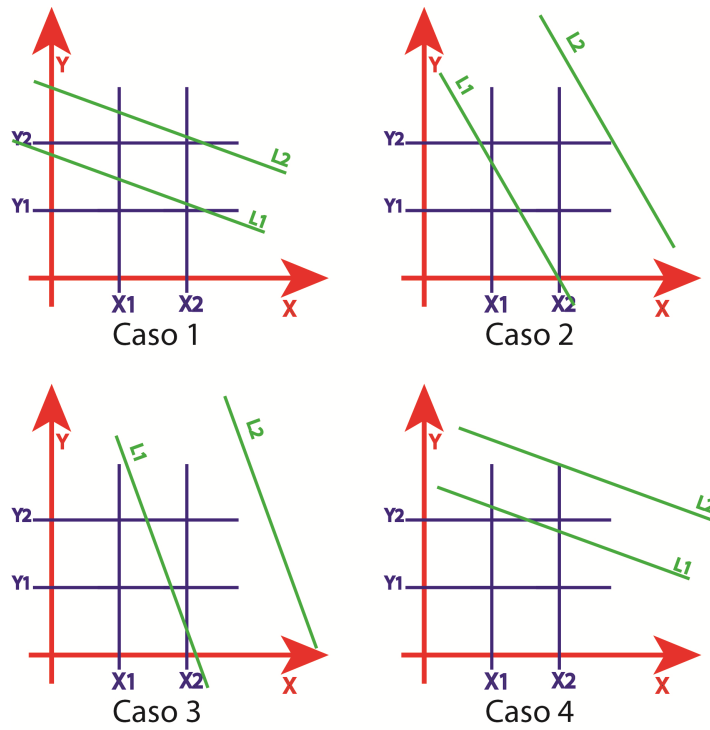


Figure 7.: Para un ángulo $-90 < \theta < 0$ y L_1 atraviesa la celda pero L_2 no, se generan 4 casos

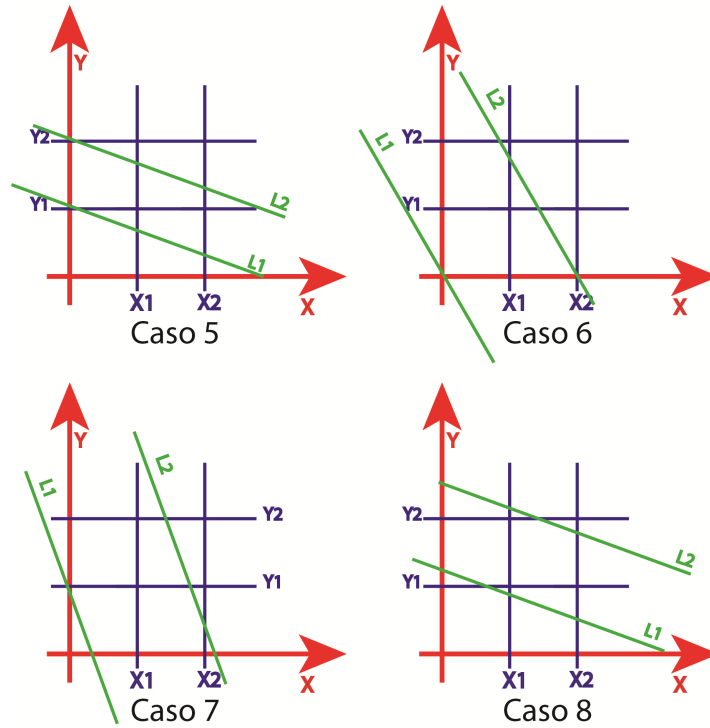


Figure 8.: Para un ángulo $-90 < \theta < 0$ y L_2 atraviesa la celda pero L_1 no, se generan 4 casos

- i. Caso 5: Se calcula la integral de L_2 entre x_1 y x_2 .
- ii. Caso 6: Se calcula la integral de L_2 entre x_1 y x_{b_1} .

iii. Caso 7: Se calcula la integral de L_2 entre x_1 y xb_1 , y la integral de L_2 entre x_1 y xb_2 . A la primera integral calculada se le resta la segunda para obtener el área deseada.

iv. Caso 8: Se calcula la integral de L_2 entre x_1 y x_2 , y la integral de L_2 entre x_1 y xb_2 . A la primera integral calculada se le resta la segunda.

c) L_1 y L_2 atraviesan la celda:

En este caso por lo menos dos de los puntos definidos para L_1 y dos de los puntos definidos para L_2 en la Ecuación 10 se encuentran dentro de la celda como se ilustra en la Figura 9

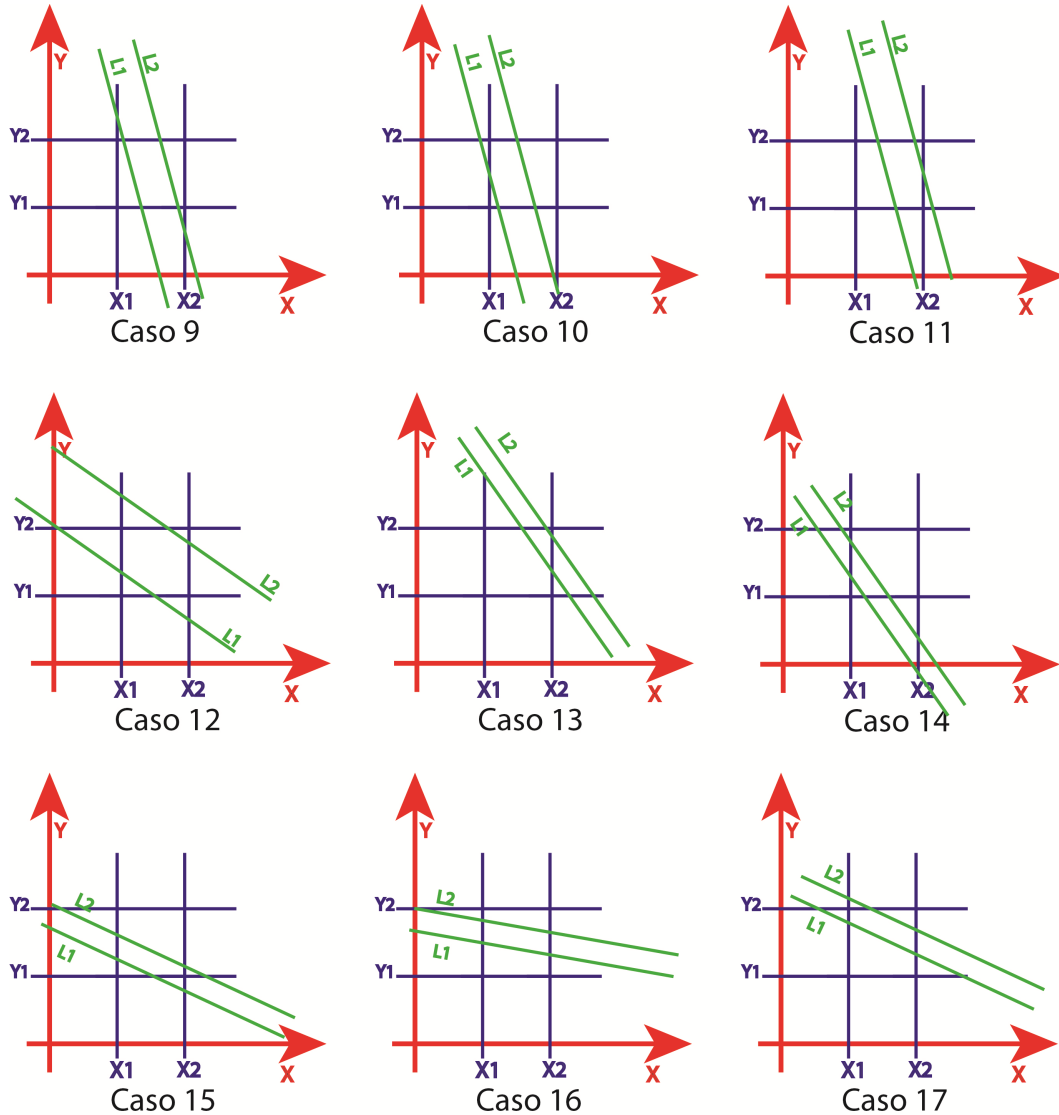


Figure 9.: Para un ángulo $-90 < \theta < 0$ y L_2 y L_1 atraviesan la celda, se generan 9 casos

i. Caso 9: Se calculan cuatro integrales; la integral de L_2 entre x_1 y xb_1 , la integral de L_1 entre x_1 y xa_1 , la integral de L_2 entre x_1 y xb_2 , y la integral de L_1 entre x_1 y xa_1 . El área deseada esta determinada por la suma entre la primera y la cuarta integral, a este resultado se le resta la suma entre la segunda y la tercer integral.

ii. Caso 10: Se calculan tres integrales; la integral de L_2 entre x_1 y xb_1 , la integral de L_1 entre x_1 y xa_1 , y la integral de L_2 entre x_1 y xb_2 . La suma entre la segunda y la tercer integral, es restada a la primera obteniendo el área deseada.

- iii. Caso 11: Se calculan cuatro integrales; la integral de L_2 entre x_1 y x_2 , la integral de L_1 entre x_1 y xa_1 , la integral de L_2 entre x_1 y xb_2 , y la integral de L_1 entre x_1 y xa_2 . El área deseada esta determinada por la suma entre la primera y la cuarta integral, a este resultado se le resta la suma entre la segunda y la tercer integral.
- iv. Caso 12: Se calculan tres integrales; la integral de L_2 entre x_1 y x_2 , la integral de L_1 entre x_1 y x_2 , y la integral de L_2 entre x_1 y xb_2 . La suma entre la segunda y la tercer integral, es restada a la primera obteniendo el área deseada.
- v. Case 13: Se calculan cuatro integrales; la integral de L_2 entre x_1 y x_2 , la integral de L_1 entre x_1 y x_2 , la integral de L_2 entre x_1 y xb_2 , y la integral de L_1 entre x_1 y xa_2 . El área deseada esta determinada por la suma entre la primera y la cuarta integral, a este resultado se le resta la suma entre la segunda y la tercer integral.
- vi. Caso 14: Se calcula la integral de L_2 entre x_1 y xb_1 a esta integral se le resta la integral de L_1 entre x_1 y xa_1 .
- vii. Caso 15: Se calcula la integral de L_2 entre x_1 y x_2 a esta integral se le resta la integral de L_1 entre x_1 y xa_1 .
- viii. Caso 16: Se calcula la integral de L_2 entre x_1 y x_2 a esta integral se le resta la integral de L_1 entre x_1 y x_2 .
- ix. Caso 17: Se calculan tres integrales; la integral de L_2 entre x_1 y x_2 , la integral de L_1 entre x_1 y x_2 , y la integral de L_2 entre x_1 y xb_2 . La suma entre la segunda y la tercer integral, es restada a la primera obteniendo el área deseada.

3. Si $\theta = 0$:

Se calcula como otro caso particular de la DLL ya que las áreas vuelven a ser rectángulos donde la altura esta definida por los limites verticales de la celda y el ancho lo determina $Ldif = \frac{LsenA}{N}$ como se ilustra en la Figura 10

- a) Caso 1: Ninguno de los dos rayos atraviesa la celda evaluada ya que se encuentran a la izquierda de ella, en este caso el $A_r = 0$.
- b) Caso 2: L_2 atraviesa la celda evaluada pero L_1 no, en este caso el área resultante es un rectángulo donde su altura está determinada por los extremos de la celda y la base depende de la incidencia de L_2

$$A_r = (y_2 - y_1) * ((Ldif * n_b) - x_1)$$
- c) Caso 3: L_1 y L_2 atraviesan la celda evaluada, en este caso el área resultante es un rectángulo donde su altura está determinada por los extremos de la celda y la base depende de la incidencia de L_2 y L_1

$$A_r = (y_2 - y_1) * (Ldif * (n_b - n_a))$$
- d) Caso 4: L_1 atraviesa la celda evaluada pero L_2 no, en este caso el área resultante es un rectángulo donde su altura está determinada por los extremos de la celda y la base depende de la incidencia de L_1

$$A_r = (y_2 - y_1) * (x_2 - (Ldif * n_a))$$
- e) Caso 5: Ninguno de los dos rayos atraviesa la celda evaluada ya que se encuentran a la derecha de ella, en este caso el $A_r = 0$.

4. Si $\theta > 0$ && $\theta < 90$:

En este caso el cálculo de las áreas es muy parecido al caso 2) la diferencia radica en que los limites sobre los que se evalúa las áreas son distintos por la pendiente de las líneas.

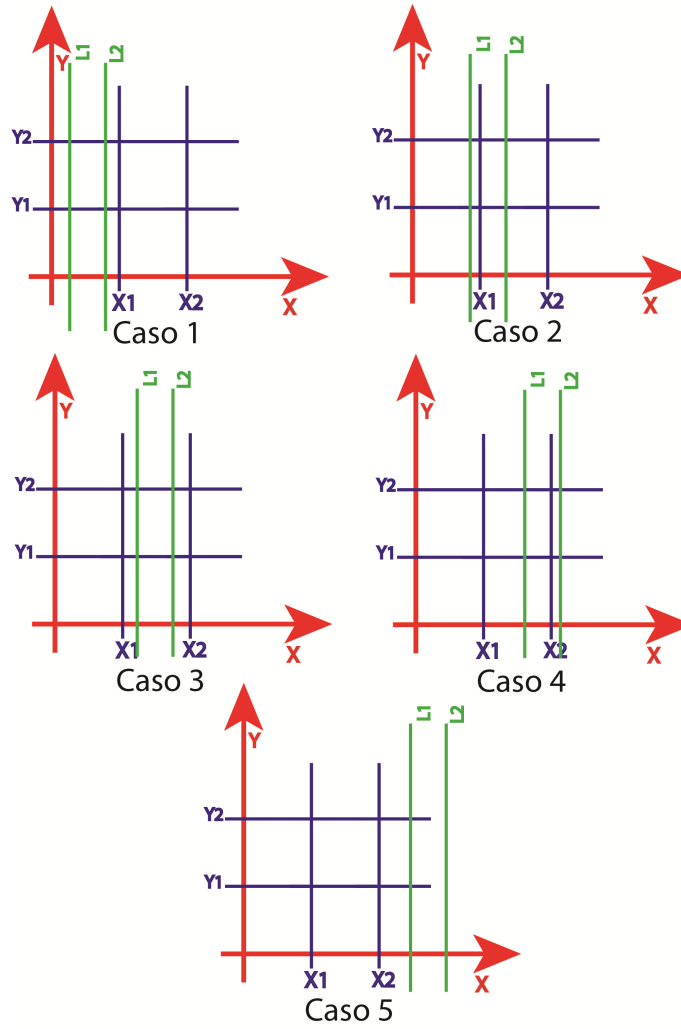


Figure 10.: Para un ángulo $\theta = 0$ se generan 5 casos

Es importante tener en cuenta que los ángulos para la DLL están definidos desde -90 a 89 siendo 0 el eje y , ya que la proyección a 90 sería la misma que en -90 .

RECONSTRUCCIÓN USANDO EL MÉTODO DE MUESTREO COMPRIMIDO

MUESTREO COMPRIMIDO

El Muestreo Comprimido es una técnica del procesamiento de señales en donde se realiza la medición de un pequeño número de combinaciones lineales no adaptativas de la señal, estas mediciones usualmente son mucho menores a la cantidad de muestras que define la señal. Por medio de estas mediciones se reconstruye la señal a través de un procedimiento no lineal.

Se va a reconstruir el vector de coeficientes de atenuación $\alpha = \{\alpha_1, \dots, \alpha_N\}^T$ por medio de la matriz de sensado \mathbf{A} (como la ilustrada en la Ecuación 4) y el vector de mediciones $\mathbf{b} = b_1, \dots, b_M^T$.

$$\mathbf{A}\alpha = \mathbf{b} \quad (11)$$

Siendo $M \ll N$ se tiene un sistema indeterminado de ecuaciones lineales, es decir, tiene infinitas soluciones. Para encontrar la solución *dispersa* ($\alpha \in \mathbb{R}^N$ con el mínimo posible de números distintos a cero) se formula el problema de optimización 12.

$$\alpha = \operatorname{argmin}_{\alpha} \|\alpha\|_0 \quad \text{suje}to \ a \ \mathbf{b} = \mathbf{A}\alpha \quad (12)$$

El problema descrito en la Ecuación 12 encontrará los K valores de la señal *dispersa*. Sin embargo esto es un problema de tiempo polinomial no determinístico (en inglés, NP-hard) ya que requiere una exhaustiva búsqueda de todos los valores $\binom{N}{K}$ distintas a cero en α . Por lo tanto la mejor aproximación en CS es minimizar la norma l_1 [7].

$$\alpha = \operatorname{argmin}_{\alpha} \|\alpha\|_1 \quad \text{s.t.} \ \mathbf{b} = \mathbf{A}\alpha \quad (13)$$

En este caso, $\alpha = \{\alpha_1, \dots, \alpha_N\}^T$ es un vector donde solamente K elementos son diferentes de cero, es decir $\|\alpha\|_0 = K$, siendo $\|\cdot\|_p$ la norma l_p . Generalmente las señales no son exactamente *dispersas* en ninguna base ortogonal, sin embargo son *comprimibles*, es decir si la magnitud de los coeficientes decaen al ser ordenados de mayor a menor. De esta manera después de cierta constante los coeficientes se aproximan a cero y se obtiene una aproximación no lineal.

Algoritmo de reconstrucción Iteraciones Recursivas de Mínimos Cuadrados con norma p

La matriz de sensado \mathbf{A} resultante en la DLL se calculará automáticamente usando el método algebraico y basándose en la retícula generada por los siguientes parámetros *Lhor*, *Lver*, *Col* y *Fil* los cuales determinan la longitud horizontal, la longitud vertical, la cantidad de filas y la cantidad de columnas respectivamente como se muestra en la Figura 11. Como se muestra en la Figura 3 para una proyección a θ , el arreglo emisor de rayos X (localizado en la parte superior izquierda) emite dichos rayos a través del objeto los cuales son recibidos en el arreglo de receptores (parte inferior derecha), la información obtenida por estos receptores es la intensidad del rayo la cual tiene relación con las densidades de las estructuras traspasadas, como se había explicado anteriormente. *LsenA* es la longitud de dicho arreglo tanto de emisor como receptor, *N-Sen* define la cantidad de emisores en el arreglo y *N-Theta* especifica la cantidad de ángulos en los cuales serán

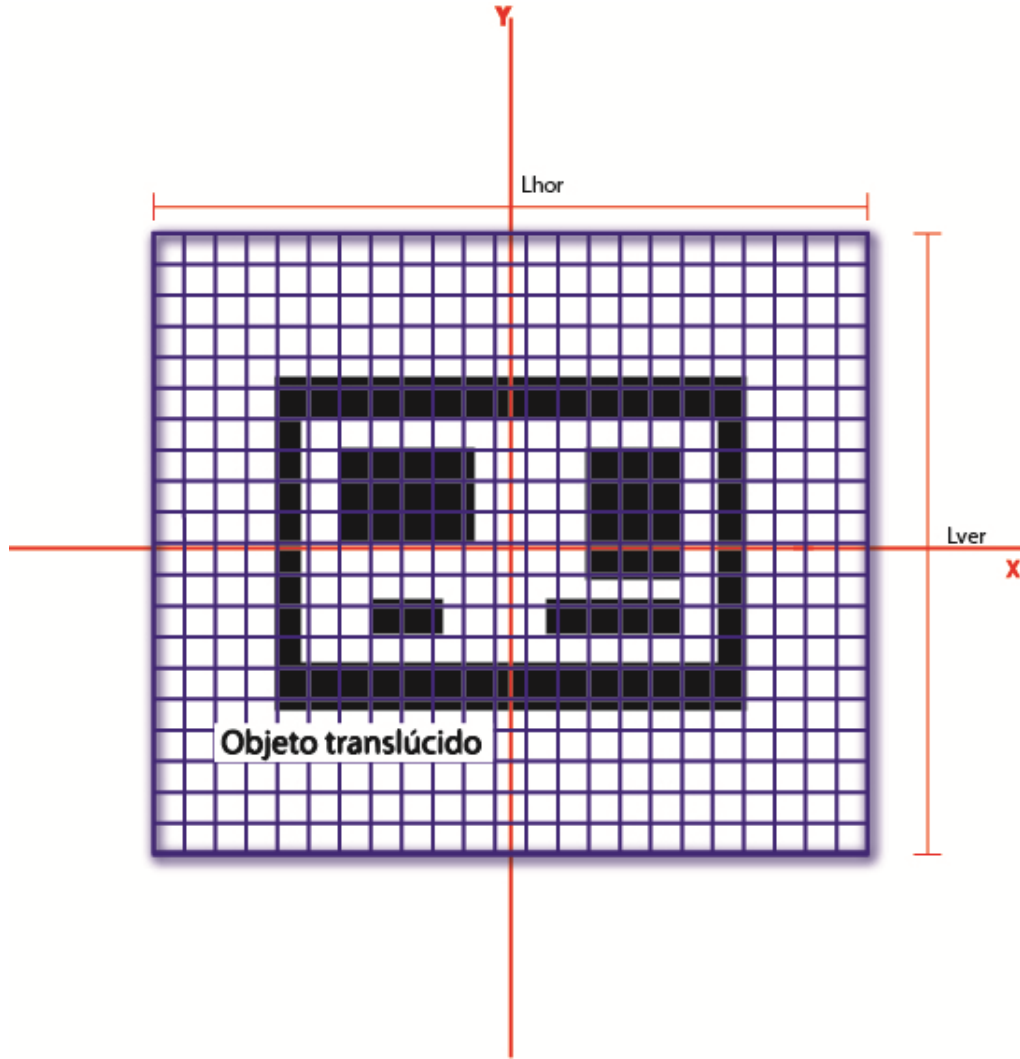


Figure 11.: Retícula sobre la cual se desarrollará la matriz de sensado, se especifican los parámetros de entrada L_{hor} , L_{ver} , Col y Fil ; los cuales corresponden a la longitud horizontal, la longitud vertical, la cantidad de columnas y filas respectivamente.

evaluadas las proyecciones.

Con los parámetros anteriormente descritos, el algoritmo de la DLL calcula las esquinas de cada una de las celdas generadas a partir de la retícula teniendo como origen el centro del objeto a reconstruir.

La matriz \mathbf{A} tendrá N columnas determinadas por la cantidad de celdas a evaluar y M filas organizadas por bloques determinados por cada proyección y en cada bloque se colocarán todos los haces del arreglo.

$$M = (N - Theta) \times (N - Sen)$$

$$N = Fil \times Col$$

Para el algoritmo se formulará el problema de optimización anteriormente descrito (Ecuación 12 y 13) para norma l_p siendo $p \leq 1$

$$\alpha = \operatorname{argmin}_{\alpha} \|\alpha\|_p \text{ s.t. } \mathbf{A}\alpha = \mathbf{b} \quad (14)$$

La forma en la que se desarrollará este problema es por medio de mínimos cuadrados como se ilustra en la Ecuación 15

$$\begin{aligned} \sum_{i=0}^{N-1} |\alpha_i|^2 |\alpha_i|^{p-2} &= \sum_{i=0}^{N-1} \frac{1}{|\alpha_i|^{2-p}} |\alpha_i|^2 \\ &= \sum_{i=0}^{N-1} \Omega_i |\alpha_i|^2 \end{aligned} \quad (15)$$

Se reemplaza la Ecuación 16 para obtener 15, donde Ω_i representa la relación entre los coeficientes calculados en la iteración k y los que se calcularán en la iteración $k + 1$.

$$\Omega_i = \frac{1}{|\alpha_i|^{2-p}} \quad (16)$$

En la Ecuación 17, Ψ es una matriz diagonal compuesta por Ω_k que es la relación calculada a través del vector estimado en la iteración anterior. Se realiza una corrección numérica en la cual si alguno de los coeficientes de α_k es igual a cero se asigna 1×10^{-9} .

$$\alpha_{k+1} = \Psi \mathbf{A}^T (\mathbf{A} \Psi \mathbf{A}^T)^\dagger \mathbf{b} \quad (17)$$

(El desarrollo de la Ecuación 17 se encuentra en el apéndice A).

Es importante tener en cuenta que para el desarrollo de la DLL el algoritmo usa la biblioteca LAPACK desarrollada por el Laboratorio de Innovación Computacional (en inglés Innovative Computing Laboratory ICL) de la universidad de Tennessee, con el fin de manejar el álgebra matricial de la forma más eficiente posible.

Adicionalmente se redujeron operaciones específicamente en la multiplicación matricial de $\Psi \mathbf{A}^T$ realizando la multiplicación de la siguiente manera:

$$\Psi \mathbf{A}^T = [\mathbf{A}_i * \psi_i]_{i=0}^{M-1} \quad (18)$$

Siendo $\mathbf{A}_i = \{A_j\}_{j=0}^{N-1}$.

Para el cálculo de la primer iteración el vector estimado es calculado de la siguiente manera:

$$\mathbf{X}(0) = \mathbf{A}^\dagger \mathbf{b} \quad (19)$$

RESULTADOS

Se documentarán cuatro pruebas; la reconstrucción de un vector aleatorio con dos bibliotecas para el manejo del algebra matricial, la reconstrucción de una imagen sintética binaria (imagen generada en computador compuesta por "1" y "0"), y se reconstruirá una imagen sintética no *disperso*; en estas tres pruebas se reconstruirán dichas imágenes variando la norma p . Finalmente se realizará una gráfica sobre la probabilidad de reconstrucción del algoritmo IRLS para la norma $p = 1$, $p = 0.7$, $p = 0.5$ y $p = 0.25$

Para estas pruebas se usará la siguiente configuración determinada por los parámetros de la DLL: Una retícula de las siguientes dimensiones 64 filas por 64 columnas, la longitud vertical será de 64 píxeles igual que la longitud horizontal. En cuanto a las proyecciones la longitud de los arreglos tanto emisor como receptor será de 64 píxeles donde se ubicaran 80 sensores y se realizaran 26 proyecciones. Dicha configuración genera una matriz de sensado de 2080×4096 mostrada en la Figura 12

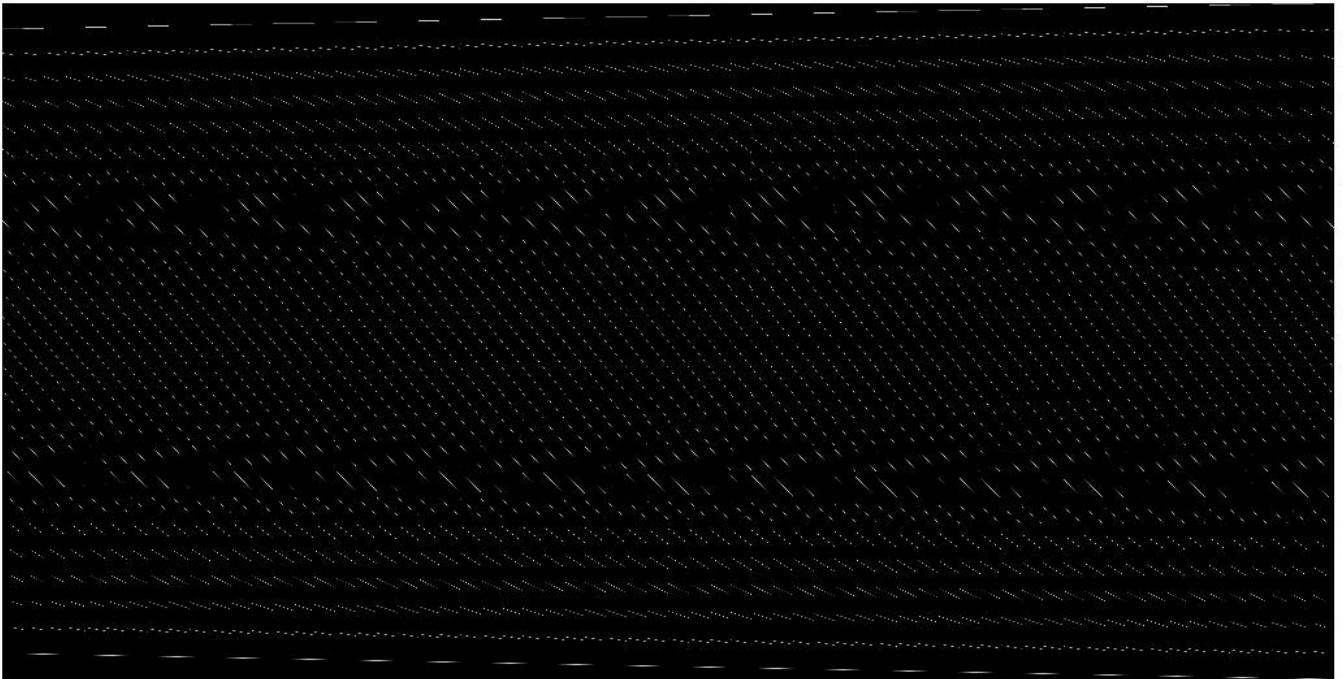


Figure 12.: Matriz de sensado con una configuración de Col=64, Fil=64, Lhor=64, Lver=64, LsenA=64, N-Sen=80, N-Theta=26

3.1 VECTOR ALEATORIO

Se reconstruye un vector aleatorio donde el 10% de los valores son distintos de cero (409 valores). La matriz de sensado es la especificada anteriormente y se muestra en la Figura 12. La reconstrucción del vector

aleatorio fue realizada con distintas normas como se muestra en la tabla 1 y 2, el algoritmo de reconstrucción realiza distinta cantidad de iteraciones para cumplir la condición $\varepsilon < 1 \times 10^{-3}$, dicha cantidad depende de la norma evaluada. Como se puede apreciar los errores cuadráticos medios entre el vector estimado y el original $\Delta \mathbf{b}$ son muy bajos (inferiores a 1×10^{-5}) por lo que se considera que la reconstrucción es perfecta. Finalmente el uso de la biblioteca *MKL INTEL* realiza las operaciones algebraicas aproximadamente 20 veces más rápido que ya que es una librería optimizada para procesadores INTEL.

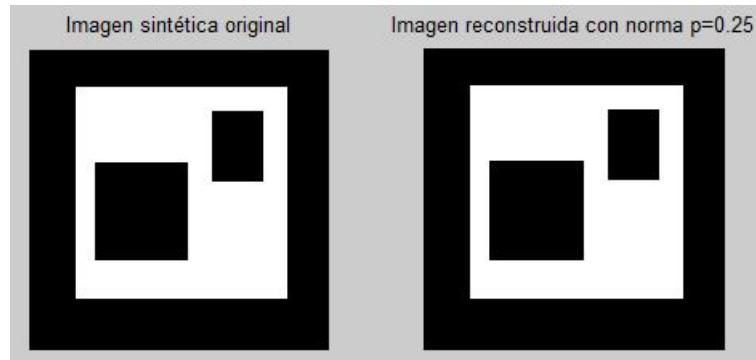
Table 1.: Reconstrucción de un vector aleatorio a distintas normas usando LAPACK

LAPACK			
IRLS Norma	Iteraciones	Error Cuadrático Medio	Tiempo (s)
1	16	1.638×10^{-5}	8879.7
0.7	7	2.988×10^{-7}	4591.9
0.5	6	1.789×10^{-6}	4032.7
0.25	5	1.665×10^{-6}	3536.3

Table 2.: Reconstrucción de un vector aleatorio a distintas normas usando MKL

MKL			
IRLS Norma	Iteraciones	Error Cuadrático Medio	Tiempo (s)
1	17	1.842×10^{-5}	370.2
0.7	7	2.901×10^{-7}	178.2
0.5	6	9.939×10^{-7}	155.3
0.25	5	1.104×10^{-6}	136.0

3.2 IMAGEN SINTÉTICA BINARIA

Figure 13.: Reconstrucción de la imagen sintética binaria con norma $p=0.25$

Con la misma matriz de sensado descrita anteriormente se realizó la reconstrucción de la imagen sintética binaria, mostrada en la Figura 13, la cual tiene 1440 valores distintos de cero; es decir, el 35% del total

de valores aproximadamente. Se puede apreciar en la Tabla 3 que la reconstrucción es perfecta ya que el error cuadrático medio entre la imagen original y la reconstrucción es inferior a 1×10^{-6} en todos los casos. Adicionalmente a medida que la norma disminuye su valor en el número de iteraciones disminuye a tal punto que en la norma 0.25 la cantidad de iteraciones es la cuarta parte de las necesarias para la norma 1.

Table 3.: Reconstrucción de la imagen sintética con IRLS a distintas normas

IRLS Norma	Iteraciones	Error Cuadrático Medio
1	23	3.16×10^{-7}
0.7	8	4.73×10^{-8}
0.5	7	5.81×10^{-13}
0.25	6	2.86×10^{-13}

3.3 IMAGEN SINTÉTICA

Usando la misma matriz de sensado se reconstruye la imagen original (Figura 14 a) con distintas normas. Es importante aclarar que esta imagen es sintética pero no es binaria es decir no es una imagen *disperso* por este motivo se usó la transformada discreta de coseno con el fin de convertir en *disperso* la imagen, sin embargo después del análisis se encuentra que a pesar de la transformación la imagen no cumple esta característica por lo que se realiza un análisis de la energía de los coeficientes; dejando solamente a los K valores con mayor energía distintos de cero.

De esta manera la Ecuación 11 se reescribe de la siguiente manera:

$$\begin{aligned}
 \mathbf{S} &= \mathbf{Q} * \alpha \\
 \mathbf{Y} &= \mathbf{A} * \mathbf{Q}^{-1} \\
 \mathbf{b} &= \mathbf{A} * \alpha \\
 \mathbf{b} &= (\mathbf{A} * \mathbf{Q}^{-1}) * (\mathbf{Q} * \alpha) \\
 \mathbf{b} &= \mathbf{Y} * \mathbf{S}
 \end{aligned} \tag{20}$$

Donde \mathbf{S} es el vector *disperso* o *comprimible*, \mathbf{Y} la nueva matriz de sensado, \mathbf{Q} es la transformada Discreta de Coseno

Al obtener el vector estimado se debe multiplicar por la transformada inversa discreta de coseno $\alpha = \mathbf{Q}^{-1} * \mathbf{S}$ para obtener la imagen original. Para este caso se usará la imagen después de suprimir el 70% de los coeficientes es decir generando una señal *comprimible* $K = 30\%$. Como se puede apreciar en la Figure 14, en c. se encuentra la reconstrucción con norma=1, en d. con norma=0.75, en e. con norma=0.5 y en f. con norma=0.25. Las reconstrucciones tienen un error $\Delta \mathbf{b} < 0.5\%$ y como se puede verificar en la Tabla 4 a medida que la norma p disminuye la cantidad de iteraciones necesarias para cumplir con que la norma euclidiana entre el vector de coeficientes estimado en la iteración k y el vector estimado en la iteración $k + 1$ sea inferior a $\varepsilon < 1 \times 10^{-2}$. Es importante resaltar que con la norma $p = 0.25$ la convergencia del algoritmo presenta algunas oscilaciones por lo tanto debe iterar más veces para cumplir con ε como es el caso de $K = 10\%$ y $K = 40\%$.

Adicionalmente en el caso de $K = 10\%$ se realiza el menor número de iteraciones y el error de la reconstrucción es menor que en los demás.

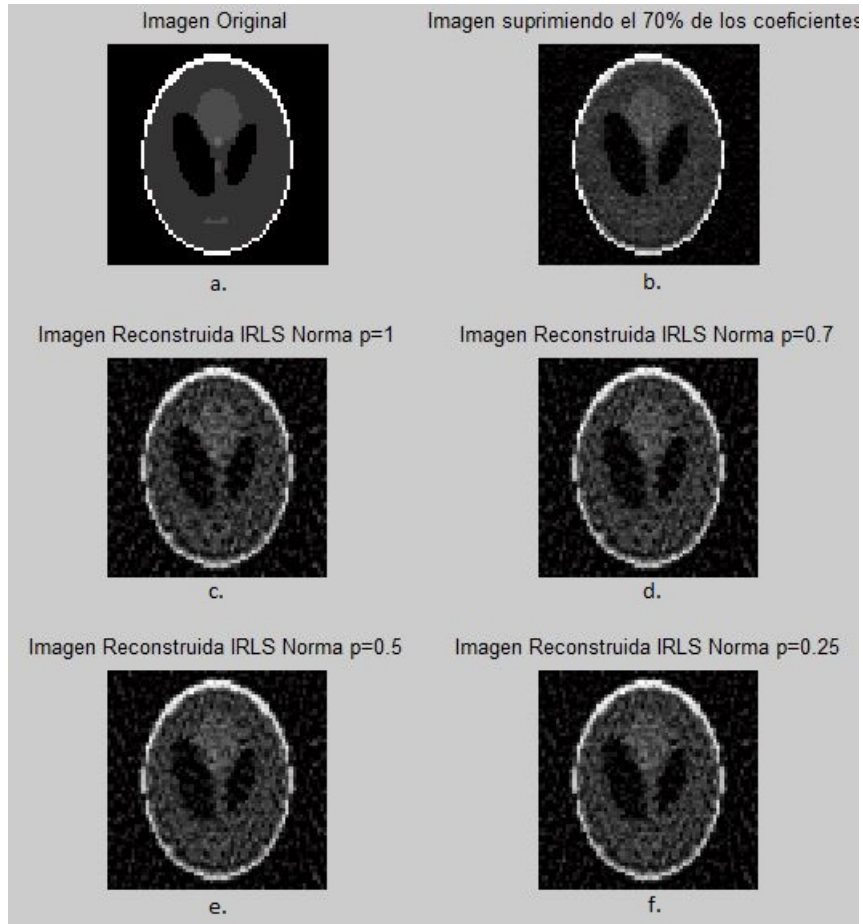


Figure 14.: Reconstrucción de la imagen no sintética (a.) suprimiendo el 80% de los coeficientes (b.), con el código de optimización IRLS ($K=20\%$) para norma=1 (c.), norma=0.7 (d.), norma=0.5 (e.) y norma=0.25 (f.)

3.4 PROBABILIDAD DE RECONSTRUCCIÓN

Para realizar la gráfica de reconstrucción se usaron matrices aleatorias compuestas por números de 0 a 1 con una precisión de 1×10^{-3} donde se varía la cantidad de filas M desde 1 hasta 32 ($N = 32$). Los vectores a reconstruir son aleatorios los cuales podrían tener valores de "0" o "1" y donde la cantidad de "1" está determinada por la variable K la cual variaba desde 1 hasta 32 para cada matriz. La probabilidad se calculó sobre mil vectores reconstruidos, con lo cual se realizaron 1024000 reconstrucciones para realizar. Dicho proceso se realizó 4 veces para calcular la probabilidad de reconstrucción para la norma $p = 1$, $p = 0.7$, $p = 0.5$ y $p = 0.25$

La Figura 15 muestra que bajo la curva la probabilidad de reconstrucción es mayor o igual al 70%. Adicionalmente se puede apreciar que la probabilidad de reconstrucción está totalmente ligada a las condiciones dadas, por ejemplo si M es inferior al 50% de los valores de N , la probabilidad dependerá de cuantos valores distintos a "0" (K) debe reconstruir el algoritmo, caso contrario sucede cuando M es superior al 80% de los valores de N , ya que sin importar los valores distintos a "0" (K) el algoritmo reconstruirá con una alta probabilidad.

La Figura 15 muestra que para la norma $p = 0.25$ se tiene una área donde la probabilidad de reconstrucción es superior que en las demás normas $p = 1$, $p = 0.7$ y $p = 0.5$ es decir, cuando $M \geq 55\%$ de los valores de

Table 4.: Reconstrucción de la imagen no sintética usando el algoritmo IRLS con distintos K y a distintas normas

K	IRLS Norma	Iteraciones	Error Cuadrático Medio
40% 1638 elementos	1	61	1.7×10^{-3}
	0.7	48	1.6×10^{-3}
	0.5	20	1.6×10^{-3}
	0.25	21	1.6×10^{-3}
30% 1228 elementos	1	62	1.5×10^{-3}
	0.7	32	1.5×10^{-3}
	0.5	26	1.4×10^{-3}
	0.25	20	1.4×10^{-3}
20% 819 elementos	1	49	1.2×10^{-3}
	0.7	35	1.2×10^{-3}
	0.5	20	1.2×10^{-3}
	0.25	15	1.2×10^{-3}
10% 409 elementos	1	56	6.6×10^{-4}
	0.7	23	6.1×10^{-4}
	0.5	17	6.2×10^{-4}
	0.25	22	6.6×10^{-4}

N con norma $p = 0.25$ existe una mayor probabilidad de reconstrucción. Sin embargo en los extremos de la gráfica $M < 50\%$ o $M > 90\%$ la probabilidad de reconstrucción es muy parecida para todas las normas.

Es importante aclarar que para esta gráfica se consideró reconstrucción perfecta cuando el error entre el vector estimado y el vector original es inferior al 1%.

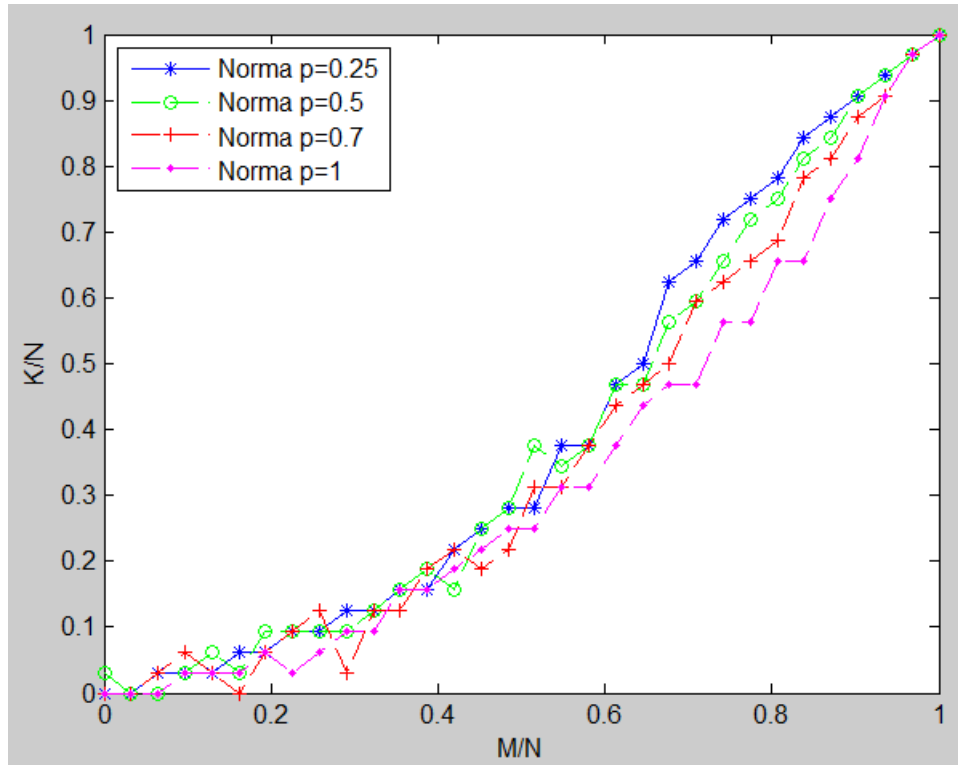


Figure 15.: Probabilidad de reconstrucción para una matriz de $M \times N$ siendo M variable de 1 a 32, con el fin de reconstruir vectores con K elementos distintos de "0", donde K también variaba de 1 a 32. La probabilidad en cada situación específica es calculada sobre 1000 reconstrucciones. Bajo la curva la probabilidad de reconstrucción es mayor o igual al 70%. El eje x está determinado por (M/N) y el eje y (K/N) .

CONCLUSIONES

La matriz de sensado calculada para una geometría en paralelo basada en proyecciones posteriores, es una matriz funcional para la reconstrucción de cortes axiales con el algoritmo de reconstrucción IRLS para norma p . El algoritmo de reconstrucción de iteraciones recursivas de mínimos cuadrados para norma 1 es un problema convexo mientras que normas inferiores a 1 no; sin embargo está probado que también converge y el uso de estas normas tal como la norma $p = 0.25$ reduce de 3 a 4 veces las iteraciones necesarias para la reconstrucción a comparación de la norma 1. Lo que le da una mayor eficiencia a la implementación en el tiempo de reconstrucción. Adicionalmente se comprobó que con norma $p = 0.25$ existe una mayor probabilidad de reconstrucción que con norma $p = 1$

BIBLIOGRAPHY

- [1] M. Danczak, K.J. Wolter, R. Rieske, and H. Roth. Application of computer tomography in electronic technology. In *Electronics Technology: Integrated Management of Electronic Materials Production, 2003. 26th International Spring Seminar on*, pages 108–111, May 2003.
- [2] Health Physics Society. Whole-body computerized tomography screening should not be performed. *Position Statement of the Health Physics Society*, Jul 2007.
- [3] Richard C. Semelka, Diane M. Armao, Jorge Elias, and Walter Huda. Imaging strategies to reduce the risk of radiation in ct studies, including selective substitution with mri. *Journal of Magnetic Resonance Imaging*, 25(5):900–909, 2007.
- [4] E.J. Candes and M.B. Wakin. An introduction to compressive sampling. *Signal Processing Magazine, IEEE*, 25(2):21–30, March 2008.
- [5] Algebraic and statistical reconstruction methods. In *Computed Tomography*, pages 201–240. Springer Berlin Heidelberg, 2008.
- [6] J. Fessler. Analytical tomographic image reconstruction methods, ch. 3.
- [7] Rama Chellappa Vishal M. Patel. *SpringerBriefs in Electrical and Computer Engineering. Sparse Representations and Compressive Sensing for Imaging and Vision*. Springer New York, 2013.

LIST OF FIGURES

- Figure 1 Reconstrucción de imagen por medio del método algebraico: Las ecuaciones lineales usadas para reconstruir están dadas por nueve proyecciones compuestas por tres ángulos y tres rayos, de esta manera se obtiene un sistema de ecuaciones de 9×9 5
- Figure 2 El rayo X de ancho $\Delta\xi$ no atraviesa toda el área del píxel (b^2). La sección de área atravesada por dicho rayo es a_{ij} para el píxel f_8 6
- Figure 3 Geometría paralela de un cuerpo translúcido a θ , se especifican los parámetros de entrada LsenA, N-Sen y N-Theta los cuales hacen referencia a la longitud de los arreglos emisor y receptor, la cantidad de sensores por arreglo y la cantidad de ángulos sobre los que se quiere proyectar 7
- Figure 4 Se modela el par de haces como líneas rectas perpendiculares donde se genera un área limitada dentro de la celda, los haces inciden a un ángulo θ , las esquinas de la celda son x_1, x_2, y_1, y_2 , la distancia entre rayos es L_{dif} 8
- Figure 5 Para un ángulo $\theta = -90$ se generan 5 casos 9
- Figure 6 Puntos de intersección de los haces del rayo con la celda, se definen 4 puntos para L1 y 4 puntos para L2 10
- Figure 7 Para un ángulo $-90 < \theta < 0$ y L1 atraviesa la celda pero L2 no, se generan 4 casos 11
- Figure 8 Para un ángulo $-90 < \theta < 0$ y L2 atraviesa la celda pero L1 no, se generan 4 casos 11
- Figure 9 Para un ángulo $-90 < \theta < 0$ y, L2 y L1 atraviesan la celda, se generan 9 casos 12
- Figure 10 Para un ángulo $\theta = 0$ se generan 5 casos 14
- Figure 11 Retícula sobre la cual se desarrollará la matriz de sensado, se especifican los parámetros de entrada Lhor, Lver, Col y Fil; los cuales corresponden a la longitud horizontal, la longitud vertical, la cantidad de columnas y filas respectivamente. 16
- Figure 12 Matriz de sensado con una configuración de Col=64, Fil=64, Lhor=64, Lver=64, LsenA=64, N-Sen=80, N-Theta=26 18
- Figure 13 Reconstrucción de la imagen sintética binaria con norma $p=0.25$ 19
- Figure 14 Reconstrucción de la imagen no sintética (a.) suprimiendo el 80% de los coeficientes (b.), con el código de optimización IRLS (K=20%) para norma=1 (c.), norma=0.7 (d.), norma=0.5 (e.) y norma=0.25 (f.) 21
- Figure 15 Probabilidad de reconstrucción para una matriz de $M \times N$ siendo M variable de 1 a 32, con el fin de reconstruir vectores con K elementos distintos de "0", donde K también variaba de 1 a 32. La probabilidad en cada situación específica es calculada sobre 1000 reconstrucciones. Bajo la curva la probabilidad de reconstrucción es mayor o igual al 70%. El eje x está determinado por (M/N) y el eje y (K/N). 23

LIST OF TABLES

Table 1	Reconstrucción de un vector aleatorio a distintas normas usando LAPACK	19
Table 2	Reconstrucción de un vector aleatorio a distintas normas usando MKL	19
Table 3	Reconstrucción de la imagen sintética con IRLS a distintas normas	20
Table 4	Reconstrucción de la imagen no sintética usando el algoritmo IRLS con distintos K y a distintas normas	22



IRLS PARA NORMA P

Del desarrollo en la Ecuación 15 se replantea la Ecuación 14 generando el Enunciado 21 (La Ecuación 15 y 14 se encuentran en la sección 2)

$$\alpha = \operatorname{argmin}_{\alpha} \alpha^T \Omega \alpha; \text{ s.t. } \mathbf{A}\alpha = \mathbf{b} \quad (21)$$

$$\Gamma = \alpha^T \Omega \alpha + \lambda^T (\mathbf{A}\alpha - \mathbf{b}) \quad (22)$$

$$\frac{\delta \Gamma}{\delta \alpha} = 0 \rightarrow \frac{\delta \Gamma}{\delta \alpha} = 2\Omega\alpha + \mathbf{A}^T \lambda$$
$$2\Omega\alpha + \mathbf{A}^T \lambda = 0 \quad (23)$$

$$\frac{\delta \Gamma}{\delta \lambda} = 0 \rightarrow \frac{\delta \Gamma}{\delta \lambda} = \mathbf{A}\alpha - \mathbf{b}$$
$$\mathbf{A}\alpha - \mathbf{b} = 0 \quad (24)$$

La Ecuación 22 se obtiene del enunciado del problema 21 y se deriva en función de α (Ecuación 23) y en función de λ (Ecuación 24)

En la Ecuación 23 se despeja α para obtener la Ecuación 25

$$\alpha = \frac{-\Omega^{-1} \mathbf{A}^T \lambda}{2} \quad (25)$$

$$\frac{-\mathbf{A}\Omega^{-1} \mathbf{A}^T \lambda}{2} = \mathbf{b}$$

$$\frac{-\lambda}{2} = (\mathbf{A}\Omega^{-1} \mathbf{A}^T)^{-1} \mathbf{b} \quad (26)$$

Se reemplaza la Ecuación 25 en la Ecuación 24 y se despeja $\frac{-\lambda}{2}$ para obtener la Ecuación 26

$$\alpha = \Omega^{-1} \mathbf{A}^T (\mathbf{A}\Omega^{-1} \mathbf{A}^T)^{-1} \mathbf{b} \quad (27)$$

Se despeja la Ecuación 16 (sección 2) y se asigna ψ (Ecuación 28). La matriz Ψ se genera colocando los valores ψ en la diagonal (Ecuación 29).

$$\Omega_i = \frac{1}{|\alpha_i|^{2-p}}$$

$$\Omega_i^{-1} = |\alpha_i|^{2-p}$$

$$\psi_i = |\alpha_i|^{2-p} \quad (28)$$

$$\Psi = \begin{bmatrix} \psi_1 & 0 & 0 & \dots & \dots & 0 \\ 0 & \psi_2 & 0 & \dots & \dots & 0 \\ 0 & 0 & \psi_3 & \dots & \dots & 0 \\ 0 & 0 & 0 & \psi_4 & \dots & 0 \\ 0 & 0 & 0 & \dots & \psi_K & 0 \\ 0 & 0 & 0 & \dots & \dots & \psi_N \end{bmatrix}$$

(29)

Reemplazando la matriz 29 en la Ecuación 27 se calcula el vector estimado en la nueva iteración partiendo como base del vector estimado de la anterior α_k ; ya que como se especifica en la Ecuación 28 la diagonal de Ψ depende de α_i que son los valores del vector estimado en la iteración inmediatamente anterior, de esta manera se obtiene la Ecuación 30.

$$\alpha_{k+1} = \Psi \mathbf{A}^T (\mathbf{A} \Psi \mathbf{A}^T)^{\dagger} \mathbf{b} \quad (30)$$

B

DESARROLLO E IMPLEMENTACIÓN DE SOFTWARE

La biblioteca de Vínculos Dinámicos DLL y los algoritmos de prueba serán encontrados en el siguiente link
<https://www.dropbox.com/sh/n7f7grxku2cn2ql/AABM4K12TSF5nZFRDfgOb5Cha?dl=0>