

UNIVERSIDAD DEL VALLE DE GUATEMALA

Facultad de Ciencias y Humanidades



Generación de datos sintéticos para el seguimiento de tumores en imágenes de tomografía computarizada en 4D para facilitar el entrenamiento de redes neuronales

Trabajo de graduación presentado por Héctor Aarón Pivaral Guerra para optar al grado académico de Licenciado en Física

Guatemala,

2024



UNIVERSIDAD DEL VALLE DE GUATEMALA

Facultad de Ciencias y Humanidades



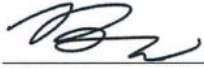
Generación de datos sintéticos para el seguimiento de tumores en imágenes de tomografía computarizada en 4D para facilitar el entrenamiento de redes neuronales

Trabajo de graduación presentado por Héctor Aarón Pivaral Guerra para optar al grado académico de Licenciado en Física

Guatemala,

2024

Vo.Bo.

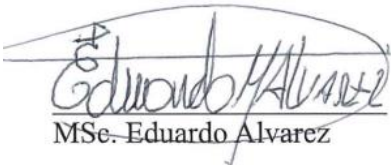


PhD. Byron Wilson

Tribunal examinador:



PhD. Byron Wilson



MSe. Eduardo Alvarez



MSc. Irene Aguilar

Fecha de aprobación del examen de  
graduación:

Guatemala, 9 de agosto de 2024

## Prefacio

Esta tesis, *Generación de datos sintéticos para el seguimiento de tumores en imágenes de tomografía computarizada en 4D para facilitar el entrenamiento de redes neuronales*, representa mi exploración en la disciplina de la física médica, un campo que capturó mi interés durante mis prácticas profesionales en el Hospital El Pilar en Guatemala. Durante estas prácticas, tuve el privilegio de participar en el equipo multidisciplinario necesario para administrar radioterapia a un tumor en la glándula lagrimal de un paciente. La observación detallada del proceso de localización del tumor mediante tomografía computarizada y la subsecuente preparación del tratamiento de radioterapia, destacaron la meticulosidad y precisión requeridas en este campo. Esta experiencia me llevó a dedicar tiempo y esfuerzo en aprender sobre la física médica con la esperanza de contribuir al cuidado de futuros pacientes.

Para involucrarme en la investigación relacionada con la física médica, contacté a Ana Diéguez, una física médica graduada de la Universidad del Valle, quien a su vez facilitó mi conexión con Byron Wilson, quien posee un PhD en Física Médica, para que supervisara el desarrollo de mi tesis.

Este trabajo se estructuró en dos fases principales debido a las limitaciones académicas y prácticas de un proyecto de licenciatura. La primera fase consistió en desarrollar un *DataLoader* para llegar a entrenar en un futuro una red neuronal. La segunda fase, que involucra el entrenamiento e implementación de la red para la localización de tumores, se propone como una continuación de este proyecto en un programa de posgrado.

Los desafíos asociados con la primera fase incluyen el manejo de temas avanzados de física médica, como la generación de imágenes DRR y la manipulación de metadatos bajo el protocolo DICOM. También incluyó el desarrollo de habilidades de programación en Python utilizando bibliotecas avanzadas como Pytorch con soporte de Nvidia CUDA. El acceso restringido a los datos clínicos, debido a las complejidades de formar un acuerdo de investigación transnacional entre nuestras universidades, limitó nuestra capacidad para extender el proyecto durante el curso de pregrado, permitiéndonos trabajar solo con un estudio de caso limitado y completamente anonimizado.

Quiero expresar mi gratitud a Ana Diéguez por facilitar el contacto necesario para desarrollar este proyecto en el ámbito de la física médica. Estoy profundamente agradecido con Byron Wilson por su consejo y apoyo en la realización de mi tesis. Además, me gustaría agradecer a mis padres, Héctor René Pivaral Díaz y Aura Gladys Guerra de Pivaral, por su apoyo constante durante mi formación académica en física.

## Cuadro 1 de abreviaturas

<b>Abreviatura</b>	<b>Significado</b>
DRR	Radiografías reconstruidas digitalmente ( <i>Digitally Reconstructed Radiograph</i> )
DICOM	Imágenes Digitales y Comunicación en Medicina ( <i>Digital Imaging and Communications in Medicine</i> )
CUDA	Arquitectura Unificada de Dispositivos de Cómputo ( <i>Compute Unified Device Architecture</i> )
CT	Tomografía Computarizada ( <i>Computed Tomography</i> )
4DCT	Tomografía Computarizada en Cuatro Dimensiones ( <i>Four-Dimensional Computed Tomography</i> )
CNN	Red Neuronal Convolutiva ( <i>Convolutional Neural Network</i> )
DNN	Red Neuronal Profunda ( <i>Deep Neural Network</i> )
CPU	Unidad Central de Procesamiento ( <i>Central Processing Unit</i> )
GPU	Unidad de Procesamiento Gráfico ( <i>Graphics Processing Unit</i> )
AAPM	Asociación Americana de Físicos en Medicina ( <i>American Association of Physicists in Medicine</i> )
CTV	Volumen Objetivo Clínico ( <i>Clinical Target Volume</i> )
PTV	Volumen Objetivo de Planificación ( <i>Planning Target Volume</i> )
ITV	Volumen Objetivo Interno ( <i>Internal Target Volume</i> )
ACR	Colegio Americano de Radiología ( <i>American College of Radiology</i> )
SBRT	Radioterapia Corporal Estereotáctica ( <i>Stereotactic Body Radiation Therapy</i> )
NSCLC	Cáncer de Pulmón de Células No Pequeñas ( <i>Non-Small Cell Lung Cancer</i> )
GPGPU	Computación de Propósito General en Unidades de Procesamiento Gráfico ( <i>General-Purpose Computing on Graphics Processing Units</i> )
ROI	Regiones de Interés ( <i>Regions of Interest</i> )
GTV	Volumen Tumoral Macroscópico ( <i>Gross Tumor Volume</i> )
API	Interfaz de Programación de Aplicaciones ( <i>Application Programming Interface</i> )
TPM	Mapa de Probabilidad del Objetivo (target probability map)

# Contenido

Prefacio.....	v
Lista de figuras.....	viii
Lista de tablas.....	ix
Sumario.....	x
1. Introducción.....	1
2. Antecedentes.....	3
3. Justificación.....	4
4. Objetivos.....	5
5. Marco teórico.....	6
5.1 Rastreo del tumor durante el movimiento respiratorio.....	6
5.2 Tomografía computarizada.....	8
5.3 Radiografías reconstruidas digitalmente.....	14
5.4 Radioterapia corporal estereotáctica.....	23
6. Metodología.....	25
6.1 GPU.....	25
6.2 Archivos DICOM.....	26
6.3 Determinación del centro de masa.....	28
6.4 Generador de radiografía reconstruida digitalmente.....	31
6.5 Caracterización del ruido en imágenes DRR.....	34
6.6 DataLoader.....	40
7. Resultados y discusión.....	42
8. Conclusiones.....	47
9. Recomendaciones.....	48
10. Bibliografía.....	49
11. Apéndices.....	52

## Lista de figuras

Figura	Página
1. Relación entre el Nivel de Brillo y la Escala de Hounsfield.....	11
2. Las cantidades $\alpha_{\text{min}}$ y $\alpha_{\text{max}}$ definen los rangos permisibles de valores paramétricos para las intersecciones del rayo con los lados de la matriz de CT.....	15
3. Geometría del generador de RRD.....	21
4. Marco de referencia de tomografía para generar la RRD.....	32
5. Gráfico de la Distribución de Poisson para Diferentes Valores de $\lambda$ .....	35
6. Ilustración de la mecánica de la operación de correlación espacial.....	36
7. Ilustración del núcleo Gaussiano para diferentes valores de $\sigma$ .....	39
8. Convolución de la imagen DRR para diferentes tamaños de kernel y valores de $\sigma$ .....	43
9. Ruido de Poisson para diferentes escalas de la imagen DRR.....	44
10. Seguimiento del centro de masa del tumor a lo largo de las 10 fases respiratorias.....	46

## Lista de tablas

Tabla	Página
1. Tiempo de ejecución para la carga de datos DICOM para el generador de DRR, con diferentes GPUs.....	44
2. Tiempo de ejecución del generador de DRR, incluyendo operaciones de ruido, con diferentes GPUs.....	45

## Sumario

El seguimiento preciso de los tumores internos en movimiento es esencial en la irradiación del cáncer de pulmón, ya que las inexactitudes pueden resultar en el fracaso para erradicar el tumor, daño al tejido sano y la imposibilidad de curar la enfermedad. Actualmente, la irradiación precisa del tumor se logra utilizando imágenes fluoroscópicas de marcadores fiduciales de alta densidad. Sin embargo, la implantación de estos marcadores es un procedimiento doloroso, potencialmente dañino y que consume tiempo. Este proyecto buscó mejorar este enfoque mediante el uso de algoritmos de aprendizaje profundo para desarrollar un sistema de seguimiento de tumores sin marcadores, empleando tecnología de imágenes fluoroscópicas similar. Al aplicar estos algoritmos avanzados, el objetivo era maximizar la información extraída de las imágenes para lograr una detección y seguimiento precisos del tumor sin la necesidad de marcadores invasivos. En este marco, se desarrolló un generador de datos para entrenar redes neuronales, que demostró su capacidad generando imágenes de proyección de Radiografía Digitalmente Reconstruida (DRR) en un promedio de 175 ms por imagen, incluyendo ruido personalizado como Gaussiano y moteado cuántico. Además, las imágenes DRR se centraron en el centro de masa, replicando la posición del tumor en el movimiento respiratorio del paciente, con alta fidelidad en la geometría y la continuidad de las reconstrucciones. Este proyecto facilitará la futura expansión para entrenar una red neuronal que rastree tumores en un 4DCT sin marcadores, promoviendo procedimientos menos invasivos y más seguros.

## Abstract

Accurate tracking of moving internal tumors is essential in lung cancer irradiation, as inaccuracies can result in failure to eradicate the tumor, damage to healthy tissue, and failure to cure the disease. Currently, precise tumor irradiation is achieved using fluoroscopic imaging of high-density fiducial markers. However, the implantation of these markers is a painful, potentially harmful, and time-consuming procedure. This project seeks to improve this approach by using deep learning algorithms to develop a marker less tumor tracking system, employing similar fluoroscopic imaging technology. By applying these advanced algorithms, the goal is to maximize the information extracted from the images to achieve accurate tumor detection and tracking without the need for invasive markers. In this framework, a data generator for training neural networks was developed, which demonstrated its capability by generating Digital Reconstructed Radiograph (DRR) projection images in 175 ms per image on average, including custom noise such as Gaussian and quantum mottle. Additionally, the DRR images were centered on the center

of mass, replicating the respiratory motion of the patient and tumor with high fidelity in geometry and continuity of the reconstructions. This project facilitates the future expansion to train a neural network to track tumors in a 4DCT without markers, promoting less invasive and safer procedures

# 1. Introducción

El campo interdisciplinario de la física médica se centra en la aplicación de principios físicos a la medicina. Una de las responsabilidades de un físico médico es comprender y aprovechar los fenómenos físicos involucrados en la adquisición, procesamiento y análisis de imágenes médicas, así como en el desarrollo de técnicas de diagnóstico y tratamiento. En la clínica, los físicos médicos aseguran la entrega segura de radiación y proporcionan el apoyo científico necesario para desarrollar nuevas técnicas de tratamiento. En el desarrollo de un rastreador de tumores sin marcadores utilizando imágenes de tomografía computarizada (CT, por sus siglas en inglés), la física médica es esencial para comprender las fuentes subyacentes de error en las técnicas de imagen y crear un marco basado en cuantificaciones para el entrenamiento y la evaluación de la tecnología desarrollada.

Este proyecto se centró en el monitoreo de tumores ubicados en áreas del cuerpo donde la respiración induce un movimiento significativo durante la fracción en los sitios de tratamiento en el pecho y el abdomen. El seguimiento de tumores y otras estrategias de gestión del movimiento son especialmente críticas en casos donde el tumor se mueve más de 5 mm en cualquier dirección, según las directrices del Grupo de Trabajo 76 (TG-76) de la Asociación Americana de Físicos en Medicina (AAPM). Si no se considera el movimiento durante la planificación del tratamiento, es posible que el tumor no reciba la dosis adecuada, además de aumentar el riesgo de dañar tejidos sanos cercanos. (Ferris, 2020).

Debido a la naturaleza extensa del proyecto, este se dividió en dos etapas para permitir una retirada temprana del proyecto con entregables en la etapa inicial. La primera etapa implicaría el desarrollo de un generador de datos para aplicaciones de aprendizaje automático. La segunda etapa consistiría en desarrollar un regresor de red neuronal convolucional (CNN, por sus siglas en inglés) que prediga la ubicación de un tumor a partir de imágenes fluoroscópicas producidas en la primera etapa.

El software creado utiliza una CT 4D de un paciente para simular imágenes fluoroscópicas. Una CT 4D es una medición del movimiento corporal en el espacio 3D mientras el paciente respira, reconstruida para cada fase del ciclo respiratorio. La adquisición de imágenes médicas utilizando tomografía computarizada en cuatro dimensiones (4DCT) implica capturar proyecciones de CT mientras se registra el patrón respiratorio y reconstruir cada fase del ciclo respiratorio individualmente. Las imágenes resultantes se registran a través de los metadatos según el estándar DICOM (Digital Imaging and Communications in Medicine). El protocolo DICOM estandariza la organización y el manejo de datos volumétricos y físicos de imágenes, facilitando

significativamente su interpretación y manipulación. Los datos registrados incluyen la identificación del paciente, la delineación de contornos que rodean el tumor, la posición inicial de la imagen, el tiempo de exposición al escáner y la orientación del paciente durante la exploración.

Usando los datos de densidad almacenados en la imagen de CT, es posible generar radiografías digitalmente reconstruidas (DRR, por sus siglas en inglés). Además, se puede agregar ruido a las imágenes DRR resultantes, como el ruido Gaussiano y el moteado cuántico, para hacer que las imágenes se asemejen a las imágenes fluoroscópicas que serían recolectadas por un dispositivo de imagen en el mundo real. A partir de este proceso, se generan imágenes fluoroscópicas 2D que incluyen la ubicación conocida del centro del tumor. Estas imágenes pueden ser utilizadas como datos de entrenamiento para una red neuronal convolucional supervisada, que aprende a correlacionar la posición del tumor en imágenes 2D con su ubicación real, optimizando así la precisión en la detección y seguimiento de tumores.

Con el generador de datos, se puede llevar a cabo el entrenamiento de la CNN, donde la CNN aprende a reconocer patrones y características distintivas de los tumores en las imágenes. Una vez entrenada, la CNN puede ser utilizada para analizar datos de imágenes adquiridas en la unidad de tratamiento, con el fin de rastrear tumores en tiempo real.

Los resultados de este estudio demostraron que el DataLoader desarrollado fue capaz de generar Radiografías Digitalmente Reconstruidas (DRR) que replica, en el movimiento respiratorio del paciente, la localización del tumor con alta fidelidad en la geometría y continuidad de las reconstrucciones. Además, se demostró la viabilidad de generar tipos de ruido Gaussiano y de Poisson, permitiendo ajustes personalizados para aproximarse a una plataforma de imágenes fluoroscópicas particular, según las necesidades específicas del estudio. La implementación de GPUs ha reducido significativamente los tiempos de ejecución del programa, ofreciendo así una ventaja considerable en la optimización de procesos.

## 2. Antecedentes

Se ha realizado investigación sobre la detección de tumores en imágenes fluoroscópicas utilizando redes neuronales profundas, notablemente en el estudio "Real-time tumor tracking using fluoroscopic imaging with deep neural network analysis" de R. Hirai et al. Este estudio utilizó la generación de imágenes de Radiografías Digitalmente Reconstruidas (DRR) como datos de entrenamiento para la localización de tumores. La generación de imágenes y el entrenamiento se llevaron a cabo utilizando programación avanzada en C++, y se utilizó equipo de alto rendimiento equipado con GPUs TitanX de NVIDIA Corporation con 12 GB de memoria.

Para este proyecto, se adoptó un enfoque que se centra exclusivamente en el uso del lenguaje Python, aprovechando su capacidad para interactuar con GPUs a través del entorno CUDA de NVIDIA. Esto facilitó el uso de herramientas como PyTorch para el procesamiento de datos. En cuanto al acceso a hardware, se utilizó servicios en la nube como Google Colab, que ofrecen acceso a GPUs con capacidades de 40 GB o 80 GB de memoria de alto ancho de banda (HBM2), lo que redujo significativamente los costos de infraestructura.

En lo que respecta a la generación de DRR, se empleó la biblioteca DiffDRR, desarrollada por Vivek y Golland. Esta biblioteca es capaz de generar proyecciones DRR a partir de escaneos de tomografía computarizada (CT), utilizando principalmente Python (99.5% de su código), lo cual fue crucial para el módulo DataLoader del proyecto.

### **3. Justificación**

El campo de la física médica ofrece una oportunidad para integrar los principios de la física con aplicaciones clínicas en la medicina, particularmente en el tratamiento del cáncer a través de la radioterapia. Esta modalidad terapéutica se enfoca en el uso dirigido de la radiación para irradiar tumores. Un componente importante de la radioterapia es la localización precisa de los tumores.

Contribuir a la mejora de estos procedimientos representa un aspecto importante de mi interés profesional y personal. La radioterapia ha demostrado ser una herramienta vital en la lucha contra el cáncer, ya que aproximadamente el 50% de los pacientes recibirán radioterapia como parte de su tratamiento. Esto ha salvado innumerables vidas y ha abierto el camino para futuras investigaciones y desarrollos en este campo. A través de esta tesis, se buscó aprender sobre el estado actual de la técnica y explorar nuevas innovaciones en el seguimiento de tumores.

En particular, este trabajo propone optimizar el procedimiento de seguimiento de tumores eliminando la necesidad de implantes fiduciales, que actualmente se utilizan para facilitar la localización precisa de tumores durante la radioterapia. Este trabajo tuvo como objetivo desarrollar un generador de datos especializado, diseñado para ayudar en el entrenamiento de futuras redes neuronales capaces de detectar tumores en imágenes de tomografía computarizada en cuatro dimensiones (4DCT). Este DataLoader proporcionará una fuente de datos simulados, replicando diversas condiciones clínicas, permitiendo un entrenamiento eficiente y efectivo de las redes neuronales.

## **4. Objetivos**

### **4.1 Objetivo institucional**

Desarrollar un código que permita la generación de un conjunto de datos adecuado que se utilizará en el futuro para entrenar modelos de redes neuronales, los cuales realizarán el seguimiento de tumores en imágenes fluoroscópicas.

### **4.2 Objetivos académicos**

Adquirir conocimiento sobre los principios fundamentales de la imagenología médica, específicamente la imagenología fluoroscópica, las estructuras de datos de imágenes médicas y cómo manipular estas estructuras.

Comprender los métodos actuales de seguimiento de tumores que utilizan marcadores fiduciales y sus limitaciones.

Desarrollar habilidades en programación y análisis de imágenes mediante la implementación y prueba de diversas técnicas de procesamiento de imágenes para detectar y rastrear tumores en imágenes fluoroscópicas.

## 5. Marco teórico

### 5.1 Rastreo del tumor durante el movimiento respiratorio

La gestión del movimiento respiratorio es importante en el tratamiento de tumores ubicados en el tórax y el abdomen, donde la respiración puede causar desplazamientos significativos del tumor, afectando la precisión de la irradiación. Según las directrices del Grupo de Trabajo 76 (TG-76) de la Asociación Americana de Físicos en Medicina (AAPM), se recomienda gestionar este movimiento cuando excede los 5 mm en cualquier dirección. Una de las limitaciones a la hora de irradiar el tumor cancerígeno debido al movimiento respiratorio es que impide el uso de márgenes estrechos alrededor del volumen objetivo clínico (CTV). Para contrarrestar esta limitación, es esencial emplear volúmenes objetivo de planificación (PTV) expandidos, conocidos como volumen objetivo interno (ITV), que permiten una cobertura adecuada del CTV y minimizan el riesgo de infra dosificación en el objetivo al mismo tiempo que se evita la irradiación de los tejidos normales adyacentes (Ferris, 2020). El sistema que se puede utilizar para irradiar el tumor es CyberKnife® (CK), un avanzado sistema de radiocirugía robótica que emplea el sistema de seguimiento respiratorio Synchrony® para gestionar el movimiento respiratorio y administrar tratamientos de radioterapia de manera precisa. El sistema de seguimiento Synchrony® emplea dos tipos de marcadores para el rastreo de tumores: externos e internos. También existe seguimiento de tumores sin marcadores, para lo cual se requieren algoritmos computacionales avanzados de patrones de imágenes.

Al combinar la información de los marcadores externos e internos, el sistema Synchrony® puede modelar y predecir el movimiento del tumor causado por la respiración. Esta modelización permite que los movimientos del robot que administra la radioterapia se adapten en tiempo real para seguir el tumor a medida que se mueve. De esta manera, el CyberKnife® puede ajustar la dirección y la intensidad del haz de radiación, asegurando que el tratamiento se entregue de manera precisa al tumor mientras minimiza la exposición a los tejidos sanos circundantes.

El seguimiento de marcadores externo de la respiración se basa en la monitorización de la superficie del cuerpo para inferir el movimiento del tumor. Este método puede utilizar distintas tecnologías, como la monitorización óptica y los sensores de movimiento en la superficie abdominal. Con el sistema de seguimiento Synchrony® se utiliza la monitorización óptica mediante LEDs como marcadores externos, que se colocan en el pecho del paciente. Estos LEDs son monitoreados por una cámara montada en el techo, la cual rastrea los movimientos de los LEDs mientras el paciente respira (Ferris, 2020). Respecto los sensores de movimiento registran el desplazamiento del abdomen durante la

respiración (Hirai 2019). Este enfoque no es invasivo, pero la correlación indirecta entre el movimiento superficial y el movimiento interno del tumor puede no ser precisa, especialmente en pacientes con patrones respiratorios irregulares. Además, requiere calibraciones constantes para mantener la precisión del seguimiento (Ferris, 2020).

El seguimiento interno del tumor con marcadores fiduciales implica la implantación de pequeños marcadores radiopacos cerca del tumor, cuya posición se rastrea. Este método proporciona una localización precisa del tumor en tiempo real y permite ajustar la administración de la radiación según el movimiento real del tumor. Sin embargo, es un procedimiento invasivo que puede causar complicaciones como neumotórax y la precisión puede verse afectada si los marcadores se desplazan de su posición original o cuando el marcador no se implanta en el tumor (Hirai 2019).

El seguimiento de tumores sin marcadores puede superar los inconvenientes de los métodos anteriores. Una de estas técnicas es el emparejamiento de plantillas con imágenes fluoroscópicas. Entonces, antes del tratamiento, se realiza una tomografía computarizada de cuatro dimensiones (4DCT) del paciente, capturando imágenes del cuerpo en diferentes fases del ciclo respiratorio para proporcionar un mapa detallado del movimiento del tumor durante la respiración. A partir de estas imágenes 4DCT, se seleccionan las imágenes más representativas que muestran la posición del tumor en varias fases de la respiración, denominadas plantillas. Estas plantillas se preparan antes del tratamiento y pueden mejorarse para reducir diferencias entre las imágenes de tomografía computarizada y las imágenes fluoroscópicas que se usarán durante el tratamiento, asegurando así que las plantillas sean lo más precisas posible. Durante el tratamiento, se utilizan imágenes fluoroscópicas en tiempo real para observar la posición del tumor. La computadora compara constantemente estas imágenes en vivo con las plantillas preparadas anteriormente, buscando coincidencias para determinar la posición exacta del tumor en cada momento específico del ciclo respiratorio. Basándose en esta comparación, el sistema de radioterapia ajusta automáticamente la dirección y el enfoque del haz de radiación para asegurar que siempre esté apuntando al tumor, incluso mientras el paciente respira. Este método no es invasivo, pero puede ralentizar el proceso, y su precisión puede verse afectada por artefactos de movimiento en las imágenes fluoroscópicas (distorsiones o errores en las imágenes).

Otro método de seguimiento sin marcadores es el uso de modelos de regresión lineal, que emplea la relación entre la posición del tumor y la posición del borde del diafragma para inferir el movimiento del tumor. Este enfoque es particularmente útil en el seguimiento de tumores en el hígado y otros sitios abdominales, donde la precisión depende de la capacidad de visualizar claramente el diafragma en las imágenes. Aunque no es invasivo, su eficacia puede verse comprometida por artefactos de movimiento (Hirai 2019).

Las redes neuronales profundas (DNN) representan un enfoque innovador para el seguimiento sin marcadores. Utilizan datos de planificación 4DCT para entrenar un modelo que identifica patrones de movimiento del tumor. Durante el tratamiento, el modelo de DNN rastrea el movimiento del tumor en tiempo real basándose en los patrones aprendidos. Este método ofrece una alta precisión y es no invasivo, pero requiere un entrenamiento

extenso con datos específicos del paciente y más investigación para validar su eficacia en entornos clínicos.

Este enfoque multidisciplinario, que combina precisión técnica con avances en inteligencia artificial, promete mejorar significativamente la eficacia y seguridad de la radioterapia en regiones afectadas por el movimiento respiratorio.

## **5.2 Tomografía computarizada (CT)**

### **5.2.1 Proceso de formación de imágenes en CT**

La Tomografía Computarizada (CT, por sus siglas en inglés) es una herramienta de diagnóstico en la física médica que utiliza rayos X para producir imágenes detalladas del interior del cuerpo. Esto se logra capturando una serie de imágenes transversales procesadas por una computadora. La CT es fundamental en la detección, diagnóstico y monitoreo de tumores cancerígenos, permitiendo a los profesionales de la salud visualizar y medir con precisión la ubicación, tamaño y extensión de los tumores.

La formación de imágenes mediante un escáner de CT se divide en tres etapas: adquisición de datos, reconstrucción de imágenes y la etapa final, que consta de cuatro procesos: visualización, post-procesamiento, almacenamiento y comunicación de imágenes.

La adquisición de datos implica la recolección de mediciones de la transmisión de rayos X del paciente. Los rayos X son emitidos por un tubo dirigido hacia el cuerpo del paciente. Diferentes tejidos, como huesos, músculos y órganos, tienen niveles variados de densidad y número atómico, lo que resulta en variaciones en el perfil de atenuación de los rayos X. Esto significa que algunos tejidos absorberán más radiación mientras que otros permitirán que más rayos X los atraviesen, dependiendo de la energía de los rayos X y las secciones transversales de interacción del medio del cual se está realizando la imagen. Detectores posicionados en el lado opuesto del paciente capturan los rayos X emergentes, creando una imagen basada en los patrones de atenuación.

La reconstrucción de imágenes se realiza enviando los datos de medición a una computadora que procesa esta información utilizando técnicas y algoritmos matemáticos específicos. Los algoritmos más comunes son la transformada de Radon y las técnicas de reconstrucción iterativa. Estos algoritmos utilizan como entrada las proyecciones del cuerpo y como salida proporcionan una representación 3D del cuerpo que podría plausiblemente crear las proyecciones (según los cálculos de los diversos algoritmos). Esto permite la creación de imágenes detalladas en 2D o 3D a partir de múltiples proyecciones obtenidas desde varias direcciones. El escáner de imágenes típicamente utilizado en aplicaciones de radiología y simulación de radioterapia es el haz de abanico, que toma una

sección transversal delgada del paciente en un patrón helicoidal que luego se une para producir una imagen 3D (Seeram, 2016).

En la etapa final, la imagen reconstruida puede visualizarse y posteriormente almacenarse (típicamente, en el formato de archivo Digital Imaging and Communications in Medicine (DICOM)), facilitando el análisis. Estos monitores han facilitado la manipulación, almacenamiento y grabación de imágenes tanto para radiólogos como para tecnólogos (Seeram, 2016).

Las imágenes pueden modificarse para hacerlas más útiles para los observadores en el proceso de interpretación y diagnóstico. Por ejemplo, las imágenes axiales transversales pueden reformatearse en secciones coronales, sagitales y paraxiales. Las imágenes también pueden someterse a otras operaciones, como suavizado de imágenes, realce de bordes, manipulación de escala de grises y procesamiento de imágenes 3D (Seeram, 2016).

El último paso implica la comunicación. Esto se refiere a la transmisión electrónica de datos de texto e imágenes desde el escáner de CT a otros dispositivos como impresoras láser, estaciones de trabajo de diagnóstico, monitores de visualización en el departamento de radiología y computadoras fuera del hospital. Las comunicaciones electrónicas en CT requieren un protocolo estándar para facilitar la conectividad entre equipos y sistemas ubicados en diversos lugares de múltiples proveedores. El estándar utilizado para este propósito es el Digital Imaging and Communications in Medicine (DICOM), establecido por el American College of Radiology (ACR) y la National Electrical Manufacturers Association (Seeram, 2016).

El estándar DICOM facilita la comprensión y manipulación de datos volumétricos y físicos de imágenes médicas de CT, ya que DICOM agrupa la información en conjuntos de datos. Incluidos en estos están la identificación del paciente dentro del archivo, la curva de los contornos que rodean el tumor, la posición de origen de la imagen, el tiempo de exposición del paciente en el escáner o la orientación del paciente (DICOM, n.d.).

### **5.2.2 Principios físicos de la tomografía computarizada**

Al revisarse los detectores que miden la radiación transmitida a través del paciente desde diversas ubicaciones, los valores de transmisión relativa (conocidos como mediciones de atenuación) se pueden calcular de la siguiente manera:

$$\text{Transmisión relativa} = \text{Log} \left( \frac{I_0}{I} \right) \quad (1)$$

Donde  $I_0$  es la intensidad de los rayos X en la fuente, y  $I$  es la intensidad de los rayos X en el detector. Los valores de transmisión relativa se envían a la computadora y se almacenan como datos en bruto. La reducción de la intensidad de un haz de radiación a medida que atraviesa un objeto se conoce como atenuación; mientras que algunos fotones

son absorbidos, otros son dispersados. La atenuación puede depender de los electrones por gramo, el número atómico, la densidad del tejido y la energía de la radiación utilizada.

Existen dos tipos de haces de rayos X a considerar: haz monocromático y haz policromático. En un haz monocromático, todos los fotones tienen la misma energía, mientras que, en un haz policromático, los fotones tienen diferentes energías. Las exploraciones CT se basan en el uso de un haz policromático. Sin embargo, para simplificar los cálculos, se puede aproximar el haz heterogéneo a uno monocromático. A través de esta aproximación, es posible aplicar la Ley de Lambert-Beer para obtener un modelo en el cual se pueda medir la atenuación. Esta ley describe una relación exponencial de la intensidad de los fotones a medida que viajan a través de los tejidos. Se expresa de la siguiente manera:

$$I = I_0 e^{-\mu x} \quad (2)$$

Donde  $I$  es la intensidad transmitida,  $I_0$  es la intensidad original,  $x$  es el grosor del objeto, y  $\mu$  es el coeficiente de atenuación lineal.  $\mu$  indica la cantidad de atenuación que ha ocurrido. El objetivo de la CT es calcular  $\mu$ , por lo que se procede a resolver la ecuación (2) para  $\mu$ :

$$\mu = -\frac{1}{x} \ln\left(\frac{I}{I_0}\right) \quad (3)$$

Los valores de  $I$  y  $I_0$  son medidos por los detectores y  $x$  es conocido. Por lo tanto, es posible calcular  $\mu$ . Considerando que la atenuación es el resultado de la absorción y dispersión de los rayos X, los rayos pueden ser atenuados debido al efecto fotoeléctrico o pueden ser atenuados y dispersados por el efecto Compton. Por lo tanto, en la ecuación (2) la atenuación  $\mu$  se da por:

$$\mu = \mu_p + \mu_c \quad (4)$$

Donde  $\mu_p$  es el coeficiente de atenuación lineal resultante de la absorción fotoeléctrica, y  $\mu_c$  es el coeficiente de atenuación lineal resultante del efecto Compton. La sección transversal fotoeléctrica aumenta con el aumento del número atómico ( $Z$ ) y la disminución de la energía de los fotones incidentes. Para la relativamente baja energía de los escáneres de CT de diagnóstico, el efecto fotoeléctrico proporciona un contraste significativo en los tejidos debido a la variación de la sección transversal con el número atómico  $Z$ , como es el caso de los huesos, y es menos prevalente en los tejidos blandos y materiales con un  $Z$  menor. El efecto Compton tiene una sección transversal significativa a mayor energía de los fotones; sin embargo, su sección transversal solo depende de la densidad eléctrica, la cual no varía mucho entre los tejidos del cuerpo. Por lo tanto, la imagen con un haz de fotones principalmente en la región de la sección transversal fotoeléctrica proporciona un mejor contraste significativo en los tejidos (Seeram, 2016).

Una vez que se mide la atenuación, estos datos de exploración se convierten en una imagen digital caracterizada por números, conocidos como números CT o Unidades Hounsfield (HU). Los números CT facilitan la conversión de las lecturas de atenuación en

una imagen CT a través de algoritmos de reconstrucción. Los números CT están correlacionados con los coeficientes de atenuación lineal  $\mu$  de los tejidos que constituyen la sección; estos números se expresan de la siguiente manera:

$$\text{Número CT} = \frac{\mu_t - \mu_w}{\mu_w} * K \quad (5)$$

Aquí,  $\mu_t$  es el coeficiente de atenuación del tejido,  $\mu_w$  es el coeficiente de atenuación del agua, y  $K$  es el factor de escala. El factor de escala típicamente se asigna un valor de 1000 para permitir una representación más refinada y detallada de las diferencias de atenuación de los rayos X en la imagen CT. En consecuencia, esto permite una visualización más precisa de los diferentes tejidos. Los números CT obtenidos con este factor se denominan escala de Hounsfield (HU). Esta escala se expresa de la siguiente manera:

$$HU_t = \frac{\mu_t - \mu_w}{\mu_w} * 1000 \quad (6)$$

La escala de Hounsfield también se establece, relativamente, con la atenuación del agua como referencia. Por lo tanto, en la escala de Hounsfield, el valor para el agua es siempre 0. Los valores positivos representan sustancias que son más densas que el agua, mientras que los valores negativos representan sustancias que son menos densas que el agua. El aire típicamente tiene una densidad menor a 0, lo que representa una HU de -1000 en esta escala, mientras que la HU del hueso depende de la densidad y composición del material biológico, lo cual varía de paciente a paciente y según la función biológica del hueso, pero típicamente es de 150 a 3000 HU (Seeram, 2016).

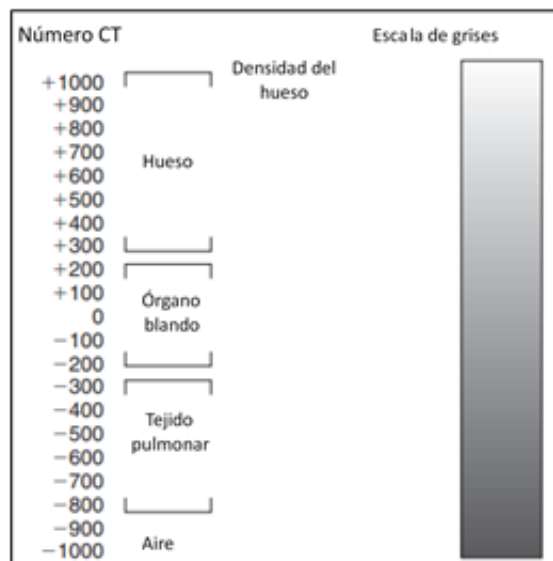


Figura 1. Relación entre el Nivel de Brillo de la imagen CT y la Escala de Hounsfield (Seeram, 2016)

Las computadoras calculan la escala de Hounsfield, y a través de estas mediciones, se crea una imagen numérica. Estas HU de CT representan un valor único del coeficiente de

atenuación lineal de los fotones en el medio, que se representa en escala de grises. Se establecen niveles de brillo correspondientes a esta escala. Los límites superiores (+3096) e inferior (-1000) de la escala representan (para un sistema de imagen de profundidad de 12 bits) blanco y negro, respectivamente. Todos los demás valores representan varios tonos de gris (Seeram, 2016). En la Figura 1, se puede observar la relación entre la escala de Hounsfield y el nivel de brillo.

### 5.2.3 Imágenes CT 4D

Cuando se trabaja con tumores ubicados cerca del diafragma, es esencial considerar su movimiento, ya que pueden desplazarse más de 1 cm debido a la respiración. Además, los movimientos de los órganos también deben tenerse en cuenta. Este movimiento de los órganos puede interferir con la calidad de las imágenes obtenidas por CT, afectando la clara identificación del tumor. Los escáneres CT capturan imágenes en un intervalo de tiempo muy corto (0.5 a 1 segundo), lo cual es considerablemente más rápido que un ciclo respiratorio completo (3 a 6 segundos). Por lo tanto, cada proyección tomada refleja la posición del tumor en ese momento específico del ciclo respiratorio, sin proporcionar una representación completa de sus movimientos y posiciones a lo largo de toda la respiración (Mayles, 2022). Además, una imagen típica, dependiendo de su longitud y parámetros de escaneo, puede tardar segundos en adquirirse, lo que es suficiente para que esté presente un artefacto de movimiento, si no se utilizan otros enfoques para tener en cuenta este movimiento. Para abordar esta situación, se han desarrollado los siguientes enfoques:

- Contención de la respiración: Se mide el volumen de aire inhalado y se instruye al paciente que lo retenga dentro de un rango de tiempo especificado (Mayles, 2022).
- Compresión abdominal: En este caso, se emplea una placa de compresión abdominal para minimizar la amplitud de la respiración y limitar el movimiento de los órganos dentro del diafragma (Mayles, 2022).
- Irradiación controlada: El haz del acelerador se activa selectivamente durante una amplitud de respiración predefinida o una fase específica de la respiración (Mayles, 2022).

A través de un escáner CT estándar, algunas de estas opciones pueden lograrse; sin embargo, para la mayoría, se requiere un escáner CT 4D, un dispositivo que facilita la captura de imágenes durante todo el ciclo respiratorio (Mayles, 2022).

El enfoque comúnmente utilizado para rastrear tumores dentro del diafragma es el enfoque de irradiación controlada. Para este método, se tiene el módulo de la tomografía computarizada 4D (CT4D). Esta CT correlacionada con la respiración permite la reconstrucción de una serie de imágenes CT, cada una correspondiente a una porción del ciclo respiratorio. Esto se logra registrando la señal respiratoria del paciente.

Para realizar una exploración CT 4D, se utilizan típicamente tres tipos de dispositivos:

- Se puede usar una pequeña caja equipada con marcadores externos, que son reflectores de infrarrojos. Se coloca en la parte superior del abdomen. Los marcadores son monitoreados por una cámara de dispositivo de carga acoplada que está equipada con iluminación infrarroja, registrando así la posición de los marcadores. A través de una interfaz, se envían señales de activación al escáner CT. Estas señales sirven como punto de referencia temporal para etiquetar cada imagen adquirida (Mayles, 2022).
- También existe la opción de usar equipos ópticos que emplean cámaras infrarrojas para recrear la estructura tridimensional del paciente, o sistemas de escaneo de superficie basados en láser. De esta manera, se enfoca un punto o área específica en la superficie superior del abdomen para monitorear los movimientos respiratorios. Una interfaz es responsable de transmitir una señal en puntos cruciales de la actividad respiratoria al escáner CT, con el objetivo de marcar las imágenes correspondientes (Mayles, 2022).
- Se puede usar un cinturón colocado alrededor de la parte superior del abdomen para detectar el movimiento respiratorio. La señal respiratoria se adquiere a través de un sensor de galga extensiométrica adjunto al paciente (Mayles, 2022).<sup>1</sup>

Antes de capturar las imágenes, se registran múltiples ciclos respiratorios para determinar la duración promedio de la respiración del paciente. La exploración CT y la adquisición de la señal respiratoria comienzan con la iniciación del tubo de rayos X, marcando el origen del tiempo. La señal respiratoria se utiliza para marcar temporalmente las imágenes en ese instante (Mayles, 2022).

Existen dos criterios fundamentales para organizar las imágenes capturadas, con el objetivo de crear una secuencia coherente y cronológica que represente diferentes fases del ciclo respiratorio. Estos se clasifican como: Amplitud y Fase. En la clasificación por amplitud, el rango completo cubierto por la señal se divide en intervalos iguales entre la inspiración (100% de amplitud) y la espiración (0% de amplitud). En la clasificación por fase, el tiempo entre dos puntos de inhalación completa, que podría considerarse el período de la señal respiratoria, se segmenta en intervalos de tiempo iguales. Siempre se recomienda reconstruir 10 series de imágenes correspondientes a una variación del 10% en amplitud o fase (Mayles, 2022).

Cada criterio tiene sus ventajas, pero algunos estudios sugieren que la categorización basada en amplitud produce menos artefactos de imagen en comparación con la categorización basada en fase. La decisión sobre qué método utilizar se basa en la observación de irregularidades en la amplitud o el patrón del ciclo respiratorio y en

---

<sup>1</sup> Las galgas extensiométricas son herramientas valiosas en la técnica aplicada de medición eléctrica de magnitudes mecánicas. Se utilizan para medir esfuerzos. El término técnico "esfuerzo" implica la deformación debido a la tensión y la compresión, que se distingue por un signo positivo o negativo. Por lo tanto, las galgas extensiométricas se emplean para medir la expansión y la contracción (OMEGA, 22 de octubre de 2023, OMEGA Strain Gauges).

seleccionar el enfoque más adecuado considerando estos factores. La clasificación de las imágenes según la fase o la amplitud es automática, pero es importante tener la opción de edición manual (Dieterich, 2016).

## 5.3 Radiografías reconstruidas digitalmente (DRR)

### 5.3.1 Utilidad de las DRR

La Radiografía Reconstruida Digitalmente (DRR, por sus siglas en inglés) es una simulación de una imagen de rayos X creada a partir de datos de tomografía computarizada (CT) y el conjunto de datos tridimensionales obtenidos durante el escaneo del paciente. Las imágenes DRR son imágenes 2D que imitan la apariencia y geometría de una imagen de rayos X desde un ángulo específico del sistema detector de rayos X. Su función principal es proporcionar un método preciso para verificar la posición y configuración del paciente durante la administración del haz de radiación. Esto se logra al permitir la visualización clara de los campos de tratamiento en relación con la anatomía específica del paciente desde una proyección particular. Asegurar una alineación precisa es esencial para mantener la consistencia entre el plan de tratamiento, utilizado para calcular la distribución de dosis, y la administración real del haz de radiación.

### 5.3.2 Algoritmo de Siddon

El proceso de generación de DRR se reduce al problema geométrico de calcular la trayectoria radiológica para un rayo específico a través de la matriz tridimensional del volumen CT (el cuerpo del paciente en la tomografía). Una densidad particular del vóxel se denota como  $\rho(i, j, k)$  y la longitud contenida por ese vóxel como  $l(i, j, k)$ . De esta manera, la trayectoria radiológica se da por:

$$d = \sum_i \sum_j \sum_k l(i, j, k) \rho(i, j, k) \quad (7)$$

Antes de la creación de este método, se había enfatizado que calcular la trayectoria radiológica no era factible porque la ecuación (7) establece un algoritmo que escala con el número de vóxeles en el volumen CT. Sin embargo, Siddon desarrolló un algoritmo que escala con la suma de las dimensiones lineales de la matriz CT (Siddon, 1985).

En este método, los vóxeles se conceptualizan de manera similar a los píxeles, pero en el espacio tridimensional. Normalmente, los píxeles crean una cuadrícula en una pantalla, formada por la intersección de líneas horizontales y verticales. Esta cuadrícula consiste en cuadrados, y cada cuadrado representa un píxel. El tamaño de cada cuadrado está determinado por el espaciamiento de estas líneas. Para los vóxeles, el concepto se extiende

a tres dimensiones utilizando conjuntos de planos paralelos en lugar de líneas. Estos planos se intersecan en ángulos rectos en tres direcciones: horizontal, vertical y perpendicularmente. Al intersecarse estos planos forman una cuadrícula de cubos, donde cada cubo representa un vóxel. Similar a los píxeles, el volumen de cada cubo vóxel está dictado por el espaciamiento entre los planos.

Volviendo al caso de los píxeles, la intersección del rayo con las líneas se calcula, en lugar de la intersección de los rayos con píxeles individuales. Esto se hace porque simplifica aún más el problema. Dado que las líneas están igualmente espaciadas, solo es necesario determinar la primera intersección y generar el resto recursivamente. La intersección del rayo consiste en dos conjuntos, uno para las líneas horizontales y otro para las líneas verticales. La intersección del rayo con los píxeles es un subconjunto de intersecciones con las líneas. Identificar ese subconjunto permite la determinación de la trayectoria radiológica (Siddon, 1985).

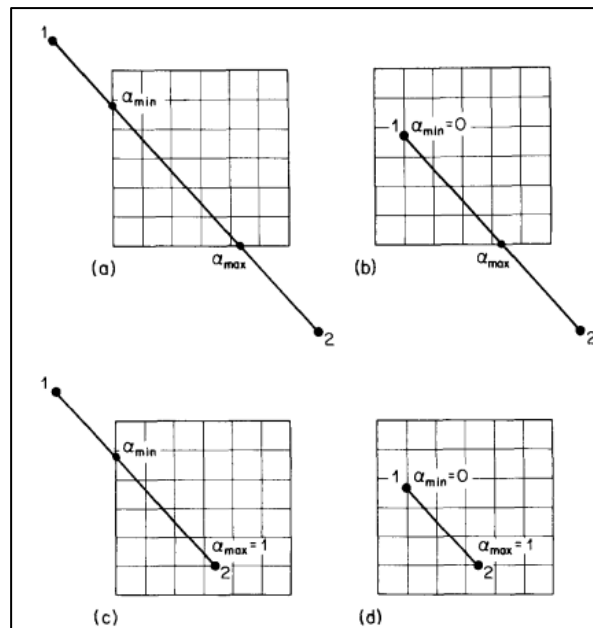


Figura 2. Las cantidades  $\alpha_{min}$  y  $\alpha_{max}$  definen el rango permisible de valores paramétricos para las intersecciones del rayo con los lados de la matriz de CT (Siddon, 1985)

En el caso de los vóxeles, el rayo desde el punto 1 hasta el punto 2 puede representarse paraméricamente por:

$$\begin{cases} X(\alpha) = X_1 + \alpha(X_2 - X_1) \\ Y(\alpha) = Y_1 + \alpha(Y_2 - Y_1) \\ Z(\alpha) = Z_1 + \alpha(Z_2 - Z_1) \end{cases} \quad (8)$$

donde el parámetro de intersección del rayo,  $\alpha$ , es cero en el punto 1 y uno en el punto 2.

Como se ve en la Figura 2(a), si ambos puntos 1 y 2 están fuera de la matriz CT, los valores paramétricos correspondientes a los dos puntos de intersección del rayo con los lados de la matriz CT se dan por  $\alpha_{min}$  y  $\alpha_{max}$ . Todas las intersecciones del rayo con planos individuales deben tener valores paramétricos que se encuentren dentro del rango ( $\alpha_{min}$ ,  $\alpha_{max}$ ). Como se observa en la Figura 2(b), cuando el punto 1 está dentro de la matriz, el valor de  $\alpha_{min} = 0$ . Cuando el punto 2 está dentro de la matriz,  $\alpha_{max} = 1$ , como se muestra en la Figura 2(c). Finalmente, cuando ambos puntos están dentro de la matriz, entonces  $\alpha_{min} = 0$  y  $\alpha_{max} = 1$  (Siddon, 1985).

La solución a la intersección del rayo con la CT implica determinar los parámetros de intersección, dentro del rango ( $\alpha_{min}$ ,  $\alpha_{max}$ ), del rayo con cada conjunto ortogonal de planos paralelos igualmente espaciados. Con respecto a cada conjunto de valores paramétricos, deben fusionarse en un solo conjunto. La medición de la longitud de un rayo situado dentro de un vóxel específico, expresada en unidades de longitud del rayo, se calcula como la diferencia entre dos valores paramétricos consecutivos del conjunto combinado. Para determinar cada punto de intersección del rayo con los vóxeles, se identifican los índices de los vóxeles involucrados. Posteriormente, la longitud de cada segmento de intersección se multiplica por la densidad del vóxel correspondiente, y los resultados se suman para calcular la trayectoria radiológica (Siddon, 1985).

Para una matriz CT de vóxeles ( $N_x - 1, N_y - 1, N_z - 1$ ), la posición de los conjuntos ortogonales de planos paralelos igualmente espaciados se da por:

$$\begin{cases} X_{plane}(i) = X_{plane}(1) + (i - 1)d_x & (i = 1, \dots, N_x) \\ Y_{plane}(i) = Y_{plane}(1) + (i - 1)d_y & (i = 1, \dots, N_y) \\ Z_{plane}(i) = Z_{plane}(1) + (i - 1)d_z & (i = 1, \dots, N_z) \end{cases} \quad (9)$$

donde  $d_x, d_y$  y  $d_z$  son las distancias entre los planos en sus respectivas dimensiones y también pueden referirse como las longitudes de los lados del vóxel.

Considerando que los valores paramétricos  $\alpha_{min}$  y  $\alpha_{max}$  se obtienen al intersectar el rayo con los lados de la matriz CT. Igualando las ecuaciones (8) y (9), se puede obtener  $X_{plane}(i) = X(\alpha_i)$ , los índices  $i = 1$  e  $i = N_x$  dan los valores paramétricos correspondiente a los lados de la matriz CT:

Si  $(X_2 - X_1) \neq 0$

$$\begin{cases} \alpha_x(1) = \frac{[X_{plane}(1) - X_1]}{X_2 - X_1} \\ \alpha_x(N_x) = \frac{[X_{plane}(N_x) - X_1]}{X_2 - X_1} \end{cases} \quad (10)$$

Con expresiones similares para  $\alpha_y(1), \alpha_y(N_y), \alpha_z(1)$  y  $\alpha_z(N_z)$ . Note que si el denominador  $(X_2 - X_1) = 0$ , significa que el rayo es perpendicular al eje x, y los valores

de  $\alpha_x$  no están definidos, lo mismo aplica para los valores de  $\alpha_y$  y  $\alpha_z$ . Para los valores no definidos de  $a_x, a_y$  y  $a_z$ , se excluyen del cálculo (Siddón, 1985).

Así, las cantidades  $\alpha_{min}$  y  $\alpha_{max}$  se definen por:

$$\begin{cases} \alpha_{min} = \max\{0, \min[a_x(1), a_x(N_x)], \min[a_y(1), a_y(N_y)], \min[a_z(1), a_z(N_z)]\} \\ \alpha_{max} = \min\{1, \max[a_x(1), a_x(N_x)], \max[a_y(1), a_y(N_y)], \max[a_z(1), a_z(N_z)]\} \end{cases} \quad (11)$$

donde las funciones min y max seleccionan de su lista de argumentos los términos mínimos y máximos, respectivamente. De todos los planos interceptados, solo ciertos planos interceptados tendrán valores paramétricos en el rango  $(\alpha_{min}, \alpha_{max})$ .

Para obtener los rangos de índices  $(i_{min}, i_{max})$ ,  $(j_{min}, j_{max})$  y  $(k_{min}, k_{max})$ , se procede a utilizar las ecuaciones (8) y (9). Sea  $(X_2 - X_1) > 0$ . Si restamos  $X_{plane}(N_x)$  con  $(N_x - i_{min})d_x$ , se puede obtener la posición del plano que corresponde a  $X(\alpha_{min})$ . Por lo tanto, para obtener  $i_{min}$  se realiza lo siguiente:

$$\begin{aligned} X_{plane}(N_x) - (N_x - i_{min})d_x &= X(\alpha_{min}) \\ X_{plane}(N_x) - (N_x - i_{min})d_x &= X_1 + \alpha_{min}(X_2 - X_1) \\ -N_x + i_{min} &= \frac{X_1 + \alpha_{min}(X_2 - X_1) - X_{plane}(N_x)}{d_x} \\ -N_x + i_{min} &= \frac{X_1 + \alpha_{min}(X_2 - X_1) - X_{plane}(N_x)}{d_x} \\ i_{min} &= N_x - \frac{[X_{plane}(N_x) - \alpha_{min}(X_2 - X_1) - X_1]}{d_x} \end{aligned} \quad (12.a)$$

Para obtener  $i_{max}$ , se comienza con  $X_{plane}(i_{max}) = X(\alpha_{max})$

$$\begin{aligned} X_{plane}(1) + (i_{max} - 1)d_x &= X_1 + \alpha_{max}(X_2 - X_1) \\ X_{plane}(1) + (i_{max} - 1)d_x &= X_1 + \alpha_{max}(X_2 - X_1) \\ i_{max} &= 1 + \frac{X_1 + \alpha_{max}(X_2 - X_1) - X_{plane}(1)}{d_x} \end{aligned} \quad (12.b)$$

Mediante esto se obtienen el rango máximo y mínimo de los índices. Para el caso  $(X_2 - X_1) < 0$ , se puede utilizar las ecuaciones (12.a) y (12.b). En este caso (12.a) se cambia  $\alpha_{min}$  por  $\alpha_{max}$ , y (12.b) se cambia  $\alpha_{max}$  por  $\alpha_{min}$ . Esto se da porque el rayo viene desde  $X_2$  intersectando primero al plano correspondiente a  $\alpha_{max}$  y, por último, intersectando

al plano correspondiente  $\alpha_{min}$  para alcanzar el punto  $X_1$ . Resumiéndose este resultado se tiene lo siguiente:

Si  $(X_2 - X_1) > 0$ ,

$$\begin{cases} i_{min} = N_x - \frac{[X_{plane}(N_x) - \alpha_{min}(X_2 - X_1) - X_1]}{d_x} \\ i_{max} = 1 + \frac{[X_1 + \alpha_{max}(X_2 - X_1) - X_{plane}(1)]}{d_x} \end{cases} \quad (13.a)$$

si  $(X_2 - X_1) < 0$ .

$$\begin{cases} i_{min} = N_x - \frac{[X_{plane}(N_x) - \alpha_{max}(X_2 - X_1) - X_1]}{d_x} \\ i_{max} = 1 + \frac{[X_1 + \alpha_{min}(X_2 - X_1) - X_{plane}(1)]}{d_x} \end{cases} \quad (13.b)$$

con expresiones similares para  $(j_{min}, j_{max})$ , y  $(k_{min}, k_{max})$  (Siddón, 1985).

Para un rango dado de índices  $(i_{min}, i_{max})$ ,  $(j_{min}, j_{max})$ , y  $(k_{min}, k_{max})$ , el conjunto de valores paramétricos  $\{\alpha_x\}$ ,  $\{\alpha_y\}$ , y  $\{\alpha_z\}$ , correspondientes a las intersecciones del rayo con los planos, se define por:

Si  $(X_2 - X_1) > 0$ ,

$$\{\alpha_x\} = \{a_x(i_{min}), \dots, a_x(i_{max})\}$$

si  $(X_2 - X_1) < 0$ ,

$$\{\alpha_x\} = \{a_x(i_{max}), \dots, a_x(i_{min})\}$$

Donde

$$\begin{aligned} \alpha_x(i) &= \frac{[X_{plane}(i) - X_1]}{X_2 - X_1} \\ \alpha_x(i) &= a_x(i - 1) + \left\lfloor \frac{d_x}{(X_2 - X_1)} \right\rfloor \end{aligned} \quad (14)$$

con expresiones similares para  $\{\alpha_y\}$  y  $\{\alpha_z\}$  (Siddon, 1985).

Los conjuntos  $\{\alpha_x\}$ ,  $\{\alpha_y\}$ , y  $\{\alpha_z\}$  se organizan en orden ascendente. Cada elemento dentro de estos conjuntos representa un punto donde el rayo se intersecta con un plano específico. Para identificar las intersecciones del rayo con los vóxeles, los conjuntos  $\{\alpha_x\}$ ,  $\{\alpha_y\}$ , y  $\{\alpha_z\}$  se combinan en un solo conjunto. Los valores paramétricos  $a_{min}$  y  $a_{max}$  se

incorporan para tener en cuenta los casos en que uno o ambos extremos del rayo se encuentran dentro de la matriz CT. El conjunto  $\{\alpha\}$  consiste en los valores  $\alpha_{min}$ ,  $\alpha_{max}$ , y los conjuntos combinados  $\{\alpha_x\}$ ,  $\{\alpha_y\}$ , y  $\{\alpha_z\}$  (Siddón, 1985).

$$\begin{aligned}\{\alpha\} &= \{\alpha_{min}, \text{merge}[\{\alpha_x\}, \{\alpha_y\}, \{\alpha_z\}], \alpha_{max}\} \\ \{\alpha\} &= \{\alpha(0), \dots, \alpha(n)\}\end{aligned}\quad (15)$$

Donde el último término tiene un índice  $n$  dado por:

$$n = (i_{max} - i_{min} + 1) + (j_{max} - j_{min} + 1) + (k_{max} - k_{min} + 1) + 1 \quad (16)$$

Dos términos adyacentes en el conjunto  $\{\alpha\}$  pueden referirse a las intersecciones del rayo con un vóxel particular. Para dos intersecciones  $m$  y  $m - 1$ , la longitud de la intersección del vóxel  $l(m)$  se define por:

$$l(m) = d_{12}[\alpha(m) - \alpha(m - 1)] \quad (m = 1, \dots, n) \quad (17)$$

donde la cantidad  $d_{12}$  es la distancia tridimensional desde el punto 1 al punto 2:

$$d_{12} = [(X_2 - X_1)^2 + (Y_2 - Y_1)^2 + (Z_2 - Z_1)^2]^{\frac{1}{2}} \quad (18)$$

El vóxel con índices  $[i(m), j(m), k(m)]$ , correspondiente a las intersecciones mencionadas  $m$  y  $m - 1$ , contiene el punto medio de las dos intersecciones. Usando las ecuaciones (8) y (9), los índices del vóxel se definen por:

$$\begin{cases} i(m) = 1 + \frac{[X_1 + \alpha_{mid}(X_2 - X_1) - X_{plane}(1)]}{d_x} \\ j(m) = 1 + \frac{[Y_1 + \alpha_{mid}(Y_2 - Y_1) - Y_{plane}(1)]}{d_y} \\ k(m) = 1 + \frac{[Z_1 + \alpha_{mid}(Z_2 - Z_1) - Z_{plane}(1)]}{d_z} \end{cases} \quad (19)$$

Donde  $\alpha_{mid}$  se define por:

$$\alpha_{mid} = \frac{[\alpha(m) + \alpha(m - 1)]}{2} \quad (20)$$

Por lo tanto, el camino radiológico de la ecuación (7) está dado por:

$$d = \sum_{m=1}^{m=n} l(m)\rho[i(m), j(m), k(m)]$$

$$d = d_{12} \sum_{m=1}^{m=n} [\alpha(m) - \alpha(m-1)] \rho[i(m), j(m), k(m)] \quad (21)$$

El algoritmo de Siddon se puede resumir en los siguientes pasos:

1. 2% del algoritmo: Calcular el rango de valores paramétricos  $(\alpha_{min}, \alpha_{max})$ .
2. 3% del algoritmo: Calcular el rango de índices  $(i_{min}, i_{max}), (j_{min}, j_{max}), (k_{min}, k_{max})$ .
3. 16% del algoritmo: Calcular el conjunto de parámetros  $\{a_x\}, \{a_y\}, \{a_z\}$ .
4. 26% del algoritmo: Unir los conjuntos para formar el conjunto  $\{\alpha\}$ .
5. 12% del algoritmo: Calcular la longitud del vóxel.
6. 41% del algoritmo: Calcular el índice del vóxel.

Los porcentajes indican la cantidad relativa de tiempo computacional dedicado a las distintas partes del algoritmo.

### 5.3.3 Vectorización del algoritmo de Siddon

Actualmente, el algoritmo de Siddon, crucial para la creación de un DRR (Radiografía Digitalmente Reconstruida), ha experimentado una mejora significativa en la eficiencia computacional gracias a la implementación de técnicas de vectorización. Esta optimización permite una integración eficiente con bibliotecas de álgebra tensorial aceleradas por GPU, asegurando un notable incremento en la velocidad y eficiencia del proceso de renderización. La vectorización ha mejorado la capacidad del algoritmo para manejar grandes volúmenes de datos, optimizando así la generación de imágenes de alta calidad. Las GPU, con su capacidad para cálculos paralelos, se han convertido en una herramienta indispensable para ejecutar eficientemente las operaciones matemáticas complejas involucradas en el proceso (Vivek, 2022).

Para la generación del DRR, en lugar de utilizar el camino radiológico de la ecuación (7), ahora es factible usar la atenuación total de energía experimentada por los rayos X. Vivek (2022) describe un proceso de generación de un DRR mediante el modelado de un sistema de proyección radiográfica idealizado, ver Figura 3. El modelo comienza definiendo la posición de la fuente de rayos X como  $\mathbf{s} \in \mathbb{R}^3$ , y un píxel objetivo en el plano del detector como  $\mathbf{p} \in \mathbb{R}^3$ . El rayo  $R(\alpha)$ , que se extiende desde  $\mathbf{s}$  hasta  $\mathbf{p}$ , se define como  $R(\alpha) = \mathbf{s} + \alpha(\mathbf{p} - \mathbf{s})$ . Este rayo comienza en  $\mathbf{s}$  ( $\alpha = 0$ ) y viaja a través del volumen de imagen hasta alcanzar el detector en  $\mathbf{p}$  ( $\alpha = 1$ ).

La fórmula para calcular la atenuación de energía de los rayos X al alcanzar el píxel  $\mathbf{p}$  se expresa mediante una integral de línea:

$$E(R) = \|\mathbf{p} - \mathbf{s}\|_2 \int_0^1 \mathbf{V}(\mathbf{s} + \alpha(\mathbf{p} - \mathbf{s})) d\alpha \quad (22)$$

Aquí,  $\mathbf{V}$  representa el volumen a través del cual pasa el rayo, mapeando de  $\mathbb{R}^3$  a  $\mathbb{R}$ . El factor  $\|\mathbf{p} - \mathbf{s}\|_2$  escala la integral, proporcionándole la dimensión de longitud a  $d\alpha$ . En el proceso de síntesis del DRR, el volumen  $\mathbf{V}$  se aproxima a un escaneo CT tridimensional discretizado. Por lo tanto, la integral continua se aproxima mediante una suma:

$$E(R) = \|\mathbf{p} - \mathbf{s}\|_2 \sum_{m=1}^{M-1} (\alpha_{m+1} - \alpha_m) \mathbf{V} \left[ \mathbf{s} + \frac{\alpha_{m+1} + \alpha_m}{2} (\mathbf{p} - \mathbf{s}) \right] \quad (23)$$

En esta suma,  $\alpha_m$  determina los puntos donde el rayo intersecta los planos ortogonales del volumen CT, con  $M$  representando el número total de intersecciones. Nótese que, a través de esta aproximación discreta, el término  $d\alpha$  en la ecuación (22) se reemplazó simplemente por  $(\alpha_{m+1} - \alpha_m)$ , y el término  $\alpha$  por la ecuación (20), dado que  $\alpha_{mid}$  identifica el vóxel actual por el cual pasa el rayo.

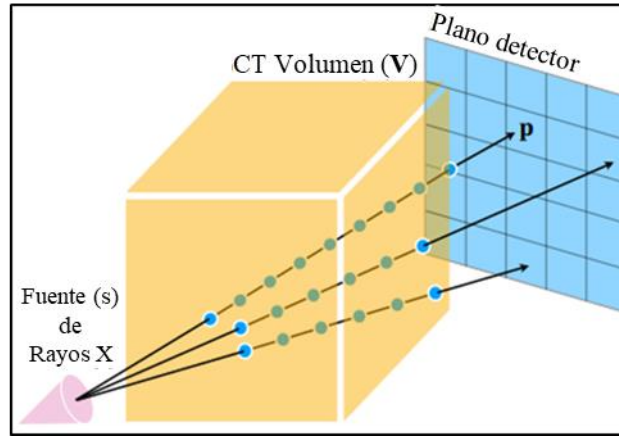


Figura 3. Geometría del generador de DRR (Vivek, 2022)

Para este caso, donde se usa la atenuación de energía, las ecuaciones (10) y (9) emplean  $X_1 = \mathbf{s}$  y  $X_2 = \mathbf{p}$ :

$$\alpha_x(i) = \frac{[X_{plane}(1) + (i - 1)d_x - \mathbf{s}_x]}{\mathbf{p}_x - \mathbf{s}_x} \quad (i = 1, \dots, N_x) \quad (24)$$

Para conveniencia, al primer plano se le asignará un índice de  $i = 0$ , por lo que  $X_{plane}(1)$  se referirá como  $b_x$ . El término  $d_x$  se denotará como  $\Delta X$ , para enfatizar que  $d_x$  es la longitud del vóxel en la dirección del eje x. Así, la ecuación (24) se da por:

$$\alpha_x(i) = \frac{[b_x + i\Delta X - s_x]}{p_x - s_x} \quad (i = 0, \dots, n_x) \quad (25)$$

con expresiones similares para  $\alpha_y(i)$ , y  $\alpha_z(i)$ . Con la ecuación (25), el algoritmo de Siddon se puede utilizar para obtener el conjunto  $\{\alpha\}$ . Sustituimos valores en el conjunto ordenado  $\{\alpha\}$  en la ecuación (23) para evaluar  $E(R)$ , que corresponde a la intensidad del píxel  $\mathbf{p}$  en el DRR sintetizado (Vivek, 2022).

La ecuación (25) es la que se debe vectorizar. Sea  $\mathbf{P} \in \mathbb{R}^{H \times W \times 3}$  conteniendo las coordenadas de vóxel 3D de un DRR con dimensión  $H \times W$ . Los valores de  $\alpha$  se calculan para las intersecciones con todos los planos  $x, y, z$  para todos  $\mathbf{p} \in \mathbf{P}$ . Por lo tanto, la ecuación (25) será:

$$\mathbf{A} = \left( \begin{pmatrix} b_x \\ b_y \\ b_z \end{pmatrix} + \begin{pmatrix} i \\ j \\ k \end{pmatrix} \otimes \begin{pmatrix} \Delta X \\ \Delta Y \\ \Delta Z \end{pmatrix} - \mathbf{s} \right) \oslash (\mathbf{P} - \mathbf{s}) \in \mathbb{R}^{H \times W \times (n_x + n_y + n_z)} \quad (26)$$

donde  $(n_x + n_y + n_z)$  son las dimensiones del volumen CT  $\mathbf{V}$ ,  $(i, j, k)$  son los índices de los vóxeles CT,  $(\Delta X, \Delta Y, \Delta Z)$  son los tamaños de los vóxeles CT, y  $\otimes$  y  $\oslash$  son los operadores de producto y división de Hadamard, respectivamente. En contraste con el algoritmo de Siddon, los valores  $(i_{min}, i_{max})$ ,  $(j_{min}, j_{max})$ , y  $(k_{min}, k_{max})$  ya no se calculan explícitamente. En su lugar, se calculan los valores mínimos y máximos de  $\alpha$ , correspondientes a cuando cada rayo entra y sale del volumen:

$$\begin{cases} \alpha_{min} = \max\{\min[a_x(0), a_x(n_x)], \min[a_y(0), a_y(n_y)], \min[a_z(0), a_z(n_z)]\} \\ \alpha_{max} = \min\{\max[a_x(0), a_x(n_x)], \max[a_y(0), a_y(n_y)], \max[a_z(0), a_z(n_z)]\} \end{cases} \quad (27)$$

Donde  $\alpha_{min}, \alpha_{max} \in \mathbb{R}^{H \times W}$ . El tensor  $\mathbf{A}$  se filtra para incluir solo valores en el rango  $[\alpha_{min}, \alpha_{max}]$ , y cada fila  $\mathbf{A}(h, w, \cdot)$  se ordena para  $h \in \{1, \dots, H\}$ ,  $w \in \{1, \dots, W\}$ . Finalmente, se evalúa la ecuación (23) con este tensor ordenado  $\mathbf{A}(h, w, \cdot)$  para calcular la intensidad de cada píxel en la DRR, completando una cadena de operaciones tensoriales vectorizadas (Vivek, 2022).

La posición 3D de la fuente de rayos X y el plano detector con relación con el volumen de CT se describen con los siguientes parámetros geométricos: radio  $\rho$ , actúa como un factor de escala; tres grados de libertad rotacionales  $(\theta, \varphi, \gamma)$ ; y tres grados de libertad traslacionales  $(bx, by, bz)$ . La posición de la fuente de rayos X, si se expresa en coordenadas esféricas, es  $\mathbf{s} = (\rho, \theta, \varphi)$ , donde  $\rho$  corresponde a la mitad de la distancia entre la fuente y el detector, mientras que  $\theta$ , y  $\varphi$  son los ángulos azimutal y polar, respectivamente. El plano detector, por su parte, se configura de manera que sea tangente a una esfera implícita en el punto diametralmente opuesto a  $\mathbf{s}$ . La orientación de este plano está determinada por una rotación alrededor del eje X por el ángulo  $\gamma$ . Anteriormente se añade la traslación  $(bx, by, bz)$  a las coordenadas de la fuente de rayos X, de manera que la imagen proyectada se centre en la posición actual de la fuente.

Vivek (2019) afirma: «Dado que reformulamos el método original de Siddon como una serie de operaciones tensoriales, nuestra versión vectorizada se beneficia de compiladores de GPU maduros y asignadores de memoria desarrollados para optimizar modelos de aprendizaje profundo a gran escala» (p. 5).

La calidad de la DRR está afectada por una serie de factores, los cuales están asociados no solo con la recolección de datos, sino también con los algoritmos de reconstrucción:

- **Volumen de Datos de CT:** El conjunto de datos de CT es crucial para definir el haz de radiación y llevar a cabo la simulación virtual. Es esencial que estos datos no solo cubran la superficie del cuerpo, sino que también abarquen toda el área que será tratada. Esto es particularmente importante, ya que el área de tratamiento puede extenderse más allá del tumor (Mayles, 2022).
- **Espesor de Corte:** La medida del espesor en los cortes de tomografía computarizada (CT) juega un papel crucial en la definición de la resolución a lo largo del eje longitudinal, estableciendo así un estándar concreto para la calidad de la Reconstrucción Radiográfica Digital (DRR). Sin embargo, la incorporación de escáneres CT helicoidales de múltiples cortes ha hecho que sea raro que el espesor de corte supere los 3 mm, siendo potencialmente tan pequeño como 0.75 mm, lo que asegura una DRR de alta calidad (Mayles, 2022).
- **Técnica de Trazado de Rayos:** Esto se refiere al método utilizado para realizar la suma. Esta técnica se basa en el uso de la intensidad del vóxel de CT más cercano para facilitar una reconstrucción rápida (Mayles, 2022).
- **Tamaño del Paso:** Este es el tamaño del incremento utilizado a lo largo de las líneas de rayos al sumar los coeficientes de atenuación para generar la DRR (Mayles, 2022).

## **5.4 Radioterapia corporal estereotáctica (SBRT)**

La Radioterapia Corporal Estereotáctica (SBRT) se utiliza en el tratamiento del cáncer de pulmón, particularmente en sus etapas iniciales. El cáncer de pulmón de células no pequeñas (NSCLC, por sus siglas en inglés) es el tipo más común de cáncer de pulmón y afecta a un número considerable de pacientes. Aproximadamente el 15-20% de los pacientes con NSCLC son diagnosticados en una etapa temprana, lo que significa que el cáncer está localizado y no se ha propagado. Anteriormente, la cirugía era la opción preferida para estos pacientes, pero algunos no pueden someterse a cirugía debido a complicaciones médicas o simplemente eligen no hacerlo (Martin, 2017).

En el pasado, la radioterapia fraccionada convencional era una alternativa al tratamiento quirúrgico, pero no proporcionaba un buen control local del tumor, lo que repercutía en resultados inferiores en comparación con la cirugía. La Radioterapia Corporal Estereotáctica (SBRT) es una técnica más moderna que tiene como objetivo administrar

menos fracciones de radiación, pero con dosis más altas. Varios estudios han demostrado que, al aumentar la dosis biológicamente efectiva por encima de cien Gy, se puede reducir el número de casos de recurrencia local y mejorar la supervivencia general en pacientes con NSCLC en etapa temprana en comparación con la radioterapia fraccionada convencional. Esto ha llevado a que la SBRT se convierta en el nuevo estándar de atención para pacientes inoperables con NSCLC en etapa temprana (Martin, 2017).

La SBRT pulmonar implica administrar altas dosis de radiación al tumor en pocas fracciones, en contraste con la radioterapia convencional. Para lograr una alta precisión en el tratamiento, es necesario considerar el movimiento del tumor dentro del pulmón, ya que esto puede afectar la entrega precisa de la radiación. Si las incertidumbres relacionadas con el movimiento no se consideran adecuadamente, pueden ocurrir fallos en la distribución de la dosis, como una cobertura inexacta del objetivo y una exposición excesiva de los tejidos sanos circundantes. Por lo tanto, en la radioterapia moderna del pulmón, es crucial controlar el movimiento del tumor y los cambios en la posición del paciente durante el tratamiento, con un conocimiento preciso de la posición del tumor para garantizar la eficacia del tratamiento y la seguridad del paciente. Los avances en las tecnologías de imagen y seguimiento continúan mejorando la precisión y efectividad de la SBRT (Caillet, 2017).

## 6. Metodología

### 6.1 GPU

#### 6.1.1 Implementación de unidades de procesamiento gráfico (GPU)

Desde el inicio de la computación, los procesadores estaban limitados a un solo núcleo. Esto significaba que cada procesador tenía una única unidad de procesamiento capaz de gestionar una tarea a la vez, ejecutando instrucciones secuencialmente; una tras otra. Con el avance de la tecnología, las Unidades Centrales de Procesamiento (CPU) modernas han evolucionado para incluir múltiples núcleos, lo que les permite manejar diversas tareas simultáneamente. Sin embargo, la fortaleza de las CPU radica en su capacidad para utilizar un alto rendimiento en cada núcleo individual para llevar a cabo tareas computacionales complejas. Debido a esta característica, las CPU no se centran en tener tantos núcleos como las Unidades de Procesamiento Gráfico (GPU), priorizando en cambio el alto rendimiento por núcleo (Intel, s.f.).

En cuanto a las GPU, este procesador, a diferencia de la CPU, consta de muchos núcleos pequeños y especializados (menos potentes en comparación con los de una CPU). Las GPU pueden tener cientos e incluso miles de núcleos trabajando. Este tipo de procesador sigue una arquitectura de computación paralela, siendo capaz de realizar múltiples cálculos o ejecuciones de procesos simultáneamente (Intel, s.f.). Las GPU son bastante útiles, especialmente en tareas que pueden dividirse en muchas operaciones más pequeñas que se ejecutan en paralelo. Por lo tanto, esta herramienta (GPU) es más eficiente en comparación con la CPU para manejar operaciones de matrices y vectores simultáneamente. Además, aunque los núcleos individuales en una GPU pueden no ser tan potentes como un núcleo de CPU, la capacidad de realizar muchas operaciones al mismo tiempo permite que las GPU sean extremadamente eficientes en términos de rendimiento y consumo de energía (Pratx, 2011).

Con el tiempo, las GPU se han convertido en dispositivos más programables y adaptables. CUDA (Compute Unified Device Architecture), un modelo de programación creado por NVIDIA, ha permitido a los programadores usar las GPU para la computación de propósito general (GPGPU). Este modelo opera a través de una interfaz de programación directa para GPU proporcionada por CUDA. En el entorno de Python, la biblioteca Pytorch destaca por ofrecer compatibilidad con CUDA, facilitando a los programadores el uso de GPU compatibles con CUDA en sus códigos (NVIDIA, s.f.).

La implementación de las GPU en el campo de la física médica abarca varias aplicaciones esenciales, incluyendo la reconstrucción avanzada de imágenes, precisión en el cálculo de dosimetría, optimización de planes terapéuticos, así como procesamiento sofisticado de imágenes médicas (Khor, 2015). En el contexto de este proyecto específico, la GPU desempeñará un papel crucial al facilitar la generación de Radiografías Reconstruidas Digitalmente (DRR) en tiempos significativamente reducidos. Esta capacidad es fundamental, considerando que en la próxima etapa de este proyecto (en el programa de maestría) se requerirá producir más de 10,000 imágenes DRR, que posteriormente serán integradas en una red neuronal convolucional para análisis y aplicaciones avanzadas (Hirai, 2019).

### **6.1.2 Google Colab**

En el código para este proyecto de tesis, se requiere una GPU para generar la DRR, utilizando Pytorch. Para ejecutar el generador de DRR utilizado en este proyecto, como mínimo, la computadora debe tener una tarjeta de video NVIDIA RTX 2080 Ti con 11 GB de memoria. Dado que no es fácil disponer de una computadora con este tipo de tarjeta de video, se eligió Google Colab como alternativa. Google Colab es una herramienta de programación basada en la nube que permite escribir y ejecutar código en Python. Ofrece acceso a recursos de computación, incluidas GPUs y Unidades de Procesamiento Tensorial (TPUs) con una memoria de hasta 15.84 GB. A través de Google Colab, se utilizaron las siguientes tarjetas de video: NVIDIA Tesla T4 con 15.84 GB, NVIDIA Tesla V100 con 32 GB, L4 con 24 GB y NVIDIA A100 con 80 GB. Estas GPUs se utilizaron mediante la interfaz CUDA 2.1.0+cu118 (Google Colab, s.f.).

## **6.2 Archivos DICOM**

### **6.2.1 Caracterización de imágenes 4DCT en formato DICOM**

Las imágenes se tomaron utilizando un escáner helicoidal Philips BigBore de 16 cortes. Este es un escáner favorable para aplicaciones de radioterapia, ya que su gran apertura de 80 cm puede acomodar una amplia variedad de configuraciones del paciente (incluyendo brazos arriba, piernas en posición de rana y pacientes de gran tamaño), así como su verdadero campo de visión de 60 cm permite una reconstrucción libre de artefactos lejos del eje central de la imagen. Las imágenes se recolectaron con un sistema de fase de sincronización utilizando el proceso de reconstrucción 4D, que creó 10 conjuntos de imágenes 3D, cada uno representando una fase diferente de la respiración del paciente.

## 6.2.2 Extracción y análisis de metadatos DICOM

Los metadatos DICOM constituyen un conjunto estandarizado de datos adjuntos a cada imagen médica en el formato DICOM. Estos metadatos son cruciales para la correcta interpretación y gestión de las imágenes médicas, ya que proporcionan información esencial. A través de los metadatos, es posible acceder a varios detalles sobre el paciente, como su nombre, identificación, fecha de nacimiento y género. También incluyen información relevante sobre el estudio, la fecha, la hora, el médico remitente y una descripción del mismo. Además, contienen datos sobre la institución, el seguimiento del paciente y el control de calidad.

Sin embargo, para este proyecto, los elementos más cruciales de los metadatos son los datos técnicos de la imagen. Estos incluyen la modalidad (tomografía computarizada, resonancia magnética o rayos X), dimensiones de la imagen, resolución y parámetros específicos relacionados con la adquisición de la imagen. La biblioteca Pydicom juega un papel importante, facilitando tanto el acceso como la gestión de estos metadatos dentro de cualquier entorno de código Python. Esta herramienta es fundamental para la consulta y aplicación efectiva de los metadatos en los procesos de análisis y gestión de imágenes médicas.

En DICOM, las secuencias son estructuras de metadatos que agrupan elementos relacionados para organizar y presentar información de manera estructurada. Las secuencias relevantes para este proyecto son:

- **ROIContourSequence:** Utilizada en radioterapia, esta secuencia detalla los contornos de las Regiones de Interés (ROI) en imágenes médicas, particularmente para el módulo de CT. Cada elemento describe un contorno de ROI, incluyendo detalles sobre la forma, posición, dimensiones en la imagen e información de referencia (aunque esto no es lo principal de esta secuencia). Los metadatos de ReferencedROI Number se utilizan para vincular los contornos de ROI definidos en la secuencia, y ContourData, que indica las coordenadas físicas de cada punto en la secuencia de contorno de ROI (Medical Imaging & Technology Alliance, s.f.).
- **StructureSetROISequence:** Proporciona información enfocada en la gestión de información descriptiva y de catalogación de las ROI, más que en detalles físicos o técnicos. Los metadatos clave para el proyecto fueron ROI Name y ROI Number (Medical Imaging & Technology Alliance, s.f.).

Además, se utilizan metadatos que no están contenidos dentro de secuencias, sino que actúan como atributos independientes:

- **ImagePositionPatient:** Define la posición tridimensional del origen de la imagen, es decir, la esquina superior izquierda del primer píxel, en relación con un sistema de coordenadas físicas en milímetros (Medical Imaging & Technology Alliance, s.f.).

- **SliceLocation:** Indica la posición de un corte individual dentro de una serie de imágenes en CT, en coordenadas físicas en milímetros (Medical Imaging & Technology Alliance, s.f.).
- **PixelSpacing:** Representa dos mediciones en milímetros que describen la distancia entre los centros de píxeles adyacentes en las direcciones horizontal y vertical (Medical Imaging & Technology Alliance, s.f.).

## 6.3 Determinación del centro de masa

### 6.3.1 Procesamiento de metadatos DICOM para la localización del tumor

En la gestión de archivos DICOM, se utiliza un archivo conocido como RTSTRUCT. Este archivo contiene los contornos delineados por el profesional y su equipo de diagnóstico (oncólogo, técnico de laboratorio y/o físico médico) en las imágenes de CT. Estos contornos, trazados meticulosamente, abarcan una variedad de órganos visualizados en la CT, así como los tumores identificados. Cada contorno se distingue por un número y un nombre únicos, facilitando su identificación y gestión.

Debido a la importancia de identificar con precisión los contornos relacionados con tumores, se desarrolló un algoritmo específico para este propósito. El objetivo del algoritmo es localizar y seleccionar, dentro del archivo RTSTRUCT, el contorno ROI correspondiente al tumor. Este proceso comienza con la consulta de la *StructureSetROISequence*, de la cual se extraen el *ROI Number* y el *ROI Name* de cada contorno. El algoritmo verifica que el *ROI Name* comience con la abreviatura "GTV," que se refiere al Volumen Tumoral Macroscópico (Gross Tumor Volume). Al identificar el *ROI Number* asociado con un *ROI Name* que comienza con "GTV," este número se almacena en una variable llamada *roinumber*.

Posteriormente, el algoritmo procede a consultar la *ROI Contour Sequence*. Durante esta fase, se revisa y compara el *ReferenceROI Number* de esta secuencia con el *roinumber* almacenado previamente. Al encontrar una coincidencia entre el *ReferenceROI Number* y el *roinumber*, este número se registra en una variable llamada *ROI Number Sequence*.

Finalmente, a través del *ROI Number Sequence*, se obtienen los datos del contorno (*Contour Data*), que proporcionan la posición exacta de cada punto del contorno que rodea al tumor en cada sección transversal del cuerpo. Con esta información detallada, es posible calcular el centro de masa del tumor utilizando un método específico descrito en la siguiente sección. Este proceso garantiza la identificación precisa y eficiente de los contornos relevantes para el tratamiento y análisis de tumores.

### 6.3.2 Metodología para el cálculo del centro de masa del tumor

El proceso comienza con la identificación de los límites máximos y mínimos en los ejes X, Y y Z del volumen que demarca el contorno que rodea al tumor. Esto se hace utilizando cada contorno del tumor extraído del archivo RTSTRUCT. Este análisis implica determinar los valores extremos de las posiciones de los puntos en los contornos en cada uno de los ejes mencionados. Identificar estos valores es necesario, ya que permite establecer un marco de referencia espacial esencial para los cálculos subsiguientes. Una vez definidos estos límites, se crea una rejilla de coordenadas que cubre completamente el área delimitada por el polígono que define el contorno alrededor del tumor. Esta rejilla asegura una representación precisa del área de interés para estudios y análisis posteriores.

Posteriormente, se crea una máscara binaria. Esta consiste en una imagen formada por píxeles de solo dos colores, típicamente blanco y negro. En ella, los píxeles blancos indican áreas fuera del contorno, mientras que los píxeles negros delimitan el área que rodea al contorno (Whybra, 2023). Para la creación de esta máscara binaria, es necesario generar polígonos cerrados para cada contorno. Un polígono es una figura geométrica plana que se forma al conectar una serie de puntos con líneas rectas. Estos puntos se llaman vértices, y las líneas rectas son los lados del polígono. Luego, se evalúa qué puntos de la rejilla de coordenadas creada previamente están ubicados dentro de este polígono; así, los puntos dentro del polígono se marcan con 1 (indicando la presencia de tejido tumoral) y los puntos fuera del polígono con 0. Estas máscaras binarias son fundamentales para el cálculo posterior del centro de masa (Whybra, 2023).

Con la rejilla de coordenadas creada y la máscara binaria en su lugar, se realiza la operación en ambos tensores mediante una multiplicación de Hadamard. Esto implica multiplicar elemento por elemento correspondiente a cada tensor. Esta operación se puede simplificar con la siguiente ecuación tridimensional  $\mathbb{R}^3$ .

$$\vec{r} = \sum_i m_i \vec{r}_i \quad (28)$$

Donde  $m_i \in \{0,1\}$  es el valor de la máscara binaria en el punto  $i$ , y  $\vec{r}_i$  es la posición vectorial física de dicho punto. Descomponemos la ecuación tridimensional (28) en sus componentes correspondientes a los ejes X, Y y Z. Luego, para cada componente, dividimos por la masa de la máscara binaria M, que se define como la suma de los valores presentes en las máscaras binarias. Este procedimiento nos permite calcular la posición del centro de masa del tumor:

$$\begin{cases} x_{cm} = \frac{1}{M} \sum_i m_i x_i \\ y_{cm} = \frac{1}{M} \sum_i m_i y_i \\ z_{cm} = \frac{1}{M} \sum_i m_i z_i \end{cases} \quad (29.a)$$

En las imágenes de CT, se pueden gestionar dos tipos de marcos de referencia relacionados: el marco de referencia del espacio de píxeles y el marco de referencia del espacio físico. El marco de referencia del espacio de píxeles se relaciona con la forma en que los píxeles están organizados en la imagen. Cada píxel representa un área específica del cuerpo escaneado. El marco de referencia del espacio físico se relaciona con la posición y orientación de la imagen en relación con el cuerpo del paciente durante el escaneo (Medical Imaging & Technology Alliance, s.f.).

La ecuación (29) describe la ubicación del centro de masa del tumor en el sistema de coordenadas del espacio físico, expresado en unidades de milímetros.

En el generador DRR, es necesario tener las posiciones en el marco de referencia del espacio de píxeles. Para convertir la posición del centro de masa en el espacio físico al espacio de píxeles, se utilizan las siguientes ecuaciones.

$$\begin{cases} i_{cm} = \frac{x_{cm} - IPP_x}{PS_x} \\ j_{cm} = \frac{y_{cm} - IPP_y}{PS_y} \\ k_{cm} = \frac{Z_{max} - z_{cm}}{PS_z} \end{cases} \quad (29.b)$$

Donde  $IPP_x$  es la posición otorgada por ImagePositionPatient en el eje X,  $IPP_y$  es la posición otorgada por ImagePositionPatient en el eje Y,  $Z_{max}$  es la posición superior del corte en el eje Z,  $PS_x$  es el espacio en milímetros entre píxeles en el eje X,  $PS_y$  es el espacio en milímetros entre píxeles en el eje Y, y  $PS_z$  es la distancia en milímetros entre cortes CT consecutivos.

Este método fue validado utilizando una técnica alternativa implementada en el software 3D Slicer, específicamente a través del análisis estadístico de segmentos. Para determinar el centro de masa en 3D Slicer, se debe acceder a la sección 'Segment Statistics' y verificar la posición del centroide de cada segmento. Este proceso emplea el concepto de **centro geométrico**, que se define como la posición promedio de todos los puntos incluidos en el segmento. Consecuentemente, representa el centro geométrico de todos los vóxeles que constituyen el segmento (3D Slicer, 2020).

## 6.4 Generador de radiografía reconstruida digitalmente (DRR)

### 6.4.1 Preparación de metadatos DICOM para la generación de DRR

Antes de aplicar el generador de DRR, es necesario calcular una matriz que represente el volumen corporal tal como se observa en la Tomografía Computarizada (CT). Este volumen se obtiene a partir de los metadatos del estándar DICOM. Es importante destacar que el proceso descrito aquí implica una modificación del código API de la biblioteca diffDRR. Esta adaptación fue necesaria para superar una restricción en el código original, que no soportaba la lectura de archivos DICOM que incluyen un tipo de archivo RTSTRUCT, además se logró una optimización del tiempo de ejecución del algoritmo. Por lo tanto, se ajustó el módulo data.py de la API de diffDRR para procesar cualquier archivo de Tomografía Computarizada Cuatro Dimensional (4DCT). Por protocolo, estos archivos generalmente se almacenan con tipos de archivo CT y RTSTRUCT.

Para construir una representación tridimensional del cuerpo a partir de una tomografía computarizada, el proceso implica ensamblar archivos DICOM, que son cortes individuales de la exploración. Inicialmente, estos cortes se organizan en la secuencia correcta según su metadato *SliceLocation*, organizándose desde la posición más alta hasta la más baja a lo largo del eje Z. De esta manera, la alineación comienza desde el corte más superior y progresa hacia abajo para asegurar que cada corte esté correctamente indexado e integrado, manteniendo la precisión y la integridad de las imágenes CT.

La construcción del volumen comienza con la creación de una matriz tridimensional, cuyas dimensiones están definidas por el número de píxeles en cada imagen CT (en las direcciones X e Y) y por el número total de imágenes (en la dirección Z). Aquí es donde el *pixel\_array* juega un papel importante. Al leerse cada archivo DICOM, se puede obtener un objeto que incluye la propiedad *pixel\_array*, que proporciona una matriz bidimensional de los valores de intensidad de los píxeles de la imagen correspondiente. Cada número en esta matriz representa la intensidad de un píxel en la imagen CT, reflejando diferencias en la densidad de los tejidos escaneados. Estos valores son esenciales para el modelo tridimensional, ya que permiten una representación detallada y diferenciada de varias estructuras anatómicas.

El código procede a llenar la matriz tridimensional con los datos de estos píxeles. Cada matriz bidimensional extraída del *pixel\_array* de una imagen CT se asigna a una capa correspondiente en la matriz tridimensional, manteniendo el orden correcto de las imágenes. Este proceso de apilamiento culmina en la formación de un modelo volumétrico completo del área escaneada.

Finalmente, se consulta el espaciamiento de los píxeles en las direcciones X, Y y Z. Para las dimensiones XY, simplemente se consulta el metadato *PixelSpacing*. En cuanto a la dirección Z, esto se obtiene midiendo la distancia entre cada corte CT consecutivo. Se calcula observando las posiciones Z de las imágenes y encontrando la diferencia entre ellas.

## 6.4.2 Establecimiento del marco de referencia del volumen CT para el generador de DRR

Como se puede observar en la Figura 4, diagramas a y b, el marco de referencia en el cuerpo del volumen CT se explica de la siguiente manera (ver la posición del cuerpo en la ilustración):

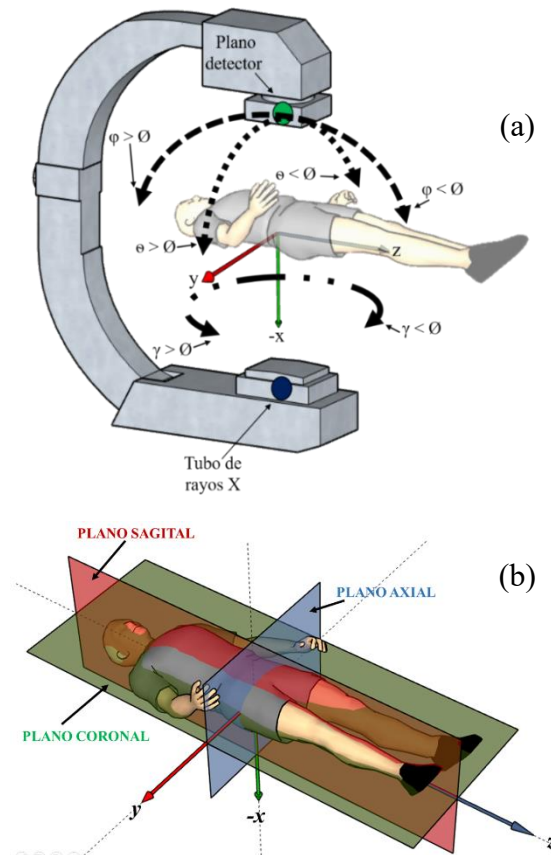


Figura 4. Marco de referencia de tomografía para generar la RRD

- Eje Y: Este eje, convencionalmente, va desde el hombro izquierdo al derecho (lateralmente-lateral) a través del cuerpo por el cual corre el plano sagital (ver diagrama b).
- Eje X: Se refiere al eje que cruza el cuerpo desde la espalda hasta el vientre (dorsoventral) por el cual corre el plano coronal (ver diagrama b).
- Eje Z: Se extiende desde la cabeza hasta los pies (céfalo-podal) a través del cuerpo por el cual corre el plano axial (ver diagrama b).

- El sistema de los rayos X y del detector será un fluoroscopio de brazo en C. Por lo tanto, los grados de libertad de rotación del fluoroscopio de brazo en C son:  $\theta$  es el ángulo acimutal,  $\varphi$  es el ángulo polar y  $\gamma$  es el ángulo de rotación (ver diagrama a).
- Usándose estos ángulos, en el proyecto se configuró para generar tres imágenes: una imagen en el plano sagital  $(-\frac{\pi}{2}, 0, \frac{\pi}{2})$ , una imagen en el plano axial  $(\frac{\pi}{2}, \frac{\pi}{2}, \frac{\pi}{2})$  y una imagen en el plano coronal  $(\pi, 0, \frac{\pi}{2})$ .

### 6.4.3 Implementación del generador de DRR usando la biblioteca DiffDRR

Una vez que se tiene el volumen CT en una matriz tridimensional y el espaciamiento de píxeles en las direcciones X, Y y Z, es posible utilizar la biblioteca diffDRR. Con esta biblioteca, se creará un objeto DRR cuyos parámetros son:

- Volumen CT
- El espaciamiento entre píxeles en las tres direcciones
- Distancia entre la fuente de rayos X y el detector
- Altura y anchura del DRR

Usándose este objeto, se utilizará el método **drr()** del objeto DRR. Este método requiere dos parámetros: el tensor de rotación de la máquina de rayos X con un arco en C, el cual se representa de la siguiente manera:

$$\text{Rotación} = (\theta, \varphi, \gamma)$$

$\theta$  es el ángulo acimutal,  $\varphi$  es el ángulo polar y  $\gamma$  es el ángulo de rotación. El tensor de traslación tridimensional de la máquina de rayos X se representa de la siguiente manera:

$$\text{Traslación} = (bx, by, bz)$$

En el tensor de traslación, en el método **drr()** para generar la imagen, siempre se asigna la posición tridimensional del centro de masa del tumor calculado en la ecuación (29.b). Luego, el método **drr()** del objeto DRR procede a usar el método de Siddon vectorizado para generar una imagen DRR centrada en el centro de masa del tumor. En este proyecto, para cada archivo DICOM 4DCT de un paciente, se generan tres imágenes DRR correspondientes a tres planos diferentes: axial, sagital y coronal. Estas imágenes se almacenarán en el conjunto de datos para ser iteradas posteriormente en un DataLoader para el entrenamiento futuro de un algoritmo de aprendizaje profundo convolucional.

En el artículo de Vivek (2022), la precisión del generador de DRR fue evaluada comparándose con un método alternativo del paquete de código abierto *Plastimatch*, que también produce DRR utilizando el procesador central (CPU). La comparación se basó en la intensidad de las imágenes DRR generadas, obteniendo un error cuadrático medio (RMSE) de  $(8.3 \pm 1.9) \times 10^{-4}$ .

## 6.5 Caracterización del ruido en imágenes DRR

### 6.5.1 Caracterización del ruido en imágenes DRR

En la fluoroscopia, ciertas propiedades son esenciales para caracterizar el ruido presente en las cadenas de imágenes de tomografía computarizada (CT). La gestión del ruido se facilita mediante el análisis de la función de transferencia de modulación. Esto proporciona medidas cuantitativas de la capacidad del sistema de imagen para mantener la claridad y nitidez de los bordes y los detalles más finos. Además, la eficiencia de la detección de los fotones juega un papel crucial, ya que mide la efectividad con la que un sistema de imagen convierte la señal de radiación recibida más reconocible (Chan, 1990).

El moteado cuántico representa un tipo de ruido inherente en la imagenología médica que proviene directamente de las propiedades fundamentales de la luz. Este ruido se origina a partir de fluctuaciones estadísticas en la distribución de los fotones emitidos por la fuente de rayos X. La llegada de fotones al detector se describe mediante una distribución de Poisson, que explica la variabilidad en el número de fotones detectados en un período fijo. Debido a esta distribución aleatoria, el número de fotones capturados en diferentes áreas del detector puede variar significativamente, resultando en una textura granulada en las imágenes de CT obtenidas (Chan, 1990).

Este ruido es crítico en el equilibrio entre la exposición a la radiación y la calidad de la imagen. Esto permitirá evitar dos extremos, que serían: El innecesario aumento del número de fotones emitidos por la fuente, y que por ende incrementa la exposición del paciente a los rayos X con el fin de mejorar la calidad de la imagen al reducir la apariencia del ruido cuántico. Por el contrario, reducir el número de fotones disminuye la exposición a la radiación pero aumenta los niveles de ruido en la imagen final, comprometiendo su calidad (Chan, 1990).

Para simular el ruido originado por el fenómeno del moteado cuántico en imágenes médicas, se aplicó una distribución de Poisson a cada píxel en la DRR. En cuanto al ruido asociado, que se gestiona a la nitidez y claridad de la imagen, se aplicará una convolución Gaussiana sobre la DRR. Este método ajusta efectivamente la dispersión de la imagen, emulando las limitaciones reales del sistema de imagen.

## 6.5.2 Ruido de Poisson en DRR

Una variable aleatoria  $X$ , que toma uno de los valores  $0, 1, 2, \dots$ , se dice que es una variable aleatoria de Poisson con parámetro  $\lambda$ , si la función de masa de probabilidad es de la forma:

$$f(X) = \frac{\lambda^X e^{-\lambda}}{X!} \quad (30)$$

donde  $\lambda > 0$ . Esta ecuación (30) es conocida como la distribución de Poisson, permitiendo calcular la probabilidad de que ocurran exactamente  $X$  eventos dentro de un intervalo de tiempo o espacio dado (Hogg, 2015).

En la etapa de procesamiento de imágenes DRR, se utilizó la función `torch.poisson()` de PyTorch para simular el ruido de Poisson. Esta función marcha tomando el valor de cada píxel en la imagen y usando este valor como  $\lambda$  para la distribución de Poisson correspondiente a ese píxel. Posteriormente, se selecciona una muestra de esta distribución para cada píxel individualmente (Pytorch, n.d.).

La distribución de Poisson se utiliza para modelar fenómenos que tienen una probabilidad fija dentro de un intervalo de tiempo específico. Representa con precisión el comportamiento de la detección de radiación en el mundo real, caracterizado por interacciones aleatorias. En este contexto, el número de fotones detectados depende de la cantidad que escapa del material e interactúa con el detector. Sin embargo, el proceso matemático de la Reconstrucción de Radiografía Digital (DRR) no considera esta aleatoriedad. Por lo tanto, la incorporación del ruido de Poisson como un paso de post procesamiento mejora el realismo de la imagen, reflejando mejor la naturaleza estocástica de la detección de fotones.

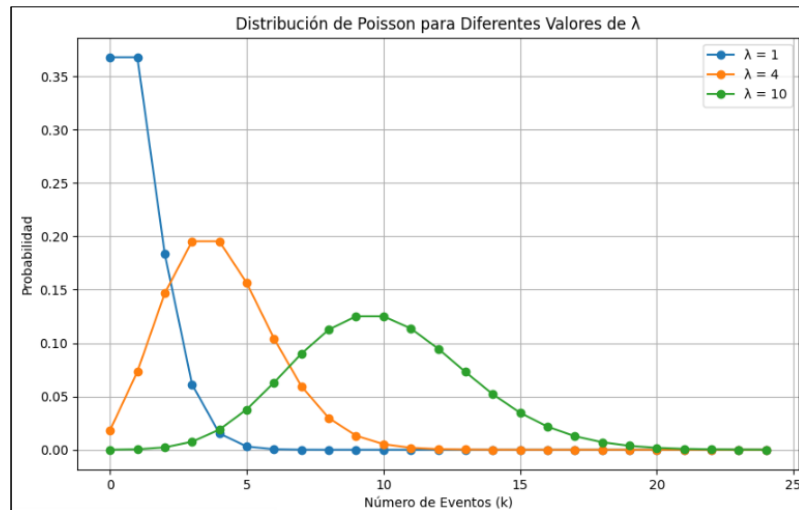


Figura 5. Gráfico de la Distribución de Poisson para Diferentes Valores de  $\lambda$

Este método implica que, para un píxel con un valor específico de  $\lambda$ , por ejemplo, 10, hay una mayor probabilidad de seleccionar números cercanos a este valor, como 9, 8, 11 o 12. Esto se debe a que, en la distribución de Poisson, los números más cercanos al valor de  $\lambda$  tienen una mayor probabilidad de ocurrencia, como se puede observar en la Figura 5. Así, esta selección se realiza de forma independiente para cada píxel. Un aspecto significativo de la función **torch.poisson()** es su capacidad para utilizar la GPU, lo que acelera significativamente el proceso de cálculo.

### 6.5.3 Ruido Gaussiano en DRR

A diferencia del ruido de Poisson, existe una limitación inherente en los detectores, ya que ningún detector mide exactamente el número de fotones de manera perfecta. Comportamientos físicos como la dispersión, el tamaño del detector y el volumen sensible del detector implican que las proyecciones en kilovoltios (kV) medidas con un detector físico no registran la atenuación total a lo largo de una línea directa. En cambio, miden una imagen que es difuminada por estos procesos físicos. Este efecto puede modelarse utilizando un filtro gaussiano, que puede calibrarse determinando la función de dispersión del punto del sistema de detección de imágenes. Este enfoque ayuda para tener en cuenta las desviaciones causadas por la dispersión en el proceso de imagen.

Este proceso implica realizar una suma ponderada de productos entre una imagen  $I$  y un Kernel  $\omega(s, t)$ , también referido como un filtro. El Kernel  $\omega(s, t)$ , que es una matriz, determina el vecindario de interacción con la imagen  $I$  a través del tamaño de la matriz. Los coeficientes de esta matriz determinan su efecto funcional en la imagen  $I$  (Woods, 2018).

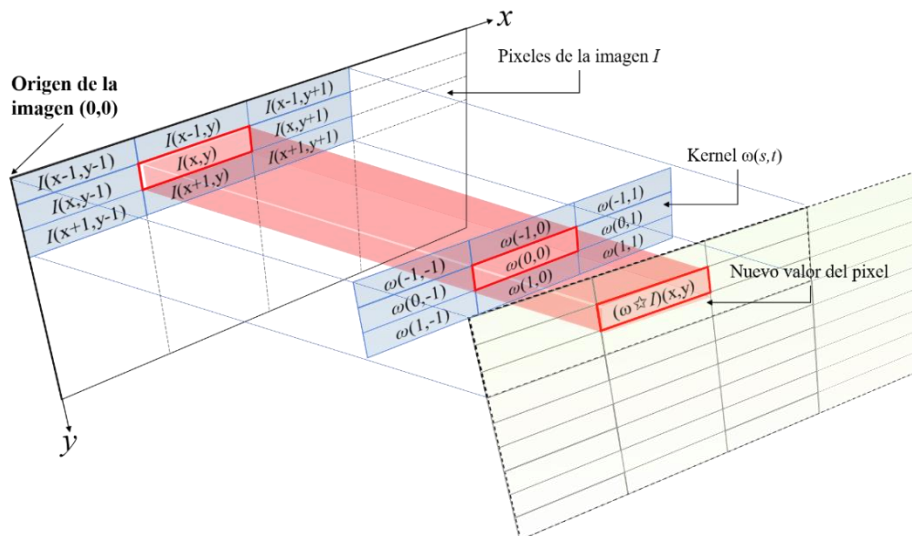


Figura 6. Ilustración de la mecánica de la operación de correlación espacial entre una imagen  $I$  y un Kernel gaussiano  $\omega(s, t)$

La Figura 6 ilustra la mecánica de la correlación espacial utilizando un Kernel  $3 \times 3$ . En cualquier punto  $(x, y)$  de la imagen, la respuesta del Kernel,  $g(x, y)$ , es la suma de los productos de los coeficientes del Kernel y los píxeles de la imagen cubiertos por el tamaño del Kernel:

$$g(x, y) = \omega(-1, -1)I(x - 1, y - 1) + \omega(-1, 0)I(x - 1, y) + \dots + \omega(0, 0)I(x, y) + \dots + \omega(1, 1)I(x + 1, y + 1) \quad (31)$$

A medida que las coordenadas  $x$  &  $y$  varían, el centro del Kernel se desplaza de un píxel de la imagen a otro, generando una nueva imagen resultante en el proceso. El coeficiente central del Kernel,  $\omega(0, 0)$ , siempre debe alinearse con el píxel en la ubicación  $(x, y)$ . Para un Kernel de tamaño  $m \times n$ , se asume que  $m = 2a + 1$  y  $n = 2b + 1$ , donde  $a$  y  $b$  son enteros no negativos. Esta suposición permite únicamente Kernels de tamaño impar en ambas direcciones de coordenadas. Con Kernels impares, la matriz tendrá un centro único, facilitando la indexación en cada operación y siendo simétricamente espacial. La Ecuación (31) puede representarse de manera generalizada. En esta representación, se utiliza el símbolo  $\star$  para denotar dicha operación (Woods, 2018).

$$(\omega \star I)(x, y) = \sum_{s=-a}^a \sum_{t=-b}^b \omega(s, t) I(x + s, y + t) \quad (32)$$

De la ecuación (32) se tiene que  $x$  &  $y$  varían de manera que el centro del Kernel visita cada píxel en  $I$  una vez, para crear la nueva imagen. Es importante destacar que la operación  $(\omega \star I)$  comparte una similitud con la operación de convolución estándar aplicada a una imagen  $I$  utilizando un Kernel  $\omega$ , representada por  $(\omega \star I)$ . La distinción fundamental entre estas dos operaciones reside en el tratamiento de la matriz del Kernel  $\omega$ ; específicamente, en la operación  $(\omega \star I)$  se omite la rotación de 180 grados de la matriz del Kernel, es decir, se omite una combinación de operaciones de reflexión, tanto horizontal como vertical, de la matriz en cuestión. En el ámbito de las matemáticas puras, la operación de convolución se caracteriza por su propiedad de generar siempre una matriz cuyas dimensiones difieren de las matrices originales con las que se inicia el proceso, específicamente la imagen y el Kernel. La operación  $(\omega \star I)$  no es una excepción a esta propiedad.

En la operación de correlación espacial, es deseable que la imagen resultante, obtenida después de aplicar el filtro  $\omega$  a la imagen  $I$ , mantenga las mismas dimensiones que la imagen original  $I$ . Este problema surge porque, cuando el Kernel se coloca sobre los píxeles en los bordes de la imagen, no tiene suficientes píxeles vecinos para realizar la suma ponderada por completo. Esto se debe a que, en los bordes, parte del Kernel se extiende más allá de la imagen, dejando áreas sin píxeles correspondientes. Por lo tanto, el Kernel solo se mueve hasta donde puede cubrir completamente los píxeles vecinos.

Para resolver este problema, se añaden píxeles adicionales en los bordes. Para ello, la imagen se extiende reflejando sus bordes. Es decir, los píxeles en el borde de la imagen se duplican hacia afuera, como si se reflejaran en un espejo. Esta técnica se conoce como “relleno por espejo” (Woods, 2018).

Ahora surge la pregunta: ¿Cuántos píxeles adicionales son necesarios en los bordes? Cuando se trabaja con Kernels de dimensiones impares, la expansión se determina dividiendo la dimensión del Kernel por dos y tomando la parte entera de este resultado. Esta operación se aplica tanto en las dimensiones horizontal como vertical del Kernel. Por ejemplo, si la dimensión horizontal del Kernel es 5, se divide 5 por 2 para obtener 2.5; se toma la parte entera, que es 2. Esto significa que se añaden dos píxeles a cada lado de la imagen en la dirección horizontal, y lo mismo se aplica en la dirección vertical. De esta manera, se asegura que el tamaño de la imagen permanezca constante después de aplicar el filtro, asegurando una cobertura completa del Kernel en los bordes. Esto solo es posible para Kernels con un tamaño impar.

En el proyecto, para realizar operaciones de correlación espacial y generar una nueva imagen usando un filtro, se utilizó la biblioteca **torch.nn.functional**, específicamente la función **conv2d()**. Esta función se aplica considerando los siguientes parámetros (PyTorch, 2024):

- **Input (Entrada):** Este parámetro corresponde a la imagen de entrada  $I$  a la cual se aplicará el filtro. La imagen debe tener la forma  $(\text{minibatch}, \text{in\_channels}, iH, iW)$ . En este caso,  $\text{minibatch} = 1$ , ya que cada imagen se procesa individualmente, en lugar de en lotes. Para  $\text{in\_channels} = 1$ , porque se trabaja con imágenes en escala de grises, donde un solo canal representa la intensidad de la luz.  $iH, iW$  son las dimensiones de la imagen, que en el proyecto son siempre  $512 \times 512$  píxeles (PyTorch, 2024).
- **Weight (Peso):** Este parámetro representa el Kernel o filtro  $\omega$  que se aplica a la imagen  $I$  a través de la correlación espacial. Su dimensión es  $(\text{out\_channels}, \text{in\_channels}/\text{groups}, kH, kW)$ . Para  $\text{out\_channels} = 1$ , se aplica un solo filtro a la imagen,  $\text{in\_channels}/\text{groups} = 1$ , porque se maneja imágenes de un solo canal.  $kH, kW$  son las dimensiones del Kernel, que pueden ser definidas por el usuario (PyTorch, 2024).
- **Stride (Desplazamiento):** Este parámetro se establece en 1 para que el Kernel se mueva un píxel a la vez (PyTorch, 2024).

Usándose una imagen con dimensiones  $(1, 1, 512, 512)$  y un Kernel de dimensiones  $(1, 1, kH, kW)$ , se puede realizar la operación de correlación espacial.

A través de la operación de correlación espacial, es posible aumentar el desenfoque en la imagen DRR. Para esto, se requiere un Kernel Gaussiano. Partiendo de la ecuación general de Gauss en dos dimensiones:

$$G(x, y) = A \exp\left(-\left(\frac{(x - \mu_x)^2}{2\sigma_x^2} + \frac{(y - \mu_y)^2}{2\sigma_y^2}\right)\right) \quad (33)$$

donde  $A$  es una constante de normalización,  $\mu_x$  y  $\mu_y$  son los centros de la distribución Gaussiana en  $x$  &  $y$  (también conocidos como medias), y  $\sigma_x$  y  $\sigma_y$  son las desviaciones

estándar de las distribuciones Gaussianas en  $x$  &  $y$ . Para lograr desenfoque en una imagen, es importante que la reducción de nitidez y la atenuación de detalles, especialmente en regiones con cambios significativos en la intensidad o color de los píxeles, sean uniformes en todas las direcciones. Además, es necesario igualar los valores de desviación estándar en  $x$  &  $y$ ,  $\sigma_x = \sigma_y$ , para evitar un suavizado que favorezca una dirección específica sobre otra. Por lo tanto, la ecuación (33) se reduce a una Gaussiana isotrópica (MathWorks, s.f.).

$$G(s, t) = A \exp\left(-\left(\frac{s^2 + t^2}{2\sigma^2}\right)\right) \quad (34)$$

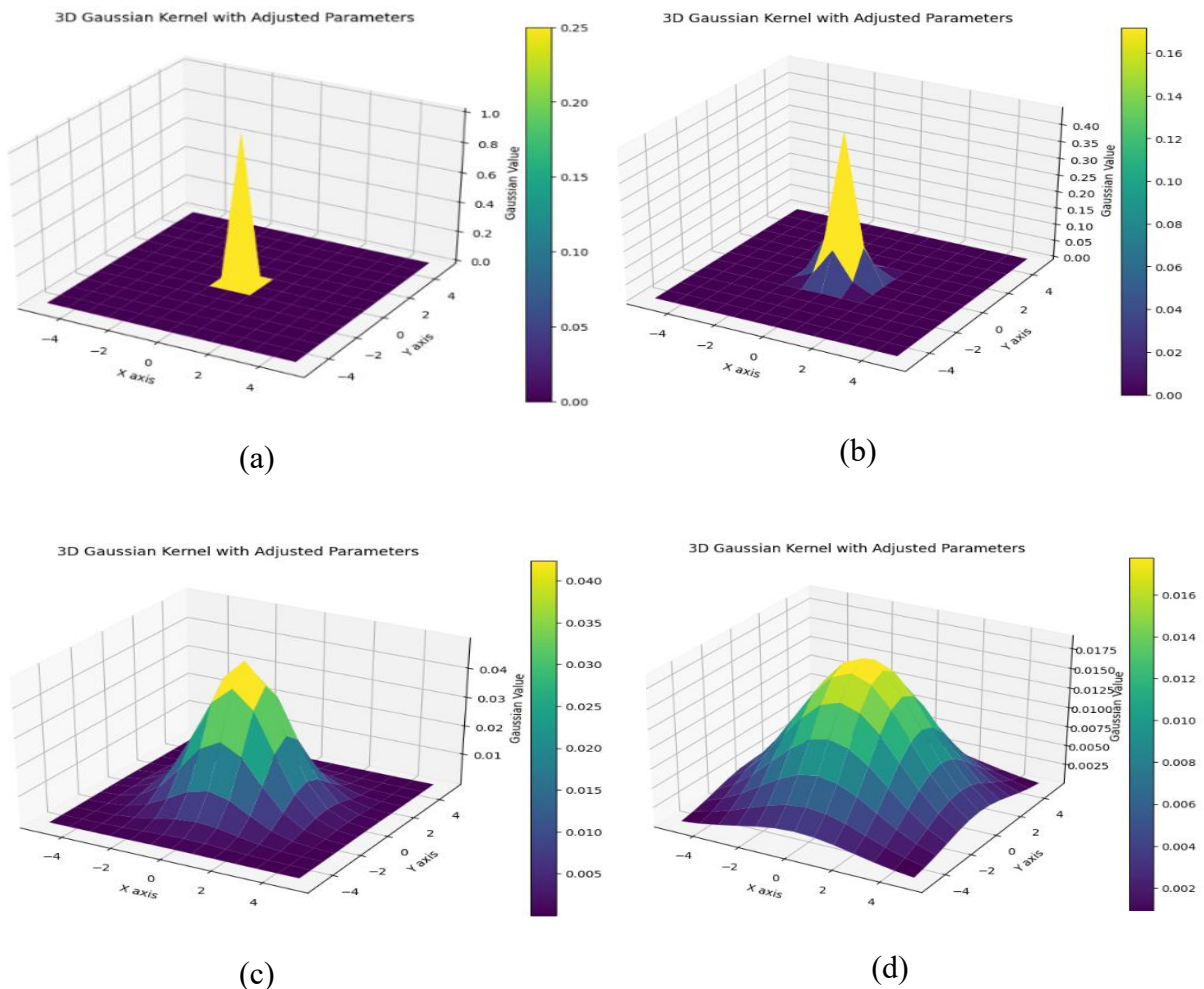


Figura 7. Ilustración del Kernel Gaussiano para diferentes valores de  $\sigma$ .

(a) para un valor de  $\sigma=0.012$ , (b) para un valor de  $\sigma=0.50$ ,  
(c) para un valor de  $\sigma=1.50$  y (d) para un valor de  $\sigma=2.50$

Denotando  $s$  y  $t$  como variables, se construye un Kernel Gaussiano con simetría circular. Esto puede observarse en la Figura 7 para un Kernel Gaussiano de dimensión  $9 \times 9$ . La forma de la curva de Gauss en estos Kernels varía según el valor de  $\sigma$ . Específicamente, a medida que  $\sigma$  aumenta, la curva de Gauss se ensancha, adoptando una forma más plana. Por el

contrario, a medida que  $\sigma$  disminuye, la curva se estrecha, resultando en una forma más puntiaguda. Para normalizar un Kernel Gaussiano, es crucial determinar la constante de normalización. Esta constante se calcula como la suma de todos los elementos en el Kernel Gaussiano. El propósito de este paso es asegurar que la suma total de los valores dentro del Kernel sea igual a uno.

Existen dos razones importantes por las cuales el Kernel Gaussiano debe ser normalizado:

1. Preservación de la **intensidad media**: Cuando un Kernel está normalizado, el valor promedio de un área de intensidad constante en la imagen original permanece igual en la imagen filtrada. Esto significa que, si una parte de la imagen tiene una intensidad uniforme, como un fondo plano de un color, esa uniformidad y nivel de intensidad no se alterarán después de aplicar el filtro (Woods, 2018).
2. Prevención de la **introducción de sesgo en la imagen**: Sin normalización, la convolución podría resultar en una imagen que sea generalmente más clara u oscura que la original. Esto es importante en el procesamiento de imágenes donde el objetivo es mantener la fidelidad de la imagen original en términos de sus niveles generales de iluminación y exposición (Woods, 2018).

## 6.6 DataLoader

El procesamiento de muestras de datos puede ser una tarea compleja en términos de mantenimiento. Para mejorar la legibilidad y la modularidad, es ideal que el código dedicado al manejo de conjuntos de datos esté separado del código utilizado para el entrenamiento de modelos. En este contexto, PyTorch ofrece dos herramientas fundamentales: `torch.utils.data.DataLoader` y `torch.utils.data.Dataset`. Estas herramientas permiten el uso de conjuntos de datos personalizados. El componente **Dataset** es responsable de almacenar las muestras y sus etiquetas correspondientes, mientras que **DataLoader** proporciona un iterable que facilita el acceso a estas muestras (PyTorch, 2024).

El **Dataset** se trata como un objeto, es decir, es una instancia de una clase que define sus propiedades y métodos. El método `__init__()` se utiliza para inicializar el **Dataset**, estableciendo los parámetros y la estructura de cómo se almacenarán los datos. El método `__len__()` devuelve el número de muestras dentro del **Dataset**, proporcionando un medio para determinar su tamaño total. Finalmente, el método `__getitem__()` se utiliza para acceder a una muestra específica y su etiqueta correspondiente, permitiendo la manipulación y el procesamiento de puntos de datos individuales (Stevens, 2020).

Por otro lado, los **DataLoaders** son iteradores que encapsulan **Datasets**. No solo proporcionan acceso a las muestras, sino que también permiten operaciones como agrupar

datos en lotes y barajarlos. La agrupación en lotes nos brinda la flexibilidad de seleccionar estrategias de muestreo personalizadas (Stevens, 2020).

En la implementación de la clase **Dataset**, se integra el generador de DRR. De este modo, al solicitar una muestra del **Dataset**, el algoritmo generador de DRR se activa automáticamente para producir el resultado deseado.

Para evaluar el rendimiento temporal del código desarrollado para la generación de imágenes en el DataLoader, se empleó la herramienta ‘%%timeit’ de IPython, ampliamente utilizada en Jupyter Notebooks. Esta herramienta permite la medición repetida y sistemática del tiempo de ejecución de bloques de código en Python, proporcionando así una estimación precisa del rendimiento.

Al usar ‘%%timeit’, el sistema ejecuta el bloque de código varias veces para minimizar las variaciones en el tiempo de ejecución que podrían ser inducidas por fluctuaciones en el sistema operativo o el entorno de ejecución. Por defecto, ‘%%timeit’ determina automáticamente el número óptimo de ejecuciones para lograr un equilibrio entre precisión y tiempo total de ejecución; en este caso, se seleccionan siete iteraciones por ciclo.

Los resultados proporcionados por ‘%%timeit’ incluyen el tiempo promedio por ciclo, que es el tiempo promedio que toma ejecutar el código, y la desviación estándar, que refleja la consistencia de estos tiempos de ejecución en varias corridas del código.

## 7. Resultados y discusión

El presente trabajo de tesis proporcionó los siguientes resultados, con sus respectivas discusiones.

Los efectos de la convolución de imágenes DRR con un Kernel Gaussiano se volvieron perceptibles al usar Kernels con dimensiones de 15x15 y mayores. Sin embargo, al aplicar Kernels de tamaño impar más pequeños que 15x15, no se percibió un efecto de desenfoque significativo, a pesar de ajustar el parámetro  $\sigma$  a 3.5. Estas observaciones sugieren que el tamaño del Kernel es crítico en la manifestación del desenfoque en imágenes procesadas con un filtro Gaussiano, siendo necesario que el tamaño del Kernel sea lo suficientemente grande y tenga una tasa de bits suficientemente alta para muestrear adecuadamente la distribución Gaussiana de la distribución del desenfoque, como se puede observar en la figura 8.

En cuanto al ruido generado por la distribución de Poisson, los resultados obtenidos se ilustran en la Figura 9. Se observa que para las imágenes DRR ajustadas a una escala de píxeles de 0 a 1, el nivel de ruido es significativamente mayor en comparación con las imágenes ajustadas a una escala de 0 a 250. Además, a medida que aumenta la escala de píxeles de la imagen, se nota una reducción progresiva del ruido de Poisson.

Como se ilustra en la Figura 5, a medida que aumenta el valor de  $\lambda$ , la distribución de Poisson se amplía, implicando una mayor variación. Sin embargo, como se observa en la Figura 9, a escalas de píxeles más bajas, se genera un ruido más perceptible en la imagen. Esta aparente contradicción se explica considerando que, aunque un valor de  $\lambda$  más alto lleva a más ruido en áreas de alta intensidad en una imagen individual (ya que el ruido de Poisson es proporcional a la intensidad del píxel), la percepción general del ruido también está influenciada por la escala de intensidad de la imagen. Esto se debe al redondeo de la intensidad de la imagen cuando se representa como un número entero. Para que el ruido de Poisson se muestree con precisión, debe haber suficiente profundidad de bits en la intensidad de la imagen, así como un modelo preciso del sistema de imágenes y la salida del tubo utilizada para la recolección de la imagen. Una profundidad de bits de 12 bits es típica para el procesamiento de imágenes, aunque puede ser computacionalmente inviable para algunas aplicaciones.

En relación con la medición del centro de masa, se compararon dos métodos: el utilizado en este proyecto y el método estadístico proporcionado por 3D Slicer. Para la medición en el eje X, se obtuvo un error absoluto medio (MAE) de 0.01270 mm; en el eje Y, el MAE fue de 0.01402 mm; y en el eje Z, fue de 0.01660 mm. El MAE promedio de estas tres

mediciones fue de 0.01445 mm. Estos resultados sugieren que el cálculo del centro de masa es tanto preciso como fiable.

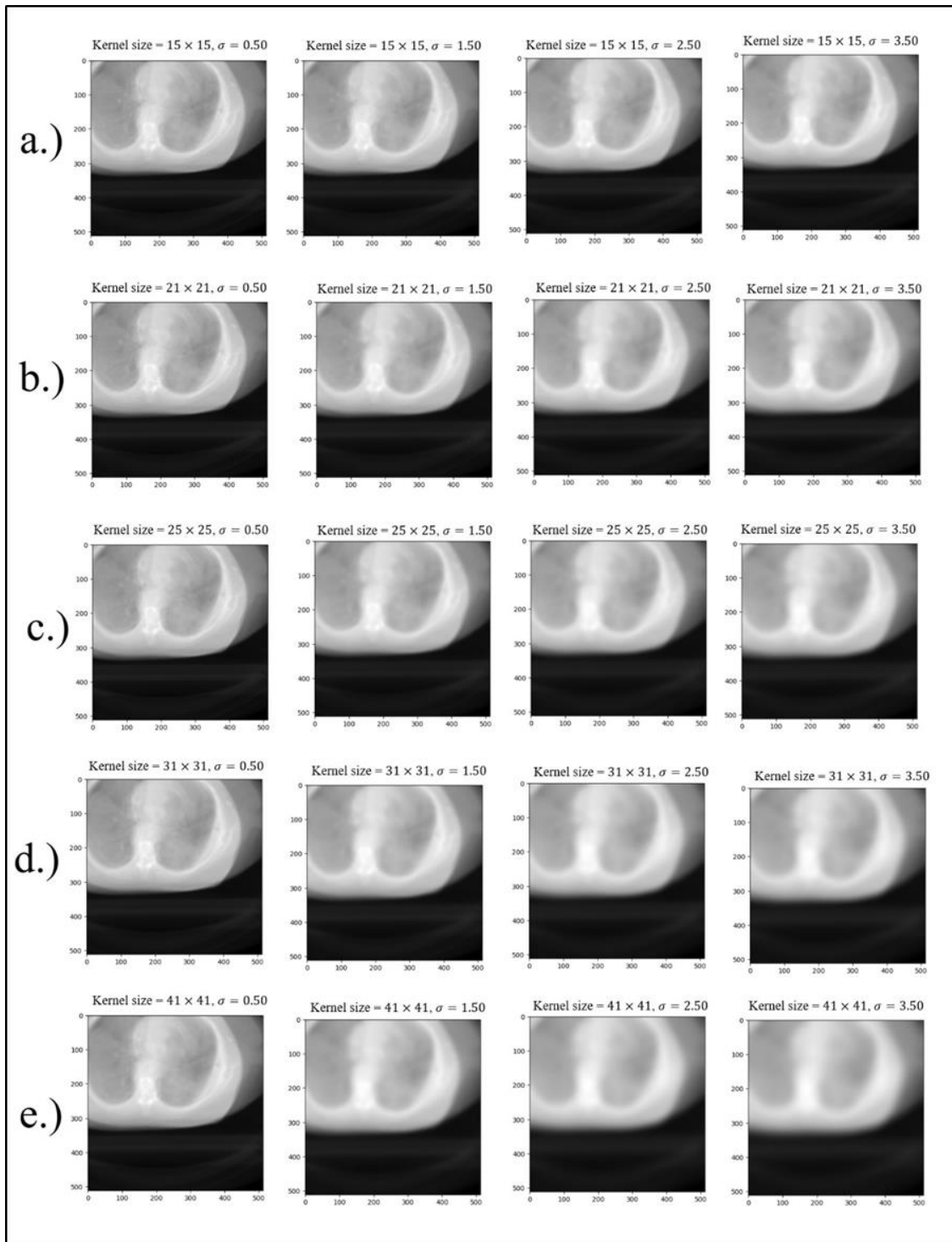


Figura 8. Efecto del Ruido Gaussiano en la imagen DRR para diferentes tamaños de Kernel y valores de  $\sigma$

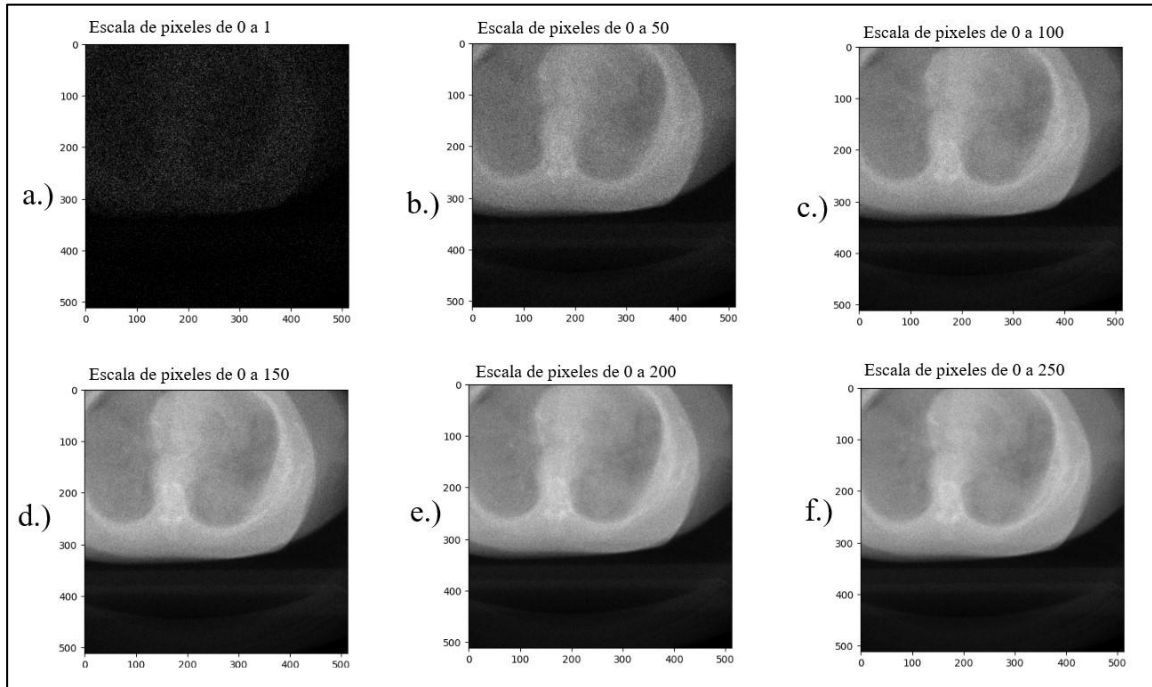


Figura 9. Ruido de Poisson en la imagen DRR, para diferentes escalas de pixeles

Se analizó la eficiencia temporal de un algoritmo diseñado para generar imágenes de entrenamiento basadas en datos DICOM. El procedimiento se dividió en dos etapas principales: la carga de datos DICOM y la generación de imágenes de radiografía digitalmente reconstruida (DRR) con inclusión de ruido.

Durante la primera etapa, referente a la carga de datos DICOM, los tiempos registrados se mantuvieron consistentemente cerca del promedio, independientemente del tipo de unidad de procesamiento gráfico (GPU) utilizada. Como se indica en la Tabla 1, el tiempo promedio de carga fue de 553 ms, con una desviación estándar de 16.6 ms.

Tabla 1. Tiempo de ejecución para la carga de datos DICOM para el generador de DRR, con diferentes GPUs				
Cronometraje: Proceso de carga de datos DICOM para la generación de imágenes				
Fase respiratoria	T4	V100	A100	LA
10%	561 ms ± 9.99 ms	576 ms ± 10.9 ms	561 ms ± 13.2 ms	534 ms ± 17.3 ms
20%	563 ms ± 11.2 ms	550 ms ± 8.43 ms	550 ms ± 11.4 ms	535 ms ± 12 ms
30%	575 ms ± 23.7 ms	569 ms ± 10.2 ms	564 ms ± 11.9 ms	528 ms ± 11.6 ms
40%	564 ms ± 8.79 ms	574 ms ± 10.9 ms	561 ms ± 22.6 ms	529 ms ± 6.84 ms
50%	551 ms ± 6.58 ms	572 ms ± 15.8 ms	554 ms ± 9.98 ms	526 ms ± 9.85 ms
60%	556 ms ± 9.39 ms	564 ms ± 9.5 ms	556 ms ± 9.74 ms	524 ms ± 6.55 ms
70%	569 ms ± 9.91 ms	576 ms ± 10.7 ms	554 ms ± 8.25 ms	529 ms ± 9.4 ms
80%	543 ms ± 9.86 ms	571 ms ± 15.9 ms	552 ms ± 18.4 ms	527 ms ± 5.3 ms
90%	571 ms ± 9.47 ms	559 ms ± 7.04 ms	549 ms ± 13.6 ms	540 ms ± 20.2 ms

La segunda etapa, relacionada con la generación de imágenes DRR, mostró una variabilidad significativa en el tiempo de procesamiento dependiendo de la GPU empleada. Según la Tabla 2, cambiar de GPU resultó en una reducción del tiempo promedio de 379 ms a 175 ms.

Tabla 2. Tiempo de ejecución del generador de DRR, incluyendo operaciones de ruido, con diferentes GPUs				
Cronometraje: Proceso de generación de imágenes				
Fase respiratoria	T4	V100	A100	LA
10%	385 ms ± 13.8 ms	256 ms ± 44.1 ms	178 ms ± 7.26 ms	295 ms ± 4.56 ms
20%	372 ms ± 5.23 ms	214 ms ± 8.71 ms	177 ms ± 7.13 ms	294 ms ± 5.46 ms
30%	375 ms ± 2.93 ms	216 ms ± 7.19 ms	175 ms ± 6.28 ms	291 ms ± 5.85 ms
40%	375 ms ± 2.42 ms	215 ms ± 3.95 ms	174 ms ± 8.94 ms	293 ms ± 9.33 ms
50%	380 ms ± 12.6 ms	211 ms ± 2.09 ms	177 ms ± 5.25 ms	291 ms ± 6.85 ms
60%	382 ms ± 14.2 ms	214 ms ± 6.12 ms	173 ms ± 8.59 ms	293 ms ± 8.26 ms
70%	385 ms ± 14.2 ms	218 ms ± 7.4 ms	177 ms ± 6.72 ms	296 ms ± 7.87 ms
80%	381 ms ± 6.98 ms	217 ms ± 6.46 ms	175 ms ± 6.16 ms	292 ms ± 8.65 ms
90%	377 ms ± 9.48 ms	220 ms ± 9.24 ms	176 ms ± 8.93 ms	295 ms ± 13.4 ms

Se sugiere que la carga de datos DICOM se realice antes del proceso de entrenamiento. De esta manera, durante el entrenamiento, el cargador de datos solo necesitaría ejecutar el proceso de generación de imágenes, optimizando el tiempo de carga en el algoritmo de entrenamiento. Según las estimaciones realizadas, cargar un total de 60,000 imágenes para entrenamiento se estima que tomaría aproximadamente 3 horas.

Es importante destacar que, para esta investigación, se utilizaron GPUs proporcionadas por servicios en la nube como Google Colab. Esta estrategia eliminó la necesidad de adquirir equipos especializados y costosos, como computadoras de alto rendimiento con tarjetas gráficas avanzadas.

Finalmente, como se puede ver en la Figura 10, las proyecciones DRR puede producir, en el movimiento respiratorio del paciente, la localización del tumor mostrando fidelidad a la geometría y continuidad de la reconstrucción.

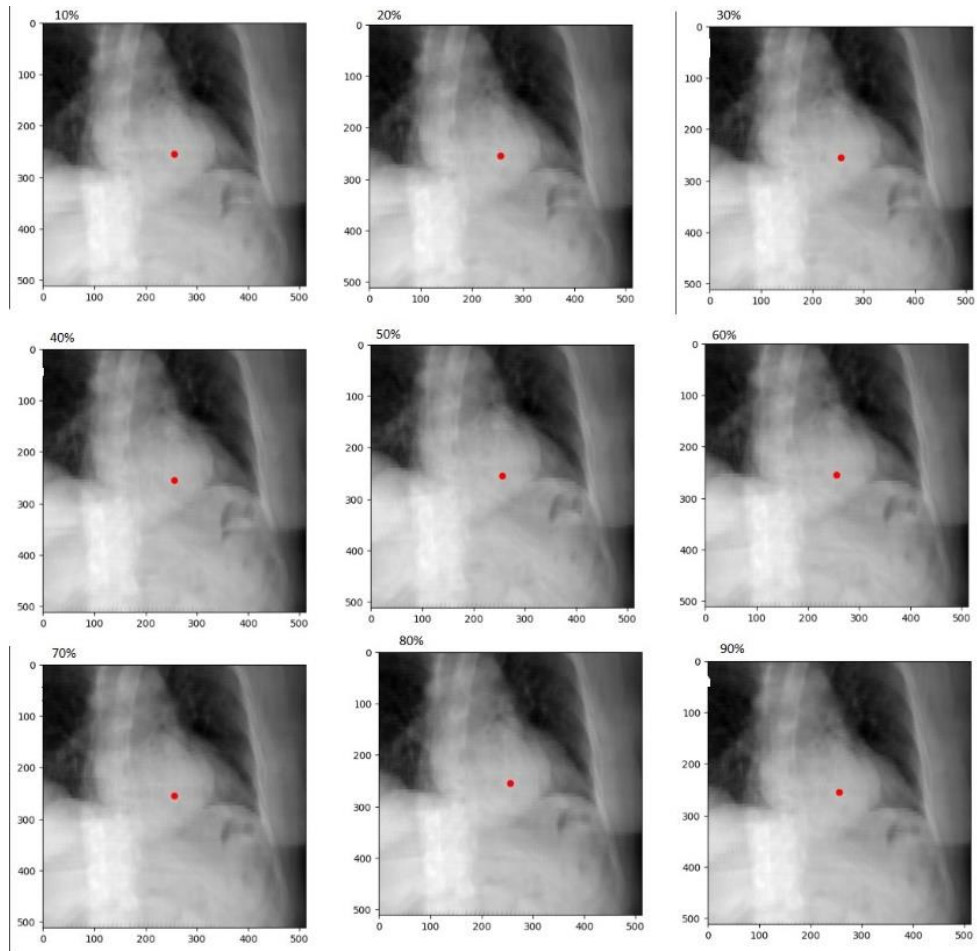


Figura 10. Imágenes DRR centradas en el centro de masa del tumor (marcado con un punto rojo) en diferentes fases del ciclo respiratorio, desde el 10% hasta el 90% de la amplitud de inspiración

## 8. Conclusiones

Los resultados de este estudio demuestran que el DataLoader desarrollado es capaz de generar Radiografías Digitalmente Reconstruidas (DRRs) que replican la posición del tumor en el movimiento respiratorio del paciente. Además, se ha probado la viabilidad de generar tipos de ruido Gaussiano y de Poisson, lo que permite ajustes personalizados según las necesidades específicas del estudio. La implementación de GPUs ha reducido significativamente los tiempos de ejecución del programa, ofreciendo así una ventaja considerable en la optimización de procesos.

Este DataLoader será una herramienta esencial para su uso en redes neuronales dirigidas al entrenamiento para la detección de tumores en 4DCT, específicamente en estudios de 4DCT.

A lo largo de este proyecto, se adquirió conocimiento fundamental sobre la manipulación y comprensión de la estructura de datos DICOM. Esto incluyó técnicas para calcular el centro de masa del tumor, determinar el volumen de CT necesario para la generación de DRR y ajustar las posiciones físicas del volumen de CT para centrar las DRRs en el tumor. Estas habilidades se desarrollaron a través de la programación en Python, destacando la importancia de la competencia técnica en el avance de la física médica.

## 9. Recomendaciones

Es importante destacar que, como en la Figura 10, no se pudo observar de manera notable el movimiento respiratorio del tumor en las diferentes fases de la respiración. Al consultar el documento del 4DCT, se llega a la misma conclusión respecto al movimiento respiratorio. Se recomienda abordar este aspecto en la segunda fase con diferentes archivos 4DCT, para verificar si es posible observar el movimiento, y si el generador DRR puede mostrarlo. En el caso de que el movimiento respiratorio sea evidente en el 4DCT, pero no en el DRR, esto es un aspecto que se podría corregir.

Una vez que el movimiento respiratorio sea visible en la imagen DRR, se podrán iterar los datos necesarios mediante el DataLoader para entrenar una red neuronal profunda (DNN) que detecte la ubicación del tumor en imágenes fluoroscópicas. Se recomienda utilizar una DNN capaz de generar un Mapa de Probabilidad del Objetivo (TPM), el cual muestra las posibles ubicaciones del tumor en la imagen. Esto proporciona una representación probabilística de la posición del tumor, permitiendo un seguimiento preciso en tiempo real durante el tratamiento (Hirai, 2019).

## 10. Bibliografía

### Referencias de internet

3D Slicer. (s. f.). *Segment Statistics Module. Slicer Documentation*. Recuperado de [https://slicer.readthedocs.io/en/v4.11/user\\_guide/modules/segmentstatistics.html](https://slicer.readthedocs.io/en/v4.11/user_guide/modules/segmentstatistics.html)

Google. (s.f.). *Google Colab*. Recuperado de <https://colab.research.google.com>

Intel. (s.f.). *Diferencias entre CPU y GPU*. Intel. Recuperado de <https://www.intel.la/content/www/xl/es/products/docs/processors/cpu-vs-gpu.html>

MathWorks. (s.f.). *Apply Gaussian Smoothing Filters to Images*. Recuperado de <https://www.mathworks.com/help/images/apply-gaussian-smoothing-filters-to-images.html>

Medical Imaging & Technology Alliance. (s.f.). *About DICOM. DICOM Standard*. Recuperado de <https://www.dicomstandard.org/about-home>

Medical Imaging & Technology Alliance. (2023). *DICOM Standard. National Electrical Manufacturers Association*. Recuperado de <https://www.dicomstandard.org/current/>

NVIDIA. (s.f.). *Documentación del Toolkit CUDA. NVIDIA Developer*. Recuperado de <https://docs.nvidia.com/cuda/>

PyTorch. (s.f.). *Loading data in PyTorch*. Recuperado de [https://pytorch.org/tutorials/beginner/basics/data\\_tutorial.html](https://pytorch.org/tutorials/beginner/basics/data_tutorial.html)

PyTorch. (s.f.). *torch.nn.Conv2d*. Recuperado de <https://pytorch.org/docs/stable/generated/torch.nn.Conv2d.html#torch.nn.Conv2d>

PyTorch. (s.f.). *torch.poisson*. Recuperado de <https://pytorch.org/docs/stable/generated/torch.poisson.html>

## Otras referencias

- Caillet, V., Booth, J. T., & Keall, P. 2017. «IGRT and Motion Management During Lung SBRT Delivery». *Physica Medica: PM: An International Journal Devoted to the Applications of Physics to Medicine and Biology: Official Journal of the Italian Association of Biomedical Physics (AIFB)*, 44: 113–122. <https://doi.org/10.1016/j.ejmp.2017.06.006>
- Chan, C. L., Sullivan, B. J., Sahakian, A. V., Katsaggelos, A. K., Swiryn, S., Hueter, D. C., & Frohlich, T. 1990. «Simulation of quantum mottle in digital angiographic images. Proceedings of SPIE». *The International Society for Optical Engineering*, 1245: 104-109.
- Dieterich, S., Ford, E., Pavord, D., & Zeng, J. 2016. *Practical Radiation Oncology Physics: A Companion to Gunderson & Tepper's Clinical Radiation Oncology*. Elsevier.
- Ferris, W. S., Kissick, M. W., Bayouth, J. E., Culberson, W. S., & Smilowitz, J. B. 2020. «Evaluation of Radixact Motion Synchrony for 3D Respiratory Motion: Modeling Accuracy and Dosimetric Fidelity». *Journal of Applied Clinical Medical Physics*. 21 (9): 96–106. <https://doi.org/10.1002/acm2.12978>.
- Gonzalez, R. C., & Woods, R. E. 2018. *Digital Image Processing*. 4th ed., Global. Pearson Education Limited.
- Gopalakrishnan, V., & Golland, P. 2022. «Fast Auto-differentiable Digitally Reconstructed Radiographs for Solving Inverse Problems in Intraoperative Imaging». *Workshop on Clinical Image-Based Procedures*: 1-11. <https://doi.org/10.48550/arXiv.2208.12737>
- Hirai, R., Sakata, Y., Tanizawa, A., & Mori, S. 2019. «Real-time Tumor Tracking Using Fluoroscopic Imaging with Deep Neural Network Analysis». *Physica Medica: PM: An International Journal Devoted to the Applications of Physics to Medicine and Biology: Official Journal of the Italian Association of Biomedical Physics (AIFB)*, 59: 22–29. <https://doi.org/10.1016/j.ejmp.2019.02.006>
- Hogg, R. V., Tanis, E. A., & Zimmerman, D. L. 2015. *Probability and Statistical Inference*. 9th ed. Pearson Education.
- Khor, H. L., Liew, S. C., & Zain, J. M. 2015. «A Review on Parallel Medical Image Processing on GPU». *2015 4th International Conference on Software Engineering and Computer Systems (ICSECS)*, Kuantan, Malaysia: 45-48. <https://doi.org/10.1109/ICSECS.2015.7333121>
- Mayles, W. P. M., Nahum, A. E., & Rosenwald, J. C. (Eds.). 2022. *Handbook of Radiotherapy Physics: Theory & Practice*. 2nd ed. Vol. 1. CRC Press.

- \_\_\_\_\_; 2022a. *Handbook of Radiotherapy Physics: Theory & Practice*. 2nd ed. Vol. 2. CRC Press.
- Menten, M. J., Wetscherek, A., & Fast, M. F. (2017). «MRI-guided Lung SBRT: Present and Future Developments». *Physica Medica: PM: An International Journal Devoted to the Applications of Physics to Medicine and Biology: Official Journal of the Italian Association of Biomedical Physics (AIFB)*, 44: 139–149. <https://doi.org/10.1016/j.ejmp.2017.02.003>
- Pratx, G., & Xing, L. 2011. «GPU Computing in Medical Physics: A Review». *Medical physics*, 38 (5): 2685–2697. <https://doi.org/10.1118/1.3578605>
- Seeram, E. 2016. *Computed Tomography: Physical Principles, Clinical Applications, and Quality Control*. 4th ed. Elsevier.
- Siddon R. L. 1985. «Fast Calculation of the Exact Radiological Path for a Three-dimensional CT Array». *Medical physics*, 12 (2): 252–255. <https://doi.org/10.1118/1.595715>
- Stevens, E., Antiga, L., & Viehmann, T. 2020. *Deep Learning with PyTorch*. Manning Publications.
- Whybra, P., & Spezi, E. 2023. «Sensitivity of Standardised Radiomics Algorithms to Mask Generation Across Different Software Platforms». *Scientific reports*, 13 (1): 1-13. <https://doi.org/10.1038/s41598-023-41475-w>

# 11. Apéndices

## Apéndice A

Repositorio GitHub:

El código utilizado en esta tesis está disponible en el siguiente repositorio de GitHub:

Repositorio: DataSet\_DataLoader\_4DCT

Este repositorio contiene dos scripts fundamentales:

1. DataSet\_DataLoader.py:

- Descripción: Este script contiene las clases y funciones necesarias para cargar y preprocesar los datos de 4DCT, para preparar los datos con el fin de entrenar una red neuronal profunda capaz de rastrear tumores en una 4DCT.

2. ct4d\_image\_generator.py:

- Descripción: Este script genera imágenes de DRR que son utilizadas en el script de DataLoader y DataSet.

Para acceder al repositorio y descargar los scripts, puede utilizar el siguiente enlace:  
[https://github.com/HectorPivaral/DataSet\\_DataLoader\\_4DCT.git](https://github.com/HectorPivaral/DataSet_DataLoader_4DCT.git)

## Apéndices B

### Iteración del Dataset para el DataLoader

A continuación, se describe el proceso de iteración sobre un Dataset de imágenes 4DCT de pacientes, especificando cómo se manejan los índices para acceder a las imágenes en diferentes fases de un ciclo de respiración.

**Inicialización del Dataset:** El Dataset está compuesto por datos de múltiples pacientes, cada uno con imágenes correspondientes a diferentes porcentajes de un ciclo de respiración (10%, 20%, 30%, 40%, 50%, 60%, 70%, 80%, 90%). El Dataset se inicializa con el directorio base que contiene los datos de los pacientes. Se crea una lista de pacientes y una lista de porcentajes de respiración.

**Método ‘\_len\_’:** Para el cálculo del total de las imágenes en el Dataset, se procede a realizar el producto del número de pacientes y el número de porcentajes de respiración, multiplicado por 3 para incluir las imágenes en los planos axial, coronal y sagital.

**Método ‘determinar\_indices’:** Toma un índice y devuelve el nombre del paciente y el porcentaje de respiración correspondiente. Para calcular estos valores, se utiliza el operador módulo (%) para determinar el índice del porcentaje de respiración y la división entera (//) para determinar el índice del paciente. Esto asegura que cada índice se mapea correctamente a un paciente y un porcentaje de respiración específicos.

**Método ‘\_getitem\_’:** Toma un índice y devuelve la imagen y su etiqueta correspondiente. Primero, ajusta el índice para manejar las tres imágenes (axial, coronal, sagital) por cada conjunto de datos. Esto se hace calculando el índice del marco usando el operador módulo y ajustando el índice base dividiendo por 3. Luego, se llama al método determinar\_indices para obtener el paciente y el porcentaje de respiración. Con esta información, se construye la ruta del archivo y se utiliza un generador de datos 4DCT para crear las imágenes en los tres planos. Finalmente, se selecciona la imagen correspondiente al marco actual y se crea una etiqueta que incluye la posición del centro de masa, el porcentaje de respiración y el marco.

### Detalle de procedimiento

**Cálculo del Índice del Marco:** Cada índice en el Dataset se divide en tres imágenes correspondientes a los planos axial, coronal y sagital. Para determinar a cuál de estos planos corresponde un índice específico, se utiliza el operador módulo (%):

El índice  $\text{idx mod } 3$  ( $\text{idx} \% 3$ ) se utiliza para determinar el plano:

- Si el resultado es 0, el plano es axial.
- Si el resultado es 1, el plano es coronal.
- Si el resultado es 2, el plano es sagital.

Por ejemplo:

- Para  $\text{idx} = 0$ ,  $0 \% 3 = 0$ , el plano es axial.
- Para  $\text{idx} = 1$ ,  $1 \% 3 = 1$ , el plano es coronal.
- Para  $\text{idx} = 2$ ,  $2 \% 3 = 2$ , el plano es sagital.

**Cálculo del Índice Base:** Después de determinar el plano, se ajusta el índice base haciendo una división entera entre el índice original  $\text{idx}$  por 3. Entonces, el índice base representa cuántos conjuntos completos de tres imágenes (axial, coronal, sagital) se han contado hasta el índice actual. Esto es importante para mapear el índice actual a su correspondiente paciente y porcentaje de respiración.

**Determinación del Porcentaje de Respiración:** Se utiliza un operador módulo entre el índice base y el número total de porcentajes, para asegurarse de que el índice se reinicie después de alcanzar el número total de porcentajes.

**Determinación del Paciente:** Se utiliza la división entera entre el índice base y el número total de porcentajes. Por ejemplo, si tenemos 9 fase de respiración, cuando el índice sea 10, obtendremos el paciente No.2.

A continuación, se presentan algunos ejemplos prácticos

**Ejemplo práctico:** Supóngase que se tiene 2 pacientes (paciente 00, paciente 01) y 9 porcentajes de respiración (10%, 20%, 30%, 40%, 50%, 60%, 70%, 80%, 90%). Cada conjunto de datos genera 3 imágenes (axial, coronal, sagital). El Dataset total tendrá 18 conjuntos de datos (2 pacientes \* 9 porcentajes), resultando en un total de 54 imágenes.

- Índice 0:
  - Índice del marco:  $0 \% 3 = 0$  (Axial)
  - Índice base:  $0 // 3 = 0$
  - Porcentaje:  $0 \% 9 = 0$  (10%)
  - Paciente:  $0 // 9 = 0$  (Paciente 00)

Resultado: Imagen Axial de Paciente 00 con 10%

- Índice 1:
  - Índice del marco:  $1 \% 3 = 1$  (Coronal)
  - Índice base:  $1 // 3 = 0$
  - Porcentaje:  $0 \% 9 = 0$  (10%)
  - Paciente:  $0 // 9 = 0$  (Paciente 00)

Resultado: Imagen Coronal de Paciente 00 con 10%

- Índice 2:
  - Índice del marco:  $2 \% 3 = 2$  (Sagital)
  - Índice base:  $2 // 3 = 0$
  - Porcentaje:  $0 \% 9 = 0$  (10%)
  - Paciente:  $0 // 9 = 0$  (Paciente 00)

Resultado: Imagen Sagital de Paciente 00 con 10%

- Índice 3:
  - Índice del marco:  $3 \% 3 = 0$  (Axial)
  - Índice base:  $3 // 3 = 1$
  - Porcentaje:  $1 \% 9 = 1$  (20%)
  - Paciente:  $1 // 9 = 0$  (Paciente 00)

Resultado: Imagen Axial de Paciente 00 con 20%

- Índice 26:
  - Índice del marco:  $26 \% 3 = 2$  (Sagital)
  - Índice base:  $26 // 3 = 8$
  - Porcentaje:  $8 \% 9 = 8$  (90%)
  - Paciente:  $8 // 9 = 0$  (Paciente 00)

Resultado: Imagen Sagital de Paciente 00 con 90%

- Índice 27:
  - Índice del marco:  $27 \% 3 = 0$  (Axial)
  - Índice base:  $27 // 3 = 9$
  - Porcentaje:  $9 \% 9 = 0$  (10%)
  - Paciente:  $9 // 9 = 1$  (Paciente 01)

Resultado: Imagen Axial de Paciente 01 con 10%