



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Escola Tècnica
Superior d'Enginyeria
Informàtica

Escola Tècnica Superior d'Enginyeria Informàtica
Universitat Politècnica de València

Desarrollo de una librería software para el
análisis y validación de imágenes 2D/3D
PET basado en el protocolo NEMA

Trabajo Fin de Grado

Grado en Ingeniería Informática

Autor: Poves Carrasco, Pablo

Tutor: Rodríguez Álvarez, María José

Tutor Externo: Pastor Esteve, Gustavo

2021 - 2022

Resumen

En este trabajo de fin de grado se detalla el diseño e implementación de una **librería de análisis de imagen PET** (Imagen por emisión de positrones) basada en el protocolo **NEMA NU-4 2008**, que permite evaluar la calidad de las imágenes en base a unas figuras de mérito, utilizando el lenguaje interpretado **Python**.

Esta librería ofrece una serie de herramientas capaces de **automatizar** total o parcialmente el proceso de análisis de **imagen PET** siguiendo dicho protocolo, además de proveer de herramientas de soporte para los resultados.

El proyecto ha sido desarrollado durante las prácticas laborales en la empresa **Bruker** y en él se incluye un apartado que detalla el impacto que ha tenido su aplicación.

A lo largo de la memoria se describen las herramientas (**Jira**, **Confluence**, **Doxygen...**) y la metodología **agile Scrum** empleadas durante el desarrollo.

Al final de este documento, encontraremos un enlace a la documentación de la librería.

Palabras clave: Python, librería, imagen PET, NEMA UN-4 2008, automatizar, análisis de imagen PET, Bruker, Jira, Confluence, Doxygen, agile, Scrum.

Abstract

This final degree project details the design and implementation of a **PET** (Positron Emission Imaging) **image analysis library** based on the **NEMA NU-4 2008** protocol, which allows the quality of the images to be assessed on the basis of figures of merit, using the **Python** interpreted language.

This library offers a series of tools capable of fully or partially **automating** the **PET image analysis** process following this protocol, as well as providing support tools for the results.

The project has been developed during the internship at **Bruker** and includes a section detailing the impact of its application.

Throughout the report, the tools (**Jira**, **Confluence**, **Doxygen**...) and the agile Scrum methodology used during development are described.

At the end of this document, you will find a link to the library documentation.

Keywords: Python, library, PET image, NEMA UN-4 2008, automating, PET image analysis, Bruker, Jira, Confluence, Doxygen, agile, Scrum.

Índice

1. Introducción	7
1.1. Objetivos	8
1.2. Estructura de la memoria	9
2. Estado del arte	11
2.1. Contexto tecnológico.....	11
2.2. Soluciones existentes.....	24
3. Análisis del problema.....	27
3.1. Solución propuesta.....	27
4. Metodología	29
5. Diseño de la solución	31
5.1. Tecnología utilizada	31
5.2. Arquitectura básica.....	35
5.3. Diseño detallado	41
5.4. Limitaciones de la librería.....	53
6. Validación.....	55
7. Impacto logrado.....	59
8. Conclusiones	61
9. Futuro	63
10. Agradecimientos	65
11. Bibliografía	67
12. Documentación de la librería.....	69



1. Introducción

La imagen, en medicina, es una de las principales y más antiguas herramientas de diagnóstico, análisis e investigación de pacientes. Posee un papel vital en procedimientos como la identificación y localización de tumores.

En las últimas décadas, los avances en ciencia y tecnología nos han permitido desarrollar técnicas para la obtención de imágenes del interior de un organismo mediante el uso de radiación o ultrasonidos, evitando así tener que hacer intervenciones quirúrgicas.

La tomografía por emisión de positrones, referida en adelante como PET (de las siglas en inglés Positron Emission Tomography), es una tecnología de imagen funcional médica, ampliamente aplicada en oncología, neurología y cardiología. Permite inferir información de un cuerpo mediante la detección de la radiación emitida por un trazador introducido en este. La principal característica de PET es que consigue obtener información y determinar la localización de determinados procesos metabólicos, como por ejemplo la consumición de glucosas de un tumor.

Los escáneres PET, son máquinas muy complejas y precisas que requieren de varios procesos de calibración y validación exhaustivos, antes de ser enviados a los usuarios finales. Estos procesos pueden realizarse de distintas formas, pero uno de los estándares más extendidos para escáneres PET de investigación con animales es el protocolo de NEMA NU-4 2008. En él se detallan una serie de procedimientos para cuantificar distintas características de las imágenes PET, que después son utilizadas para determinar la calidad de imagen de un escáner. Junto con estas características, se incluye un proceso para la obtención de dichas cuantificaciones, procesos que tradicionalmente se han hecho a mano o con ayuda de algún software de análisis de imagen como AMIDE, pero siempre con la intervención humana.

Lo que se propone en este trabajo es la creación de una suite de herramientas (una librería) desarrollada en Python, para facilitar la creación de scripts que automatice total o parcialmente el proceso de la obtención de dichos parámetros de las imágenes PET, así como habilitar otros tipos de análisis y extracción de información de carácter más general.

1.1. Objetivos

El objetivo principal de este proyecto es crear una serie de herramientas software en forma de librerías de Python, que permitan la automatización total o parcial del apartado de análisis de calidad de imágenes, durante el proceso de validación de escáneres PET 3D preclínicos dedicados a la investigación en pequeños animales.

En el caso de este proyecto, y como resultado de ser una colaboración de prácticas laborales con la empresa Bruker (Empresa biotecnológica cuyas actividades, entre otras, incluyen la calibración y validación de escáneres PET 3D), las librerías se centrarán alrededor de los procedimientos de validación que ellos emplean en sus instalaciones, entre ellos el protocolo NEMA.

Como segundo objetivo, la librería proveerá de herramientas básicas y flexibles capaces de adaptarse a muchos otros procedimientos y estándares de validación, permitiendo extender o construir nuevas librerías de análisis de imagen a partir de esta.

Parte del reto al construir estas herramientas, es tener en cuenta el alto coste computacional y de memoria principal que puede llegar a tener su ejecución (las imágenes 3D pueden llegar a pesar más de 1GB). Por tanto, otro de los objetivos de este proyecto es intentar obtener la mayor eficiencia que Python pueda proveer.

1.2. Estructura de la memoria

Esta memoria se divide en 12 apartados.

En el que nos encontramos ahora, el primero, se encuentran la introducción y los objetivos del proyecto.

En el segundo, se realiza un estudio del panorama actual de las herramientas y los procesos de análisis de calidad de imagen PET que se emplean. Este apartado comienza con una introducción a conceptos necesarios para comprender el contexto en que se desarrolla el trabajo, y finaliza con una crítica a las soluciones existentes.

En el tercer apartado se exponen con más detalle los inconvenientes de las soluciones actuales y se introduce la solución propuesta.

En el apartado cuarto se describe la metodología y herramientas empleadas para la organización de las tareas del proyecto.

El quinto es el apartado más largo de la memoria. En él se detallan, tanto las tecnologías empleadas para el desarrollo de la librería, como la estructura y diseño de la librería en sí.

En el sexto apartado se documentan las pruebas de validación que la librería ha tenido que superar para considerarse apta para su uso.

En el apartado 7 se comenta cual ha sido el impacto de este proyecto en la empresa en que se ha desarrollado.

En el octavo, se encuentran las conclusiones, se comenta si se han alcanzado los objetivos y cuáles han sido las habilidades adquiridas durante la realización del trabajo.

En el noveno se comentan los posibles desarrollos y ampliaciones futuras que puede tener la librería.

El décimo apartado se compone de un comentario de agradecimiento a las partes implicadas en este proyecto.

En el undécimo apartado, se documentan las fuentes que se han empleado en el trabajo.

Por último, en el duodécimo apartado, se encuentra un enlace a la documentación del código de la librería.

2. Estado del arte

Para poder comprender el contexto en el que se encuentran, tanto el estado del arte como el resto del proyecto, es necesario introducir al lector en una serie de conceptos y términos propios de la materia que se está tratando.

2.1. Contexto tecnológico

Vóxel

Una imagen 3D está formada por vóxeles (del inglés *volumetric pixel*). Los vóxeles son el equivalente a los píxeles de las imágenes 2D, salvo por la diferencia de que estos tienen un volumen.

Los vóxeles pueden tener distintas dimensiones, pero para este trabajo consideraremos los vóxeles como polígonos cúbicos con un valor asociado que representa la intensidad de este. Los vóxeles en las imágenes PET 3D preclínicas rondan un tamaño de arista de entre 1 y 0,25 milímetros de longitud normalmente.

Una imagen PET 3D consiste, en términos de informática, en una larga serie de números en coma flotante cuyos valores representan la intensidad de los vóxeles de la imagen.

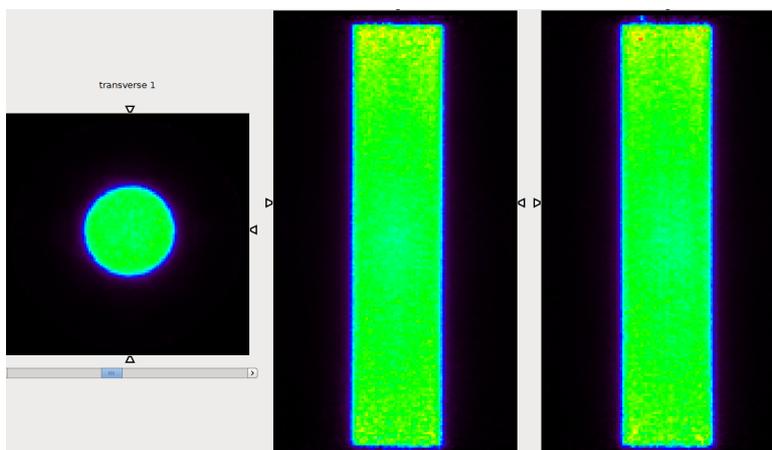


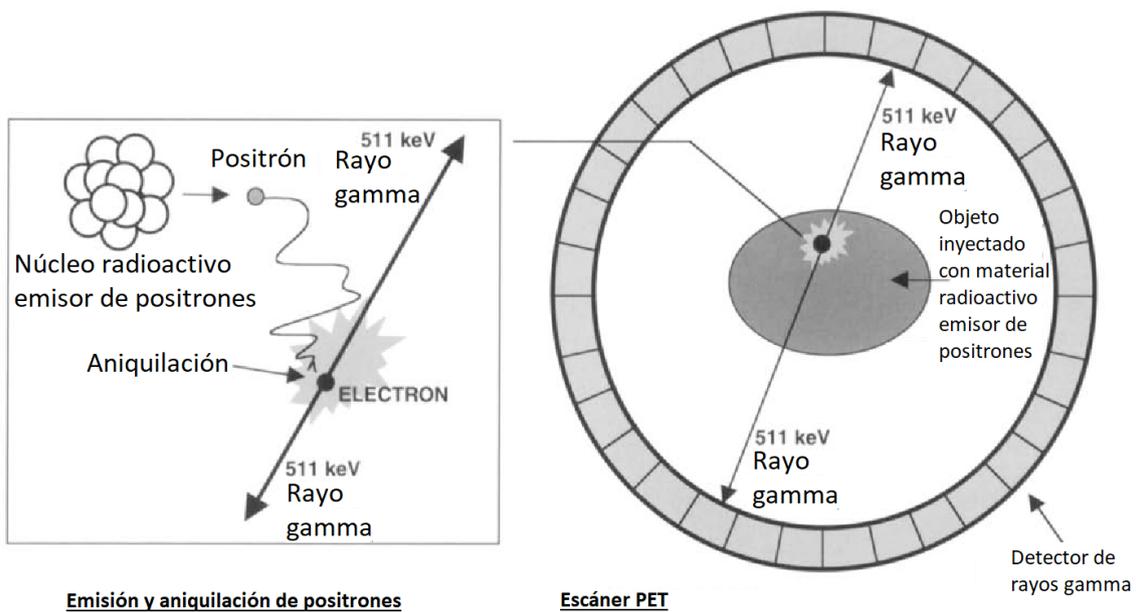
Figura 1 – Imagen PET de ejemplo mostrando las perspectivas transversal, coronal y sagital respectivamente.

Imágenes PET

Cuando hablamos de intensidad de un vóxel en imágenes PET, realmente, nos referimos al número de desintegraciones por unidad de tiempo que han tenido lugar dentro del volumen definido por el vóxel.

Los escáneres PET utilizan anillos de detectores que permiten captar pares de fotones de radiación gamma, fruto de la desintegración de los positrones (antipartícula del electrón) cuando colisionan con los electrones del ambiente. Los positrones son emitidos por trazadores radioactivos introducidos en el organismo estudiado.

Figura 3 – Diagrama mostrando la emisión y la desintegración de un positrón dentro de un anillo de



detectores de un escáner PET.

Utilizando algoritmos de reconstrucción, a cada vóxel del espacio del campo de visión de un escáner se le asigna un número que representa la cantidad de eventos de desintegración que se estima han sido emitidos en ese vóxel. Una región con un alto número de desintegraciones indica una alta acumulación del trazador radioactivo.

En el caso de algunos trazadores (como el Flúor-18) permiten conocer las áreas del organismo con mayor consumo o acumulación de glucosa como pueden ser el cerebro o los tumores.

Volumen de interés (VOI)

Un VOI (de sus siglas en inglés Volume Of Interest) hace referencia a una región en el espacio de una imagen 3D sobre la que deseamos conocer información de los vóxeles que la conforman. Usualmente los VOI se suelen definir en una imagen con formas cilíndricas o rectangulares y permiten conocer datos como la intensidad media de la región o intensidad máxima y mínima.

Fantoma (o maniquí)

La palabra “fantoma”, proveniente de la palabra en inglés *phantom*, consiste en un objeto con cavidades en las que se introduce una disolución de agua e isótopos radioactivos, con unas medidas determinadas y tolerancias muy bajas.

El propósito de estos objetos es el de proveer tanto un marco de referencia al mundo real, como la posibilidad de realizar distintas pruebas a la hora de capturar imágenes en los escáneres.

En este proyecto se habla en concreto de 3 tipos de fantomas.

- **El fantoma de uniformidad:** consiste en un cilindro hueco de radio y longitud conocidos.



Figura 4 – Imagen del fantoma de uniformidad.

- **El fantoma de fuente puntual:** es una pequeña semilla de Na22 radioactivo. (normalmente del tamaño de 1 milímetro cúbico).



Figura 5 – Imagen del fantoma de fuente puntual.

- **El fantoma de calidad de imagen de NEMA:** (en adelante fantoma NEMA IQ, de *Image Quality*) consiste en un objeto compuesto por una serie de cámaras huecas interconectadas.

Es un fantoma propio del estándar NEMA NU-4 2008 para cuantificar la calidad de una imagen PET y será detallado más adelante.



Figura 6 – Imagen del fantoma de calidad de imagen del protocolo NEMA.

Cuantificación de la calidad de una imagen PET 3D

Hay diferentes parámetros que determinan la calidad de una imagen y diferentes formas de cuantificar y calcular cada uno de ellos.

Con el propósito de estandarizar la cuantificación de la calidad de una imagen PET 3D en escáneres para animales, nace el protocolo NEMA NU-4 2008 (nos referiremos a él como “protocolo NEMA” en adelante en el proyecto).

El protocolo NEMA ha sido desarrollado en una colaboración entre los principales fabricantes de escáneres PET (Philips y General Electrics), científicos de diversas universidades (Oxford, Cambridge, Washington University...) y la asociación nacional de fabricantes eléctricos en EE. UU. (o NEMA de sus siglas en inglés *National Electrical Manufacturers Associaton*).

El protocolo NEMA propone cuatro figuras de mérito (mediciones) para cuantificar la calidad de una imagen PET 3D:

- La resolución espacial
- La uniformidad
- Los coeficientes de recuperación
- El ratio de desbordamiento

El documento expone detalladamente que fantoma ha de usarse para obtener los datos de cada figura de mérito y el proceso y cálculos asociados para lograrlo.

En este proyecto se pretende desarrollar una serie de herramientas que sean capaces de realizar las mediciones de estas cuatro figuras de mérito de forma totalmente automática, dejando para el usuario simplemente la comparación de los resultados obtenidos con los esperados.

Resolución espacial

La resolución espacial consiste en la capacidad del escáner y los procesos de reconstrucción de imagen implicados, de resolver estructuras pequeñas dentro de los objetos capturados en las imágenes.

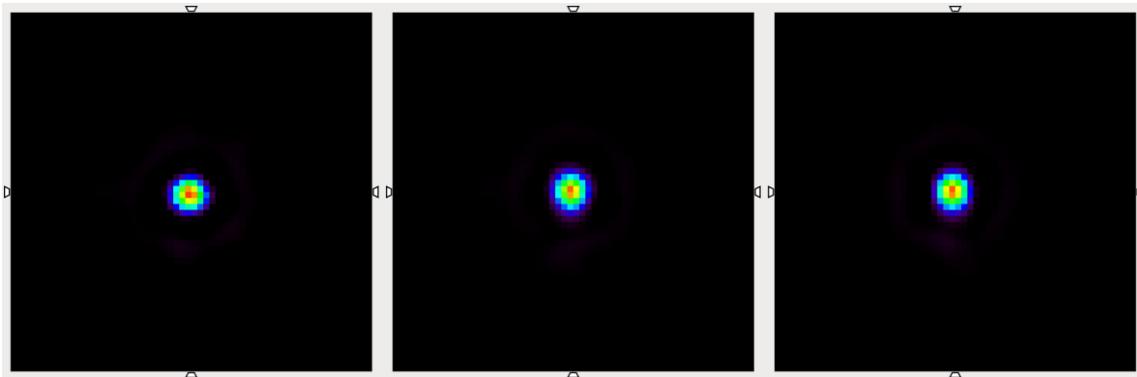


Figura 7 – Imagen PET del fantoma de fuente puntual.

El valor concreto de la resolución espacial proporciona un tamaño característico por debajo del cual el escáner es incapaz de resolver.

Para cuantificar la resolución espacial en los escáneres PET, el protocolo NEMA propone un método que precisa del uso del fantoma de fuente puntual. Consiste en realizar una captura de este, y medir sus dimensiones en los tres ejes de la imagen para comprobar cuánto se aproximan los valores obtenidos a los reales.

Para realizar esta medición, primero se localiza el centro del fantoma en la imagen.

Después, se analizan independientemente los 3 ejes de vóxeles que atraviesan esa ubicación. Se ajusta una función gaussiana a cada eje y se obtiene la anchura a media altura (FWHM) para obtener el tamaño del objeto en cada dimensión, intentando recuperar las dimensiones de 1 milímetro cúbico del fantoma.

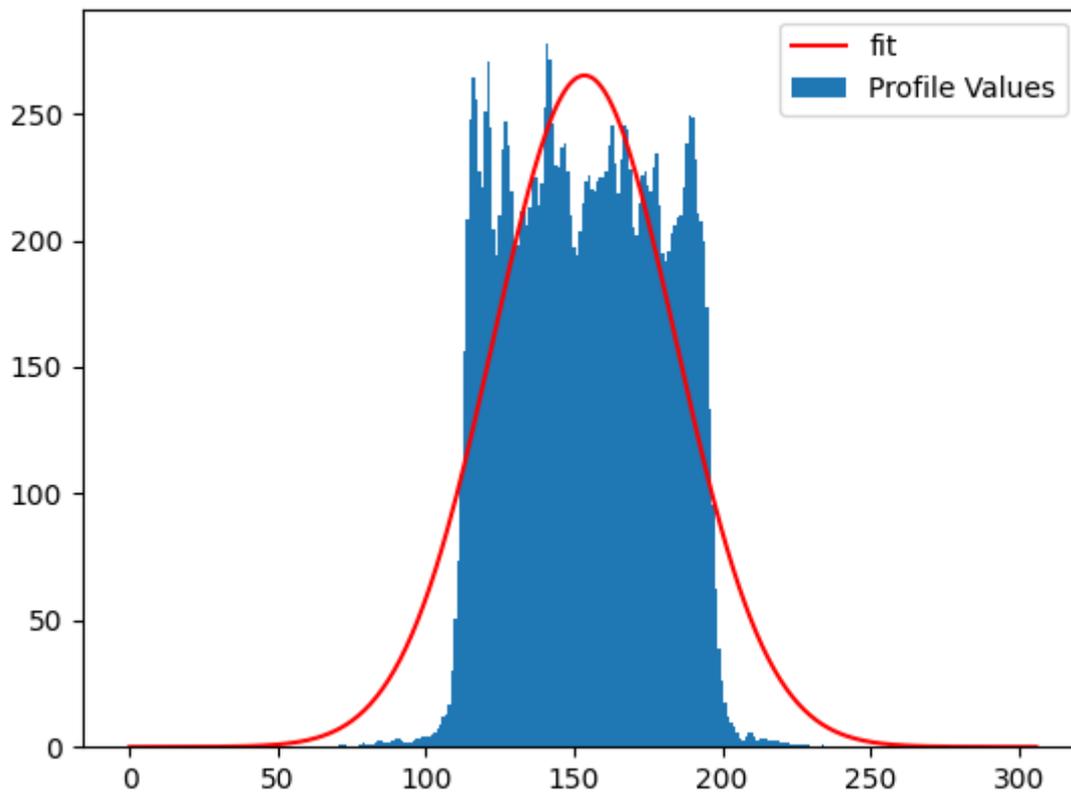


Figura 8 –Gráfico mostrando el ajuste de la función gaussiana a los datos de los vóxeles del eje X que atraviesan el centro de una fuente puntual. Imagen obtenida mediante el uso de las herramientas del módulo ProfileLib de la librería de este proyecto.

Uniformidad

La uniformidad es una forma de medir la variabilidad de la actividad recuperada en la imagen después del proceso de reconstrucción. Esta figura de mérito mide la variabilidad obtenida para una región que se sabe homogénea (variabilidad nula) y proporciona una cuantificación del ruido que se introduce en el proceso de formación de la imagen.

Según el protocolo NEMA, la uniformidad puede medirse al contrastar distintos valores estadísticos obtenidos a partir de un VOI cilíndrico, de dimensiones especificadas, definido en la región de uniformidad del fantoma NEMA IQ.



Figura 9 – Diagrama del fantoma NEMA IQ indicando la ubicación de la región de uniformidad.

La región de uniformidad consiste en una zona amplia de la cavidad, lo que permite obtener una gran cantidad de datos del fantoma para realizar la estadística. Cuanto mayor sea el volumen de datos, más precisa será la medición.

El tamaño del VOI en el fantoma NEMA IQ está definido por el protocolo NEMA.

Para obtener una medida con mayor estadística que la que el fantoma NEMA IQ ofrece, se puede realizar la misma operación empleando un fantoma de uniformidad. Al ser de mayor volumen, permite definir un VOI de mayor tamaño y, por tanto, obtener mayor cantidad de datos para realizar la estadística. Este proceso no está definido específicamente en el protocolo NEMA pero es una medición equivalente.



Figura 10 – Imagen del fantoma de uniformidad indicando en rojo, la ubicación de la región de uniformidad.

Los valores de la uniformidad son necesarios para calcular tanto los coeficientes de recuperación como los ratios de desbordamiento.

Coeficientes de recuperación

Los coeficientes de recuperación son los ratios de actividad (radioactiva) observada en la imagen en comparación a la actividad real del radioisótopo en el fantoma.

En el protocolo NEMA se describe un proceso de adquisición de los datos necesarios relativamente complejo.

La medición se realiza en la región de las barras del fantoma NEMA IQ.

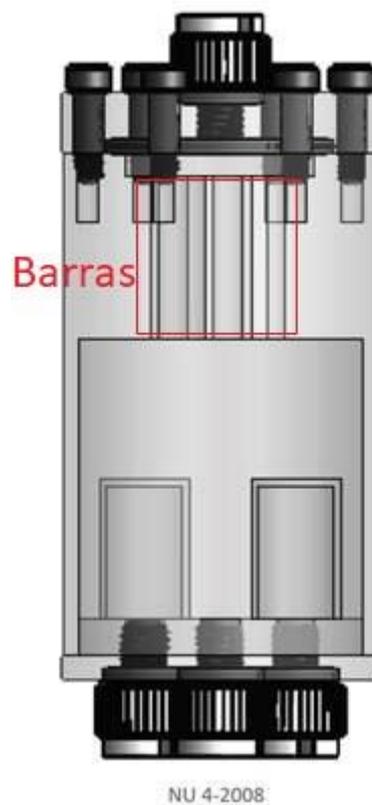


Figura 11 – Diagrama del fantoma NEMA IQ indicando la ubicación de la región de las barras.

Esta región está formada por 5 barras, cada una de un diámetro distinto (5, 4, 3, 2 y 1 milímetros de diámetro), alrededor del eje central del fantoma, a una distancia de 7 milímetros y separadas equitativamente entre sus centros. Cada barra tiene un diámetro distinto para comprobar el ratio de recuperación en diferentes volúmenes.

En primer lugar, se suman por planos transversales los valores de los vóxeles que pertenecen a esta región (dejando un espacio al principio y al final) en un único plano en el que observamos 5 círculos, uno por cada barra.

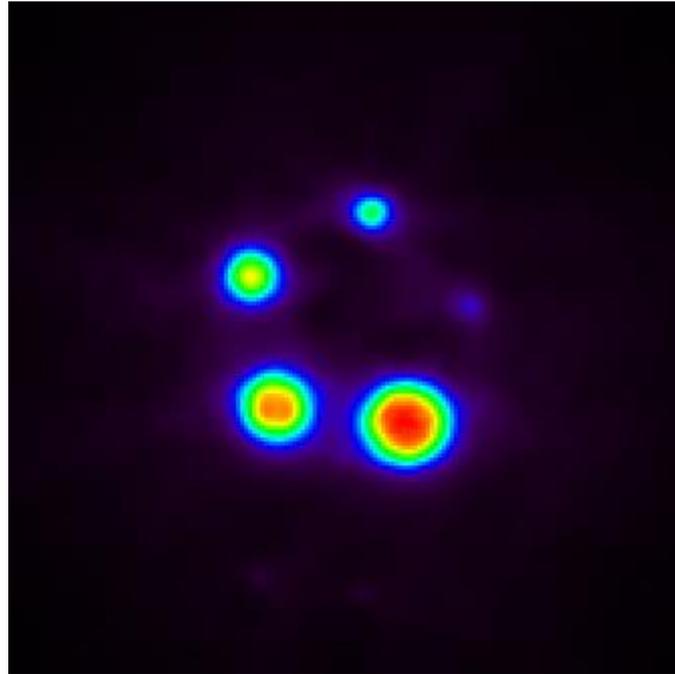


Figura 12 – Imagen PET del plano resultante del sumatorio de los planos transversales de la región de barras del fantoma IQ involucrados en el cálculo de los coeficientes de recuperación. Imagen obtenida mediante el uso de las herramientas del módulo Validations de la librería de este proyecto.

Después, se define un VOI en cada barra del plano resultante para obtener la localización del vóxel que ha sumado el mayor conteo de eventos en cada una de ellas.

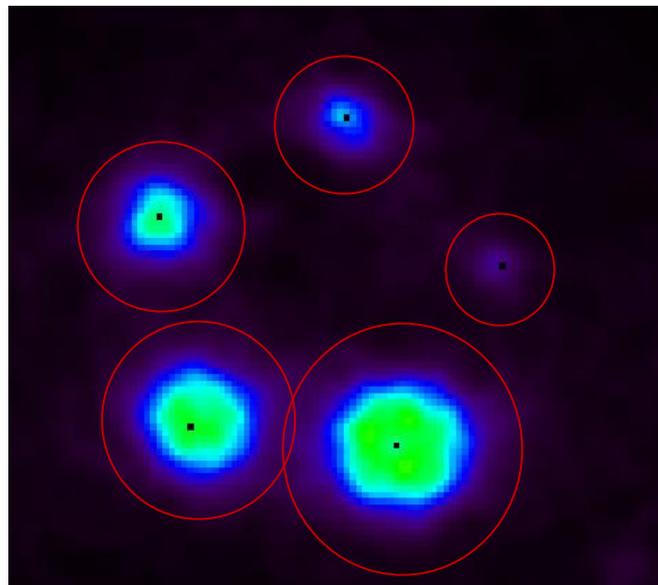


Figura 13 – Imagen PET de un plano transversal de la región de barras del fantoma IQ involucrados en el cálculo de los coeficientes de recuperación. En rojo se indica el VOI considerado para cada barra. El vóxel negro en el interior de cada barra muestra la ubicación del vóxel con el valor máximo de estas,

obtenido del plano de la figura 12. Imagen obtenida mediante el uso de las herramientas del módulo Validations de la librería de este proyecto.

Por último, tomando la imagen original, se recuperan los valores de los vóxeles que corresponden a estas coordenadas a lo largo de los planos transversales que se han sumado, generando así una lista de valores de vóxeles para cada barra de la región.

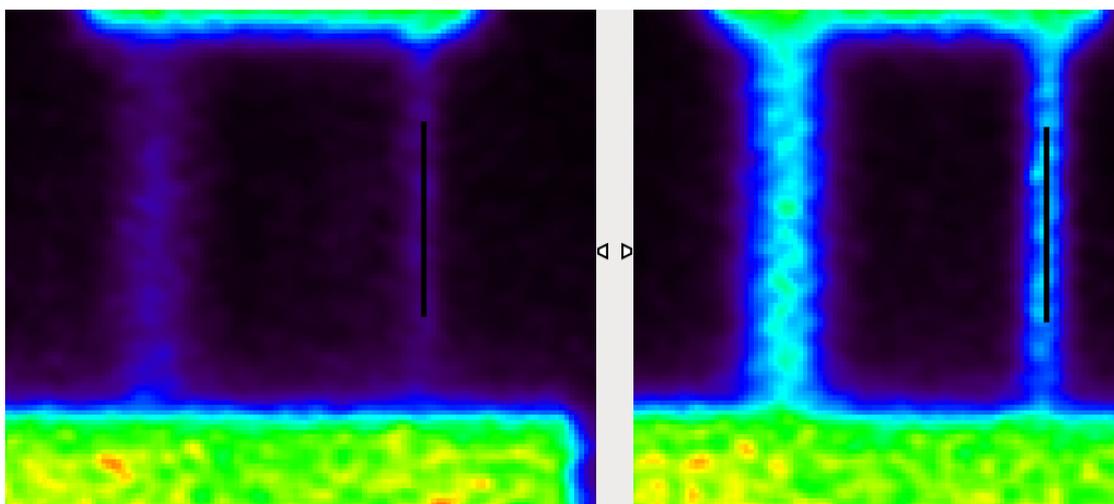


Figura 14 – Imagen PET de la región de las barras, perspectivas coronal y sagital. Los vóxeles negros en el interior de la barra indican los vóxeles considerados para el cálculo de los coeficientes de recuperación de dicha barra. Imagen obtenida mediante el uso de las herramientas del módulo Validations de la librería de este proyecto.

Estos valores son los que serán empleados para obtener las mediciones pertinentes para calcular los coeficientes de recuperación de cada barra.

Ratio de desbordamiento

El ratio de desbordamiento hace referencia al ratio de actividad entre las zonas “frías” (con poca actividad radioactiva) y la zona “caliente” (la región de uniformidad del fantoma NEMA IQ).

Esta medición se realiza en la región del aire y del agua del fantoma NEMA IQ. Consta de dos pequeñas cavidades estancas dentro de la propia cavidad del radioisótopo del fantoma, una de ellas rellena de agua y la otra simplemente con aire.



Figura 15 – Diagrama del fantoma NEMA IQ indicando la ubicación de la región del agua y el aire.

Las mediciones se realizan sobre los vóxeles ubicados en 2 VOI, uno en la región del agua y otro en la región del aire del fantoma.

2.2. Soluciones existentes

Actualmente, existe poca información disponible alrededor de los procedimientos que se emplean en distintas instituciones de investigación, o centros de empresas privadas, para analizar la calidad de imagen de los escáneres PET 3D siguiendo el protocolo NEMA.

En el caso de la empresa en la que se realizaron las prácticas laborales, Bruker, el análisis de imagen se desarrolla de forma manual, apoyándose en software de visualización y análisis de imagen, tal como AMIDE (Figura 16), para identificar y

ubicar las regiones interesantes del fantoma en la imagen, definir los VOI en las regiones pertinentes y recabar la información que se precise.

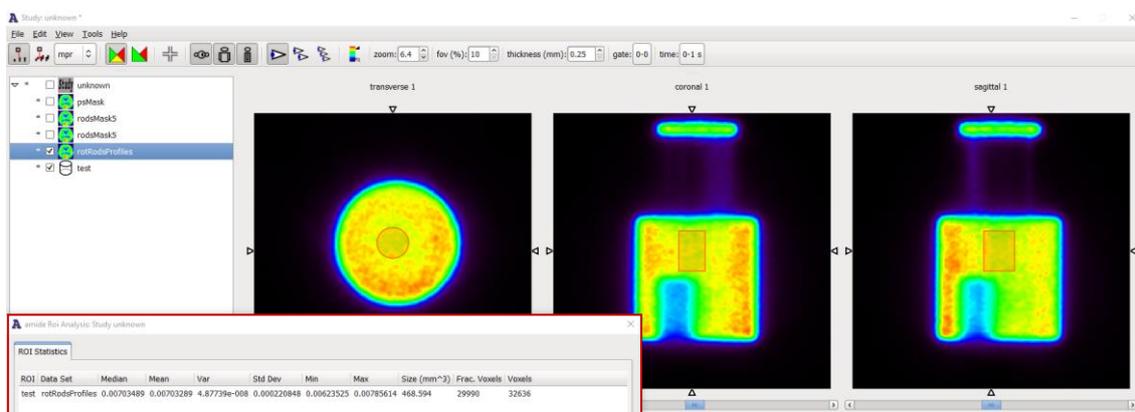


Figura 16 – Imagen de AMIDE mostrando una imagen PET de un fantoma NEMA IQ con un VOI definido en la región de la uniformidad. Marcado en rojo se encuentran las estadísticas del VOI calculadas por AMIDE.

Como excepción para el protocolo NEMA, ya que es un proceso que se hace con cierta frecuencia, se emplea un script que, introduciendo las coordenadas de las regiones de interés, calcula y devuelve los valores necesarios para realizar la validación automáticamente. Esto, sin embargo, no elimina el paso anterior, que conlleva que una persona abra y visualice la imagen para poder ubicar las regiones de interés manualmente.

Por otro lado, más allá de Bruker, existen una serie de herramientas software para crear flujos de procesamiento de imagen PET. Algunos ejemplos serían: APPIAN, PMOD, Pypes, MIAKAT y NiftyPET. Aunque muchos de estos ejemplos contienen algunas de las herramientas básicas que incluye este proyecto (en concreto la capacidad de definir VOIs), ninguno está enfocado a analizar la calidad de la imagen ni seguir un protocolo estandarizado como NEMA. Muchos incluyen visores de imagen con opciones para realizar un proceso similar al anteriormente mencionado en Bruker, es decir, manual.

Entre los ejemplos destacamos NiftyPET, una gran librería desarrollada en Python (al igual que la solución propuesta en este proyecto) con gran variedad de herramientas para reconstrucción y análisis de imágenes PET. Sin embargo, al igual que el resto, no tienen herramientas para cuantificar la calidad de la imagen.

3. Análisis del problema

El principal problema de las soluciones actuales es que no incluyen la capacidad de realizar un análisis de calidad de imagen personalizado, o acorde al protocolo NEMA. Relegan en visores de imagen para realizar esta tarea de forma manual u otorgan las herramientas en un entorno cerrado que no permite acoplarlas a un sistema mayor para automatizar las pruebas.

Esto provoca varios problemas de repetibilidad y reproducibilidad de las pruebas, ya que, el simple hecho de tomar la misma región de interés de una prueba anterior que esté desplazada un pixel en cualquier dirección puede resultar en una diferencia notable en los resultados o, incluso, ser la diferencia entre una prueba realizada correctamente o no.

Además, como cualquier proceso repetitivo realizado de forma manual, induce un punto de cuello de botella en el flujo de trabajo a la hora de verificar la calidad de imagen de un escáner o método de reconstrucción.

3.1. Solución propuesta

Frente a esta problemática, el proyecto ofrece al usuario un abanico de herramientas que le facilitan, tanto el desarrollo de procedimientos de análisis de imagen automáticos personalizados, como integrar estos procedimientos (y los basados en el protocolo NEMA ya incluidos) en su propio software.

Esta solución posibilita, por ejemplo, la generación de flujos de pruebas de calidad de imagen no supervisados que permitan realizar análisis reproducibles y repetibles, además de abrir las puertas a la creación de flujos de pruebas de grandes volúmenes de imágenes.

4. Metodología

Dada la gran falta de información disponible sobre herramientas similares a la que se va a desarrollar, no quedaban claros cuales iban a ser todos los requisitos y las necesidades del proyecto. Tan solo contábamos con los objetivos finales y los prerrequisitos básicos para ello.

Debido a esto, y a la comodidad de trabajar con metodologías *agile* por experiencias pasadas, se decidió tomar una metodología ágil *Scrum*.

La metodología Scrum es una metodología *agile* muy flexible. Permite adaptar los requisitos y la estructura del proyecto a las necesidades que se van encontrando durante el desarrollo de este.

Un ejemplo de esto sería el momento en que nos percatamos de que el posicionamiento de un fantoma dentro de un detector es responsabilidad de un ser humano. Por tanto, podríamos esperar que las imágenes tuvieran los objetos ligeramente desalineados con los ejes de coordenadas del espacio. Por ello se decidió revisar parte de la implementación ya realizada y probada para incluir la posibilidad de realizar rotaciones en esta.

Una vez el objetivo y la metodología fueron decididos, la estrategia de trabajo consistió en ir elaborando las distintas herramientas de forma independiente, comprobando su funcionamiento individual antes de continuar con la siguiente. Cuando se terminaba un módulo de herramientas debía comprobarse que todas funcionaban entre sí sin producir ningún tipo de fallo.

Las *daily* (reuniones diarias) características de *scrum*, no fueron estrictamente necesarias para mantener al tanto del desarrollo como para resolver impedimentos que podían surgir debido al desconocimiento sobre algunos aspectos del tema que se estaba tratando, imagen PET.

Cada semana se realizaba una reunión de control sobre el proyecto. Se revisaban los objetivos alcanzados, los objetivos pendientes y, si era necesario, ampliar o recortar algún requisito u objetivo.



Desarrollo de una librería software para el análisis y validación de imágenes 2D/3D PET basado en el protocolo NEMA

Además, el desarrollo y sus objetivos (o hitos) se repartieron en sprints mensuales. De esta forma se podía llevar fácilmente el seguimiento del proyecto y hacer retrospectiva de sprints anteriores para mejorar la organización de los futuros.

Para organizar las tareas que iban surgiendo conforme avanzaba el proyecto, modificarlas, tomar anotaciones y poder conocer el estado global del proyecto, se decidió por el uso de Trello.

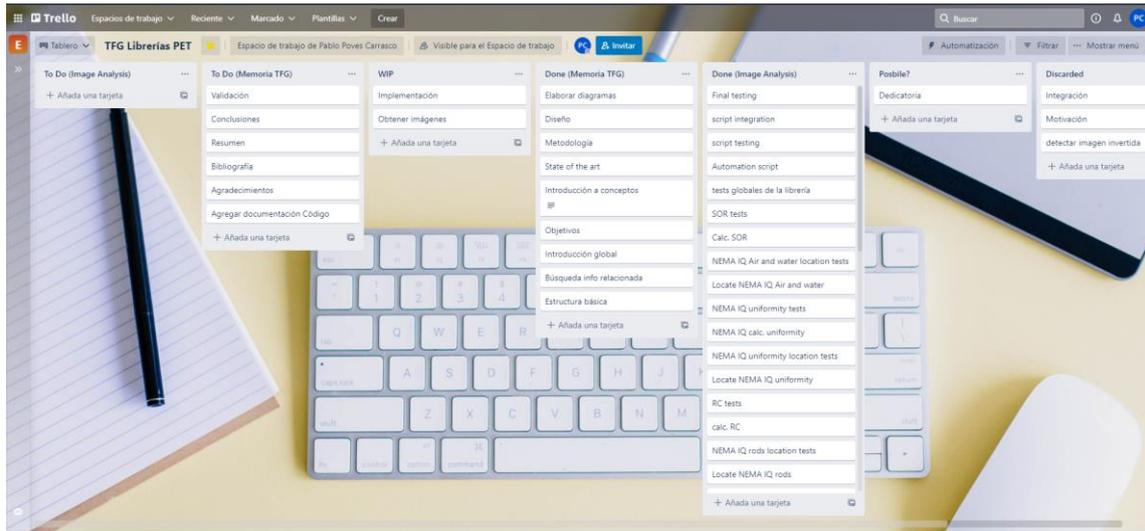


Figura 17 – Imagen del tablero de Trello empleado durante el proyecto.

Trello es una herramienta online gratuita (con opciones de mejora de pago) desarrollada por Atlassian para administración de proyectos basado en tarjetas visuales estilo Kanban (otra metodología *agile*). Esta herramienta fue especialmente útil hacia el final de la realización del proyecto, durante la elaboración de esta memoria, para visualizar el trabajo pendiente o en curso y poner objetivos en el tiempo antes de la entrega final.

5. Diseño de la solución

Antes de introducirnos en la arquitectura y el diseño de la solución propuesta, vamos a concretar el marco de tecnologías que se han empleado para el desarrollo del proyecto.

5.1. Tecnología utilizada

Jira

Durante el proyecto, y debido a que luego parte de este iba a ser integrado en la rama de producción de Bruker, se empleó el software Jira (comentado más adelante) con un repositorio Git integrado para el control de versiones.

Jira, al igual que Trello, es una herramienta online de administración de tareas de un proyecto desarrollado por Atlassian. A diferencia de Trello, Jira no usa el sistema de tarjetas de tarea visuales sino que opta por una gestión en forma de flujos de trabajo y está más enfocado al desarrollo de software.

Durante la realización del proyecto, y como se explicó en el apartado de metodología, se empleó Trello para la gestión de las tareas. Sin embargo, dado que el sistema de control de versiones estaba construido mediante Jira, esta herramienta también fue empleada.

Confluence

Confluence es otra herramienta desarrollada por Atlassian, a diferencia de Trello y Jira, en lugar de centrarse en la gestión de tareas de un proyecto se centra en la gestión de la documentación corporativa. Realiza un papel similar al de una wiki privada y permite almacenar documentación y artículos, y relacionarlos con tareas o flujos de trabajo de Jira.

Durante la fase final de la realización del proyecto, se generó la documentación relacionada con las librerías y se integró con el resto de la documentación de Bruker.

Además, permitió acceso a documentos e información que fueron relevantes para el desarrollo del código de este trabajo.

Visual Studio Code

Visual Studio Code es un editor de código desarrollado por la compañía Microsoft. Durante el proyecto, este fue el entorno principal de desarrollo de las librerías. Gracias a su popularidad, posee una gran cantidad de *plugins* de utilidad a la hora de desarrollar código en el lenguaje Python.

Python

Python es un lenguaje de programación interpretado, multiparadigma y es el empleado en el desarrollo de las librerías de este proyecto. Son varias las razones que llevaron a decidirse por este lenguaje por encima de otros.

En primer lugar, Python tiene una gran comunidad de desarrolladores y una variedad muy amplia y rica de librerías de apoyo. Esto es elemental, ya que algunos de los algoritmos o cálculos empleados en este proyecto ya han sido programados en algunas de ellas. Gracias a esto, con solo importarlas no es necesario tener que rehacerlo.

En segundo lugar, Python es un lenguaje muy estable y ampliamente empleado en ámbitos de desarrollo para ciencia de datos y computación, lo que propicia que entre las librerías disponibles, se encuentren con más probabilidad las herramientas que puedan servirnos.

Además, Python ofrece un modo interactivo, ideal para depurar código que está realizando gran cantidad de cálculos con volúmenes importantes de datos.

Por último, Python soporta orientación a objetos. Durante la carrera de ingeniería informática, los lenguajes orientados a objetos son los más empleados en las asignaturas que incluyen programación. De hecho, el primer lenguaje que se imparte es Java, y el que se emplea durante la asignatura de Ingeniería de Software es C#. Ambos poseen orientación a objetos. Esto se traduce en familiaridad a la hora de programar con Python.

Librerías Python empleadas

Durante el proyecto, como ya mencionamos, se emplearon algunas librerías para agilizar el desarrollo.

NumPy

NumPy es una potente librería que ofrece herramientas de alto nivel de cálculo matemático complejo, además de proporcionar un manejo de grandes volúmenes de datos, mucho más rápido de lo que sería posible en simple Python.

Matplotlib

Matplotlib es una librería de visualización de datos que está bien integrada con NumPy.

Scikit-image

Scikit-image ofrece múltiples herramientas para el tratamiento y análisis de imágenes tanto 2D como 3D.

Astropy

Astropy es una librería que, aunque enfocada a la astronomía, incluye potentes herramientas de tratamiento de datos.

Doxygen

Doxygen es una herramienta de generación de documentación que soporta una amplia variedad de lenguajes de programación. Su atractivo principal es su amplia capacidad de configuración y su sencillez a la hora de comentar el código usando su sintaxis.

Además, Doxygen ofrece la generación de este código en distintos formatos. Destacamos la creación de un portal web interactivo en HTML, que proporciona herramientas de búsqueda y facilita la navegación en él, y la generación de documentación en LaTeX, también con indexación interactiva.



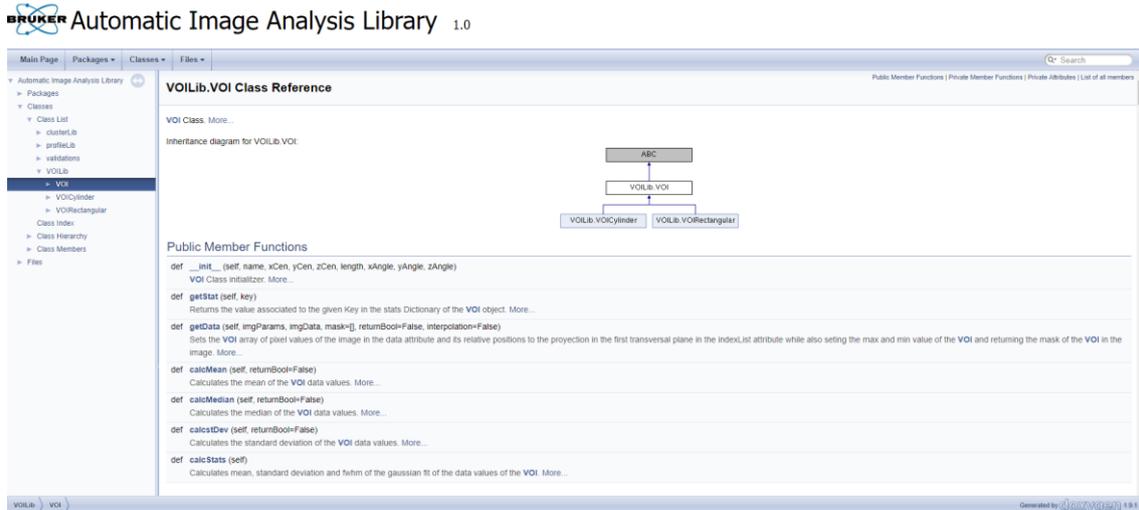


Figura 18 – Imagen ejemplo documentación Doxygen en HTML.

En este proyecto se generó la documentación en ambos formatos. El formato HTML fue empleado durante el desarrollo de las herramientas de más alto nivel de la propia librería. El documento LaTeX fue convertido en PDF y pasó a formar parte de la documentación corporativa de Bruker en Confluence.

6.4 VOILib.VOI Class Reference

61

Public Member Functions

- `def __init__(self, name, xCen, yCen, zCen, length, xAngle, yAngle, zAngle)`
VOI Class initializer.
- `def getStat(self, key)`
Returns the value associated to the given Key in the stats Dictionary of the VOI object.
- `def getData(self, imgParams, imgData, mask=[], returnBool=False, interpolation=False)`
Sets the VOI array of pixel values of the image in the data attribute and its relative positions to the projection in the first transversal plane in the indexList attribute while also setting the max and min value of the VOI and returning the mask of the VOI in the image.
- `def calcMean(self, returnBool=False)`
Calculates the mean of the VOI data values.
- `def calcMedian(self, returnBool=False)`
Calculates the median of the VOI data values.
- `def calcstDev(self, returnBool=False)`
Calculates the standard deviation of the VOI data values.
- `def calcStats(self)`
Calculates mean, standard deviation and fwhm of the gaussian fit of the data values of the VOI.

Private Member Functions

- `def _calcRotMatrix(self)`
Calculates the rotation matrix of the given rotation angles for each axis.
- `def _calcTransToOOMatrix(self)`
Calculates the translation matrix of the VOI to the ordinate origin.
- `def _calcTransBackToCoords(self)`
Calculates the translation matrix of the VOI from the ordinates origin back to its original coordinates.
- `def _setVoxelIndexList(self, voxelIndexList)`
Sets the VOI voxelIndexList to the given array of values.
- `def _setVoxelDataAndPos(self, voxelData, voxelPositions)`
Sets the VOI voxelData and voxelPositions to the given arrays of values.
- `def _setSum(self, sum)`
Sets the VOI sum to the given value.
- `def _addSum(self, value)`
Adds the given value to the current value of the VOI sum attribute.
- `def _setStat(self, key, val)`
If the key does not figure in the stats Dictionary of the VOI object, it will add a new one to it with the given value.
- `def _rotatePosition(self, pos, imgParams)`
Returns the position of the voxel that corresponds to the rotated given position based on the rotation matrix of the VOI.

Figura 19 – Imagen ejemplo documentación Doxygen en LaTeX.

5.2. Arquitectura básica

La librería se compone de 6 módulos agrupados en 3 bloques de abstracción. Estos tres bloques se comportan, por así decirlo, como si de una torre se tratara, colocando en lo alto de esta las herramientas de mayor abstracción y en la base las de menor.

En el bloque base encontramos los módulos con las herramientas más básicas, genéricas y de menor abstracción. Estas son (prácticamente) independientes entre sí y están orientadas a los usuarios más expertos.

En el segundo nivel, encontramos herramientas de más alto nivel de abstracción que el anterior, que han sido creadas, en gran medida, usando las herramientas de la base y, por lo tanto, dependen de ella pero no de los bloques superiores.

Por último encontramos el tercer bloque, construido con las herramientas de la base y que también depende del bloque inferior para funcionar. Es el bloque de mayor nivel de abstracción y está orientado a usuarios con poca experiencia en programación.

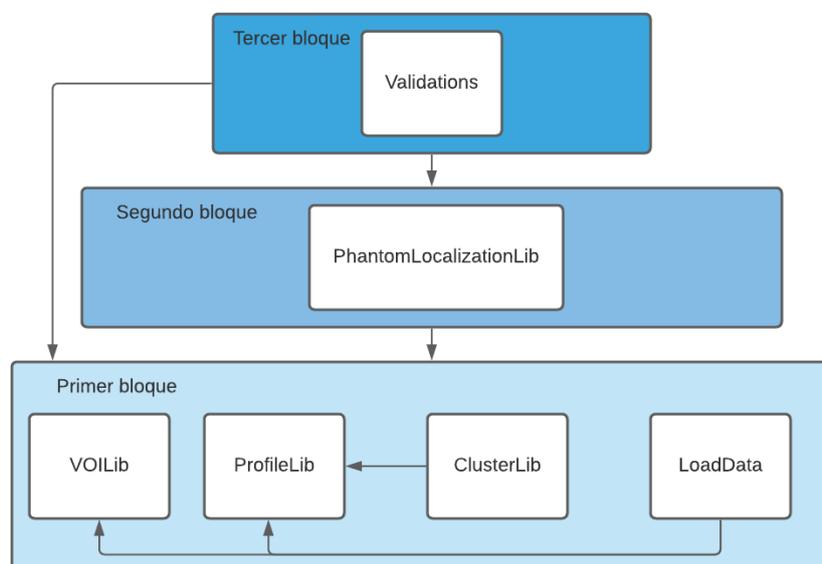


Figura 20 – Diagrama de dependencias y niveles de abstracción de los módulos de la librería del proyecto.

Ya que en Python no existe estrictamente el atributo privado, para atributos y métodos, se han marcado usando la convención “_” antes del nombre en la implementación del código.

VOILib

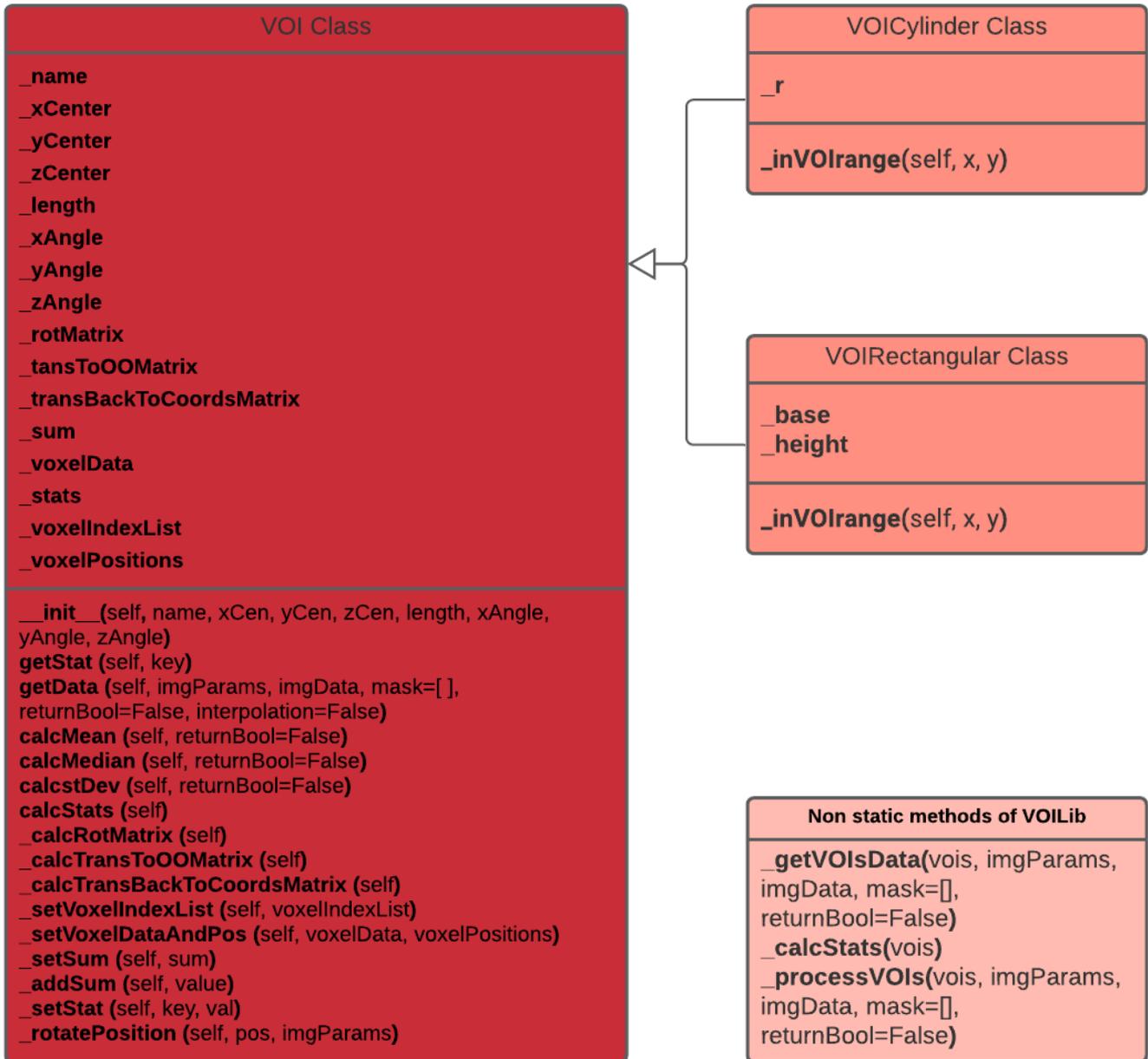


Figura 21– Diagrama del módulo VOILib

ProfileLib

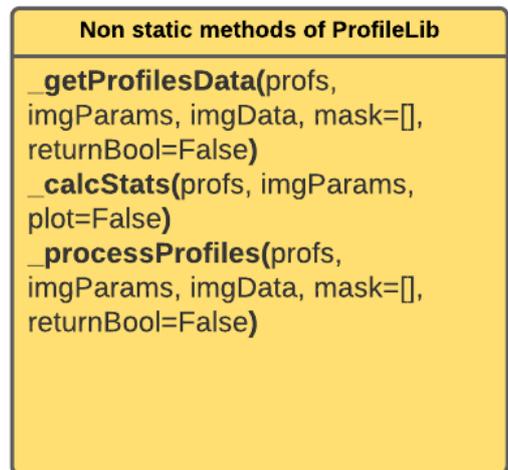


Figura 22– Diagrama del módulo ProfileLib



ClusterLib

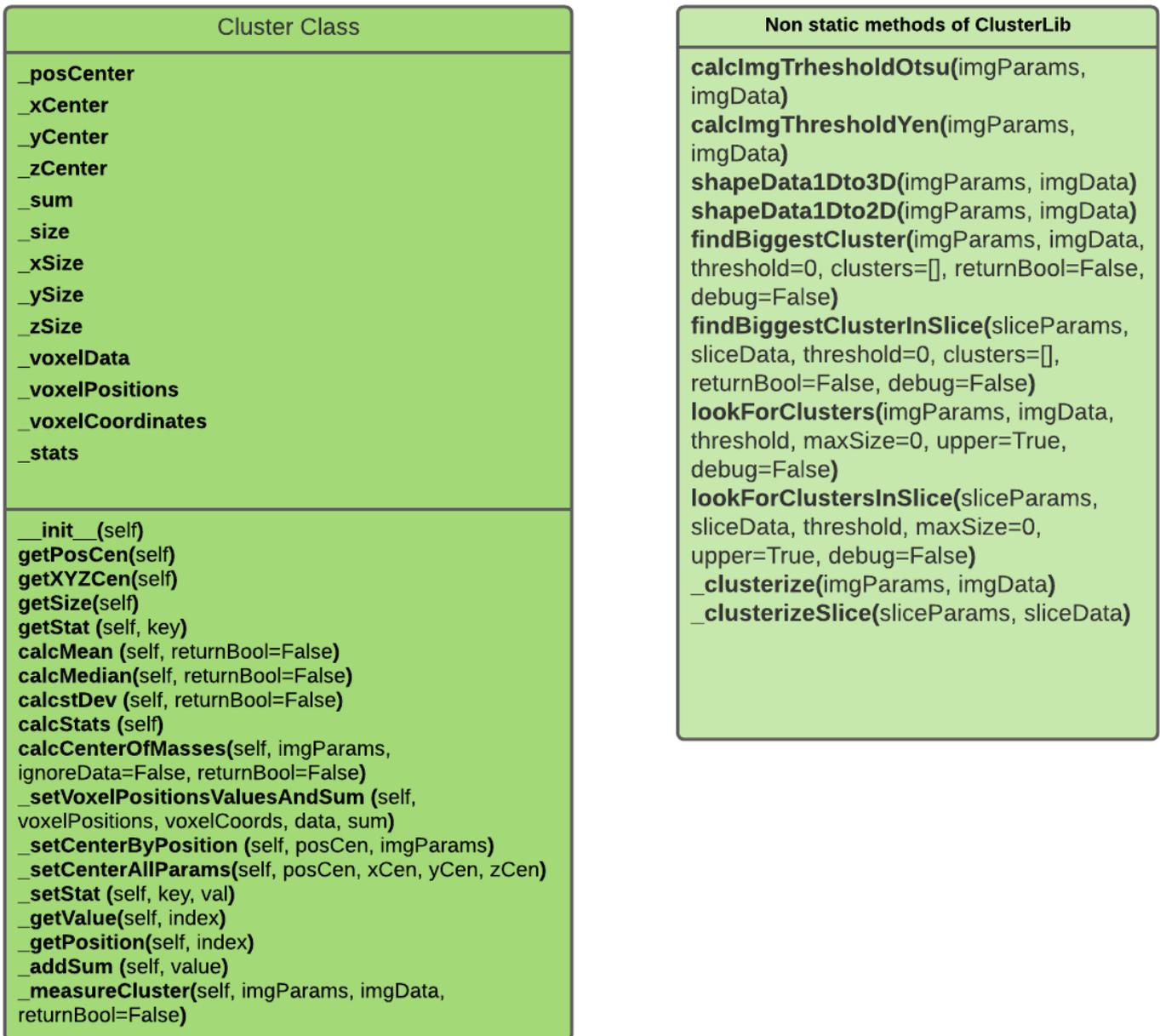


Figura 23– Diagrama del módulo ClusterLib

PhantomLocalizationLib

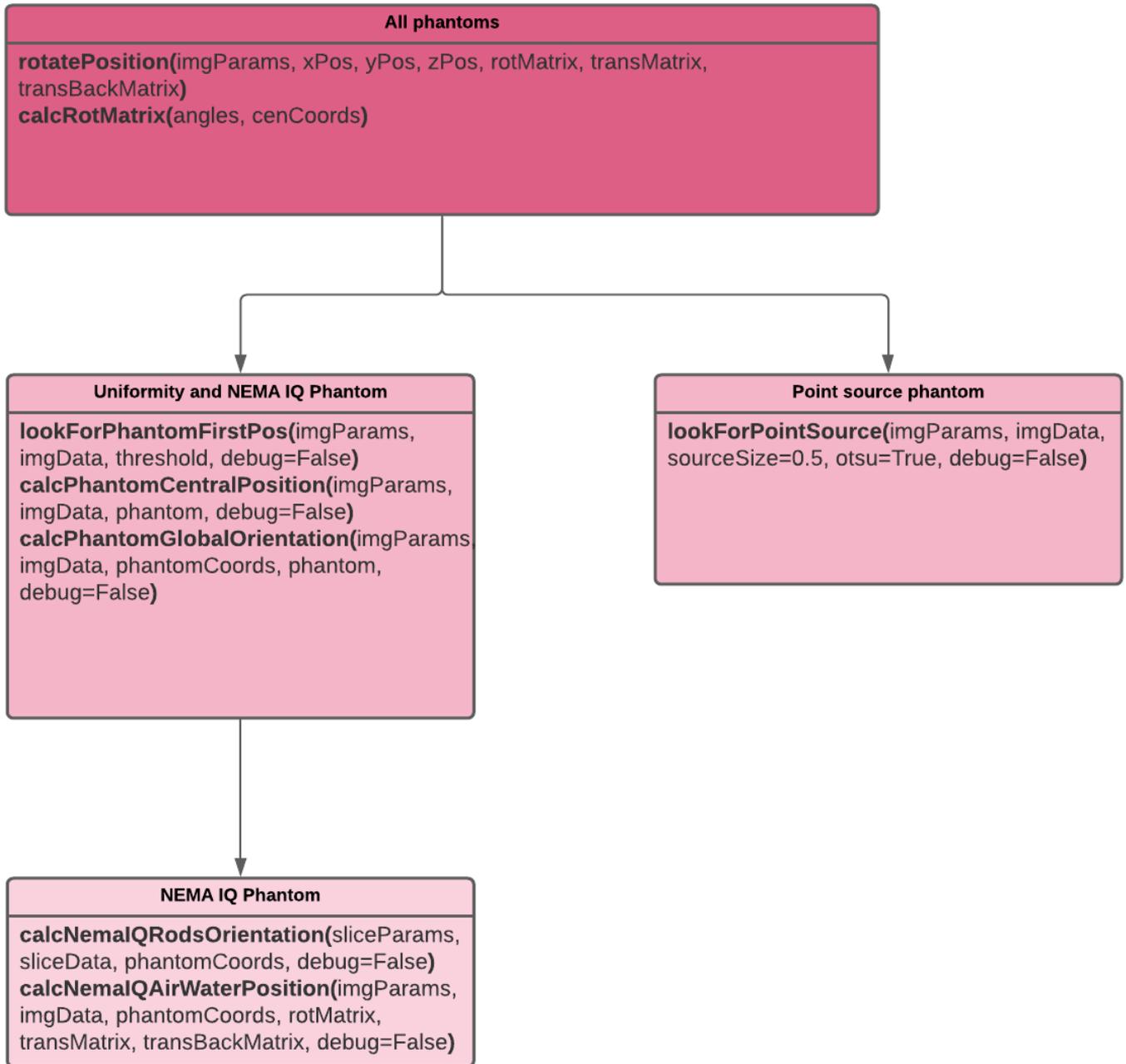


Figura 24– Diagrama del módulo PhantomLocalizationLib



Validations

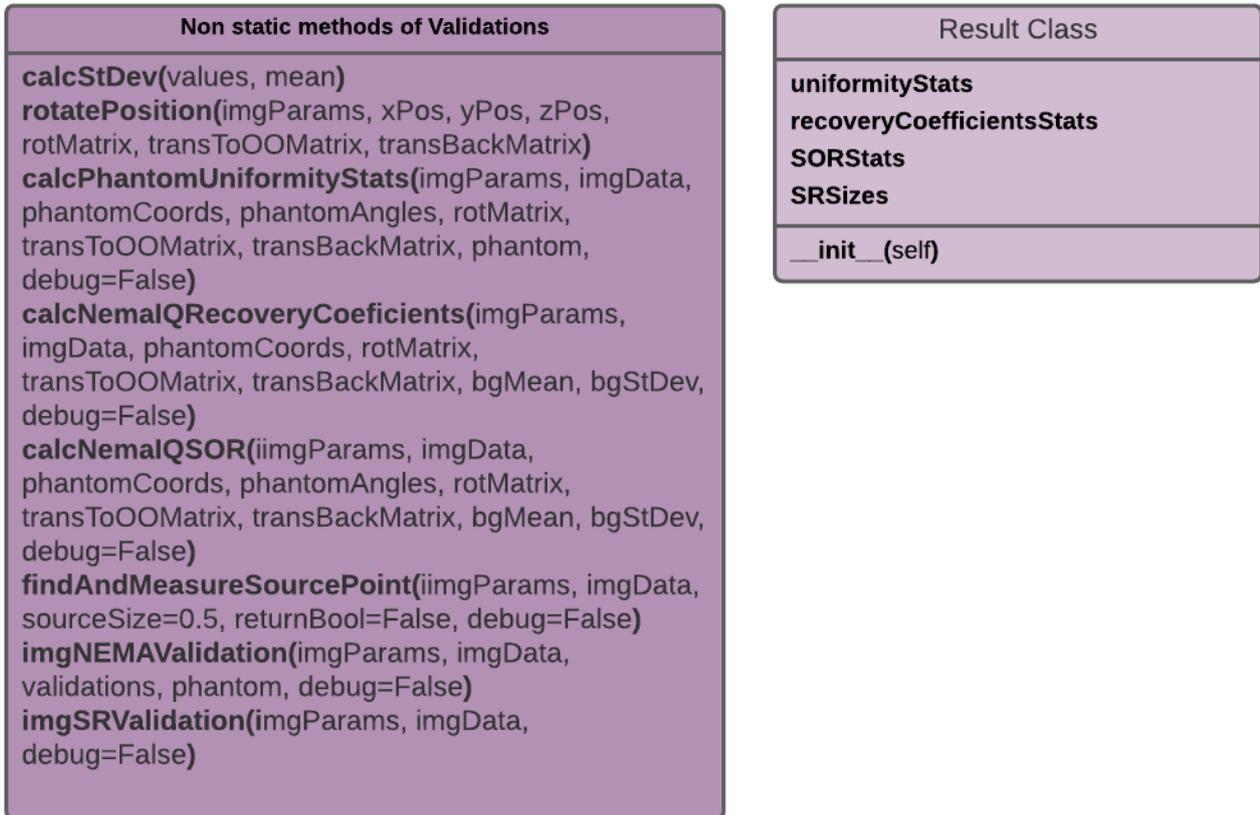


Figura 25– Diagrama del módulo Validations

LoadData

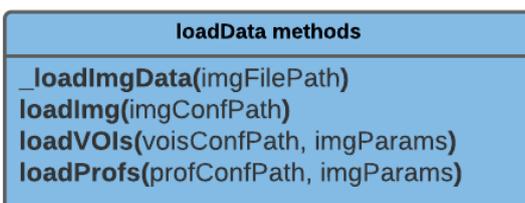


Figura 26– Diagrama del módulo LoadData

5.3. Diseño detallado

VOILib

El módulo VOILib ofrece la posibilidad de declarar VOIs en cualquier lugar del espacio tridimensional.

En concreto, posee la capacidad de declarar VOIs con formas cilíndricas y rectangulares. Para ello, emplea la clase VOI abstracta, que contiene todos los atributos y métodos comunes a todos los “VOI”, de la que se heredan las clases “VOICylinder” y “VOIRectangular”. Estas últimas agregan sus propios atributos y la implementación de un método en sobrecarga propio.

Una vez definidos, VOILib permite (si se encuentran dentro del espacio de la imagen claro) recabar información sobre la región y realizar cálculos para extraer estadísticas sobre esta.

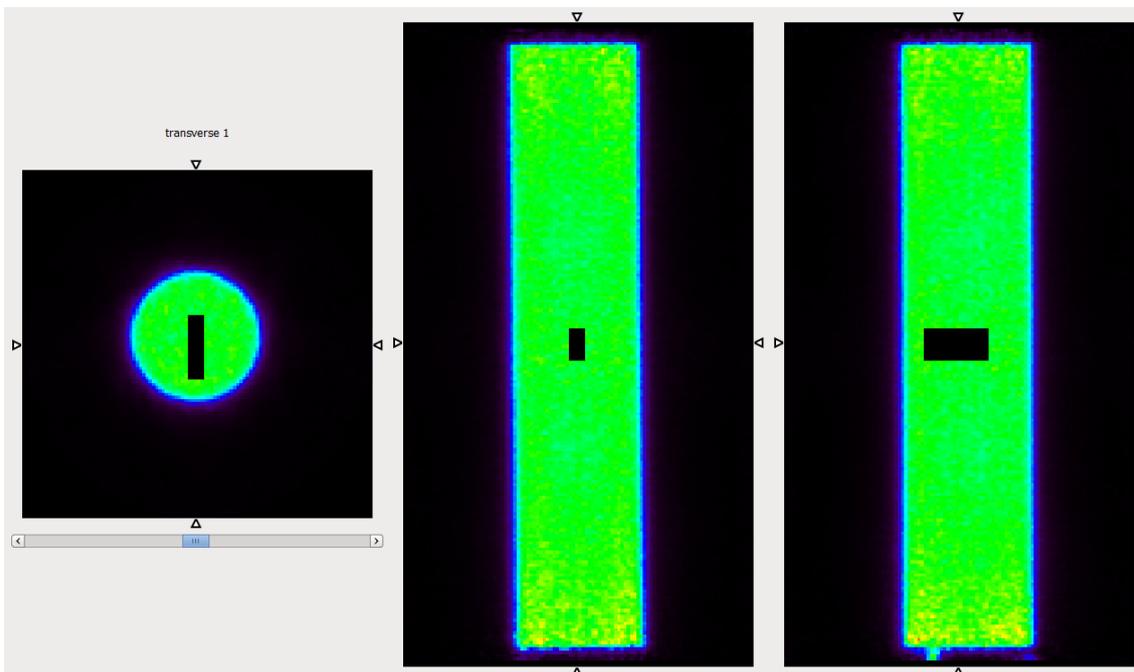


Figura 27 – Imagen PET del fantoma de uniformidad. Marcado en negro, la máscara de un VOI rectangular de dimensiones 5 x 20 x 10 milímetros. Imagen obtenida mediante las herramientas del módulo VOILib de la librería de este proyecto.

VOILib permite declarar y procesar múltiples VOI simultáneamente y de manera independiente de la imagen, lo que permite tanto usar el mismo VOI en distintas imágenes como usar con distintos VOI en la misma imagen e incluso, trabajar con distintos VOI y distintas imágenes.

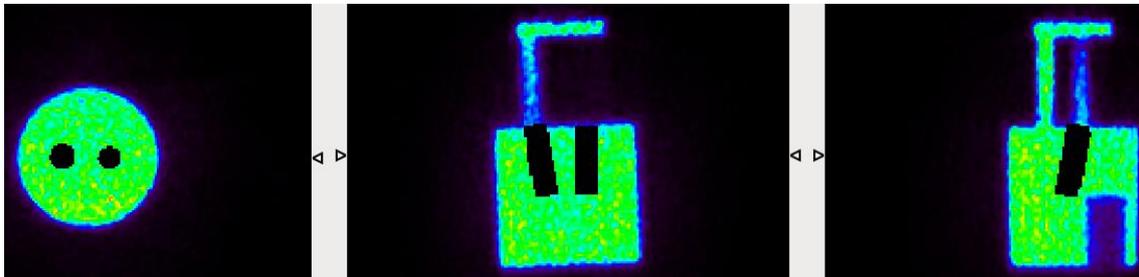


Figura 28 – Imagen PET de la máscara de 2 VOIs en el fantoma NEMA IQ. Imagen obtenida mediante las herramientas del módulo VOILib de la librería de este proyecto.

Las estadísticas que puede extraer el módulo sobre una región son la media, la mediana, el máximo valor, el mínimo valor, la desviación estándar y el número de vóxeles que la componen.

Una de las características principales de VOILib es la capacidad de declarar los VOI con rotaciones en los tres ejes del espacio. Estas rotaciones se hacen mediante matrices de rotación eulerianas.

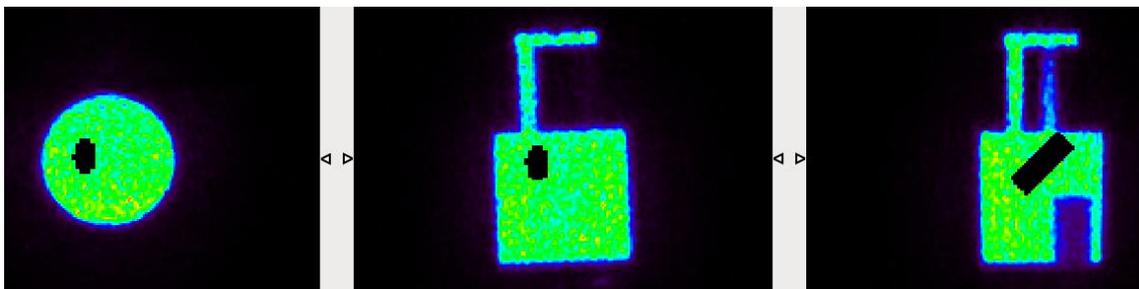


Figura 29 – Imagen PET de la máscara IQ un VOI rotado 45° en el eje X en el fantoma NEMA. Imagen obtenida mediante las herramientas del módulo VOILib de la librería de este proyecto.

ProfileLib

El módulo ProfileLib permite definir y procesar perfiles (líneas) en cualquier lugar y dirección del espacio para recabar información y estadísticas sobre los vóxeles que son atravesados por estas. Para ello se apoya en la clase “Profile”.

Para definir un perfil tan solo es necesario escoger un punto en el espacio (unas coordenadas X, Y, Z), un ángulo de inclinación, e indicar el eje sobre el que se inclina este (X, Y o Z).

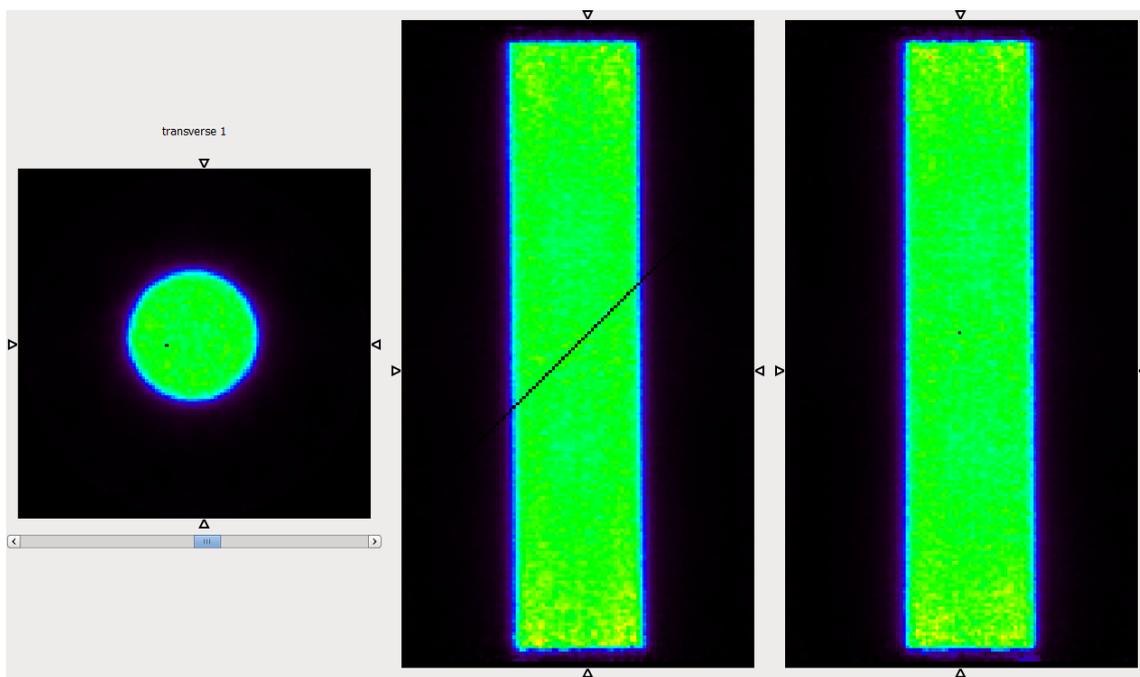


Figura 30 – Imagen PET del fantoma de uniformidad. Marcado en negro, un perfil con un ángulo de 45° en el plano coronal. Imagen obtenida mediante las herramientas del módulo ProfileLib de la librería de este proyecto.

Similar en muchos aspectos al módulo VOILib, ProfileLib permite declarar y procesar tantos perfiles como se desee simultáneamente.

También como VOILib, ofrece la posibilidad de usar múltiples perfiles sobre una misma imagen, emplear los mismos perfiles en múltiples imágenes o trabajar con distintos perfiles en distintas imágenes.

Este módulo permite extraer varias estadísticas sobre los vóxeles que atraviesan un perfil. Además de las básicas (media, máximo, mínimo, desviación estándar), ProfileLib posee la capacidad de calcular la distancia que recorre en el espacio de la imagen y

realizar un ajuste gaussiano de los valores del perfil que también se puede visualizar (ver Figura 8).

Gracias a ello, ProfileLib permite calcular el FWHM del ajuste gaussiano de los datos y, por tanto, medir estructuras de objetos en la imagen. Esto es especialmente útil, por ejemplo, para calcular la resolución espacial.

Además, ProfileLib ofrece la capacidad de definir y trabajar con perfiles en distintos planos simultáneamente.

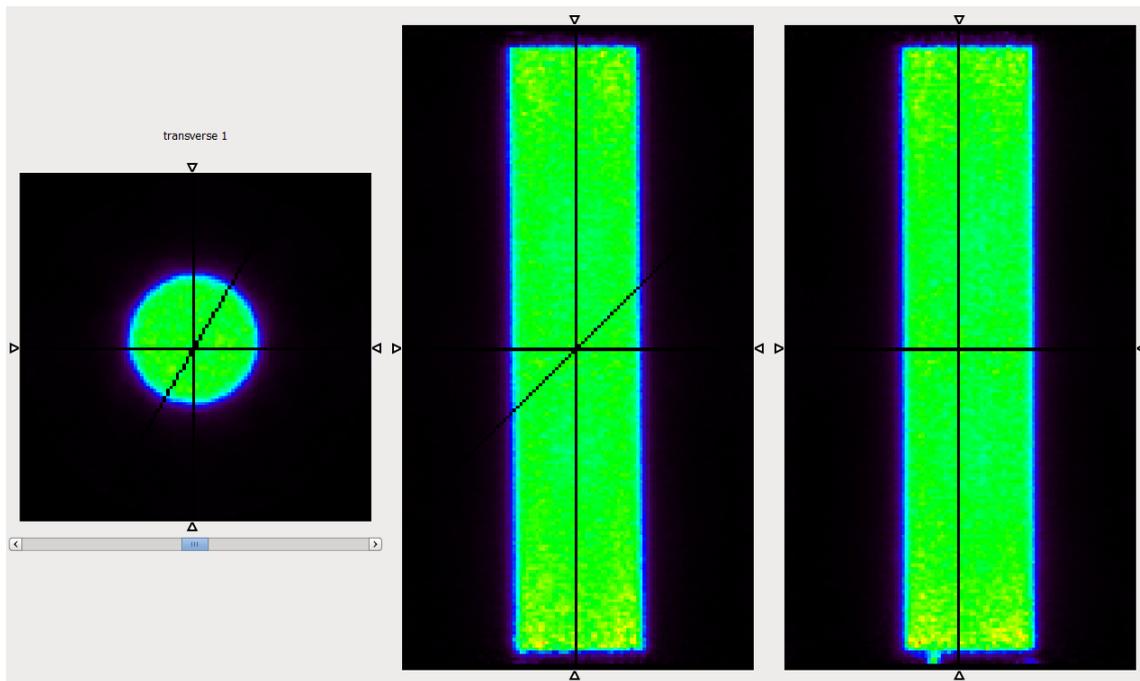


Figura 31 – Imagen PET del fantoma de uniformidad. Marcado en negro, las máscaras de múltiples perfiles con distinta inclinación y en distintos planos. Imagen obtenida mediante las herramientas del módulo ProfileLib de la librería de este proyecto.

ClusterLib

ClusterLib es un módulo que permite *clusterizar* (agrupar en clústeres de vóxeles) una imagen, distinguiendo actividad de fondo automáticamente.

Para poder clusterizar una imagen, ClusterLib ofrece herramientas de umbralización automática de la imagen. Esto es, la obtención de un valor que permite discriminar actividad de fondo. ClusterLib hace principalmente uso del algoritmo de Otsu, pero también tiene implementado el método de Yen.

Una vez clusterizada la imagen, apoyándose en la clase “Cluster”, permite obtener información y estadística sobre los clústeres encontrados.

La clase “Cluster” representa un conjunto de vóxeles que reúnen los siguientes requisitos:

- Son contiguos entre ellos
- Superan la condición de *umbralización* (comparación con un valor umbral) automática

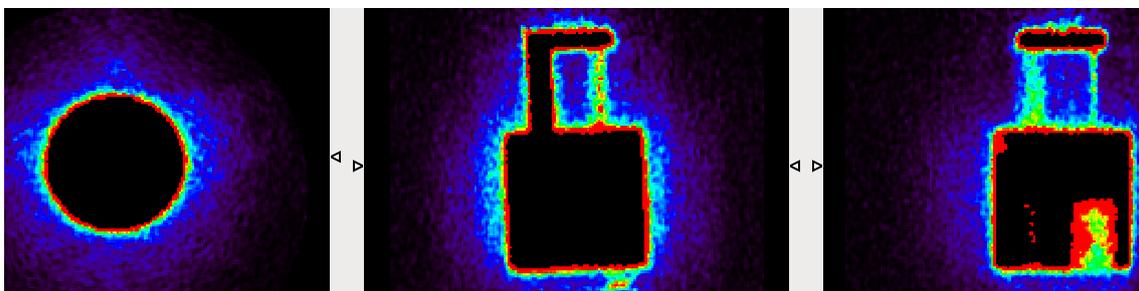


Figura 32 – Imagen PET de una máscara del fantoma NEMA IQ tras ser clusterizada por paso alto. Marcado en negro en el interior del objeto, los vóxeles que han superado la condición de umbralización automática. Imagen obtenida mediante las herramientas del módulo ClusterLib de la librería de este proyecto.

La condición de umbralización consiste en determinar si el valor de un vóxel, en comparación con el valor de umbralización, es:

- Mayor: el vóxel se considera actividad
- Menor: el vóxel se considera fondo

Esto no restringe la clusterización a solo la actividad, si no que ClusterLib ofrece la posibilidad de clusterizar tanto por paso alto (agrupar la actividad) como por paso bajo (agrupar el fondo).

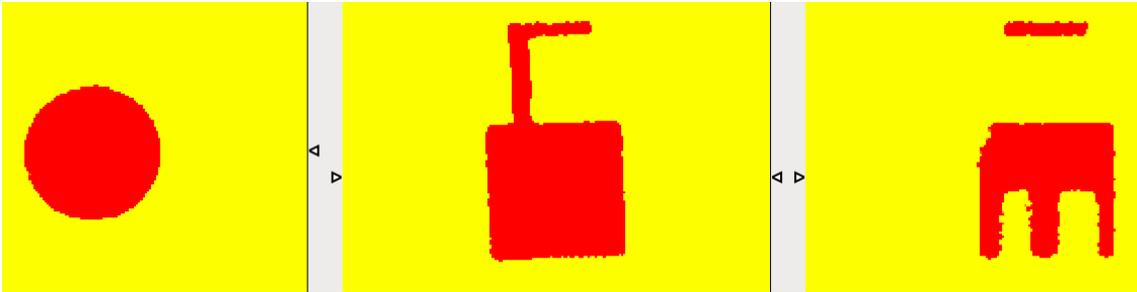


Figura 33 – Imagen PET de una máscara del fantoma NEMA IQ tras ser clusterizada por paso bajo. Marcado en amarillo, los vóxeles que han superado la condición de umbralización automática. Imagen obtenida mediante las herramientas del módulo ClusterLib de la librería de este proyecto.

Al igual que el resto de los módulos de herramientas de análisis genéricas de esta librería, incluye el cálculo de las estadísticas básicas sobre los valores del clúster. En cambio, y exclusivamente en este módulo, permite calcular los centros de masas de los clústeres y, empleando ProfileLib, medir las dimensiones en los ejes X, Y, Z de estos.

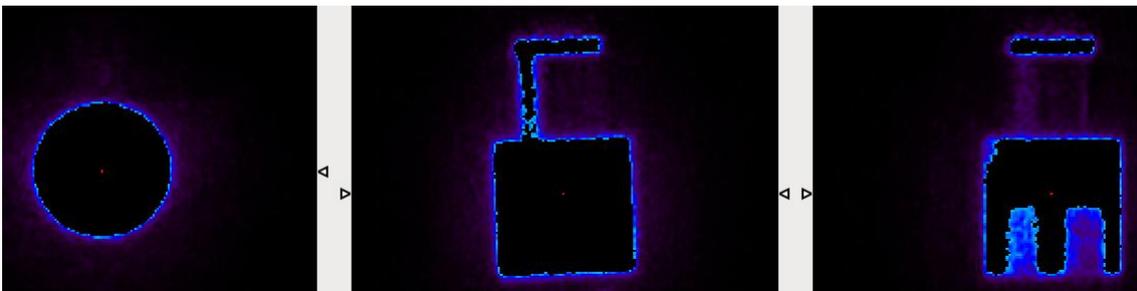


Figura 34 – Imagen PET de una máscara del fantoma NEMA IQ de la Figura 25 tras ser clusterizada por paso alto. Marcado en negro en el interior del objeto, los vóxeles que han superado la condición de umbralización automática. Marcado en rojo, el centro de masas calculado para el clúster. Imagen obtenida mediante las herramientas del módulo ClusterLib de la librería de este proyecto.

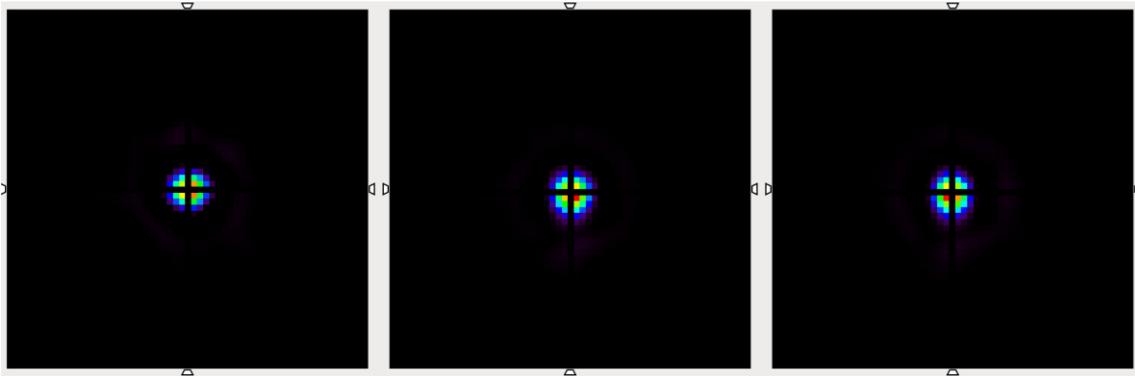


Figura 35 – Imagen PET del fantoma de fuente puntual tras procesar 3 perfiles (uno en cada eje) que atraviesan su centro de masas. Marcado en negro en el interior del objeto, las máscaras de los perfiles. Imagen obtenida mediante las herramientas del módulo ClusterLib de la librería de este proyecto.

LoadData

LoadData es un módulo auxiliar cuyo propósito principal es agilizar las pruebas de la librería, así como ofrecer una primera capa de configuración para probar scripts.

Este módulo ofrece herramientas para leer archivos de configuración que permiten cargar imágenes y sus parámetros (tamaño en milímetros del espacio de la imagen y el tamaño de lado de los vóxeles), definir VOIs y definir perfiles sin tener que introducirlos en forma de código.

```

voisConf.txt: Bloc de notas
Archivo Edición Formato Ver Ayuda
Name Type X Y Z Len Rad/(Base) (Height)
air 1 -9.25 7.75 23.5 7.5 4
water 1 2 -3.75 23.5 7.5 2
back 1 -3.5 1.5 8.5 10 11.25
sqr 2 0 0 0 5 19 4

```

Figura 36 – Imagen de un archivo de configuración de VOIs de ejemplo para el módulo LoadData

PhantomLocalizationLib

Como su nombre indica, PhantomLocalizationLib es un módulo que ofrece herramientas de localización automática de fantasmas en una imagen.

Se apoya en las herramientas del bloque base (VOILib, ProfileLib y ClusterLib) y permite obtener la ubicación y orientación de una serie de fantasmas en el espacio de una imagen.

Además, en el caso del fantoma NEMA IQ, permite localizar también sus distintas regiones.

Este módulo de la librería incluye herramientas para la localización de tres fantasmas:

- **El fantoma de fuente puntual**

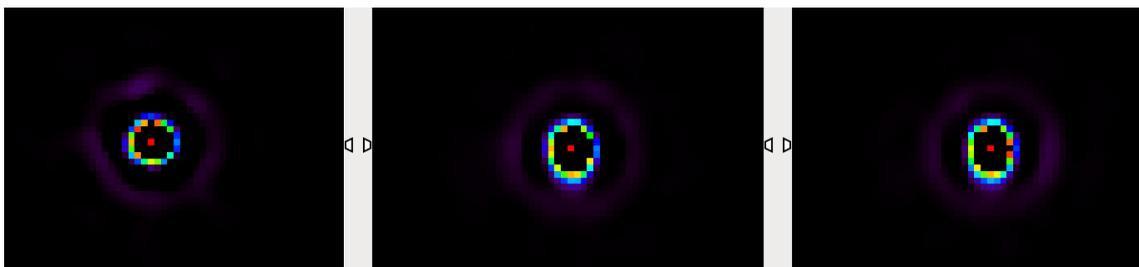


Figura 37 – Imagen PET de la máscara del fantoma de fuente puntual clusterizado por paso alto. Marcado en negro en el interior del anillo, los vóxeles que han superado la condición de umbralización. Marcado en rojo, el centro de masas calculado para el clúster. Imagen obtenida mediante las herramientas del módulo ClusterLib de la librería de este proyecto.

- **El fantoma de uniformidad**

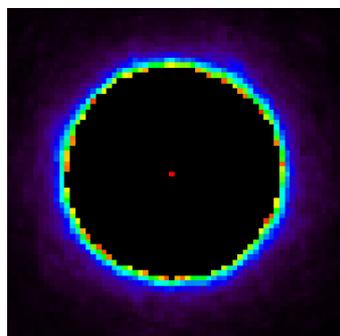


Figura 38 – Imagen PET de un plano transversal de la región de uniformidad del fantoma de uniformidad clusterizado por paso alto. Marcado en negro en el interior del anillo, los vóxeles que han superado la condición de umbralización. Marcado en rojo, el centro de masas calculado para el clúster. Imagen obtenida mediante las herramientas del módulo ClusterLib de la librería de este proyecto.

- **El fantoma NEMA IQ y sus regiones:**

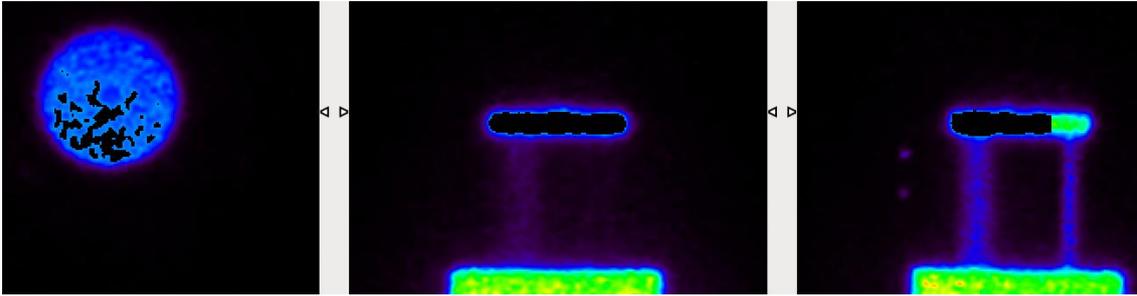


Figura 39 – Imagen PET del fantoma NEMA IQ clusterizado parcialmente por paso alto. Marcado en negro en el interior del objeto, los vóxeles que han superado la condición de umbralización. Esta imagen es un ejemplo de primera aproximación de la ubicación del objeto del método de localización de fantasmas. La clusterización se detiene cuando alcanza un tamaño de clúster determinado, ya que se considera que el objeto ha sido encontrado. Imagen obtenida mediante las herramientas del módulo `PhantomLocalizationLib` de la librería de este proyecto.

- La región de uniformidad

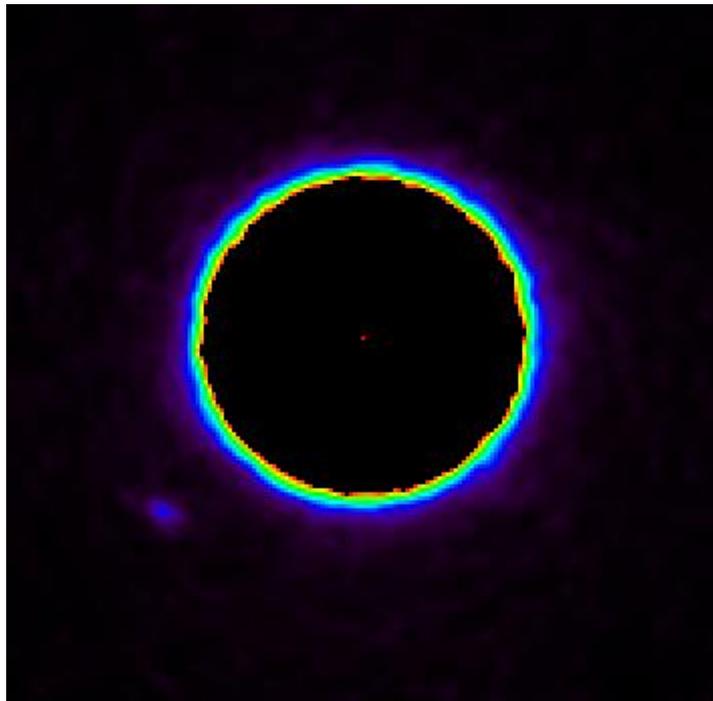


Figura 40 – Imagen PET de un plano transversal de la región de uniformidad del fantoma NEMA IQ clusterizado por paso alto. Marcado en negro en el interior del anillo, los vóxeles que han superado la condición de umbralización. Marcado en rojo, el centro de masas calculado para el clúster. Imagen obtenida mediante las herramientas del módulo `ClusterLib` de la librería de este proyecto.

○ La región de las barras

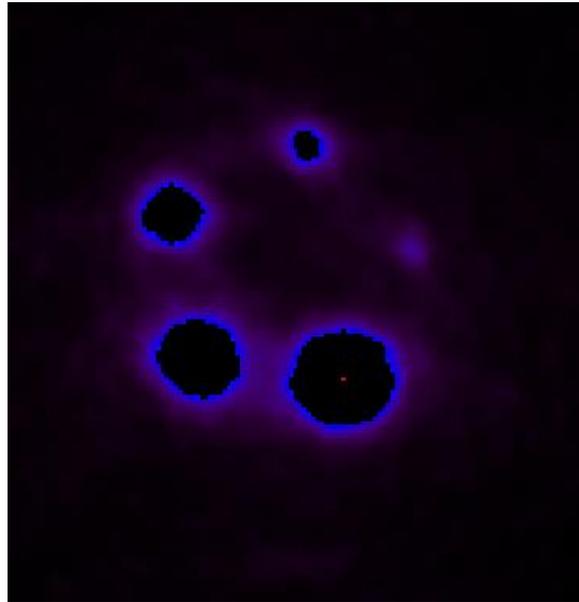


Figura 41 – Imagen PET de un plano transversal de la región de las barras del fantoma NEMA IQ clusterizado por paso alto. Se puede observar que la herramienta ha detectado 4 de las 5 barras.

Marcado en negro en el interior de cada anillo, los vóxeles que han superado la condición de umbralización. Marcado en rojo, el centro de masas calculado para el clúster de la barra de 5mm. Imagen obtenida mediante las herramientas del módulo ClusterLib de la librería de este proyecto.

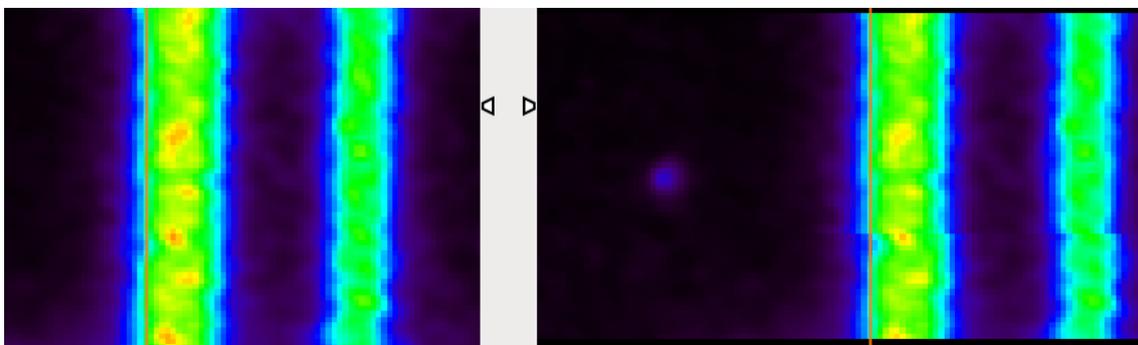


Figura 42 – 2 imágenes PET distintas de un plano coronal de la región de las barras del mismo fantoma NEMA IQ. Se puede observar que la herramienta ha corregido una leve inclinación del objeto desplazando ciertos planos un vóxel a la derecha, obteniendo así mediciones más precisas más adelante.

Marcado en rojo, en el interior de la barra de 5 mm, una línea vertical para ayudar a visualizar el cambio. Imagen obtenida mediante las herramientas del módulo PhantomLocalizationLib de la librería de este proyecto.

- La región del aire y el agua

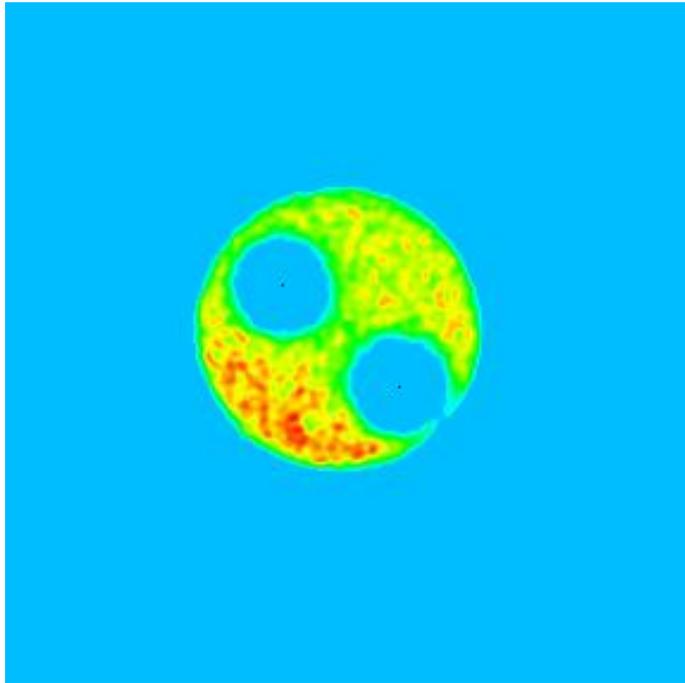


Figura 43 – Imagen PET de un plano transversal de la región del aire y el agua del fantoma NEMA IQ clusterizado por paso bajo. Marcado en azul, los vóxeles que han superado la condición de umbralización. Marcado en negro, los centros de masas calculados para los clústeres del aire y del agua. La herramienta también ha formado un clúster distinto con los vóxeles del fondo. Imagen obtenida mediante las herramientas del módulo PhantomLocalizationLib de la librería de este proyecto.

Validations

El módulo Validations incluye una serie de métodos que permiten realizar las mediciones pertinentes para las distintas figuras de mérito propuestas por el protocolo NEMA.

Además, también ofrece una serie de métodos y una clase de soporte de gestión de resultados, tanto para facilitar la integración de la librería en otros proyectos como para agilizar la creación de scripts de automatización de pruebas de calidad de imagen.

Validations se apoya en los módulos de herramientas básicas para realizar las mediciones, además de necesitar las ubicaciones que el módulo de localización de fantomas provee para ello.

Este módulo ofrece soporte para las figuras:

- **Uniformidad**

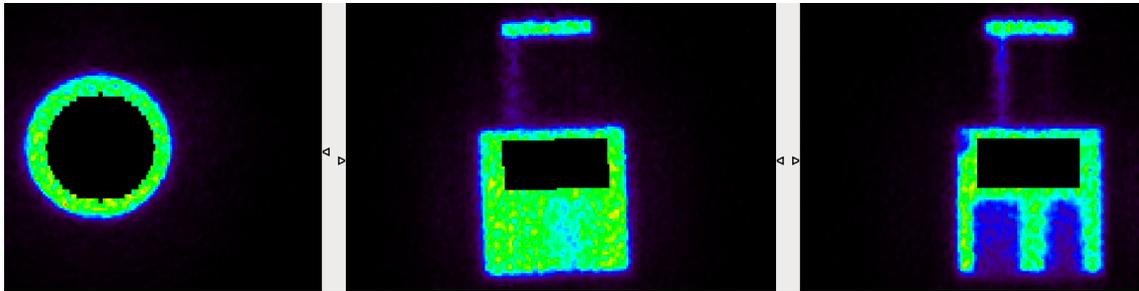


Figura 44 – Imagen PET de la máscara del VOI en la región de uniformidad del fantoma NEMA IQ.
Imagen obtenida mediante las herramientas del módulo VOILib de la librería de este proyecto.

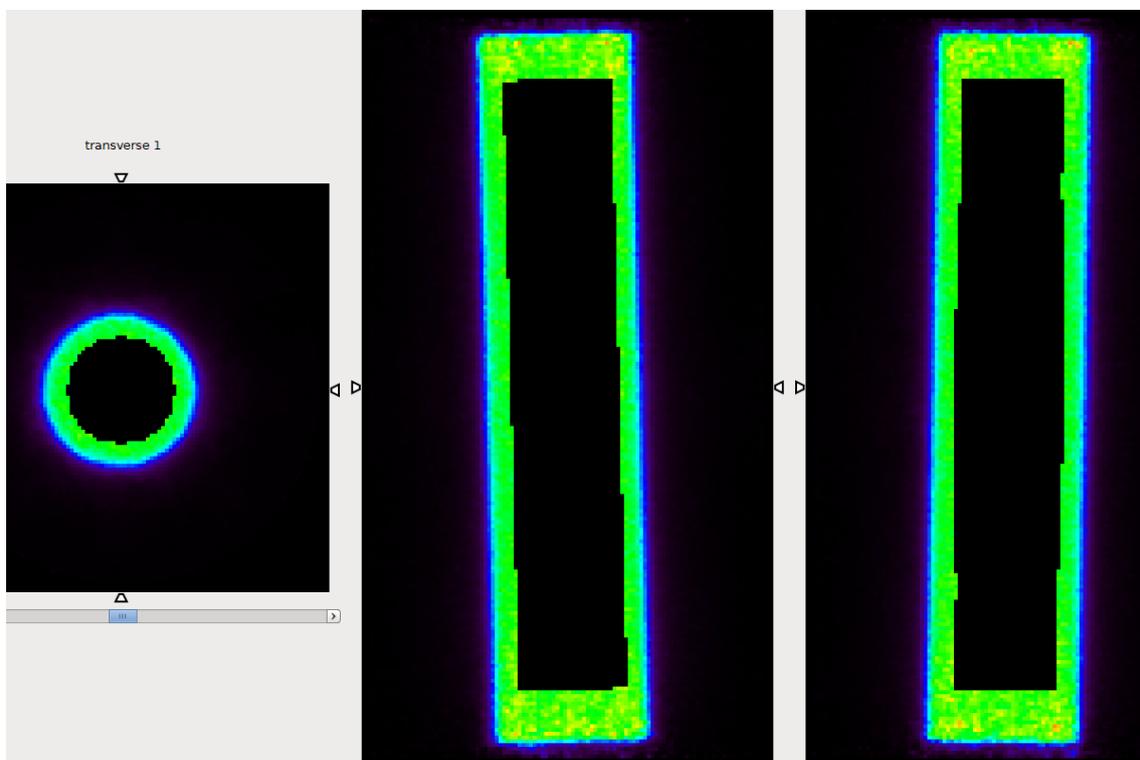


Figura 45 – Imagen PET de la máscara del VOI en la región de uniformidad del fantoma de uniformidad.
Imagen obtenida mediante las herramientas del módulo VOILib de la librería de este proyecto.

- **Resolución espacial (Figuras 8 y 35)**
- **Coefficientes de recuperación (Figuras 12, 13 y 14)**
- **Ratios de desbordamiento**



Figura 46 – Imagen PET de la máscara del VOI en la región de agua del fantoma de uniformidad. Imagen obtenida mediante las herramientas del módulo VOILib de la librería de este proyecto.

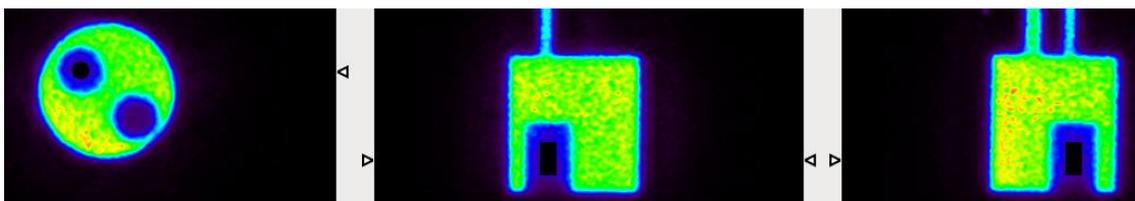


Figura 47 – Imagen PET de la máscara del VOI en la región de aire del fantoma de uniformidad. Imagen obtenida mediante las herramientas del módulo VOILib de la librería de este proyecto.

5.4. Limitaciones de la librería

La librería posee ciertas limitaciones que deben conocerse antes de trabajar con ella. Esto garantizará mayor rendimiento y precisión en los cálculos de los resultados. En algunos casos, la librería podría no llegar a funcionar adecuadamente si no se siguen las advertencias expuestas a continuación.

Respecto al posicionamiento u orientación de los fantomas en la imagen:

- La librería considera que el fantoma NEMA IQ está orientado de tal forma que, si consideramos como perspectiva coronal la figura 9, las barras se colocan en la parte más alejada de la perspectiva transversal del observador. Si esta condición no se cumple, la librería no será capaz de localizar el fantoma NEMA IQ en la imagen.
- La librería es capaz de corregir pequeñas desviaciones en la orientación del objeto de no más de 10° . Superado este punto, algunas correcciones pueden no ser suficientes para recuperar con precisión todos los datos. 10° es un valor máximo pero se recomienda que los objetos estén lo mejor alineados posible en el escáner.

Respecto a la calidad mínima de la imagen:

- No hay datos sobre la calidad de la imagen mínima, ya que no se ha tenido la oportunidad de trabajar con imágenes de calidad demasiado pobre, pero se estima que, imágenes con una calidad extremadamente baja, pueden no ser aptas para el procesamiento en la librería. Esto puede ser de conocimiento relevante si se piensa utilizar la librería para, por ejemplo, el desarrollo de nuevos algoritmos de reconstrucción. Por otro lado, la librería cuenta con mecanismos de detección de puntos calientes (vóxeles con un valor extremadamente alto, producidos por pequeños errores en métodos de reconstrucción iterativos, que pueden alterar el proceso de umbralización automática) y detección de exceso de desbordamiento en las regiones del aire y el agua (cuando el borde entre alguna de las regiones con el exterior del objeto no es percibido por el módulo de clusterización).

6. Validación

La validación del proyecto ha constado de distintas etapas en varios niveles de los módulos de las librerías.

Primero y durante el desarrollo, se realizaron pruebas para cada método de cada módulo por separado, asegurando no solo la eliminación de errores de sintaxis, si no la minimización de cualquier error que pudiera surgir durante la ejecución, comprobando todos los escenarios posibles. Aquellos errores que sobrepasaban la capacidad del código de realizar su función (véase, que la imagen presentaba una calidad demasiado mala para que la herramienta de clusterización fuera consistente o que el objeto estuviera desviado más de 10° en alguno de sus ejes) fue considerado como limitación propia de la librería para tener en cuenta a priori de trabajar con ella.

Una vez un módulo había sido finalizado, se probaba su funcionamiento de una forma más general, analizando su comportamiento como el conjunto de operaciones de los métodos, más que su funcionamiento individual. Observando si los métodos interactuaban con corrección entre sí, especialmente si se requería del uso de métodos de otros módulos.

Durante las anteriores dos fases, también se observaba el rendimiento del código a nivel de uso de capacidad de cómputo y de memoria principal del sistema en que se trabajaba. En más de una ocasión tuvieron que rectificarse y rediseñarse algunos aspectos en beneficio de una mayor eficiencia computacional, especialmente el almacenamiento dinámico de máscaras de la imagen durante la ejecución (la librería NumPy y sus herramientas de *array* brillaron especialmente en este caso).

Con el código probado para manejar cualquier excepción que pudiera darse, se procedió a verificar que no hubiera errores de lógica y que los módulos realizaban las operaciones que se pretendían con exactitud.

Debido a que no existía nada similar para comprobar resultados, se procedió a comparar los resultados de la librería con los obtenidos con el anterior método sobre las mismas imágenes. Se estableció que los resultados no deberían oscilar más de un 2% de los del método original para considerarse satisfactorios.

Además, debido a que el anterior procedimiento no consideraba las rotaciones, se decidió que se realizaran comprobaciones manuales empleando AMIDE. Se reproducían los VOI y los perfiles, empleando las máscaras de estos generadas por la librería, y se comprobaba que los resultados eran exactos hasta, al menos, 4 decimales.

Como paso extra de validación de las herramientas, se generaron imágenes sintéticas de los distintos fantasmas y se les introdujo artificialmente ruido estadístico. Después, se emplearon las herramientas para localizar y obtener estadísticas sobre las imágenes, y se comprobó que estos eran idénticos a los introducidos artificialmente.

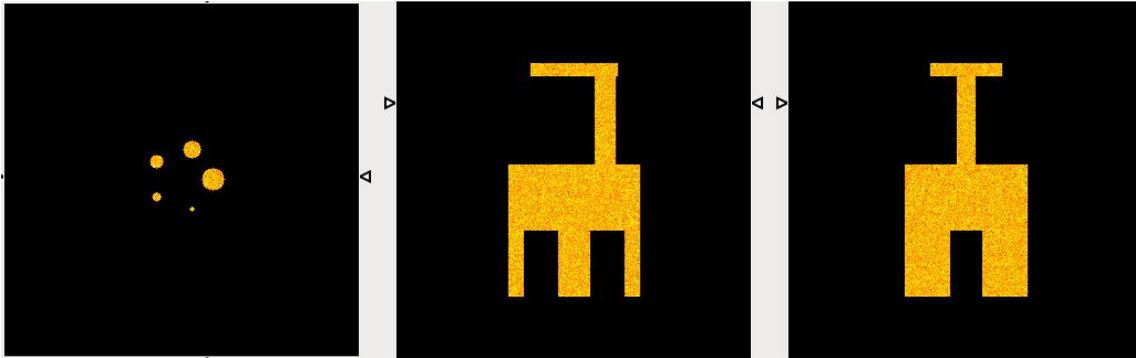


Figura 48 – Imagen generada artificialmente de un fantoma NEMA IQ con ruido estadístico que sigue una distribución normal con $\sigma = 0,1$.

Superadas satisfactoriamente todas estas etapas, se procedió a realizar pruebas de las validaciones a gran escala.

Para las pruebas que requerían emplear los fantomas de NEMA IQ y uniformidad, se emplearon múltiples imágenes obtenidas mediante diferentes métodos de reconstrucción y distintas correcciones.

```

2021-11-10, 09:04:15 - Test Case Evaluation Started
-----
2021-11-10, 09:41:48 - TC23 Done Successfully in [REDACTED] sec.
TC23_1 Reconstruction Done in [REDACTED] sec.
TC23_1 Validation Done
TC23_1 Image Validation Done
Results:
PHANTOM TYPE: IQ
*****
NEMA Validations: SUCCESS
Details:
Uniformity
=====
                %STD,      Mean,      minVal,      maxVal,
-----
MEASURED: [REDACTED]
EXPECTED: [REDACTED]
ERRORS:      0.5 0.003
OBTAINED ERRORS: 0.00395 0.0025
VALIDATION:  True True
RESULT:      True
*****
Recovery Coefficients
=====
Rod size:      5mm,      4mm,      3mm,      2mm,      1mm
-----
%STD: [REDACTED]
MEAN: [REDACTED]
EXPECTED %STD: [REDACTED] | Error: 1.0
EXPECTED Mean: [REDACTED] | Error: 0.03
OBTAINED %STD ERROR: 0.19112 0.19624 0.16267 0.08209 0.43387
OBTAINED MEAN ERROR: -0.00649 -0.00616 -0.0099 -0.01289 -0.0257
VALIDATION:  True True True True True
RESULT:      True
*****
Spill Over Ratio
=====
                Region      SOR      %STD
-----
AIR: [REDACTED]
WATER: [REDACTED]
EXPECTED AIR: [REDACTED]
EXPECTED WATER: [REDACTED]
AIR ERRORS: 0.05 2.0
WATER ERRORS: 0.05 2.0
AIR OBTAINED ERRORS: -0.01238 -0.00187
WATER OBTAINED ERRORS: -0.00588 5e-05
VALIDATION:  AIR-True WATER-True
RESULT:      True

```

Figura 49 – Extracto de un informe de resultados del protocolo NEMA en una imagen de fantoma NEMA IQ. Los valores de estadística han sido ocultados por derechos de propiedad de Bruker. En la imagen se observa el rango de error permitido y el error obtenido por la librería. El informe completo, en este caso, contenía los resultados de la librería en 8 imágenes distintas.

En el caso de la resolución espacial, se realizaron mediciones de las dimensiones en los 3 ejes de un fantoma de fuente puntual en más de 1000 imágenes distintas. Se comprobaba que el resultado no oscilase más de un 25% de su tamaño real (lo que se traducía en una diferencia de 1 vóxel de 0.25 milímetros de lado de tamaño).

520	0.0	-21.0	0.0	1.2157128164356337	1.1065189763807928	1.1849573299791087
521	0.0	-16.0	0.0	1.2906622095460292	1.1024975741027965	1.0738975683206506
522	0.0	-11.0	0.0	1.2194779740255763	1.204736258829003	1.0461809064938805
523	0.0	-6.0	0.0	1.09889838771629	1.0892477478419667	0.9790628680065953
524	0.0	-1.0	0.0	1.0259179041601445	1.0448393668734974	0.9225249366247098
525	0.0	4.0	0.0	1.136830342634649	1.0980706164211922	0.9398171504206984
526	0.0	9.0	0.0	1.1467223734601244	1.155629267496064	0.9494384725063064
527	0.0	14.0	0.0	1.1923477183419777	1.084407632102822	1.008873678475633
528	0.0	19.0	0.0	1.4040440630065913	1.1121997194518629	1.0961661255681376
529	0.0	24.0	0.0	1.5688759889232236	1.0659782702817502	1.1900709336646131
530	-21.0	0.0	0.0	1.0808299629204547	1.2218271705036146	1.1455602202758717
531	-16.0	0.0	0.0	1.0760216224461048	1.2376574731467858	1.0116281879665034
532	-11.0	0.0	0.0	1.1271335457403795	1.1807541811968967	0.9645572481514914
533	-6.0	0.0	0.0	1.1616337578702576	1.1426256575584801	0.980273786625637
534	-1.0	0.0	0.0	1.1069010620831404	1.1685909927901455	0.939817908171385
535	4.0	0.0	0.0	1.3042002135349038	1.112894623549647	0.9567054488312011
536	9.0	0.0	0.0	1.1722706079471172	1.083073274503006	1.0197356386824858
537	14.0	0.0	0.0	1.1602912077018572	1.2415360623128637	0.9636852619628502
538	19.0	0.0	0.0	1.1257920997223607	1.4767540853860588	1.1115000592946005
539	24.0	0.0	0.0	1.1368058016399072	1.354819867707411	1.142133634547968

Figura 50 – Extracto de un informe de resultados de resolución espacial en una imagen de fantoma NEMA IQ. De izquierda a derecha: índice de la imagen analizada, las 3 coordenadas de la posición del fantoma en el espacio de la imagen, las tres dimensiones medidas del fantoma. El informe completo, en este caso, contenía los resultados de la librería en 571 imágenes distintas. Para esta prueba en concreto, los tiempos de adquisición del escáner (tiempo que tarda en realizar la captura) fueron muy cortos, por lo que el rango de error fue ampliado al 40%.

7. Impacto logrado

Finalizado el desarrollo y superadas satisfactoriamente todas las pruebas, la librería fue entregada a la empresa en la que fue desarrollada y donde se realizaron las prácticas laborales, Bruker, en la que actualmente se está empleando en numerosos desarrollos.

En primer lugar y como finalidad original del proyecto, la librería ha sido integrado en la suite de pruebas de imagen PET de Bruker como proceso de validación final de calidad del software.

Esto ha significado la obtención de un método de validación de imagen repetible, reproducible y no supervisado, lo que supone un ahorro de tiempo y esfuerzo.

También empleando la librería, se han desarrollado scripts que realizan controles de calidad rutinarios automáticamente en los equipos de los clientes.

Además, como base de los desarrollos que impliquen generación de imágenes PET, se instauró como protocolo analizar los resultados empleando la librería, de tal forma que, se genera el caso de pruebas una vez termina el desarrollo.

Por último, las herramientas más genéricas de análisis y de localización de fantasmas que provee la librería, están siendo empleadas en varias ramas de desarrollo dentro de Bruker (por ejemplo, un mapa de la resolución espacial en el campo de visión del escáner).

8. Conclusiones

Podemos concluir afirmando que se han alcanzado los objetivos propuestos. La librería provee de las herramientas necesarias para crear fácilmente scripts automáticos de validación de imagen según el protocolo NEMA.

Las herramientas de cálculo incluidas, aunque potentes, no suponen una gran carga computacional en los equipos en los que se ejecutan.

Y, por último, los módulos de herramientas de análisis más genéricos posibilitan la fácil creación y expansión de la misma librería.

Como garantía del éxito de este proyecto, podemos certificar que, a fecha de hoy, la empresa Bruker ha integrado en los equipos de los clientes, y emplea habitualmente en sus desarrollos, la librería de análisis de calidad de imagen PET del proyecto.

Gracias a la realización de este proyecto he podido poner en práctica conocimientos obtenidos durante la carrera de distintas disciplinas. Por ejemplo las metodología agile de Gestión de Proyectos, los sistemas de control de versiones dado con Azure en Ingeniería de Software, o la programación orientada a objetos vista a lo largo de la carrera.

Por otra parte, también tuve la oportunidad de adquirir y ampliar los conocimientos obtenidos en el grado de Ingeniería Informática. Como, por ejemplo, aprender un nuevo lenguaje (Python), a partir de los empleados durante los cursos, y buenas prácticas a la hora de escribir y documentar el código de un proyecto.

Indudablemente, no se trata de un proyecto de gran repercusión, sino que está orientado a un sector muy especializado de la informática médica. Pero, como pude aprender en esa asignatura, son pequeños desarrollos como estos los que posibilitan el avance y la creación de grandes progresos en cualquier campo, especialmente en la investigación médica.

9. Futuro

Estas herramientas son ahora propiedad de la empresa Bruker. Aun así, durante la finalización del proyecto se elaboró una lista de mejoras y posibles expansiones de la librería que podrían ser de interés.

Las principales mejoras consisten en agregar la posibilidad de analizar imágenes con el fantoma NEMA IQ en la orientación opuesta y ampliar las funcionalidades de depuración presentes en el código.

Además sería interesante paralelizar el código existente ya que se obtendría una sustancial mejora en los tiempos de ejecución.

Por último, y dada la naturaleza propia de las librerías, no existe realmente un límite a lo que pueda expandirse el proyecto. Pueden agregarse la localización de nuevos tipos de fantasmas, o la automatización de distintos procesos de análisis más allá del protocolo NEMA.

10. Agradecimientos

En este apartado me gustaría transmitir mi más sincero agradecimiento a todas las partes implicadas en el desarrollo de este proyecto.

A mi tutora, María José Rodríguez Álvarez, por su implicación y su paciencia a pesar de los contratiempos surgidos.

A mi cotutor y mentor en Bruker, Gustavo Pastor Esteve, por sus consejos y su dedicación en el proyecto. Por ayudarme a desenvolverme en un tema tan complejo como es la imagen PET y aportar su perspectiva tanto al código como a la memoria de este proyecto.

A la empresa Bruker, por darme la oportunidad de realizar las prácticas y el trabajo de fin de grado con su equipo en Valencia. Unas brillantes personas de las que he aprendido valores e importantes lecciones de vida, más allá del ámbito de la informática y la empresa, y lo que es formar parte de un equipo multidisciplinar y, a la vez, tan unido y familiar.

Mención especial al Dr. Josep F. Oliver de Bruker, por ayudarme a comprender y documentar en este proyecto el apartado de física involucrado en las imágenes PET.

11. Bibliografía

Protocolo NEMA

NEMA. 29 septiembre 2008. NEMA NU-4 2008: Performance of small animal positron emission tomographs (PETs). EE.UU. [publicación en línea]. Disponible en: <<https://www.nema.org/standards/view/Performance-Measurements-of-Small-Animal-Positron-Emission-Tomographs>> [Consulta: noviembre 2021]

Laura Moliner, Maria J. Rodríguez-Alvarez, Juan V. Catret, Antonio González, Víctor Ilisie, José M. Benloch. 2019. NEMA Performance Evaluation of CareMiBrain dedicated brain PET and Comparison with the wholebody and dedicated brain PET systems. Scientific Reports Nature Research. [publicación electrónica]. [Consulta: noviembre 2021]

Estado del arte

Cherry SR, Gambhir SS. 2001. Use of positron emission tomography in animal research. [publicación en línea]. Disponible en: <<https://www.semanticscholar.org/paper/Use-of-positron-emission-tomography-in-animal-Cherry-Gambhir/9c2cd050387ed354ef6907801aa29308cc737276>> [Consulta: noviembre 2021]

NiftyPET. Fecha desconocida. NiftyPET Documentation. [web en línea]. Disponible en: <<https://niftypet.readthedocs.io/en/latest/intro/>> [Consulta: noviembre 2021]

Markiewicz, P.J., Ehrhardt, M.J., Erlandsson, K. 2018. NiftyPET: a High-throughput Software Platform for High Quantitative Accuracy and Precision PET Imaging and Analysis. [publicación en línea]. Disponible en: <<https://link.springer.com/article/10.1007/s12021-017-9352-y>> [Consulta: noviembre 2021]

Funck Thomas, Larcher Kevin, Toussaint Paule-Joanne, Evans Alan C., Thiel Alexander. 2018. APPIAN: Automated Pipeline for PET Image Analysis. [publicación en línea]. Disponible en: <<https://www.frontiersin.org/articles/10.3389/fminf.2018.00064/full>> [Consulta: noviembre 2021]

APPIAN. Fecha desconocida. APPIAN Documentation. [web en línea]. Disponible en: <<https://github.com/APPIAN-PET/APPIANblob/master/documentation/USERGUIDE.md>> [Consulta: noviembre 2021]

Pmod. Fecha desconocida. Pmod. [web en línea]. Disponible en: <<https://www.pmod.com/web/>> [Consulta: noviembre 2021]

Roger Gunn, Christopher Coello, Graham Searle. Mayo 2016. Molecular Imaging And Kinetic Analysis Toolbox (MIAKAT) - A Quantitative Software Package for the Analysis of PET Neuroimaging Data. [publicación en línea]. Disponible en <https://jnm.snmjournals.org/content/57/supplement_2/1928> [Consulta: noviembre 2021]

M. Savio, Michael Schutte, Manuel Graña and Igor Yakushev. Abril 2017. Pypes: Workflows for Processing Multimodal Neuroimaging Data. [Publicación en línea]. Disponible en: <<https://www.frontiersin.org/articles/10.3389/fninf.2017.00025/full>> [Consulta: noviembre 2021]

Otsu

The lab book pages. Fecha desconocida. Otsu thresholding. [web en línea]. Disponible en: <<http://www.labbookpages.co.uk/software/imgProc/otsuThreshold.html>> [Consulta: noviembre 2021]

Bryan S. Morse. Enero 2000. Lecture 4: Thresholding. [Publicación en línea]. Disponible en: <https://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/MORSE/threshold.pdf> [Consulta: noviembre 2021]

Yen

Carlos A. Cattaneo, Ledda I. Larcher, Ana I. Ruggeri, Andrea C. Herrera, Enrique M. Biasoni. Noviembre 2011. Métodos de umbralización de imágenes digitales basados en entropía de Shannon y otros. [Publicación en línea]. Disponible en: <<https://cimec.org.ar/ojs/index.php/mc/article/viewFile/3951/3868>> [Consulta: noviembre 2021]

12. Documentación de la librería

La documentación de la librería es demasiado extensa para incluirla en este documento.

Puede accederse a ella a través del siguiente enlace:

https://drive.google.com/file/d/13cppoA5Ll6vwzhHs_mVWbmmmeTS1gxw-w/view?usp=sharing