



UNIVERSIDAD DE SEVILLA

FACULTAD DE FÍSICA

GRADO EN FÍSICA

TRABAJO FIN DE GRADO

**Automatización en la detección de patrones
circulares de pequeño tamaño en imágenes
médicas para finalidad de Quality Assurance**

Autor: Rafael Fernández Espinosa

Tutor: Miguel Antonio Cortés Giraldo

Cotutor: José Macías Jaén

A mis tíos Rafa y Marcos.

A mi pequeña Dorami.

Agradecimientos

Quienes me conocen sabrán que el mero hecho de llegar a escribir estas líneas reviste en mi caso una vasta importancia. Sabrán, también, que desde que puse un pie en la Facultad de Física hace casi una década lo hice temeroso pero convencido de que un día podría decir que soy físico y de que mi trabajo tendría una aplicación directa —y positiva— en la sociedad, por pequeña que fuera. Semejante determinación debe cultivarse con templanza y paciencia pero, sobre todo, requiere del apoyo de muchas personas a lo largo de todos estos años. Ahora es mi turno de agradecer todo ese apoyo a esas personas.

En primer lugar, me gustaría dar las gracias a Miguel Antonio Cortés Giraldo y José Macías Jaén por toda su labor y el trato que me han brindado como tutores de este trabajo. No es sencillo encontrar a personas tan dedicadas y comprensivas, y yo he tenido la suerte de encontrar tales cualidades en ambos. Tampoco me olvido de Gerardo Tarrero y Rosa Otero, profesores de Física y Lengua Castellana en Bachillerato, respectivamente. Él plantó la semilla de mi curiosidad y pasión por la Física. Ella consiguió impulsar mi interés por la literatura y supo hacerme ver la importancia de la gramática en la redacción.

Toda mi gratitud a Belén, mi pareja, por la infinita paciencia y comprensión que ha tenido conmigo; no estaría aquí de no ser por su cariño, su profesionalidad y su impecable asesoramiento. También al incansable Nacho Requena, por abrirme a un mundo de escritores y escritoras sin los cuales no hubiese tenido ni la soltura ni las ideas para llevar a cabo la redacción de este trabajo.

Por último, quiero agradecer a mi familia y amigos todo el apoyo y los ánimos a lo largo de la carrera. No habría sido posible sin vosotros.

Llegados a este punto quiero que quienes me conocen ahora sepan que, finalmente, lo conseguí. Que ya estoy aquí. Que al final, perseverando, se llega. Y que mi trabajo ha servido para aportar un pequeño grano de arena en la lucha contra el cáncer.

Resumen

El cáncer sigue permaneciendo como una de las enfermedades más mortíferas a las que se ha enfrentado la humanidad. En su labor por combatir esta enfermedad, los servicios de radiofísica hospitalaria disponen de aceleradores lineales (LINAC) con los que se trata a los pacientes mediante el bombardeo de radiación ionizante, lo cual exige una precisión genuinamente elevada. Por ello, se llevan a cabo controles de calidad como la calibración entre los láseres de la sala, alineados con la herramienta conocida como Iso-Align™, y el punto de radiación del LINAC. Este punto, conocido como isocentro de radiación, consiste en el lugar geométrico donde el acelerador irradia. La necesidad de intervención humana en algunos procesos conlleva un coste de tiempo y una menor precisión que, aunque se encuentren dentro de los niveles de tolerancia exigidos, pueden tener margen de mejora. A fin de asegurar una mayor eficacia resulta imperativo erradicar toda intervención humana en el proceso, para lo cual se ha desarrollado un algoritmo basado en la transformada de Hough en Python con objeto de automatizar la determinación del punto de corte de los láseres para su posterior ajuste con las coordenadas del isocentro. Se ha obtenido una reducción de los tiempos de ejecución del 99 %, mientras que los resultados obtenidos respecto de la metodología original presentan una discrepancia del orden del milímetro o menor, aunque su reproducibilidad no ha resultado total, puesto que se han conseguido procesar con éxito siete de las nueve imágenes del Iso-Align disponibles. A pesar de esto, ha sido posible la implementación de esta metodología en aceleradores lineales, lo que abre las puertas a la completa automatización de los procesos de control de calidad y a una mayor eficiencia en ellos y, por ende, mejora la calidad de vida tanto de pacientes como de radiofísicos.

Palabras clave— Isocentro, Hough transform, Iso-Align, LINAC

Abstract

Cancer still remains one of the deadliest diseases humanity has ever faced. In their battle against it, hospital radiophysics services make use of linear particle accelerators (LINAC) to treat patients by using ionizing radiation. Techniques like this demand a really high precision and, therefore, they implement quality assurance controls such as the calibration of the lasers in the room—which are aligned thanks to the tool known as Iso-Align™—with the point where the LINAC beam hits. This point, known as radiation isocenter, is the geometric location which the accelerator irradiates. The need for human intervention in some processes entails time costs and a lower precision in which, although they are within tolerance limits, there is room for improvement. To ensure a greater efficiency it is imperative to remove all human intervention in the process, for which we developed an algorithm with Python based on Hough's transform with the main goal of automatically locating the point in which lasers intersect, with a view to adjusting it with the isocenter's coordinates. A 99 % time saving has been achieved, while the results obtained for the isocenter differ about one millimetre at most, although its reproductibility is not fully assured, since seven out of nine of the Iso-Align images that were available were successfully processed. Nevertheless, this new methodology has been implemented with success in linear accelerators, which constitutes a milestone towards full automation of quality assurance controls and a better efficiency in them, improving quality of life for patients and radiophysicists alike.

Keywords— Isocenter, Hough transform, Iso-Align, LINAC

Índice general

| | | |
|----------|--|-----------|
| 1 | Introducción y objetivos del trabajo | 1 |
| 1.1 | Introducción | 1 |
| 1.2 | Motivación y objetivos | 5 |
| 2 | Fundamentos teóricos | 6 |
| 2.1 | Conceptos y definiciones previas | 6 |
| 2.1.1 | Acelerador lineal de electrones | 6 |
| 2.1.2 | Isocentros e Iso-Align™ | 9 |
| 2.2 | Transformada de Hough | 10 |
| 2.2.1 | La problemática de la detección de límites en imágenes | 11 |
| 2.2.2 | Funcionamiento de la transformada de Hough | 12 |
| 3 | Metodología | 17 |
| 3.1 | Análisis del isocentro: retrospectiva de la metodología original | 17 |
| 3.1.1 | Trazado de perfiles circulares en una estrella de radiación | 18 |
| 3.1.2 | Centro de referencia: punto de corte de los láseres | 20 |
| 3.2 | Implementación de la transformada de Hough en Python | 22 |
| 3.2.1 | Tratamiento previo de la imagen | 22 |
| 3.2.2 | El método <i>HoughCircles</i> | 24 |
| 3.2.3 | Valor medio y varianza como alternativa a la regresión lineal | 26 |
| 3.2.4 | Representación y punto de corte | 26 |
| 4 | Resultados | 28 |
| 4.1 | Detección de patrones circulares en imágenes de Iso-Align™ y determinación del punto de corte de los láseres | 28 |
| 4.2 | Comprobación de la robustez del código. Desviación respecto de la metodología original | 30 |

| | |
|---|-----------|
| 5 Conclusiones | 35 |
| Bibliografía | 37 |
| A Código empleado en el desarrollo del algoritmo | 39 |
| B Imágenes del Iso-Align TM analizadas | 50 |
| C Resultados del análisis de imágenes de Iso-Align | 55 |

Capítulo 1

Introducción y objetivos del trabajo

1.1 Introducción

El cáncer en general y la carrera por encontrar su cura en particular son dos de las asignaturas pendientes que más urgen aprobar a la sociedad moderna. Si bien el progreso frente a la enfermedad resulta tangible, este es lento y las cifras aún son preocupantes: en torno a 10 millones de personas mueren al año con el cáncer como *mortis causa* [Sung et al., 2021]. Por otra parte, el número de defunciones por cáncer ha aumentado en un 75 % desde 1990, mientras que el índice de mortalidad (n.º de muertes por cada 100 000 personas) ha aumentado un 21 % [Roser, Max, 2020]. Aunque puedan parecer alarmantes, estas cifras son esperables, ya que tanto los cambios demográficos como el avance de la tecnología conllevan una mayor detección de casos y, por desgracia, también un mayor registro de defunciones. No obstante, no todas las conclusiones que se pueden extraer de estos datos son estrictamente negativas; considerando la tasa de mortalidad desglosada por edades se observa que el incremento se debe, principalmente, al envejecimiento de la población. Si se ajustan los datos excluyendo dicho cambio demográfico, la tasa cae un 15 %, puesto que la mayoría de muertes por cáncer se producen en personas de avanzada edad (70 años en adelante), mientras que en la población más joven la cifra se reduce considerablemente, tal y como se puede apreciar en las Figuras 1.1 y 1.2 [Roser, Max, 2020].

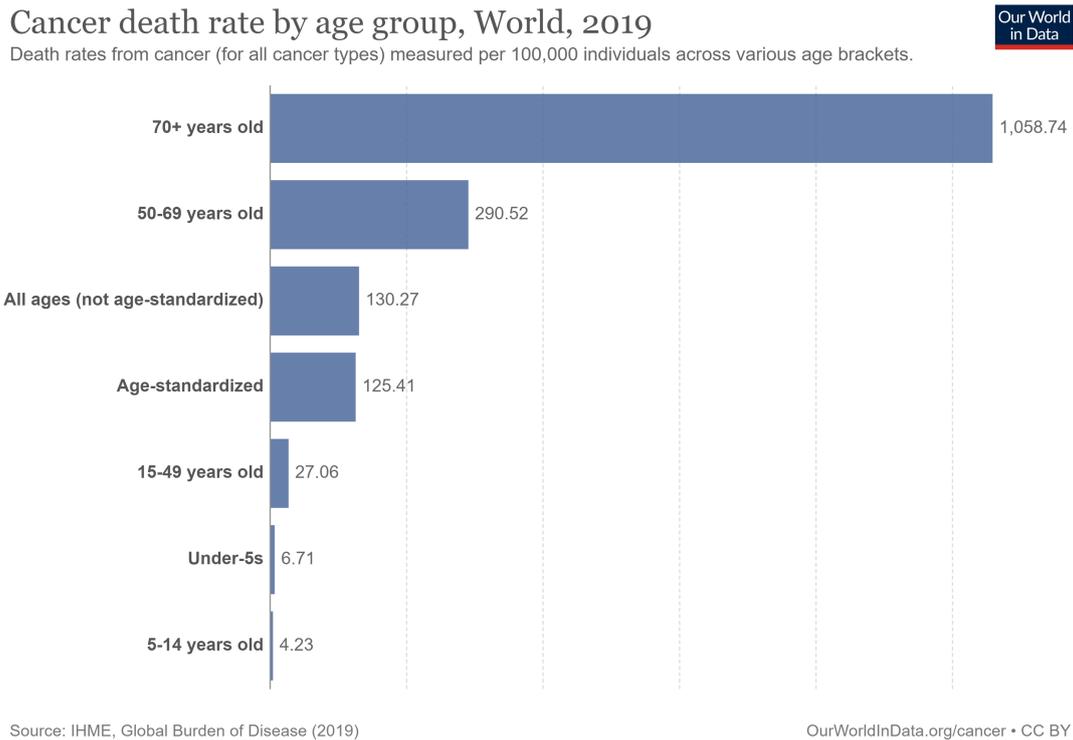
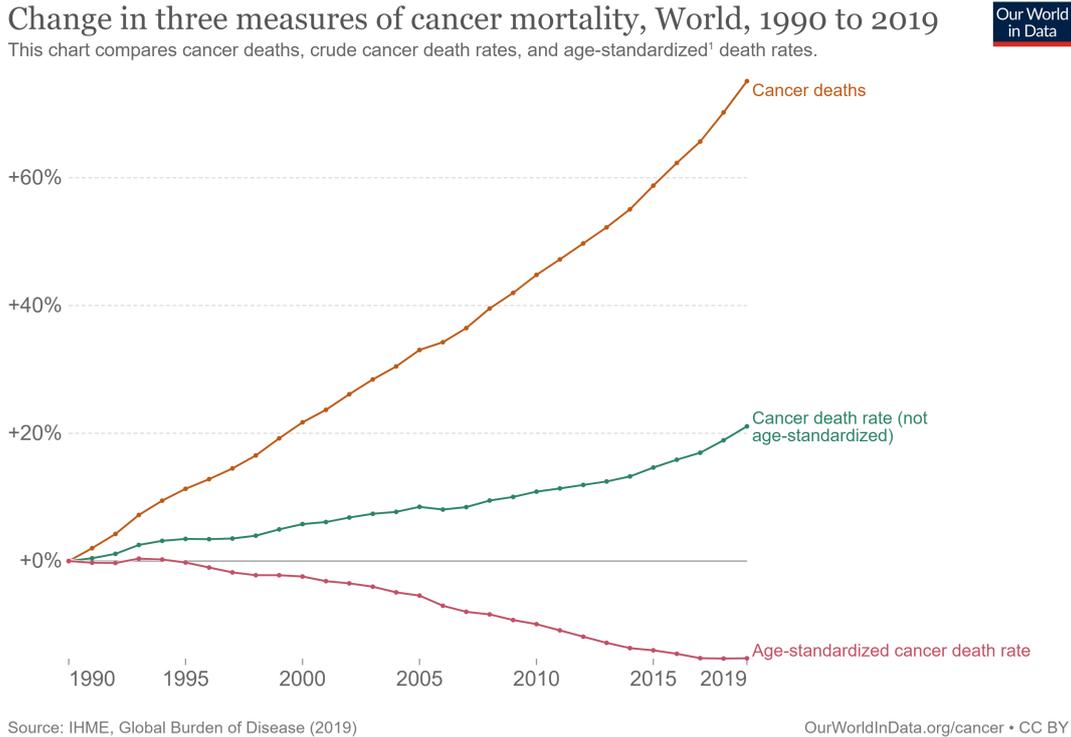


Figura 1.1: Tasa de mortalidad por grupos de edad.[Roser, Max, 2020]

Entre otras muchas disciplinas, la Física ha desempeñado un papel crítico en la medicina, así como en la lucha de la humanidad contra el cáncer. Desde su descubrimiento en 1896 por Henri Becquerel, la radiación ionizante —esto es, que posee la suficiente energía como para interactuar con la materia e ionizarla— se ha estado empleando en el campo de la medicina de formas muy diversas. Quizá el ejemplo más notorio sea el de los rayos X, ahora imprescindibles para realizar ciertos diagnósticos en los hospitales, aunque la lista ha crecido exponencialmente conforme el entendimiento sobre este campo se ha ido incrementando. Los TAC, los PET y técnicas como la protonterapia o radioterapia, donde se tratan tumores mediante el empleo de dicho tipo de radiación, ponen de manifiesto cómo de relevante ha resultado esta disciplina para el bienestar de la población mundial. Es en esta última técnica en la que nos centramos en el presente trabajo.

En palabras del Consejo Nacional de Especialistas en Ciencias de la Salud (CNECS), la radiofísica hospitalaria «es la especialidad sanitaria que se ocupa de medir y valorar las radiaciones, con el fin de contribuir a la correcta planificación, aplicación e investigación de las técnicas radiológicas que la Física pone a disposición de la Medicina»[CNECS, 1996]. Semejante definición resulta más que pertinente para nuestro trabajo, puesto que hace



¹ **Age standardization:** Age standardization is an adjustment that makes it possible to compare populations with different age structures, by standardizing them to a common reference population. [Read more: How does age standardization make health metrics comparable?](#)

Figura 1.2: Evolución anual de la mortalidad del cáncer, en tres tipos de mediciones.[Roser, Max, 2020]

uso del término «planificación». El fenómeno de la radiación puede ser un arma de doble filo, y no habría sido posible llegar a las técnicas que se usan hoy día ni a tales estándares de calidad sin una previa planificación supervisada por un físico, o bien, para ser más exactos, un radiofísico.

En el momento de su descubrimiento, los rayos X se empleaban de forma indiscriminada, véase en fotografías, cosméticos y, por supuesto, en radiografías, sin considerar los efectos que podría tener una exposición prolongada y sin control a la radiación. Las consecuencias fueron, como nos ha enseñado la historia, nefastas. En España, tras el accidente en el Hospital Clínico Universitario Lozano Bleza (Zaragoza, año 1990) [Argos, Lucía, 1991] en el que murieron cerca de una treintena de personas por una calibración errónea del acelerador lineal, se decretó la necesidad de la presencia de físicos especializados en radiación ionizante en todos los hospitales, ejerciendo de puente entre la Medicina y la Física y velando por que todos los aparatos estén correctamente calibrados. Se había creado la especialización de la radiofísica hospitalaria.

Entre otras muchas tareas, el papel del radiofísico hospitalario consiste, esencialmente, en asegurarse de que el paciente reciba la dosis correcta de radiación de manera exacta en el lugar necesario. Para ello, actualmente se hace uso de los aceleradores lineales (LINAC) de electrones, dispositivos que emiten un bombardeo de radiación y pueden adaptarse a la forma concreta del tumor, minimizando así los daños por quemaduras y exposición a la radiación en el tejido sano. Ahora bien, la herramienta no lo es todo; también debe usarse correctamente. Para hacer un uso adecuado y responsable de los aceleradores lineales se necesita un estricto y riguroso plan de control de calidad (en lo sucesivo «QA», de «*quality assurance*» en inglés).

En el Real Decreto 1566/1998, publicado en el Boletín Oficial del Estado (BOE) de 17 de julio, se establecen los criterios de calidad a los que se deberá ceñir escrupulosamente cualquier entidad u organización que se disponga a realizar operaciones de carácter radioterapéutico. En este se impone la obligatoriedad de introducir programas de QA sobre los LINAC, detallando una serie de parámetros de seguridad, niveles de acción —como referencia para detectar anomalías y actuar en consecuencia— y tolerancias varias con respecto a distintos aspectos de los mismos, con el objeto de conseguir garantizar el adecuado funcionamiento de las instalaciones que se haga uso [BOE, 1998]. Cómo se diseñen y se lleven a cabo dichos programas recae directamente en el servicio de radiofísica hospitalaria, y no sale adelante sin el visto bueno del radiofísico encargado de la supervisión de las instalaciones.

Aunque cada programa de QA sea único, podemos dividir las pruebas que se realizan en dos grupos: dosimétricas y geométricas. Las primeras tienen como objetivo que la dosis administrada a los pacientes sea exactamente la que el equipo desee administrar, mientras que las segundas velan porque el bombardeo de partículas ionizantes de exactamente donde se indica. A lo largo del trabajo nos centraremos en estas últimas, donde trataremos de automatizar una parte del procedimiento de hallar el isocentro de radiación —este es, el punto exacto donde incidirá la radiación proveniente del LINAC—, con el objetivo de eliminar la componente humana y así agilizar el proceso de QA.

En el trabajo se hará uso de la transformada de Hough, algoritmo diseñado para reconocer patrones en imágenes, con objeto de detectar los puntos de calibración del maniquí que se ha usado para calibrar los láseres de la habitación donde se encuentra el

LINAC. Estos láseres intersectan en un punto que, idealmente, debe coincidir con el punto donde el rayo del acelerador incide, conocido como isocentro de radiación. Finalmente, una vez se haya conseguido automatizar el proceso se someterá al programa a reiteradas pruebas para comprobar que tanto el tiempo de computación como la precisión se han mejorado respecto de la metodología original.

1.2 Motivación y objetivos

Tal y como señalábamos en la introducción, el cáncer sigue siendo una de las principales causas de muerte en la sociedad actual. Aunque la tecnología y nuestro conocimiento acerca de la enfermedad haya avanzado queda mucho por hacer y, a pesar de sus redoblados esfuerzos, los recursos y el tiempo de las divisiones de radiofísica hospitalaria son limitados.

Este trabajo surge de la necesidad de agilizar los procedimientos de QA en los hospitales para conseguir una puesta en marcha más rápida de los aceleradores lineales, así como de eliminar la componente humana en ciertos procedimientos que requieren una mayor precisión, lo que implica que el personal radiofísico pueda encargarse de otras tareas más urgentes o bien reducir el tiempo necesario para realizar tales procedimientos. Todo con el objetivo principal de alcanzar una puesta a punto de los aceleradores más eficaz para interrumpir lo menos posible el tratamiento de los pacientes oncológicos.

Considerando todo lo relatado anteriormente, el presente trabajo tiene dos objetivos:

- desarrollo de un algoritmo que automatice la detección de patrones circulares en las imágenes del maniquí de calibración usadas para hallar el punto de corte de los láseres en proceso de calibración geométrica;
- comprobación de la calidad y robustez del algoritmo comparando el punto de corte hallado por este y el determinado mediante el uso de la metodología tradicional, y sometiéndolo a distintas imágenes maniquí.

Capítulo 2

Fundamentos teóricos

Antes de sumergirnos de lleno en el trabajo, resulta necesario profundizar en algunos de los conceptos y dispositivos que hemos presentado en los puntos anteriores, con objeto de asegurar que el lector se encuentre en el punto de partida óptimo para comprender lo que se discute en las siguientes páginas.

2.1 Conceptos y definiciones previas

2.1.1 Acelerador lineal de electrones

Los aceleradores lineales (LINAC) son, en esencia, dispositivos cuyo trabajo consiste en acelerar partículas cargadas eléctricamente. Reciben el adjetivo «lineal» debido a que la partícula acelerada en cuestión transita una única vez a lo largo de una trayectoria rectilínea, al contrario de lo que sucede con los aceleradores tipo ciclotrón o sincrotrón [Jameson et al., 2015].

Originalmente, un acelerador lineal estaba compuesto por una serie de cilindros metálicos aislados conocidos como tubos de deriva, con un campo eléctrico oscilante aplicado en el espacio entre estos encargado de acelerar las partículas cargadas (véase el esquemático de la Figura 2.1). Estas partículas, sometidas a un potencial variable del orden de la radiofrecuencia entre cada pareja de cilindros, irán experimentando un aumento de velocidad a lo largo del eje de movimiento, la cual permanecerá constante en el momento en que atraviesan alguno de los tubos aislados [Jameson et al., 2015].

Debido a este aumento progresivo de la velocidad surge la necesidad de ir aumentando la longitud de los tubos de deriva, puesto que se debe mantener la sincronización del paso de la partícula entre el espaciado con el periodo de variación del potencial eléctrico para que las partículas viajeras sigan experimentando una aceleración.

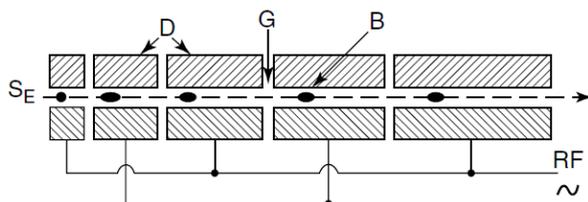


Figura 2.1: Esquema de un acelerador lineal ideado por R. Wideroe. S es una fuente de iones positivos. D representa un esquema de los tubos de deriva. G consiste en el espaciado entre tubos. B consiste en el haz de iones. RF señala que la frecuencia del pulso de potencial es del orden de la radiofrecuencia. [Jameson et al., 2015]

En este trabajo nos centraremos en los aceleradores lineales de electrones, cuya velocidad se acerca rápidamente a la de la luz. Debido a esto, el planteamiento anterior no funciona, puesto que se necesitarían frecuencias sumamente elevadas con objeto de conseguir la sincronización deseada con el paso de los electrones. Para remediar esto se diseñó una guía de ondas —esto es, una cavidad metálica hueca que permite la transmisión de ondas electromagnéticas confinadas en ella— que recibe el nombre de guía de ondas de iris (*iris-loaded waveguide*), por la cadena de coronas circulares dispuestas paralelamente en el interior de la cavidad cilíndrica (Figura 2.2). La cavidad central de la cadena, el iris, ejerce la función de filtro para radiofrecuencia con un ancho de banda relativamente estrecho [Jameson et al., 2015]. Esto implica que la relación de fase entre dos cavidades consecutivas dependa exclusivamente de la frecuencia, lo que significa que dada una cierta frecuencia podemos disponer las dimensiones oportunas para que la onda electromagnética viaje con una velocidad de fase igual a la de los electrones eyectados. Esto asegura la sincronización y, por ende, la aceleración de las partículas a lo largo de la guía. Cabe destacar que, para evitar cualquier tipo de desaceleración de los electrones o pérdida de energía, es conveniente alcanzar un buen vacío en la guía de ondas.

En el caso de los LINAC de electrones usados en radiofísica hospitalaria, encontramos un esquemático común a todos los distintos tipos en la Figura 2.3. Siguiendo las ideas expuestas anteriormente, el funcionamiento, a grandes rasgos, es el que sigue:

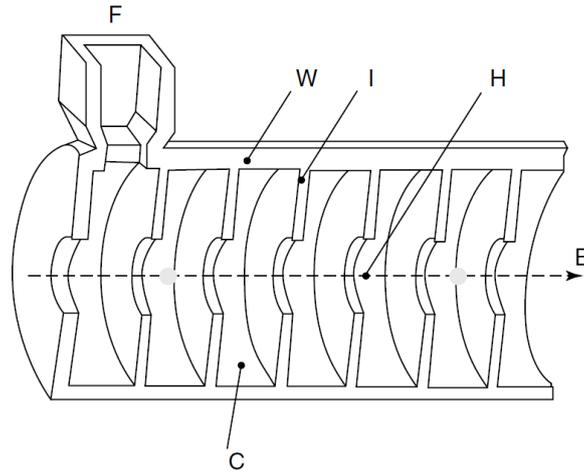


Figura 2.2: Esquema de la guía de ondas de iris, donde I son las coronas circulares; H los huecos circulares en dichas coronas (iris); W la guía de ondas; C las distintas celdas; F la fuente de energía y B el eje por el que se transmiten las ondas electromagnéticas. [Jameson et al., 2015]

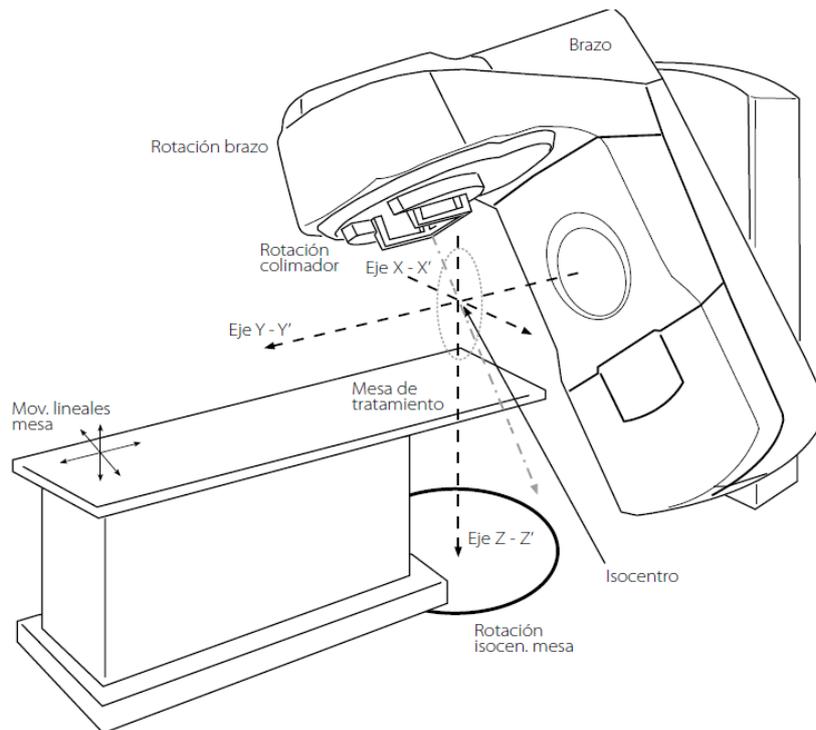


Figura 2.3: Diagrama de un acelerador lineal de electrones empleado en radiofísica hospitalaria.[Pinza-Molina and Lliso-Valverde, 2009]

Partiendo de un filamento de tungsteno se genera un haz de electrones mediante emisión termoiónica que pasa a la guía de onda, donde se acelera hasta alcanzar la velocidad deseada. Mediante el empleo de imanes cuadrupolares se ayuda a la concentración de electrones del haz [Jameson et al., 2015], que se redirige a través del brazo (*gantry*) hasta otra muestra de tungsteno —conocida como filtro—, donde los electrones inciden y

se emite radiación en forma de fotones por bremsstrahlung, esto es, radiación resultante de la deceleración de portadores de carga, como son los electrones. Dichos fotones salen disparados hacia todas direcciones y en un espectro energético diverso, de modo que se precisa de una serie de colimadores y cámaras de ionización al objeto de focalizar y mejorar la calidad del haz que finalmente incidirá en las masas tumorales de los pacientes. Con el haz listo solo falta indicar la forma óptima de este para abarcar la mayor superficie posible de masa tumoral y la menor de tejido sano. Para esto se hace uso de los colimadores multilamina (MLC, del inglés «*multi-leaf collimator*»), que consisten en una serie de hojas de un cierto metal, habitualmente tungsteno, situadas paralelamente las unas a las otras y que pueden desplazarse a voluntad y de forma independiente, otorgando el perfil deseado para el rayo fotónico emergente (Figura 2.4).

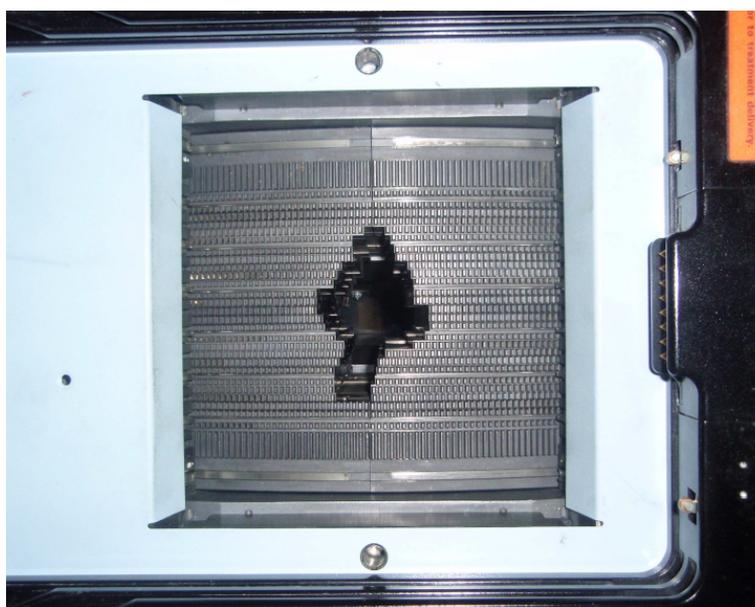


Figura 2.4: Ejemplo de colimador multilamina. [Cortés Giraldo, 2011]

2.1.2 Isocentros e Iso-Align™

Son muchas y muy variadas las maneras de definir el isocentro. En lo que a nosotros nos atañe —es decir, en radiofísica hospitalaria— se puede clasificar el isocentro en dos tipos: mecánico y de radiación. A saber:

- **Isocentro mecánico:** centro del círculo que mejor ajusta la trayectoria dibujada desde la posición de fuente cero (posición en la que colimador y *gantry* tengan

ángulo de 0°). Consiste en el punto en el espacio donde *gantry*, colimador y mesa rotan [Zhang et al., 2015].

- **Isocentro de radiación:** para un ángulo del *gantry* dado, el isocentro de radiación se encuentra en el centro de la esfera de menor tamaño que atraviese la línea emergente del *gantry* [Zhang et al., 2015]. Es el punto en el espacio donde intersectan los rayos de radiación central, y viene dado por el círculo de menor radio que cumpla con las propiedades de intersectar con todas las líneas de los rayos de radiación y sea tangente a cada línea dentro de una terna.

A lo largo del documento nos centraremos en el isocentro de radiación, cuya calibración comienza con el alineado de láseres dispuestos en la habitación donde se encuentra el LINAC. Estos láseres emiten planos cuyo corte fija, a priori, el centro de referencia de radiación para la máquina. Para afinar la calibración y verificar que se irradia en el punto deseado, se hace uso de maniqués dados por láminas de Iso-Align™, una herramienta multitarea empleada para QA y calibración de los LINAC (Figura 2.5).



Figura 2.5: Ejemplo de Iso-Align. (Civco Radiotherapy <https://civcort.com/ro/physics-qa/isoalign/isoalign-MTIAD1.htm>. Accedido en agosto de 2023.)

2.2 Transformada de Hough

Patentada por Paul V. C. Hough en 1962 [Hough, 1962] y originalmente diseñada para utilizarse en un detector de partículas, la transformada de Hough consiste, en líneas generales, en una técnica de extracción de elementos, i.e. una técnica que se encarga de transformar datos aún sin procesar en datos numéricos que puedan ser procesados

manteniendo la información del conjunto de datos original. El objetivo principal de la transformada consiste en encontrar determinados patrones en imágenes.

Entre sus muchas aplicaciones, la transformada de Hough se usa principalmente para tratamiento de imágenes, aunque estas pueden ser tantas como uno se proponga. En el presente trabajo se usará para la detección de los patrones circulares de la imagen de Iso-Align, de modo que es preciso profundizar en cómo funciona.

2.2.1 La problemática de la detección de límites en imágenes

Según Ballard, en una imagen la información más relevante sobre un objeto suele venir prescrita en la forma de su contorno [Ballard, 1981]. Ahora bien, el principal problema que surge a la hora de buscar estas formas consiste en conseguir distinguir el contorno de la figura en concreto que estamos buscando. Por lo general, en aquellas imágenes donde queremos extraer la información de una figura determinada habrá otro tipo de elementos cuyas figuras se solaparán o mezclarán con las nuestras, dificultando el proceso de extracción de datos que buscamos. Para ilustrar mejor esto, usaremos como ejemplo una imagen con una variedad de patrones bien definida, como puede ser la imagen de la Figura 2.6.

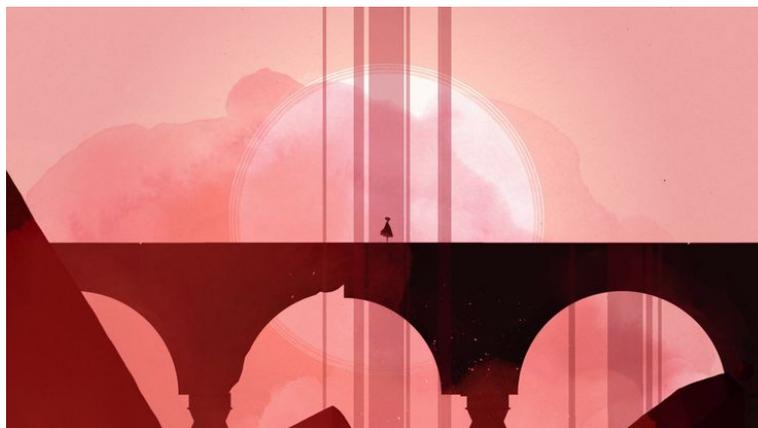


Figura 2.6: Ejemplo de imagen con figuras muy marcadas, como pueden ser el puente o las cascadas (GRIS, Nomada Studio).

En ella podemos apreciar diversos patrones: rectilíneos, circulares e incluso irregulares, como las nubes o el polvo. Supongamos que nos interesa extraer exclusivamente la información relativa a los primeros. Para ello, aplicamos un pequeño tratamiento a la imagen que muestre solo los bordes más marcados (Figura 2.7). Aquí se manifiesta el

problema que anunciábamos al inicio del punto: si bien es cierto que se ha conseguido realizar una buena criba de formas indeseadas, sigue resultando complicado indicar a nuestro ordenador cuáles son las figuras cuya información necesitamos y cuáles debe descartar porque no son necesarias. En este momento es en el que entra la transformada de Hough, que plantea una potente solución a este tipo de dificultades y consigue distinguir patrones rectilíneos o circulares empleando un número de parámetros muy reducido. Cómo lo hace se describe a continuación.

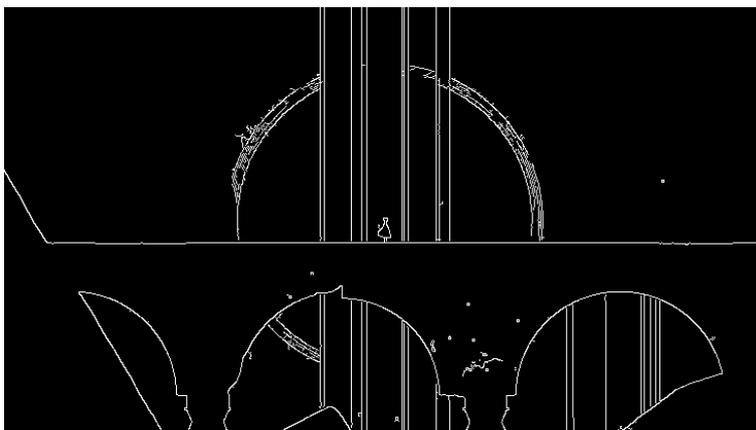


Figura 2.7: Imagen ejemplo con los bordes más relevantes señalados (GRIS, Nomada Studio).

2.2.2 Funcionamiento de la transformada de Hough

En nuestro caso, como el uso de la transformada radica únicamente en encontrar patrones simples, estudiaremos la transformada de Hough empleada para curvas analíticas con la forma $f(x, \mathbf{a}) = 0$, esto es, curvas que pueden ser representadas por una expresión analítica, donde x es un punto de la imagen y \mathbf{a} un vector paramétrico [Ballard, 1981]. Llevaremos a cabo dicho análisis con el ejemplo más simple: la detección de líneas.

Supongamos que tenemos una nube de puntos de borde esto es, puntos cuyos *píxeles* han superado cierto umbral como para ser seleccionados como borde de una figura y queremos detectar la línea recta que dibujan algunos de ellos (Figura 2.8). Sabemos que la ecuación de una recta de pendiente m y ordenada en el origen n viene dada por

$$y = mx + n. \quad (2.1)$$

En dicha ecuación x e y son variables, mientras que m y n son parámetros fijos. La manera en la que la transformada de Hough actúa es invirtiendo estos roles, es decir, dejando que los parámetros m y n sean variables para cada punto de la recta y fijando los valores de x e y en cada caso. La ecuación de la recta pasaría a ser entonces

$$n = y_i - mx_i, \quad (2.2)$$

donde el subíndice i hace referencia a cada punto de borde empleado para dibujar la recta.

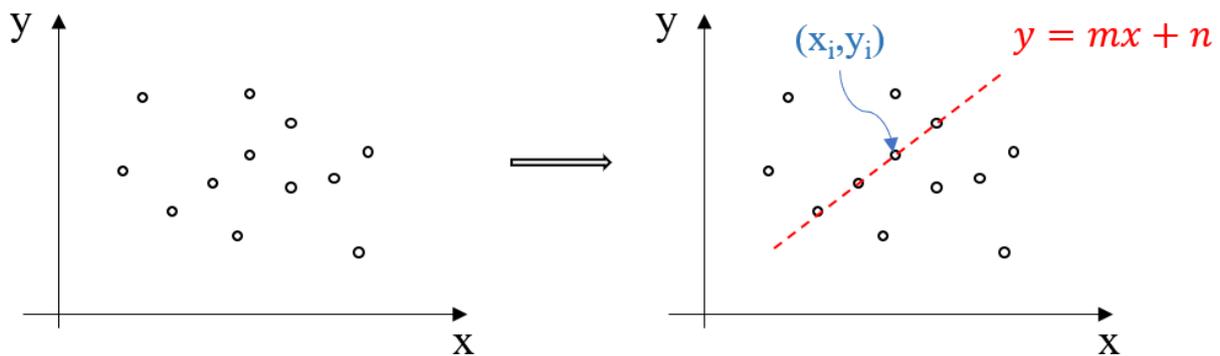


Figura 2.8: Nube de puntos de borde con patrón rectilíneo oculto

En este punto y como con todo tipo de transformadas tendremos que distinguir entre dos espacios de trabajo. A saber:

- **Espacio imagen:** espacio original, donde se encuentra la imagen y cuyos puntos corresponden a coordenadas en esta.
- **Espacio de parámetros:** espacio lugar de la transformada de Hough, cuyos puntos indican la presencia de patrones figurativos.

Aplicando la transformada de Hough a los puntos de la nube pasamos entonces del espacio imagen al espacio de parámetros, donde se dibujará una recta para cada par de valores (x_i, y_i) que cumplan con la ecuación (2.2). El resultado que cabría esperar consiste en una serie de líneas con distinta dirección pero con una característica en común: todas intersectan en el punto (m, n) en el espacio de parámetros. Por contra, aquellos puntos que no formen parte de la recta en el espacio imagen no pasarán por dicho punto en el espacio de parámetros. Podemos apreciar esto en la Figura 2.9. Generalizando para un

conjunto de rectas en el espacio imagen obtendremos un conjunto de puntos asociados a dichas rectas en el espacio de parámetros.

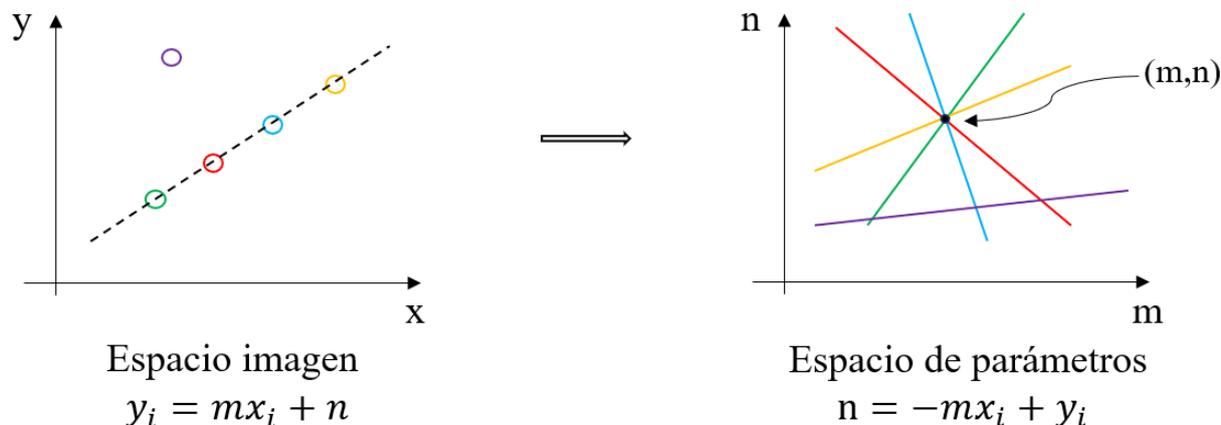


Figura 2.9: Transformada de Hough aplicada en los puntos de la nube.

Por el momento, siguiendo este procedimiento hemos conseguido distinguir aquellos puntos de borde del resto, quedándonos exclusivamente con los que dibujan una línea recta. No obstante, existen maneras más eficaces de aplicar la transformada si encontramos la parametrización adecuada. Por ejemplo, para el caso de las rectas resulta más indicado hacer uso de la parametrización en coordenadas polares, con ρ y θ como parámetros [Duda and Hart, 1972]. Definimos entonces la ecuación de la recta como

$$x \cos \theta + y \sin \theta = \rho. \quad (2.3)$$

Aplicando la transformada, x e y permanecerán constantes, mientras que ρ y θ serán variables. Como las funciones son sinusoidales, podremos distinguir como rectas en el espacio imagen aquellos puntos en el espacio de parámetros en los que intersectan todas las funciones sinusoidales correspondientes a los puntos de la recta original (véase la Figura 2.10).

No obstante, independientemente de la parametrización escogida existe la posibilidad de afinar aún más la detección de bordes y contornos con este procedimiento. Aunque el algoritmo original ideado por Hough no hacía uso de ello, en los procedimientos actuales se hace uso de la información provista por los bordes para señalar una orientación en la figura detectada, esto es, indicar la dirección hacia la cual se «cierra» la figura considerando la información del gradiente de la imagen en escala de grises. Esta técnica es especialmente

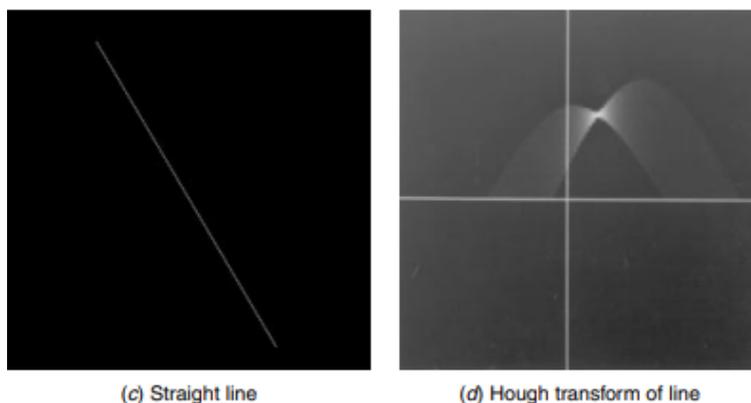


Figura 2.10: Transformada de Hough aplicada a una recta usando parametrización polar.[Pratt, 2007]

eficaz para casos como la detección de patrones elípticos o circulares y será la llevada a cabo en el trabajo, puesto que consigue reducir aún más el número de parámetros.

La manera de llevar todo este razonamiento al terreno de la computación consiste en emplear lo que se conoce como matriz de acumulación, $A(\mathbf{a})$. Esta matriz determinará si los puntos detectados forman la figura deseada o no siguiendo un procedimiento de votación basado en las ideas expuestas en el ejemplo anterior, donde nuestro vector de parámetros es $\mathbf{a} = (m, n)$, por lo que la matriz de acumulación podremos definirla como $A(\mathbf{a}) = A(m, n)$.

Inicialmente, cuantizamos el espacio de parámetros subdividiéndolo en secciones cuadrangulares del tamaño que más convenga (*píxeles* en nuestro caso). Cada recuadro en dicha división será un elemento de la matriz de acumulación, todos con un valor inicial impuesto de cero. Por cada punto de borde detectado —recordemos, pares de (x_i, y_i) que cumplan con la ecuación (2.2)— se suma un voto en el elemento correspondiente de la matriz, de modo que en el punto del espacio de parámetros cuyo elemento de la matriz de acumulación sea más elevado se corresponderá con el punto donde convergen todas las líneas asociadas a puntos de borde en el espacio imagen (Figura 2.11).

Conociendo cómo funciona, si aplicamos un algoritmo basado en la transformada de Hough para patrones rectilíneos al ejemplo de la Figura 2.6 acabamos obteniendo el resultado deseado, donde se consigue la distinción de dichos patrones del resto, dando pie a poder indicar las figuras que se precisen de la imagen (Figura 2.12).

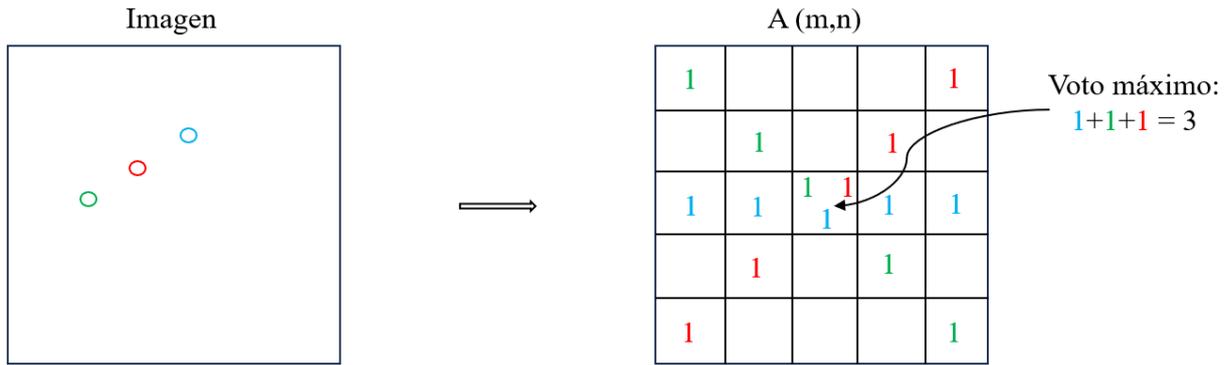


Figura 2.11: Ejemplo visual del funcionamiento de la matriz de acumulación.

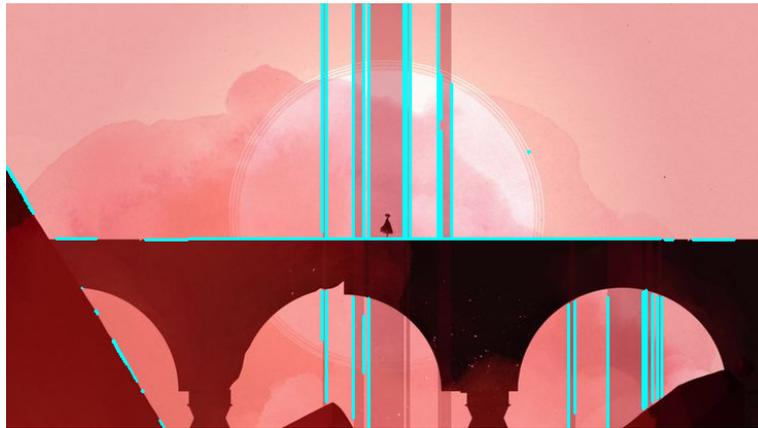


Figura 2.12: Ejemplo de imagen donde se han conseguido distinguir los patrones rectilíneos del resto (GRIS, Nomada Studio).

Como veníamos anunciando, la transformada de Hough tiene tantas aplicaciones como puedan imaginarse, y la detección de figuras puede extrapolarse a un método de detección más general. En nuestro caso haremos uso del algoritmo basado en la detección de patrones circulares, cuya función analítica no es otra que la ecuación del círculo:

$$(x - a)^2 + (y - b)^2 = r^2, \quad (2.4)$$

mientras que el procedimiento es el mismo que el descrito en el ejemplo para líneas rectas.

Capítulo 3

Metodología

3.1 Análisis del isocentro: retrospectiva de la metodología original

Una vez conocida la utilidad de la transformada de Hough, el principal objetivo de este trabajo consiste en hacer uso de ella para patrones circulares con el objeto de automatizar la detección de dichos patrones en las imágenes de Iso-Align y así agilizar el calibrado del LINAC. Como ya hemos comentado, cada servicio de radiofísica cuenta con su propio procedimiento a la hora de localizar el isocentro. En el caso del Hospital Universitario Virgen Macarena (en lo sucesivo «HUVM»), ubicado en Sevilla (España), se hace uso de un *software* desarrollado por la unidad de gestión de radioterapia oncológica y radiofísica hospitalaria, cuyo fin último consiste en determinar el isocentro de radiación y alinear los láseres con este.

El funcionamiento del programa informático divide la búsqueda del isocentro en dos bloques: trazado de perfiles circulares en una estrella de radiación y búsqueda del centro de referencia de radiación hallando el punto de corte de los láseres alineados con ayuda del Iso-Align. Aunque el presente trabajo se centrará en este último bloque, es necesario describir el funcionamiento del programa original en su totalidad para tener el contexto completo de su funcionamiento.

3.1.1 Trazado de perfiles circulares en una estrella de radiación

Una de las pruebas de control de calidad en los LINAC consiste en verificar que el sistema MLC se encuentre centrado respecto al eje de rotación del colimador. Para ello, se hace uso de una película radiocrómica que se dispone de forma perpendicular al eje de radiación. Con las láminas del MLC definiendo un segmento rectangular se irradia a distintos ángulos de rotación del colimador, dibujando una estrella de radiación en la película (Figura 3.1) [Pinza-Molina and Lliso-Valverde, 2009]. En el centro de esta estrella, una vez se encuentra bien calibrada, se halla el isocentro de radiación del LINAC.



Figura 3.1: Estrella de radiación usada para verificar el sistema MLC del acelerador lineal. (Cortesía del Servicio de Radiofísica del HUVM)

Para fijar el isocentro de radiación, el *software* trazará unos perfiles circulares alrededor de un punto que el usuario establece a su elección. Estos perfiles se conocen como perfiles de intensidad, cuya utilidad no resulta de especial relevancia en nuestro caso. La manera de indicar el punto a partir del cual se dibujan dichos perfiles es a mano, introduciendo la imagen de la estrella de radiación en el programa y, seleccionando la herramienta pertinente, haciendo clic en el punto de intersección de todos los haces de la estrella.

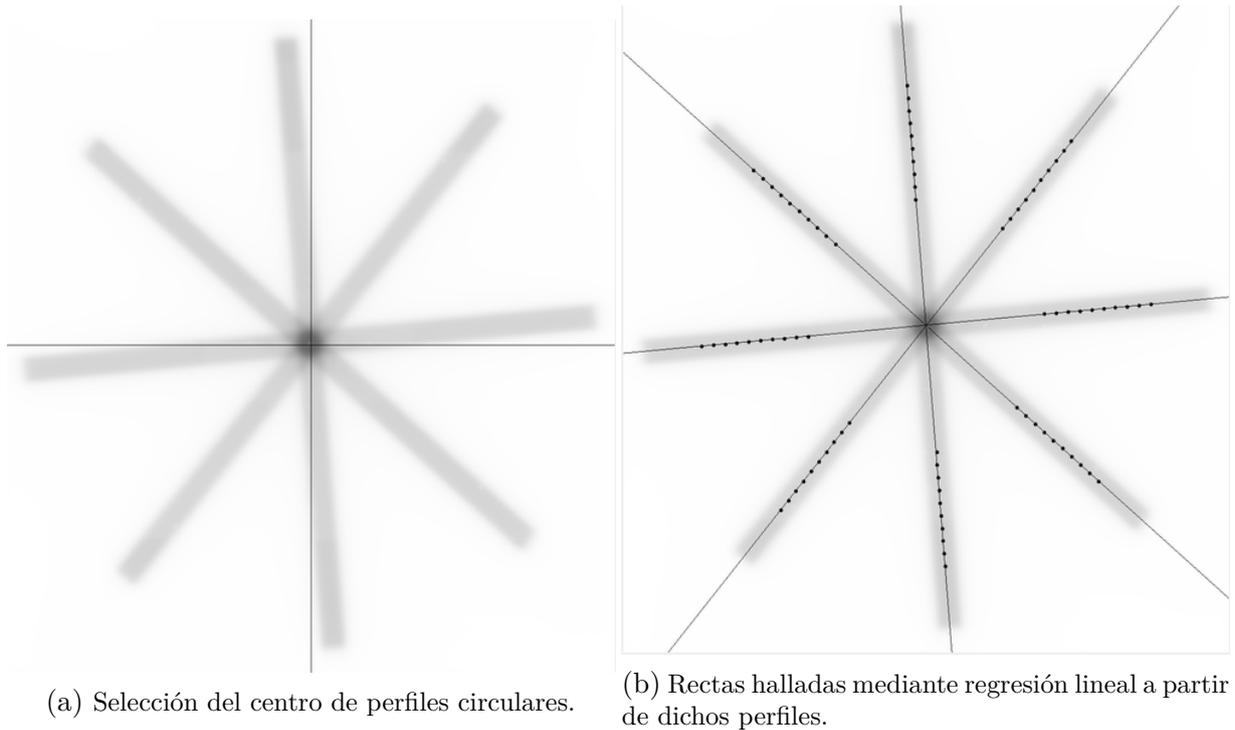


Figura 3.2: Determinación del centro de la estrella de radiación. (Cortesía del Servicio de Radiofísica del HUVM)

Una vez indicado el punto, el programa se encargará de dibujar rectas para cada brazo de la estrella a partir de la regresión lineal de puntos dibujados por la intersección de los perfiles circulares con cada brazo, sirviéndose de la información provista por los perfiles de intensidad (Figura 3.2). Dichas rectas intersectarán entre sí, idealmente en el isocentro de radiación. No obstante, si hacemos *zoom* al centro de la estrella encontraremos que las rectas no convergen en un único punto (Figura 3.3).

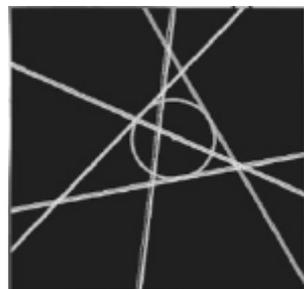


Figura 3.3: *Zoom* al punto de intersección entre las rectas en el centro de la estrella. [Depuydt et al., 2012]

La manera en la que el programa encuentra el isocentro de radiación consiste en, siguiendo su definición, buscar ternas de líneas y dibujar un círculo que sea tangente a

cada línea de la terna. Como se producen múltiples intersecciones se da lugar a varias ternas cuyos círculos internos tienen un radio distinto. Se considera que el isocentro de radiación lo determina el círculo con menor radio de todas las ternas posibles y que intersecte con todas las rectas dibujadas, como puede verificarse en la Figura 3.3.

3.1.2 Centro de referencia: punto de corte de los láseres

La posición del isocentro hallada anteriormente es la considerada como verdadera. No obstante, puede ocurrir que los láseres de la habitación no estén alineados correctamente, esto es, que el punto de corte entre ambos no se encuentre en la misma posición que el isocentro de radiación, lo que puede suponer que la mesa no esté correctamente alineada con el colimador y, por tanto, se irradie en un punto no deseado. Para ello, el programa lleva a cabo un segundo proceso, en este caso centrándose en la búsqueda de dicho punto de corte, sirviéndose de la ayuda provista por el Iso-Align.

Para encontrar el centro de referencia se hace uso de las marcas radiopacas que definen los láseres en el Iso-Align, dando lugar a una imagen con una serie de puntos de referencia con distintas orientaciones para hallar el centro de la forma más exacta posible. Esto, que se conoce como imagen del Iso-Align (Figura 3.4), será nuestro punto de partida.

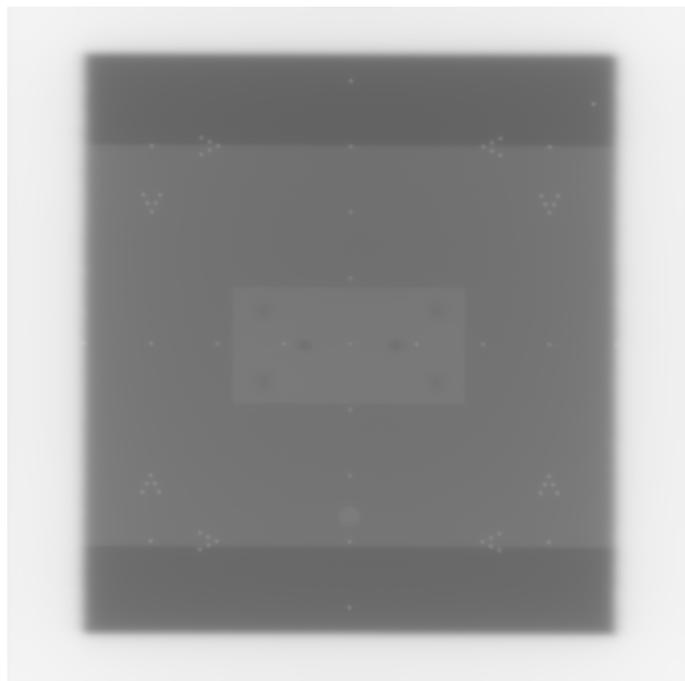


Figura 3.4: Imagen del Iso-Align usada para calibración computerizada. (Cortesía del Servicio de Radiofísica del HUVM)

Empleando dichos puntos de referencia (las marcas blanquecinas presentes en la imagen del Iso-Align) pueden trazarse rectas calculadas mediante regresión lineal que intersecten en el centro exacto de la imagen, definiendo así el punto de corte de los láseres, pues estos están alineados con el Iso-Align. La manera en la que se indican las coordenadas de tales puntos es señalándolas manualmente, haciendo *zoom* a la imagen del Iso-Align y haciendo clic individualmente en cada punto para dibujar las rectas horizontal y vertical (Figura 3.5).

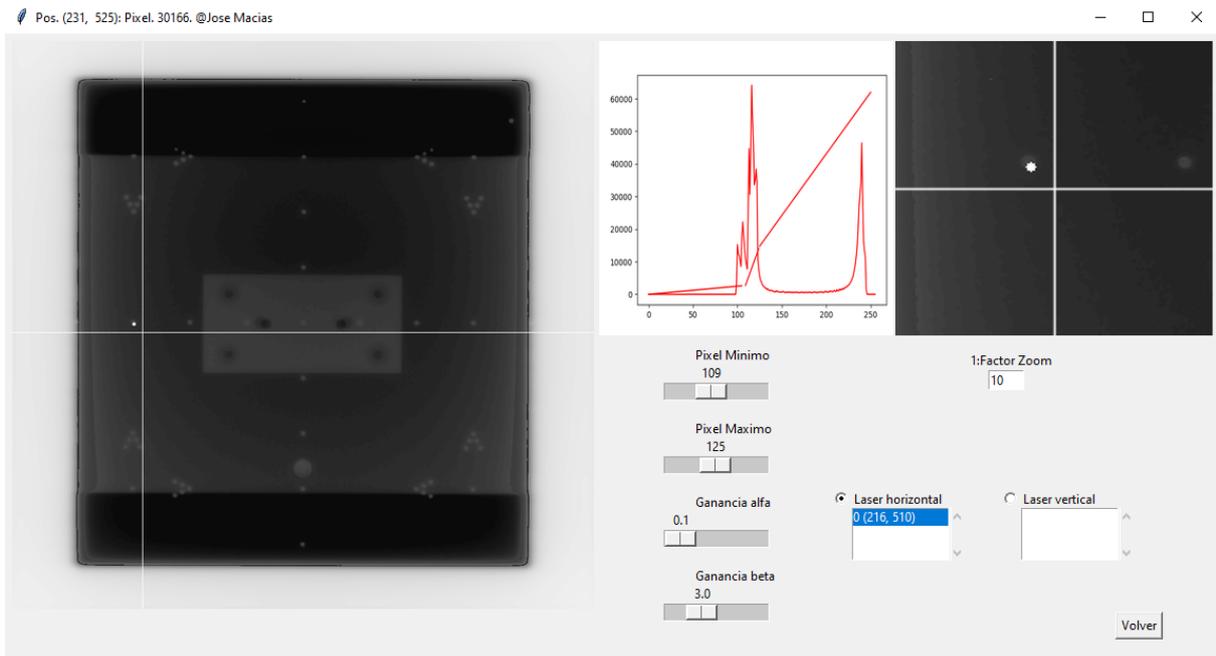


Figura 3.5: *Software* empleado para la detección del isocentro de radiación. (Cortesía de su creador y desarrollador, José Macías Jaén, HUVM)

Con ambas rectas definidas, el programa halla el punto de corte de los láseres. Este punto, el centro definido por las rectas de los marcadores radiopacos, a su vez coincidentes con los láseres, debe coincidir con el centro de la estrella de radiación. No obstante, esto no suele ocurrir habitualmente, de modo que a veces es necesario recentrar los láseres para que ambos centros coincidan, lo que supone la calibración total y correcta del isocentro.

Llevados a cabo ambos procedimientos, el programa genera un archivo PDF que recapitula toda la información obtenida, indicando la posición del isocentro, el número de *píxeles* por milímetro de la imagen, el radio del isocentro y el punto de corte entre los láseres, entre otros, con el objeto de tener toda la información relevante en un mismo documento para consulta del equipo radiofísico.

3.2 Implementación de la transformada de Hough en Python

El anterior procedimiento, aunque es plenamente funcional y sus resultados entran dentro de los parámetros sugeridos por la *American Association of Physicists in Medicine* (AAPM) [Klein et al., 2009], así como por la Sociedad Española de Física Médica [Pinza-Molina and Lliso-Valverde, 2009] e impuestos en el Real Decreto 1566/1998, de 17 de julio [BOE, 1998], es un proceso que resulta costoso en cuanto a tiempo empleado por el personal del servicio de radiofísica a la hora de llevar a cabo las sesiones de QA, puesto que deben tener especial cuidado al fijar el centro más exacto para cada punto, tan exacto como les permita discernir la vista. Teniendo esto en cuenta, vemos que el papel que puede desempeñar la transformada de Hough es crítico si buscamos perfeccionar el *software* y agilizar el proceso de detección del isocentro de radiación. Con una detección automatizada de los patrones circulares que suponen los puntos de referencia de la imagen del Iso-Align erradicamos toda intervención humana en el proceso, evitando así posibles imprecisiones a la hora de señalar dichos puntos en el programa para las respectivas regresiones lineales, además del recorte de tiempo que ello supone.

Es por este motivo que hemos desarrollado un código que, incorporado al programa original, se encargue de llevar a cabo el proceso anterior de forma automática sirviéndonos de la transformada como base. La manera que tiene el programa de llevar esto a cabo es la que sigue.

3.2.1 Tratamiento previo de la imagen

Atendiendo a la imagen del Iso-Align encontramos muchas más marcas radiopacas de las que realmente necesitamos, como por ejemplo algunas en las esquinas con forma de flecha, como puede apreciarse en la Figura 3.4. En nuestro caso dichas marcas no son necesarias para encontrar el centro de la imagen, de modo que deben ser descartadas para que no interfieran en las regresiones lineales. Conseguimos esto mediante el empleo de recortes en las regiones de interés, esto es, en las zonas circundantes a las marcas radiopacas que forman las líneas vertical y horizontal, respectivamente (véase la Figura 3.6). Sin embargo, primero hay que extraer toda la información posible de la imagen.

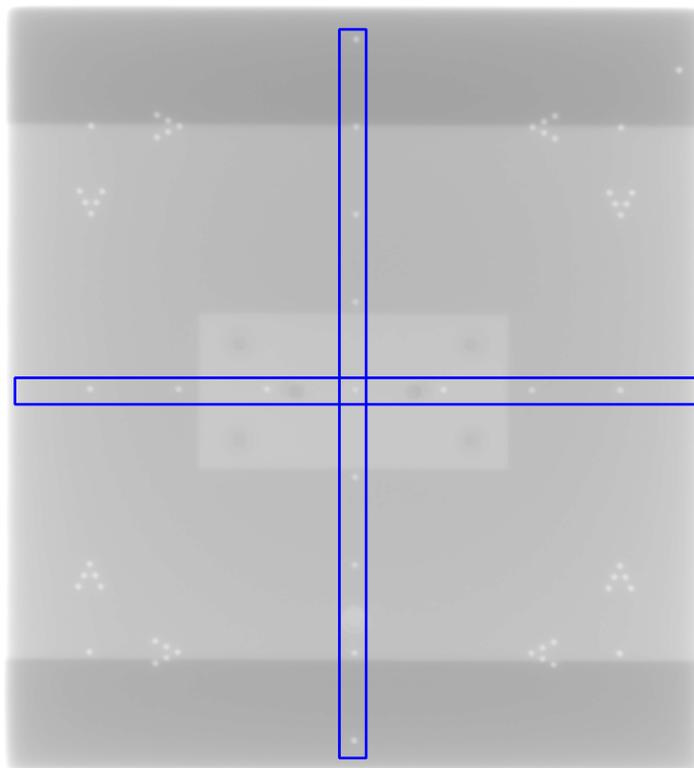


Figura 3.6: Imagen del Iso-Align con regiones de interés señaladas.

La primera dificultad técnica a tener en cuenta es el hecho de que la imagen del Iso-Align viene en un formato de datos llamado DICOM (del inglés «*Digital Imaging and Communications in Medicine*» y escrito como `.dcm`), un estándar de gestión, almacenamiento y transmisión de información e imágenes en medicina ¹. Para extraer toda la información recopilada usaremos el método `pydicom.dcmread()`, de la librería `pydicom.filereader`. Una vez hemos indicado al programa que lea el archivo DICOM cargamos la imagen como una matriz de datos, cuyos elementos harán referencia a la información visual en cada *píxel*. La dimensión de dicha matriz es 1024×1024 *píxeles*.

Una vez hecho esto basta con que, de forma aproximada, le indiquemos al programa qué elementos de la matriz de imagen original pertenecen al recorte que queremos realizar para cada región de interés. Para llevar esto a cabo, debemos señalar el ancho y el largo del recorte en *píxeles*, usando como referencia de origen (el elemento $A_{1,1}$ de la matriz original) el *píxel* situado en la esquina superior izquierda de la imagen original. De esta

¹Información extraída de DICOM <https://dicom.nema.org/medical/dicom/current/output/pdf/part01.pdf>. Accedido en septiembre de 2023.

manera hacemos recortes rectangulares de 1024 *píxeles* de largo y unos 30 de ancho, lo que da el resultado que se puede observar en la Figura 3.6.

Con los recortes de las regiones de interés realizados podemos comenzar con la detección de patrones circulares. No obstante, antes hay que indicar si el recorte que estamos empleando es el vertical o el horizontal. La manera de hacer esto es apelando a sus dimensiones: si el número de *píxeles* comprendido entre los elementos de fila de la matriz es menor o igual al fijado por la región de interés, nos encontraremos en el caso del corte horizontal, mientras que cualquier otro caso corresponderá al corte vertical. Una vez hecho esto se pasa cada recorte a escala de grises mediante el método `cvtColor()` de la librería de tratamiento de imágenes *Open Source Computer Vision* (OpenCV). También se hace uso de un filtro de mediana con `medianBlur()`, de la misma librería, que se encargará de limpiar la imagen de impurezas, el cual se ejecuta en cada elemento de esta y reemplaza cada *píxel* con la mediana de los *píxeles* dentro de un cuadrado de vecinos alrededor del *píxel* tratado. [Szeliski, 2022]

3.2.2 El método *HoughCircles*

La imagen del Iso-Align ya ha recibido el tratamiento necesario para emplear la transformada de Hough. En Python dicha transformada existe como método de la librería OpenCV tanto para patrones lineales (`HoughLinesP()`, usado previamente en el ejemplo del capítulo anterior) como para patrones circulares (`HoughCircles()`), que es lo que usaremos en el código.

De forma general, `HoughCircles()` es un método que depende de una serie de parámetros que el usuario deberá indicar para optimizar la detección de patrones circulares. Dichos parámetros son los que siguen ²:

- **«*image*»**: imagen de entrada. Debe ser de un único canal y en 8 bits. Si la imagen original es a color hay que hacer un paso previo a escala de grises;
- **«*method*»**: método que se va a emplear para detectar patrones circulares en la imagen de entrada. En nuestro caso se hará uso de `HOUGH_GRADIENT`, que realiza una segunda votación empleando un gradiente que determine la dirección hacia la

²Extraído de *Open Source Computer Vision* (OpenCV) <https://t.ly/KD2L2>. Acceso: agosto de 2023.

cual se cierra la figura analizada;

- «***dp***»: relación (ratio) inversa entre la resolución del acumulador y la resolución de la imagen. Cuanto mayor sea ***dp***, menor será la matriz del acumulador y viceversa;
- «***minDist***»: distancia mínima fijada entre los centros de los círculos detectados (x, y) . Si es muy pequeña pueden detectarse círculos espurios mientras que si es muy grande pueden detectarse demasiado pocos;
- «***param1***»: parámetro que fija el umbral del gradiente para llevar a cabo la detección de bordes;
- «***param2***»: valor umbral para el acumulador en `HOUGH_GRADIENT`. Cuanto más pequeño se fije el umbral, mayor será la cantidad de círculos detectados. Cuanto mayor, más círculos se devolverán potencialmente;
- «***minRadius***»: fija el valor mínimo que debe tener el radio de los círculos detectados;
- «***maxRadius***»: fija el valor máximo que debe tener el radio de los círculos detectados.

Cada recorte precisa de una parametrización distinta para el método `HoughCircles()`, de modo que, una vez hacemos la distinción del recorte con el que se está trabajando, indicamos asimismo los parámetros empleados para la transformada en cada recorte. Como se requiere que el usuario indique dichos parámetros, ha sido necesario un proceso de ensayo y error para encontrar la combinación más general y óptima posible de parámetros.

Una vez hemos dispuesto correctamente dichos parámetros para cada recorte, el método `HoughCircles()` nos devuelve una tupla indicando toda la información de los círculos detectados en la imagen del Iso-Align, esto es, las coordenadas del centro y el radio de cada círculo (x, y, r) . Separamos la información de las coordenadas del centro del radio y creamos una nueva lista que incluya únicamente dichas coordenadas con el objeto de realizar la regresión lineal pertinente, repitiendo el proceso en ambos recortes.

3.2.3 Valor medio y varianza como alternativa a la regresión lineal

A la hora de llevar a cabo la regresión lineal surge otro problema cuya resolución resulta imprescindible para continuar con el proceso. Programar un modelo de regresión lineal para rectas cuya pendiente es infinita (como es en el caso del recorte vertical) puede convertirse en una tarea cuya dificultad excede los límites de los objetivos marcados, de modo que se hace una serie de cálculos con vistas a paliar este inconveniente.

Comenzamos hallando el valor medio (\bar{x}) entre los valores de las abcisas de los círculos detectados en el recorte vertical, así como su desviación estándar (σ). Con ambas magnitudes determinadas pasamos a definir el cociente entre ambas como:

$$C = \frac{\sigma}{\bar{x}} \quad (3.1)$$

Si tenemos un conjunto de puntos que no esten completamente alineados ocurrirá que $\sigma \neq 0$ y, por lo tanto, $C \neq 0$, lo cual nos indica que la pendiente es finita, de modo que se puede hallar su valor mediante el modelo de regresión lineal establecido en el código. Por contra, en el caso puramente vertical, esto es, aquel cuya pendiente de la recta de regresión es infinita, encontramos que cualquier abcisa de cualquier punto detectado determina el valor medio, mientras que $\sigma \rightarrow 0$, de modo que nos basta con unir los centros de dos de los círculos detectados para dibujar la recta de mejor ajuste. Cabe señalar que, en el desarrollo del trabajo, hemos considerado el caso puramente vertical únicamente para el caso en que $\sigma = 0$ y, por ende, $C = 0$. Podemos visualizar mejor este razonamiento atendiendo a la Figura 3.7.

3.2.4 Representación y punto de corte

Una vez hemos conseguido realizar la regresión lineal en cada recorte llega el momento de hallar el punto de intersección entre ambas rectas. No obstante, una vez más surge la necesidad de establecer una bifurcación en el procedimiento según el caso en el que nos encontremos respecto de los círculos detectados en el recorte vertical, atendiendo a lo que hemos visto en la sección anterior.

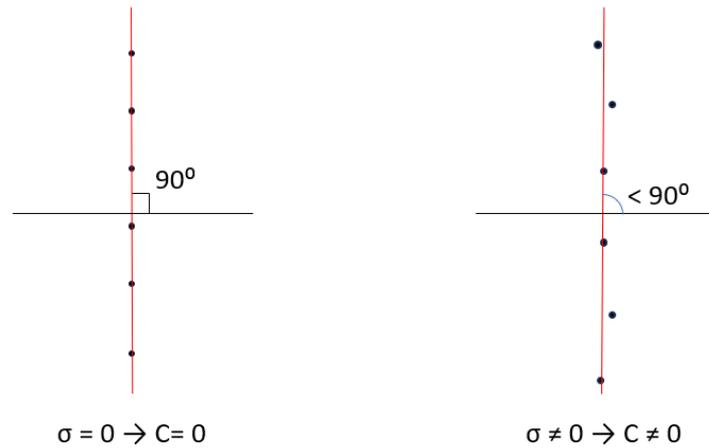


Figura 3.7: Ejemplo de ajuste lineal para el caso puramente vertical (izquierda) y pendiente finita (derecha).

Para el caso de pendiente finita, esto es, menor a 90° según nuestro criterio, nos encontramos con dos ecuaciones de la recta, de modo que hallar el punto de corte entre ambas resulta sencillo; basta con solucionar el sistema de ecuaciones. Es en el caso de la pendiente infinita donde vuelve a aparecer el problema, puesto que no tenemos la ecuación de la recta de forma explícita. La manera de hallar el punto de corte en este caso será la siguiente: cuando dos rectas intersectan quiere decir que, en el punto de corte, ambas ecuaciones de la recta se cumplen, dando el mismo resultado. Haremos uso de esto teniendo en cuenta que, en el caso puramente vertical, las abscisas tienen el mismo valor, dentro de las tolerancias marcadas, para todos los puntos detectados. Sustituyendo dicho valor en la ecuación de la recta obtenida mediante regresión lineal en el recorte horizontal obtendremos el valor de la ordenada correspondiente al punto de corte.

Con el punto de corte determinado, solo queda representar todos los datos extraídos en la imagen del Iso-Align, de modo que pueda tenerse una segunda comprobación (en este caso, visual) de que el procedimiento se ha llevado a cabo correctamente.

Capítulo 4

Resultados

Con el algoritmo completamente desarrollado, queda atender a los resultados obtenidos para distintas imágenes del Iso-Align. No obstante, lejos de considerar el procedimiento como exitoso, aún queda comprobar su reproducibilidad y que los resultados obtenidos sean fidedignos a los de la metodología original, añadiendo el ahorro de tiempo de la nueva metodología.

4.1 Detección de patrones circulares en imágenes de Iso-AlignTM y determinación del punto de corte de los láseres

Aplicando la rutina basada en la transformada de Hough para patrones circulares y siguiendo el procedimiento descrito en la metodología obtenemos el siguiente resultado para una imagen del Iso-Align:

- Punto de corte de los láseres: $(512, 507) \pm 1$ px
- Tiempo de ejecución del programa: $0,34 \pm 0,01$ s

Visualmente, este resultado puede verificarse en la Figura 4.1.

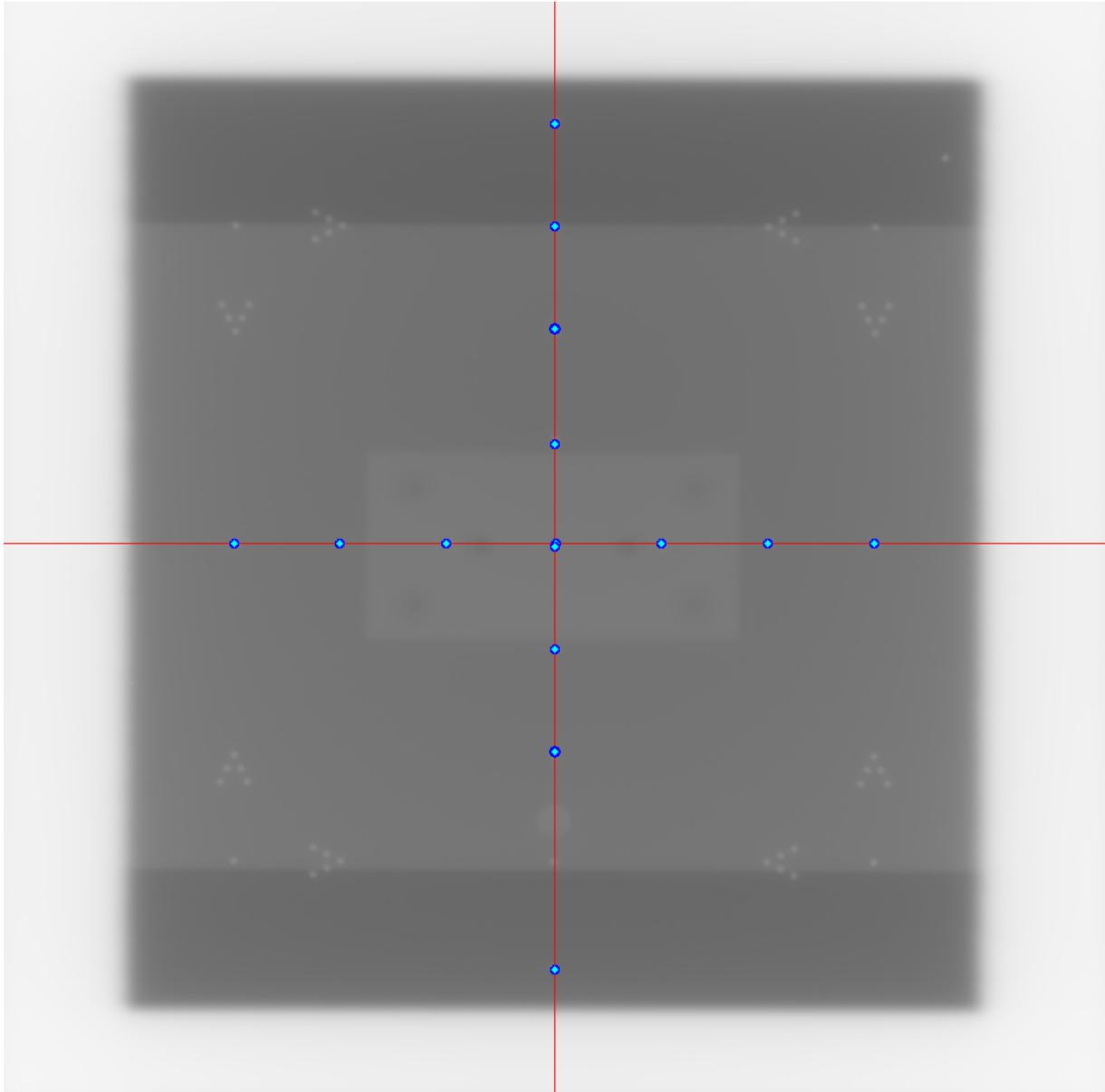


Figura 4.1: Imagen del Iso-Align con el centro determinado haciendo uso del algoritmo basado en la transformada de Hough (muestra del 15/02/2023).

4.2 Comprobación de la robustez del código. Desviación respecto de la metodología original

A priori, si nos fijamos únicamente en los resultados obtenidos en la sección anterior podría parecer que, en efecto, el algoritmo desarrollado en el presente trabajo supone una mejoría con respecto a la metodología tradicional. No obstante, para determinar de forma cuantitativa que así es resulta necesario realizar una comparativa entre ambos procedimientos, para lo cual hemos ejecutado diez veces el procedimiento que se lleva a cabo en la actualidad, cronometrando cada intento. Pueden apreciarse tales resultados en las Tablas 4.1 y 4.2.

| 15/12/2023 | X_o (px) | Y_o (px) | X_a (px) | Y_a (px) | t_o (s) | t_a (s) |
|-----------------|--------------|--------------|------------|------------|-----------|-------------|
| Intento 1 | 511,7 | 507,1 | 512 | 507 | 90 | 0,34 |
| Intento 2 | 512,6 | 507,7 | 512 | 507 | 66 | 0,34 |
| Intento 3 | 512,2 | 508,8 | 512 | 507 | 65 | 0,36 |
| Intento 4 | 512,0 | 508,8 | 512 | 507 | 76 | 0,43 |
| Intento 5 | 512,3 | 508,2 | 512 | 507 | 62 | 0,37 |
| Intento 6 | 512,2 | 508,8 | 512 | 507 | 67 | 0,40 |
| Intento 7 | 512,0 | 507,4 | 512 | 507 | 64 | 0,38 |
| Intento 8 | 512,0 | 508,8 | 512 | 507 | 58 | 0,41 |
| Intento 9 | 511,8 | 508,0 | 512 | 507 | 63 | 0,38 |
| Intento 10 | 512,2 | 507,1 | 512 | 507 | 62 | 0,40 |
| MEDIA | 512,1 | 508,1 | 512 | 507 | 67 | 0,38 |
| σ_x (px) | 0,24 | | | | | |
| σ_y (px) | 0,72 | | | | | |

Tabla 4.1: Comparativa entre los resultados de la metodología original y la actual para la obtención del centro en la imagen del Iso-Align del 15/12/2023. El subíndice «o» hace referencia a los datos obtenidos por la metodología original, mientras que «a» corresponde a la metodología desarrollada en este trabajo.

| | |
|---------------------------------|------|
| $\max(\Delta X)$ (px) | 0,67 |
| $\max(\Delta Y)$ (px) | 1,86 |
| $\min(\Delta X)$ (px) | 0,0 |
| $\min(\Delta Y)$ (px) | 0,14 |
| $\langle \Delta X \rangle$ (px) | 0,13 |
| $\langle \Delta Y \rangle$ (px) | 1,12 |

Tabla 4.2: Diferencias absolutas entre las coordenadas X e Y del punto de corte de los láseres encontradas por ambas metodologías para la imagen del Iso-Align del 15/02/2023. Se contemplan los casos de diferencia máxima, mínima y promedio.

Antes que nada, es necesario señalar que hemos decidido dar una estimación de la incertidumbre más adecuada que en la metodología original. Esto se debe a que, en dicha metodología, la precisión por defecto con la que se reporta la medida es del orden de 10^{-12} px. Sin embargo, cada vez que ejecutamos el programa tradicional, al depender de la precisión del humano los resultados obtenidos pueden diferir entre sí para cada intento, de modo que la precisión dada por este método no es realista, mientras que en la metodología actual siempre se obtiene el mismo resultado con una precisión de ± 1 px.

La primera deducción que podemos hacer atendiendo a los datos de la Tabla 4.1 consiste en que, mientras que hay una cierta desviación entre las coordenadas encontradas mediante la metodología tradicional, en la actual esta desaparece por completo, garantizando su reproducibilidad. Por otra parte, la diferencia entre tiempos resulta abrumadora, anotando una diferencia promedio de más de 60 segundos, lo que supone un recorte en el tiempo de ejecución en torno del 99,4%. Además, puede apreciarse en la Tabla 4.2 que las diferencias entre ambos procedimientos son mínimas, teniendo menos de 2 px de diferencia para el caso de máxima diferencia, lo cual, considerando una resolución de 4 px por milímetro, no supone ni medio milímetro de diferencia.

Para esta primera imagen del Iso-Align los resultados son bastante positivos. No obstante, lejos de asegurar el éxito del nuevo procedimiento creemos necesario analizar una serie de imágenes con objeto de verificar que el código es reproducible y que sus resultados son fiables a la hora de hallar el punto de corte entre los láseres. Para esto, llevamos a cabo el mismo procedimiento en nueve imágenes de Iso-Align, sacadas en distintas pruebas de QA para LINAC distintos. Mostramos los resultados de dos de ellas en las tablas siguientes, mientras que el resto pueden encontrarse en el Anexo correspondiente.

Aunque atendiendo a los resultados anteriores queda patente la mejora que presenta la nueva metodología, hemos considerado oportuno mostrar también un caso problemático, este es, el correspondiente a la imagen del Iso-Align tomada el 05/07/2023, presentado en las tablas 4.5 y 4.6 y que puede observarse de forma visual en la Figura 4.2. En este caso tanto las desviaciones máxima como promedio pasan del milímetro de diferencia. Esto, principalmente, puede deberse a la elección de los parámetros escogidos en el recorte vertical, ya que en este caso sólo se detecta un punto en dicho recorte, y quizá pueda subsanarse este error escogiendo otra configuración de parámetros distinta. Sin embargo,

a pesar de esto la desviación sigue siendo asumible, lo que indica que, incluso en el peor de los casos la nueva metodología funciona correctamente y presenta unos resultado fiables.

| 08/03/2023 | X_o (px) | $Y_o(px)$ | $X_a(px)$ | $Y_a(px)$ | $t_o(s)$ | $t_a(s)$ |
|-----------------|--------------|--------------|------------|------------|-----------|-------------|
| Intento 1 | 510,2 | 512,7 | 512 | 513 | 76 | 0,46 |
| Intento 2 | 510,0 | 514,4 | 512 | 513 | 86 | 0,38 |
| Intento 3 | 510,7 | 514,7 | 512 | 513 | 73 | 0,38 |
| Intento 4 | 509,6 | 514,7 | 512 | 513 | 73 | 0,38 |
| Intento 5 | 509,6 | 515,0 | 512 | 513 | 80 | 0,37 |
| Intento 6 | 510,4 | 514,4 | 512 | 513 | 65 | 0,35 |
| Intento 7 | 510,7 | 514,0 | 512 | 513 | 77 | 0,40 |
| Intento 8 | 509,8 | 514,3 | 512 | 513 | 61 | 0,37 |
| Intento 9 | 510,6 | 514,4 | 512 | 513 | 90 | 0,38 |
| Intento 10 | 510,0 | 514,4 | 512 | 513 | 66 | 0,39 |
| MEDIA | 510,2 | 514,3 | 512 | 513 | 75 | 0,39 |
| σ_x (px) | 0,41 | | | | | |
| σ_y (px) | 0,59 | | | | | |

Tabla 4.3: Comparativa entre los resultados de la metodología original y la actual para la obtención del centro en la imagen del Iso-Align del 08/03/2023. El subíndice «o» hace referencia a los datos obtenidos por la metodología original, mientras que «a» corresponde a la metodología desarrollada en este trabajo.

| | |
|---------------------------------|------|
| $\max(\Delta X)$ (px) | 2,40 |
| $\max(\Delta Y)$ (px) | 2,0 |
| $\min(\Delta X)$ (px) | 1,30 |
| $\min(\Delta Y)$ (px) | 0,30 |
| $\langle \Delta X \rangle$ (px) | 1,84 |
| $\langle \Delta Y \rangle$ (px) | 1,30 |

Tabla 4.4: Diferencias absolutas entre las coordenadas X e Y del punto de corte de los láseres encontradas por ambas metodologías para la imagen del Iso-Align del 08/03/2023. Se contemplan los casos de diferencia máxima, mínima y promedio.

Por último, cabe mencionar que de los nueve casos analizados el nuevo programa se ha ejecutado con éxito en siete. Existen dos casos donde no hemos podido detectar ningún punto en el recorte vertical, de modo que no hemos podido realizar la regresión lineal pertinente ni hallar el punto de corte de los láseres en ambos casos. La principal razón por la que puede deberse esto radica en el mismo motivo que para el caso discutido anteriormente: la elección de parámetros por parte del usuario.

| 05/07/2023 | X_o (px) | Y_o (px) | X_a (px) | Y_a (px) | t_o (s) | t_a (s) |
|-----------------|--------------|--------------|------------|------------|-----------|-------------|
| Intento 1 | 516,1 | 507,3 | 512 | 508 | 80 | 0,35 |
| Intento 2 | 516,8 | 508,9 | 512 | 508 | 73 | 0,34 |
| Intento 3 | 516,8 | 508,7 | 512 | 508 | 84 | 0,34 |
| Intento 4 | 516,6 | 509,1 | 512 | 508 | 72 | 0,34 |
| Intento 5 | 517,2 | 509,1 | 512 | 508 | 77 | 0,34 |
| Intento 6 | 517,0 | 508,4 | 512 | 508 | 75 | 0,34 |
| Intento 7 | 516,3 | 508,7 | 512 | 508 | 77 | 0,36 |
| Intento 8 | 516,0 | 507,6 | 512 | 508 | 76 | 0,35 |
| Intento 9 | 515,9 | 508,9 | 512 | 508 | 68 | 0,35 |
| Intento 10 | 515,2 | 509,6 | 512 | 508 | 69 | 0,33 |
| MEDIA | 516,4 | 508,6 | 512 | 508 | 75 | 0,34 |
| σ_x (px) | 0,58 | | | | | |
| σ_y (px) | 0,66 | | | | | |

Tabla 4.5: Comparativa entre los resultados de la metodología original y la actual para la obtención del centro en la imagen del Iso-Align del 05/07/2023. El subíndice «o» hace referencia a los datos obtenidos por la metodología original, mientras que «a» corresponde a la metodología desarrollada en este trabajo.

| | |
|---------------------------------|------|
| $\max(\Delta X)$ (px) | 5,20 |
| $\max(\Delta Y)$ (px) | 1,60 |
| $\min(\Delta X)$ (px) | 3,20 |
| $\min(\Delta Y)$ (px) | 0,40 |
| $\langle \Delta X \rangle$ (px) | 4,39 |
| $\langle \Delta Y \rangle$ (px) | 0,63 |

Tabla 4.6: Diferencias absolutas entre las coordenadas X e Y del punto de corte de los láseres encontradas por ambas metodologías para la imagen del Iso-Align del 05/07/2023. Se contemplan los casos de diferencia máxima, mínima y promedio.

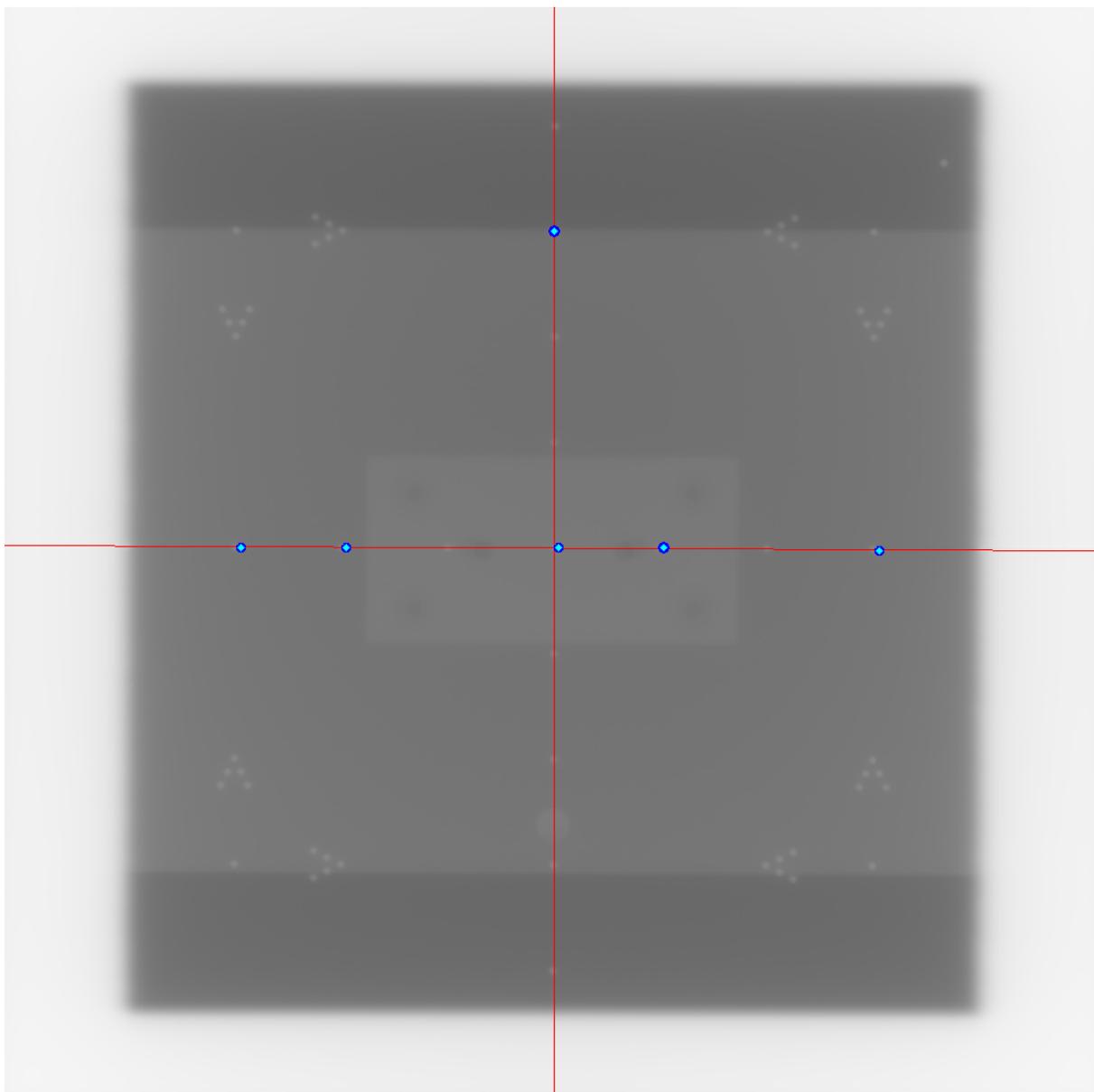


Figura 4.2: Imagen del Iso-Align con el centro determinado haciendo uso del algoritmo basado en la transformada de Hough (muestra del 05/07/2023).

Capítulo 5

Conclusiones

Se ha conseguido llevar a cabo una nueva manera de hallar el punto de corte entre los láseres para los procedimientos de calibración del isocentro de radiación y QA en radiofísica hospitalaria. La principal ventaja que supone este procedimiento consiste en la no intervención del usuario en el proceso, seguida de una reducción del tiempo empleado realmente significativa. Mientras que los resultados resultan satisfactorios, obteniendo una tasa de funcionamiento de la rutina en siete de las nueve imágenes del Iso-Align usadas, una desviación del orden de 1 *mm* en el peor de los casos y presentando una reducción temporal de aproximadamente el 99 %, resulta ineludible señalar la posible pérdida de generalidad de la metodología que, debido a la previa elección de parámetros por parte del usuario en la detección de patrones circulares, puede llevar a retrasos puntuales si resulta necesario modificar dichos parámetros para que puedan obtenerse resultados para algunas imágenes.

Por otra parte, aunque ligeras, las desviaciones en los puntos de cortes detectados contienen fuentes de error probablemente debidas al redondeo de decimales, paso necesario para que la rutina funcione, así como al filtro de limpieza por mediana realizado en el tratamiento previo de la imagen. Sin embargo, hemos podido comprobar de primera mano que con respecto a la metodología original resulta una clara mejora puesto que, debido a la intervención humana, los resultados obtenidos en dicha metodología son distintos cada vez que se ejecuta, mientras que en nuestro procedimiento se obtiene el mismo resultado en cada ejecución con una dispersión nula.

No obstante, a pesar de todo esto se ha conseguido implementar esta rutina con éxito en los LINAC de Servicio de Radiofísica en el HUVM, corroborando su viabilidad con una puesta en marcha inmediata, además de la capacidad de ser reproducible en otros dispositivos y en otros organismos.

Entre la filosofía y las cualidades de la nueva metodología se presenta la capacidad de poder modificarse y llevar a cabo mejoras rápidamente mediante procedimientos como el aprendizaje automático, dejando como futuros proyectos a realizar conseguir que sea la propia computadora quien decida qué parámetros son los más acertados para cada recorte, optimizando el número de puntos obtenidos y descartando aquellos que son espurios. Asimismo, un siguiente paso a dar puede ser que este proceso se realice directamente sobre la película radiocrómica en tiempo real, sin necesidad de digitalizar la imagen del Iso-Align. Todo considerando como base lo descrito en el presente trabajo.

Aunque queda un largo camino por delante en la lucha contra el cáncer, podemos afirmar con vehemencia que con el presente trabajo hemos avanzado en la dirección correcta, sentando las bases para lo que puede llegar a ser unos procedimientos de control de calidad más ágiles y menos fatigosos para el equipo de técnicos y radiofísicos, eficiencia que a la postre lleva asociada una repercusión directa (y positiva) en la calidad de la administración del tratamiento de radioterapia a los pacientes.

Bibliografía

- [Argos, Lucía, 1991] Argos, Lucía (1991). El accidente del clínico de Zaragoza, una cadena de fallos humanos única en el mundo, según los expertos. https://elpais.com/diario/1991/10/12/sociedad/687222014_850215.html?event_log=go. *El País*. Accedido en septiembre de 2023.
- [Ballard, 1981] Ballard, D. (1981). Generalizing the Hough transform to detect arbitrary shapes. *Pattern Recognition*, 13:111–122.
- [BOE, 1998] BOE (1998). Real Decreto 1566/1998, de 17 de julio, por el que se establecen los criterios de calidad en radioterapia. *Boletín Oficial del Estado*, 206, de 28 de agosto de 1998. <https://www.boe.es/eli/es/rd/1998/07/17/1566/con>.
- [CNECS, 1996] CNECS (1996). *Radiofísica Hospitalaria*, pages 591–604. Ministerio de Sanidad. Gobierno de España.
- [Cortés Giraldo, 2011] Cortés Giraldo, M. A. (2011). Desarrollos y aplicaciones de GEANT4 para radioterapia y microdosimetría en detectores y circuitos integrados.
- [Depuydt et al., 2012] Depuydt, T., Penne, R., Verellen, D., Hrbáček, J., Lang, S., Leysen, K., Vandevondel, I., Poels, K., Reynders, T., Gevaert, T., Duchateau, M., Tournel, K., Boussaer, M., Cosentino, D., Garibaldi, C., Solberg, T. D., and Ridder, M. D. (2012). Computer-aided analysis of star shot films for high-accuracy radiation therapy treatment units. *Physics in Medicine and Biology*, 57:2997–3011.
- [Duda and Hart, 1972] Duda, R. O. and Hart, P. E. (1972). Use of the hough transformation to detect lines and curves in pictures. *Communications of the ACM*, 15:11–15.
- [Hough, 1962] Hough, P. (1962). Method and means for recognizing complex patterns.

- US Patent. No. 3 069 654 Dec. 18, 1962.
- [Jameson et al., 2015] Jameson, R., Bisognano, J., and Lapostolle, P. (2015). *Linear Accelerators*, pages 495–532. John Wiley Sons, Ltd.
- [Klein et al., 2009] Klein, E. E., Hanley, J., Bayouth, J., Yin, F.-F., Simon, W., Dresser, S., Serago, C., Aguirre, F., Ma, L., Arjomandy, B., Liu, C., Sandin, C., and Holmes, T. (2009). AAPM Task Group 142 report: Quality assurance of medical accelerators. *Medical Physics*, 36:4197–4212.
- [Pinza-Molina and Lliso-Valverde, 2009] Pinza-Molina, C. and Lliso-Valverde, F. (2009). *Control de calidad en aceleradores de electrones para uso médico*. Sociedad Española de Física Médica; Madrid (España).
- [Pratt, 2007] Pratt, W. K. (2007). *Digital image processing : PIKS Scientific inside (Third Edition)*. Wiley-Interscience; Los Altos, CA.
- [Roser, Max, 2020] Roser, Max (2020). Is the world making progress against cancer? <https://ourworldindata.org/progress-against-cancert>. Our World in Data. Accedido en agosto de 2023.
- [Sung et al., 2021] Sung, H., Ferlay, J., Siegel, R. L., Laversanne, M., Soerjomataram, I., Jemal, A., and Bray, F. (2021). Global cancer statistics 2020: Globocan estimates of incidence and mortality worldwide for 36 cancers in 185 countries. *CA: A Cancer Journal for Clinicians*, 71(3):209–249.
- [Szeliski, 2022] Szeliski, R. (2022). *Computer Vision: Algorithms and Applications (2nd Edition)*. Springer Nature Switzerland AG; Cham (Switzerland).
- [Zhang et al., 2015] Zhang, M., Zhou, S.-M., and Qu, T. (2015). What do we mean when we talk about the linac isocenter? *International Journal of Medical Physics, Clinical Engineering and Radiation Oncology*, 04:233–242.

Apéndice A

Código empleado en el desarrollo del algoritmo

```
1 import cv2
2 import numpy as np
3
4 img = cv2.imread("gris5.jpg")
5 #cv2.imshow("GRIS",img)
6 gray = cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)
7 edges = cv2.Canny(gray,25,150) #25, 150
8
9 lines = cv2.HoughLinesP(edges, 1, np.pi/180,70)
10
11 for line in lines:
12     x1,y1,x2,y2 = line[0]
13     cv2.line(img,(x1,y1), (x2,y2),(255,255,0),2)
14 cv2.imshow("Edges",edges)
15 cv2.imshow("Imagen",img)
16 cv2.imwrite("Bordes.png",edges)
17 cv2.imwrite("GRISLin.png",img)
18 cv2.waitKey(0)
19 cv2.destroyAllWindows()
```

Listing A.1: Código para la detección de patrones rectilíneos en la imagen ejemplo de la Figura 2.12.

```

1 # ---Deteccion de patrones circulares en una lamina de isoalign mediante
   el metodo de la transformada de Hough (HoughCircles y HOUGH_GRADIENT
   en OpenCSV)---
2
3 # La idea principal es que el codigo lea la imagen en DCM
   correspondiente al isoalign, tome un corte horizontal y otro vertical
   de las regiones donde interesa y en esos cortes sea donde se va a
   realizar la deteccion de patrones circulares mediante el metodo
   HoughCircles. Es probable que en alguno de los cortes resulte
   necesario aplicar una modificacion del brillo y el contraste a la
   imagen. Una vez se hayan encontrado de forma satisfactoria los
   circulos detectados, se proceder a realizar un ajuste lineal en
   ambas tuplas de puntos (horizontal y vertical), con el objetivo de
   obtener dos rectas que al intersectar nos den el punto que buscamos.
4
5 # PASO 0 - PREAMBULO:
6
7 # Comenzamos importando todas las librerias y paquetes que necesitemos
   usar en el script:
8
9 import cv2
10 import numpy as np
11 import pydicom
12 from PIL import Image
13 import sys
14
15 # sys nos va a ayudar en el proceso de debugging, saliendo de la
   ejecucion en la linea de codigo que queramos aplicando el metodo sys.
   exit().
16
17 import pandas as pd
18 import csv
19 from sklearn.linear_model import LinearRegression
20 import matplotlib.pyplot as plt
21
22 # Usaremos plt para el caso en que representemos la recta de mejor
   ajuste obtenida mediante la regresion lineal en LinearRegression.
23
24

```

```

25 # PASO 1 - IMPORTAR IMAGEN DESDE ARCHIVO DICOM:
26
27 # Vamos a extraer y guardar en una matriz de pixeles la imagen desde el
    archivo de entrada (.dcm):
28
29 isoalign = pydicom.dcmread('isoalign.dcm') # Comenzamos leyendo el
    archivo dcm.
30 matriz = isoalign.pixel_array # Cargamos la imagen como una matriz de
    datos para comprobar si los elementos de dicha matriz son FLOATS o,
    por el contrario, ENTEROS (INTEGERS).
31 cv2.imwrite("Isoalign.png",matriz)
32 #print(matriz.shape) # Tama o de la matriz. Sale (1024,1024).
33 #print(matriz) #Aqui comprobamos que la matriz que se me devuelve es de
    16 bytes (2^16), ya que en la matriz los elementos pasan del valor
    60000.
34
35 # PASO 2 - RECORTE DE LA MATRIZ EN REGIONES DE INTERES (HORIZONTAL Y
    VERTICAL):
36
37 # Horizontal:
38
39 nfila_h_i = 494
40 nfila_h_f = 524
41 ncol_h_i = 0
42 ncol_h_f = 1024
43 matrizh = matriz[nfila_h_i:nfila_h_f , ncol_h_i:ncol_h_f]
44 cv2.imwrite("out_h.png",matrizh)
45
46 #Vertical:
47
48 nfila_v_i = 0
49 nfila_v_f = 1024
50 ncol_v_i = 494
51 ncol_v_f = 524
52 matrizv = matriz[nfila_v_i:nfila_v_f , ncol_v_i:ncol_v_f]
53 cv2.imwrite("out_v.png",matrizv)
54
55 # PASO 3 - DETECCION DE PATRONES CIRCULARES MEDIANTE LA TRANSFORMADA DE
    HOUGH:

```

```

56
57 # Es en este punto en el que vamos a definir una funcion que lleve a
    cabo todo el nucleo del trabajo (identificar la imagen, aplicar
    contraste, detectar los patrones circulares...). No obstante, antes
    definimos la funcion con la que llevaremos a cabo el modelo de
    regresion lineal:
58
59 def linearreg(x,y):
60     # Para el modelo de regresion lineal necesitamos que x este
    redimensionado:
61     X = np.array(x).reshape((-1,1))
62     Y = np.array(y)
63     Xarray = np.array(x) # Dejamos este array sin hacer reshape para
    dibujar la linea.
64     model = LinearRegression().fit(X,Y)
65
66     # Con el modelo ya invocado y los puntos desglosados ya podemos
    obtener los resultados (pendiente, ordenada en el origen y
    coeficiente de correlacion) del ajuste lineal:
67
68     # Coeficiente de correlacion (R^2):
69
70     r_sq = model.score(X,Y)
71     print(f"Coeficiente de correlacion: {r_sq}")
72
73     # Pendiente (m) y ordenada en el origen (n):
74
75     print(f"Pendiente: {model.coef_}")
76     m = model.coef_
77
78     print(f"Ordenada en el origen: {model.intercept_}")
79     n = model.intercept_
80
81     # Ahora que tenemos todos los datos de la regresion lineal pasamos a
    dibujar como ha quedado el ajuste:
82     # Esto solo esta para comprobar que esta todo bien. No es necesario
    representarlo.
83     xinic= Xarray.min(axis=0)
84     xfin = Xarray.max(axis=0)

```

```

85     yinic = Y.min(axis=0)
86     yfin = Y.min(axis = 0)
87
88     scatter = plt.scatter(X,Y, color="blue")
89     plt.plot(list(range(xinic,xfin)), [m*X+n for X in range(xinic,xfin)
90 ], color="red")
91     plt.show()
92     cv2.waitKey(0)
93     cv2.destroyAllWindows()
94     return m, n
95
96 def Hough (fichero):
97
98     im = cv2.imread(fichero)
99     output = im.copy()
100
101     # Para poder diferenciar entre el recorte vertical y el horizontal
102     vamos a usar como primer criterio las dimensiones de los recortes.
103     Hemos forzado a que el recorte horizontal tenga como dimension
104     (30,1024) y el vertical (1024,30), de modo que esto nos resultara de
105     utilidad para que la funcion consiga distinguirlas:
106
107     nfils = output.shape[0]
108     ncols = output.shape[1]
109
110     if nfils <= 30: # Corte horizontal
111         dp = 2.99
112         minDist = 41
113         param1 = 50
114         param2 = 16
115         minRadius = 1
116         maxRadius = 2
117
118     else: # Corte vertical
119
120         # En el caso del corte vertical es necesario aplicar una
121         modificacion al brillo y contraste del recorte para mejorar la
122         deteccion de patrones circulares. Definimos los valores de CONTRASTE
123         (alpha) y BRILLO (beta):

```

```

116
117     alpha = 1.21
118     beta = -99
119
120     # Aplicamos el metodo convertScaleAbs, que va a escalar, tomar
valores absolutos y convertir la imagen en tipo 8-bit:
121
122     output = cv2.convertScaleAbs(output,alpha = alpha, beta = beta)
123
124     dp = 12
125     minDist = 20
126     param1 = 50
127     param2 = 12
128     minRadius = 1
129     maxRadius = 2
130
131     #Los parametros anteriores son propios del metodo HoughCircles, que
es el que vamos a usar para detectar dichos patrones circulares en
los recortes.
132
133     # //////////// BREVE EXPLICACION DEL METODO HOUGHCIRCLES ////////////:
134
135     # De forma general, el metodo HoughCircles toma los siguientes
parametros:
136
137     # cv2.HoughCircles(image, method, dp, minDist, param1, param2,
minRadius,maxRadius)
138
139     # Desarrollamos la explicacion de cada parametro por separado:
140
141     # image: Imagen de un unico canal en 8 bit. Si la imagen original es
a color hay que hacer un paso previo a escala de grises.
142
143     # method: Metodo que se va a utilizar para detectar patrones
circulares en las imagenes. Actualmente el unico metodo implementado
es cv2.HOUGH_GRADIENT.
144
145     # dp: Relacion (ratio) inversa entre la resolucion del acumulador y
la resolucion de la imagen. Basicamente, cuanto mayor sea dp, menor

```

```

sera la matriz del acumulador.
146
147 # minDist: Distancia minima entre el centro de los circulos
detecados (x,y). Si minDist es muy baja se pueden detectar muchos
circulos espurios; si es muy grande, demasiado pocos.
148
149 # param1: Valor del gradeiente usado para llevar a cabo la deteccion
de bordes.
150
151 # param2: Valor umbral para el acumulador en el metodo cv2.
HOUGH_GRADIENT. Cuanto menor sea el umbral, mayor sera la cantidad de
circulos detectados (incluyendo circulos espurios); cuanto mayor,
mas circulos se devolveran potencialmente.
152
153 #min y maxRadius: fijan el tama o minimo y maximo, respectivamente,
del radio en pixeles.
154
155 #////////////////////////////////////
156
157 # HoughCircles solo puede tratar con imagenes en escalas de grises,
de modo que hacemos un poco de tratamiento de imagen:
158
159 gray = cv2.cvtColor(output, cv2.COLOR_BGR2GRAY)
160 gray = cv2.medianBlur(gray,5)
161
162 # Con la imagen pasada a escala de grises, aplicamos el metodo para
hallar los circulos en cada caso:
163
164 circles = cv2.HoughCircles(gray, cv2.HOUGH_GRADIENT, dp, minDist,
165                             param1, param2, minRadius, maxRadius)
166
167 # En circles se nos devuelve la posicion del centro y el radio (x,y,
r) de cada circulo detectado en una tupla. No obstante, dichos
circulos no vienen como enteros, de modo que redondeamos los
decimales y pasamos a enteros:
168
169 detected_circles = np.uint16(np.around(circles))
170
171 # Una vez obtenemos las tuplas de circulos detectados para cada caso

```

```

, pasamos a crear nuestras propias tuplas extrayendo solo la posicion
de los centros de dichos circulos:
172
173 # Como estamos trabajando con una tupla de tuplas (es decir, una
lista que va de 3 en 3 coordenadas (x,y,r)), para hallar las
posiciones x e y de cada caso por separado tenemos que realizar una
iteracion, comenzando por el primer elemento dentro del primer
elemento de la lista y asi. Entonces:
174
175 x_center = [row[0] for row in detected_circles[0]]
176 y_center = [row[1] for row in detected_circles[0]]
177
178 # Aunamos esas coordenadas en una unica matriz, para lo cual
tendremos que hacer una iteracion:
179
180 xy = [i for i in zip(x_center,y_center)]
181
182 # Obtenemos finalmente una tupla con solo las coordenadas x,y
correspondiente a los centros de los circulos detectados:
183
184 XY =[]
185 for (x,y,r) in detected_circles[0,:]:
186     if r<=5: #Aqui hay que filtrar por radio, a pesar de que en
HoughCircles este se delimito con maxRadius.
187         xy = (x,y)
188         XY.append(xy)
189         cv2.circle((output),(x,y),r,(255,0,0),2) # Dibujamos el
contorno de los circulos detectados.
190         cv2.circle((output),(x,y),1,(255,255,0),2) # Dibujamos el
centro de los circulos detectados.
191
192 print("Coordenadas de los centros detectados: \n",XY)
193 cv2.imshow("Circulos detectados",output)
194 cv2.waitKey(0)
195
196 # Pasamos ahora a realizar el ajuste lineal y a representar todo en
cada recorte y en la imagen general. Comenzamos separando de XY los
valores de x y de y en dos matrices:
197

```

```

198
199     equis = []
200     ygriega = []
201
202     for (x,y) in XY:
203         xdots = x
204         ydots = y
205         equis.append(xdots)
206         ygriega.append(ydots)
207
208     # Una vez hecho esto vamos a hallar el valor medio y la desviacion
    estandar (sigma) de los puntos x para deducir si es o no necesario
    aplicar la regresion lineal:
209
210     xmedia = np.mean(equis)
211     sigma = np.std(equis)
212
213     # Ahora dividimos sigma entre el valor medio de x:
214
215     ratio = sigma/xmedia
216
217     print("Media de los x:", xmedia)
218     print("Desviacion estandar en x:",sigma)
219     print("Relacion entre la desviacion estandar y el valor medio:",
    ratio)
220
221     # Con la relacion entre el valor medio y la desviacion estandar
    podemos volver a segregar la funcion para saber si se precisa o no de
    ajuste lineal:
222
223     if ratio != 0:
224         m,n = linearreg(equis,ygriega)
225     else:
226         m = 0
227         n = 0
228     return m, n, detected_circles ,x_center ,y_center , ratio
229
230 # Representamos ya todo en la imagen general del Iso-Align:
231

```

```

232 mh,nh,detected_circles_h,x_centerh,y_centerh,ratioh = Hough("out_h.png")
233 mv,nv,detected_circles_v,x_centerv,y_centerv,ratiov = Hough("out_v.png")
234
235 #Representamos todo en la imagen del Iso-Align:
236 image = cv2.imread("Isoalign.png")
237
238 # Lineas:
239
240 # Recta horizontal:
241
242 yfin_h = mh*1024 + nh
243 yfinal_h = int(np.around(yfin_h))
244 yinicial_h = int(np.around(nh+nfila_h_i))
245 start_point_h = (0,yinicial_h)
246 end_point_h = (1023,yfinal_h + nfila_h_i)
247 linearh = cv2.line(image,start_point_h,end_point_h,(0,0,255),1)
248
249 #Recta vertical:
250
251 # Aqui volvemos a hacer un desglose por si se da el caso de que la recta
    obtenida no es puramente vertical. De no serlo el cociente entre
    sigma y el valor medio (ratiov) sera distinto de cero, de modo que
    necesitamos hacer uso de la ecuacion explicita de la recta para
    hallar un punto inicial y final de la misma, ya que en este caso
    tendremos un valor de la pendiente y la ordenada en el origen para la
    recta vertical bien definido ( $y = mv*x + nv$ ). Para hallar el punto
    de corte en este caso se deben igualar las dos ecuaciones de la recta
    , hallar la x del corte y luego sustituir en alguna de las dos
    ecuaciones de la recta para hallar la y del corte.
252
253 # No obstante, si tenemos una recta puramente vertical (pendiente
    infinita) encontramos que la coordenada x del punto de corte es
    cualquiera de dicha recta. Entonces, para sacar la coordenada y
    sustituimos dicho punto en la ecuacion de la recta horizontal.
254
255 if ratiov !=0:
256     xinic_v = -nv/mv # Esto es asi porque hemos hecho  $y = 0$ 
257     xfin_v = -nv/mv + 1023/m # Aqui  $y = 1023$ 
258     start_point_v = (xinic_v,0)

```

```

259     end_point_v = (xfin_v,1023)
260     # Igualamos ecuaciones y sacamos x e y del corte:
261     ndiff = nh-nv
262     mdiff = mv - mh
263     xcorte =ndiff/mdiff
264     ycorte = mv*xcorte + nv + ncol_v_i
265
266 else:
267     start_point_v = (x_centerv[0] + ncol_v_i,0)
268     end_point_v = (x_centerv[0] + ncol_v_i, 1023)
269     linearv = cv2.line(image,start_point_v,end_point_v,(0,0,255),1)
270     xcorte = start_point_v[0]
271     yc = mh*xcorte+nh
272     ycorte = int(np.around(yc+nfila_h_i)) #Sumamos el nfila_h_i para
referenciar de donde hemos realizado el corte.
273
274 # Circulos:
275 for (x, y, r) in detected_circles_h[0,:]:
276     if r<=5:
277         cv2.circle((image),(x,y+nfila_h_i),r, (255,0,0),2) # Contorno.
278         cv2.circle((image),(x,y+nfila_h_i),1, (255,255,0),3) # Centro.
279
280 for (x, y, r) in detected_circles_v[0,:]:
281     if r<=6:
282         cv2.circle((image),(x+ncol_v_i,y),r, (255,0,0),2) # Contorno.
283         cv2.circle((image),(x+ncol_v_i,y),1, (255,255,0),3) # Centro.
284
285 cv2.imshow("Isoalign alineado",image)
286 cv2.waitKey(0)
287
288 # PARTE FINAL - PUNTOS DE LAS RECTAS Y PUNTO DE INTERSECCION:
289
290 print("Puntos de la recta horizontal:\n",start_point_h,end_point_h)
291 print("Puntos de la recta vertical:\n",start_point_v,end_point_v)
292
293 pcorte = (xcorte,ycorte)
294 print("Punto de corte entre las rectas horizontal y vertical:\n",pcorte)

```

Listing A.2: Algoritmo basado en la transformada de Hough para automatizar la detección de patrones circulares en las imágenes de Iso-Align.

Apéndice B

Imágenes del Iso-Align TM analizadas

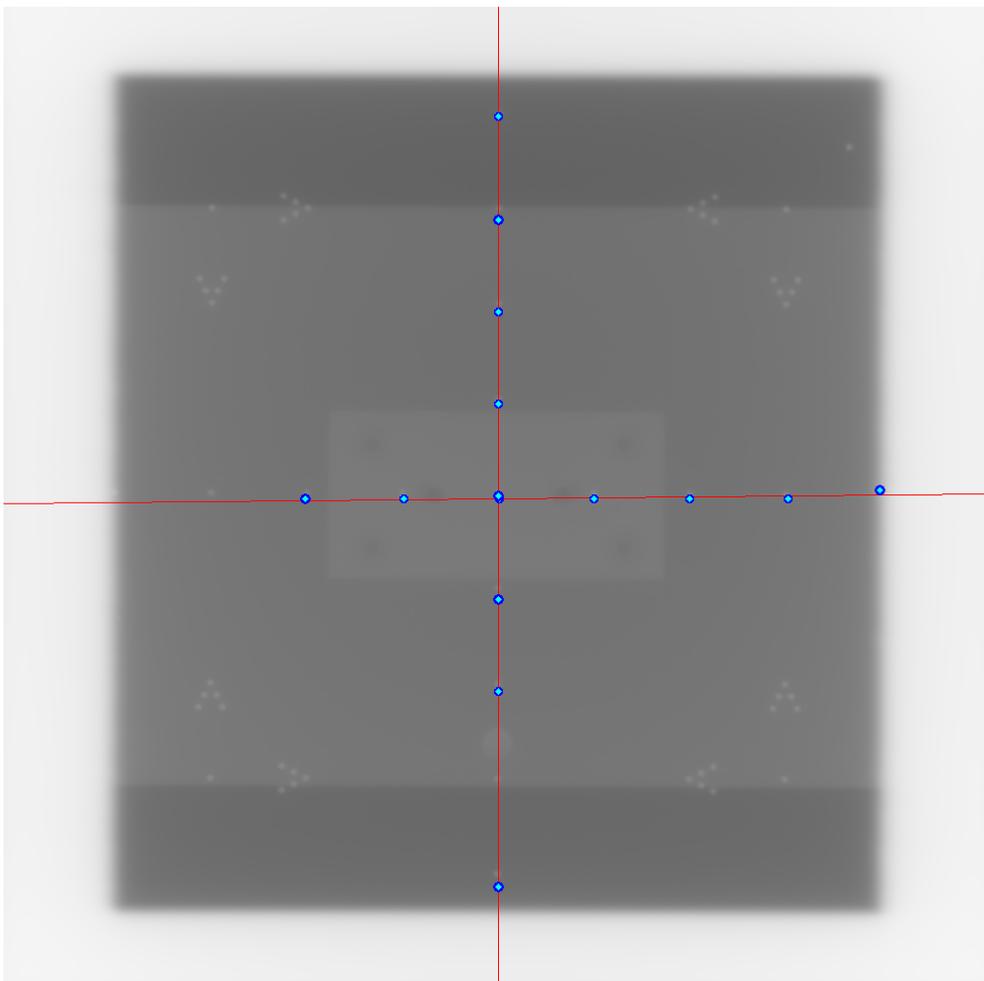


Figura B.1: Imagen del Iso-Align analizada. Muestra del 03/02/2023.

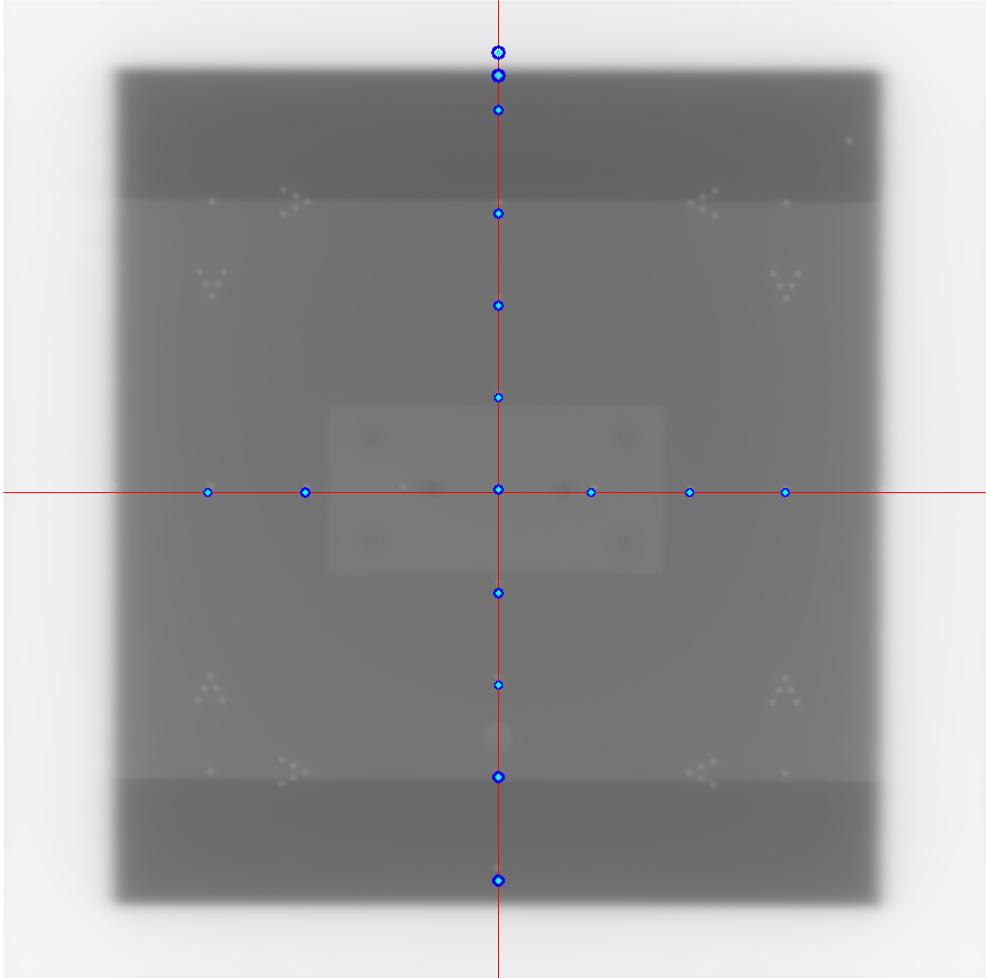


Figura B.2: Imagen del Iso-Aling analizada. Muestra del 08/03/2023.

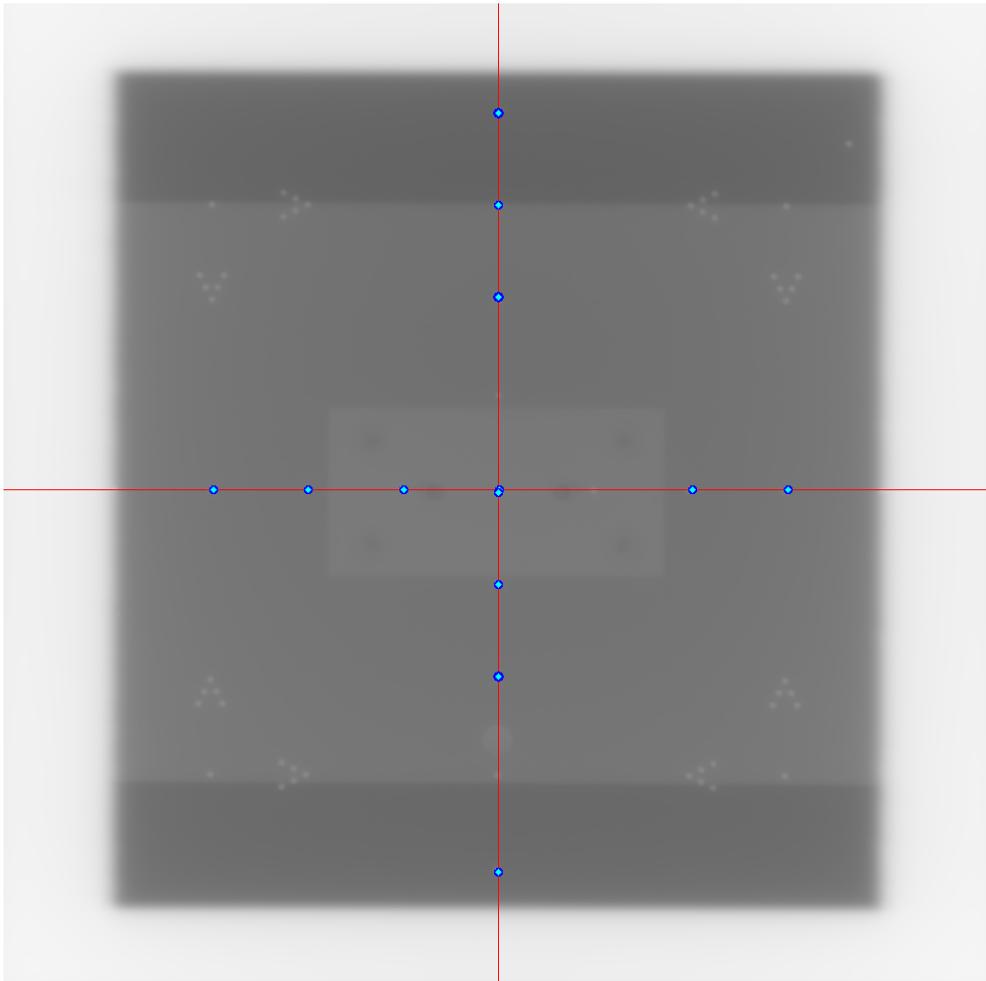


Figura B.3: Imagen del Iso-Aling analizada. Muestra del 02/06/2023.

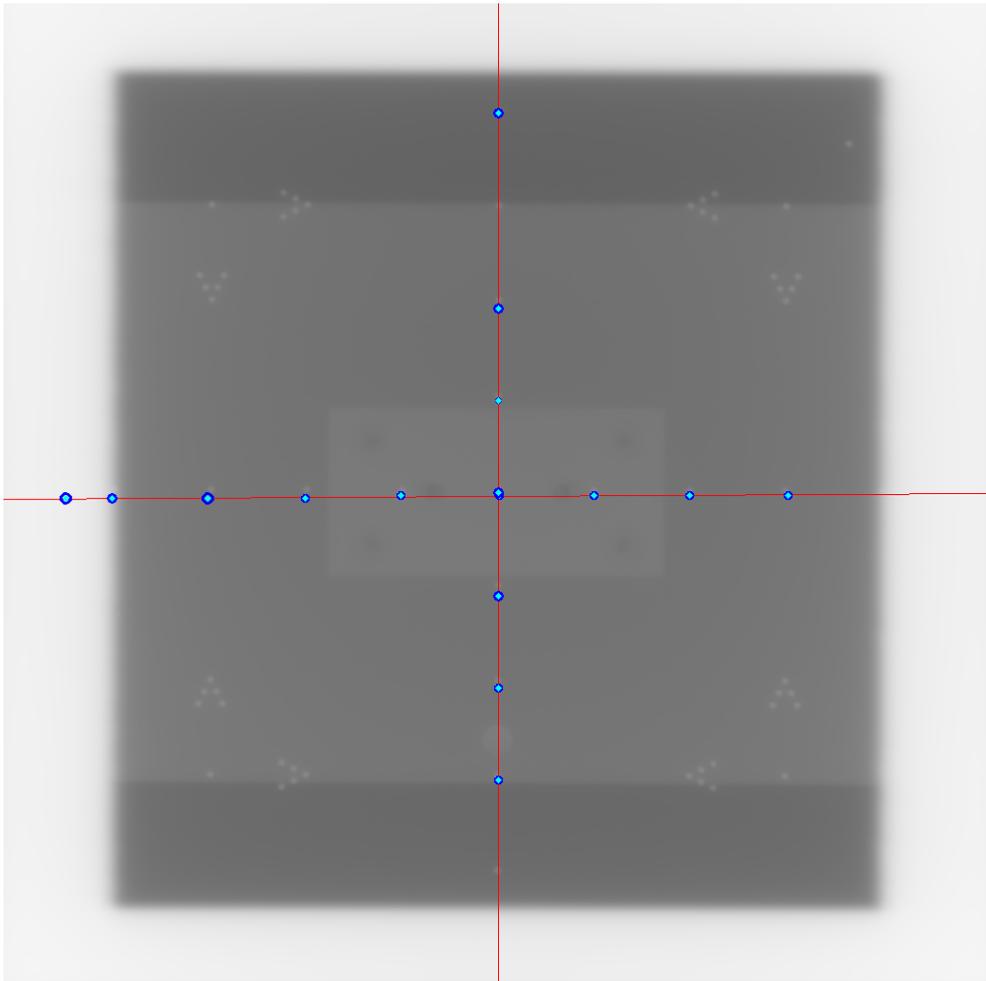


Figura B.4: Imagen del Iso-Aling analizada. Muestra del 21/06/2023.

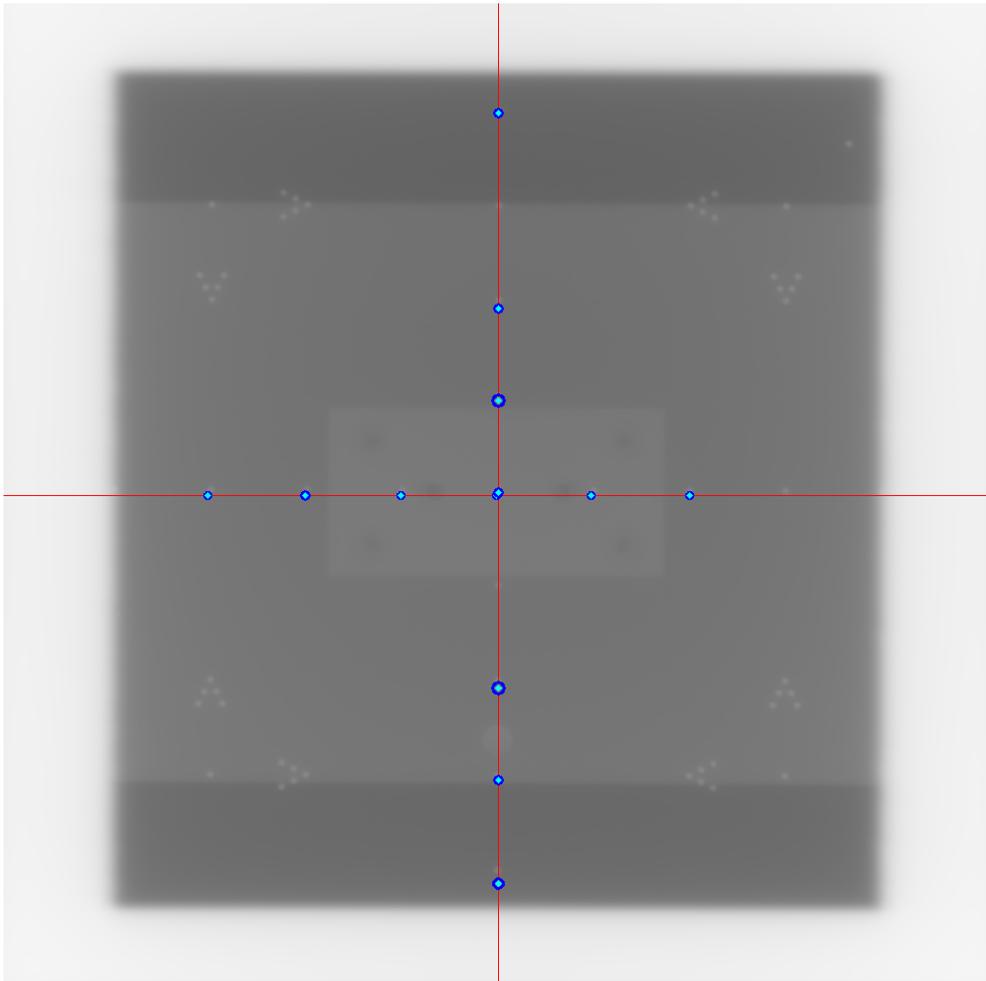


Figura B.5: Imagen del Iso-Aling analizada. Muestra del 23/02/2021.

Apéndice C

Resultados del análisis de imágenes de Iso-Align

| 02/06/2023 | X_o (px) | Y_o (px) | X_a (px) | Y_a (px) | t_o (s) | t_a (s) |
|-----------------|--------------|--------------|------------|------------|-----------|-------------|
| Intento 1 | 514,1 | 508,2 | 512 | 507 | 65 | 0,45 |
| Intento 2 | 513,3 | 507,8 | 512 | 507 | 75 | 0,42 |
| Intento 3 | 514,6 | 507,8 | 512 | 507 | 72 | 0,44 |
| Intento 4 | 514,1 | 507,3 | 512 | 507 | 88 | 0,42 |
| Intento 5 | 513,3 | 508,0 | 512 | 507 | 88 | 0,37 |
| Intento 6 | 513,7 | 507,6 | 512 | 507 | 97 | 0,37 |
| Intento 7 | 513,4 | 507,3 | 512 | 507 | 92 | 0,38 |
| Intento 8 | 513,3 | 508,0 | 512 | 507 | 96 | 0,43 |
| Intento 9 | 513,7 | 507,8 | 512 | 507 | 79 | 0,37 |
| Intento 10 | 513,7 | 508,0 | 512 | 507 | 89 | 0,36 |
| MEDIA | 513,7 | 507,8 | 512 | 507 | 84 | 0,40 |
| σ_x (px) | 0,41 | | | | | |
| σ_y (px) | 0,29 | | | | | |

Tabla C.1: Comparativa entre los resultados de la metodología original y la actual para la obtención del centro en la imagen del Iso-Align del 02/06/2023. El subíndice «o» hace referencia a los datos obtenidos por la metodología original, mientras que «a» corresponde a la metodología desarrollada en este trabajo.

| | |
|---------------------------------|------|
| $\max(\Delta X)$ (px) | 2,60 |
| $\max(\Delta Y)$ (px) | 1,20 |
| $\min(\Delta X)$ (px) | 1,30 |
| $\min(\Delta Y)$ (px) | 0,30 |
| $\langle \Delta X \rangle$ (px) | 1,72 |
| $\langle \Delta Y \rangle$ (px) | 0,78 |

Tabla C.2: Diferencias absolutas entre las coordenadas X e Y del punto de corte de los láseres encontradas por ambas metodologías para la imagen del Iso-Align del 02/06/2023. Se contemplan los casos de diferencia máxima, mínima y promedio.

| 23/02/2021 | X_o (px) | Y_o (px) | X_a (px) | Y_a (px) | t_o (s) | t_a (s) |
|-----------------|--------------|--------------|------------|------------|-----------|-------------|
| Intento 1 | 509,8 | 515,0 | 512 | 513 | 96 | 0,48 |
| Intento 2 | 510,0 | 514,1 | 512 | 513 | 82 | 0,35 |
| Intento 3 | 510,6 | 514,9 | 512 | 513 | 108 | 0,35 |
| Intento 4 | 509,8 | 514,7 | 512 | 513 | 108 | 0,35 |
| Intento 5 | 510,0 | 514,7 | 512 | 513 | 94 | 0,37 |
| Intento 6 | 510,0 | 514,1 | 512 | 513 | 71 | 0,37 |
| Intento 7 | 510,2 | 514,1 | 512 | 513 | 79 | 0,36 |
| Intento 8 | 509,6 | 513,0 | 512 | 513 | 83 | 0,36 |
| Intento 9 | 510,4 | 515,6 | 512 | 513 | 97 | 0,42 |
| Intento 10 | 509,8 | 514,7 | 512 | 513 | 72 | 0,37 |
| MEDIA | 510,0 | 514,5 | 512 | 513 | 89 | 0,38 |
| σ_x (px) | 0,29 | | | | | |
| σ_y (px) | 0,67 | | | | | |

Tabla C.3: Comparativa entre los resultados de la metodología original y la actual para la obtención del centro en la imagen del Iso-Align del 23/02/2021. El subíndice «o» hace referencia a los datos obtenidos por la metodología original, mientras que «a» corresponde a la metodología desarrollada en este trabajo.

| | |
|---------------------------------|------|
| $\max(\Delta X)$ (px) | 2,40 |
| $\max(\Delta Y)$ (px) | 2,60 |
| $\min(\Delta X)$ (px) | 1,40 |
| $\min(\Delta Y)$ (px) | 0,0 |
| $\langle \Delta X \rangle$ (px) | 1,98 |
| $\langle \Delta Y \rangle$ (px) | 1,49 |

Tabla C.4: Diferencias absolutas entre las coordenadas X e Y del punto de corte de los láseres encontradas por ambas metodologías para la imagen del Iso-Align del 23/02/2021. Se contemplan los casos de diferencia máxima, mínima y promedio.

| 03/02/2023 | X_o (px) | Y_o (px) | X_a (px) | Y_a (px) | t_o (s) | t_a (s) |
|-----------------|--------------|--------------|------------|------------|-----------|-------------|
| Intento 1 | 512,7 | 514,1 | 512 | 513 | 82 | 0,43 |
| Intento 2 | 513,0 | 515,6 | 512 | 513 | 67 | 0,38 |
| Intento 3 | 512,4 | 514,4 | 512 | 513 | 101 | 0,36 |
| Intento 4 | 512,9 | 514,4 | 512 | 513 | 82 | 0,38 |
| Intento 5 | 512,4 | 515,3 | 512 | 513 | 77 | 0,43 |
| Intento 6 | 512,1 | 515,3 | 512 | 513 | 71 | 0,37 |
| Intento 7 | 512,8 | 514,6 | 512 | 513 | 70 | 0,42 |
| Intento 8 | 513,1 | 514,7 | 512 | 513 | 61 | 0,36 |
| Intento 9 | 513,0 | 513,7 | 512 | 513 | 72 | 0,36 |
| Intento 10 | 512,8 | 514,1 | 512 | 513 | 70 | 0,38 |
| MEDIA | 512,7 | 514,6 | 512 | 513 | 75 | 0,39 |
| σ_x (px) | 0,30 | | | | | |
| σ_y (px) | 0,58 | | | | | |

Tabla C.5: Comparativa entre los resultados de la metodología original y la actual para la obtención del centro en la imagen del Iso-Align del 03/02/23. El subíndice «o» hace referencia a los datos obtenidos por la metodología original, mientras que «a» corresponde a la metodología desarrollada en este trabajo.

| | |
|---------------------------------|------|
| $\max(\Delta X)$ (px) | 1,10 |
| $\max(\Delta Y)$ (px) | 2,60 |
| $\min(\Delta X)$ (px) | 0,10 |
| $\min(\Delta Y)$ (px) | 0,70 |
| $\langle \Delta X \rangle$ (px) | 0,72 |
| $\langle \Delta Y \rangle$ (px) | 1,62 |

Tabla C.6: Diferencias absolutas entre las coordenadas X e Y del punto de corte de los láseres encontradas por ambas metodologías para la imagen del Iso-Align del 03/02/23. Se contemplan los casos de diferencia máxima, mínima y promedio.

| 21/06/2023 | X_o (px) | Y_o (px) | X_a (px) | Y_a (px) | t_o (s) | t_a (s) |
|-----------------|--------------|--------------|------------|------------|-----------|-------------|
| Intento 1 | 511,6 | 514,8 | 512 | 514 | 81 | 0,41 |
| Intento 2 | 512,1 | 513,7 | 512 | 514 | 78 | 0,36 |
| Intento 3 | 511,6 | 514,2 | 512 | 514 | 76 | 0,36 |
| Intento 4 | 512,1 | 513,9 | 512 | 514 | 71 | 0,36 |
| Intento 5 | 511,7 | 515,2 | 512 | 514 | 87 | 0,39 |
| Intento 6 | 511,4 | 513,2 | 512 | 514 | 80 | 0,37 |
| Intento 7 | 512,2 | 513,3 | 512 | 514 | 85 | 0,36 |
| Intento 8 | 511,4 | 513,8 | 512 | 514 | 81 | 0,36 |
| Intento 9 | 511,7 | 513,6 | 512 | 514 | 77 | 0,37 |
| Intento 10 | 511,8 | 514,3 | 512 | 514 | 79 | 0,37 |
| MEDIA | 511,8 | 514,0 | 512 | 514 | 80 | 0,37 |
| σ_x (px) | 0,28 | | | | | |
| σ_y (px) | 0,60 | | | | | |

Tabla C.7: Comparativa entre los resultados de la metodología original y la actual para la obtención del centro en la imagen del Iso-Align del 21/06/2023. El subíndice «o» hace referencia a los datos obtenidos por la metodología original, mientras que «a» corresponde a la metodología desarrollada en este trabajo.

| | |
|---------------------------------|------|
| $\max(\Delta X)$ (px) | 0,60 |
| $\max(\Delta Y)$ (px) | 1,20 |
| $\min(\Delta X)$ (px) | 0,10 |
| $\min(\Delta Y)$ (px) | 0,10 |
| $\langle \Delta X \rangle$ (px) | 0,24 |
| $\langle \Delta Y \rangle$ (px) | 0,0 |

Tabla C.8: Diferencias absolutas entre las coordenadas X e Y del punto de corte de los láseres encontradas por ambas metodologías para la imagen del Iso-Align del 21/06/2023. Se contemplan los casos de diferencia máxima, mínima y promedio.