

Trabajo Fin de Grado

Grado en Ingeniería de las Tecnologías Industriales

Medición con sistemas embarcados de la calidad geométrica de vías ferroviarias: determinación de la posición del carril mediante visión artificial y optimización con restricciones

Autor: Jorge Galán Cáceres

Tutor: Dr. Juan S. Valverde García

Dpto. Matemática Aplicada II
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla

Sevilla, 2020



Trabajo Fin de Grado
Grado en Ingeniería de las Tecnologías Industriales

**Medición con sistemas embarcados de la calidad
geométrica de vías ferroviarias: determinación
de la posición del carril mediante visión
artificial y optimización con restricciones**

Autor:

Jorge Galán Cáceres

Tutor:

Dr. Juan S. Valverde García

Dpto. Matemática Aplicada II
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla

Sevilla, 2020

Trabajo Fin de Grado: Medición con sistemas embarcados de la calidad geométrica de vías ferroviarias: determinación de la posición del carril mediante visión artificial y optimización con restricciones

Autor: Jorge Galán Cáceres
Tutor: Dr. Juan S. Valverde García

El tribunal nombrado para juzgar el trabajo arriba indicado, compuesto por los siguientes profesores:

Presidente:

Vocal/es:

Secretario:

acuerdan otorgarle la calificación de:

El Secretario del Tribunal

Fecha:

Agradecimientos

Agradezco el trabajo realizado por mi tutor Juan Valverde García y mi cotutor Eduardo Briaes Palacios, de la empresa Virtualmechanichs S.L., que me ha permitido poder completar este Trabajo Fin de Grado. Muchos de los desarrollor y algoritmos usados en el presente trabajo están inspirados en la tesis de Eduardo [1].

Doy las gracias a mi familia, por acompañarme en este camino. Sin ellos todo hubiera sido más complicado.

Resumen

El objetivo de este Trabajo Fin de Grado ha sido desarrollar un algoritmo de optimización para determinar la posición del carril de una vía en el mundo real a partir de las medidas tomadas por las cámaras de un sistema embarcado en un tren.

Comenzamos con una presentación general del proyecto en el que se engloba esta tarea. Hacemos también un breve repaso al estado del arte y a la normativa que debe cumplir un proyecto de este estilo.

Para resolver la tarea hemos desarrollado la cinemática de seguimiento del movimiento de una cámara con la cual podremos obtener la posición en 3D de un punto del carril que aparezca en la imagen en 2D tomada por la cámara. Para poder llevar a cabo dicha transformación, hemos implementado la calibración para las cámaras instaladas. Ambos desarrollos teóricos se explican detalladamente en el documento.

Además, hemos modelado matemáticamente el perfil de la sección de un carril. Este perfil teórico es el que usan nuestros algoritmos de optimización para resolver el problema.

El problema propuesto es de tipo matemático y combina aspectos geométricos, computacionales y de optimización. Para resolver este problema, hemos desarrollado tres algoritmos que nos proporcionan. Explicamos en qué se basa cada uno y cómo los hemos implementado en Python.

Por último hemos llevado a cabo una serie de experimentos para poder evaluar nuestros algoritmos. Estos los hemos hecho con un banco de pruebas de medidas generadas artificialmente y con datos reales de mediciones realizadas en trenes.

Por último, hacemos un análisis de los resultados obtenidos y debatimos sobre los posibles cambios y mejoras que podríamos implementar en los algoritmos propuestos.

Índice

<i>Resumen</i>	III
1 Introducción	1
1.1 Historia	1
1.2 Motivación	3
1.3 Estado del arte	5
1.4 El proyecto vmRail	5
1.4.1 Introducción	5
1.4.2 Características generales de vmRail	6
1.4.3 Especificaciones de vmRail	11
2 Modelado matemático del problema	17
2.1 Cinemática del seguimiento de movimiento usando visión por computador	17
2.1.1 Cinemática de la cámara	17
2.1.2 Coordenadas homogéneas de un vector	20
2.1.3 Parámetros intrínsecos de la cámara	20
2.1.4 Parámetros extrínsecos de la cámara	21
2.1.5 Relación entre puntos de la imagen y puntos del mundo	22
2.1.6 Seguimiento de movimiento usando visión por computador	23
2.2 Calibración de las cámaras	24
2.2.1 Calibración de cámaras usando el método de Zhang	24
Búsqueda de las matrices de homografía	27
Búsqueda de parámetros intrínsecos	29
Búsqueda de parámetros extrínsecos	31
Optimización de los parámetros intrínsecos y extrínsecos	32
Procedimiento de calibración antes de las inspecciones	34
2.3 Modelado del perfil del raíl	35
2.3.1 Función trazadora de arcos	36
2.3.2 Función trazadora de rectas	37
2.3.3 Tramos UIC54	37
Cabeza del raíl	37
Alma del raíl	44
Base del raíl	45
3 Algoritmos de optimización	51
3.1 Explicación teórica de los algoritmos	51
3.1.1 Algoritmo 1: Optimización con dos nubes de puntos	53
3.1.2 Algoritmo 2: ICP	53
3.1.3 Algoritmo 3: Optimización con función explícita	55
3.2 Pruebas de los algoritmos	58
3.2.1 Resultados con banco de pruebas	58

Experimento 1. Medidas de perfil completo sin ruido	60
Experimento 2. Medidas de perfil completo con ruido de pequeña amplitud	62
Experimento 3. Medidas de perfil completo con ruido de gran amplitud	64
Experimento 4. Medidas de perfil incompleto sin ruido	66
Experimento 5. Medidas de perfil incompleto con ruido de pequeña amplitud	68
Experimento 6. Medidas de perfil incompleto con ruido de gran amplitud	70
Experimento 7. Influencia de la condición inicial	72
3.2.2 Resultados con datos reales	76
Experimento 1. Análisis en punto kilométrico 1	77
Experimento 2 . Análisis en punto kilométrico 2	80
Experimento 3. Análisis de un tramo de viaje	83
4 Conclusiones y futuras mejoras	85
4.1 Conclusiones	85
4.2 Futuras mejoras	85
<i>Índice de Figuras</i>	87
<i>Índice de Tablas</i>	89
<i>Bibliografía</i>	91

1 Introducción

1.1 Historia

La aparición de las redes ferroviarias en Europa cambió drásticamente la vida a sus habitantes [2]. Antes de su invención, las personas dependían de otros medios de transporte como los caballos y los carros que no satisfacían las necesidades reales de las personas. Los viajes eran largos, pesados y peligrosos, y el envío de mercancías demasiado lento y costoso. Era común que algunos productos no llegaran a determinadas zonas del mundo porque no salía rentable económicamente o porque no se podía garantizar que fueran a llegar en buen estado a su destino. Existían sistemas de raíles para vagones en las minas e incluso existían carros manuales que podían aprovechar dichas vías, pero no fue hasta la invención del motor de vapor que los trenes comenzaron a ser una opción viable y provechosa para la población.

Es por eso que muchos expertos defienden que el motor de vapor fue uno de los cambios necesarios para que se produjera la Revolución Industrial. Aparte de lo que supuso en términos de producción industrial, a partir de ese momento podían transportar veloz y seguramente lo producido, además de maximizar beneficios, lo cual trajo una época de prosperidad económica para los grandes empresarios.

Aunque en sus primeros años de vida no fue un medio utilizado por las clases bajas, no tardó mucho en extenderse. Tanto que se convirtió en el medio de transporte usado por una gran parte de la población para ir al trabajo. Esto produjo un cambio en la concepción y la perspectiva del tiempo y la distancia. Reducción en los desplazamientos laborales, menor tiempo de espera para los envíos, abastecimiento de productos que de otra manera nunca hubieran llegado a determinados lugares...En definitiva, la implantación de un sistema ferroviario fue un aspecto diferencial a la hora de analizar el desarrollo e impacto de la Revolución Industrial en la vida del pueblo.

La implantación de estas redes fue un proceso lento debido a la magnitud de las obras requeridas para garantizar las infraestructuras necesarias y por la gran inversión que este proyecto demandaba. La primera línea férrea abierta al uso público para transporte de viajeros y mercancías se puso en funcionamiento en 1825 en Inglaterra [2]. Dicha línea unía las ciudades de Stockton y Darlington, pero puede considerarse que el primer ferrocarril como lo conocemos hoy en día no apareció hasta 1830 cuando se inauguró la línea Liverpool-Manchester. Esta línea era la encargada de unir el puerto de Liverpool, por donde entraba el algodón proveniente de América, con la ciudad de Manchester, conocida por aquel entonces por su gran industria textil. Esto supuso un impacto económico importante para ambas ciudades, a la vez que ayudó a aligerar el tráfico de las carreteras convencionales.

Ya por aquel entonces comenzaron a instaurarse los primeros estándares de vías ferroviarias. Tanto en cuanto al tipo de instalaciones, protecciones y protocolos de comunicación como las medidas o morfologías que debían usarse para uniformizar las redes ferroviarias a nivel estatal o incluso continental. Por ejemplo, en la línea de Stockton de 1825 comentada anteriormente se decidió usar un ancho de vía de 1,435 m ya que era el utilizado en las vías mineras y así se les podía dar una segunda vida a los vagones. Para ello, George Stephenson (creador de la línea de Stockton y hombre importante en la construcción de la línea de Liverpool) construyó una locomotora adaptada a este ancho de vía. Tuvieron bastante éxito y se vendieron incluso a países extranjeros que habían decidido utilizar la misma anchura.

Sin embargo, esta medida fue pronto cuestionada y se demostró que teóricamente era preferible usar un ancho de vía mayor. Uno de los grandes ferroviarios ingleses, I. Sambard K. Brunel construyó en el año 1835 la línea Great Western con un ancho de vía de 2,10 m. A pesar de su prestigio, no muchos siguieron su iniciativa, ya que se dieron cuenta que lo más importante era unificar todas las líneas estatales para facilitar su interconexión y optimizar la fabricación de locomotoras. Aún así esta línea siguió operando con estas especificaciones hasta 1882, año en el que se cambió al ancho común de 1,435 m.

En España, la primera línea se construyó en Cuba en el año 1837 con un ancho de 1,435m. Esa misma anchura se hubiera seguido utilizando en territorio nacional si no fuera porque en 1843, debido a que se estaba discutiendo la concesión de la línea Barcelona-Mataró, el Gobierno nombró una comisión para estudiar el sistema de concesiones ferroviarias. Al año siguiente, dicha comisión presentó un informe en el que se recomendaba el uso de un ancho de vía de 1,674 m. Como se había visto en Inglaterra un mayor ancho de vía permitía construir locomotoras más potentes y emplear gálilos más grandes. Aunque también presentaba inconvenientes, como la dificultad de interconexión con otras redes, en aquel momento parecieron más importantes las ventajas. Al final resultó que las ventajas no se aprovecharon, ya que se compraron las locomotoras inglesas normales y se adoptó un gálilo un poco mayor que el estándar europeo.

A lo largo de las siguientes décadas, se produjo un gran aumento del número de redes ferroviarias por todo Europa. Bélgica, Francia o Alemania, entre otros muchos, formaron redes kilométricas y complejas interconexiones entre ellas. Esto supuso un problema en algunos países debido a dos motivos: la falta de capital y la mala planificación. Muchos países no querían endeudarse con otras potencias por lo que no pudieron construir al ritmo que deseaban. Otros, sin embargo, construyeron demasiado y para finales de siglo había algunos países que incluso comenzaron a desmontar redes debido a que eran demasiado densas y suponían costes inasumibles.

Desde entonces hasta prácticamente hoy en día, el tren ha continuado siendo un elemento fundamental en la vida de las personas y en el crecimiento económico de muchos países. No se podría entender el mundo como lo hacemos hoy en día sin este invento. Además de las principales redes de cada país, se construyeron numerosas redes secundarias que facilitaron el transporte a personas y comunicó comunidades que habían estado más aisladas debido a la dificultad de acceso a ellas.

No pasó mucho tiempo hasta que este invento se modificó para poder utilizarlo dentro de las ciudades, lo que conocemos como el metro. Este medio de transporte es uno de los más utilizados hoy en día para desplazamientos dentro de una ciudad, como puede ser el trayecto al trabajo. La importancia de los sistemas de metro es cada vez mayor y ello conlleva que numerosos estudios se realicen para mejorar los vehículos, las vías o para mejorar su mantenimiento. Aquí es donde entra el papel de la empresa Virtualmechanichs S.L., donde realicé unas prácticas en el verano de 2019, con su sistema TGMS/vmRail (Track Geometry Measurement System) y en lo que se centra este trabajo Fin de Grado: el desarrollo de un algoritmo de optimización para estimar la posición del carril en una imagen.

A lo largo de este Trabajo Fin de Grado se explicarán diversos aspectos del proyecto vmRail, de su cinemática y de los algoritmos empleados para resolver la tarea encomendada. En lo que queda de Capítulo 1, explicaremos otras soluciones que se han aplicado o se han estudiado para resolver el problema de posicionamiento de la vía, es decir, el estado del arte. También haremos un repaso general del proyecto vmRail de Virtualmechanichs S.L.

En el Capítulo 2 mostraremos los desarrollos matemáticos que han sido necesarios para resolver dos problemas de gran importancia a la hora de resolver la tarea: la calibración de las cámaras y la creación de una función que modele el perfil del carril. La calibración de las cámaras es imprescindible para poder extraer información de las imágenes captadas por nuestra cámara. Debido a que queremos conocer una posición en 3D a partir de una posición en 2D que nos da la imagen, es necesario crear un modelo de calibración y definir un procedimiento para implementarlo. Generar una función que modele el perfil de la sección del carril será imprescindible para resolver nuestro programa de optimización, puesto que lo usaremos de referencia para computar la función de coste de nuestro algoritmo, como se explicará en el Capítulo 2.

En el Capítulo 3, por su parte, mostraremos las tres soluciones propuestas, representadas por los tres algoritmos diferentes que utilizamos para estimar la posición del carril. Además, realizaremos una serie de experimentos

para testear las soluciones que ofrecen cada uno y poder determinar cuáles son las carencias y puntos fuertes de cada uno de ellos.

Por último, en el Capítulo 4, analizaremos las conclusiones que obtenemos de las pruebas realizadas en el capítulo anterior y propondremos posibles mejoras que hagan que nuestros algoritmos sean más robustos o proporcionen una mejor respuesta.

1.2 Motivación

El metro se ha convertido en uno de los medios de transporte más utilizados en las principales ciudades, además de un claro representante de la importancia de una ciudad. No es casualidad que las mayores redes de metro se encuentren en las ciudades más poderosas e influyentes del mundo: Madrid, Pekín, Londres, Moscú, Nueva York... Por ejemplo, se estima que más de 60 millones de pasajeros utilizan el metro de Madrid al año, llegando a tener picos de hasta 2,6 millones de viajeros en un solo día, como el día de Black Friday de 2018 [3].

Ranking de sistemas de metro del mundo

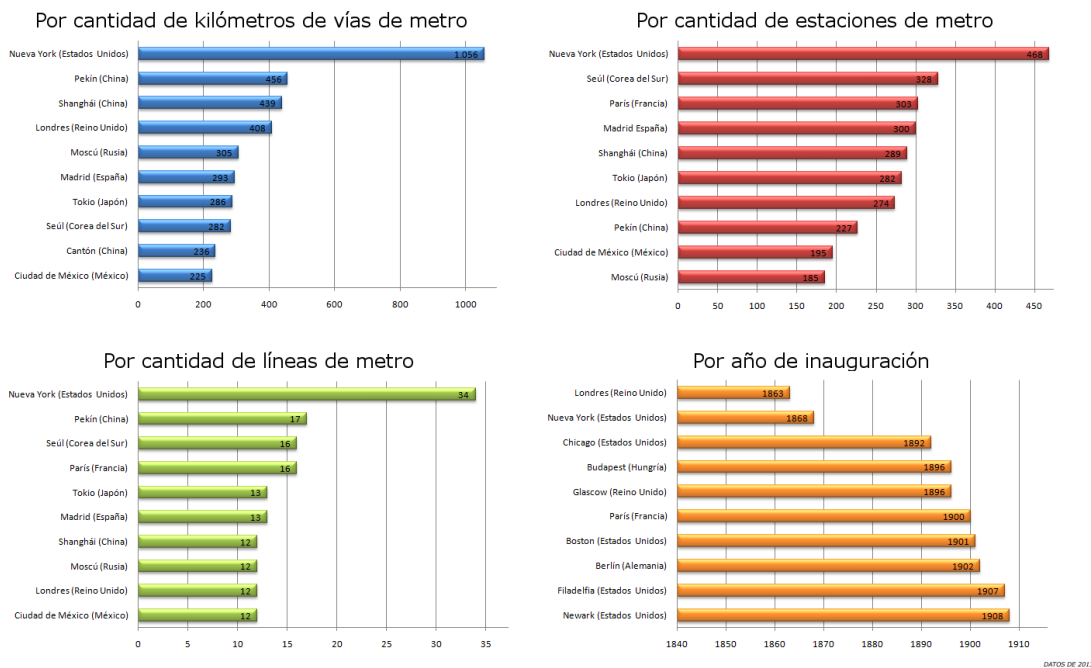


Figura 1.1 Ranking de los metros del mundo.

Mirando estas cifras y las que aparecen en la Figura 1.1 podemos hacernos una idea de lo importante que se ha vuelto este medio de transporte para comprender la vida actual de nuestra sociedad. Para ir al trabajo, para ir de compras o incluso para hacer turismo, el metro se ha convertido en un elemento fundamental de nuestras vidas y, en ocasiones, hasta en un icono de la ciudad. Es por ello que se emplean muchos recursos de los ayuntamientos para mejorar el servicio cada año y numerosos estudios se hacen para ello.

Aparte de los gastos obvios en personal, construcción de nuevas instalaciones y limpieza, una gran parte de la inversión de una ciudad en su metro se la lleva el mantenimiento de los trenes y la red vial. Este gasto se hace, además de para poder cumplir lo que dictamina la normativa UNE-EN 13848, por tres razones: disminuir el riesgo de accidente, aumentar la vida útil de los equipos y, por último pero no por ello menos importante, garantizar un servicio de calidad y con unos estándares de confort adecuados para sus usuarios.

Basándose en estos últimos parámetros surge la necesidad de la creación de un sistema como vmRail. La empresa Virtualmechanichs S.L. decidió crear un sistema novedoso que midiera diversos aspectos de la red vial, desde la calidad geométrica de la vía, las vibraciones que registra el vehículo a lo largo de un viaje o el desgaste del perfil del raíl. La digitalización de este proceso es necesaria para poder obtener la información necesaria para realizar las tareas de mantenimiento preventivo adecuadas para garantizar que todo cumple con los estándares mínimos. El sistema propuesto se conoce como TGMS, que corresponde a las siglas en inglés de sistema de medición de geometría de vía (Track Geometry Measurement System).

Este tipo de mediciones se suelen hacer con vehículos especiales equipados con los sistemas y sensores necesarios para medir la geometría de la vía. Hay dos opciones: los trenes laboratorios y los carros auscultadores. Los trenes laboratorio son vehículos ferroviarios equipados con muchos sensores tales como Unidades de medición inerciales (IMU), láseres de posición u odómetros entre otros muchos. Estos vehículos son muy caros y necesitan una gran inversión por lo que no son muy comunes.

La otra solución propuesta son los carros auscultadores. Estos son unos equipos que normalmente son empujados a través de la vía por un operario cualificado para realizar la medición de los distintos parámetros de la vía. Dependiendo del carro estará preparado para medir unas características u otras. El proceso es lento y tiene que hacerse en horas de no funcionamiento del servicio para no interrumpir el correcto funcionamiento del sistema.

Sin embargo, lo novedoso del TGMS es que va embarcado en el vehículo, por lo que podemos aprovechar los viajes diarios del metro o el tren para tomar las medidas. Además, como puede estar funcionando regularmente podremos detectar los errores o problemas antes que si realizamos unas mediciones periódicas con los sistemas antiguos. Indudablemente este sistema reduce el trabajo necesario para garantizar la calidad del sistema vial, mejora la vida útil de los elementos del vehículo y garantiza un mejor servicio para los usuarios del servicio de transporte público. Además, las soluciones existentes presentadas anteriormente (carros auscultadores y trenes laboratorios) son soluciones muy costosas, de unos 100.000 € la primera y de hasta 40M € la segunda [4].

Dentro de este proyecto, que explicaremos con más detalle en la sección 1.4, nos encargaremos de trabajar en el desarrollo de un algoritmo de optimización que nos proporcione la posición del raíl en cada medición. Para ello tendremos que buscar cuál es la mejor transformación espacial afín (Sección 3.1) que hace que nuestras medidas se ajusten al carril ideal (Sección 2.3). Una vez que hayamos encontrado dicha transformación podremos calcular la posición del carril en el mundo real en 3D, lo que nos permitirá determinar si se cumplen los estándares geométricos de vía estipulados por las autoridades pertinentes. En España, estos estándares están recogidos en las normas UNE de aplicaciones ferroviarias.

Este trabajo Fin de Grado representa una pequeña parte del proyecto general y se encuadra en el marco de la Tesis Doctoral de Ingeniería Industrial de Eduardo Briaies Palacios [1], que ha sido mi cotutor durante el desarrollo de este trabajo. Muchos de los desarrollos matemáticos expuestos en este documento fueron realizados por él o por otros miembros de la empresa y yo los he utilizado para poder resolver la tarea que se me había encargado.

Este trabajo se ha implementado en su totalidad en Python ya que es un lenguaje muy potente, con numerosas funciones útiles para nuestro trabajo y con facilidad de exportación a otros sistemas u otros lenguajes. Además se trata de un lenguaje de código abierto, por lo que no nos encontraremos limitaciones a la hora de implementar nuestro código en el proyecto. Para ejecutar nuestros códigos y poder realizar los test para comprobar su validez hemos usado un ordenador con una CPU Intel(R) Core(TM) i7-10750-H 2.60 GHz.

1.3 Estado del arte

Debido a la importancia del proceso de medición de la calidad geométrica de la vía, como hemos expuesto en los apartados anteriores, ya hay ciertos proyectos enfocados en solucionar esta tarea. Vamos a tratar de explicar el enfoque que se le ha dado a este problema por otros investigadores.

El trabajo propuesto por Yue Yang et al. [5] propone un método preciso y rápido para inspeccionar la calidad geométrica de la vía. Este estudio comienza con una motivación parecida a la del vmRail. Las grandes velocidades que llegan a alcanzar los trenes en estos tiempos (hasta de 300 km/h en algunos casos) ha hecho que la calidad de la vía sea fundamental tanto para el mantenimiento preventivo de las instalaciones como para evitar accidentes. Un desgaste excesivo del perfil del raíl podría incluso provocar el descarrilamiento de un tren, con las horribles consecuencias que esto supondría.

Los métodos de inspección de raíles se pueden dividir en dos grandes grupos: los que necesitan contacto para realiza la medición y los que no. Los que no necesitan el contacto, como este, usan diversos sensores como un láser [6], para poder medir el perfil del raíl en cada punto de la vía. Además, para llevar a cabo este método es necesario obtener la medida exacta que tendría dicho carril para poder compararlo después con las medidas y determinar el desplazamiento sufrido respecto a la posición esperada o deseada.

Para realizar el emparejamiento entre las medidas y el raíl ideal se usan diversos algoritmos de *matching* de nubes de puntos. Por ejemplo, una buena propuesta para solucionar este problema es usar los algoritmos de RANSAC (Random sample consensus) o SAC-IA (sample consensus inicial alignment) para obtener una primera estimación de la pose inicial [7, 8]. Una vez que tenemos esta primera estimación, se procede a realizar un ajuste más exacto utilizando algoritmos como el ICP (Iterative Closest Point) o alguna de sus variantes [9–11]. Otros proponen la utilización de algoritmos basados en el método de los mínimos cuadrados, aunque estos algoritmos no están muy utilizados en este área, ya que son muy costosos computacionalmente.

Sin embargo, la solución que propone el estudio que estamos presentando [5] está basada en el algoritmo FGR (Fast Global Registration). Este algoritmo introduce el método Gauss-Newton para resolver un problema de mínimos cuadrados no lineal, apoyado por una función robusta de estimación de pesos. Este método no necesita alineación inicial y proporciona resultados muy robustos como se puede observar en el estudio realizado por Koltun et al. [12]. Todas las soluciones comentadas se basan en algoritmos matemáticos de *matching* y estimación de posiciones, con diversas variantes que pueden ir desde el peso que se le da a los diferentes puntos hasta el criterio elegido para emparejar los puntos de las dos nubes de puntos.

1.4 El proyecto vmRail

1.4.1 Introducción

La empresa Virtualmechanichs S.L. ha trabajado en diferentes proyectos del ámbito industrial con diversas compañías, pero siempre centrándose en la simulación de sistemas y la validación experimental. La continua necesidad de comprobar la validez de los modelos teóricos es la que daba sentido al desarrollo de las diversas soluciones experimentales propuestas por la empresa. Una vez que se desarrollaron los modelos matemáticos necesarios para llevar a cabo el proyecto y se podía plantear aplicar esta solución en la vida real, se decidió transmitir este conocimiento al sector ferroviario industrial.

En el verano de 2019 durante unas prácticas en Virtualmechanichs S.L., comencé a trabajar en este problema y seguí hasta diciembre de 2020. La primera tarea que tuve que realizar fue acondicionar diferentes códigos, como la calibración de las cámaras que veremos en la Sección 2.2, al lenguaje común del sistema final: Python. Una vez resuelto este problema comencé a trabajar en el desarrollo de los algoritmos de optimización necesarios para poder identificar la posición del carril en el mundo a partir de las imágenes tomadas por las cámaras del sistema vmRail.

La motivación de este proyecto, como hemos mencionado antes, es la de poder digitalizar una tarea que hoy en día es costosa y poco eficiente. Esta digitalización nos permite obtener información fiable y poder tomar

decisiones basándonos en ella. Esto supondría un ahorro de recursos para las empresas mantenedoras del sector ferroviario, lo cual acabaría repercutiendo en el cliente final, es decir, los usuarios de los sistemas públicos de transporte.

1.4.2 Características generales de vmRail

La solución propuesta para resolver el problema de la medición de la calidad geométrica de la vía es un sistema embarcado que se coloca en el propio vehículo y que cuenta con una serie de sensores y equipos de procesamiento de datos necesarios para poder ofrecer los resultados esperados. Los sensores en cuestión son los siguientes:

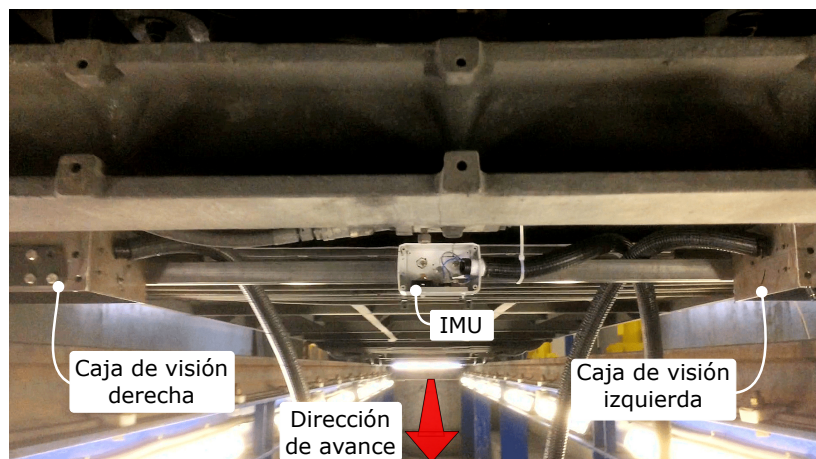


Figura 1.2 Sistema compacto de sensores. IMU en el centro de la imagen y las dos cámaras de visión a los lados..

- **IMU.** Se utiliza para medir la aceleración, velocidad angular y el campo magnético terrestre. También lleva implementado un algoritmo integrado de estimación de la orientación absoluta del vehículo basado en algoritmos de fusión de sensores. Son paquetes de sensores que normalmente incluyen acelerómetros, giróscopos y magnetómetros. A veces, también es común que incluyan sensores de temperatura y GPS, entre otros, para mejorar o apoyar las medidas de los otros sensores. Podemos verlo en el centro de la imagen en la Figura 1.2.
- **Sistema de visión artificial.** Este sistema está formado tanto por las cámaras de vídeo como los láseres planos integrados. Esto permite que se pueda realizar una reconstrucción 3D del carril en el sistema de referencia del vehículo. Tenemos uno para cada carril, como se puede observar en la Figura 1.2.
- **Odómetro.** Se utiliza para la medición del avance y la velocidad de avance del vehículo. Para ello utiliza las lecturas del tacogenerador del vehículo.
- **Sistema de adquisición de datos.** Es el sistema encargado de gestionar los sensores, sincronizar los datos medidos y enviarlos al ordenador de auscultación.
- **Ordenador de auscultación.** Es el encargado de procesar las imágenes de las cámaras y guardar las mediciones del resto de sensores.
- **Ordenador de abordó.** Se encarga de gestionar el sistema de adquisición y la recopilación de los datos generados.

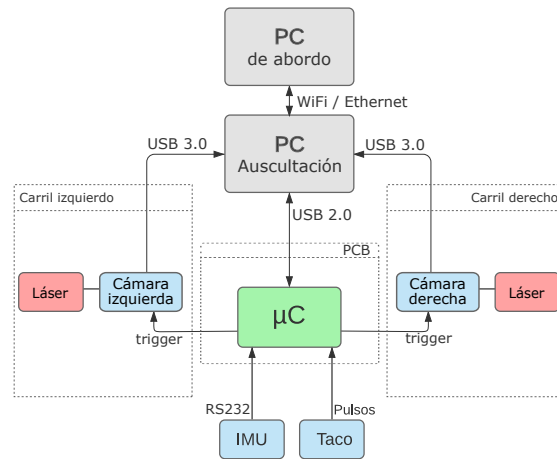


Figura 1.3 Esquema de la arquitectura electrónica y sensorial del sistema.

Con nuestro sistema queremos identificar las posibles irregularidades de la vía para poder reconstruir su geometría. Esta geometría viene dado por cuatro variables independientes, que son el desplazamiento lateral y vertical de cada uno de los dos carriles. Estos valores se denominan como y^{lir} y z^{lir} para el carril izquierdo (left irregularity), y y^{rir} y z^{rir} para el carril derecho (right irregularity), como se muestra en la Figura 1.4. Estos desplazamientos vienen provocados por irregularidades que pueden ser verticales u horizontales, o lo que es lo mismo, vienen dadas por irregularidades en el ancho de vía, la nivelación, el peralte y la alineación, términos que vamos a definir acorde a la norma UNE 13848 en la Sección 1.4.3.

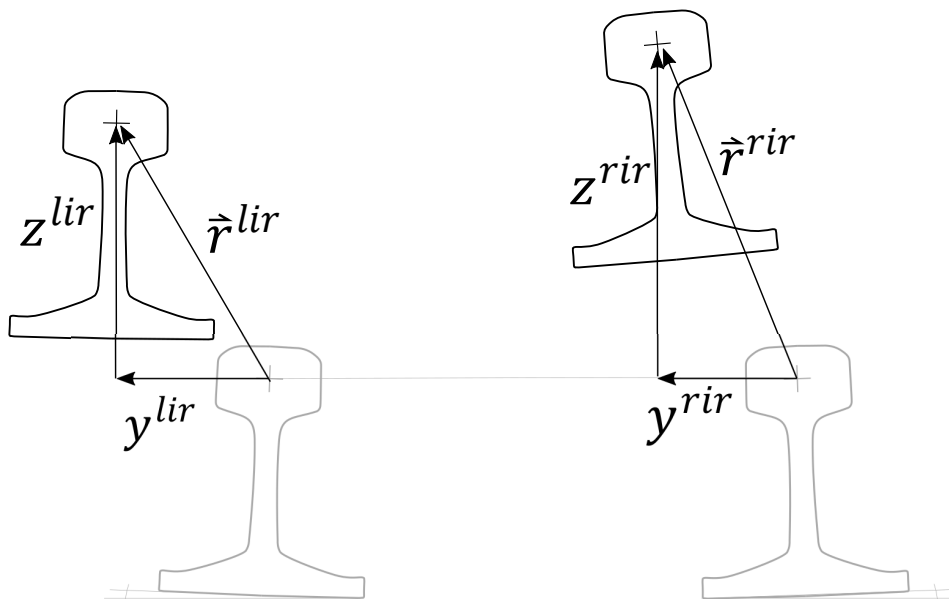


Figura 1.4 Desplazamientos de los carriles debido a las irregularidades de la vía.

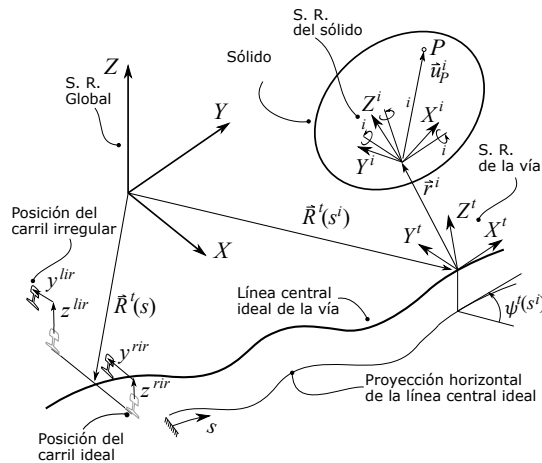


Figura 1.5 Global Frame $\langle X, Y, Z \rangle$ y Track Frame $\langle X^t, Y^t, Z^t \rangle$.

Para poder modelar las relaciones entre los sistemas de referencia que conforman el sistema, hemos tenido que usar fundamentos de cinemática de múltiples sólidos rígidos basados en el libro de Shabana de Dinámica de sistemas multicuerpo [13]. Esto nos permite conocer en qué posición del mundo real está por ejemplo una medida realizada por la cámara o por la IMU. Los elementos principales del sistema TGMS son el sensor inercial (IMU) unido rígidamente al eje de las ruedas, el conjunto de cámaras y láseres enfocando al lateral de los carriles (una cámara y un láser por raíl) y la señal del odómetro del vehículo. El esquema del sistema está mejor explicado en las siguiente figuras:

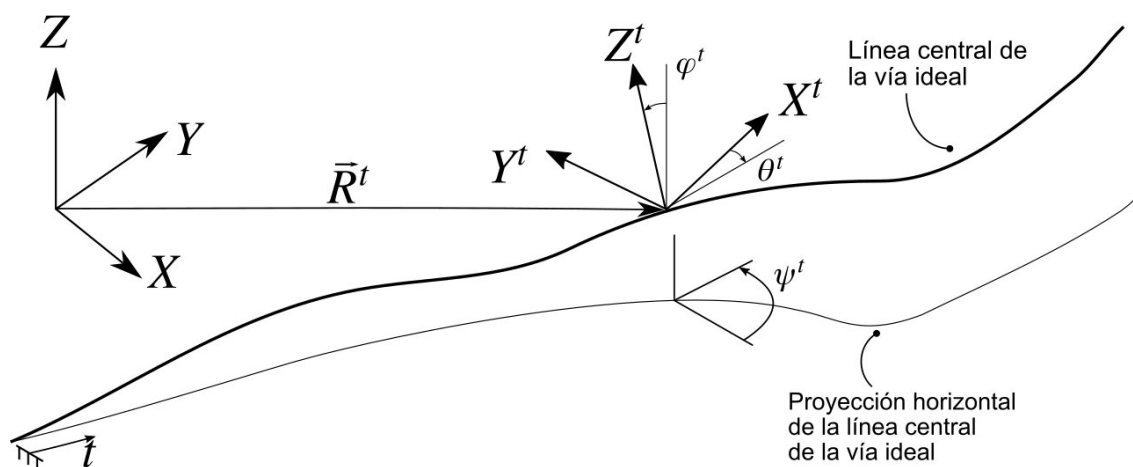


Figura 1.6 Global Frame $\langle X, Y, Z \rangle$ y Track Frame $\langle X^t, Y^t, Z^t \rangle$.

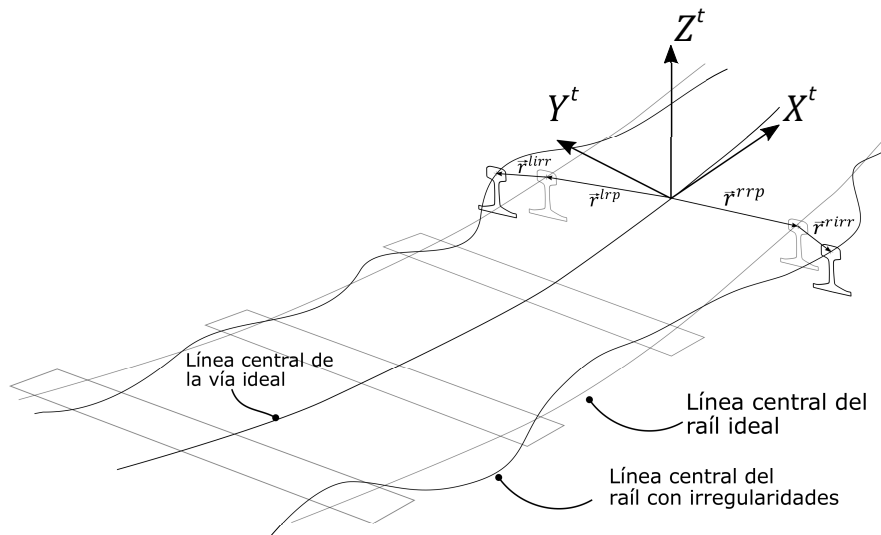


Figura 1.7 Irregularidades de las secciones del carril en su línea central.

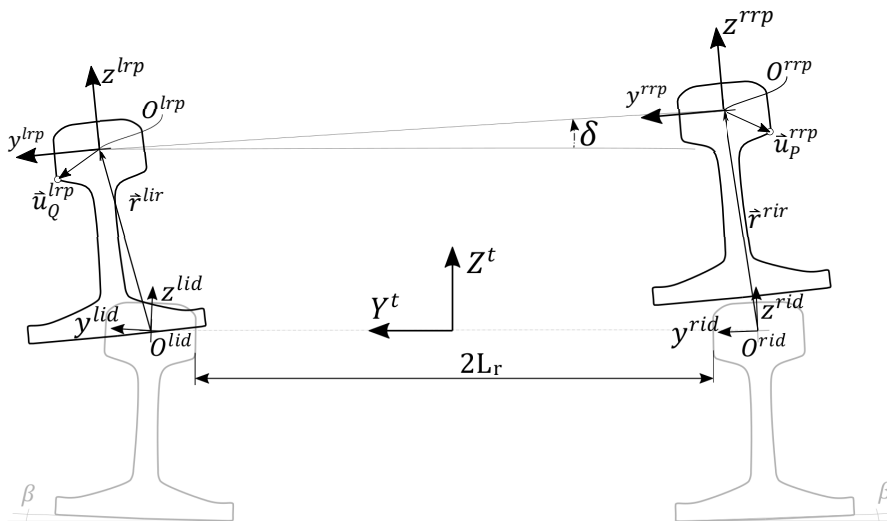


Figura 1.8 SR de una sección transversal de los carriles en la vía (rail profiles) ideal y real: $\langle x^{lrp}, y^{lrp}, z^{lrp} \rangle, \langle x^{rrp}, y^{rrp}, z^{rrp} \rangle, \langle x^{lid}, y^{lid}, z^{lid} \rangle, \langle x^{rid}, y^{rid}, z^{rid} \rangle$.

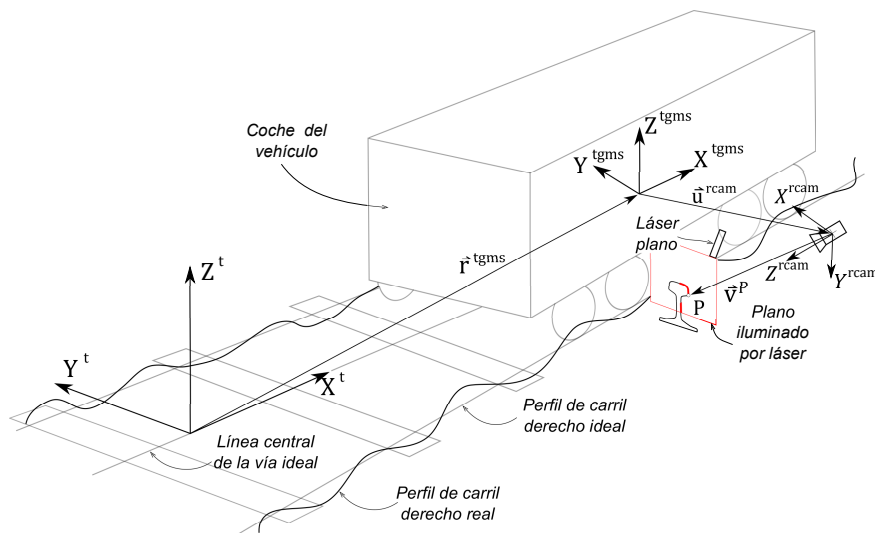


Figura 1.9 SR del vehículo instrumentado: *track frame*, *wheelset i frame* $\langle X^{wi}, Y^{wi}, Z^{wi} \rangle$ y *cameras frames*, $\langle X^{lcam}, Y^{lcam}, Z^{lcam} \rangle$, $\langle X^{rcam}, Y^{rcam}, Z^{rcam} \rangle$.

Como se puede apreciar en las Figuras 1.6, 1.8, 1.7 y 1.9, se utilizan seis sistemas de referencia (SR) para desarrollar el planteamiento del problema. Son los siguientes:

- **El sistema de referencia inercial y global, *global frame* (GF) $\langle X, Y, Z \rangle$.** Es un sistema fijo en el espacio, se coloca al comienzo del trayecto
- **El sistema de referencia de la vía, *track frame* (TF) $\langle X^t, Y^t, Z^t \rangle$.** Además de ser un SR es un campo definido en función del valor de la coordenada de arco s , definida a lo largo de la vía. La posición y orientación de este sistema de referencia respecto a GF es una función del arco s a lo largo de la línea central de la vía.
- **El sistema de referencia del sólido con el que es solidario el TGMS, al que llamaremos con las siglas TGMS $\langle X^{tgms}, Y^{tgms}, Z^{tgms} \rangle$.** Este sistema de referencia acompaña al sólido del vehículo en el que se encuentran los sensores, por lo que su cinemática estará influenciada por las irregularidades de la vía. Una geometría irregular procará movimientos de este SR en base a la dinámica del sistema multicuerpo.
- **Los sistemas de referencia de los raíles irregulares del perfil del raíl izquierdo, *left rail-profile frame* (LRPF), $\langle X^{lrp}, Y^{lrp}, Z^{lrp} \rangle$ y del perfil del raíl derecho, *right rail-profile frame* (RRPF), $\langle X^{rrp}, Y^{rrp}, Z^{rrp} \rangle$** Estos sistemas de referencia se definen para cada valor de la coordenada de la longitud de arco. Estos SR están unidos a las cabezas de los carriles para cada una de las secciones definidas dentro del plano $Y^t Z^t$.
- **Los sistemas de referencia de los raíles ideales del perfil izquierdo y derecho, *left ideal rail-profile frame* (LIDF) y *right ideal rail-profile frame* (RIDF)** Se definen a partir de la posición del origen del TF en la línea central de la vía.
- **Los sistemas de referencia de las cámaras izquierda, *left camera frame* (LCF) $\langle X^{lcam}, Y^{lcam}, Z^{lcam} \rangle$, y derecha, *right camera frame* (RCF) $\langle X^{rcam}, Y^{rcam}, Z^{rcam} \rangle$** Las curvas que se forman debido a la intersección entre el plano iluminado por el láser y la superficie de la cabeza de los carriles es adquirido por las cámaras según sus sistemas de referencia. El SR se define de tal manera que los ejes Z apunten al carril y los otros dos ejes formen sendos planos paralelos a los planos de las imágenes tomadas. Esto será de mucha utilidad para realizar el desarrollo matemático de la cinemática del seguimiento del movimiento como se explicará en la sección 2.1.

Aunque para el desarrollo del proyecto general todos estos sistemas de referencia son de vital importancia, para el desarrollo de nuestra tarea sólo haremos referencia a los SR *global frame* (GF), *track frame* (TF) y los de las cámaras derecha e izquierda.

1.4.3 Especificaciones de vmRail

Como cualquier proyecto de ámbito industrial, existen normas y reglas que regulan las características que debe cumplir nuestro sistema. En este caso, se trata de la norma UNE-EN 13848 [14–19]. En estas normas se definen los conceptos importantes para el proyecto además de las condiciones que estos tienen que cumplir.

Primero comenzamos haciendo una serie de aclaraciones y explicaciones de términos que son utilizados para referirse a las características de un sistema vial y que son importantes para seguir el desarrollo del proyecto. Dichas definiciones se encuentran en la norma UNE-EN 13848-1 referida a las aplicaciones ferroviarias, centrándose en la calidad geométrica de la vía [14].

En la citada norma encontramos las siguientes definiciones referidas a la geometría de la vía que vamos a presentar y explicar. El primero es el sistema de coordenadas de la vía. Este se puede ver en la figura 1.10.

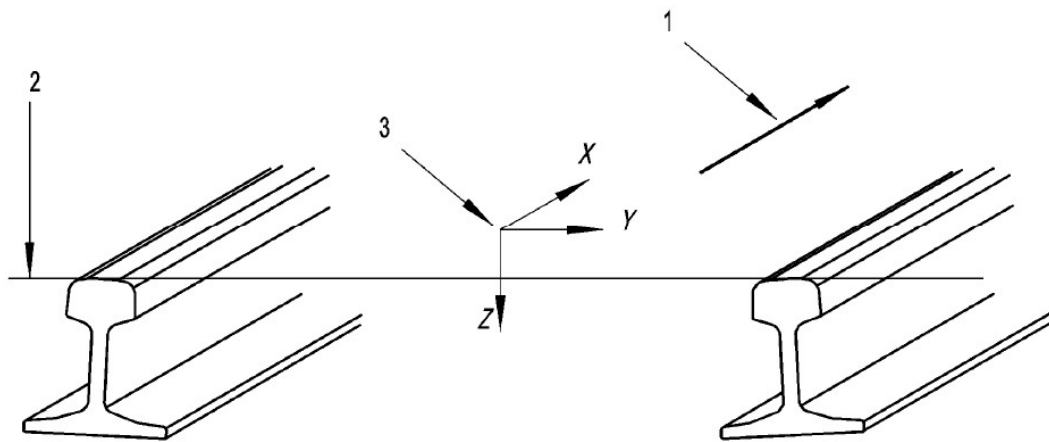


Figura 1.10 Sistema de coordenadas de una vía según la norma UNE referente a la caracterización geométrica de la vía.

Como podemos ver en la figura anterior, el sistema de referencias está definido por (3) y determina la dirección X como la dirección de avance del vehículo, Z la altura e Y el movimiento a lo largo del ancho de vía. Además se define la dirección de avance del desplazamiento como (1) que coincide con la dirección del eje X . Por último, se muestra la superficie de rodadura (2) que, como su nombre indica, es la superficie sobre la que se mueven las ruedas del tren o Metro. Formalmente, su definición es la superficie curva definida por el desplazamiento longitudinal de una línea recta perpendicular al eje de la vía y tangente a ambas tablas de rodadura. La tabla de rodadura es la superficie superior de la cabeza del carril.

La segunda definición importante que nos presenta la norma, es el ancho de vía G . Este es la menor distancia entre líneas perpendiculares a la superficie de rodadura cuya intersección con cada perfil de la cabeza del carril está situada en el punto P , en el rango de 0 a Z_p por debajo de la superficie de rodadura. Z_p siempre es 14mm . Cabe definir también, que la cabeza del carril es la parte superior de su sección, mientras que el alma es la parte delgada intermedia y la base es la parte inferior que se apoya en el suelo. Para la definición de este ancho de vía hace falta diferenciar entre dos situaciones: cuando la cabeza del carril está desgastada y cuando no, ya que esto hace que cambie la posición del punto P . Se puede apreciar esta diferencia fácilmente comparando las dos imágenes siguientes (Figuras 1.11 y 1.12)

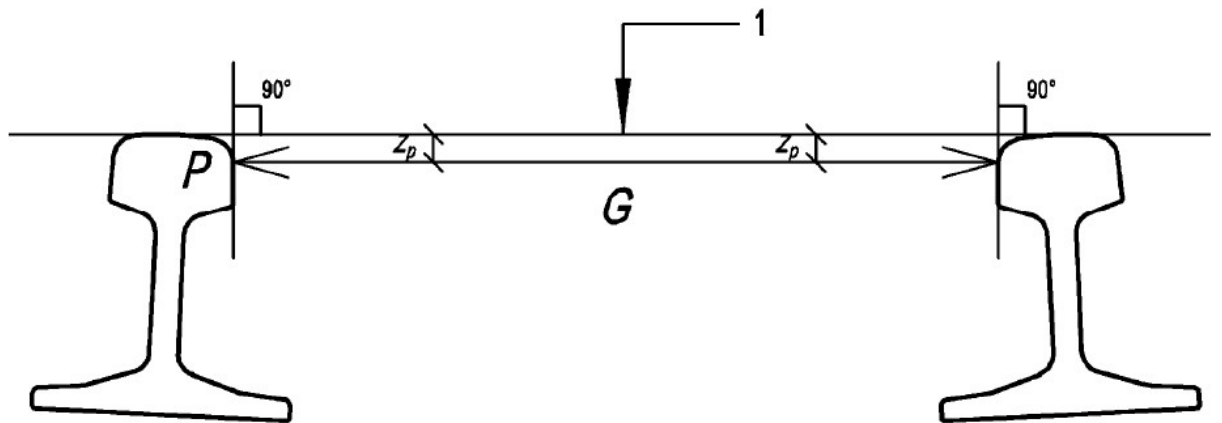


Figura 1.11 Ancho de vía para carril nuevo.

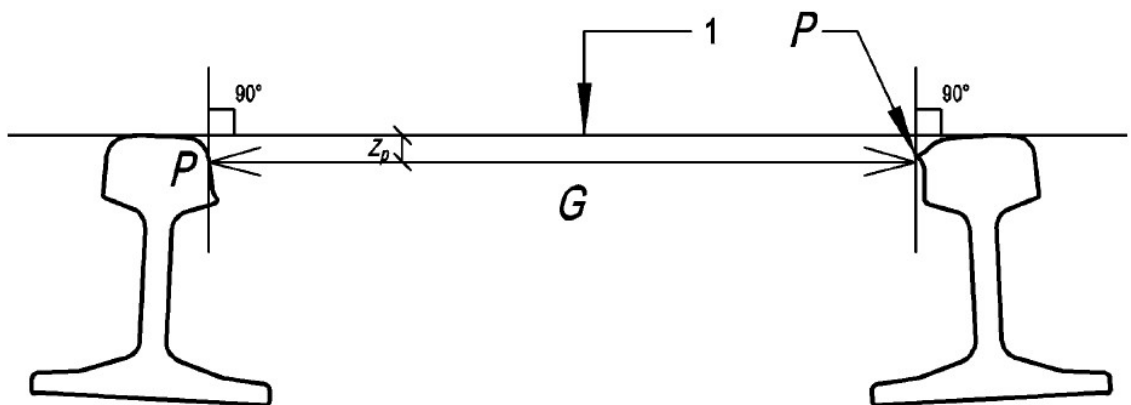


Figura 1.12 Ancho de vía para carril desgastado.

La tercera definición es la de nivelación longitudinal, que es la variación Z_p según el eje Z de alturas consecutivas de la tabla de rodadura de cada carril, expresada como la desviación de la posición vertical media. Podemos también verlo definido gráficamente en la Figura 1.13. En ella se marcan la tabla de rodadura (1) y la línea de referencia (2), elementos importantes para definir el movimiento del vehículo sobre el carril.

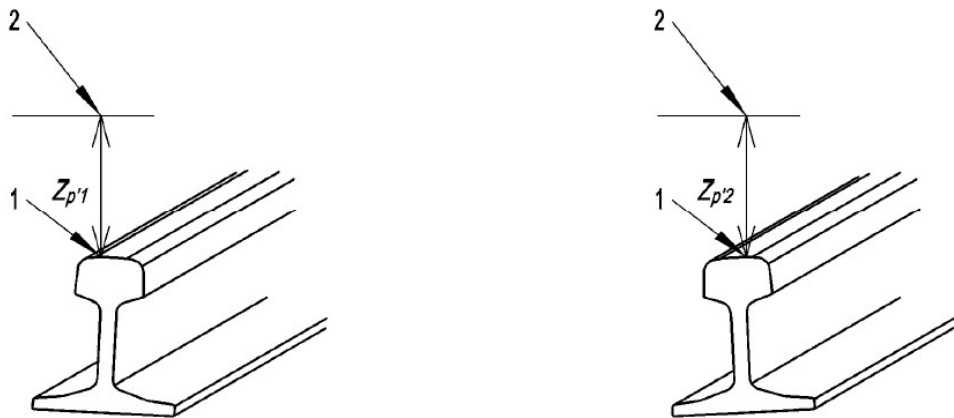


Figura 1.13 Nivelación longitudinal de la vía.

La siguiente definición es la de nivelación transversal. Esta queda definida por la diferencia en altura de tablas de rodadura adyacentes calculada según el ángulo entre la superficie de rodadura y un plano de referencia horizontal. Se expresa como la altura del cateto vertical de un triángulo rectángulo con una hipotenusa de valor igual al ancho de vía nominal más el ancho de la cabeza del carril, redondeado a los 10 mm más próximos. Esto puede apreciarse en la Figura 1.14.

NOTA Para anchos de vía nominales de 1 435 mm la hipotenusa es de 1 500 mm de longitud.
 Para anchos de vía nominales de 1 524 mm la hipotenusa es de 1 600 mm de longitud.
 Para anchos de vía nominales de 1 668 mm la hipotenusa es de 1 740 mm de longitud.

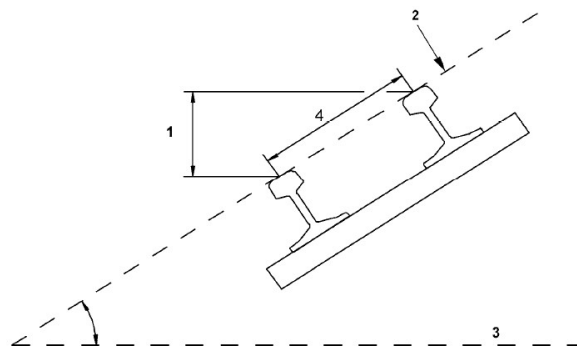


Figura 1.14 Nivelación transversal de la vía.

A continuación definimos el término de alineación. Se trata de la variación y_p según el eje Y de posiciones consecutivas del punto P sobre cada carril, expresada como desviación de la posición horizontal media (línea de referencia). Esto se ve reflejado en la Figura 1.15

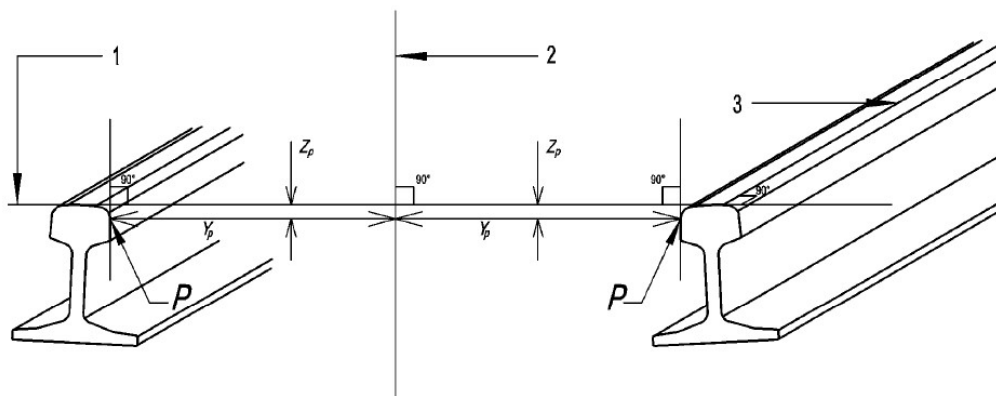


Figura 1.15 Alineación.

La última definición necesaria es la de alabeo, que es la diferencia algebraica entre dos nivelaciones transversales tomadas a una distancia de separación definida, expresada generalmente como un gradiente entre los dos puntos de medición.

En esta norma, además de definir los conceptos y términos importantes se estipula la manera de medirlos, los requisitos que deben cumplir tanto los sistemas de medición como las instalaciones medidas y el rango de validez de los valores tomados.

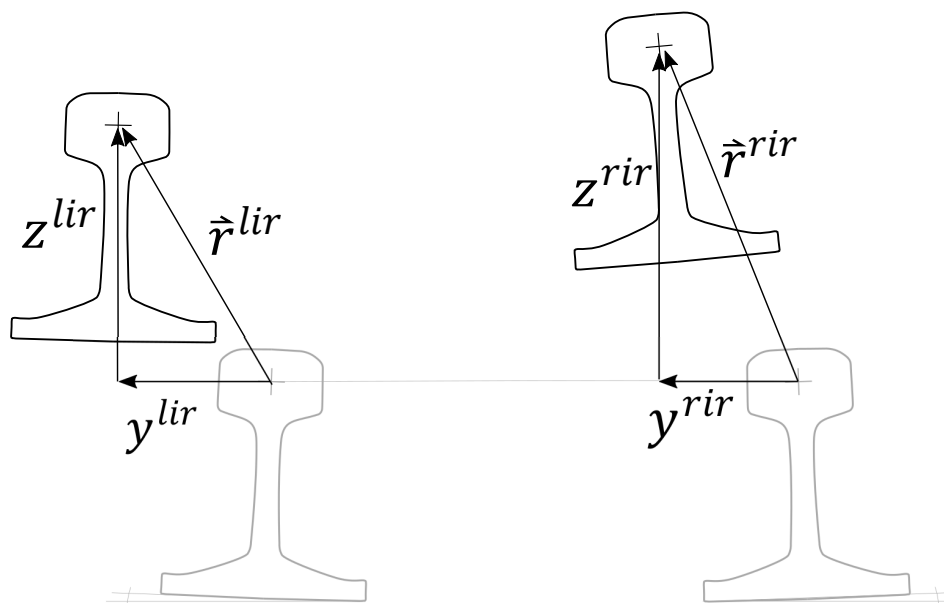


Figura 1.16 Desplazamientos de los carriles debido a las irregularidades de la vía.

Las irregularidades que medimos con nuestros instrumentos de medida las podremos utilizar para reconstruir la geometría de la vía. Dichas irregularidades se representan como una traslación en el plano YZ para cada uno de los carriles (Figura 1.16). Los valores se miden como y^{lir} , z^{lir} para el carril izquierdo y y^{rir} , z^{rir} para el derecho. El superíndice *ir* significa *irregularities*.

La norma establece que estas irregularidades se deben medir y clasificar en estas categorías:

- Las irregularidades laterales (Figura 1.17)
- Las irregularidades verticales (Figura 1.18)

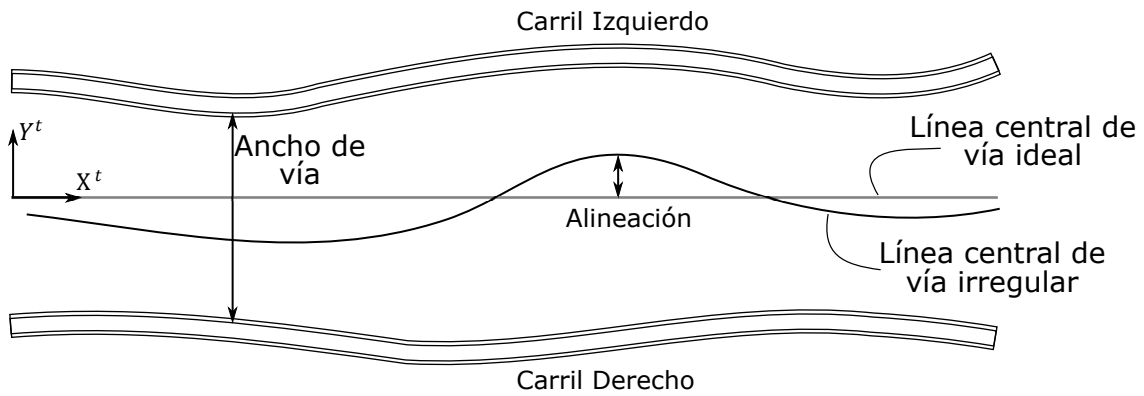


Figura 1.17 Irregularidades laterales.

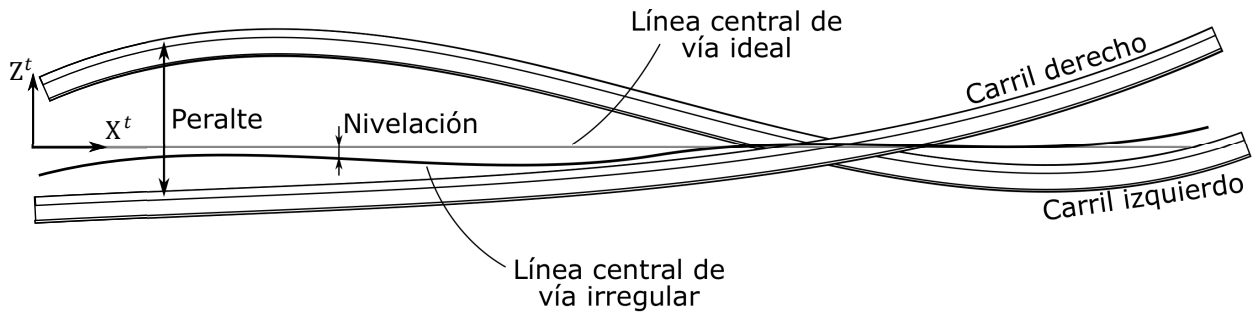


Figura 1.18 Irregularidades verticales.

2 Modelado matemático del problema

En este capítulo vamos a mostrar todos los desarrollos matemáticos utilizados en el transcurso de este Trabajo Fin de Grado. Principalmente son 3 los problemas que requieren una explicación profunda: la cinemática de las cámaras, la calibración de estas y el modelado de la curva del perfil del carril ideal.

2.1 Cinemática del seguimiento de movimiento usando visión por computador

2.1.1 Cinemática de la cámara

En esta sección procedemos a explicar la cinemática usada para encontrar el movimiento de un cuerpo usando una cámara [1], que equivale al modelo que vamos a usar para identificar las posiciones en 3D en el mundo de los puntos capturados por nuestra cámara embarcada en el metro.

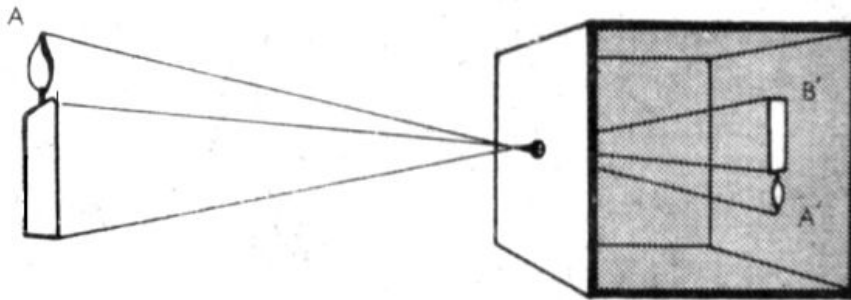


Figura 2.1 Modelo esquemático de una cámara estenopeica.

Para comenzar, necesitamos mostrar el modelo esquemático de una cámara. Para ello usamos lo que es conocido como modelo *pinhole* de una cámara. Está inspirado en las cámaras antiguas (Figura 2.1) que se usaban al principio de la era de las cámaras. La luz entraba a la cámara a través de un pequeño agujero y la imagen se iba formando en la parte trasera de una caja oscura donde se había colocado una placa fotosensible. Hubo numerosos estudios de cómo afectaba el tamaño de la apertura, el material utilizado para capturar las imágenes y el ángulo de dicha placa. Aunque las cámaras modernas ya no usan este principio de funcionamiento, sino que usan lentes, este modelo sigue siendo muy utilizado para relacionar la posición de los puntos de la imagen con su posición en el mundo real. La cámara que vamos a utilizar es la que aparece en la figura 2.2



Figura 2.2 Cámara Ximea xiQ USB3.0 MQ003CG-CM. Cámara de tamaño reducido, muy útil para esta aplicación.

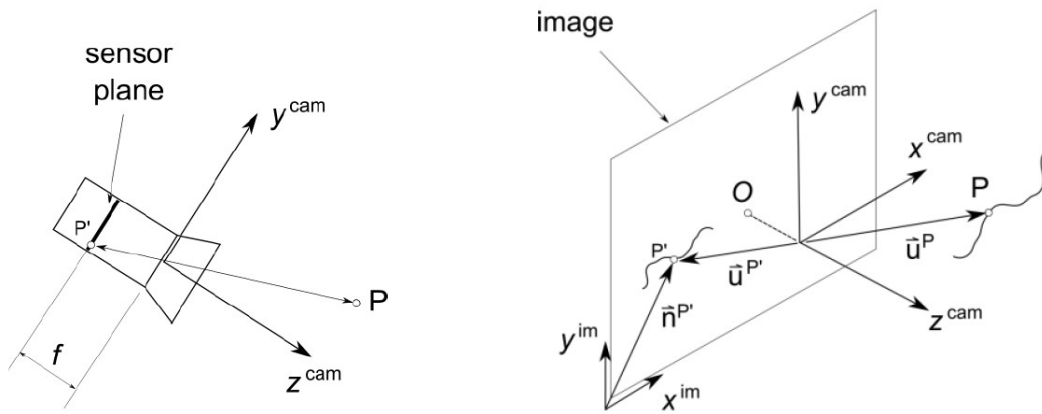


Figura 2.3 Vectores de posición en el sistema de referencia de la imagen y de la cámara.

Asumiendo que un punto P del mundo real aparece como P' en la imagen como puede observarse en la parte derecha de la Figura 2.3, podemos definir los vectores de posición de ambos vectores de la siguiente manera, siendo f la distancia focal de la cámara:

$$\bar{\mathbf{u}}^P = \begin{bmatrix} u_x^P \\ u_y^P \\ u_z^P \end{bmatrix}, \quad \bar{\mathbf{u}}^{P'} = \begin{bmatrix} u_x^{P'} \\ u_y^{P'} \\ -f \end{bmatrix} \quad (2.1)$$

La barra superior de los vectores significa que las coordenadas de dichas componentes están referenciadas al sistema de referencia de la cámara. Más tarde se introducirán dichas variables en el modelo del mundo real. Si observamos la parte izquierda de la Figura 2.3 se puede ver claramente que el valor de $u_z^{P'}$ siempre es $-f$. Además, podemos observar que los vectores $\bar{\mathbf{u}}^P$ y $\bar{\mathbf{u}}^{P'}$ están relacionados porque son colineales y sus componentes cumplen la siguiente expresión:

$$\frac{u_x^P}{u_x^{P'}} = \frac{u_y^P}{u_y^{P'}} = \frac{u_z^P}{-f} \quad (2.2)$$

Esta ecuación la sacamos si aplicamos el teorema de Tales a los triángulos similares que forman los vectores de posición con sus componentes. La ecuación 2.2 también puede escribirse de la siguiente manera:

$$\bar{\mathbf{u}}^{P'} = \frac{-f}{u_z^P} \bar{\mathbf{u}}^P \quad (2.3)$$

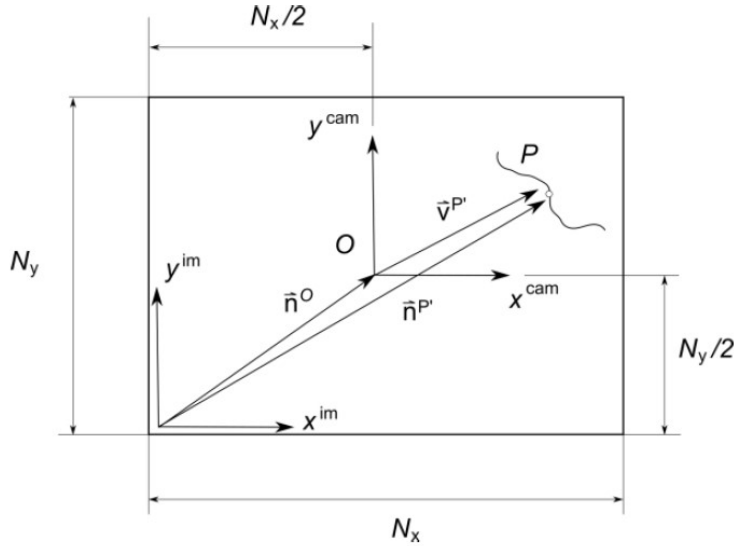


Figura 2.4 Sistemas de referencia en el plano de la imagen.

En la Figura 2.3 podemos ver la representación de la imagen o plano del sensor. Las componentes en este sistema de los vectores de posición $\mathbf{n}^{P'}$ no están representadas en unidades de longitud sino en píxeles. Un pixel es la unidad mínima de medida de una imagen. Representa cada uno de los puntos que forma la imagen en cuestión. Podemos ver que los vectores $\mathbf{n}^{P'}$ y $\bar{\mathbf{v}}^{P'}$ (proyección de $\bar{\mathbf{u}}^{P'}$ en el plano de la imagen) pueden relacionarse entre ellos si realizamos las siguientes suposiciones:

- 1. El eje óptico z^{cam} es perpendicular al plano del sensor y x^{cam} e y^{cam} son paralelos a x^{im} e y^{im} respectivamente.
- 2. El eje óptico z^{cam} intersecta al plano del sensor exactamente en el centro, de manera que el punto de intersección $\mathbf{n}^O = \begin{bmatrix} N_x/2 \\ N_y/2 \end{bmatrix}$, siendo N_x y N_y el número total de píxeles en las direcciones x e y .

Según estas condiciones, el vector $\bar{\mathbf{v}}^{P'}$ contiene los siguientes elementos:

$$\bar{\mathbf{v}}^P = \begin{bmatrix} u_x^{P'} \\ u_y^{P'} \end{bmatrix} \quad (2.4)$$

Es sencillo determinar que:

$$\mathbf{n}^{P'} = \mathbf{n}^O + \frac{1}{s} \bar{\mathbf{v}}^P = \mathbf{n}^O - \frac{f}{s u_z^P} \bar{\mathbf{u}}^P, \quad (2.5)$$

donde s es una constante que nos proporciona el tamaño del pixel en ancho o largo en metros (la mayoría de las cámaras usan el mismo tamaño de pixel aunque hay algunas que lo cambian). Para llegar a esta conclusión hemos tenido que usar las ecuaciones 2.3 y 2.4.

2.1.2 Coordenadas homogéneas de un vector

En este momento, es conveniente introducir el concepto de coordenadas homogéneas de un vector, ya que es una herramienta muy utilizada en la teoría de visión por computador y será introducida posteriormente en el desarrollo de esta sección. Un vector, en 2D o 3D, se representa por 3 o 4 componentes respectivamente cuando usamos coordenadas homogéneas:

$$\mathbf{v} = \begin{bmatrix} v_x \\ v_y \end{bmatrix} \Rightarrow \underline{\mathbf{v}} = \begin{bmatrix} av_x \\ av_y \\ a \end{bmatrix}, \quad \underline{\mathbf{v}} = \text{hom}(\mathbf{v}), \quad (2.6)$$

$$\mathbf{w} = \begin{bmatrix} w_x \\ w_y \\ w_z \end{bmatrix} \Rightarrow \underline{\mathbf{w}} = \begin{bmatrix} bw_x \\ bw_y \\ bw_z \\ b \end{bmatrix}, \quad \underline{\mathbf{w}} = \text{hom}(\mathbf{w}), \quad (2.7)$$

donde $\underline{\mathbf{v}}$ y $\underline{\mathbf{w}}$ son las versiones homogéneas de \mathbf{v} y \mathbf{w} , y a y b son números reales no nulos, que representan un factor de escala. Para obtener \mathbf{v} y \mathbf{w} a partir de sus coordenadas homogéneas solo necesitamos dividir las dos o tres primeras componentes (dependiendo de si estamos en el caso 2D o 3D) por la última. Claramente el valor de los factores a y b es irrelevante, ya que se puede realizar una rápida conversión si se cambia el factor. Podemos ver en la siguiente expresión que tanto $\underline{\mathbf{v}}_1$ como $\underline{\mathbf{v}}_2$ son representaciones en coordenadas homogéneas de un mismo vector \mathbf{v} y que ambas son equivalentes.

$$\underline{\mathbf{v}}_1 = \begin{bmatrix} av_x \\ av_y \\ a \end{bmatrix} \equiv \begin{bmatrix} cv_x \\ cv_y \\ c \end{bmatrix} = \underline{\mathbf{v}}_2 \quad (2.8)$$

2.1.3 Parámetros intrínsecos de la cámara

Utilizando la representación homogénea del vector $\mathbf{n}^{\mathbf{P}'}$, según lo explicado en la Sección 2.1.2, la ecuación 2.5 puede ser escrita de la siguiente manera, lo que nos va a permitir introducir el concepto de matriz de parámetros intrínsecos de una cámara \mathbf{M}^{int} .

$$\underline{\mathbf{n}}^{\mathbf{P}'} = \begin{bmatrix} -\frac{f}{s} & 0 & \frac{N_x}{2} \\ 0 & -\frac{f}{s} & \frac{N_y}{2} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \bar{u}_x^{\mathbf{P}'} \\ \bar{u}_y^{\mathbf{P}'} \\ \bar{u}_z^{\mathbf{P}'} \end{bmatrix} = \mathbf{M}^{\text{int}} \bar{\mathbf{u}}^{\mathbf{P}'} \quad (2.9)$$

La matriz \mathbf{M}^{int} es una matriz 3x3 que relaciona la posición en coordenadas homogéneas de un punto de la imagen con su posición en el plano de la cámara. Los parámetros intrínsecos son N_x, N_y, f y s .

Esta ecuación 2.9 relaciona la posición de los puntos de la imagen $\mathbf{n}^{\mathbf{P}'}$ con sus posiciones en el mundo real $\bar{\mathbf{u}}^{\mathbf{P}'}$. la ecuación puede usarse en una única dirección: podemos obtener la posición en el sistema de la cámara una vez que conocemos la del mundo real, pero no al contrario. Para obtener la posición en la imagen tenemos que aplicar la ecuación 2.9 y aplicar la operación $\mathbf{n}^{\mathbf{P}'} = \text{hom}^{-1}(\underline{\mathbf{n}}^{\mathbf{P}'})$, que consiste únicamente en dividir las dos primeras componentes de las coordenadas homogéneas entre la tercera (asumiendo que estamos en un

caso de transformación 3D-2D) y ya obtenemos las coordenadas en el plano de la imagen. Sin embargo, no podemos aplicar la misma técnica a los puntos de la imagen, porque no podríamos obtener el valor de $\bar{\mathbf{u}}^P$. Esto es consistente con la idea de que no podemos obtener una posición 3D a partir de una imagen 2D. Para solucionar esto se podría utilizar un plano láser u otras técnicas. Para esta aplicación hemos usado el láser que aparece en la figura 2.5.



Figura 2.5 Láser rojo de proyección plana.

La definición de los elementos de \mathbf{M}^{int} solo puede ser obtenido a partir de los parámetros intrínsecos si la suposición asumida anteriormente es válida: el eje óptico es perpendicular al plano del sensor de la cámara en el punto central. En caso contrario, la matriz tendría la siguiente forma:

$$\mathbf{M}^{int} = \begin{bmatrix} \alpha & \gamma & n_x^O \\ 0 & \beta & n_y^O \\ 0 & 0 & 1 \end{bmatrix} \tag{2.10}$$

Esta matriz tiene la misma forma que la definida anteriormente en la ecuación 2.9, con la excepción del término γ no nulo, que representa la asimetría de los ejes de la imagen. Por lo tanto, para encontrar los valores de los parámetros intrínsecos debemos encontrar los valores de α , γ , β y \mathbf{n}^O .

2.1.4 Parámetros extrínsecos de la cámara

Asumimos que la posición y la orientación de la cámara con respecto al sistema de referencia del mundo $\langle X, Y, Z \rangle$ está definidas por el vector \mathbf{r}^{cam} y por la matriz de transformación de la cámara \mathbf{A}^{cam} ambas mostradas en la Figura 2.6. Por lo tanto el vector de posición de cualquier punto P , utilizando la definición de transformación afín, está definido por la siguiente ecuación:

$$\mathbf{r}^P = \mathbf{r}^{cam} + \mathbf{A}^{cam} \bar{\mathbf{u}}^P \tag{2.11}$$

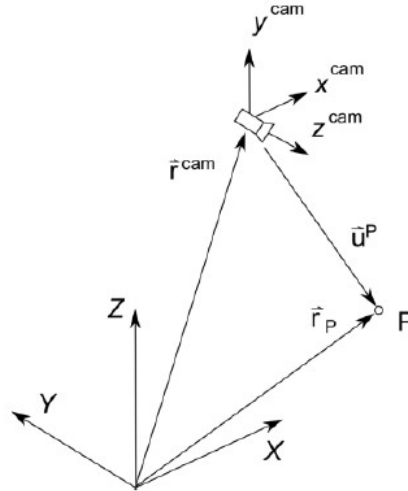


Figura 2.6 Posición de la cámara en el mundo.

Esta última ecuación puede ser invertida, ya que \mathbf{A}^{cam} es ortogonal, y obtendríamos la posición del punto respecto al sistema de referencia de la cámara:

$$\bar{\mathbf{u}}^{\text{P}} = (\mathbf{A}^{\text{cam}})^T (\mathbf{r}^{\text{P}} - \mathbf{r}^{\text{cam}}) \quad (2.12)$$

La expresión puede escribirse de una manera más compacta utilizando la matriz de transformación 3x4, algo que es muy común cuando tratamos con cinemática de robots:

$$\bar{\mathbf{u}}^{\text{P}} = [(\mathbf{A}^{\text{cam}})^T \quad -\mathbf{r}^{\text{cam}}] \begin{bmatrix} \mathbf{r}^{\text{P}} \\ 1 \end{bmatrix} = \mathbf{M}^{\text{ext}} \begin{bmatrix} \mathbf{r}^{\text{P}} \\ 1 \end{bmatrix} \quad (2.13)$$

La matriz \mathbf{M}^{ext} es la matriz de los parámetros extrínsecos, que nos proporciona la posición y orientación de la cámara respecto al sistema de referencia del mundo. Este sistema de referencia del mundo es el que hayamos definido como referencia absoluta de nuestro sistema. Como hemos explicado en la sección 1.4, para nuestro proyecto este sistema es el que se encuentra en el punto medio entre los dos raíles y acompaña solidariamente al cuerpo del tren durante su movimiento.

2.1.5 Relación entre puntos de la imagen y puntos del mundo

Sustituyendo la ecuación 2.13 en la ecuación 2.9 obtenemos lo siguiente:

$$\underline{\mathbf{n}}^{\text{P}'} = \mathbf{M}^{\text{int}} \mathbf{M}^{\text{ext}} \begin{bmatrix} \mathbf{r}^{\text{P}} \\ 1 \end{bmatrix} \quad (2.14)$$

Esta ecuación relaciona la posición de los puntos en la imagen y la posición de dichos puntos en el mundo. Como se ha explicado anteriormente en la ecuación 2.9, esta relación es unidireccional, podemos obtener $\underline{\mathbf{n}}^{\text{P}'}$ conocido \mathbf{n}^{P} , pero no al contrario. Esta última ecuación 2.14 puede escribirse también de la siguiente manera, donde c representa un factor de escala desconocido:

$$c \begin{bmatrix} \mathbf{n}^{P'} \\ 1 \end{bmatrix} = \mathbf{M}^{\text{int}} \mathbf{M}^{\text{ext}} \begin{bmatrix} \mathbf{r}^P \\ 1 \end{bmatrix} \quad (2.15)$$

2.1.6 Seguimiento de movimiento usando visión por computador

Como hemos explicado a lo largo de la sección el seguimiento de un punto P no puede ser realizado usando una única cámara, puesto que nos faltaría una componente por determinar. Sin embargo, esto es posible si dicho punto P se mueve sobre una superficie cuya ecuación en el sistema del mundo es conocida. Asumimos que la ecuación de esta supuesta superficie tiene la forma $f(x,y,z) = 0$. Si le añadimos esta ecuación a la ecuación 2.15 obtenemos el siguiente sistema de ecuaciones algebraicas:

$$\begin{cases} c \begin{bmatrix} \mathbf{n}^{P'} \\ 1 \end{bmatrix} = \mathbf{M}^{\text{int}} \mathbf{M}^{\text{ext}} \begin{bmatrix} \mathbf{r}^P \\ 1 \end{bmatrix} \\ f(r_x^P, r_y^P, r_z^P) = 0 \end{cases} \quad (2.16)$$

Este es un sistema de 4 ecuaciones con 4 incógnitas: las tres componentes de \mathbf{r}^P y el factor de escala c . Un caso particular de esta ecuación ocurre cuando el punto P se mueve sobre un plano. En este caso el sistema de ecuaciones anterior (Ecuación 2.16) queda de la siguiente manera:

$$\begin{cases} c \begin{bmatrix} \mathbf{n}^{P'} \\ 1 \end{bmatrix} = \mathbf{M}^{\text{int}} \mathbf{M}^{\text{ext}} \begin{bmatrix} \mathbf{r}^P \\ 1 \end{bmatrix} \\ Ar_x^P + Br_y^P + Cr_z^P + D = 0 \end{cases} \quad (2.17)$$

Las constantes A , B , C y D son los parámetros que definen nuestro plano, y se corresponden con las componentes del vector normal a él mismo. Si dicho plano es, por ejemplo, el plano $\langle X, Y \rangle$ nuestra ecuación 2.17 se transforma quedando con la siguiente forma:

$$c \begin{bmatrix} \mathbf{n}^{P'} \\ 1 \end{bmatrix} - \mathbf{M}^{\text{int}} \mathbf{M}^{\text{ext}} \begin{bmatrix} r_x^P \\ r_y^P \\ 0 \\ 1 \end{bmatrix} \quad (2.18)$$

Con este último cambio, ya solo tendríamos 3 ecuaciones y 3 incógnitas, simplificando su resolución.

Por último, estudiaremos el caso en el que tengamos dos cámaras observando el punto P . En este caso, sería posible realizar el seguimiento del movimiento sin conocer la ecuación del plano o superficie por la que se desplaza nuestro punto de interés. Si les llamamos cámara 1 y cámara 2, por ejemplo, las ecuaciones que resolverían el problema de seguimiento del punto P formarían el siguiente sistema:

$$\begin{cases} c_1 \begin{bmatrix} \mathbf{n}_1^{P'} \\ 1 \end{bmatrix} - \mathbf{M}_1^{\text{int}} \mathbf{M}_1^{\text{ext}} \begin{bmatrix} \mathbf{r}^P \\ 1 \end{bmatrix} = 0 \\ c_2 \begin{bmatrix} \mathbf{n}_2^{P'} \\ 1 \end{bmatrix} - \mathbf{M}_2^{\text{int}} \mathbf{M}_2^{\text{ext}} \begin{bmatrix} \mathbf{r}^P \\ 1 \end{bmatrix} = 0 \end{cases} \quad (2.19)$$

Los subíndices 1 y 2 hacen referencia a qué cámara pertenecen los puntos. Este sistema de ecuaciones 2.19 es sobredeterminado pero compatible, con 6 ecuaciones y 5 incógnitas. Podrían ser resueltas por un algoritmo de mínimos cuadrados basado en la pseudoinversa de la matriz de coeficientes.

Una vez descrito el modelo de sistemas de referencia y vectores que vamos a usar para resolver el problema, podemos darnos cuenta de que dependemos de la correcta calibración de las cámaras para obtener las matrices M^{int} y M^{ext} . Es por ello que el problema de la calibración de las cámaras es necesario y muy importante. Dicho proceso y la técnica utilizada para ello se tratarán más adelante a lo largo de este capítulo, en la Sección 2.2.

2.2 Calibración de las cámaras

Como hemos explicado en la Sección 2.1, para poder obtener la posición 3D de los puntos que obtenemos de las cámaras en 2D, debemos aplicar algún tipo de proceso de calibración que nos permita obtener información métrica de las imágenes. Al ser un proceso necesario para poder utilizar las cámaras en proyectos de este tipo, ha sido un ámbito en el que se han desarrollado numerosas soluciones utilizando distintas técnicas. A grandes rasgos podemos dividir estas técnicas en dos categorías: calibración fotogramétrica y auto calibración. Sus principales características son:

- **Calibración fotogramétrica.** La calibración se lleva a cabo con la observación de un objeto 3D del que se conoce su geometría. Normalmente este objeto está compuesto por varios (dos o tres) planos colocados perpendicularmente entre ellos. Este proceso requiere costosos aparatos de medición y calibración pero da resultados muy eficientes y robustos.
- **Auto calibración.** Las técnicas englobadas en este conjunto no necesitan un objeto de calibración. Desplazando la cámara antes una imagen estática podemos obtener parámetros intrínsecos de la cámara. Teóricamente, con tres desplazamientos de la cámara podríamos obtener la reconstrucción 3D de la imagen. Estas técnicas son muy flexibles y nos permiten obtener resultados sin necesidad de otros aparatos externos, ya sean de medida o calibración. Sin embargo, hay muchos parámetros que estimar por lo que si el proceso no se hace con suficiente precisión podemos obtener resultados poco fiables.

Para este proyecto hemos decidido usar el método propuesto por Zhengyou Zhang, encuadrado en la primera categoría de algoritmos de calibración expuestos anteriormente. Para esta calibración únicamente necesitamos que la cámara observe un patrón planar (Figura 2.7) en al menos dos orientaciones diferentes conocidas. El modelado matemático del problema lo hemos tomado de [1] y la tarea que realizamos fue implementarlo en Python para poder ser integrado en vmRail. Pasamos a explicar dicho proceso de calibración.

2.2.1 Calibración de cámaras usando el método de Zhang

El método presentado por Zhang en el año 2000 en [20] calcula los valores de las matrices M^{int} y M^{ext} para una cámara dada. En la versión que hemos utilizado [1], utilizamos el patrón de calibrado conocido como ChessBoard Pattern, que como su nombre indica, es una plantilla con cuadrados blancos y negros. Podemos ver en la Figura 2.9 dicho patrón y la configuración habitual que se usa: tres patrones ortogonales entre sí formando un triedro rectangular. El vértice de dicho triedro se toma como origen de coordenadas del sistema $\langle X, Y, Z \rangle$ y los 3 patrones representarían los planos $\langle X, Y \rangle$ (horizontal), $\langle Y, Z \rangle$ (panel trasero) y $\langle X, Z \rangle$ (panel lateral). El objetivo último de calcular estas matrices es que sin ellas no podemos ser capaces de, a partir de las coordenadas 2D de los puntos en la imagen, obtener la posición 3D en el mundo real.

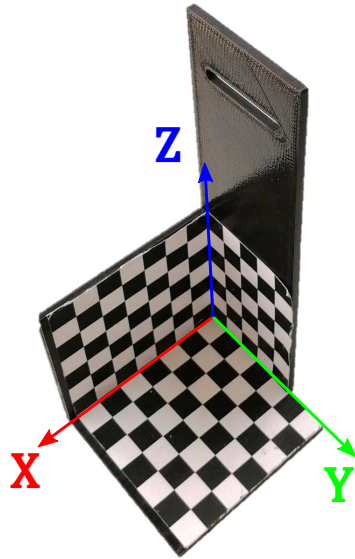


Figura 2.7 Patrón de calibración utilizado.

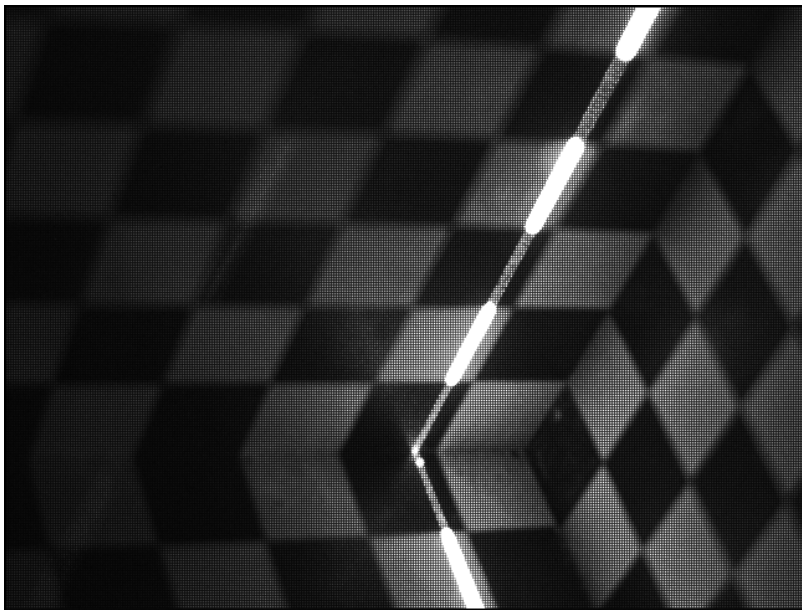


Figura 2.8 Patrón de calibración cámara izquierda.

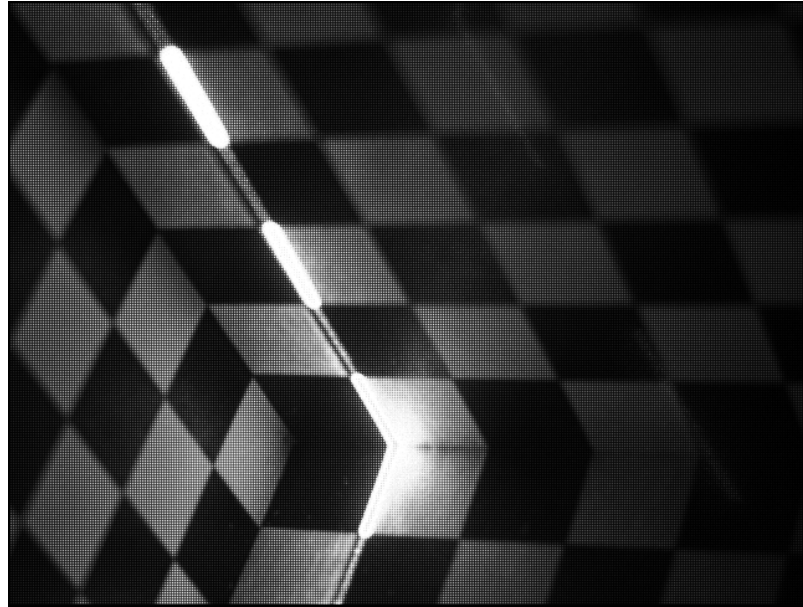


Figura 2.9 Patrón de calibración cámara derecha.

El método de Zhang requiere como datos de entrada la posición de unos puntos en el mundo, el vector \mathbf{r}^P , y su correspondencia con los puntos de la imagen $\mathbf{n}^{P'}$. En este ejemplo, hemos utilizado 10 puntos en cada uno de los planos, lo que resulta en un total de 30 puntos de calibración. La Figura 2.10 nos muestra la posición de los puntos de calibración en el sistema de referencia de la imagen a la izquierda, y a la derecha en el sistema de referencia del mundo. Estos puntos no se corresponden con los representados en la Tabla 2.1, pero sirven para ejemplificar cómo serían dichos puntos. Lo que buscamos ahora son las matrices de parámetros intrínsecos y extrínsecos de la cámara para poder pasar de nuestros puntos 2D de la imagen a las posiciones 3D de dichos puntos en el mundo real.

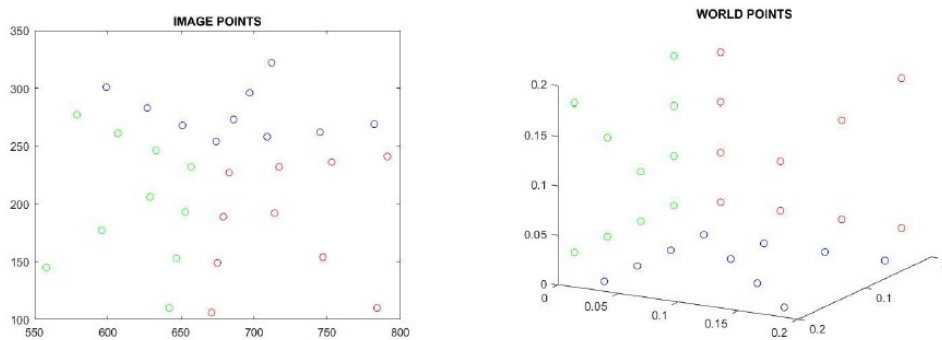


Figura 2.10 Puntos de calibración en la imagen (izq.) y en el mundo (dcha.).

Cuando tenemos las coordenadas de los puntos seleccionados y medidos para realizar la calibración de la cámara, el método de calibración de Zhang sigue 4 pasos:

- Encontrar las matrices de homografía
- Calcular los parámetros intrínsecos
- Calcular los parámetros extrínsecos
- Optimizar los parámetros intrínsecos e extrínsecos calculados anteriormente

Tabla 2.1 Vectores de posición de los puntos de calibración en los sistemas de la imagen y del mundo.

Plano X-Y					Plano X-Z					Plano Y-Z				
$n^{P'}$ (pix.)		r^P (mm.)			$n^{P'}$ (pix.)		r^P (mm.)			$n^{P'}$ (pix.)		r^P (mm.)		
674	254	25	25	0	657	232	25	0	25	683	227	0	25	25
686	273	75	75	0	629	206	75	0	75	714	192	0	75	75
697	296	125	125	0	596	177	125	0	125	747	154	0	125	125
712	322	175	175	0	558	145	174	0	175	784	110	0	175	175
709	258	25	75	0	653	193	25	0	75	679	189	0	25	75
745	262	25	125	0	647	153	25	0	125	675	149	0	25	125
782	269	25	175	0	642	110	25	0	175	671	106	0	25	175
651	268	75	25	0	633	246	75	0	25	717	232	0	75	25
627	283	125	25	0	607	261	125	0	25	753	236	0	125	25
599	301	175	25	0	579	277	175	0	25	791	241	0	175	25

La matriz de homografía representa la transformación homográfica que sufren los puntos para pasar de un sistema a otro. La homografía es toda transformación proyectiva que determina una correspondencia entre dos figuras geométricas planas, de forma que a cada uno de los puntos y las rectas de una de ellas les corresponden, un punto y una recta de la otra. A continuación, pasamos a explicar detalladamente cada uno de los pasos que acabamos de exponer.

Búsqueda de las matrices de homografía

La aplicación de la ecuación 2.15 a los puntos de los diferentes planos definidos por nuestro modelo de calibración (Figura 2.10) nos proporciona una transformación plano-plano también conocida como homografía [1]. Debido a que las primeras 3 columnas de nuestra matriz M^{ext} representan una matriz de rotación, pueden ser interpretadas como las componentes de los vectores unitarios del sistema del mundo representados en el sistema de la cámara. Esto se modela matemáticamente de la siguiente manera:

$$M^{ext} = [(A^{cam} \quad -\bar{r}^{cam})] = [\bar{i}^{wd} \quad \bar{j}^{wd} \quad \bar{k}^{wd} \quad -\bar{r}^{cam}], \tag{2.20}$$

donde el superíndice wd representa el sistema de referencia del mundo (wd=World). La aplicación de la ecuación 2.15 a los puntos X-Y da lugar a la siguiente expresión:

$$c \begin{bmatrix} n^{P'} \\ 1 \end{bmatrix} = M^{int} [\bar{i}^{wd} \quad \bar{j}^{wd} \quad \bar{k}^{wd} \quad -\bar{r}^{cam}] \begin{bmatrix} r_x^P \\ r_y^P \\ 0 \\ 1 \end{bmatrix} = M^{int} [\bar{i}^{wd} \quad \bar{j}^{wd} \quad -\bar{r}^{cam}] \begin{bmatrix} r_x^P \\ r_y^P \\ 1 \end{bmatrix} \tag{2.21}$$

Si llamamos matriz de homografía a la siguiente matriz 3x3:

$$\mathbf{H}^{XY} = \mathbf{M}^{\text{int}} [\bar{\mathbf{i}}^{\text{wd}} \quad \bar{\mathbf{j}}^{\text{wd}} \quad -\bar{\mathbf{r}}^{\text{cam}}], \quad (2.22)$$

entonces la ecuación 2.21 puede ser reescrita de la siguiente manera:

$$c \begin{bmatrix} n_x^{P'} \\ n_y^{P'} \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{H}_{11}^{XY} & \mathbf{H}_{12}^{XY} & \mathbf{H}_{13}^{XY} \\ \mathbf{H}_{21}^{XY} & \mathbf{H}_{22}^{XY} & \mathbf{H}_{23}^{XY} \\ \mathbf{H}_{31}^{XY} & \mathbf{H}_{32}^{XY} & \mathbf{H}_{33}^{XY} \end{bmatrix} \begin{bmatrix} r_x^P \\ r_y^P \\ 1 \end{bmatrix} \quad (2.23)$$

Debido a que tenemos 10 puntos en cada plano, podríamos escribir 10 versiones de la ecuación anterior 2.23. Además, para cada punto P el factor de escala c , que aparece en la ecuación anterior, cambia y lo podemos calcular de la siguiente manera:

$$c = \mathbf{H}_{31}^{XY} r_x^P + \mathbf{H}_{32}^{XY} r_y^P + \mathbf{H}_{33}^{XY} \quad (2.24)$$

Sustituyendo este resultado en la primera y segunda ecuación del sistema descrito (Ecuación 2.23) y reordenando los términos, obtenemos la siguiente expresión:

$$\begin{aligned} (\mathbf{H}_{31}^{XY} r_x^P + \mathbf{H}_{32}^{XY} r_y^P + \mathbf{H}_{33}^{XY}) n_x^{P'} - (\mathbf{H}_{11}^{XY} r_x^P + \mathbf{H}_{12}^{XY} r_y^P + \mathbf{H}_{13}^{XY}) &= 0 \\ (\mathbf{H}_{31}^{XY} r_x^P + \mathbf{H}_{32}^{XY} r_y^P + \mathbf{H}_{33}^{XY}) n_y^{P'} - (\mathbf{H}_{21}^{XY} r_x^P + \mathbf{H}_{22}^{XY} r_y^P + \mathbf{H}_{23}^{XY}) &= 0 \end{aligned} \quad (2.25)$$

Considerando los elementos de la matriz de homografía como las incógnitas de un sistema formado por dos ecuaciones homogéneas lineales, la ecuación 2.25 podría reescribirse como:

$$\begin{bmatrix} -r_x^P & -r_y^P & -1 & 0 & 0 & 0 & n_x^{P'} r_x^P & n_y^{P'} r_y^P & n_x^{P'} \\ 0 & 0 & 0 & -r_x^P & -r_y^P & -1 & n_y^{P'} r_x^P & n_y^{P'} r_y^P & n_y^{P'} \end{bmatrix} \begin{bmatrix} \mathbf{H}_{11}^{XY} \\ \mathbf{H}_{12}^{XY} \\ \mathbf{H}_{13}^{XY} \\ \mathbf{H}_{21}^{XY} \\ \mathbf{H}_{22}^{XY} \\ \mathbf{H}_{23}^{XY} \\ \mathbf{H}_{31}^{XY} \\ \mathbf{H}_{32}^{XY} \\ \mathbf{H}_{33}^{XY} \end{bmatrix} = \mathbf{L}_i^{XY}(\mathbf{n}^{P'}, \mathbf{r}^P) \hat{\mathbf{H}}^{XY} = 0 \quad (2.26)$$

La matriz $\mathbf{L}_i^{XY}(\mathbf{n}^{P'}, \mathbf{r}^P)$ tiene una dimensión de 2×9 y depende de la posición del punto P en la imagen y en el sistema de referencia del mundo. Por su parte, la matriz $\hat{\mathbf{H}}^{XY}$ contiene los términos de la matriz de homografía para el plano $\langle X, Y \rangle$ y tiene un tamaño de 9×1 . Podemos observar que podemos obtener un conjunto de dos ecuaciones homogéneas para cada uno de los puntos P en el plano $\langle X, Y \rangle$. Agrupando todos los resultados, nuestra ecuación queda de la siguiente forma:

$$\begin{bmatrix} \mathbf{L}_1^{XY} \\ \mathbf{L}_2^{XY} \\ \vdots \\ \mathbf{L}_n^{XY} \end{bmatrix} \hat{\mathbf{H}}^{XY} = \mathbf{L}^{XY} \hat{\mathbf{H}}^{XY} = 0, \quad (2.27)$$

donde \mathbf{L}^{XY} es una matriz $2nXY \times 9$, siendo nXY el número de puntos utilizado para la calibración en el plano $\langle X, Y \rangle$ (en este caso $nXY=10$). Debido a que el sistema de ecuaciones 2.27 es homogéneo, podemos obtener soluciones no triviales para $\hat{\mathbf{H}}^{XY}$ si el sistema es sobredeterminado (más ecuaciones que incógnitas). En este caso se puede demostrar que la solución óptima del sistema 2.27, utilizando la minimización de la suma de errores cuadráticos como criterio, es el autovector asociado al menor autovalor de la matriz $(\mathbf{L}^{XY})^T \mathbf{L}^{XY}$. Formulándolo con una nomenclatura parecida a la que usaremos cuando lo implementemos en Python, la búsqueda del valor óptimo queda de la siguiente manera:

$$\begin{aligned} [\lambda, \phi] &= eig((\mathbf{L}^{XY})^T \mathbf{L}^{XY}) \\ [\lambda_{min}, i_{min}] &= min(\lambda) \\ \hat{\mathbf{H}}^{XY} &= \phi(:, i_{min}) \end{aligned} \quad (2.28)$$

El razonamiento que hemos hecho para el plano $\langle X, Y \rangle$ podría plantearse análogamente para los otros dos planos que necesitamos calibrar ($\langle X, Z \rangle$, $\langle Y, Z \rangle$) para encontrar las matrices \mathbf{H}^{XZ} y \mathbf{H}^{YZ} . Debido a que el sistema es homogéneo, con este procedimiento obtenemos \mathbf{H}^{XY} multiplicado por un determinado factor de escala. Esto significa que $e\hat{\mathbf{H}}^{XY}$ es también solución del sistema, siendo e cualquier número real.

Búsqueda de parámetros intrínsecos

Tomando como punto de partida la ecuación 2.2.1, podemos deducir que los vectores unitarios $\bar{\mathbf{i}}^{wd}$ y $\bar{\mathbf{j}}^{wd}$ son la primera y la segunda columna de la matriz $(\mathbf{M}^{int})^{-1} \mathbf{H}^{XY}$, lo cual representamos en las siguientes expresiones:

$$\begin{aligned} \bar{\mathbf{i}}^{wd} &= (\mathbf{M}^{int})^{-1} \mathbf{h}_1^{XY}, \\ \bar{\mathbf{j}}^{wd} &= (\mathbf{M}^{int})^{-1} \mathbf{h}_2^{XY}, \end{aligned} \quad (2.29)$$

donde \mathbf{h}_i^{XY} es la i -ésima columna de \mathbf{H}^{XY} . Ahora intentaremos representar la ecuación anterior en forma matricial. Debido a que $\bar{\mathbf{i}}^{wd}$ y $\bar{\mathbf{j}}^{wd}$ son vectores unitarios y ortogonales, podemos deducir las siguientes ecuaciones, donde $(\mathbf{M}^{int})^{-T} = ((\mathbf{M}^{int})^{-1})^T$.

$$\left. \begin{aligned} (\bar{\mathbf{i}}^{wd})^T \bar{\mathbf{j}}^{wd} &= 0 \\ \bar{\mathbf{i}}^{wd} &= (\bar{\mathbf{j}}^{wd})^T \bar{\mathbf{j}}^{wd} \end{aligned} \right\} \Rightarrow \begin{cases} (\mathbf{h}_1^{XY})^T (\mathbf{M}^{int})^T (\mathbf{M}^{int})^{-1} \mathbf{h}_2^{XY} = 0 \\ (\mathbf{h}_1^{XY})^T (\mathbf{M}^{int})^{-T} (\mathbf{M}^{int})^{-1} \mathbf{h}_1^{XY} - (\mathbf{h}_2^{XY})^T (\mathbf{M}^{int})^{-T} (\mathbf{M}^{int})^{-1} \mathbf{h}_2^{XY} = 0 \end{cases} \quad (2.30)$$

Utilizando notación matricial, la matriz \mathbf{B} se define de la siguiente forma:

$$\mathbf{B} = (\mathbf{M}^{int})^{-T} (\mathbf{M}^{int})^{-1} = \begin{bmatrix} \frac{1}{\alpha^2} & -\frac{\gamma}{\alpha^2 \beta} & \frac{n_y^o \gamma - n_x^o \beta}{\alpha^2 \beta} \\ -\frac{\gamma}{\alpha^2 \beta} & \frac{\gamma}{\alpha^2 \beta} + \frac{1}{\beta^2} & -\frac{n_y^o \gamma - n_x^o \beta}{\alpha^2 \beta} - \frac{n_y^o}{\beta^2} \\ \frac{n_y^o \gamma - n_x^o \beta}{\alpha^2 \beta} & -\frac{n_y^o \gamma - n_x^o \beta}{\alpha^2 \beta} - \frac{n_y^o}{\beta^2} & \frac{(n_y^o \gamma - n_x^o \beta)^2}{\alpha^2 \beta} + \frac{(n_y^o)^2}{\beta^2} + 1 \end{bmatrix} \quad (2.31)$$

Esta matriz ha sido calculada explícitamente en término de los parámetros intrínsecos de la cámara 2.10 y podemos observar que \mathbf{B} es una matriz simétrica. La ecuación 2.30 incluye 3 productos de la siguiente forma:

$$\begin{aligned}
& (\mathbf{h}_i^{XY})^T \mathbf{B} \mathbf{h}_i^{XY} = \\
& = \begin{bmatrix} \mathbf{h}_{i1}^{XY} \mathbf{h}_{j1}^{XY} & \mathbf{h}_{i1}^{XY} \mathbf{h}_{j2}^{XY} + \mathbf{h}_{i2}^{XY} \mathbf{h}_{j1}^{XY} & \mathbf{h}_{i2}^{XY} \mathbf{h}_{j2}^{XY} & \mathbf{h}_{i3}^{XY} \mathbf{h}_{j1}^{XY} + \mathbf{h}_{i1}^{XY} \mathbf{h}_{j3}^{XY} & \mathbf{h}_{i3}^{XY} \mathbf{h}_{j2}^{XY} + \mathbf{h}_{i2}^{XY} \mathbf{h}_{j3}^{XY} & \mathbf{h}_{i3}^{XY} \mathbf{h}_{j3}^{XY} \end{bmatrix} \begin{bmatrix} B_{11} \\ B_{12} \\ B_{22} \\ B_{13} \\ B_{23} \\ B_{33} \end{bmatrix} = \\
& = \mathbf{v}_{ij}^{XY} \hat{\mathbf{B}}
\end{aligned} \tag{2.32}$$

La matriz \mathbf{v}_{ij}^{XY} tiene una dimensión de 1×6 y está contruida usando los elementos de \mathbf{H}^{XY} , mientras que la matriz $\hat{\mathbf{B}}$ es una matriz 6×1 que contiene los elementos de \mathbf{B} . Utilizando esta nomenclatura, las dos ecuaciones de 2.30 pueden reescribirse como:

$$\begin{bmatrix} \mathbf{v}_{12}^{XY} \\ \mathbf{v}_{11}^{XY} - \mathbf{v}_{22}^{XY} \end{bmatrix} \hat{\mathbf{B}} = 0 \tag{2.33}$$

A partir de las matrices \mathbf{H}^{XZ} y \mathbf{H}^{YZ} podemos obtener 4 ecuaciones más como esta, lo que nos definiría el sistema completo, que tendría la siguiente forma:

$$\begin{bmatrix} \mathbf{v}_{12}^{XY} \\ \mathbf{v}_{11}^{XY} - \mathbf{v}_{22}^{XY} \\ \mathbf{v}_{12}^{XZ} \\ \mathbf{v}_{11}^{XZ} - \mathbf{v}_{22}^{XZ} \\ \mathbf{v}_{12}^{YZ} \\ \mathbf{v}_{11}^{YZ} - \mathbf{v}_{22}^{YZ} \end{bmatrix} \hat{\mathbf{B}} = \mathbf{V} \hat{\mathbf{B}} = 0 \tag{2.34}$$

Donde podemos observar que la matriz \mathbf{V} tiene un tamaño 6×6 . Haciendo el mismo razonamiento que hicimos para la búsqueda de los valores de las matrices de homografía en la Sección 2.2.1, podemos resolver el sistema ya que vuelve a ser un sistema homogéneo lineal. Nuestro problema se puede formular de la siguiente forma [1]:

$$\begin{aligned}
& [\lambda, \phi] = \text{eig}(\mathbf{V}^T \mathbf{V}) \\
& [\lambda_{min}, i_{min}] = \min(\lambda) \\
& \hat{\mathbf{B}} = \phi(:, i_{min})
\end{aligned} \tag{2.35}$$

El último paso consiste en encontrar la relación entre las componentes de la matriz B y los parámetros intrínsecos de la cámara. Esto puede realizarse utilizando las siguientes fórmulas explícitas:

$$\begin{aligned}
n_y^O &= \frac{B_{12}B_{13} - B_{11}B_{23}}{B_{11}B_{22} - B_{12}^2} \\
\lambda &= B_{12} - \frac{B_{13}^2 + n_y^O(B_{12}B_{13} - B_{11}B_{23})}{B_{11}} \\
\alpha &= -\sqrt{\frac{\lambda}{B_{11}}} \\
\beta &= -\sqrt{\lambda B_{11}B_{11}B_{22} - B_{12}^2} \\
\gamma &= -\frac{B_{12}^2 \alpha^2 \beta}{\lambda} \\
n_x^O &= \frac{\gamma n_y^O}{\beta} - \frac{B_{13} \alpha^2}{\lambda}
\end{aligned} \tag{2.36}$$

Búsqueda de parámetros extrínsecos

A partir de la ecuación , los vectores unitarios del sistema del mundo se representan como:

$$\begin{aligned}
\bar{\mathbf{i}}_{XY}^{wd} &= e(\mathbf{M}^{int})^{-1} \mathbf{h}_1^{XY} \\
\bar{\mathbf{j}}_{XY}^{wd} &= e(\mathbf{M}^{int})^{-1} \mathbf{h}_2^{XY}
\end{aligned} \tag{2.37}$$

En las ecuaciones recién escritas, el superíndice XY representa que los vectores unitarios han sido calculados a partir de las matrices de homografía XY y e representa el factor de escala incógnita que puede ser solución de nuestra matriz de homografía, como hemos explicado en la Sección 2.2.1. Debido a que los vectores son unitarios, podemos calcular el valor de e como:

$$\begin{aligned}
e_1 &= \frac{1}{\|(\mathbf{M}^{int})^{-1} \mathbf{h}_1^{XY}\|} \\
e_2 &= \frac{1}{\|(\mathbf{M}^{int})^{-1} \mathbf{h}_2^{XY}\|}
\end{aligned} \tag{2.38}$$

En el caso ideal $e = e_1 = e_2$. Sin embargo, debido a que el cálculo de \mathbf{H}^{XY} se ha realizado haciendo una aproximación de mínimos cuadrados, en la práctica podemos comprobar que $e_1 \neq e_2$. Utilizando este último resultado, podemos calcular de nuevo los vectores unitarios del sistema del mundo:

$$\begin{aligned}
\bar{\mathbf{i}}_{XY}^{wd} &= \frac{(\mathbf{M}^{int})^{-1} \mathbf{h}_1^{XY}}{\|(\mathbf{M}^{int})^{-1} \mathbf{h}_1^{XY}\|} \\
\bar{\mathbf{j}}_{XY}^{wd} &= \frac{(\mathbf{M}^{int})^{-1} \mathbf{h}_2^{XY}}{\|(\mathbf{M}^{int})^{-1} \mathbf{h}_2^{XY}\|} \quad \bar{\mathbf{k}}_{XY}^{wd} = \bar{\mathbf{i}}_{XY}^{wd} \times \bar{\mathbf{j}}_{XY}^{wd}
\end{aligned} \tag{2.39}$$

El vector de posición del origen del sistema de referencia de la cámara 2.6 se representaría de la siguiente manera usando los últimos cálculos mostrados:

$$\bar{\mathbf{r}}_{XY}^{cam} = -e(\mathbf{M}^{int})^{-1} \mathbf{h}_3^{XY} = -\frac{e_1 + e_2}{2} (\mathbf{M}^{int})^{-1} \mathbf{h}_3^{XY} \tag{2.40}$$

Como puede apreciarse, hemos aproximado el factor de escala e como la media de los dos errores e_1 y e_2 definidos previamente. Utilizando estas expresiones, la matriz de parámetros extrínsecos de la cámara puede calcularse a partir de la matriz de homografía del plano XY como:

$$\mathbf{M}_{XY}^{\text{ext}} = [\bar{\mathbf{i}}_{XY}^{\text{wd}} \quad \bar{\mathbf{j}}_{XY}^{\text{wd}} \quad \bar{\mathbf{k}}_{XY}^{\text{wd}} \quad -\bar{\mathbf{r}}_{XY}^{\text{cam}}] = [(\mathbf{A}_{XY}^{\text{cam}})^T - \bar{\mathbf{r}}_{XY}^{\text{cam}}] \quad (2.41)$$

Debido a que la matriz de homografía viene de una aproximación, como hemos puntualizado anteriormente en esta sección, la resultante obtenida de $\mathbf{A}_{XY}^{\text{cam}}$ no es exacta y no cumple la propiedad de ortogonalidad de las matrices, es decir $(\mathbf{A}_{XY}^{\text{cam}})^T \neq (\mathbf{A}_{XY}^{\text{cam}})^{-1}$. Existe un proceso para solucionar este problema y poder obtener la "verdadera" matriz de rotación. Este proceso comienza con el cálculo de los autovalores de la matriz $\mathbf{A}_{XY}^{\text{cam}}$. Después, ya podemos calcular los valores corregidos. Para ello tenemos que calcular el producto de los autovalores izquierdos con los autovalores derechos de la matriz:

$$\begin{aligned} [U, S, V] &= \text{svd}(\mathbf{A}_{XY}^{\text{cam}}) \\ (\mathbf{A}_{XY}^{\text{cam}})_{\text{corrected}} &= UV \end{aligned} \quad (2.42)$$

Con un razonamiento similar, podemos obtener la matriz de parámetros extrínsecos para los otros planos $\langle X, Z \rangle$ e $\langle Y, Z \rangle$. Las fórmulas utilizadas para ello, análogas a las expuestas a lo largo de esta sección para el plano $\langle X, Y \rangle$, son:

$$\begin{aligned} \bar{\mathbf{i}}_{XZ}^{\text{wd}} &= \frac{(\mathbf{M}^{\text{int}})^{-1} \mathbf{h}_1^{\text{XZ}}}{\|(\mathbf{M}^{\text{int}})^{-1} \mathbf{h}_1^{\text{XZ}}\|} \\ \bar{\mathbf{k}}_{XZ}^{\text{wd}} &= \frac{(\mathbf{M}^{\text{int}})^{-1} \mathbf{h}_2^{\text{XZ}}}{\|(\mathbf{M}^{\text{int}})^{-1} \mathbf{h}_2^{\text{XZ}}\|} \\ \bar{\mathbf{j}}_{XZ}^{\text{wd}} &= \bar{\mathbf{i}}_{XZ}^{\text{wd}} \times \bar{\mathbf{j}}_{XZ}^{\text{wd}} \\ \bar{\mathbf{r}}_{XZ}^{\text{cam}} &= -\frac{e_1 + e_2}{2} (\mathbf{M}^{\text{int}})^{-1} \mathbf{h}_3^{\text{XZ}} \\ \mathbf{M}_{XZ}^{\text{ext}} &= [\bar{\mathbf{i}}_{XZ}^{\text{wd}} \quad \bar{\mathbf{j}}_{XZ}^{\text{wd}} \quad \bar{\mathbf{k}}_{XZ}^{\text{wd}} \quad -\bar{\mathbf{r}}_{XZ}^{\text{cam}}] = [(\mathbf{A}_{XZ}^{\text{cam}})_{\text{corrected}}^T - \bar{\mathbf{r}}_{XZ}^{\text{cam}}] \\ \bar{\mathbf{j}}_{YZ}^{\text{wd}} &= \frac{(\mathbf{M}^{\text{int}})^{-1} \mathbf{h}_1^{\text{YZ}}}{\|(\mathbf{M}^{\text{int}})^{-1} \mathbf{h}_1^{\text{YZ}}\|} \\ \bar{\mathbf{k}}_{YZ}^{\text{wd}} &= \frac{(\mathbf{M}^{\text{int}})^{-1} \mathbf{h}_2^{\text{YZ}}}{\|(\mathbf{M}^{\text{int}})^{-1} \mathbf{h}_2^{\text{YZ}}\|} \\ \bar{\mathbf{i}}_{YZ}^{\text{wd}} &= \bar{\mathbf{j}}_{YZ}^{\text{wd}} \times \bar{\mathbf{k}}_{YZ}^{\text{wd}} \\ \bar{\mathbf{r}}_{YZ}^{\text{cam}} &= -e (\mathbf{M}^{\text{int}})^{-1} \mathbf{h}_3^{\text{YZ}} = -\frac{e_1 + e_2}{2} (\mathbf{M}^{\text{int}})^{-1} \mathbf{h}_3^{\text{YZ}} \\ \mathbf{M}_{YZ}^{\text{ext}} &= [\bar{\mathbf{i}}_{YZ}^{\text{wd}} \quad \bar{\mathbf{j}}_{YZ}^{\text{wd}} \quad \bar{\mathbf{k}}_{YZ}^{\text{wd}} \quad -\bar{\mathbf{r}}_{YZ}^{\text{cam}}] = [(\mathbf{A}_{YZ}^{\text{cam}})_{\text{corrected}}^T - \bar{\mathbf{r}}_{YZ}^{\text{cam}}] \end{aligned} \quad (2.43)$$

Sin embargo, debemos obtener una solución única para \mathbf{M}^{ext} . Es por ello que debemos encontrar una versión unificada a partir de los tres resultados obtenidos, uno para cada plano de calibración. Este cálculo se realiza en el último paso del proceso de calibración.

Optimización de los parámetros intrínsecos y extrínsecos

La parametrización de la matriz de rotación se puede realizar utilizando diversas técnicas como pueden ser los ángulos de Euler, cuaterniones... Para este trabajo, sin embargo, es conveniente usar un método conocido como los parámetros Rodriguez, ya que esta formulación nos permite realizar el promedio entre rotaciones diferentes. Estos parámetros son definidos utilizando la rotación del ángulo α (γ en la Figura 2.11) y el vector unitario \mathbf{e} , que, según está definido en el teorema de rotaciones finitas de Euler, representan el ángulo y la

dirección del giro al que tenemos que someter al sistema de referencia del mundo para que este sea paralelo al de la cámara. Los parámetros están definidos tal que:

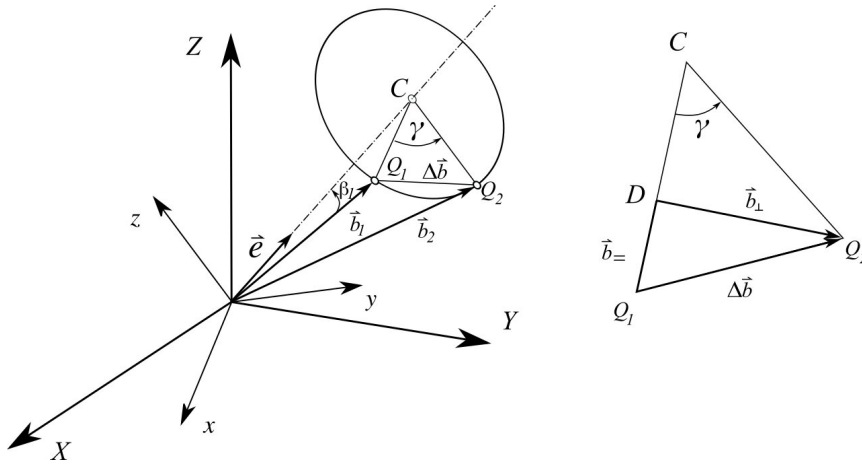


Figura 2.11 Esquema del teorema de Euler de las rotaciones finitas.

$$\mathbf{g}^{\text{cam}} = \tan\left(\frac{\alpha}{2}\right)\mathbf{e} \quad (2.44)$$

La matriz de rotación definida en función de los parámetros Rodriguez tiene la siguiente forma [1]:

$$\mathbf{A}^{\text{cam}} = \mathbf{I}_3 + \frac{2}{1 + (\mathbf{g}^{\text{cam}})^T \mathbf{g}^{\text{cam}}} (\mathbf{g}^{\text{cam}} + \mathbf{g}^{\text{cam}} \mathbf{g}^{\text{cam}}), \quad (2.45)$$

donde $\bar{\mathbf{g}}^{\text{cam}}$ es la matriz antisimétrica asociada a $\mathbf{g}^{\text{cam}} = [g_1^{\text{cam}} \ g_2^{\text{cam}} \ g_3^{\text{cam}}]^T$ construida como:

$$\bar{\mathbf{g}}^{\text{cam}} = \begin{bmatrix} 0 & -g_3^{\text{cam}} & g_2^{\text{cam}} \\ g_3^{\text{cam}} & 0 & -g_1^{\text{cam}} \\ -g_2^{\text{cam}} & g_1^{\text{cam}} & 0 \end{bmatrix} \quad (2.46)$$

La ecuación 2.45 puede ser invertida para poder realizar el cálculo de los parámetros:

$$\mathbf{g}^{\text{cam}} = \frac{1}{\text{trace}(\mathbf{A}^{\text{cam}})} \begin{bmatrix} A_{23}^{\text{cam}} - A_{32}^{\text{cam}} \\ A_{31}^{\text{cam}} - A_{13}^{\text{cam}} \\ A_{12}^{\text{cam}} - A_{21}^{\text{cam}} \end{bmatrix} \quad (2.47)$$

Si ahora utilizamos la representación de la orientación de las cámaras mediante cuaterniones [1] $(\mathbf{A}_{XY}^{\text{cam}})_{\text{corrected}}$, $(\mathbf{A}_{XZ}^{\text{cam}})_{\text{corrected}}$ y $(\mathbf{A}_{YZ}^{\text{cam}})_{\text{corrected}}$, corregidas con la ecuación 2.42, obtenemos las rotaciones $\mathbf{g}_{XY}^{\text{cam}}$, $\mathbf{g}_{XZ}^{\text{cam}}$ y $\mathbf{g}_{YZ}^{\text{cam}}$. A partir de estas expresiones podemos obtener una representación de una rotación combinada de la siguiente forma:

$$\mathbf{g}^{\text{cam}} = \frac{\mathbf{g}_{XY}^{\text{cam}} + \mathbf{g}_{XZ}^{\text{cam}} + \mathbf{g}_{YZ}^{\text{cam}}}{3} \quad (2.48)$$

Por último, podemos transformar esta expresión en función de los parámetros de Rodriguez a matriz de rotación ordinaria gracias a la ecuación 2.45, obteniendo la matriz de parámetros extrínsecos resultante. Gracias a esta optimización realizada, podremos obtener los valores óptimos de los parámetros intrínsecos y extrínsecos.

Este resultado óptimo será aquel que minimice el error cuadrático medio de todos los puntos del sistema de referencia del mundo proyectados en el sistema de la imagen con los puntos de nuestra imagen. Para realizar esta optimización podemos usar diversos métodos como Nelder-Mead o Levenberg-Marquadt, entre otros muchos. Tras realizar las pruebas determinamos que el mejor es el método de Limited-Memory BFGS ya que es el que menores errores nos daba [1].

Procedimiento de calibración antes de las inspecciones

Explicados los desarrollos teóricos necesarios para realizar la calibración, procedemos a realizar una calibración con el sistema montado en el vehículo. Una vez que hemos instalado el sistema hay que colocar, orientar y enfocar correctamente nuestra cámara para poder obtener una imagen con calidad y que nos proporcione información relevante del carril.

Una vez preparado el montaje de las cámaras y sensores que vamos a utilizar, procedemos a la calibración. Como se ha visto, la calibración parte de las posiciones conocidas de los puntos del calibrador en la imagen y su correspondencia con los puntos en el sistema de referencia del mundo. Para conseguir esto necesitaremos conocer la posición y orientación con respecto al TGMS en todo instante.

A partir de los modelos 3D podremos extraer los puntos del calibrador en el mundo y sus coordenadas, utilizando únicamente las fotos tomadas por nuestra cámara. Podemos observar que el calibrador está colocado de forma que sus ejes son paralelos a los del TGMS, lo que facilita la relación entre ambos, y permite obtener de manera directa la relación entre los puntos en el sistema de referencia del TGMS y sus correspondientes coordenadas en el sistema de referencia del mundo. Con este montaje también obtenemos las constantes que definen el plano de nuestro láser.

Una vez que hemos llevado a cabo todos los pasos de calibración expuestos en esta sección, ya somos capaces de transformar los puntos de cada imagen 2D a 3D, gracias a las matrices \mathbf{M}^{int} y \mathbf{M}^{ext} de cada cámara.

2.3 Modelado del perfil del raíl

En este apartado vamos a explicar el proceso y las expresiones matemáticas utilizadas para el cálculo del perfil que vamos a usar como referencia para calcular la posición del carril medido. Como ya hemos explicado, usaremos este carril teórico para utilizarlo como carril ideal para nuestro algoritmo. Recordamos que el objetivo definitivo de los algoritmos es encontrar la transformación afín que consigue que la nube de puntos de las medidas obtenidas pro la cámara se ajusten al perfil del carril ideal. De esta manera podremos estimar la posición en la que se encuentra el carril en cada punto kilométrico del trayecto y poder detectar irregularidades o tramos que no cumplan algunos de los estándares óptimos para los parámetros de calidad geométrica de la vía, explicados en la sección 1.2.

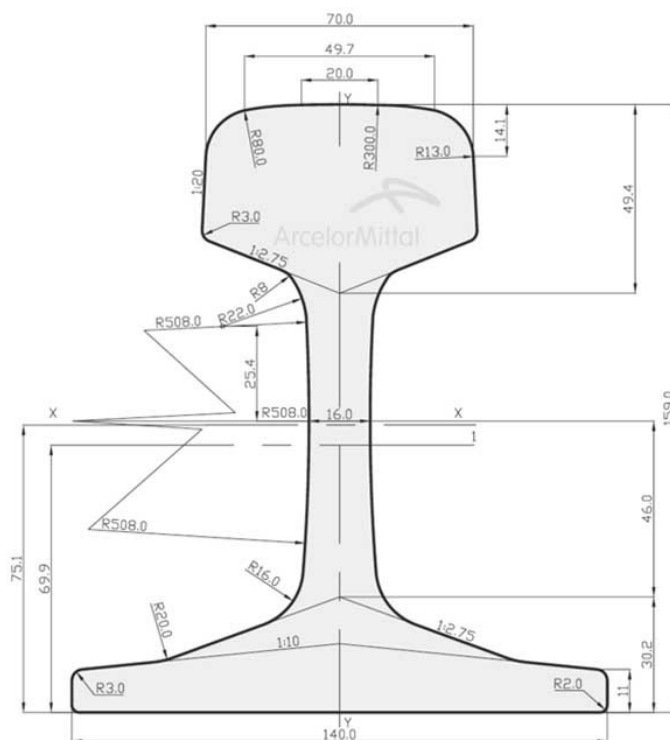


Figura 2.12 Perfil del carril UIC54. Estipulado por las normas UNE.

Según indica la norma [14] el perfil que usaremos será el UIC54, el cual podemos ver en la Figura 2.12. Las partes fundamentales desde arriba a abajo son: cabeza, alma y base del raíl.

Cómo se verá más adelante en el capítulo 3, la curva la vamos a necesitar representar como una nube de puntos y como una función analítica, dependiendo del algoritmo que usemos. Con la nube de puntos no habrá problemas en modelar el perfil completo, pero a la hora de tratar con funciones analíticas nos damos cuenta que solo podemos modelar una de las mitades por que si no la función sería multievaluada.

Esto podría suponer un problema si necesitaríamos computar la sección del carril completo. Sin embargo, de la manera en la que están colocados nuestros sensores, nosotros solo vamos a ver una de las mitades del perfil en cada cámara, por lo que con modelar la mitad derecha o izquierda será suficiente. Por ello podemos modelarla como una función y -proyectable. Para todos los cálculos de esta sección se harán para la mitad derecha del perfil UIC54 y usaremos la simetría respecto al eje vertical para poder alternar entre carril derecho e izquierdo.

Para poder representar la curva hemos decidido dividir el perfil (la mitad derecha) en los 14 tramos que refleja el plano del perfil: 4 rectas y 10 arcos de circunferencia como podemos observar en 2.13). Sin embargo,

el plano del perfil UIC54 (Figura 2.12) no proporciona explícitamente todos los datos necesarios para su modelado, por lo que habrá parámetros que sacaremos utilizando métodos matemáticos y de dibujo técnico. Además, hemos tenido que calcular los puntos de tangencia entre las curvas. Por último, es necesario recalcar que usamos como referencia para los ángulos la horizontal, el origen de las X es la recta vertical que divide el perfil en dos mitades simétricas y el origen del eje Y es el punto medio de la base del perfil, es decir, el punto en el que eje X y la recta inferior del carril se cortan.

Perfil del carril UIC54 subdividido en tramos

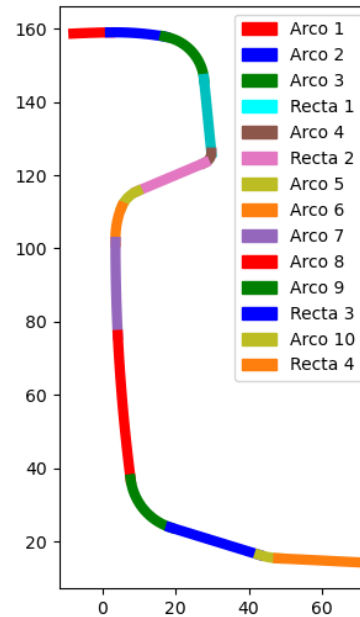


Figura 2.13 División por tramos del perfil UIC54.

Para cada tipo de curva (arco o recta) hemos creado una función que las traza, las cuales vamos a explicar brevemente a continuación.

2.3.1 Función trazadora de arcos

Esta función necesita que le pasemos como argumentos de entrada el centro (C), el radio (R), el rango de ángulo que quieres generar $([\alpha_0, \alpha_1])$ y el número de puntos que deseas que se generen.

Una vez que obtiene los parámetros aplica la siguiente fórmula:

$$arc = C + R * \begin{vmatrix} \cos \alpha \\ \sin \alpha \end{vmatrix}, \quad con : \quad \alpha = [\alpha_0, \alpha_1]$$

De esta forma la salida de nuestra función será un vector de puntos (x,y) que representan el arco definido.

2.3.2 Función trazadora de rectas

Esta función tiene como argumentos de entrada los puntos del eje x ($[x_0, x_1]$) que delimitan el segmento que queremos, los parámetros m y n característicos de nuestra recta y , al igual que para la anterior función 2.3.1, el número de puntos que queremos que formen parte de nuestro segmento.

Con estos parámetros traza la recta usando la propia definición de una recta:

$$y = m * x + n, \quad \text{con: } x = [x_0, x_1].$$

2.3.3 Tramos UIC54

Ahora que sabemos como funcionan las funciones que nuestro programa principal generador del perfil va a usar, comenzamos a explicar el proceso para la determinación de los parámetros de cada parte de la curva completa y sus puntos de tangencia. La nomenclatura de los parámetros con dos subíndices será la siguiente: el primero indica si es el comienzo (0) o el fin (1) de la curva y el segundo subíndice hará referencia a la curva a la que pertenece. Para los parámetros que solo tienen un subíndice, este hace referencia al arco o a la recta de la que forman parte.

Cabeza del raíl

-Arco 1:

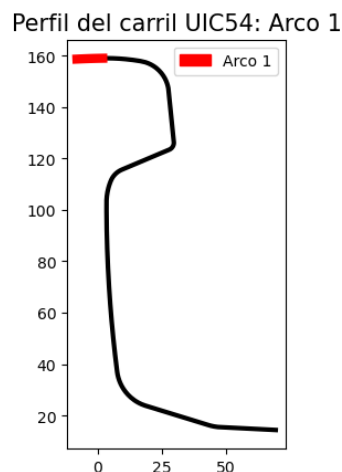


Figura 2.14 Arco 1 del perfil UIC54.

Observando el plano del perfil UIC54 podemos saber que $\alpha_{01} = 0rad$, $R_1 = 300$ mm y $C_1 = [0, 159 - 300]$ mm. Solo nos queda encontrar el final del rango de ángulos para poder trazarlo. Para ello, observando el dibujo y usando trigonometría deducimos que:

$$\alpha_{11} = \arccos \frac{\frac{1}{2} * D_1}{R}, \quad \text{con: } D_1 = d_1 = 20.$$

-Arco 2:

Perfil del carril UIC54: Arco 2

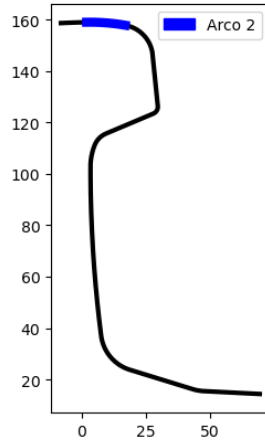


Figura 2.15 Arco 2 del perfil UIC54.

Para esta parte, tenemos $R_2 = 80 \text{ mm}$ y $\alpha_{02} = \alpha_{11}$ pero no disponemos del centro del arco ni del ángulo final. Para encontrar el centro, debemos darnos cuenta de que para las dos curvas sean tangentes, el centro debe encontrarse en la recta que une C_1 y el último punto del primer arco. Sabiendo eso, lo calculamos de la siguiente manera:

$$C_2 = C_1 + (R_1 - R_2) * \begin{vmatrix} \cos \alpha_{11} \\ \sin \alpha_{11} \end{vmatrix}$$

Para el cálculo de α_{12} volvemos a usar la trigonometría y lo determinamos con la siguiente fórmula:

$$\alpha_{12} = \arccos \left(\frac{D_2}{R_2} + \cos \alpha_{02} \right), \quad \text{con: } D_2 = \frac{1}{2} * (d_2 - d_1)$$

-Arco 3:

Perfil del carril UIC54: Arco 3

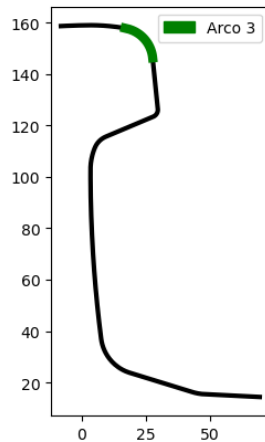


Figura 2.16 Arco 3 del perfil UIC54.

Para este último arco de la cabeza aplicamos el mismo razonamiento que para el arco 2. Conocemos $R_3 = 13$ mm, $\alpha_{03} = \alpha_{12}$ y tenemos que encontrar tanto C_3 como α_{13} . De nuevo sabemos que el centro debe encontrarse en la recta que une el centro anterior C_2 y el último punto del arco 2. Para encontrar α_{13} aplicamos el mismo razonamiento matemático. Las fórmulas para ello, análogas a las del apartado anterior son:

$$C_3 = C_2 + (R_2 - R_3) * \left| \frac{\cos \alpha_{12}}{\sin \alpha_{12}} \right|$$

$$\alpha_{13} = \arccos \left(\frac{D_3}{R_3} + \cos \alpha_{03} \right), \quad \text{con: } D_3 = \frac{1}{2} * (d_3 - d_2)$$

-Arco 4:

Perfil del carril UIC54: Arco 4

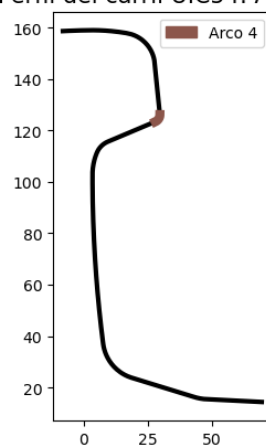


Figura 2.17 Arco 4 del perfil UIC54.

El arco 4 es el que va entre las dos rectas de la cabeza. Lo calculamos antes porque debido a los datos que se nos proporcionan, son necesarios los puntos de tangencia para delimitar los segmentos de las rectas 1 y 2 que pertenecen al perfil. Para este cálculo, y sabiendo que $R_4 = 3$ mm, hemos creado un sistema de ecuaciones que nos calculará los puntos finales de ambas rectas que hacen que exista un arco con dicho radio que haga tangencia con ambas. La nomenclatura será de la siguiente manera:

- Los subíndices 1 y 2 se refieren a las rectas 1 y 2 respectivamente.
- Los Δ_1 y Δ_2 representan lo que hay que sumar a los puntos finales de las rectas para llegar al centro del arco.
- Los ángulos β_1 y β_2 son los ángulos de las direcciones perpendiculares a ambas rectas, puesto que sabemos que el punto de tangencia debe encontrarse en una dirección normal a las rectas con las que estamos trabajando. Sus valores son: $\beta_1 = \frac{1}{20} + \frac{\pi}{2}$ y $\beta_2 = -\frac{1}{2.75}$
- Los puntos x_1, y_1, x_2, y_2 son las incógnitas de nuestro sistema y se corresponden a ese teórico punto final que estamos buscando.

Definimos los valores de los incrementos como:

$$\Delta_{x1} = -3 * \sin \beta_1, \quad \Delta_{y1} = -3 * \cos \beta_1, \quad \Delta_{x2} = -3 * \sin \beta_2, \quad \Delta_{y2} = 3 * \sin \beta_2$$

Y el sistema de ecuaciones a resolver sería el siguiente:

$$\begin{cases} x_1 + \Delta_{x1} = x_2 + \Delta_{x2} \\ y_1 + \Delta_{y1} = y_2 + \Delta_{y2} \end{cases}$$

Una vez que tenemos los puntos de tangencia, los usamos para calcular la posición del centro del arco 4. Dicho punto tiene que ser, por definición de punto de tangencia, el siguiente:

$$C_4 = \begin{pmatrix} x_1 + \Delta_{x1} \\ y_1 + \Delta_{y1} \end{pmatrix}$$

Y los ángulos de inicio y fin del arco son:

$$\alpha_{04} = -\frac{\pi}{2} + \beta_2, \quad \alpha_{14} = \frac{\pi}{2} + \beta_1$$

-Recta 1:

Perfil del carril UIC54: Recta 1

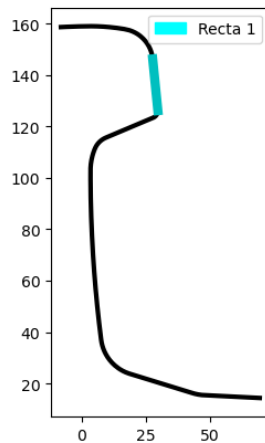


Figura 2.18 Recta 1 del perfil UIC54.

Observando el plano proporcionado en la Figura 2.12 podemos ver que la pendiente de nuestra recta es $m = -\frac{1}{20}$ y los puntos de inicio y final del segmento son los que se corresponden con el último punto del arco 3 y el punto de tangencia entre la recta 1 y el arco 4. Esos datos son suficientes para determinar nuestra recta gracias a la función descrita en 2.3.2.

Siguiendo el orden lógico ahora deberíamos describir el trazado de la recta 2, pero no es posible puesto que no conocemos el otro punto necesario, el punto de tangencia con el arco 5. Para conocerlo debemos conocer los arcos 5, 6 y 7 que vienen a continuación, ya que, una vez resueltos, nos permitirán calcular la tangencia entre la recta 2 y el arco 5. Debido a los datos de los que disponemos vamos a empezar por el arco 5 y 6, después la línea 2 y ,por último, el arco 7.

-Arco 5:

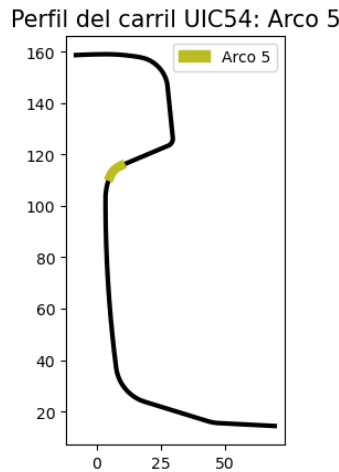


Figura 2.19 Arco 5 del perfil UIC54.

Aunque el arco 5 y el 6 podrían considerarse parte del alma del raíl, lo hemos metido en la parte de la cabeza porque, como ya se ha explicado, los datos de estos dos arcos son necesarios para calcular la recta 2, que, indudablemente, pertenece a la cabeza. Es por ello que consideraremos a efectos de este capítulo, que el alma del carril comienza en el arco 7.

Para calcular el centro C_5 vamos a usar una técnica muy utilizada en dibujo técnico para encontrar puntos de tangencia y centros de las circunferencias tangentes. Observando el plano del perfil podemos determinar que C_5 se encontrará en la intersección entre una recta paralela a la recta 2 separado a una distancia igual a R_5 y una circunferencia con centro C_6 de radio $R_{aux} = R_6 - R_5$. A su vez, para calcular la posición de C_6 necesitaremos algunos valores pertenecientes al arco 7. Los valores y las expresiones necesarias para calcular dicho centro son las siguientes:

$$\begin{cases} R_5 = 8 \text{ mm}, & R_6 = 22\text{mm}, & R_7 = 508\text{mm} \\ \alpha_{06} = \alpha_{17} = -\arcsen \frac{25.4}{R_7 - R_6} + \pi \\ C_7 = \begin{bmatrix} 8+R_7 \\ 30.2+46 \end{bmatrix} \end{cases}$$

$$C_6 = C_7 + (R_7 - R_6) * \begin{bmatrix} \cos \alpha_{17} \\ \sin \alpha_{17} \end{bmatrix}$$

Podemos observar que el valor de 25.4 mm lo sacamos del plano del UIC54

Una vez que tenemos C_6 , ya solo necesitaríamos calcular C_5 y el rango de ángulos para poder trazar el arco 5. Para calcular el valor del centro del arco 5 tenemos que resolver el sistema de ecuaciones que explicamos anteriormente: una recta paralela a la línea 2 (y_{aux} con $m_{aux} = \frac{1}{2.75}$ y $n_{aux} = (159 - 49.4) \sqrt{R_5^2 + (R_5 m_{aux})^2}$)

a una distancia de 8 mm y la circunferencia auxiliar de radio la resta entre R_6 y R_5 ($R_{aux} = R_6 - R_5$). Se expresa matemáticamente de la siguiente manera:

$$\begin{cases} y = m_{aux}x + n_{aux} \\ (x - C_{6x})^2 + (y - C_{6y})^2 = R_{aux}^2 \end{cases}$$

Ya solo nos falta calcular el rango de ángulo para tener completamente definido el arco. Las expresiones de dichos ángulos se pueden sacar de manera intuitiva a partir del plano.

$$\begin{cases} \alpha_{05} = \pi - \arctan 2.75 \\ \alpha_{15} = \pi + \arctan \frac{C_{6y} - C_{5y}}{C_{6x} - C_{5x}} \end{cases}$$

Con estos datos ya podemos usar nuestra función trazadora de arcos para definir completamente la nube de puntos que representará nuestro arco 6 en el perfil ideal del UIC54.

-Arco 6:

Perfil del carril UIC54: Arco 6

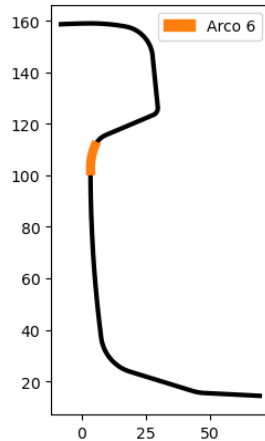


Figura 2.20 Arco 6 del perfil UIC54.

Para este arco no vamos a necesitar hacer muchos más cálculos porque los realizados para el arco 5 se usan ahora también. Lo único que si será necesario calcular es el ángulo final de dicho arco, que corresponde a la siguiente expresión:

$$\alpha_{16} = \arctan \frac{C_{6y} - C_{7y}}{C_{6x} - C_{7x}}$$

-Recta 2:

Perfil del carril UIC54: Recta 2

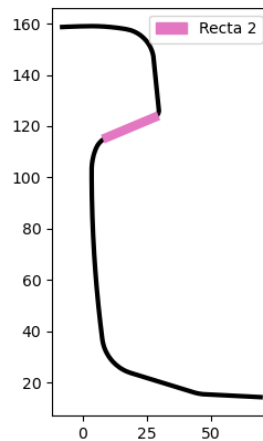


Figura 2.21 Recta 2 del perfil UIC54.

Si recordamos, a la hora de modelar la cabeza del raíl habíamos dejado la recta 2 sin definir ya que no teníamos el punto final de la recta, que correspondía al punto de tangencia entre esta recta y el arco 5. Una vez que ya tenemos definido el arco 5 podemos trazar la recta sabiendo que el punto inicial es el punto de tangencia con el arco 4 y el punto final el punto de tangencia con el arco 5. Los parámetros de la recta son :

$$\begin{cases} m_2 = \frac{1}{2.75} \\ n_2 = 159 - 49.4 \end{cases}$$

Ya con la definición de la recta 2 podemos dar por concluido el modelado de la cabeza de la sección del raíl.

Alma del raíl

Una vez que hemos definido completamente la cabeza del perfil del raíl, comenzamos a modelar el alma, que es la parte intermedia entre cabeza y base y se corresponde a la parte más delgada de nuestra sección.

-Arco 7:

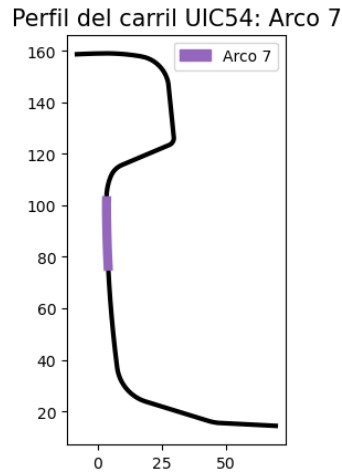


Figura 2.22 Arco 7 del perfil UIC54.

Debido a los cálculos que ya hemos definido antes como necesarios para calcular los parámetros de las curvas pertenecientes a la cabeza del raíl, en este apartado solo necesitaremos definir el rango de ángulos de este arco para poder pasárselo como argumento a nuestra función.

Dichos ángulos son los siguientes:

$$\begin{cases} \alpha_{07} = \pi - \arcsin \frac{25.4}{R_7} \\ \alpha_{17} = \pi \end{cases}$$

A continuación deberíamos calcular el arco 8, que se corresponde con la parte inferior del alma. Sin embargo, no disponemos de su punto final puesto que se trata del punto de tangencia con el arco 9, el cual no puede ser definido sin definir previamente la base, debido a los datos de los que disponemos. Es por ello, que vamos a pasar a la sección de la base del raíl y es allí donde definiremos el arco 8, aunque no pertenezca a la base del raíl, sino al alma.

Base del raíl

-Arco 9:

Perfil del carril UIC54: Arco 9

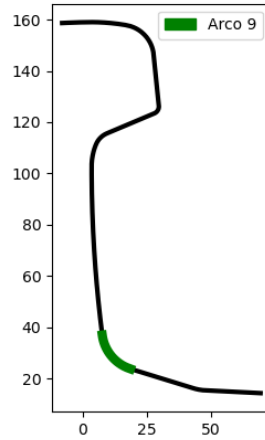


Figura 2.23 Arco 9 del perfil UIC54.

Para calcular este nuevo arco debemos primero encontrar su centro C_9 . Si observamos el plano del que disponemos, podemos deducir que dicho centro se encontrará en la intersección de una recta auxiliar paralela a la recta 3 a una distancia igual a R_9 , que es igual a 16 mm, y una circunferencia auxiliar de radio $R_{aux} = 508 - 16$ mm y centro $C_7 = C_8$. Para resolver este necesitamos definir una serie de parámetros que son los siguientes.

$$\begin{cases} m_{aux} = -\frac{1}{2.75} \\ n_{aux} = 30.2 + \sqrt{16^2 + (16(\frac{1}{2.75}))^2} \\ x_{c_{aux}} = 508 + 8 \\ y_{c_{aux}} = 46 + 30.2 \\ R_{c_{aux}} = 508 - 16 \end{cases}$$

Una vez definidos estos parámetros resolvemos el sistema:

$$\begin{cases} y = m_{aux}x + n_{aux} \\ (x - x_{c_{aux}})^2 + (y - y_{c_{aux}})^2 = R_{c_{aux}}^2 \end{cases} \quad (2.49)$$

Ya tenemos el centro C_9 pero no conocemos los ángulos necesarios para definir el arco. El ángulo final, en este caso el ángulo en el que se produce la tangencia con la recta 3, tiene la siguiente expresión:

$$\alpha_{19} = -\arctan(-2.75) + \pi$$

Para el otro ángulo, debemos volver a aplicar técnicas de dibujo técnico. Viendo el plano proporcionado podemos observar que dicho punto de tangencia tiene que encontrarse en la intersección del arco 8 y una recta

que pase por los centros C_8 y C_9 . Para encontrar dicho punto debemos resolver este sistema de ecuaciones, siendo m_{aux} la pendiente de esa recta que pasa por ambos centros y que tiene la forma $m_{aux} = \frac{C_{9y} - C_{8y}}{C_{9x} - C_{8x}}$:

$$\begin{cases} y = m_{aux}x + n_{aux} \\ (x - C_{9x})^2 + (y - C_{9y})^2 = R_9^2 \end{cases}$$

La posición del punto de tangencia la hemos hallado resolviendo el sistema de ecuaciones anterior. La solución que obtenemos conforma el punto $P_{tang} = [x_{tang}, y_{tang}]$ que es el punto de tangencia con el arco 8. Ya solo nos queda encontrar el ángulo para el que se da dicha tangencia que, haciendo un desarrollo trigonométrico podemos deducir que es:

$$\alpha_{09} = 2\pi + \arctan \frac{y_{tang} - C_{9y}}{x_{tang} - C_{9x}}$$

Con todos estos datos ya podemos describir la curva de este arco y generar los puntos que formarán parte de

nuestra nube de puntos del perfil UIC54.

Para esta primera tangencia usamos el mismo razonamiento que para calcular la tangencia entre el arco 4 y la recta 1. Procedemos pues a definir los parámetros que usaremos para calcular el punto final de la recta 3, que será el punto de tangencia que buscamos.

-Arco 8:

Perfil del carril UIC54: Arco 8

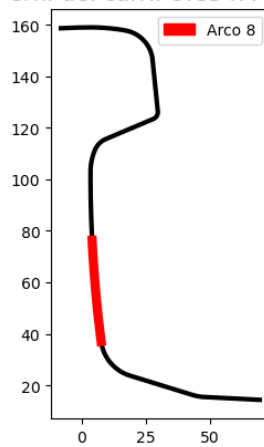


Figura 2.24 Arco 8 del perfil UIC54.

Con lo calculado en el apartado anterior ya podemos generar el arco 8 que completaría el alma de nuestro carril. Esto es posible porque $\alpha_{18} = \alpha_{09}$ y $\alpha_{08} = \pi$.

-Recta 3:

Perfil del carril UIC54: Recta 3

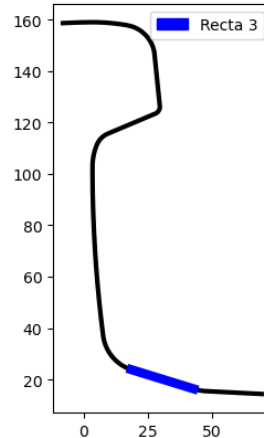


Figura 2.25 Recta 3 del perfil UIC54.

Las 3 últimas partes de nuestro perfil están relacionadas a la hora de calcularlas. Es necesario calcular los puntos de tangencia del arco 10 con ambas rectas para poder definir los intervalos de x en los que existen nuestras rectas. Para ello tenemos que hacer un planteamiento similar al realizado para calcular los puntos de tangencia del arco 4 con las rectas 1 y 2. Por ello, debemos definir los siguientes parámetros :

$$\beta_3 = \pi - \frac{1}{2.75}, \quad \beta_4 = \pi - \frac{1}{10}$$

$$R_{10} = 20$$

$$\Delta_{x3} = R_{10} \sin \beta_3, \quad \Delta_{y3} = -R_{10} \cos \beta_3, \quad \Delta_{x4} = R_{10} \sin \beta_4, \quad \Delta_{y4} = -R_{10} \cos \beta_4$$

Y el sistema de ecuaciones a resolver sería el siguiente:

$$\begin{cases} x_3 + \Delta_{x3} = x_4 + \Delta_{x4} \\ y_3 + \Delta_{y3} = y_4 + \Delta_{y4} \end{cases}$$

Conociendo los puntos de tangencia tanto de la recta 3 (x_{13}) como de la recta 4 (x_{04}) calculados en el sistema anterior, podemos trazarlas. El punto de inicio de x_3 es:

$$x_{03} = C_{9y} + \Delta_{x3}$$

-Recta 4:

Perfil del carril UIC54: Recta 4

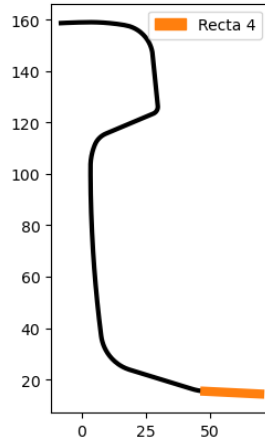


Figura 2.26 Recta 4 del perfil UIC54.

Gracias a los resultados obtenidos en el apartado de la recta 3, solo necesitamos el punto final del intervalo. Observando el plano podemos ver que $x_{14} = 70$.

-Arco 10:

Perfil del carril UIC54: Arco 10

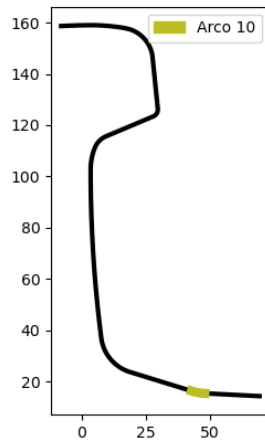


Figura 2.27 Arco 10 del perfil UIC54.

Por último, para finalizar esta sección del documento, pasamos a definir los intervalos de ángulos en los que definimos nuestro arco 10. Para ello solo debemos observar el plano de la Figura 2.12 puesto que los valores son inmediatos observando los valores de las pendientes de las rectas con las que tiene que hacer tangencia. Sus expresiones son:

$$\alpha_{010} = -\frac{\pi}{2} - \frac{1}{2.75}$$

$$\alpha_{110} = -\frac{\pi}{2} - \frac{1}{10}$$

Con este último arco ya podemos dar por terminada la definición de la nube de puntos que usaremos para modelar nuestro perfil ideal. Como podemos observar nos faltaría por modelar la parte inferior de la base del perfil, pero esos puntos nunca aparecen en nuestras medidas, por lo que no será necesario modelarlos para usar nuestros algoritmos.

Este modelo matemático lo hemos implementado en Python en una función llamada puntos UIC54. Esta función toma como parámetros los tramos que deseamos generar (de los 14 recientemente explicados) y el número de puntos que queremos en cada uno de los tramos. Por defecto el número de puntos por tramos lo hemos definido como 25. La función nos devuelve una nube de puntos que representan el perfil de la sección del carril, que usaremos para computar la transformación óptima de las medidas.

3 Algoritmos de optimización

En este apartado procedemos a explicar los tres algoritmos desarrollados e implementados para resolver el problema propuesto. Dicho problema consiste en encontrar la transformación espacial afín que hace que nuestras medidas se ajusten mejor al carril de referencia modelado. Esta transformación consiste en una rotación \mathbf{R} y una traslación \mathbf{t} que definiremos más adelante.

De los algoritmos propuestos, hay dos que se tratan de una optimización utilizando el método Nelder-Mead con la función `optimize` de la librería `numpy` de Python. Se trata del algoritmo 1 y 3. Por otra parte el algoritmo 3 es el Iterative Closest Point (ICP). Para estos tres algoritmos hemos usado la función `puntos UIC54` explicada en la Sección 2.3.3 para usarlo como perfil ideal con el que compararemos las medidas obtenidas por las cámaras y el láser de nuestro sistema embarcado `vmRail`. Cada uno de los algoritmos tiene sus peculiaridades que se tratarán de explicar para poder diferenciarlos.

El principio básico de los tres algoritmos es la búsqueda de la transformación espacial afín que hace que se ajusten mejor nuestros puntos de medidas al carril ideal que representa al perfil `UIC54`. Esta transformación es una composición de una traslación y una rotación. El objetivo de dichos algoritmos es la de minimizar la función objetivo o función de coste que definiremos más adelante.

Además se presentan los resultados que se obtienen con cada uno de ellos. Para ellos realizamos dos tandas de pruebas diferentes. Primero, ejecutamos los algoritmos con bancos de pruebas generados artificialmente para simular las medidas que obtendríamos con nuestros sensores. De esta manera podemos predecir el comportamiento que tendrán en la aplicación real y comprobar cómo puede afectar al resultado la cantidad, la distribución y la calidad de dichos datos medidos. Posteriormente, se probarán los algoritmos con datos reales obtenidos de mediciones realizadas por `Virtualmechanichs S.L.` Por último, se comparan los resultados para decidir cuál es el algoritmo óptimo que pueda ser útil para esta aplicación. Para ello nos basaremos en la estabilidad, robustez y velocidad de resolución, sin olvidarnos de la validez de la solución obtenida.

3.1 Explicación teórica de los algoritmos

A continuación, presentamos una breve explicación teórica de cada uno de los algoritmos y los fundamentos matemáticos usados. Para ello vamos a necesitar explicar una serie de conceptos que se utilizarán en algunos o en todos los algoritmos.

La primera, y posiblemente una de las más importantes, es la de la matriz de transformación espacial a la que denominaremos \mathbf{M} [21]. Esta transformación se denomina transformación afín. Está compuesta por una matriz de rotación \mathbf{R} y un vector de traslación \mathbf{t} que se definen de la siguiente manera.

$$\mathbf{R} = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix}, \quad \mathbf{t} = \begin{bmatrix} tx \\ ty \end{bmatrix}$$

Para poder aplicar las transformaciones simultáneamente se crea la matriz de transformación \mathbf{M} antes mencionada, que tiene la siguiente forma:

$$\mathbf{M} = \begin{bmatrix} \cos \theta & -\sin \theta & tx \\ \sin \theta & \cos \theta & ty \\ 0 & 0 & 1 \end{bmatrix}$$

La última fila de la matriz \mathbf{M} es necesaria cuando trabajas con un conjunto de puntos en coordenadas homogéneas, como es el caso. La coordenada (3,3) se corresponde a la escala que se le quiera aplicar a nuestro conjunto de puntos. Para el caso de estudio se aplica un escalado de 1, aunque podría estudiarse si es necesario un cambio de escala a nuestras medidas para que se ajusten mejor al carril ideal. Este cambio de escala podría venir motivado por errores en la medida y distorsiones provocadas por el movimiento del tren. Sin embargo, a la vista de los datos de los que se disponen para este estudio, esto no parece necesario.

Como para cualquier problema de optimización, necesitaremos una función de coste que minimizar, la cual nos servirá para poder evaluar la validez de nuestra solución óptima y nos permitirá compararla con otras durante la resolución del problema. Nosotros hemos decidido usar el error cuadrático medio de las distancias entre la nube de medidas, que denominamos \mathbf{A} , y la nube de puntos que representan el carril ideal, \mathbf{B} .

Para computar esta distancia es necesario emparejar los puntos de ambas nubes. Esto se hará siguiendo el criterio del punto más cercano, aunque se podrían haber escogido otros criterios [22]. Las nubes de puntos son de diferente tamaño, ya que tenemos más puntos de medidas que los que usamos para modelar el carril. Los emparejamientos no son restrictivos, es decir puede haber dos puntos de las medidas emparejados con el mismo de la nube de puntos del carril ideal. Esto hace que haya puntos que no se emparejen con ninguno y estos puntos no se contabilizarán a la hora de calcular las distancias entre las nubes.

La función de coste que usamos se formula matemáticamente de la siguiente forma:

$$J = \sum_{i=1}^n \sqrt{(b_{ix} - a_{ix})^2 + (b_{iy} - a_{iy})^2}, \quad n: \text{número de parejas A-B}$$

Como se ha explicado anteriormente, trabajamos con dos nubes de puntos de distintos tamaños definidos de la siguiente manera :

$$\mathbf{A} = \begin{bmatrix} a_{1x}, & a_{1y} \\ a_{2x}, & a_{2y} \\ \vdots & \vdots \\ a_{n_a x}, & a_{n_a y} \end{bmatrix} \quad (3.1)$$

$$\mathbf{B} = \begin{bmatrix} b_{1x}, & b_{1y} \\ b_{2x}, & b_{2y} \\ \vdots & \vdots \\ b_{n_b x}, & b_{n_b y} \end{bmatrix} \quad (3.2)$$

Cabe destacar que n_a y n_b representan la cantidad de puntos que tenemos en la nube de las medidas y en la nube de puntos del carril ideal, respectivamente.

3.1.1 Algoritmo 1: Optimización con dos nubes de puntos

El primer algoritmo usa la nube de puntos de la medida **A** (ecuación 3.1) y la nube de puntos de nuestro perfil ideal **B** (ecuación 3.2) y las compara. Usando una función de optimización basándose en el método de Nelder-Mead trata de buscar la matriz de transformación que hace que nuestras medidas se ajusten mejor al carril ideal.

A nuestro algoritmo le pasamos ambas nubes de puntos y él trata de alcanzar el mínimo de la función de coste, en este caso el error cuadrático medio entre las parejas de puntos. Los puntos se emparejan por proximidad usando una función de búsqueda de vecinos más próximos (Closest Neighbours) de la librería *sklearn* de Python. Esta función empareja los puntos que están a menor distancia, pero puede emparejar varios puntos de las medidas con el mismo punto del carril ideal, lo que podría provocar que el algoritmo se quedara atascado en un mínimo local si la condición inicial no fuera suficientemente buena.

Partiendo de una condición inicial, nuestro algoritmo va acercándose iterativamente a la solución óptima hasta que alcanza los requisitos mínimos de error máximo. Es decir, el programa calcula la transformación que produce menos error entre parejas de puntos, la aplica a los puntos de medida y vuelve a calcular el mínimo de la función de coste. En cada iteración se calculan de nuevo las parejas de puntos que se van a comparar. Progresivamente nuestros puntos irán ajustándose al raíl ideal.

3.1.2 Algoritmo 2: ICP

El segundo algoritmo es un algoritmo que ya ha sido usado para aplicaciones parecidas a la que nos atañe y en otros muchos campos. El ICP (Iterative Closest Point) [10, 11] es un algoritmo utilizado para minimizar la distancia entre dos nubes de puntos basándose en álgebra lineal. Se ha utilizado para diversos fines como puede ser reconstrucción 3D de escáneres cerebrales o para localizar robots basándose en el mapeado que hagan de un entorno. Los pasos que sigue el algoritmo son los siguientes:

- Para cada punto de nuestra medida busca el vecino más próximo de nuestra nube de puntos de referencia (carril ideal UIC54). Para ello usa la función `ClosestNeighbours` de la librería *sklearn* explicada en el apartado anterior 3.1.1.
- Estima la combinación de rotación y translación que mejor alinea nuestras nubes de puntos, minimizando el error entre los pares encontrados en el paso anterior.
- Transforma los puntos con la matriz de transformación.
- Vuelve a iterar. Esto se repetirá hasta que se cumpla el requisito de tolerancia o error máximo admitido o si se supera el número de iteraciones permitido.

Básicamente, es un algoritmo que, iterativamente, va acercándose a la solución que mejor ajusta las dos nubes de puntos. Sin embargo esa solución puede no ser la óptima, ya que en algunos casos el algoritmo puede tender a un mínimo local del cual no es capaz de salir.

A continuación vamos a explicar mejor las fórmulas y métodos matemáticos que se utilizan para desarrollar este algoritmo [10]. Para ello, primero necesitamos explicar la notación que usaremos para referirnos a los elementos que intervienen.

- Nuestro conjunto de puntos **A** se corresponde con la nube de puntos de medidas de nuestro sistemas embarcados en el vehículo y la nube de puntos **B** es el modelo del perfil UIC54 explicado en el Capítulo 2 de este texto.
- Llamaremos **R** a la matriz de rotación y **t** al vector de traslación (x,y) que aplicaremos a nuestros puntos de medidas en cada iteración.

Una vez que conocemos la notación que usaremos en este desarrollo procedemos a explicar con más detalle cada uno de los pasos del algoritmo.

En el primer paso tenemos que buscar las parejas de puntos para computar la distancia que intentaremos minimizar. Estas parejas solo son válidas para una iteración, ya que una vez que se les aplica la transformación a los puntos de nuestra nube \mathbf{A} , será necesario calcular de nuevo las parejas de puntos. Para realizar estos emparejamientos hay distintas estrategias, ya sea a la hora de elegir qué puntos de la nube queremos tener en cuenta, como el algoritmo usado para emparejarlos. Nosotros hemos decidido usar todo el conjunto de las medidas para esta tarea, ya que no tenemos ninguna manera de darle más peso a unos puntos que a otros. Si conociéramos que uno de los puntos se corresponde a un punto característico del raíl ideal, el vértice inferior lateral de la cabeza, por ejemplo; podríamos forzar que ese emparejamiento se realizara siempre con dicho punto del carril ideal y ponderar más a esa pareja a la hora de calcular la distancia. Esto podría mejorar la convergencia de nuestro algoritmo y reducir el tiempo computacional necesario. Hay que tener en cuenta, que este proceso tendrá que realizarse para cada imagen que obtengamos durante el viaje del tren, por lo que una pequeña reducción del tiempo de ejecución en una imagen podría suponer una mejora sustancial a la hora de procesar un viaje completo.

Para el algoritmo de emparejamiento hemos usado la función implementada en Python de `ClosestNeighbours`. Esta función viene dentro de la librería `sklearn` que proporciona numerosas técnicas basadas en Machine Learning para la resolución de diversos problemas de Clustering, ordenamiento y muchos más.

Por lo tanto, atendiendo a las premisas que hemos definido, nuestro conjunto de parejas de puntos \mathbf{C} quedaría formulado de la siguiente manera:

$$\mathbf{C} = \{(i,j) | a_i \in \mathbf{A} \text{ y } b_j \in \mathbf{B} \text{ son puntos emparejados}\}$$

Esta operación es una de las que más recursos computacionales consume por lo que es muy importante elegir adecuadamente tanto el algoritmo de emparejamiento como el conjunto de puntos de nuestra nube \mathbf{A} que usamos.

A continuación, podríamos aplicar un filtro para rechazar aquellos puntos que pensamos que están demasiado lejos como para ser importantes a la hora del cálculo de la distancia. Para ello podríamos determinar una distancia máxima aceptable y desechar aquellas parejas que la superen. Esta distancia máxima podríamos sacarla de manera experimental, por ejemplo, observando los resultados que obtenemos en las pruebas y analizando cómo afectaría dicha distancia máxima a la convergencia del algoritmo. Nosotros, sin embargo, no hemos aplicado este filtro ya que estamos analizando desviaciones muy pequeñas del raíl por lo que sería difícil definir dicho umbral de distancia.

El segundo paso del algoritmo ICP consiste en calcular la transformación espacial (rotación y traslación) que alinea nuestras nubes de puntos. Debido a que usamos la suma de errores cuadráticos medios como medida de lo bueno que es nuestro algoritmo podemos definir nuestro problema de minimización de la siguiente manera:

$$(\mathbf{R}^*, \mathbf{t}^*) = \min_{\mathbf{R}, \mathbf{t}} \sum_{(i,j) \in \mathbf{C}} \|b_j - (\mathbf{R}^* a_i - \mathbf{t})\|^2$$

Para la resolución de este problema se han estudiado hasta 4 técnicas diferentes [23]. Nosotros hemos elegido el algoritmo basado en descomposición de valores singulares (SVD) por sus buenos resultados en términos de velocidad, precisión y estabilidad. Para llevarlo a cabo, primero tenemos que definir los centros de gravedad de ambas nubes de puntos, teniendo en cuenta que $|\mathbf{C}|$ es el número de parejas creadas, de la siguiente manera:

$$\bar{a} = \frac{1}{|\mathbf{C}|} \sum_{(i,j) \in \mathbf{C}} a_i, \quad \bar{b} = \frac{1}{|\mathbf{C}|} \sum_{(i,j) \in \mathbf{C}} b_i$$

Centralizando nuestras nubes de puntos en función de su centro de gravedad conseguimos extraer la traslación del problema de minimización aligerando bastante la carga computacional del algoritmo. Con esta transformación el problema de minimización antes descrito queda de la siguiente manera:

$$\mathbf{R}^* = \min_{\mathbf{R}} \sum_{(i,j) \in \mathbf{C}} \|(b_j - \bar{b}) - \mathbf{R}(a_i - \bar{a})\|^2$$

Este problema puede resolverse haciendo la SVD (Singular Values Decomposition) de la matriz siguiente y definiendo $\mathbf{R}^* = \mathbf{U}\mathbf{V}^T$:

$$K = \sum_{(i,j) \in \mathbf{C}} (b_j - \bar{b}) * (a_i - \bar{a})^T = \mathbf{U}\mathbf{D}\mathbf{V}^T$$

Es posible que \mathbf{R}^* no se trate de una rotación pura, sino que incluya también una reflexión. Este caso particular se soluciona fácilmente multiplicando la tercera columna de la matriz \mathbf{U} por -1. Finalmente, una vez calculada la rotación \mathbf{R}^* el vector de traslación óptimo se computa de la siguiente forma:

$$\mathbf{t}^* = \bar{b} - \mathbf{R}^* \bar{a}$$

Nuestro algoritmo se detendrá una vez que el cambio de error de una iteración a otra sea inferior a un umbral especificado por nosotros. Además, para evitar un posible bucle infinito en el caso de divergencia de nuestro algoritmo, es bueno definir un umbral máximo también para el número de iteraciones.

Como vemos el algoritmo es parecido al primero, explicado en la Sección 3.1.1, en cuanto al orden y los pasos previos al cálculo de la transformación espacial óptima, pero es más rápido debido a que el álgebra lineal requiere menos coste computacional que las funciones de optimización utilizadas en los otros algoritmos.

3.1.3 Algoritmo 3: Optimización con función explícita

El último algoritmo que se presenta es el más complejo. En los dos anteriores usábamos dos nubes de puntos para calcular las transformaciones necesarias. El hecho de usar nubes de puntos nos ofrecía flexibilidad a la hora de elegir el número de puntos que queríamos usar para la referencia y los tramos de dicho perfil que queríamos. Pero esa flexibilidad también nos plantea un problema: al ser una nube de puntos la precisión de la curva depende de la cantidad de puntos que cojas, ya que cuantos menos puntos haya, más separación hay entre ellos y menos se parecerá a la curva del perfil ideal que queremos representar. Esto hace que para tener una precisión alta debamos tratar con grandes cantidades de puntos, lo que puede llegar a hacer que el coste computacional del algoritmo sea demasiado alto.

En este algoritmo no usamos una nube de puntos como referencia, sino la función explícita que representa nuestro perfil del carril. Para ello hemos usado una función definida a trozos que modela la mitad del carril que vamos a comparar con las medidas. Dicha función tiene la forma $x = f(y)$ porque si la modeláramos como una $y = f(x)$ no podríamos hacerlo de esta manera, ya que habría varios valores de x para distintas y . Es por ello que decidimos proyectarla sobre el eje y y nos evitamos ese problema. Se podría haber girado el carril y obtendríamos lo mismo.

A continuación, vamos a expresar matemáticamente la forma que tendrá nuestra ecuación explícita del perfil UIC54. Primero debemos definir la lista de intervalos que le permitirá a nuestro algoritmo saber en qué tramo se encuentra. Para referirnos a los puntos de inicio y fin de un intervalo usaremos el primer o último punto de los definidos para cada tramo de la curva, ya sea arco o recta, como puede verse en la Sección 2.3. El subíndice -1 representa el último punto de la curva en cuestión y el 0 el primero, recorriendo nuestro perfil

desde la cabeza a la base del raíl. Sin embargo, a la hora de definir los intervalos hemos empezado por la base y continuamos hacia arriba a lo largo de nuestra curva, lo que puede ser confuso, de ahí la aclaración.

Cabe destacar que el primer punto de una de las curvas se corresponde con el último del anterior, por lo que los intervalos que vamos a definir ahora podrían haberse definido de manera diferente pero teniendo el mismo significado y valor. Para que sirva de apoyo visual para comprender la división realizada del perfil, volvemos a adjuntar la figura de los tramos del perfil UIC54 (Figura 3.1)

Perfil del carril UIC54 subdividido en tramos

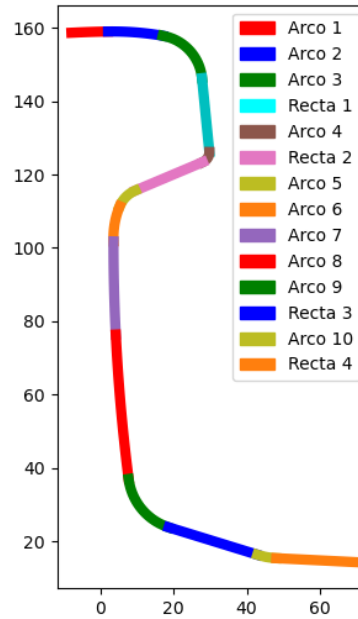


Figura 3.1 División por tramos del perfil UIC54.

$$\left. \begin{aligned}
 \text{Intervalo1} &= [\text{Recta4}_{-1}, \text{Recta4}_0) \\
 \text{Intervalo2} &= [\text{Recta4}_0, \text{Arco10}_0) \\
 \text{Intervalo3} &= [\text{Arco10}_0, \text{Recta3}_0) \\
 \text{Intervalo4} &= [\text{Recta3}_0, \text{Arco9}_0) \\
 \text{Intervalo5} &= [\text{Arco9}_0, \text{Arco8}_0) \\
 \text{Intervalo6} &= [\text{Arco8}_0, \text{Arco7}_0) \\
 \text{Intervalo7} &= [\text{Arco7}_0, \text{Arco6}_0) \\
 \text{Intervalo8} &= [\text{Arco6}_0, \text{Arco5}_0) \\
 \text{Intervalo9} &= [\text{Arco5}_0, \text{Recta2}_0) \\
 \text{Intervalo10} &= [\text{Recta2}_0, \text{Arco4}_0) \\
 \text{Intervalo11} &= [\text{Arco4}_0, \text{Recta1}_0) \\
 \text{Intervalo12} &= [\text{Recta1}_0, \text{Arco3}_0) \\
 \text{Intervalo13} &= [\text{Arco3}_0, \text{Arco2}_0) \\
 \text{Intervalo14} &= [\text{Arco2}_0, \text{Arco1}_0]
 \end{aligned} \right\} \quad (3.3)$$

Una vez que tenemos los intervalos definidos, tenemos que definir las ecuaciones que modulan el perfil en cada uno de ellos. Debido a que esta función es del tipo $x = f(y)$ no podemos usar las funciones definidas anteriormente en las secciones 2.3.1 y 2.3.2. Es por ello que definimos la siguiente función definida a trozos:

$$x = \begin{cases} -10y + (Recta4_{0x} - (-10Recta4_{0y})) & \text{si } y \in \text{Intervalo1} \\ -\sqrt{R_{10}^2 - (y - C_{10y})^2} + C_{10x} & \text{si } y \in \text{Intervalo2} \\ -2.75y + (Recta3_{0x} - (-2.75Recta3_{0y})) & \text{si } y \in \text{Intervalo3} \\ -\sqrt{R_9^2 - (y - C_{9y})^2} + C_{9x} & \text{si } y \in \text{Intervalo4} \\ -\sqrt{R_8^2 - (y - C_{8y})^2} + C_{8x} & \text{si } y \in \text{Intervalo5} \\ -\sqrt{R_7^2 - (y - C_{7y})^2} + C_{7x} & \text{si } y \in \text{Intervalo6} \\ -\sqrt{R_6^2 - (y - C_{6y})^2} + C_{6x} & \text{si } y \in \text{Intervalo7} \\ -\sqrt{R_5^2 - (y - C_{5y})^2} + C_{5x} & \text{si } y \in \text{Intervalo8} \\ 2.75y + (Recta2_{0x} - (2.75Recta2_{0y})) & \text{si } y \in \text{Intervalo9} \\ \sqrt{R_4^2 - (y - C_{4y})^2} + C_{4x} & \text{si } y \in \text{Intervalo10} \\ -\frac{1}{20}y + (Recta1_{0x} - (-\frac{1}{20}Recta1_{0y})) & \text{si } y \in \text{Intervalo11} \\ \sqrt{R_3^2 - (y - C_{3y})^2} + C_{3x} & \text{si } y \in \text{Intervalo12} \\ \sqrt{R_2^2 - (y - C_{2y})^2} + C_{2x} & \text{si } y \in \text{Intervalo13} \\ \sqrt{R_1^2 - (y - C_{1y})^2} + C_{1x} & \text{si } y \in \text{Intervalo14} \end{cases} \quad (3.4)$$

Para los otros algoritmos usábamos el sumatorio de los errores cuadráticos medios entre las parejas de puntos de la nube del perfil ideal y y de la nube de las medidas. Para este algoritmo, en cambio, vamos a usar una fórmula parecida, pero, como no se emparejan puntos, tiene una pequeña modificación. Siendo y^* el valor de la coordenada y del punto de la función más cercano al punto de las medidas b_i al que estamos buscando pareja y $f_{UIC54}(y^*)$ el valor de la función analítica en dicho punto, nuestra función de coste tendrá la siguiente expresión:

$$J = \sum_{i=1}^n \sqrt{(b_{ix} - f_{UIC54}(y^*))^2 + (b_{iy} - y^*)^2}$$

El algoritmo funciona de manera similar al 3.1.1: empieza con una transformación lineal que aplica a las medidas, vuelve a calcular la transformación que minimiza el error entre la nube de puntos y nuestra función y aplica la nueva transformación.

Una de las diferencias es que si en los dos primeros algoritmos se necesita un emparejamiento entre puntos de la nube de medidas y puntos del carril ideal para computar el error que se usa en la función de coste, en este algoritmo lo que minimizamos es la distancia entre cada punto y el punto más cercano de la función. Para ello debemos realizar una búsqueda del punto más cercano de la curva a cada punto de la nube de puntos. Para ello usamos una función de minimización implementada en Python, pero debido a que tiene que buscar ese punto para cada uno de los de la nube de medidas y que debe hacer muchas llamadas a la función que modela nuestro perfil explícito (ecuación 3.4) se vuelve un proceso muy lento y costoso computacionalmente.

Hablaremos de esto en el Capítulo 4, donde propondremos mejoras para nuestros algoritmos o nuevas soluciones que podrían mejorar lo ya hecho. Debido a que no se encuentra suficientemente desarrollado y no

tiene sentido compararlo con los otros que dan bastantes mejores resultados, vamos a excluirlo de las pruebas para determinar el algoritmo óptimo para resolver la tarea en la que estamos trabajando.

3.2 Pruebas de los algoritmos

Como parte final del proceso de diseño de los algoritmos explicados a lo largo de este capítulo, procedemos a probarlos y analizar los resultados que obtenemos. Haremos pruebas con bancos de pruebas y con datos reales obtenidos en diversas mediciones realizadas por Virtualmechanichs S.L. Mostraremos los resultados obtenidos, las comparaciones entre los resultados que nos proporciona cada algoritmo y las conclusiones que podemos extraer de ellos.

3.2.1 Resultados con banco de pruebas

El primer paso en este proceso fue probarlo con datos creados artificialmente en vez de por los obtenidos en las mediciones realizadas por parte de Virtualmechanichs S.L. Al utilizar estos datos ficticios, de los cuales conocemos su forma y contenido, podemos ver los comportamientos de los algoritmos ante diversas modificaciones como puede ser la cantidad de datos, el ruido de las mismas o la influencia de las condiciones iniciales en el proceso de optimización. Además, esta fase fue útil durante el desarrollo de los códigos, puesto que nos facilitó la búsqueda de errores en los programas y nos sirvió para depurarlos. Para poder estudiar todos los casos que nos parecieron relevantes diseñamos 7 experimentos diferentes. Estos son:

- **Experimento 1:** Medidas de perfil completo sin ruido
- **Experimento 2:** Medidas de perfil completo con ruido de pequeña amplitud
- **Experimento 3:** Medidas de perfil completo con ruido de gran amplitud
- **Experimento 4:** Medidas de perfil incompleto sin ruido
- **Experimento 5:** Medidas de perfil incompleto con ruido de pequeña amplitud
- **Experimento 6:** Medidas de perfil incompleto con ruido de gran amplitud
- **Experimento 7:** Influencia de la condición inicial

Para simular nuestras medidas hemos usado la función puntos UIC54 explicada en el Capítulo 2 creada para generar nuestro carril ideal. Para poder calcular la transformación que ajusta estas supuestas medidas al carril ideal también le aplicaremos una rotación y una traslación con los siguientes valores $\theta = -\frac{\pi}{8} rad$ y $\mathbf{t} = \begin{bmatrix} 10 \\ 10 \end{bmatrix} mm$.

Cuando decimos que usamos el carril completo es que vamos a generar nuestras supuestas medidas con todos los tramos del perfil UIC54. Cuando hablamos de perfil incompleto es que no usaremos todos los tramos, sino únicamente los que hemos observado que son los que aparecen en las medidas reales. Esto sucede porque, debido al ángulo en el que se encuentra colocada la cámara respecto al carril, la parte inferior de la cabeza del raíl nunca es observado por nuestras cámaras. Para generar el ruido que simule la incertidumbre típica de cualquier sensor, en este caso la cámara, hemos usado una distribución normal implementada en Python.

Además, para que los resultados fueran comparables hemos inicializado dicho error con un seed común del generador de números aleatorios, de manera que los ruidos aplicados sean iguales para ambos algoritmos. Cuando hablamos de pequeña amplitud hemos usado una desviación típica de 0.2 mm y cuando hablamos de gran amplitud es igual a 2 mm. Tomando $N(\mu, \sigma^2)$ como una distribución normal de media μ y varianza σ^2 , la expresión matemática que modela ambos ruidos serían las siguientes:

- Ruido de pequeña amplitud: $N(0,0.2)$
- Ruido de gran amplitud: $N(0,2)$

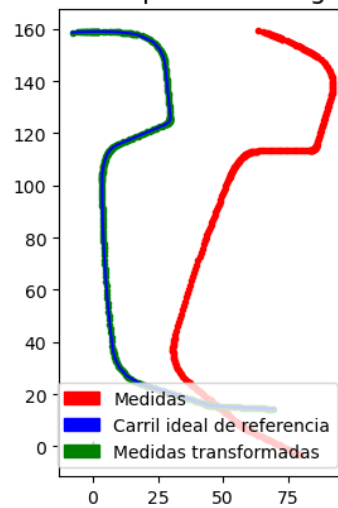
Para cada uno de los casos expuestos arriba, realizaremos cinco pruebas para cada algoritmo y mostraremos como resultado la media de ellos, excepto para el último experimento, para el cuál haremos tres pruebas con diferentes condiciones iniciales y mostraremos los resultados de todas ellas. Además, analizaremos cuál de los algoritmos es el que mejor se comporta e intentaremos explicar cuales pueden ser los motivos para obtener los diferentes resultados. El error que se muestra en las tablas de esta sección está medido según la función de coste de los algoritmos, mostrada en el Capítulo 3 y dividido entre el número de parejas. Es decir, el error obtenido es el la media del error para cada pareja de puntos. Por último, es necesario aclarar que todos los resultados han sido obtenidos con un ordenador con un procesador Intel(R) Core(TM) i7-10750-H 2.60 GHz.

Experimento 1. Medidas de perfil completo sin ruido

Para este primer experimento generamos como supuestas medidas (nube de puntos A según lo visto en la Sección 3.1) el perfil completo y no le aplicamos ningún tipo de ruido. Los resultados obtenidos son los siguientes:

Algoritmo 1: Optimización con dos nubes de puntos

Resultados experimento algoritmo 1

**Figura 3.2** Resultados algoritmo 1 experimento 1.**Algoritmo 2: ICP:**

Resultados experimento algoritmo 2

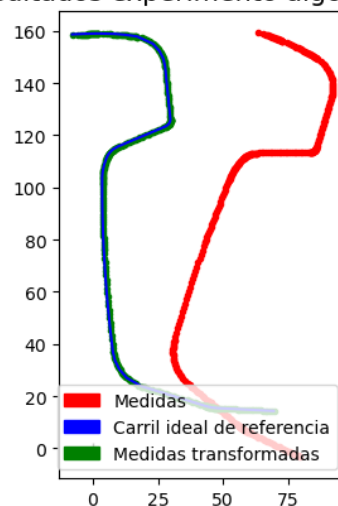
**Figura 3.3** Resultados algoritmo 2 experimento 1.

Tabla 3.1 Resultados del experimento 1. Perfil completo sin ruido..

Algoritmo	Tiempo (s)	Error (mm)	$t_x(mm)$	$t_y(mm)$	θ (rad)
Algoritmo 1 (Opt. nube puntos)	0.6594	1.0038	-5.4119	-13.0656	0.3926
Algoritmo 2 (ICP)	0.0089	1.0239	-5.0423	-13.2956	0.3947

Para determinar si el error es bueno lo vamos a comparar con el ancho de la parte de la cabeza del raíl con la que estamos comparando nuestras medidas. Este es de 35 mm por lo que un error de 1 mm, como el que estamos obteniendo, representa un error del 2.85%. Podemos decir que los resultados son muy buenos, de hecho debido al buen ajuste que conseguimos no se logran diferenciar las curvas verdes y azul, correspondientes a las medidas corregidas sobre el carril ideal de referencia. Observando los resultados observamos que el ángulo que obtenemos es el contrario al que le hemos aplicado a nuestras medidas creadas artificialmente, como podíamos esperar. Sin embargo las traslaciones no se acercan tanto a esos valores esperados. Esto es debido a que como la rotación se hace respecto al origen de coordenadas y no respecto al centroide de la nube de puntos, el giro no se hace sobre sí mismo, sino que crea también un desplazamiento.

Experimento 2. Medidas de perfil completo con ruido de pequeña amplitud

Para este segundo experimento generamos como supuestas medidas el perfil completo con ruido de pequeña amplitud ($\sigma^2=0.2$). Los resultados obtenidos son los siguientes:

Algoritmo 1: Optimización con dos nubes de puntos

Resultados experimento algoritmo 1

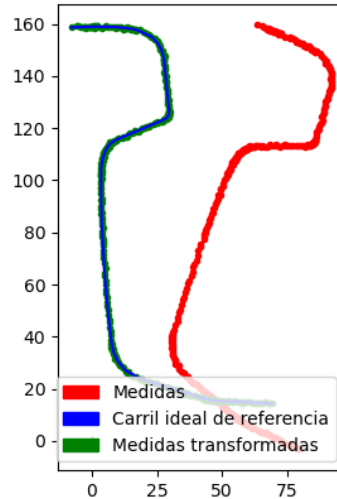


Figura 3.4 Resultados algoritmo 1 experimento 2.

Algoritmo 2: ICP

Resultados experimento algoritmo 2

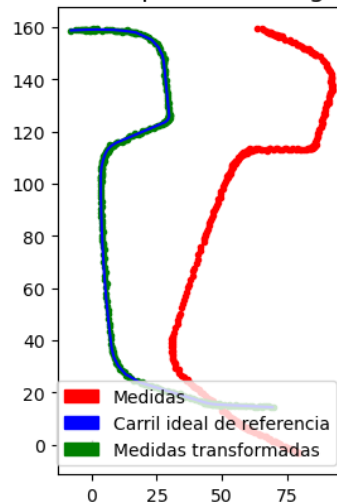


Figura 3.5 Resultados algoritmo 2 experimento 2.

Tabla 3.2 Resultados del experimento 2. Perfil completo con poco ruido..

Algoritmo	Tiempo (s)	Error (mm)	$t_x(mm)$	$t_y(mm)$	θ (rad)
Algoritmo 1 (Opt. nube puntos)	0.4543	1.0254	-5.5469	-13.0198	0.3916
Algoritmo 2 (ICP)	0.0121	1.0305	-5.2200	-13.1247	0.3936

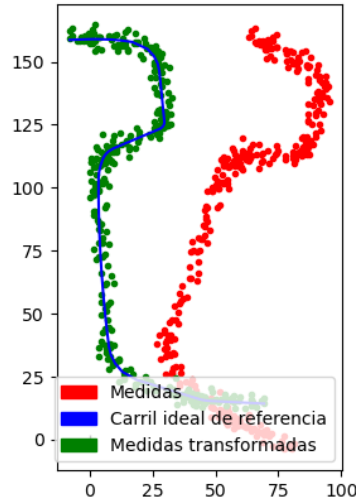
Volvemos a obtener valores del error en torno al 2.8% del ancho de la cabeza del carril.

Experimento 3. Medidas de perfil completo con ruido de gran amplitud

Para este tercer experimento generamos como supuestas medidas el perfil completo con ruido de gran amplitud ($\sigma^2=2$). Los resultados obtenidos son los siguientes:

Algoritmo 1: Optimización con dos nubes de puntos

Resultados experimento algoritmo 1

**Figura 3.6** Resultados algoritmo 1 experimento 3.**Algoritmo 2: ICP**

Resultados experimento algoritmo 2

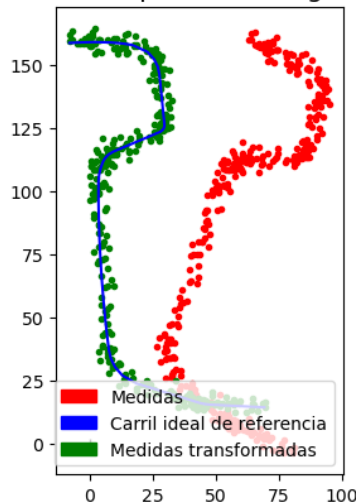
**Figura 3.7** Resultados algoritmo 2 experimento 3.

Tabla 3.3 Resultados del experimento 3. Perfil completo con mucho ruido..

Algoritmo	Tiempo (s)	Error (mm)	$t_x(mm)$	$t_y(mm)$	θ (rad)
Algoritmo 1 (Opt. nube puntos)	0.3894	1.4747	-6.5956	-12.4312	0.3833
Algoritmo 2 (ICP)	0.0149	1.4758	-6.2773	-12.4596	0.3859

Como vemos en los resultados de la Tabla 3.3, el error aumenta respecto a los obtenidos en los experimentos anteriores. Esto es debido a que al añadirle el ruido a las medidas el algoritmo tiene más dificultades para poder encontrar esa transformación. Sin embargo sigue estando en el rango del 4%.

Experimento 4. Medidas de perfil incompleto sin ruido

Para este cuarto experimento generamos como supuestas medidas el perfil incompleto con los tramos que hemos creído convenientes, ya que son los que nos aparecen en las medidas reales. En este experimento no le aplicamos ruido a dichos puntos generados. Los resultados obtenidos son los siguientes:

Algoritmo 1: Optimización con dos nubes de puntos

Resultados experimento algoritmo 1

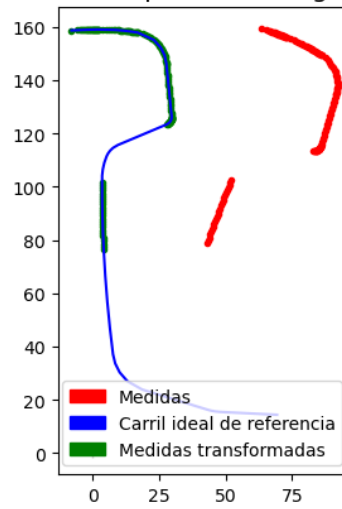


Figura 3.8 Resultados algoritmo 1 experimento 4.

Algoritmo 2: ICP

Resultados experimento algoritmo 2

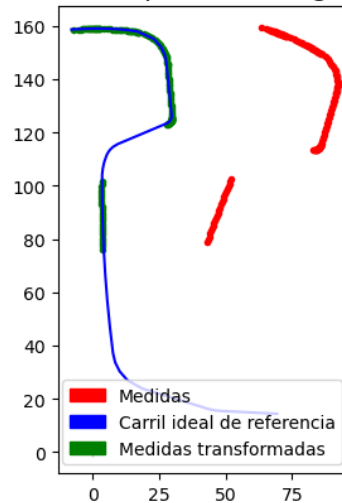


Figura 3.9 Resultados algoritmo 2 experimento 4.

Tabla 3.4 Resultados del experimento 4. Perfil incompleto sin ruido..

Algoritmo	Tiempo (s)	Error (mm)	$t_x(mm)$	$t_y(mm)$	θ (rad)
Algoritmo 1 (Opt. nube puntos)	0.7712	0.9489	-5.4119	-13.0656	0.3926
Algoritmo 2 (ICP)	0.0099	0.9864	-6.6769	-13.1565	0.3822

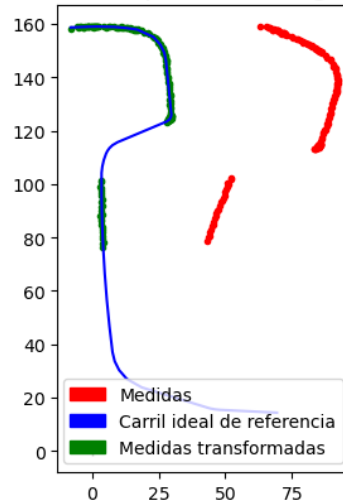
Como vemos, los tiempos aumentan considerablemente. Es lógico ya que es más difícil emparejar dos nubes de puntos con la misma forma que 2 que tienen geometrías diferentes.

Experimento 5. Medidas de perfil incompleto con ruido de pequeña amplitud

Para este quinto experimento hemos creado el perfil con los tramos importantes y le aplicamos un ruido de pequeña amplitud ($\sigma^2=0.2$). Los resultados obtenidos son los siguientes:

Algoritmo 1: Optimización con dos nubes de puntos

Resultados experimento algoritmo 1

**Figura 3.10** Resultados algoritmo 1 experimento 5.**Algoritmo 2: ICP**

Resultados experimento algoritmo 2

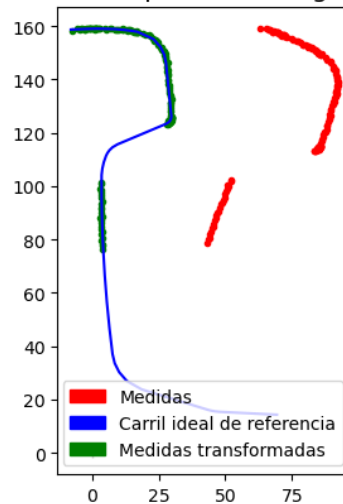
**Figura 3.11** Resultados algoritmo 2 experimento 5.

Tabla 3.5 Resultados del experimento 5. Perfil incompleto con poco ruido..

Algoritmo	Tiempo (s)	Error (mm)	$t_x(mm)$	$t_y(mm)$	θ (rad)
Algoritmo 1 (Opt. nube puntos)	0.6287	0.9841	-5.9290	-13.0535	0.3892
Algoritmo 2 (ICP)	0.0127	0.9881	-6.6415	-12.9592	0.3833

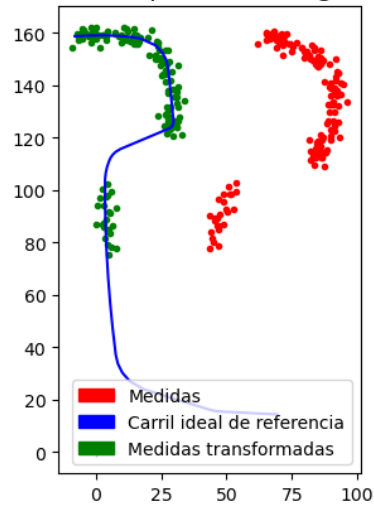
En este experimento podemos apreciar el mismo efecto que tuvo el ruido para el carril completo: aumenta un poco el error pero manteniéndose en valores de orden similar.

Experimento 6. Medidas de perfil incompleto con ruido de gran amplitud

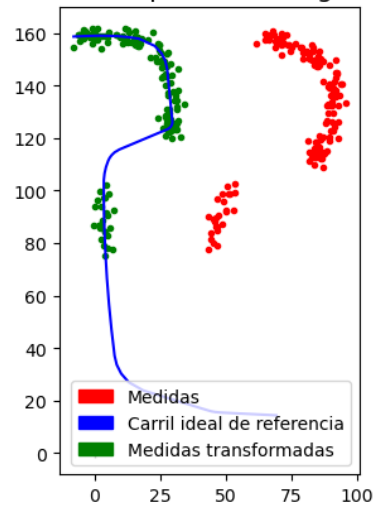
En el sexto experimento volvemos a utilizar únicamente las partes del carril que hacen que nuestras supuestas medidas se parezcan más al caso real y le aplicamos un ruido de gran amplitud ($\sigma^2=2$).

Algoritmo 1: Optimización con dos nubes de puntos

Resultados experimento algoritmo 1

**Figura 3.12** Resultados algoritmo 1 experimento 6.**Algoritmo 2: ICP**

Resultados experimento algoritmo 2

**Figura 3.13** Resultados algoritmo 2 experimento 6.

Vamos a analizar primero estos seis experimentos ya que todos se basan en el cambio de puntos que usamos y el ruido que le metemos a las medidas, por lo que podemos sacar patrones de comportamiento. Observando

Tabla 3.6 Resultados del experimento 6. Perfil incompleto con mucho ruido..

Algoritmo	Tiempo (s)	Error (mm)	$t_x(mm)$	$t_y(mm)$	θ (rad)
Algoritmo 1 (Opt. nube puntos)	0.5707	1.4343	-4.2197	-13.0936	0.4047
Algoritmo 2 (ICP)	0.0127	1.4444	-7.0869	-13.1999	0.3828

los resultados, tanto gráficos como numéricos, podemos ver que ambos algoritmos dan resultados muy parecidos y de error reducido, lo que hace incluso pensar que las gráficas estén repetidas. Cabe aclarar que el error que se computa es la suma de la distancias entre ambas nubes de puntos entre el número de parejas.

Además, podemos ver que los algoritmos son bastante robustos a los parámetros antes comentados (número de puntos, tramos y ruido) porque dan resultados similares para todos los casos. Sin embargo, hay que destacar que el algoritmo 2 es mucho más rápido que el 1, aspecto que tendremos que tener en cuenta a la hora de elegir el algoritmo que usaremos para resolver el problema. Como ya explicamos en el Capítulo 3, esta mayor velocidad se debe a que el algoritmo 2 utiliza álgebra lineal que es computacionalmente menos costosa.

Experimento 7. Influencia de la condición inicial

En este último experimento vamos a estudiar cómo afectan los cambios de la condición inicial. Para no volver a repetir los seis experimentos realizados anteriormente cogemos como base los puntos utilizados para el experimento 6, es decir, el carril con los tramos más importantes y con ruido de gran amplitud. También hay que aclarar que para los experimentos anteriores la condición inicial utilizada ha sido $\mathbf{p}_0 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$. Las dos primeras componentes del vector se corresponden con t_x y t_y , siendo la tercera el ángulo θ . Debemos alzar que estas condiciones iniciales son los valores por los que nuestro algoritmo va a empezar su búsqueda de la transformación óptima para minimizar el error entre ambas nubes, no la posición a la que desplazamos nuestras medidas para simular la irregularidad, como hemos explicado en la sección anterior. Después de probar con un gran número de condiciones iniciales, decidimos que las 3 que usaremos son las siguientes, ya que muestran una serie de comportamientos de nuestros algoritmos que consideramos oportuno recalcar:

$$\left\{ \begin{array}{l} \mathbf{p}_01 = \begin{bmatrix} -9 \\ -9 \\ \frac{\pi}{10} \end{bmatrix} \\ \mathbf{p}_02 = \begin{bmatrix} 100 \\ 100 \\ 0 \end{bmatrix} \\ \mathbf{p}_03 = \begin{bmatrix} 0 \\ 0 \\ -\pi \end{bmatrix} \end{array} \right. \quad (3.5)$$

Las tres condiciones iniciales han sido elegidas para estudiar los tres casos más importantes. La primera es una condición inicial que suponemos que es buena puesto que es muy cercana a la transformación que hemos aplicado a nuestras medidas simuladas, con los signos cambiados puesto que queremos anular esa transformación. La segunda es una condición de partida en la que nos encontramos muy lejos de la solución óptima en términos de la traslación espacial. Por último, la tercera, es el caso en el que la condición inicial nos pone nuestra nube de puntos en un ángulo muy lejano al correcto. Los resultados que obtenemos para cada uno de estos caso son:

Condición inicial 1 (\mathbf{p}_01):

Resultados experimento algoritmo 1

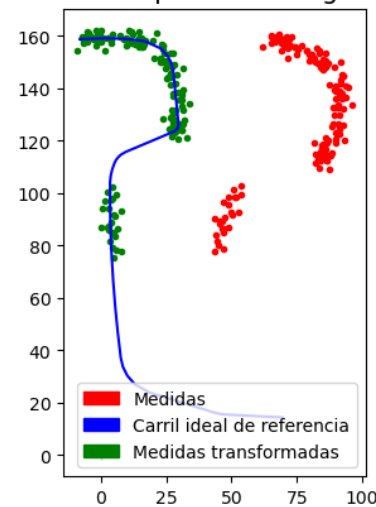


Figura 3.14 Resultados algoritmo 1 experimento 7.1 .

Resultados experimento algoritmo 2

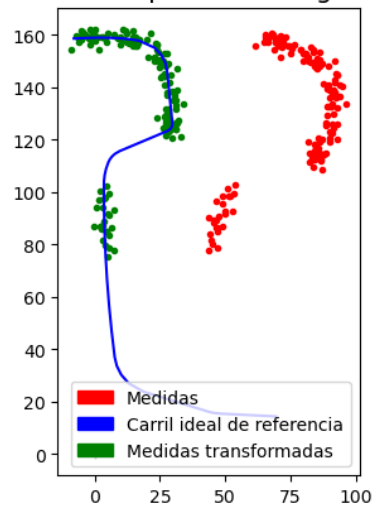


Figura 3.15 Resultados algoritmo 2 experimento 7.1.

Tabla 3.7 Resultados del experimento 7: Condición inicial 1.

Algoritmo	Tiempo (s)	Error (mm)	t_x (mm)	t_y (mm)	θ (rad)
Algoritmo 1 (Opt. nube puntos)	0.5391	1.4343	-4.2197	-13.0936	0.4047
Algoritmo 2 (ICP)	0.0185	1.4363	-4.9747	-12.9429	0.3994

Condición inicial 2 (p_0^2):

Resultados experimento algoritmo 1

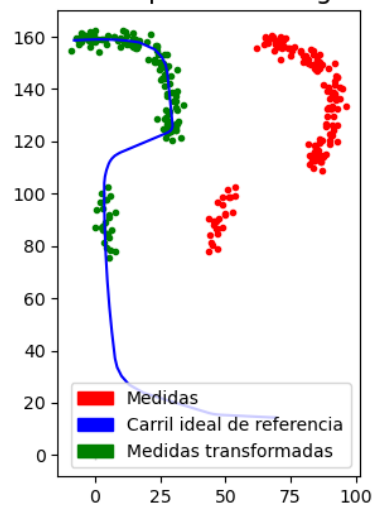


Figura 3.16 Resultados algoritmo 1 experimento 7.2.

Resultados experimento algoritmo 2

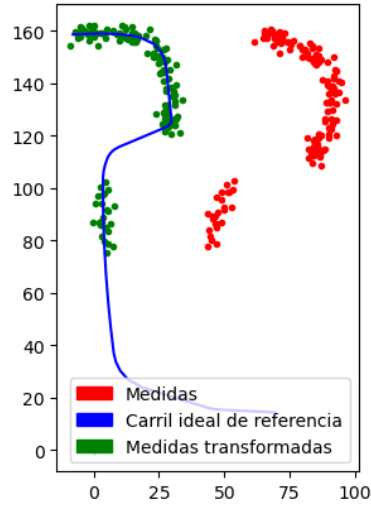


Figura 3.17 Resultados algoritmo 2 experimento 7.2.

Tabla 3.8 Resultados del experimento 7: Condición inicial 2.

Algoritmo	Tiempo (s)	Error (mm)	$t_x(mm)$	$t_y(mm)$	θ (rad)
Algoritmo 1 (Opt. nube puntos)	0.6939	1.4343	-4.2197	-13.0936	0.4047
Algoritmo 2 (ICP)	0.0119	1.4358	-4.8883	-12.9569	0.4002

Condición inicial 3 (p_03):

Resultados experimento algoritmo 1

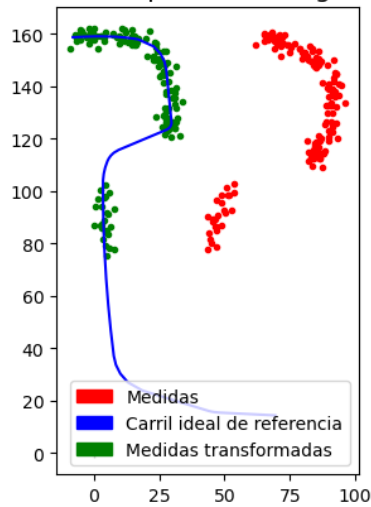


Figura 3.18 Resultados algoritmo 1 experimento 7.3.

Resultados experimento algoritmo 2

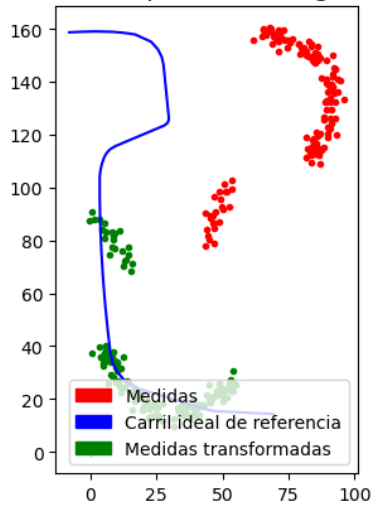


Figura 3.19 Resultados algoritmo 2 experimento 7.3.

Tabla 3.9 Resultados del experimento 7: Condición inicial 3.

Algoritmo	Tiempo (s)	Error (mm)	t_x (mm)	t_y (mm)	θ (rad)
Algoritmo 1 (Opt. nube puntos)	0.5351	1.4343	-4.2197	-13.0936	0.4047
Algoritmo 2 (ICP)	0.0229	2.1945	-37.5151	171.3393	2.1879

Podemos sacar dos conclusiones de estos últimos experimentos:

- El Algoritmo 1 es sensible a una mala condición inicial para la traslación. Aunque vemos que el error es parecido, ha tardado más que para todos los otros experimentos como se puede apreciar en la Tabla 3.9 .
- El Algoritmo 2 tiene mal funcionamiento para una mala condición inicial en términos de ángulo. Podemos ver que se ha quedado atascado en un mínimo local porque el error tiene un valor razonable y similar a los obtenidos para todas las otras condiciones iniciales pero la solución no es la esperada. Se puede apreciar visualmente en la gráfica pero, además, si nos fijamos en los valores obtenidos para la rotación y la traslación óptima que aparecen en la Tabla 3.9, los parámetros calculados distan mucho de los esperados.

3.2.2 Resultados con datos reales

En esta sección vamos a ver como se comportan nuestro algoritmos a la hora de trabajar con datos reales. Dichos datos fueron obtenidos por la empresa Virtualmechanichs S.L. para poder realizar pruebas en su sistema vmRail. La estrategia será similar a la llevada a cabo en las pruebas con datos ficticios: realizaremos diferentes experimentos y para cada uno de ellos realizaremos cinco repeticiones.

El valor mostrado en las tablas que aparecen a lo largo de la sección son la media de esos cinco resultados. Los datos que tenemos están guardados en conjuntos muy grandes de imágenes, del orden de 10^5 para 5 minutos de tramo (a una frecuencia de 250 Hz), pero no utilizaremos todas. Para los dos primeros experimentos, vamos a usar dos grupos de cinco imágenes en diversos tramos del viaje en cuestión para que puede haber una cierta diversidad de resultados. Esto puede ser interesante por que suponemos que en las inmediaciones de un punto kilométrico las medidas serán todas similares, mientras que si cogemos dos tramos lo suficientemente espaciados podremos obtener una mayor disparidad en las medidas, lo que hará que podamos estudiar adecuadamente la robustez de los algoritmos. Habrá tramos con pendiente, curvas o zonas en las que el tren necesite ir a menor velocidad y todas ellas tendrán que ser estudiadas.

Todos los test y sus resultados son originales, realizados en el desarrollo de este Trabajo Fin de Grado. Los algoritmos se programaron en Python y se integraron en el programa general de análisis de viaje desarrollado por Virtualmechanichs S.L..

Además, realizaremos un tercer experimento que consiste en analizar cómo analizan los algoritmos tramos más largos de viajes. Para ello haremos que cada uno de los algoritmos trabaje con un tramo de 10^4 imágenes y mostraremos los resultados. Sin embargo debido a que el algoritmo 1 tarda del orden de 10^2 más que el algoritmo 2, podemos prescindir del primero para este último experimento.

Se discutirá en el capítulo 4 los puntos fuertes y débiles de este algoritmo, al igual que los del resto, y se propondrán posibles mejoras. Se podrá observar en las figuras y en los valores obtenidos, que, obviamente, son peores que con los datos creados por nosotros. Esto es debido a que los sensores introducen errores en las medidas y , por lo tanto, las nubes de puntos no son tan perfectas, lo que dificulta el correcto funcionamiento de los algoritmos.

Por lo tanto, procedemos a mostrar los 3 experimentos llevados a cabo y sus resultados.

Experimento 1. Análisis en punto kilométrico 1

Algoritmo 1: Optimización con dos nubes de puntos

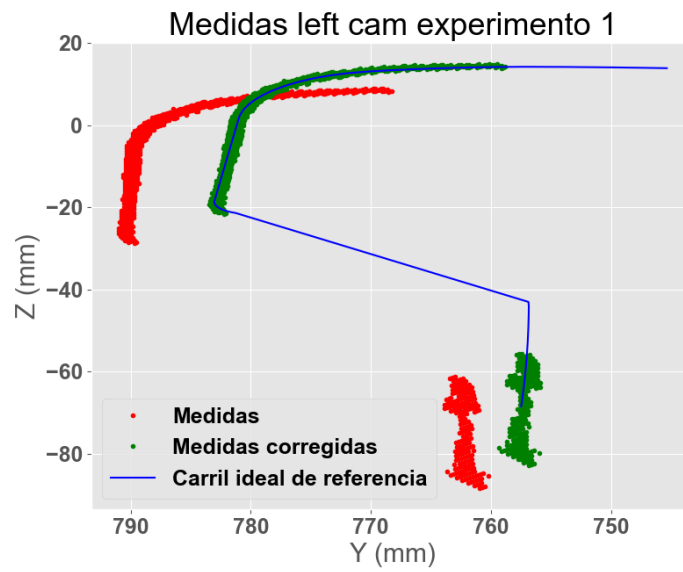


Figura 3.20 Resultados algoritmo 1 experimento 1 para la cámara izquierda.

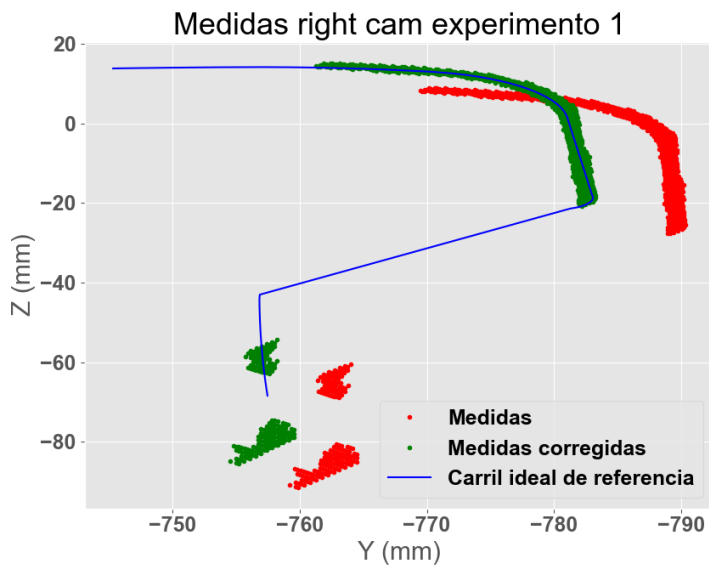


Figura 3.21 Resultados algoritmo 1 experimento 1 para la cámara derecha.

Algoritmo 2: ICP

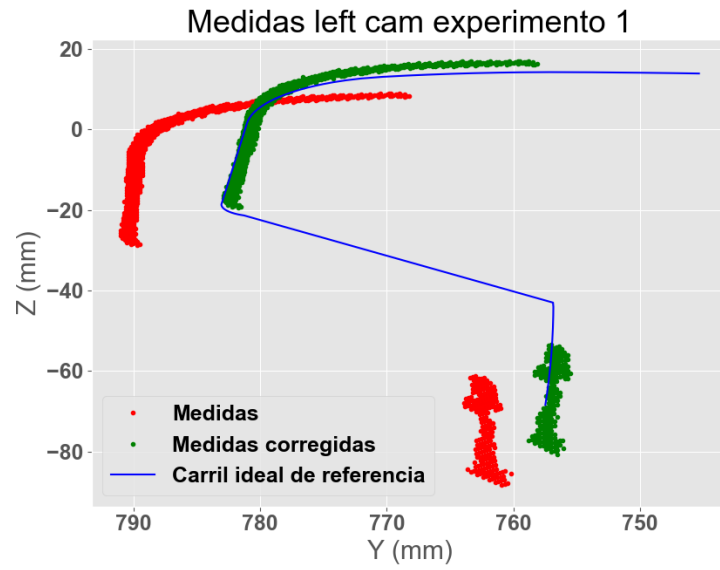


Figura 3.22 Resultados algoritmo 2 experimento 1 para la cámara izquierda.

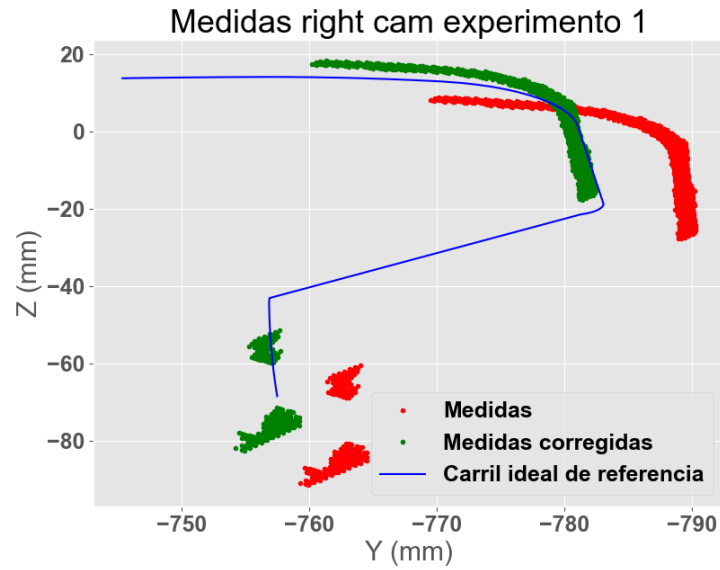


Figura 3.23 Resultados algoritmo 2 experimento 1 para la cámara derecha.

Tabla 3.10 Resultados del experimento 1.

Algoritmo	Tiempo (s)		Error (s)	
	Right cam	Left cam	Right cam	Left cam
Algoritmo 1 (Opt. nube puntos)	1.1733	1.2342	1.2876	1.0647
	Right cam	Left cam	Right cam	Left cam
Algoritmo 2 (ICP)	0.0788	0.0446	1.5419	1.3171

Experimento 2 . Análisis en punto kilométrico 2

Algoritmo 1: Optimización con dos nubes de puntos

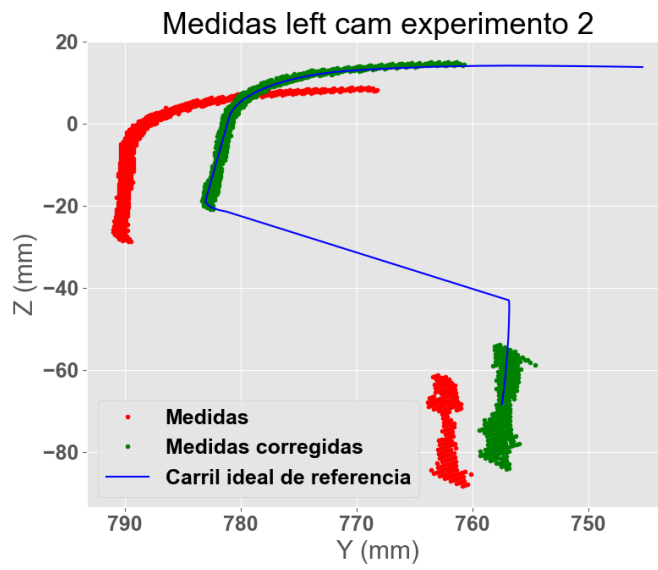


Figura 3.24 Resultados algoritmo 1 experimento 2 para la cámara izquierda.

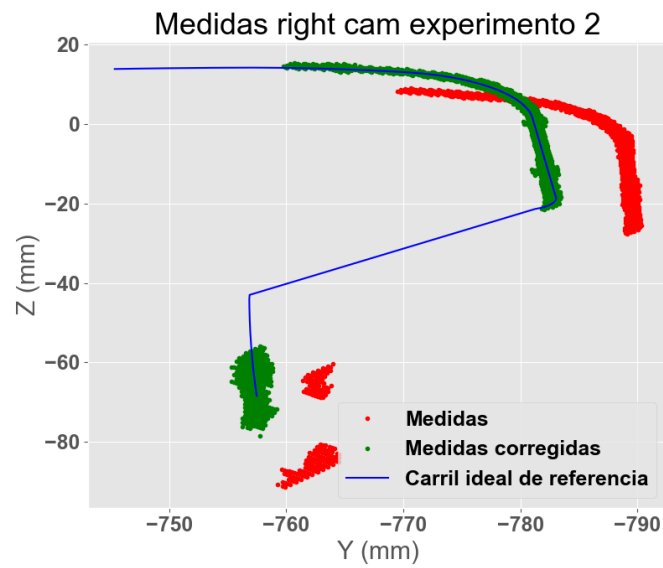


Figura 3.25 Resultados algoritmo 1 experimento 2 para la cámara derecha.

Algoritmo 2: ICP

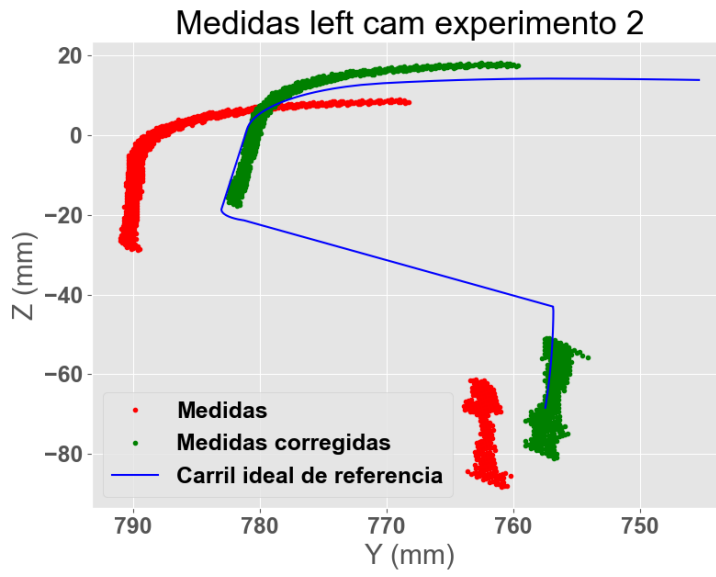


Figura 3.26 Resultados algoritmo 2 experimento 2 para la cámara izquierda.

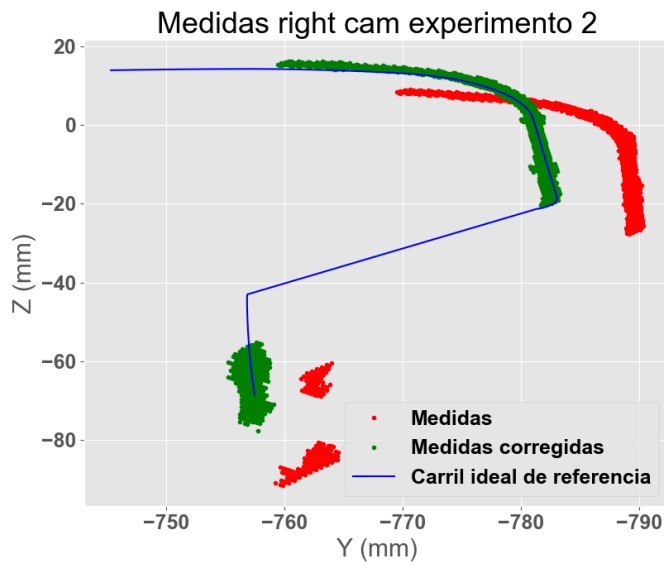


Figura 3.27 Resultados algoritmo 2 experimento 2 para la cámara derecha.

Tabla 3.11 Resultados del experimento 2.

Algoritmo	Tiempo (s)		Error (s)	
	Right cam	Left cam	Right cam	Left cam
Algoritmo 1 (Opt. nube puntos)	0.9811	1.9601	2.1896	1.2325
	Right cam	Left cam	Right cam	Left cam
Algoritmo 2 (ICP)	0.0719	0.0419	0.9835	1.4687

Experimento 3. Análisis de un tramo de viaje

Algoritmo 1: Optimización con dos nubes de puntos

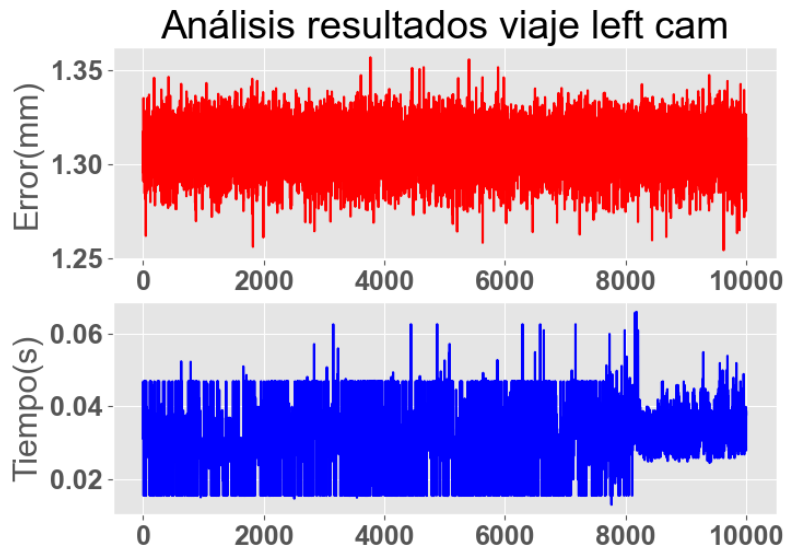


Figura 3.28 Resultados del análisis del viaje para la cámara izquierda.

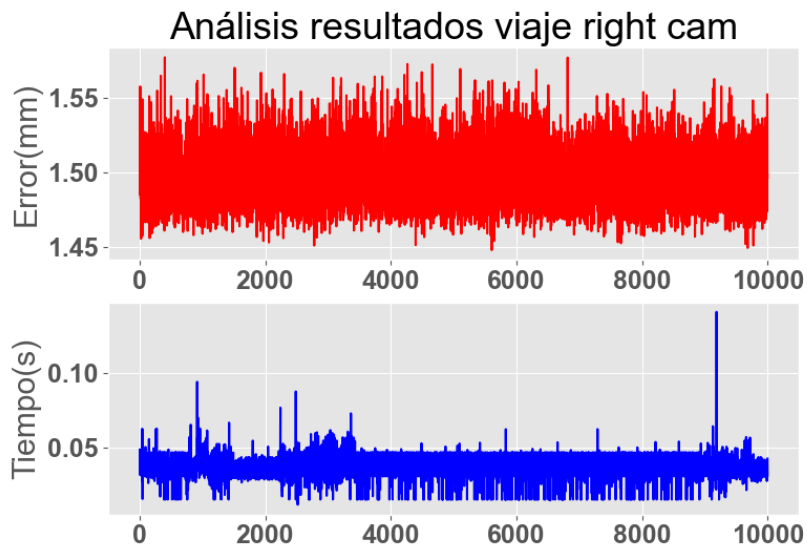


Figura 3.29 Resultados del análisis del viaje para la cámara derecha.

Tabla 3.12 Resultados del experimento 2.

Algoritmo 2 (ICP)	Error medio	Tiempo medio por imagen (s)	Tiempo total (s) [min]
Right cam	1.4959	0.0384	383.8051 [6.4]
Left cam	1.3091	0.0327	327.3757 [5.4]

Vemos que los resultados son bastante reducidos (aproximadamente del 4.3% de la mitad del ancho de la cabeza del carril) y parecidos a los que habíamos obtenido en las pruebas con datos generados artificialmente, por lo que demuestra que el algoritmo 2 es robusto y que da las mismas prestaciones en tramos cortos o tramos largos de la vía.

4 Conclusiones y futuras mejoras

4.1 Conclusiones

Vamos ahora a reflejar las conclusiones que hemos sacado analizando los resultados obtenidos en las secciones 3.2.1 y 3.2.2 . Para ello analizaremos uno a uno los algoritmos y daremos los pros y contras de cada uno.

Algoritmo 1

- Se trata de un algoritmo robusto, que proporciona una buena respuesta al problema de optimización planteado.
- Presenta poco error cuadrático en la distancia entre las nubes de puntos.
- Buena respuesta ante condiciones iniciales muy alejadas de la solución buscada.
- Por contra, es un algoritmo bastante lento y hace que no se pueda plantear su implementación debido a la cantidad de imágenes que tenemos que analizar.

Algoritmo 2

- Es un algoritmo robusto al igual que el 1, da una respuesta coherente para tramos largos del viaje del vehículo.
- Es un algoritmo muy veloz lo que le hace útil para la tarea propuesta.
- Error reducido, con valores parecidos a los del algoritmo 1, aunque en un tiempo mucho menor.
- Su principal debilidad es la sensibilidad a las condiciones iniciales como pudimos ver en la sección 3.2.1, sobre todo a la variable que representa el giro de la nube de puntos.

Valorando los pros y los contras de cada algoritmo, podemos afirmar que el **algoritmo 2** es el que mejor se adapta a las necesidades del proyecto. Su principal debilidad, la sensibilidad frente a las condiciones iniciales, se puede solucionar de dos maneras. La primera es estimar una condición inicial basándose en ensayos anteriores o analizando las gráficas que muestran el ajuste de las nubes. La otra opción es utilizar como condición inicial el resultado obtenido en la imagen anterior. Debido a que podemos suponer que no habrá una gran variación de la posición de las medidas de una imagen a otra, ya que supondría que existe una rotura en el raíl o que hay un fallo en los sensores, esta condición inicial será muy próxima a la solución buscada, por lo que se podrá reducir el tiempo y coste computacional del análisis del viaje.

4.2 Futuras mejoras

El proyecto tiene una serie de posibles mejoras que vamos a exponer a continuación. Estas las propondremos sobre todo para el algoritmo 2 que es el que ha proporcionado los mejores resultados en los ensayos.

- Implementación de las partes más costosas del algoritmo en lenguaje C, ya que es un lenguaje mucho más rápido que Python, debido a que este es un lenguaje interpretado.

- Mejora del proceso de emparejamientos de puntos en los algoritmos del Capítulo 3. La función *sklearn* funciona gracias al Machine Learning y tiene pequeñas posibles variaciones. Una de ellas puede ser la función o estrategia usada para comenzar a buscar vecinos o el parámetro que determine la elección de un punto u otro. Esto podría reducir el coste y el tiempo de ejecución del algoritmo.
- Implementación de un nuevo código que sea mas eficiente para los algoritmos 1 y 3, ya que en la versión actual no pueden competir con el 2. Estas mejoras estarán orientadas a optimizar el código desarrollado y buscar variantes de los algoritmos de búsquedas de parejas. Una posible idea es cambiar la manera en la que se emparejan los puntos de las nubes, como se refleja en el estudio realizado por Rusinkiewicz [22]. Esas pequeñas mejoras podrían hacer ue el comportamiento global del algoritmo mejorara y pudiera acercarse a los resultados obtenidos por el ICP.
- Paralelización de los cálculos usando la GPU. Debido a que el algoritmo debe realizar la misma tarea una serie de veces, podríamos acelerar el proceso de análisis de un viaje si preparamos el algoritmo para que haga los cálculos en paralelo.
- Realización de más test con datos reales para poder comprobar que el algoritmo funciona en situaciones diferentes.

Índice de Figuras

1.1	Ranking de los metros del mundo	3
1.2	Sistema compacto de sensores. IMU en el centro de la imagen y las dos cámaras de visión a los lados.	6
1.3	Esquema de la arquitectura electrónica y sensorial del sistema	7
1.4	Desplazamientos de los carriles debido a las irregularidades de la vía	7
1.5	Global Frame $\langle X, Y, Z \rangle$ y Track Frame $\langle X^t, Y^t, Z^t \rangle$	8
1.6	Global Frame $\langle X, Y, Z \rangle$ y Track Frame $\langle X^t, Y^t, Z^t \rangle$	8
1.7	Irregularidades de las secciones del carril en su línea central	9
1.8	SR de una sección transversal de los carriles en la vía (rail profiles) ideal y real: $\langle x^{lrp}, y^{lrp}, z^{lrp} \rangle, \langle x^{rrp}, y^{rrp}, z^{rrp} \rangle, \langle x^{lid}, y^{lid}, z^{lid} \rangle, \langle x^{rid}, y^{rid}, z^{rid} \rangle$	9
1.9	SR del vehículo instrumentado: track frame, wheelset i frame $\langle X^{wi}, Y^{wi}, Z^{wi} \rangle$ y cámaras frames, $\langle X^{lcam}, Y^{lcam}, Z^{lcam} \rangle, \langle X^{rcam}, Y^{rcam}, Z^{rcam} \rangle$	10
1.10	Sistema de coordenadas de una vía según la norma UNE referente a la caracterización geométrica de la vía	11
1.11	Ancho de vía para carril nuevo	12
1.12	Ancho de vía para carril desgastado	12
1.13	Nivelación longitudinal de la vía	13
1.14	Nivelación transversal de la vía	13
1.15	Alineación	14
1.16	Desplazamientos de los carriles debido a las irregularidades de la vía	14
1.17	Irregularidades laterales	15
1.18	Irregularidades verticales	15
2.1	Modelo esquemático de una cámara estenopeica	17
2.2	Cámara Ximea xiQ USB3.0 MQ003CG-CM. Cámara de tamaño reducido, muy útil para esta aplicación	18
2.3	Vectores de posición en el sistema de referencia de la imagen y de la cámara	18
2.4	Sistemas de referencia en el plano de la imagen	19
2.5	Láser rojo de proyección plana	21
2.6	Posición de la cámara en el mundo	22
2.7	Patrón de calibración utilizado	25
2.8	Patrón de calibración cámara izquierda	25
2.9	Patrón de calibración cámara derecha	26
2.10	Puntos de calibración en la imagen (izq.) y en el mundo (dcha.)	26
2.11	Esquema del teorema de Euler de las rotaciones finitas	33
2.12	Perfil del carril UIC54. Estipulado por las normas UNE	35
2.13	División por tramos del perfil UIC54	36
2.14	Arco 1 del perfil UIC54	37
2.15	Arco 2 del perfil UIC54	38
2.16	Arco 3 del perfil UIC54	38
2.17	Arco 4 del perfil UIC54	39

2.18	Recta 1 del perfil UIC54	40
2.19	Arco 5 del perfil UIC54	41
2.20	Arco 6 del perfil UIC54	42
2.21	Recta 2 del perfil UIC54	43
2.22	Arco 7 del perfil UIC54	44
2.23	Arco 9 del perfil UIC54	45
2.24	Arco 8 del perfil UIC54	46
2.25	Recta 3 del perfil UIC54	47
2.26	Recta 4 del perfil UIC54	48
2.27	Arco 10 del perfil UIC54	48
3.1	División por tramos del perfil UIC54	56
3.2	Resultados algoritmo 1 experimento 1	60
3.3	Resultados algoritmo 2 experimento 1	60
3.4	Resultados algoritmo 1 experimento 2	62
3.5	Resultados algoritmo 2 experimento 2	62
3.6	Resultados algoritmo 1 experimento 3	64
3.7	Resultados algoritmo 2 experimento 3	64
3.8	Resultados algoritmo 1 experimento 4	66
3.9	Resultados algoritmo 2 experimento 4	66
3.10	Resultados algoritmo 1 experimento 5	68
3.11	Resultados algoritmo 2 experimento 5	68
3.12	Resultados algoritmo 1 experimento 6	70
3.13	Resultados algoritmo 2 experimento 6	70
3.14	Resultados algoritmo 1 experimento 7.1	72
3.15	Resultados algoritmo 2 experimento 7.1	73
3.16	Resultados algoritmo 1 experimento 7.2	73
3.17	Resultados algoritmo 2 experimento 7.2	74
3.18	Resultados algoritmo 1 experimento 7.3	74
3.19	Resultados algoritmo 2 experimento 7.3	75
3.20	Resultados algoritmo 1 experimento 1 para la cámara izquierda	77
3.21	Resultados algoritmo 1 experimento 1 para la cámara derecha	77
3.22	Resultados algoritmo 2 experimento 1 para la cámara izquierda	78
3.23	Resultados algoritmo 2 experimento 1 para la cámara derecha	78
3.24	Resultados algoritmo 1 experimento 2 para la cámara izquierda	80
3.25	Resultados algoritmo 1 experimento 2 para la cámara derecha	80
3.26	Resultados algoritmo 2 experimento 2 para la cámara izquierda	81
3.27	Resultados algoritmo 2 experimento 2 para la cámara derecha	81
3.28	Resultados del análisis del viaje para la cámara izquierda	83
3.29	Resultados del análisis del viaje para la cámara derecha	83

Índice de Tablas

2.1	Vectores de posición de los puntos de calibración en los sistemas de la imagen y del mundo	27
3.1	Resultados del experimento 1. Perfil completo sin ruido.	61
3.2	Resultados del experimento 2. Perfil completo con poco ruido.	63
3.3	Resultados del experimento 3. Perfil completo con mucho ruido.	65
3.4	Resultados del experimento 4. Perfil incompleto sin ruido.	67
3.5	Resultados del experimento 5. Perfil incompleto con poco ruido.	69
3.6	Resultados del experimento 6. Perfil incompleto con mucho ruido.	71
3.7	Resultados del experimento 7: Condición inicial 1	73
3.8	Resultados del experimento 7: Condición inicial 2	74
3.9	Resultados del experimento 7: Condición inicial 3	75
3.10	Resultados del experimento 1	79
3.11	Resultados del experimento 2	82
3.12	Resultados del experimento 2	83

Bibliografía

- [1] E. Briales, “Presentación de un método novedoso de medición de la calidad geométrica de la vía desde un sistema embarcado en un vehículo comercial (tesis doctoral industrial),” 2020.
- [2] V. S. Blanco, *Desarrollo de las líneas férreas en Europa*, 1994. [Online]. Available: <https://dialnet.unirioja.es/servlet/articulo?codigo=2778334>
- [3] *Informe corporativo Metro de Madrid 2018*, 2018. [Online]. Available: <https://www.metromadrid.es/sites/default/files/documentos/Portal%20de%20transparencia/Memorias/INFORME%20CORPORATIVO%202018.pdf>
- [4] [Online]. Available: <https://www.railwaygazette.com/infrastructure/talgo-to-supply-330-km/h-dr-avril-inspection-train/54792.article>
- [5] Y. Yang, L. Liu, B. Yi, and F. Chen, “An accurate and fast method to inspect rail wear based on revised global registration,” *IEEE Access*, vol. 6, pp. 57 267–57 278, 2018.
- [6] X. Z, L. Q, M. Q, and Z. Q., “A 3d laser profiling system for rail surface defect detection.” August 2017.
- [7] R. R., F. JM., and P. M., “A comparative analysis of ransac techniques leading to adaptive real-time random sample consensus.” 2008. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-540-88688-4_37
- [8] O. Chum and J. Matas, “Optimal randomized ransac,” *IEEE Transaction on pattern analysis and machine intelligence*, vol. 30, no. 8, Agosto 2008. [Online]. Available: <http://cmp.felk.cvut.cz/~chum/papers/chum-pami08.pdf>
- [9] Y. Yang, L. Liu, B. Yi, and F. Chen, “Dynamic inspection of a rail profile under affine distortion based on the reweighted-scaling iterative closest point method,” *Measurement Science and Technology*, vol. 30, no. 11, p. 115202, sep 2019. [Online]. Available: <https://doi.org/10.1088%2F1361-6501%2F201911115202>
- [10] T. Z. Jochen, J. Schmidt, and H. Niemann, “Point set registration with integrated scale estimation,” 2005.
- [11] W. Burgard, C. Stachniss, M. Bennewitz, D. Tipaldi, and L. Spinello. Introduction to mobile robotics: Iterative closest point algorithm. [Online]. Available: <http://ais.informatik.uni-freiburg.de/teaching/ss14/robotics/slides/18-icp.pdf>
- [12] Q.-Y. Zhou, J. Park, and V. Koltun, “Fast global registration,” in *ECCV*, 2016. [Online]. Available: <http://vladlen.info/papers/fast-global-registration.pdf>
- [13] A. Shabana, *Dynamics of Multibody Systems*, 01 2005.
- [14] CETREN, “UNE-EN 1384 8-1. aplicaciones ferroviarias. vía. calidad de la geometría de la vía. parte 1:caracterización de la geometría de vía.” 2020.
- [15] AENOR, “UNE-EN 1384 8-2. ferroviaria-vía-calidad de la geometría- 2.sistemas de medición. vehículos de registro.” 2007.

-
- [16] —, “UNE-EN 1384 8-3. ferroviaria-vía-calidad de la geometría- 3.sistemas de medición. máquinas de construcción y mantenimiento.” 2010.
- [17] —, “UNE-EN 1384 8-4. aplicaciones ferroviarias-vía-calidad de la geometría de la vía. parte 4: Sistemas de medición. dispositivos manuales y de bajo peso.” 2012.
- [18] —, “UNE-EN 1384 8-5. aplicaciones ferroviarias-vía-calidad de la geometría de la vía. parte 5: Niveles de calidad geométrica. plena vía y aparatos de vía.” 2018.
- [19] CETREN, “UNE-EN 1384 8-6. ferroviaria-vía-calidad de la geometría- 6. caracterización de la calidad de geometría de la vía.” 2014.
- [20] Z. Zhang, “A flexible new technique for camera calibration,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 11, pp. 1330–1334, 2000.
- [21] A. J. Botero, *Descripciones y Transformaciones Espaciales*, 2005.
- [22] S. Rusinkiewicz and M. Levoy, “Efficient variants of the icp algorithm,” in *Proceedings Third International Conference on 3-D Digital Imaging and Modeling*, 2001, pp. 145–152.
- [23] D. Eggert, A. Lorusso, and R. Fisher, “Estimating rigid body transformations: a comparison of four major algorithms,” *Machine Vision and Applications*, March 1997.