ESCUELA DE INGENIERÍAS INDUSTRIALES

Departamento: **INGENIERÍA DE SISTEMAS Y AUTOMÁTICA**

Área de Conocimiento: **INGENIERÍA DE SISTEMAS Y AUTOMÁTICA**

TRABAJO FIN DE GRADO

CONSTRUCCIÓN DE MAPAS DE EXTERIORES CON UN LIDAR 3D EMBARCADO EN UN ROBOT MÓVIL

Grado en Ingeniería Electrónica, Robótica y Mecatrónica

Autor: FRANCISCO ANAYA PALACIOS

Tutor: CIPRIANO GALINDO ANDRADES

Cotutor: JAVIER GONZÁLEZ JIMÉNEZ

MÁLAGA, Junio de 2024

Agradecimientos

A mi familia,

porque mis logros son gracias a su apoyo incondicional.

RESUMEN

El avance significativo en el campo de la robótica móvil de exteriores en los últimos años se materializa en la implementación de vehículos autónomos capaces de detectar los elementos de su entorno, así como de localizarse durante la navegación.

El presente Trabajo de Fin de Grado explora la creación de mapas densos de puntos 3D mediante la utilización de un sensor LiDAR integrado en una plataforma robótica móvil empleando herramientas innovadoras del estado del arte. Una vez construido, se estudia la georreferenciación y orientación de los mapas mediante tecnología GPS, logrando así un sistema de referencia absoluto para la zona mapeada. Además, se estudia la utilización de los mapas generados como método de localización mediante un algoritmo de registro entre nubes de puntos ICP (Iterative Closest Point) y se analizan los resultados obtenidos.

Este trabajo implica una tarea compleja, ya que se afronta el manejo y almacenamiento de grandes volúmenes de datos, lo que requiere de técnicas específicas de filtrado, gestión y visualización para optimizar la relación entre carga computacional y precisión de los mapas. La implementación se lleva a cabo utilizando el software estándar para aplicaciones robóticas ROS2.

PALABRAS CLAVE

Robótica móvil, Mapa 3D, Nube de puntos, Odometría LiDAR, Odometría LiDAR-Inercial (LIO), Algoritmo ICP.

ABSTRACT

The significant advancement in the field of outdoor mobile robotics in recent years is evidenced by the implementation of autonomous vehicles capable of detecting elements in their environment, as well as localizing themselves during navigation.

This Thesis explores the creation of dense 3D point maps using a LiDAR sensor integrated into a mobile robotic platform employing innovative state-of-the-art tools. Once constructed, the georeferencing and orientation of the maps are studied using GPS technology, thereby achieving an absolute reference system for the mapped area. Additionally, the use of the generated maps as a localization method is studied through a point cloud registration algorithm, ICP (Iterative Closest Point), and the obtained results are analyzed.

This work involves a complex task as it addresses the handling and storage of large volumes of data, which requires specific filtering, management, and visualization techniques to optimize the balance between computational load and map accuracy. The implementation is carried out using the standard software for robotic applications, ROS2.

KEYWORDS

Mobile Robotics, 3D Map, Point Cloud, LiDAR Odometry, LiDAR-Inertial Odometry (LIO), ICP algorithm.

UNIVERSIDAD DE MÁLAGA

ESCUELA DE INGENIERÍAS INDUSTRIALES

DECLARACIÓN DE ORIGINALIDAD DEL PROYECTO/TRABAJO FIN DE GRADO

D./ Dña.: Francisco Anaya Palacios
DNI/Pasaporte: 26301927R
Correo: 0619999026@uma.es
Titulación: Ingeniería Electrónica, Robótica y Mecatrónica
Título del Proyecto/Trabajo: Construcción de mapas de exteriores con un LiDAR 3D embarcado en un robot móvil

DECLARA BAJO SU RESPONSABILIDAD

Ser autor/a del texto entregado y que no ha sido presentado con anterioridad, ni total ni parcialmente, para superar materias previamente cursadas en esta u otras titulaciones de la Universidad de Málaga o cualquier otra institución de educación superior u otro tipo de fin.

Así mismo, declara no haber trasgredido ninguna norma universitaria con respecto al plagio ni a las leyes establecidas que protegen la propiedad intelectual, así como que las fuentes utilizadas han sido citadas adecuadamente.

En Málaga, a 24 de junio de 2024

Fdo.: ……

ÍNDICE DE CONTENIDOS

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

TABLA DE SIGLAS Y ACRÓNIMOS

Siglas/Acrónimos	Significado	Traducción
DoF	Degree of Freedom	Grado de libertad
EKF	Extended Kalman Filter	Filtro de Kalman extendido
EPSG	European Petroleum Survey Group	-
GIS/SIG	Geographical Information System	Sistema de información geográfica
GPS	Global Positioning System	Sistema de Posicionamiento Global
ICP	Iterative Closest Point	Punto más cercano iterativo (algoritmo)
IKF	Iterated Kalman Filter	Filtro de Kalman iterado
IMU	Inertial Measurement Unit	Unidad de medición inercial
LiDAR	Light Detection and Ranging	Detección y medición de luz
LIO	LiDAR-Inertial Odometry	Odometría por sensor LiDAR y sensor inercial
MSE/ECM	Mean Squared Error	Error Cuadrático Medio
RGB	Red, Green and Blue	Rojo, verde y azul
ROS	Robot Operating System	Sistema operativo robótico
RTK	Real-Time Kinematic	Cinemática en tiempo real
SLAM	Simultaneous Localization and Mapping	Localización y mapeo simultáneos
TFG	Trabajo Fin de Grado	-
UTM	Universal Transverse Mercator	Sistema de coordenadas universal transversal de Mercator
Vóxel	Volumetric Pixel	Píxel volumétrico
WSG84	World Geodetic System 1984	Sistema Geodésico Mundial 1984
2D	Bidimensional	-
3D	Tridimensional	-

1. INTRODUCCIÓN

En el escenario actual en el que la tecnología se muestra capaz de conectar el mundo real y el mundo digital, se alza como necesario el desarrollo de sistemas de creación de modelos de la realidad y/o mapas que permitan recrear la información tridimensional del entorno. En el campo de la robótica móvil de exteriores, destaca además la necesidad de investigar este terreno dada la alta demanda de sistemas de mapeado precisos y de métodos de localización basados en la información extraíble del entorno.

El sensor LiDAR (Light Detection and Ranging) se ha consolidado como la tecnología fundamental en la percepción del entorno en 3 dimensiones, dadas sus características de rango amplio y gran precisión (Figura 1). El complejo tratamiento de estos datos requiere de herramientas software sofisticadas y de un post-procesamiento que adquiere un gran peso en la generación de los mapas 3D.

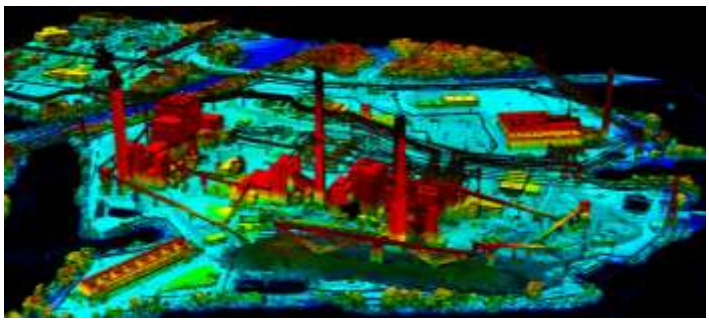


***Figura 1: Mapa tridimensional de alta densidad generado a partir de datos LiDAR**. En la imagen se muestra parte de la infraestructura eléctrica del Estado de Texas. Fuente: https://www.power-grid.com/td/communication-technology/lidar-mapping-of-texas-electrical-infrastructure-underway-for-rural-broadband-project/#gref*

La generación de mapas 3D con gran precisión, requiere de la fusión de la información tridimensional de las nubes de puntos LiDAR con información obtenida por otros sensores. La combinación de datos de un sensor LiDAR y un sensor IMU utilizada en este proyecto, permite una implementación robusta que proporciona información precisa tanto sobre el entorno como sobre la posición del dispositivo móvil.

1.1. MOTIVACIÓN

La utilización de sensores GPS como base de los sistemas de localización es el procedimiento estandarizado en la actualidad debido a su referenciación global y a su alta precisión. Sin embargo, la señal GPS puede encontrarse inaccesible en diferentes zonas y su información puede ser de baja precisión en condiciones adversas, por lo que se requiere de la exploración de métodos basados en otros sensores, como lo es el LiDAR en este proyecto. Esta técnica puede ser clave para la localización en ubicaciones en las que el recorrido mezcla tanto zona exterior como zona interior.

Por tanto, el uso de los mapas construidos con nubes de puntos puede resultar de gran utilidad, dado que la comparación entre estos y las nubes obtenidas por el sensor LiDAR genera

información valiosa para la estimación de la posición relativa del vehículo dentro del sistema de referencia del mapa. Además, la georreferenciación de este mapa permite la conversión de localización relativa a localización absoluta aplicando la transformación apropiada.

1.2. OBJETIVOS

Se define una línea de trabajo compuesta por los siguientes hitos:

- Construcción de un mapa denso de puntos preciso que ofrezca una representación digital fiel de la realidad y que constituya una base sólida para el proyecto. Esto requiere del estudio de los datos y sensores disponibles y de las posibilidades que ofrecen las herramientas del estado del arte, de forma que se puedan obtener resultados óptimos y con un alto nivel de detalle. Dentro de esta tarea se incluye el post-procesado de las nubes de puntos.

- Análisis de los mapas y comparación, previa georreferenciación, con modelos existentes de la zona. Para esto se investigan diferentes herramientas de visualización con el objetivo superponer la nube de puntos que conforma el mapa a capas con información 2D y/o 3D de la zona escaneada, siendo de estudio el formato y la compatibilidad de estos conjuntos de datos. Durante la georreferenciación se aporta un grado mayor de información, de forma que los mapas pasan a contener información en un marco global y no tan solo en el sistema de referencias relativo a la realización del mapeado.

- Estudio de la utilidad de los mapas generados en el diseño de procedimientos y/o implementación de aplicaciones útiles para la localización y estimación de odometría de robots móviles. Este último punto contempla el uso de algoritmos basados en ICP para la obtención de transformaciones que minimicen el error entre la nube de puntos obtenida por el LiDAR y el mapa de puntos construido, de forma que se puedan utilizar como transformaciones correctoras de estimaciones iniciales de pose, buscando resultados de gran precisión y/o mejoras significativas en casos de estimación inicial poco precisa.

1.3. HERRAMIENTAS UTILIZADAS

En cuanto al sistema robótico, se utiliza una plataforma Hunter 2.0 UGV sobre la que se dispone un sistema sensorial compuesto los siguientes dispositivos: un sensor LiDAR Ouster OS1-32 de gran potencia y alta resolución (incorpora además un sensor IMU de 6 ejes sin información de la orientación absoluta) y un receptor GPS-RKT Emlid Reach RS2+ con gran precisión en entornos favorables para la obtención de localización global. Todo esto controlado por un PC a bordo con procesador AMD Ryzen 7 con memoria RAM de 32 GB y disco duro de estado sólido SSD de 500 GB.

A continuación, se enumeran las herramientas software utilizadas para el desarrollo del proyecto. Para el desarrollo software en Python y C++, se utiliza el editor de código Visual Studio Code. Se hace uso de la distribución de ROS2 Humble Hawksbill para la

implementación de los nodos necesarios, junto con las herramientas de ROS2 RVIZ2 y PCL Viewer para visualizar la información recogida por el sistema sensorial y la herramienta RQT Graph para visualizar y verificar el esquema de nodos a tiempo real. Como herramienta de SLAM (mapeado y localización de forma simultánea), se hace uso del paquete de software FASTLIO2[3], el cual se basa en el procesamiento de la información proporcionada por sensores LiDAR y sensores inerciales para la estimación de la odometría del dispositivo móvil, permitiendo además el almacenamiento de las nubes de puntos registradas. Este paquete surge como mejora de la versión anterior FAST-LIO[4] y ofrece una mayor robustez y eficiencia. Para la visualización de nubes de puntos y capas 2D y 3D de la zona se utiliza la plataforma ArcGIS.

1.4. ESTRUCTURA DEL DOCUMENTO

La presente memoria se divide en seis capítulos, de forma que cada uno constituye una parte independiente del global del proyecto.

- En el capítulo de Introducción se realiza una descripción general del contexto en el que se enmarca el proyecto para situar al lector.

- En el segundo capítulo, se realiza una descripción del estado del arte de las tecnologías LiDAR, así como de los métodos de localización en robótica móvil relacionados con el proyecto.

- En el tercer capítulo, se detallan las herramientas hardware y software utilizadas para el desarrollo del proyecto.

- En el capítulo de Implementación, se realiza una descripción del procedimiento seguido para la generación y procesamiento de los mapas 3D y se muestran los resultados de los mapas generados.

- En el capítulo de Pruebas de localización, se describen las diferentes pruebas realizadas en las que se estudia el uso de los mapas con un algoritmo basado en ICP y se muestran los resultados obtenidos.

- En el capítulo de Conclusiones y Líneas Futuras se realiza un análisis general del trabajo realizado y los resultados obtenidos durante el desarrollo del proyecto completo y se comentan las posibles líneas de trabajo que surgen a partir de este trabajo.

2. ESTADO DEL ARTE

El estado del arte de este proyecto abarca un amplio espectro de conocimientos y especialidades tecnológicas relacionados con la robótica móvil. A continuación, se presenta un análisis detallado de la situación actual, que ha servido como base de investigación y proporciona el contexto del desarrollo del presente proyecto.

2.1. TECNOLOGÍAS LIDAR

La tecnología LiDAR, acrónimo de Light Detection and Ranging, se ha consolidado como método estandarizado para aplicaciones de generación de modelos tridimensionales precisos del entorno. Los sensores basados en esta tecnología se denominan activos dado que emiten pulsos de luz hacia las áreas deseadas y captan la luz reflejada por las superficies existentes, midiendo el tiempo transcurrido entre emisión y recepción[1] de los pulsos para calcular la distancia a los objetos (1) (Figura 2). A continuación, se presenta una visión general del estado del arte de estas tecnologías y de sus aplicaciones específicas en el dominio de la robótica.

$$d = \frac{ToF}{v} \tag{1}$$

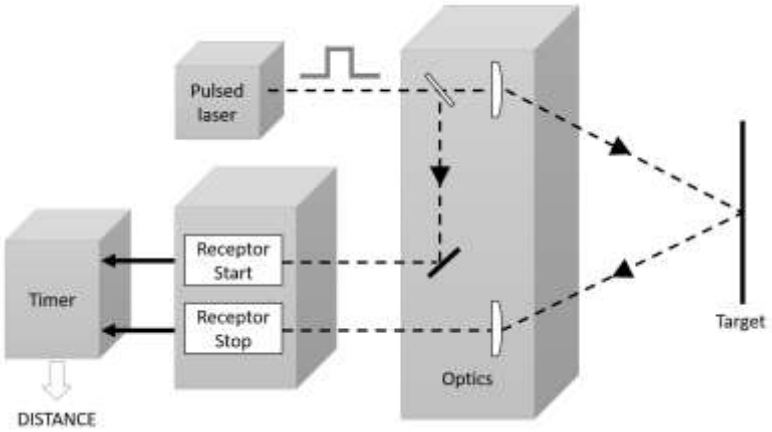


Figura 2: Modo de funcionamiento de un sensor LiDAR. *Los pulsos de luz se envían hacia la zona de interés y se utilizan además para activar el receptor y medir el tiempo hasta que se recibe la luz rebotada. Fuente: https://www.mdpi.com/2076-3417/9/19/4093#*

La implementación del primer sensor LiDAR tuvo lugar en 1961 por parte de la empresa Hughes Aircraft Company para aplicaciones de seguimiento de satélites basándose en la reciente invención del láser. Desde entonces, las aplicaciones del LiDAR han ido adquiriendo presencia en diversos campos entre los que destacan la topografía, la navegación de vehículos

[1] Time of Flight: Tiempo transcurrido entre la emisión y la recepción de un haz de luz utilizado para calcular la distancia a la superficie detectada.

autónomos y la robótica. La evolución de esta tecnología se ha visto reflejada en la creación de diferentes tipos de sensor según el modo de funcionamiento. Entre los más utilizados (ver Figura 3), se encuentran el sensor LiDAR de matriz en fase óptica o sensor LiDAR OPA (Optical Phased Array), el sensor giratorio y el sensor MEMS (Microelectromechanical System) basado en tecnología electromecánica.

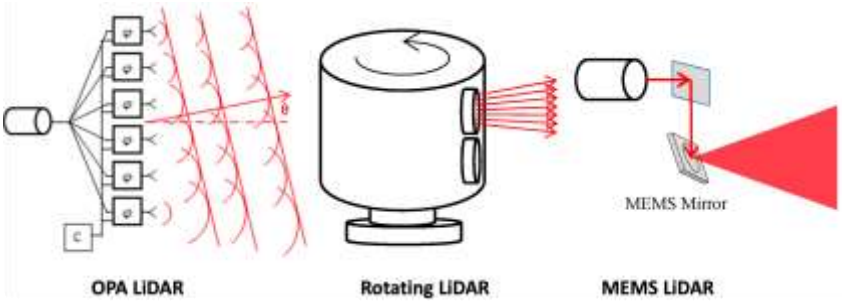


Figura 3: Tipos de sensor LiDAR más utilizados según su modo de funcionamiento según el modo de emisión de luz. *Fuente: https://www.mdpi.com/2072-666X/11/5/456*

De igual forma, existe una clasificación de los sensores LiDAR giratorios de tipo mecánico según el barrido que utilizan para escanear el entorno. Esta se muestra en la Figura 4.

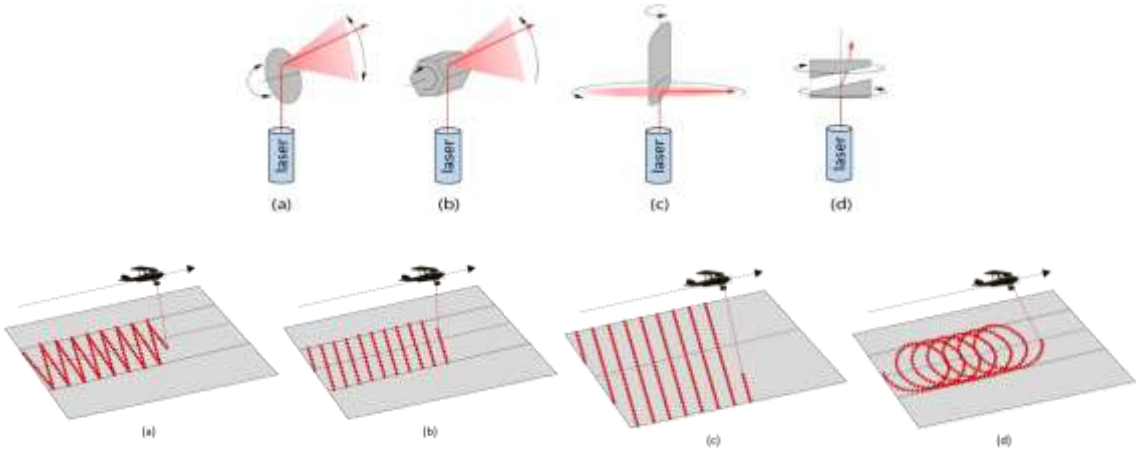


Figura 4: Diferentes tipos de LiDAR según el patrón de escaneo. *a: espejo oscilante, b: espejo poligonal giratorio, c: espejo giratorio, d: prisma dispersivo o de rejilla. Fuente: https://www.yellowscan.com/es/knowledge/what-are-the-different-scan-patterns-of-lidar-systems/*

Además de los sensores LiDAR de escaneo, existe una variante llamada LiDAR Flash basado en la iluminación de una zona completa en cada emisión de luz. Este sensor utiliza una matriz de sensores sin componentes móviles, lo que permite la captura de la información tridimensional del entorno con una mayor velocidad, obteniendo como desventajas resoluciones y rangos de detección menores. Para este proyecto se utiliza un sensor LiDAR giratorio de escaneo en paralelo, cuyo esquema de funcionamiento se muestra en la Figura 5.

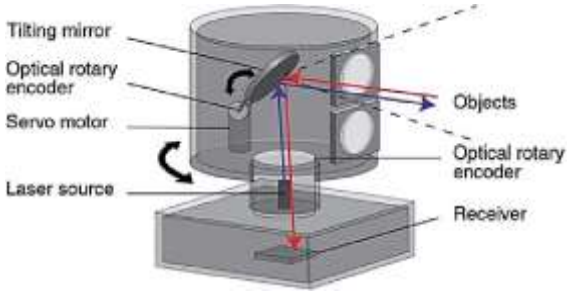


Figura 5: Esquema de partes y funcionamiento de un sensor LiDAR giratorio. *Fuente: https://www.researchgate.net/figure/Principle-of-mechanical-spinning-LiDAR-6_fig5_358910737*

El desarrollo de las tecnologías LiDAR muestra un avance continuo en términos de precisión, miniaturización y velocidad de escaneo y procesamiento de datos. En el momento actual, estas mejoras permiten la integración de sensores LiDAR de tamaño reducido con precisiones del orden del milímetro y con alta velocidad de escaneo en sistemas robóticos y drones compactos. La aplicación de esta tecnología destaca en el ámbito de la robótica dada la alta demanda de aplicaciones de navegación autónoma en entornos de gran escala para tareas de localización y mapeo simultáneos (SLAM). Destaca su utilización en situaciones en las que otros métodos de posicionamiento global, como GPS, no están disponibles, si bien en los últimos años las investigaciones se centran en la fusión de los distintos sensores, dando lugar a un amplio abanico de sistemas multi-sensor basados en LiDAR 3D[8]. Otra de las tareas más extendidas de la tecnología LiDAR en robótica es la generación de mapas de ocupación del entorno tanto bidimensionales como tridimensionales, ver Figura 6.

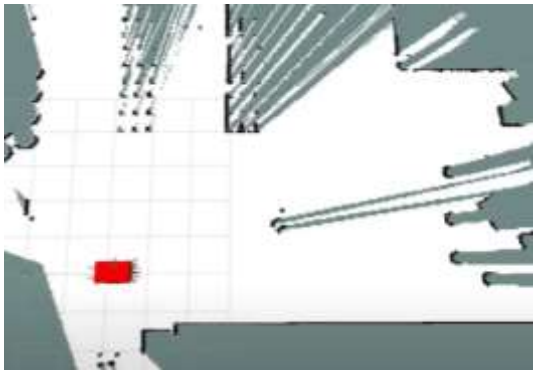


Figura 6: Generación de una cuadrícula de ocupación con un sensor LiDAR 2D. *En el mapa generado se aprecia cómo algunos haces de luz han atravesado los huecos entre los obstáculos. Fuente: https://www.blackcoffeerobotics.com/blog/webots-with-ros-simulation-overview*

Otro de los campos en los que mayor presencia de sensores LiDAR se puede apreciar es el de la conducción autónoma (Figura 7). Los vehículos autónomos se sirven de la información tridimensional precisa que proporcionan estos sensores para una navegación autónoma eficiente y segura. La utilización de estos sensores permite a los vehículos crear mapas en tiempo real, pudiendo de esta forma ajustar la ruta para evitar obstáculos y/o reaccionar ante posibles cambios en el entorno. Dado que son sensores activos, las condiciones de oscuridad no afectan a su desempeño, lo que los convierte en fiables para la conducción nocturna. Sin embargo, otras condiciones climáticas adversas, como lluvia intensa, pueden ocasionar resultados erróneos.

Figura 7: Ejemplo de nube de puntos LiDAR y detección de obstáculos obtenida por un coche autónomo de la empresa Google. *En la imagen se aprecia la detección tanto de otros vehículos como de personas y la identificación del tipo de obstáculo. Fuente: https://www.popsci.com/cars/article/2013-09/google-self-driving-car/*

En el escenario actual en el que la miniaturización y la alta precisión de los sensores LiDAR se encuentran en un estado avanzado, las investigaciones de mejoras se centran en la reducción del alto coste que suelen suponer. Es de destacar además el reciente avance que se ha observado gracias a la implementación de técnicas de visión por computador y deep learning en el procesamiento de los datos LiDAR, lo que permite una mejora sustancial en la extracción de información de alto nivel de gran interés en la navegación[7].

2.2. MAPAS 3D

Los mapas 3D son reconstrucciones digitales utilizadas para describir el entorno. Estos pueden estar conformados por la combinación de nubes de puntos que, además de contener información tridimensional, pueden incluir datos sobre diferentes variables medidas por distintos sensores e incluso información añadida en el post-procesado. Entre estas, podemos encontrar el color RGB, lo que se utiliza para conseguir una representación visual y realista. También es común encontrar información medida por el sensor LiDAR sobre la intensidad registrada en el haz de luz recibida, de la que se puede extraer una idea sobre la naturaleza de la superficie detectada, como pueden ser la reflectividad, la opacidad o la textura. Como datos añadidos en post-procesamiento, es común observar una clasificación de las diferentes partes del mapa según sus características, cosa que se puede conseguir mediante la aplicación de métodos basados en visión por computador, machine learning y deep learning.

El gran volumen que presentan los conjuntos de datos con información 3D de alta precisión de zonas de gran extensión requiere de una gestión eficaz, lo cual es esencial para la creación de mapas 3D de utilidad y cuyo uso no suponga gran complejidad y requiera de hardware y software asequibles. Estos conjuntos adquieren de forma rápida el orden de los millones de puntos, por lo que suele reducirse la información de cada punto a las coordenadas espaciales únicamente, siendo posible la utilización de nubes de puntos con información de otras características como se señaló anteriormente, aumentando esto en gran medida el volumen de datos. Por tanto, se requiere de soluciones avanzadas para el almacenamiento, procesamiento y posterior análisis de estos conjuntos de datos.

Una de las técnicas de actualidad en las que destaca la creación de mapas 3D es la cartografía 3D. Esta utiliza datos de elevación y datos GPS para la creación de representaciones tridimensionales digitales de una zona de interés y se encuentra en auge dado su uso estandarizado en planificación urbana y su gran utilidad en estudios de impacto ambiental. La adquisición de datos en cartografía 3D se puede llevar a cabo mediante diferentes tecnologías, entre las que destacan LiDAR y fotogrametría (Figura 8). Para el procesamiento de estos datos se utilizan herramientas software específicas del contexto de los Sistemas de Información Geográfica (SIG). La georreferenciación de los mapas 3D mediante la aportación de las coordenadas geográficas permite la integración de los datos en un marco de referencia global. Este procedimiento tiene especial interés dentro de la planificación territorial. Además, herramientas como el aprendizaje automático y la inteligencia artificial automatizan el procesamiento de los datos, lo cual sirve como base para aplicaciones de predicción de fenómenos naturales y clasificación de terrenos.

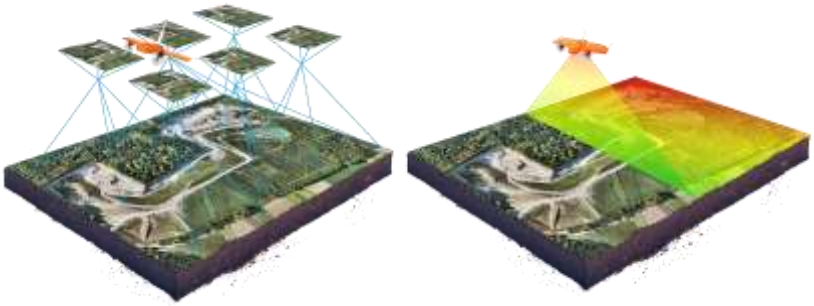


Figura 8: Comparativa del procedimiento de mapeado por fotogrametría (I) y LiDAR (II) en cartografía 3D. *Fuente: https://wingtra.com/es/dron-fotogrametria-vs-lidar/*

La creación de mapas 3D juega un papel fundamental de igual modo en áreas de reciente creación como la realidad virtual (VR), la realidad aumentada (AR) y la unión de ambas, denominada realidad mixta (MR). Los mapas 3D permiten la representación digital de entornos reales de gran interés en las aplicaciones visuales que integran estas tecnologías. En el ámbito de la realidad virtual, estos mapas permiten la visualización de cualquier entorno de forma inmersiva, mientras que en la realidad aumentada se utilizan de forma superpuesta a la realidad, por lo que el procesamiento tridimensional de los datos adquiere una gran relevancia.

2.2.1. Voxelización

La voxelización, como se muestra en la Figura 9, es el proceso de conversión de datos tridimensionales a su equivalente en una matriz tridimensional con un tamaño de cuadrícula específico, normalmente mayor. La unidad mínima que compone la matriz tridimensional se define como vóxel. Por tanto, un vóxel es el equivalente al píxel resultante de la ampliación del espacio 2D a 3D, de ahí su denominación como acrónimo de “volumetric pixel” o “píxel volumétrico”.

Figura 9: Voxelización de diferente tamaño sobre un modelo 3D con forma de conejo. *El tamaño de voxelización decrece de izquierda a derecha incrementando el nivel de detalle y el volumen de los datos. Fuente: https://www.researchgate.net/figure/Point-cloud-voxel-model-in-different-scales-a-Point-clouds-model_fig4_340823107*

El proceso de voxelización pasa por la definición de la cuadrícula espacial según el tamaño de voxelización, el muestreo del espacio para evaluar la información presente en cada volumen elemental de la cuadrícula y la asignación de valores a cada vóxel. Este permite la reducción sustancial del tamaño de los datos, manteniendo información en todas las zonas con un menor nivel de detalle. Por esto, es una técnica de post-procesado fundamental en la gestión de grandes conjuntos de datos, destacando su uso en el campo de la computación grafica.

2.3. LOCALIZACIÓN

La determinación de la posición y orientación de un robot de manera precisa es esencial para la navegación y el mapeo, por lo que la localización ha sido objeto de investigación paralelo al desarrollo del campo de la robótica. El presente proyecto aborda diversos métodos de localización en robótica móvil fundamentados en las tecnologías descritas en este apartado. A continuación, se detallan estos métodos analizando su actualidad y los retos existentes.

2.3.1. Tecnología GPS

El Sistema de Posicionamiento Global o GPS es un sistema de navegación que permite la determinación de la posición absoluta de un receptor en la mayor parte de la superficie terrestre utilizando la información proveniente de la comunicación con diferentes satélites pertenecientes a una constelación (Figura 10). Dado que las mediciones obtenidas a través de receptores GPS son absolutas e independientes, este es el método de localización en tiempo real más extendido para vehículos autónomos y robots de exteriores.

Según el modo de funcionamiento se distinguen diferentes categorías:

- GPS autónomo (Stand-Alone GPS): Es el modo básico de funcionamiento de esta tecnología, en el que los receptores GPS calculan su posición únicamente mediante la recepción de señales de satélites.
- GPS diferencial (DGPS): Este método utiliza estaciones de referencia con ubicación conocida para realizar correcciones diferenciales.
- Cinemática en tiempo real (RKT): Esta categoría utiliza estaciones base para realizar correcciones en tiempo real obteniendo una precisión centimétrica. Es el modo utilizado en este proyecto.

- GPS cinemático post-procesado (PPK): Es un método similar al RKT en el que las correcciones se realizan de forma offline.

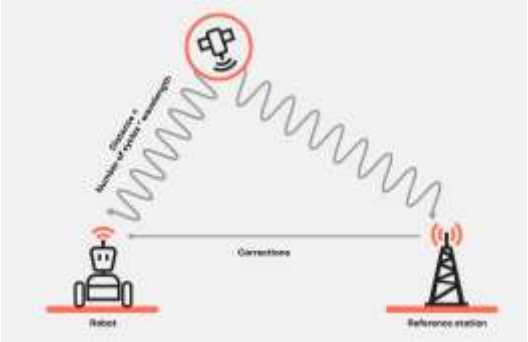


Figura 10: Funcionamiento de un sistema de localización GPS-RKT. *Las medidas de posición obtenidas en relación con un satélite se corrigen utilizando estaciones de referencia con posición exacta conocida. Fuente: https://www.u-blox.com/en/technologies/rtk-real-time-kinematic*

La mayor problemática que enfrenta esta tecnología es la interferencia con otras señales y/o el bloqueo en entornos urbanos densos, dado que las señales GPS son relativamente débiles. De esta forma, la precisión de las medidas puede variar pasando de covarianzas despreciables a errores del orden del kilómetro, lo que puede resultar excesivo para su utilización en numerosas tareas. Por tanto, es necesario investigar otros métodos que sirvan como alternativa a estas situaciones y resulten eficaces en entornos urbanos e incluso en interiores.

2.3.2. Estimación de la odometría y SLAM

La odometría, a diferencia del posicionamiento absoluto por GPS descrito, es una técnica de posicionamiento relativo en robótica móvil que estima la pose del robot por acumulación de los movimientos que realiza, partiendo de una posición inicial conocida. Esta técnica se basa generalmente en las mediciones obtenidas por encoders situados en las ruedas del vehículo. Sin embargo, dada la inexactitud de estos sensores y el constante crecimiento de los errores acumulativos, existen diferentes alternativas de estimación de odometría basados en información sensorial de distintos tipos, como sensores IMU, LiDAR, cámaras, etc. La fusión de los datos de varios de estos sensores ofrece soluciones más robustas a cambio de una mayor carga computacional y permite construir un mapa del entorno y estimar la pose del robot de forma simultánea, procedimiento denominado SLAM (Simultaneous Localization and Mapping). A continuación, se detallan las técnicas con mayor presencia en la actualidad.

La odometría LiDAR[6] es la técnica de estimación de la pose mediante el procesamiento de la información geométrica del entorno obtenida por un sensor LiDAR. Esta realiza comparaciones periódicas de las nubes de puntos anteriores con la nube de puntos actual con el objetivo de obtener la transformación que hace las hace coincidir minimizando el error. Con esta información, es posible calcular el movimiento que ha llevado a cabo el robot y obtener así su localización de forma acumulativa. La utilización de esta técnica para la creación

de mapas del entorno de forma simultánea también se denomina LOAM (LiDAR Odometry and Mapping).

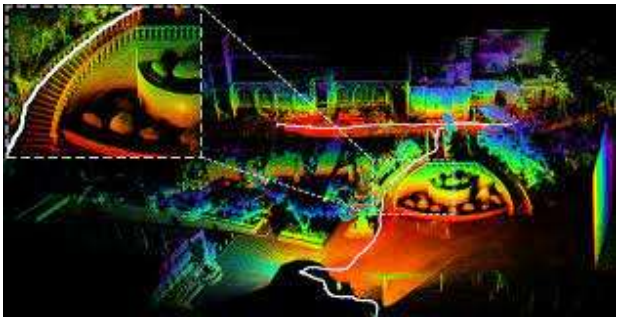


Figura 11: Ejemplo de LiDAR Odometry and Mapping de un sensor Livox. *Se muestra en detalle la trayectoria seguida por el dispositivo durante el mapeo. Fuente: https://jiaronglin.com/project/proj_loam_livox/*

Esta solución está sujeta a diferentes problemáticas, como pueden ser errores en las medidas del sensor LiDAR y errores ocasionados por entornos dinámicos. Con el objetivo de solventarlas, se suelen utilizar otros sensores de forma simultánea. Los datos de estos sensores se fusionan con los datos LiDAR obteniendo una implementación más robusta.

Uno de los métodos de localización más utilizados en la actualidad es el de odometría LIO (LiDAR-Inertial Odometry), el cual se basa en la utilización de sensores LiDAR y sensores inerciales IMU fusionados. Esta combinación de datos se lleva a cabo con el objetivo de obtener una estimación más precisa y robusta a la que se obtiene de la utilización de cada sensor de forma individual, por lo que la fusión juega un papel fundamental en la implementación de esta técnica. Dada la precisión de sus resultados y la posibilidad de funcionamiento tanto en interior como en exterior, se ha convertido en una de las técnicas más utilizadas en robótica para aplicaciones de SLAM.

La odometría visual, por otro lado, se basa en la utilización de cámaras para determinar la posición y orientación del dispositivo mediante el análisis de secuencias de imágenes. Para cada imagen se estima la posición desde la que se ha capturado para coincidir con el mapa 3D (Figura 12).

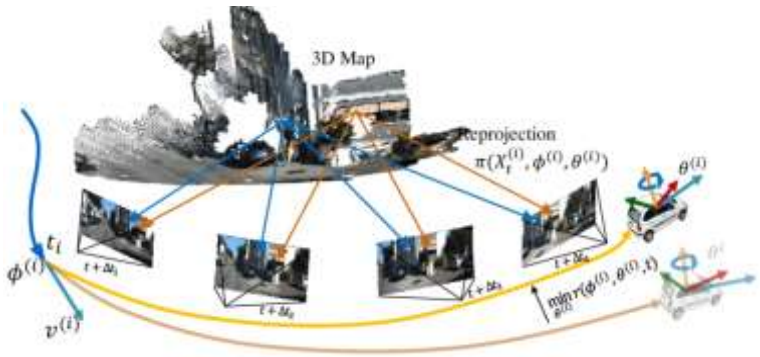


Figura 12: Ejemplo de funcionamiento de odometría visual. *Las diferentes imágenes capturadas se analizan con respecto al mapa 3D para obtener la pose del robot en cada instante. Fuente: https://www.researchgate.net/figure/The-illustration-of-our-visual-odometry-framework-The-initial-state-vti-of-current_fig2_332103736*

La utilización de sensores pasivos, como cámaras sin iluminación, puede dar lugar a resultados erróneos en escenarios oscuros, por lo que en estas situaciones su uso se sustituye por el de sensores LiDAR.

Con el objetivo de mejorar la precisión y la robustez de la estimación de la odometría y de mejorar la localización en entornos dinámicos, se han desarrollado técnicas avanzadas que permiten una fusión optimizada de los datos, entre las que destacan el Filtro de Partículas, el Filtro de Kalman y el algoritmo ICP. El Filtro de Partículas o Filtro de Montecarlo está diseñado para implementaciones en entornos altamente no lineales, dado que utiliza un conjunto de hipótesis que representa la distribución de posibles estados del robot y realiza actualizaciones periódicas en base a las observaciones de los sensores proporcionadas durante la navegación. Esta técnica es ampliamente utilizada en aplicaciones de SLAM en la actualidad y presenta como reto la disminución de la alta demanda computacional necesaria para grandes conjuntos de partículas, lo que resulta necesario para localización precisa en entornos con grandes dimensiones.

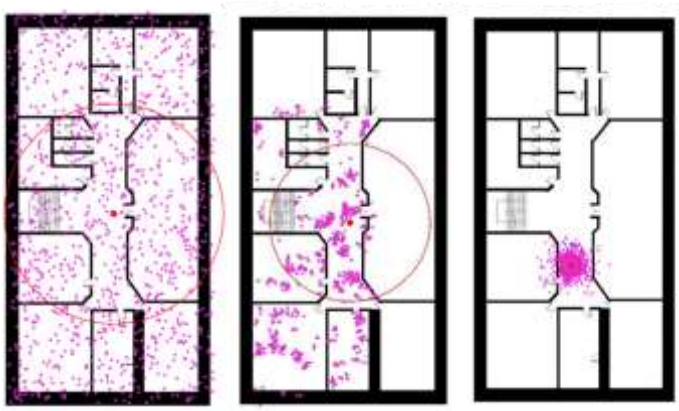


Figura 13: Funcionamiento del Filtro de Partículas en localización. *Se muestran 3 situaciones ordenadas cronológicamente en las que se puede apreciar cómo las partículas o hipótesis del estado del robot convergen a la zona de la posición real, reduciéndose la desviación estándar de la distribución de hipótesis tras cada actualización, disminuyendo así la incertidumbre. Fuente: https://www.researchgate.net/figure/Particle-filter-for-localization-Left-the-initial-state-of-particles-Middle-particles_fig2_328992992*

El Filtro de Kalman realiza predicciones de posición en base a los estados anteriores y a las mediciones, generalmente ruidosas, de los sensores y actualiza el estado del sistema tras cada nueva estimación. El punto clave del funcionamiento de este filtro es la combinación ponderada de predicción y medición con respecto a la incertidumbre de cada una, de forma que se optimiza la generación de predicciones mediante la utilización de la información de los diferentes errores como herramienta. La versión extendida de este filtro o EKF (Extended Kalman Filter) es ampliamente utilizada en navegación autónoma y SLAM dado su enfoque para sistemas no lineales y manejo del ruido de los sensores. De igual forma que para el Filtro de Partículas, el mayor reto que afronta este método es la optimización de la carga computacional, junto con la creación y utilización de modelos precisos del entorno.

El algoritmo ICP (Iterative Closest Point) es un método utilizado para alinear nubes de puntos 3D utilizado en campos como la robótica y la realidad aumentada. Este algoritmo realiza un procedimiento iterativo (ver Figura 14) en el que, partiendo de dos nubes de puntos y una estimación inicial de la situación espacial de ambas, realiza modificaciones (traslaciones y rotaciones) sobre una de ellas y lleva a cabo una serie de cálculos para intentar minimizar el error medido entre ambas nubes de puntos y conseguir un mejor solapamiento.

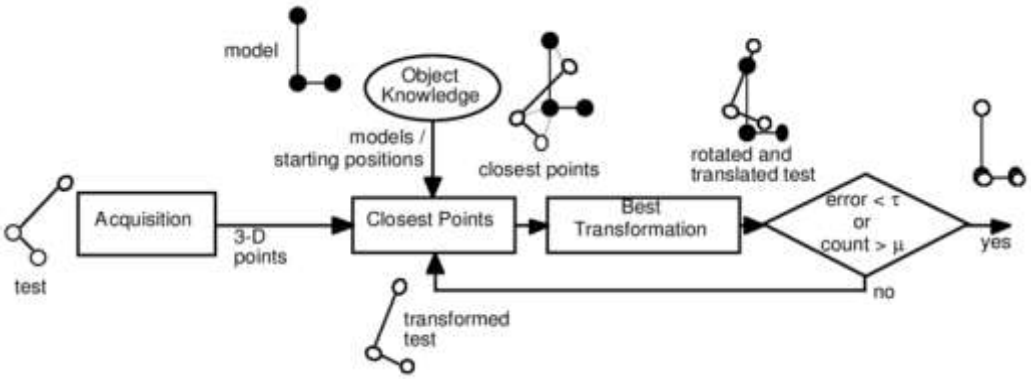


Figura 14: Esquema del modo de funcionamiento del algoritmo ICP. *El conjunto de datos adquirido se transforma de forma iterativa comenzando por una estimación inicial hasta que el error de comparación con el modelo es menor que el umbral definido. Fuente: https://www.researchgate.net/figure/Working-principle-of-the-iterative-closest-point-algorithm-ICP_fig4_228792259*

Para esto, es posible comenzar por un filtrado de los puntos a procesar mediante técnicas como la voxelización de forma que la carga computacional sea menor. Una vez se tienen las nubes a procesar, se define una serie de parámetros como la distancia umbral a partir de la cual no se considera coincidencia entre dos puntos, el número de iteraciones a realizar y el error máximo bajo el cual se considera la convergencia del algoritmo. El proceso iterativo de desplazamiento y rotación de la nube de puntos se realiza a medida que se calculan los errores promedio, analizando para cada iteración si se han cumplido los requisitos de los umbrales predefinidos o si se debe seguir iterando. Una vez el proceso finaliza, ya sea por convergencia o por alcance del máximo de iteraciones permitidas, se obtienen como resultado tanto los valores de error obtenidos tras la aplicación del proceso, como la transformación espacial que minimiza el error de comparación.

Dentro de la robótica, existen numerosas aplicaciones del algoritmo ICP, convirtiéndolo en fundamental dentro de campos como la navegación autónoma, el SLAM o localización y mapeo de forma simultánea (ver Figura 15)[10] y la manipulación de objetos.

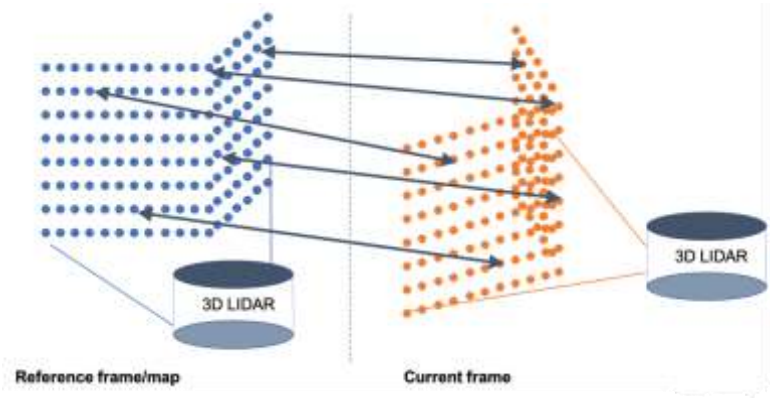


Figura 15: Registro de puntos entre dos nubes para distintas posiciones del sensor LiDAR. *La información de las diferentes nubes se compara con el objetivo de obtener los puntos de correspondencia en el proceso denominado scan matching. Fuente: https://www.kudan.io/blog/3d-lidar-slam-the-basics/*

Estas aplicaciones presentan un aumento reciente dados los avances producidos por las investigaciones en optimización del algoritmo, estableciéndose como herramienta robusta y precisa a la hora de mejorar tanto mapas 3D como estimaciones de la pose de robots móviles. Entre los desafíos actuales en este campo destacan la influencia de la densidad de las nubes de punto comparadas y la dependencia de una estimación inicial correcta, así como la reducción del elevado requerimiento computacional que supone.

En este proyecto se utiliza la versión de ICP punto a punto, por la cual se estudia la correspondencia entre puntos dada su precisión y rápida convergencia. En el estado del arte de este algoritmo coexisten otras versiones, como son ICP punto a plano, en la que la correspondencia se establece entre los puntos de una nube y los planos tangentes a la otra nube y la versión generalizada G-ICP que une las anteriores. Además, existen algoritmos diseñados para el registro de nubes de puntos similares [11], entre los que destacan el algoritmo Normal Distributions Transform (NDT) y el algoritmo Fast Global Registration (FGR).

3. HERRAMIENTAS HARDWARE Y SOFTWARE UTILIZADAS

Para el desarrollo de este proyecto se ha utilizado el conjunto de sensores y dispositivos electrónicos explicados a continuación, interconectados y dispuestos sobre una plataforma robótica móvil (ver Figura 16).

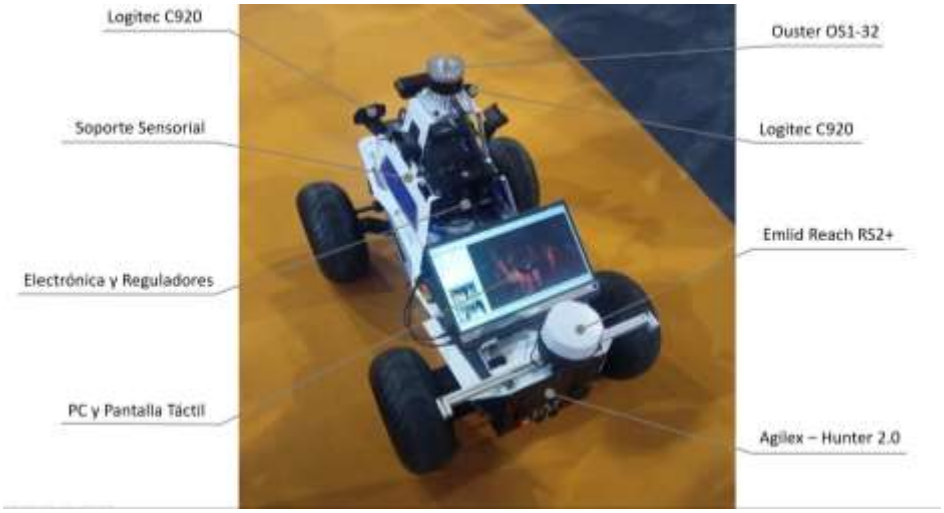


Figura 16: Sistema robótico móvil compuesto por plataforma robótica y sensórica/electrónica.

3.1. PLATAFORMA ROBÓTICA

La plataforma robótica utilizada es el modelo Hunter 2.0 UGV (Unmanned Ground Vehicle – Vehículo terrestre no tripulado) (Figura 17) de la empresa Agilex. El cuerpo de la plataforma está construido en acero y utiliza dos servomotores para conseguir una velocidad variable. El guiado se realiza mediante control remoto. Esta plataforma utiliza el sistema de Ackerman de dirección para el giro de las ruedas delanteras y cuenta con tracción trasera y suspensión independiente de las ruedas delanteras. Las características principales de esta plataforma se recogen en la Tabla 1.

Característica	Valor
Peso	65 kg
Dimensiones	980 x 745 x 380 mm
Distancia entre ejes	650 mm
Potencia	1200 W
Batería	24 V / 30 Ah
Velocidad máxima	10 km/h
Autonomía máxima	22 km

Tabla 1: Características de la plataforma robótica Hunter 2.0 UGV.

Figura 17: Plataforma robótica Hunter 2.0 UGV de Agilex. *Fuente: https://global.agilex.ai/*

Para la administración e intercomunicación de los dispositivos, la plataforma se ha equipado con un computador con procesador AMD Ryzen 7 4800U, memoria RAM de 32 GB, disco duro de estado sólido (SSD) de 500 GB y 6 conectores USB.

3.2. SENSORES

A continuación, se ofrece una descripción detallada de los distintos dispositivos de detección y medición que se han integrado en el sistema robótico para el desarrollo del proyecto.

3.2.1. SENSOR LiDAR + IMU

Para la obtención de datos LiDAR se utiliza el sensor Ouster OS1-32 (Figura 18), el cual incorpora además una Unidad de Medición Inercial (IMU).

Figura 18: Sensor LiDAR Ouster OS1-32. *Fuente: https://ouster.com/*

El Ouster OS1-32 es un sensor LiDAR de alta resolución y rango medio que destaca por su gran potencia y rendimiento para su bajo peso. Está diseñado para uso en automatización industrial, vehículos autónomos, cartografía y robótica. Este dispositivo cuenta con

controladores de ROS y C++ de código abierto. Las características de este sensor se muestran en la Tabla 2.

Característica	Valor
Alcance máximo	200 m
Campo de visión horizontal	360º
campo de visión vertical	45º (+/- 22.5º)
Resolución angular	0.01º
Resolución horizontal	1024
Resolución vertical	32
Rango mínimo	0.5 m

Tabla 2: Características del LiDAR Ouster OS1-32.

El sensor Ouster OS1-32-U cuenta además con un sensor IMU modelo IAM-20680HT de InvenSense. Esta IMU de 6 ejes (giroscopio de 3 ejes y acelerómetro de 3 ejes) con tecnología MotionTracking cuenta con una frecuencia de 100 muestras por segundo con latencia máxima de 10 ms.

3.2.2. GPS

El receptor GPS utilizado es el Emlid Reach RS2+ (Figura 19). Este receptor GPS-RKT con precisión de hasta el orden del centímetro está diseñado para robótica móvil dada su tasa de actualización de hasta 10 Hz. La precisión del sensor en los distintos modos de funcionamiento se muestra en la siguiente tabla:

Modo de funcionamiento	Precisión vertical	Precisión horizontal
Estático	8 mm + 1 ppm	4 mm + 0.5 ppm
PPK	10 mm + 1 ppm	5 mm + 0.5 ppm
RTK	14 mm + 1 ppm	7 mm + 1 ppm

Tabla 3: Precisión del receptor GPS Emlid Reach RS2+ para los distintos modos de funcionamiento.

Figura 19: Sensor GPS Emlid Reach RS2+. *Fuente: https://emlid.com/reachrs2plus/*

3.3. ROS

ROS (Robot Operating System)[5] es un middleware o marco de trabajo de software de código abierto estandarizado para el desarrollo y creación de aplicaciones en robótica. ROS se lanzó en 2007 por los estudiantes de doctorado Keenan Wyrobek y Eric Berger con objetivo de crear un marco de trabajo flexible para la programación de dispositivos asociados con la robótica y para su interconexión y ejecución simultánea. Dentro de este middleware son compatibles los lenguajes de programación C++ y Python, permitiendo su selección en base a las diferentes características de cada uno, siendo posible la interacción entre nodos implementados con cada lenguaje.

La evolución de este middleware dio lugar a su nueva versión descentralizada ROS2. En esta se prescinde del nodo Master, de forma que ahora son los propios nodos los que comunican al resto su aparición y desaparición. Dentro de ROS2 se crean distribuciones o conjuntos de versiones de paquetes de forma que los desarrolladores puedan trabajar con versiones estables mientras se trabaja en la actualización y mejora de las distribuciones futuras. Para este proyecto se ha utilizado la distribución de ROS2 Humble Hawksbill, lanzada en mayo de 2022. Para cada distribución se define además una fecha de fin de vida útil o end of life (EOL), siendo la ROS2 Humble en mayo de 2027. La elección de esta distribución se basa en la búsqueda de una implementación con software actualizado, manteniendo la seguridad de la estabilidad necesaria dado el tiempo transcurrido desde su lanzamiento y de un tiempo de vida útil considerablemente amplio.

Para la visualización de la información recogida por los sensores y de la información 3D y de odometría generada por FAST-LIO2, se utiliza la herramienta RVIZ2 dada su compatibilidad con todos los tipos de datos utilizados durante en este trabajo (nubes de puntos, trayectorias, sistemas de referencias, etc).

Durante el desarrollo del proyecto, se ha hecho uso de la herramienta Rosbag2 de ROS2, la cual permite la grabación, reproducción y manipulación de mensajes. Esta se ha utilizado para capturar los mensajes enviados por los sensores durante los mapeos y para la reproducción de estos y otros conjuntos de datos en condiciones idénticas sobre los que realizar diferentes pruebas y obtener una evaluación precisa y confiable.

3.4. FAST-LIO2

Fast-Lio2 (Fast LiDAR-Inertial Odometry 2)[3] es un paquete de ROS de odometría LiDAR-inercial versátil, rápido y robusto. El paquete de ROS2 integra un sensor LiDAR y un sensor inercial (IMU) utilizando un filtro de Kalman iterado (IKF)[1], de modo que se consigue estimar la odometría del dispositivo móvil de forma eficiente y fiable. Dentro de las características de esta versión del paquete destacan:

- Mapeo incremental utilizando ikd-Tree[9].
- Admisión de una mayor variedad de sensores LiDAR, incluyendo los sensores LiDAR giratorios como Ouster.
- Posibilidad de funcionamiento con IMU externa.

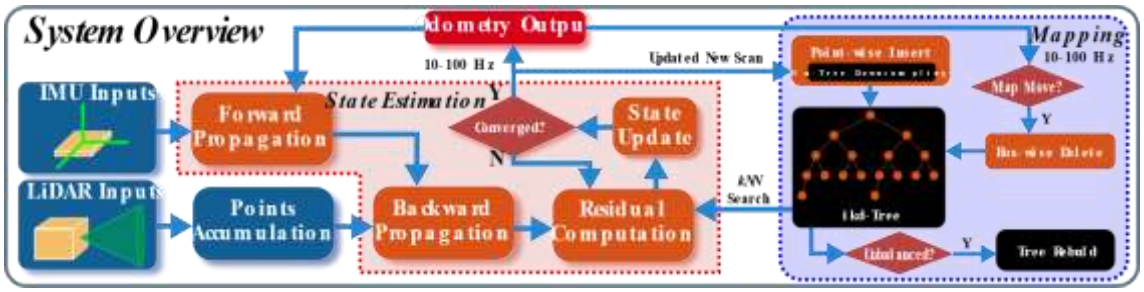


Figura 20 : Diagrama de funcionamiento de FAST_LIO2. *Fuente: https://github.com/hku-mars/FAST_LIO?tab=readme-ov-file*

Por tanto, este algoritmo se presenta como una mejora significativa en comparación con su versión predecesora, FAST-LIO. Los resultados que ofrece se generan con una mayor eficiencia en el tratamiento de datos y muestran un aumento en la precisión en cuanto a la estimación de la posición y orientación del robot. Además, la aplicación de odometría directa[2] permite una mejora notable en el tiempo de procesamiento.

Esta versión de FAST_LIO 2.0 para ROS2 tiene como requisito una versión de Ubuntu igual o superior a la 20.04 y es compatible con distribuciones de ROS a partir de Foxy, siendo recomendable la utilización de Humble, la cual se utiliza para el desarrollo de este proyecto. Otros requerimientos son las bibliotecas PCL y Eigen y el driver de livox para ROS2.

Una vez se cumplen los requerimientos del paquete, este procesa la información recibida por los sensores mediante los topics definidos y los procesa, utilizando el tipo de datos para LiDAR con los campos XYZI, incluyendo información espacial de los obstáculos y de intensidad de la luz rebotada. Estos datos LiDAR se combinan utilizando la información de la odometría y es posible la modificación del código inicial para su guardado con el fin de su posterior procesamiento. Para ello se utiliza la adaptación de la estructura de datos de puntos en un espacio de árbol k-d llamada IKD-Tree. Esta implementación optimiza el manejo de los datos y permite un aumento en la velocidad de procesamiento, dado que organiza el conjunto de puntos de forma que tareas repetitivas como búsqueda, eliminación y adición de puntos se realiza de manera sencilla. De esta forma, se genera información de utilidad sobre odometría y nubes de puntos combinadas.

Para la estimación de la odometría del dispositivo móvil se utiliza un filtro de Kalman extendido iterado estrechamente acoplado (a diferencia de otras herramientas similares que utilizan un acoplamiento débil), llevando a cabo un procesamiento de los datos de cada sensor por separado y con antelación al fusionado de ambos. El procedimiento iterativo del filtro tiene por objetivo un ajuste continuo de las estimaciones del estado del dispositivo móvil generadas tras la recepción de cada nuevo conjunto de datos, permitiendo así un refinamiento de las

[2] La odometría directa es el método por el cual se utilizan los datos obtenidos por cámaras o sensores LiDAR sin realizar un pre-procesamiento para extraer información de las características del entorno, como superficies o bordes.

estimaciones iniciales que produce resultados más robustos y fiables, otorgando, por lo tanto, una mayor fiabilidad a la odometría que genera FAST-LIO2.

La utilización de este paquete de ROS2 resulta interesante tanto para aplicaciones de localización, dada la robustez de su estimación de odometría, como para la generación de modelos digitales de entornos reales. Debido a la mejora en la eficiencia y la velocidad de procesamiento, FAST-LIO2 se destaca como una elección interesante tanto para procesamiento offline como online, ya que es capaz de generar los resultados en tiempo real.

La elección de esta herramienta se basa en la utilización de sensores LiDAR e IMU y en la robustez y precisión de sus resultados, puesto que el hecho de combinar ambos sensores en la misma herramienta genera una fiabilidad mayor, dado que la información ajustada y refinada en cada iteración para la estimación de la odometría se utiliza en las iteraciones futuras para las nuevas estimaciones. Durante el desarrollo de este proyecto se hace uso de la información que genera FAST-LIO2 en dos tareas distintas. En primer lugar, se combinan las nubes que la herramienta crea durante el recorrido del robot y, por otro lado, se utiliza la información que genera sobre la pose del robot como ground truth[3] para las pruebas de mejoras en la estimación de localización por su precisión y robustez.

3.5. ARCGIS

ArcGIS es una plataforma integral de software para Sistemas de Información Geográfica (GIS) desarrollada por la compañía Environmental Systems Research Institute (ESRI). Esta plataforma puede ser utilizada para la creación, gestión, análisis y visualización de diferentes tipos de datos geoespaciales, estableciéndose en los últimos años como un estándar dentro de la industria SIG, además de proporcionar soluciones avanzadas para la investigación científica. Durante el desarrollo del proyecto, se ha hecho uso de la plataforma para la visualización de los mapas georreferenciados generados y para la validación mediante la comparación con modelos de 2 y 3 dimensiones de la zona mapeada.

ArcGIS se destaca por la potencia de sus herramientas de visualización, posibilitando al usuario la creación de mapas y representaciones espaciales con gran nivel de detalle. Entre las ventajas de la utilización de esta plataforma, destacan: precisión, flexibilidad y escalabilidad, comunidad y soporte e innovación continua.

ArcGIS proporciona el acceso a una gran cantidad de bases de datos geoespaciales de una gran variedad de tipos y estilos, los cuales son fácilmente descargables y pueden utilizarse tanto para definición como base del proyecto como para superposición a otra base (pudiendo superponer una o varias capas). Además, ArcGIS permite la integración de datos provenientes de fuentes variadas como pueden ser satélites, drones, sensores, bases de datos externas, etc. Esta plataforma cuenta con un gran abanico de herramientas para la visualización de los datos

[3] *Ground Truth* es el término que se utiliza en robótica móvil para definir la medida de posición más precisa disponible. Esta información se usa como referencia para la comparación y/o validación de otros datos.

espaciales, siendo destacable la dualidad de visualización 2D y 3D. Además, se encuentran disponibles herramientas de generación de representaciones dinámica, lo que permite una visualización interesante y una comprensión mayor de los datos, especialmente en 3D. Dentro de las múltiples utilizaciones que puede tener la plataforma, cabe destacar las más comunes: planificación urbana y territorial, gestión de recursos naturales, investigación científica y superposición de modelados.

La elección de ArcGIS para la visualización de los mapas creados y su georreferenciación se basa en la gran compatibilidad y las posibilidades que ofrece la plataforma en cuanto al tratamiento de nubes de puntos, además de la posibilidad de combinación de distintos tipos de datos, lo cual permite la utilización de forma simultánea de capas con información tanto 2D como 3D con capas de nubes de puntos. Esto permite la capacidad de analizar los mapas de puntos generados mediante comparación con la información de las capas 3D proporcionadas por la plataforma, así como analizar la calidad de la información del sensor GPS.

4. IMPLEMENTACIÓN

A continuación, se detalla el procedimiento de generación de los mapas de puntos 3D utilizando como ejemplo los mapas generados durante la realización del proyecto. El edificio mapeado es el de la ETSII (Escuela Técnica Superior de Ingeniería Informática), con dirección Bulevar Louis Pasteur 35, Puerto de la Torre, 29071 Málaga[4].

Figura 21: Ubicación del edificio objeto del mapeado en la ciudad de Málaga.

El paso inicial del proceso es la creación de un archivo rosbag que capture los mensajes enviados por los topics de interés durante el mapeo para poder analizar la información disponible y poder construir los mapas de puntos offline y realizar múltiples reproducciones para obtener la información de interés. Los topics capturados por rosbag durante los mapeos se muestran en la Tabla 4.

[4] Enlace a la ubicación del edificio en Google Maps: https://maps.app.goo.gl/91JtpTH7KYEyUPkB7

Nombre	Tipo	Información
/ouster/points	sensor_msgs/msg/PointCloud2	Nube de puntos capturada por el LiDAR
/ouster/imu	sensor_msgs/msg/Imu	Aceleración lineal y velocidad angular (6 ejes) capturadas por la IMU del Ouster
/hunter/fix	sensor_msgs/msg/NavSatFix	Información GPS

Tabla 4: Topics capturados por rosbag durante los diferentes mapeos realizados.

Con este conjunto de datos es posible tanto obtener información global del mapeo (número de mensajes publicados por topics, frecuencias, discontinuidades, etc) como replicar la situación en cualquier momento.

4.1. CREACIÓN DEL MAPA CON FAST-LIO2

Para utilizar la herramienta FAST-LIO2 en la generación de mapas de puntos 3D es necesario realizar los siguientes ajustes:

- Ajuste del código para la utilización del modelo del LiDAR utilizado y depuración requerida (el código original está enfocado a un LiDAR LIVOX).
- Ajuste de las características específicas del sistema robótico, como son el nombre de los topics utilizados y la posición relativa del LiDAR respecto de la IMU.

Con el objetivo de procesar la información tridimensional obtenida a través del LiDAR, es necesario introducir la transformación de la pose del sensor LiDAR respecto del sensor IMU. Para el sensor utilizado en este proyecto, las matrices de transformación son las siguientes[5]:

$$T_{lidar\_to\_sensor} = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0.036180 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2)$$

$$T_{imu\_to\_sensor} = \begin{bmatrix} 1 & 0 & 0 & 0.006253 \\ 0 & 1 & 0 & -0.011775 \\ 0 & 0 & 1 & 0.007645 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3)$$

En primer lugar, será necesario obtener la matriz de transformación de sensor a LiDAR, la cual se puede calcular como inversa de la matriz de transformación de LiDAR a sensor, compuesta por el vector de rotación traspuesto y el vector de traslación con el signo opuesto (Ecuación 4).

$$T_{sensor\_to\_lidar} = (T_{lidar\_to\_sensor})^{-1} = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & -0.036180 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4)$$

[5] Matrices obtenidas del manual de usuario del sensor Ouster OS1 con unidades en metros. Disponible en: https://data.ouster.io/downloads/software-user-manual/software-user-manual-v2p0.pdf

La matriz de transformación de la pose del sensor IMU a la del sensor LiDAR se obtiene de la multiplicación de las matrices anteriores. El resultado para este caso es la matriz de transformación de la Ecuación 5.

$$T_{imu\_to\_lidar} = T_{sensor\_to\_lidar} \cdot T_{imu\_to\_sensor} = \begin{bmatrix} -1 & 0 & 0 & -0.006253 \\ 0 & -1 & 0 & 0.011775 \\ 0 & 0 & 1 & -0.028535 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5)$$

Por otro lado, se ha modificado el código principal para generar pequeños mapas independientes de forma periódica utilizando un iterador para facilitar una gestión más eficiente de los datos y evitar desbordamientos. Esta implementación permite además la reubicación de los primeros archivos de forma que se libere constantemente la memoria disponible posibilitando la creación de mapas de grandes extensiones sin limitaciones por el espacio de almacenamiento disponible. Estos mapas de bajo volumen pueden ser combinados a posteriori para la generación del mapa global. Una vez ajustado el código se lanza el nodo de mapeo mediante el siguiente comando:

ros2 launch fast_lio mapping_ouster32.launch.py

Esto abre automáticamente una ventana de RVIZ2. De forma simultánea, para la generación del mapa de manera offline (también se puede realizar sin el paso intermedio por un archivo bag dada la velocidad de los resultados producidos por FAST-LIO2) se reproduce el bag mediante el comando:

ros2 bag play "nombre del archivo bag"

Una vez se reciben los primeros mensajes de nubes de puntos, estas comienzan a mostrarse interpretando el movimiento del robot en el momento del mapeo si la información de los sensores es buena y los ajustes necesarios se han realizado correctamente (ver Figura 22).

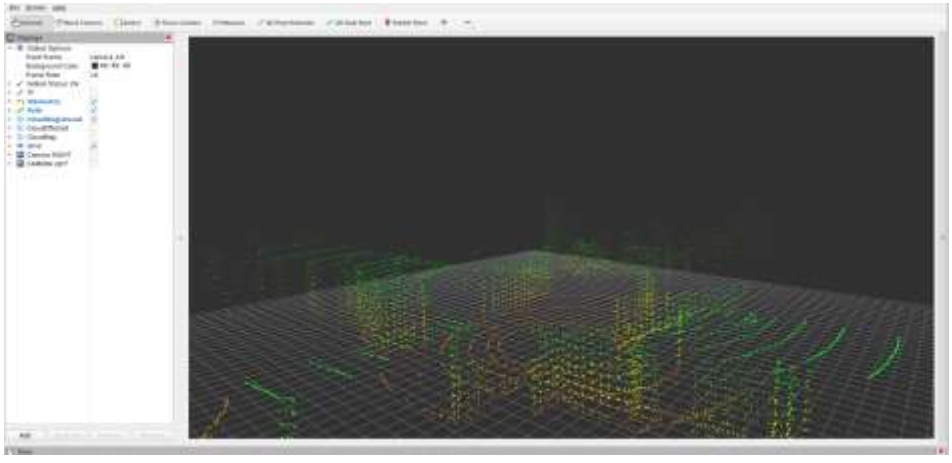


Figura 22: Primeras nubes de puntos capturadas y enviadas por el LiDAR.

El diagrama de ROS obtenido por RQT para el funcionamiento de FAST-LIO2 con un archivo bag se muestra en la siguiente imagen:

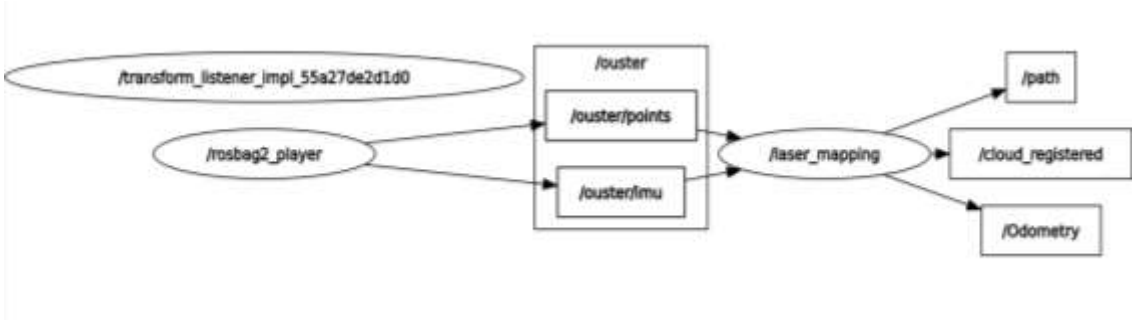


Figura 23: Diagrama de comunicación rqt_graph durante la ejecución FAST-LIO2 con la reproducción de un archivo bag de ROS2.

En la Figura 24 se muestra el proceso de construcción del mapa del archivo rosbag 2 desde el comienzo.

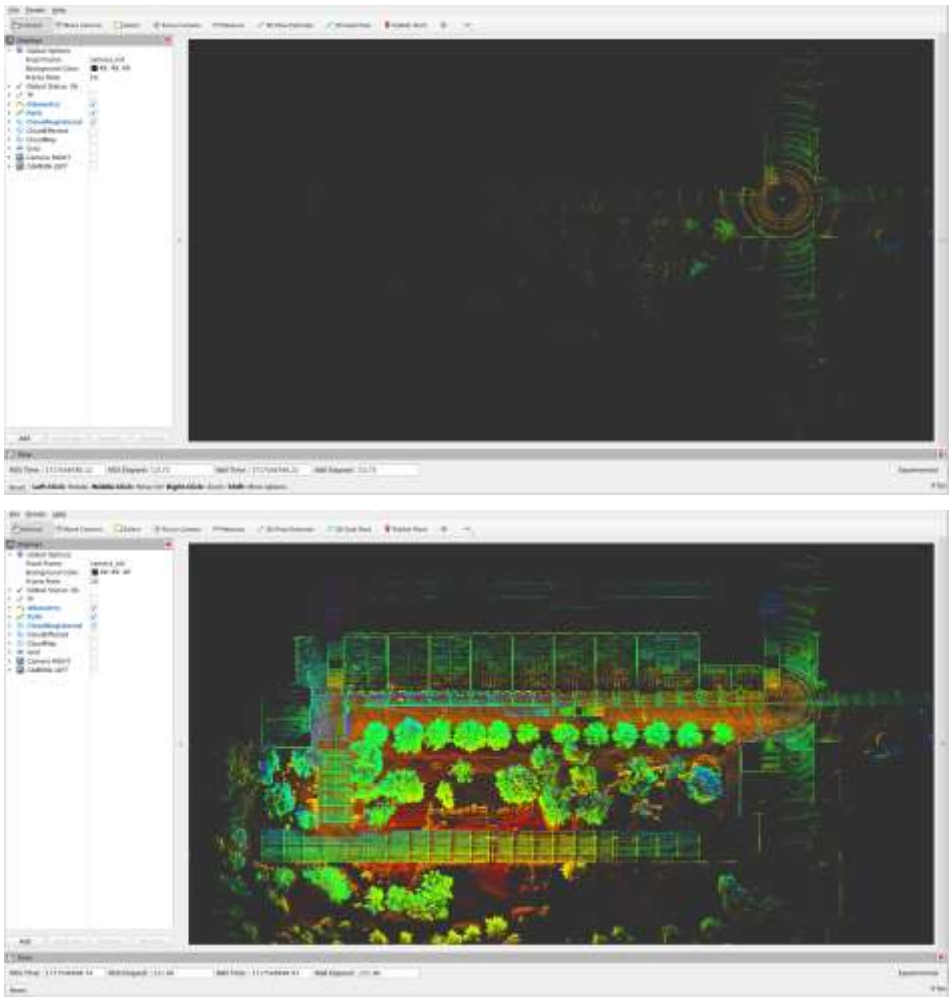


Figura 24: Proceso de construcción del mapa del archivo rosbag 2. *En la imagen superior se muestra la posición inicial del robot antes de comenzar a desplazarse desde una vista cenital. En la imagen inferior*

se muestra el mapa de puntos de la parte exterior del edificio casi completo tras el desplazamiento circular del robot.

El sistema de referencia se define como la posición inicial del robot, y a partir de este se añaden continuamente los puntos de correspondencia utilizando la información del sensor IMU y de las propias nubes. En la Figura 25 se puede observar la información generada tanto de trayectoria como de pose del robot (representada por el conjunto de ejes) durante el mapeado.

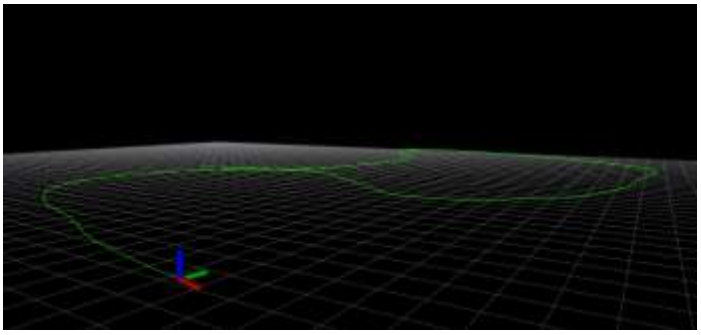


Figura 25: Trayectoria y pose del robot generadas durante la ejecución de FAST-LIO2.

4.2. PROCESAMIENTO DE LAS NUBES DE PUNTOS

Una vez se obtienen las nubes de puntos iniciales, es necesario llevar a cabo una serie procedimientos de post-procesado. Mediante la combinación de varias de estas nubes, es posible la creación de mapas de mayor o menor tamaño. Para esto basta con concatenar los puntos de los distintos archivos, dado que cada nube de puntos resulta de la traslación y rotación de las nubes generadas por el LiDAR según la pose del robot en cada momento.

Tras la combinación de todos los puntos en un único mapa, es conveniente realizar un filtrado debido a la menor precisión de las medidas del sensor LiDAR en relación al aumento de la distancia al objeto, o bien por un posible interés en una zona o sección concreta, como puede ser la visualización de cortes horizontales de edificios (ver Figura 26).

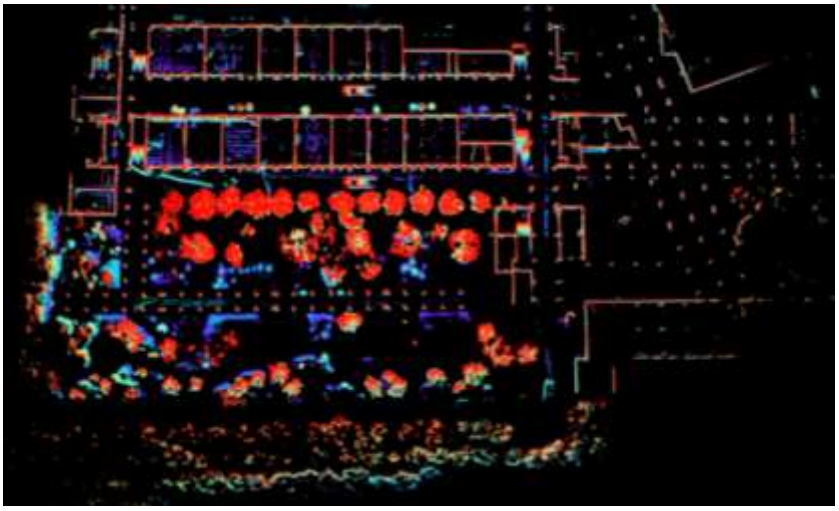


Figura 26: Mapa de la rosbag 2 filtrado en z con sigma 1. *Gracias al filtrado se puede observar la planta del edificio, así como el corte horizontal de los elementos presentes (véase árboles y columnas).*

Este filtrado es conveniente para una visualización clara de las nubes de puntos mediante una distribución de color según el valor relativo de la componente de cada punto en el mapa, ya que la eliminación de los puntos más alejados del núcleo del mapa permite que la información se encuentre más concentrada. A continuación, se muestra la comparativa de la visualización de un mapa de puntos original y un mapa de puntos filtrado en z.

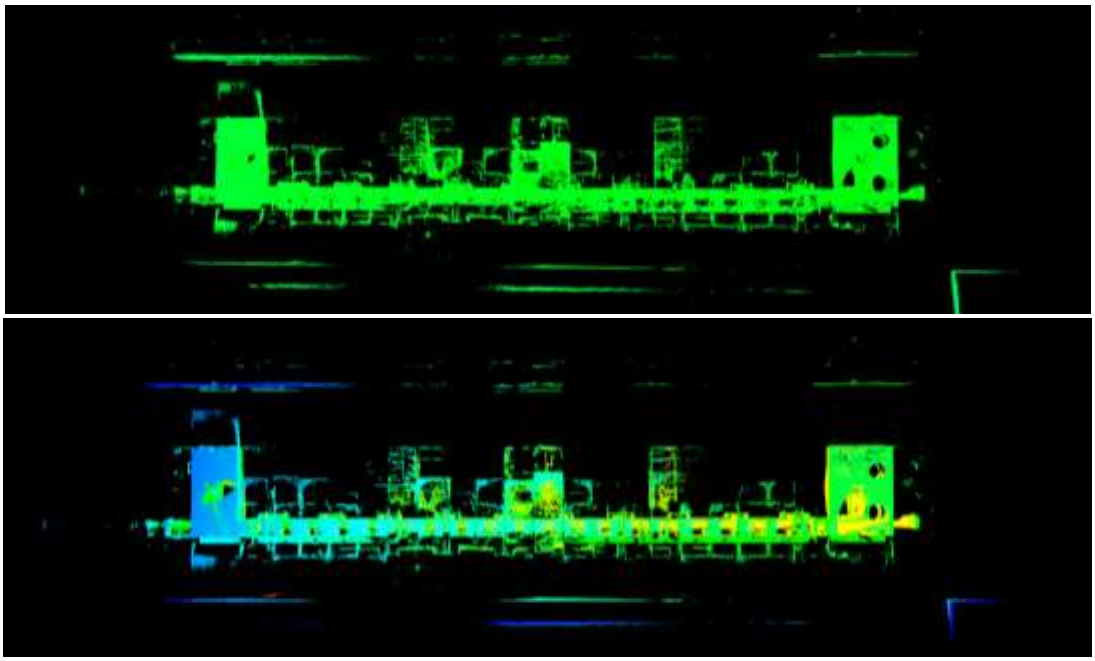


Figura 27: Variación de la visualización según el campo Z al filtrar los puntos más alejados (II) en comparación al mapa original (I). *Se observa con mayor claridad la información que contiene el mapa de puntos al realizar el filtrado dada la mayor variación del color por altura.*

Otro de los pasos más habituales en el post-procesamiento de nubes de puntos es el de voxelización, debido al aumento del tamaño de los datos según crece la escala de la zona mapeada. Dada la generación de pequeños submapas independientes, la voxelización puede realizarse tanto para cada submapa antes de realizar la combinación, como de una única vez para el mapa completo. Cada opción presenta ventajas y puntos a analizar. Para el caso del voxelizado del mapa completo, las ventajas son una matriz de puntos homogénea y una única ejecución. En el caso del voxelizado por submapas, destaca como punto más favorable la posibilidad de crear grandes mapas sin tener que manejar volúmenes de datos significativamente grandes. Este último método tiene como desventaja la posibilidad de realizar un cálculo incorrecto de los centros de los vóxeles por no tener en cuenta la disposición de los distintos submapas entre ellos, lo que podría dar lugar a irregularidades por las que la zona entre puntos podría llegar a superar notablemente la distancia de voxelización.

Con el objetivo de disminuir la carga computacional, se lleva a cabo la voxelización de los mapas completos estudiando el valor del tamaño de voxelización que optimice la relación entre el volumen de los datos y la precisión de la información (ver Figura 28).

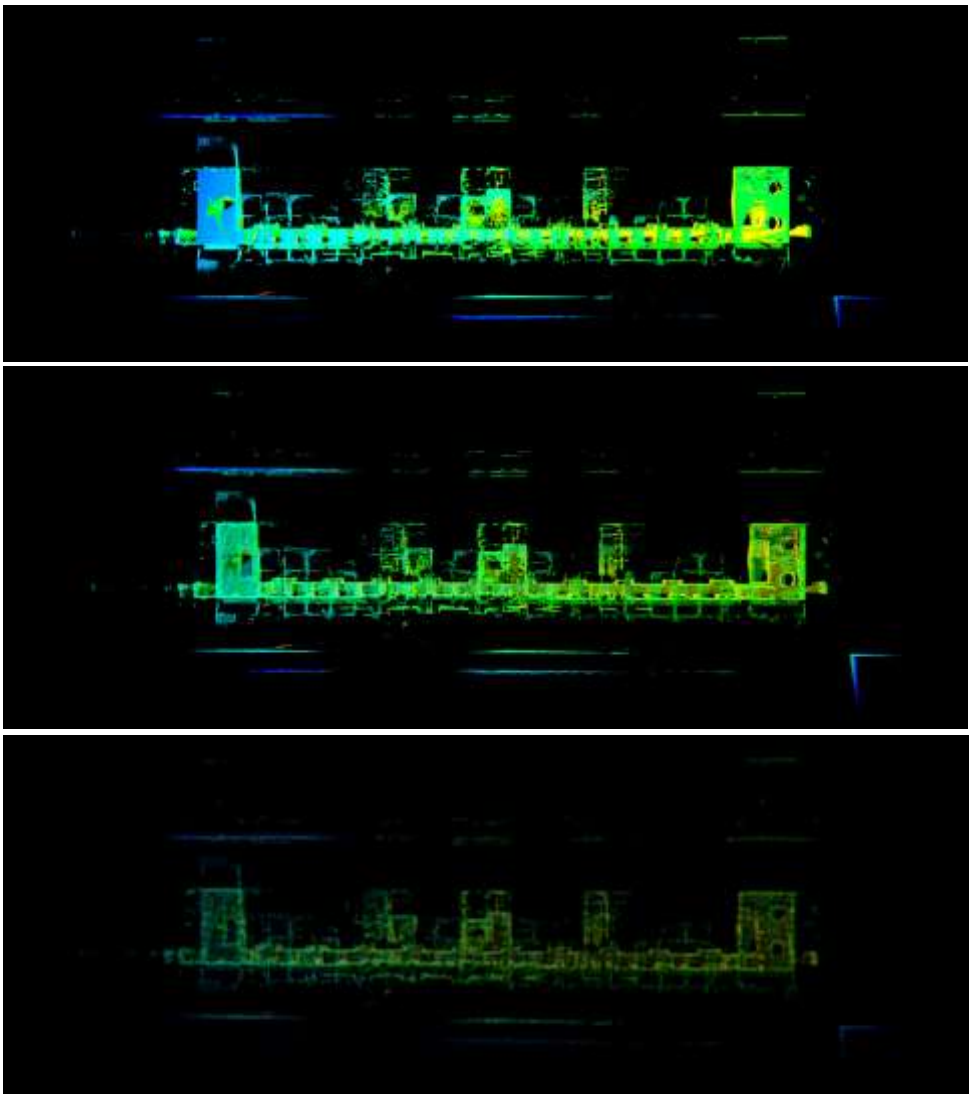


Figura 28: Comparativa de la vista cenital del mapa de puntos del Mapa 3 sin voxelización (I), con voxelización de 0.2 metros (II) y con voxelización de 0.5 metros (III). *Se aprecia una clara disminución de la densidad de puntos y la pérdida de información conforme aumenta el tamaño de voxelización.*

La variación del tamaño de voxelización tiene un fuerte impacto en el tamaño del archivo, como se muestra en la Tabla 5.

Tamaño de voxelización (m)	Tamaño del archivo (KB)	Reducción de tamaño (%)
0	352246	100
0.01	43746	12.4191
0.02	42059	11.9402
0.05	36099	10.2482
0.1	23755	6.7439
0.2	11961	3.3956
0.5	2594	0.7364
1	617	0.1752

Tabla 5: Variación del tamaño de un archivo de puntos 3D de extensión pcd según el tamaño de voxelización.

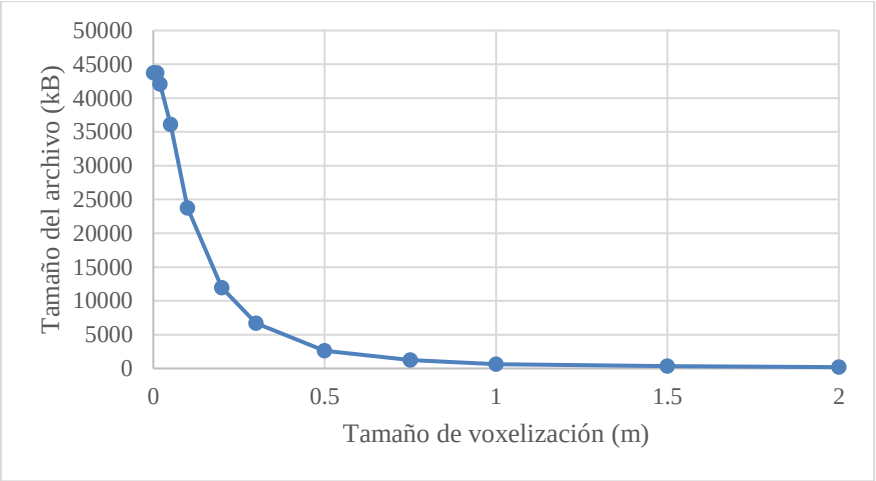


Figura 29: Variación del tamaño de un archivo de puntos 3D de extensión pcd según el tamaño de voxelización. *Se puede observar una tendencia clara de disminución del tamaño del archivo conforme aumenta el tamaño de voxelización similar a una relación hiperbólica.*

Con el objetivo de posibilitar la compatibilidad de los mapas generados y permitir su lectura por herramientas software de visualización de nubes de puntos y otros tipos de datos, se ha desarrollado un código en lenguaje Python que permite convertir los archivos con extensión pcd (Point Cloud Data) a archivos con extensión las, utilizada para los datos obtenidos por sensores láser LiDAR. De esta forma, los archivos que contienen la información de los mapas pueden ser comparados con capas de diferentes tipos de datos en programas como ArcGIS.

4.3. MAPAS GENERADOS

A continuación, se detalla la información característica de los mapas de exteriores generados durante el desarrollo del proyecto convenientemente post-procesados. Asimismo, se muestran descriptivas de los mapas. A modo de complemento, se proporcionan la información y las imágenes descriptivas de un mapa adicional generado como prueba de mapeado en interiores en el Anexo 1.

4.3.1. Mapa 1 – Patio exterior

El primer mapa construido resulta del trayecto del robot por el patio exterior del edificio. Para la generación de este mapa, se ha recorrido una distancia de 90.80 m durante el mapeo, generándose 14 archivos bag con un tamaño total de 1.86 GB. Este mapa se utiliza además para las pruebas de localización (ver el apartado PRUEBAS DE LOCALIZACIÓN Y RESULTADOS). En el siguiente enlace se puede visualizar la generación de este mapa: GENERACIÓN Y VISUALIZACIÓN DEL MAPA 1

En la Tabla 6 se muestran las características del Mapa 1.

Característica	Valor
Nubes fusionadas	1266
Número de puntos procesados	41484288
Número de puntos del mapa inicial	6419565
Número de puntos tras voxelizar	1114421
Tamaño del archivo inicial	150459 KB
Tamaño del archivo tras voxelizar	13060 KB

Tabla 6: Características del Mapa 1.

A continuación, se muestran el recorrido realizado durante el mapeado (Figura 30), la vista cenital y las dimensiones del Mapa 1 (Figura 31) e imágenes descriptivas.

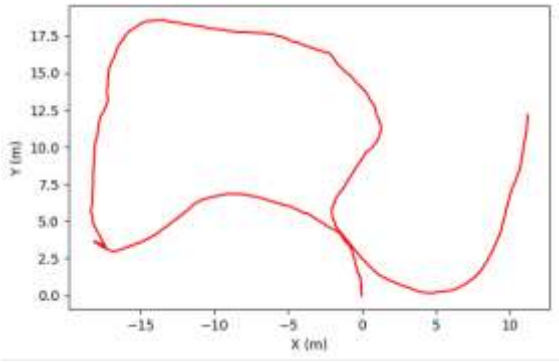


Figura 30: Trayectoria recorrida por el robot durante el primer mapeado.

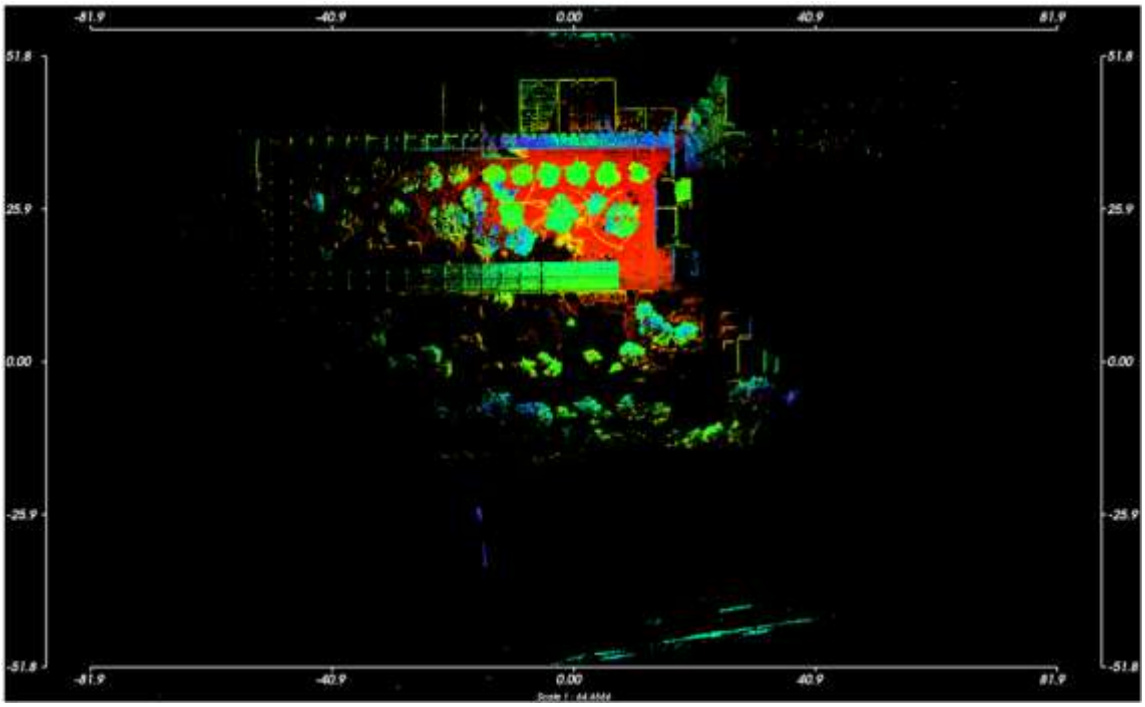


Figura 31: Vista cenital y medidas en metros del Mapa 1. *Se puede apreciar la detección de puntos en la fachada del edificio que se encuentra al otro lado de la carretera.*

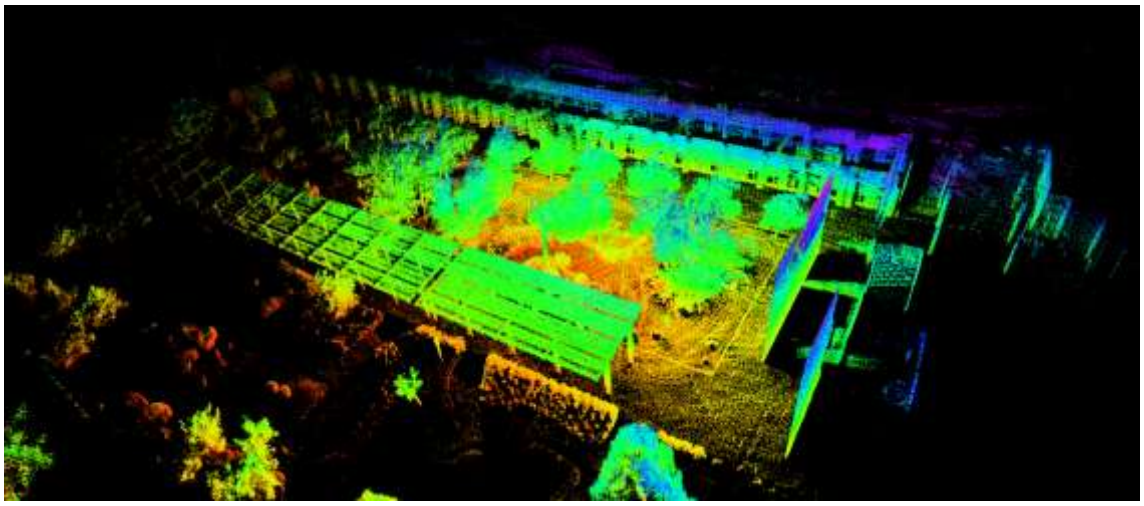


Figura 32: Vista oblicua del Mapa 1.

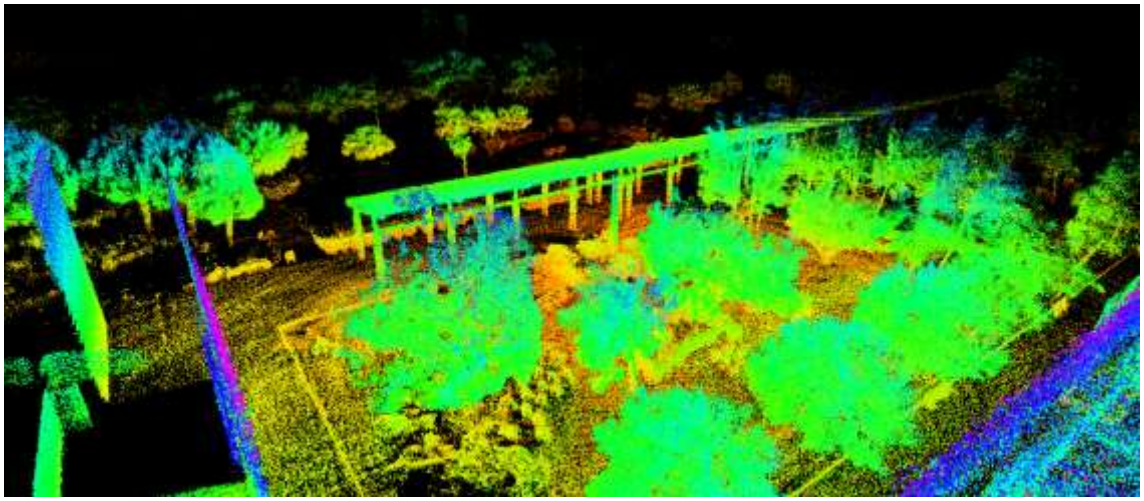


Figura 33: Vista del patio exterior.

4.3.2. Mapa 2 – Edificio Completo

Con el objetivo de crear un mapa del edificio completo, se ha recorrido la planta baja alternando tanto zonas exteriores como interiores. Durante este mapeo, el robot ha recorrido una distancia de 959.18 metros y se han generado 30 archivos bag con un tamaño total de 3.01 GB. En el siguiente enlace se puede visualizar la generación de este mapa: GENERACIÓN Y VISUALIZACIÓN DEL MAPA 2

En la Tabla 7 se muestran las características del mapa y los archivos generados. Desde la información generada por el sensor LiDAR hasta el mapa voxelizado se ha llevado a cabo la reducción del número de puntos en un 97.6583 %. Esto se ha producido en dos pasos secuenciales: durante la ejecución de FAST-LIO2 se han seleccionado el 11.9387 % de los

puntos por correspondencia; mientras que, tras la voxelización con tamaño de 1 centímetro, el número de puntos resultante representa un 19.6145 % de los puntos existentes previamente.

Característica	Valor
Nubes fusionadas	8597
Número de puntos procesados	281706496
Número de puntos del mapa inicial	33632054
Número de puntos tras voxelizar	6596773
Tamaño del archivo inicial	788252 KB
Tamaño del archivo tras voxelizar	77307 KB

Tabla 7: Características del Mapa 2.

A continuación, se muestran la trayectoria recorrida por el robot durante el segundo mapeado (Figura 34), la vista cenital y las medidas del Mapa 2 (Figura 35) y varias imágenes descriptivas.

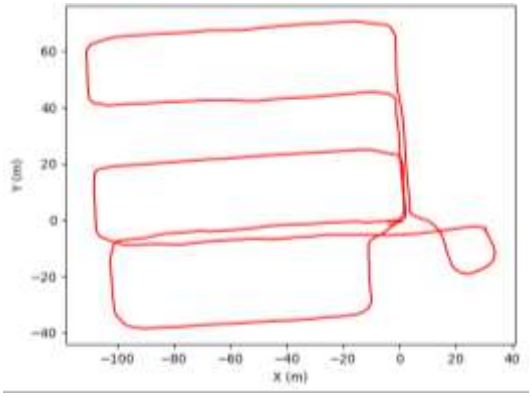


Figura 34: Trayectoria recorrida por el robot durante el segundo mapeado.

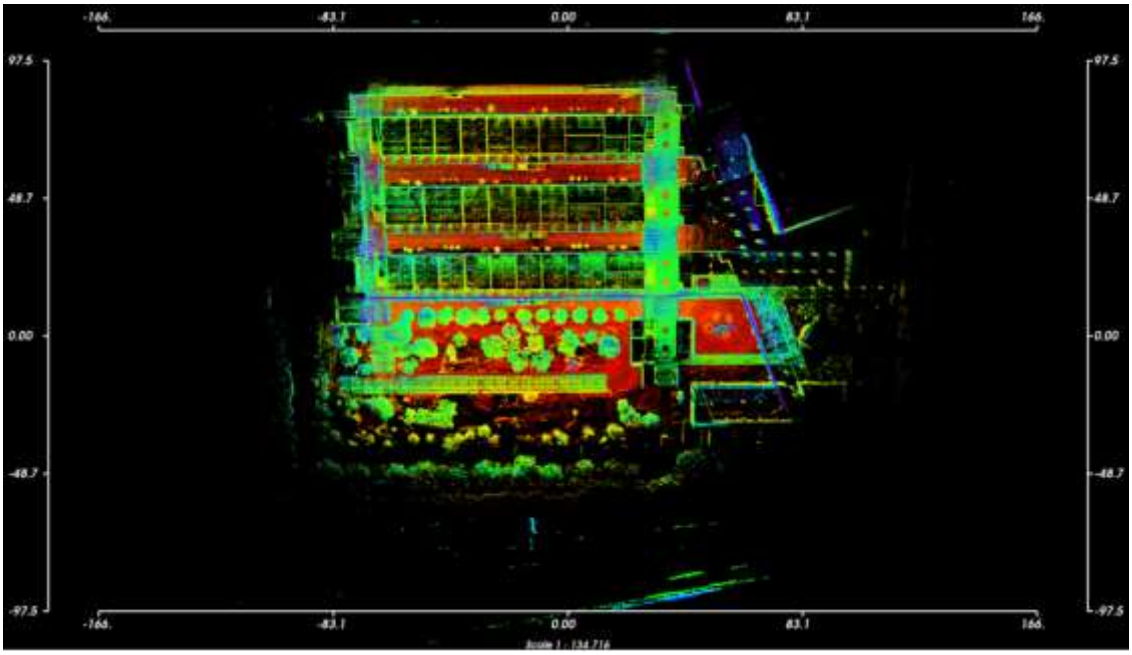


Figura 35: Vista cenital y medidas en metros del Mapa 2.

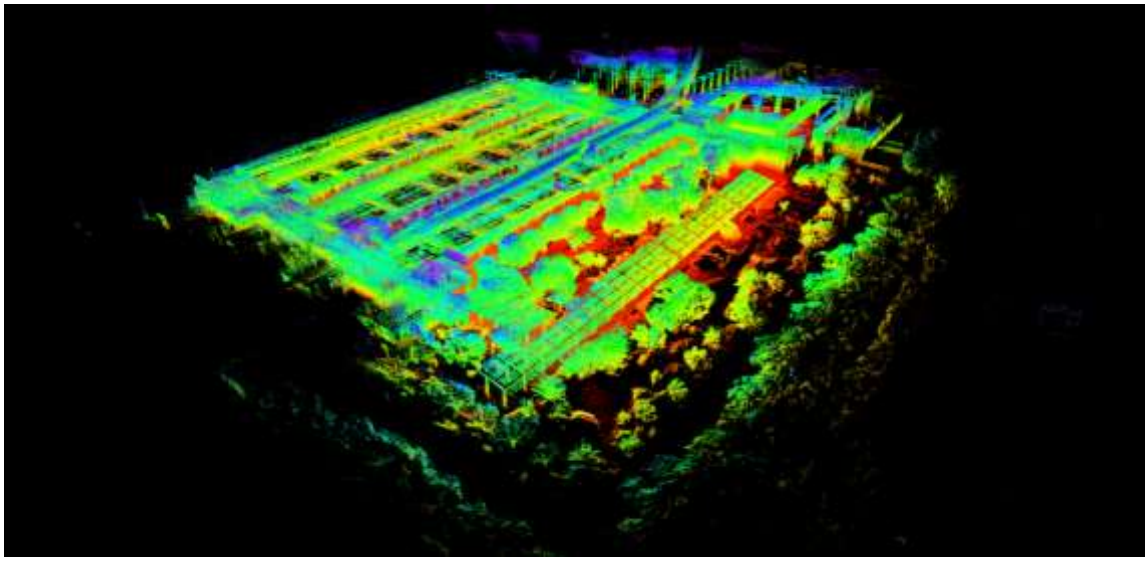


Figura 36: Vista oblicua del Mapa 2.

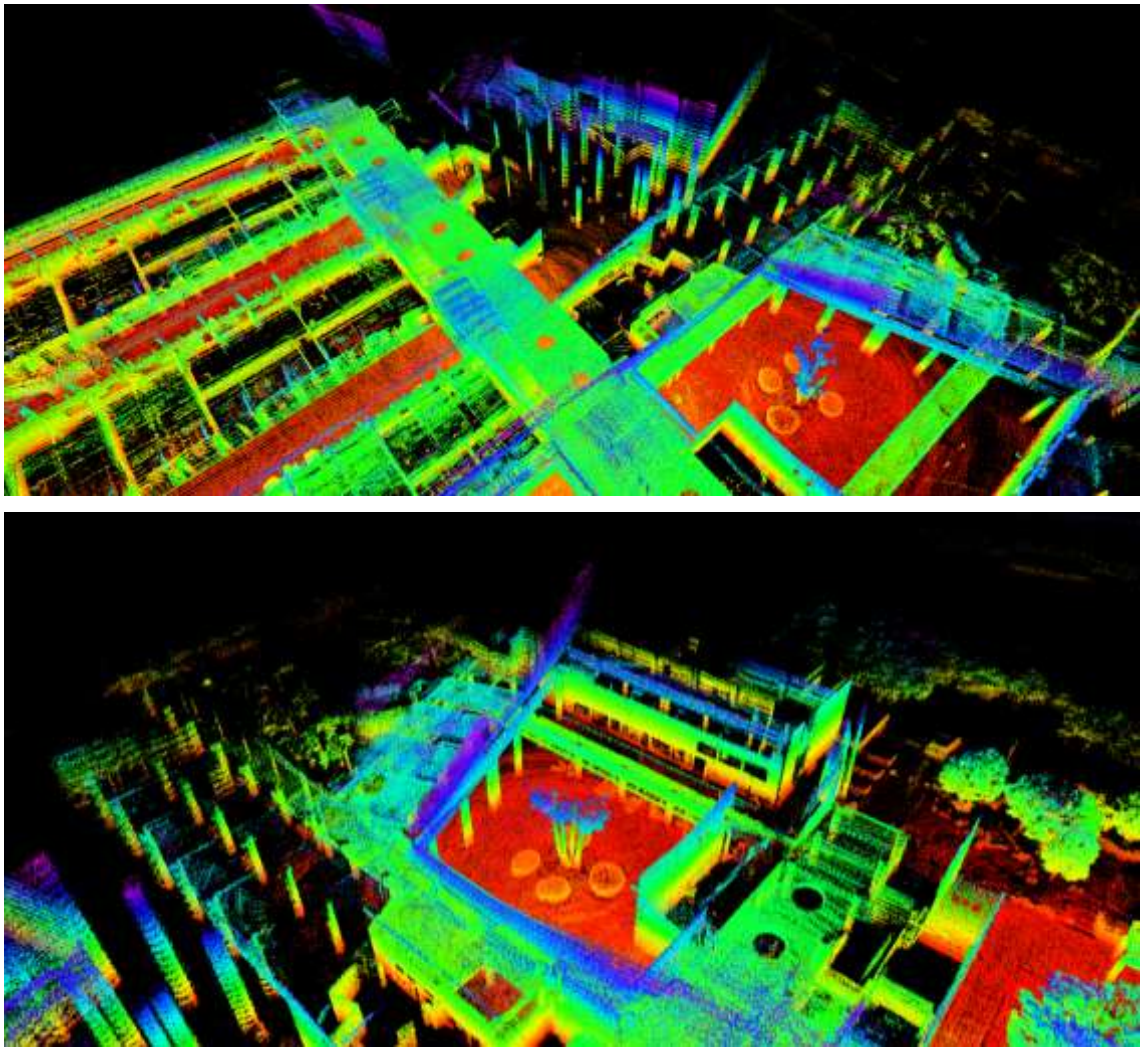


Figura 37: Detalles de la zona de columnas y del patio interior.

4.3.3. Georreferenciación y orientación

Dada la problemática asociada a la información obtenida por los sensores GPS su modo de funcionamiento, se requiere de un análisis de los datos disponibles y de su precisión para conseguir una georreferenciación correcta. Las situaciones adversas a las que se enfrentan los sensores de este tipo, como a la hora de medir la localización en entornos en los que hay presentes edificios y otros elementos, producen fenómenos como el decremento sustancial de la precisión o incluso la discontinuidad de los datos generados (fenómeno altamente probable en situaciones similares las que se han estudiado de mapeado de semiexteriores).

Por tanto, se ha desarrollado un procedimiento que busca utilizar la información con mayor precisión para la georreferenciación y orientación del mapa. Para esto, se comienza con el desarrollo de un programa en lenguaje Python que analiza la precisión de las mediciones del GPS buscando actualizar la información de los sensores para los mensajes del GPS con covarianza mínima en el eje x medida hasta el momento. Por tanto, este programa almacena la medida del GPS y la información de la odometría del robot para el momento de mayor precisión. Durante la reproducción del archivo rosbag1 se obtiene la evolución de las mejoras de covarianza en x recogida en la Tabla 8.

Tiempo transcurrido	Covarianza en x
5.26904869	2.893401
6.27160501	0.933156
7.26802301	0.603729
8.26924276	0.262144
17.2686009	0.186624
19.2722156	0.173056
20.2750847	0.16
21.2746713	0.135424
22.2845304	0.123904
24.2706292	0.1024
38.2752132	0.092416
42.2787199	0.082944
49.2809119	0.063504

Tabla 8: Evolución de la covarianza del GPS en x para la Rosbag1. *Se observa un decremento de la escala de metros a la escala de centímetros en la covarianza en x.*

Para la información de la orientación, se ha utilizado el cálculo de la orientación en el planeta mediante el cálculo del ángulo en el plano xy (yaw) del vector generado por dos medidas de GPS en el primer movimiento del robot (momento inicial y final del primer recorrido en línea recta) sobre la horizontal. Dado que el robot comienza el recorrido en dirección positiva del eje x del sensor IMU según está dispuesto en el sistema robótico y el mapa comparte este sistema de referencias, la orientación del mapa se consigue de la rotación del ángulo yaw medido. Para el primer mapa generado se ha medido experimentalmente un desfase angular de 0.6º, mientras que para el segundo mapa generado ha sido de 3.7º. Esto puede deberse a la inexactitud del GPS y la posible desviación del robot en el primer movimiento. Por ello, una alternativa de mayor precisión resulta de la utilización de un sensor

con información sobre la orientación absoluta como una brújula digital o un sensor IMU de 9 ejes en el instante inicial del mapeo.

Una vez se obtiene esta información, se realiza la georreferenciación mediante 3 pasos secuenciales:

1) En primer lugar, se lleva a cabo el giro con centro en el origen del sistema de referencias inicial de la nube según la información de orientación absoluta obtenida al principio del recorrido.
2) A continuación, se utilizan las coordenadas de la odometría del robot en el momento de mayor precisión del GPS para la definición de un nuevo sistema de referencias con centro en ese punto. Esto se consigue restando a las coordenadas de cada punto de la nube las coordenadas de la odometría.
3) Por último, y dado que el mapa ya se encuentra orientado y se conocen las coordenadas medidas por el GPS en el punto que ahora sirve de origen, basta con sumar a las coordenadas de cada punto del mapa 3D las coordenadas x e y según UTM correspondientes a la latitud y longitud medidas.
   Para esto, se realiza la conversión de formato entre códigos numéricos EPSG, códigos utilizados para cartografía e identificación de sistemas de coordenadas. Las coordenadas iniciales aportadas por el GPS se encuentran en el código EPSG:4326 del World Geodetic System 1984 (WGS84) y se convierten a EPSG:25830, el cual se utiliza para ubicar puntos en la zona UTM 30 N[6], a la que pertenece Málaga (Figura 38).

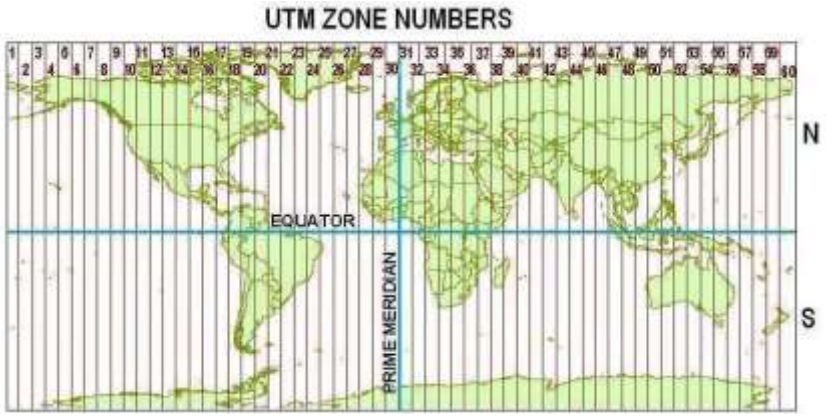


Figura 38: Mapa del mundo con la división por zonas según UTM. *Se puede apreciar la ubicación de Málaga en la zona 30 N. Fuente:* ***https://www.researchgate.net/figure/Universal-Transverse-Mercator-UTM-coordinate-system-is-a-standard-set-of-map_fig3_342330834***

[6] En este caso, la letra N se refiere a la división norte del planeta. Existe una división por latitudes en la que Málaga se encuentra en el cuadrante 30 S.

4.4. VISUALIZACIÓN JUNTO A MODELOS 2D Y 3D DE LA ZONA

La georreferenciación de los mapas, junto con las posibilidades de representación de distintos tipos de datos superpuestos, permite la comparación de estos con modelos tanto 2D como 3D de la zona mapeada. A continuación, se muestran diferentes vistas de los mapas junto con capas de distinto tipo. La inclusión de los mapas debidamente georreferenciados en el sistema de coordenadas geoespacial se describe en el Anexo 2. Para la verificación de la georreferenciación y de la validez de los mapas, se utiliza una capa 3D de edificaciones de la zona de Málaga.

4.4.1. MAPA 1 – PATIO EXTERIOR

A continuación, se muestran diferentes vistas del mapa de puntos generado a partir del archivo rosbag 1 en superposición a diferentes capas tanto 2D como 3D.

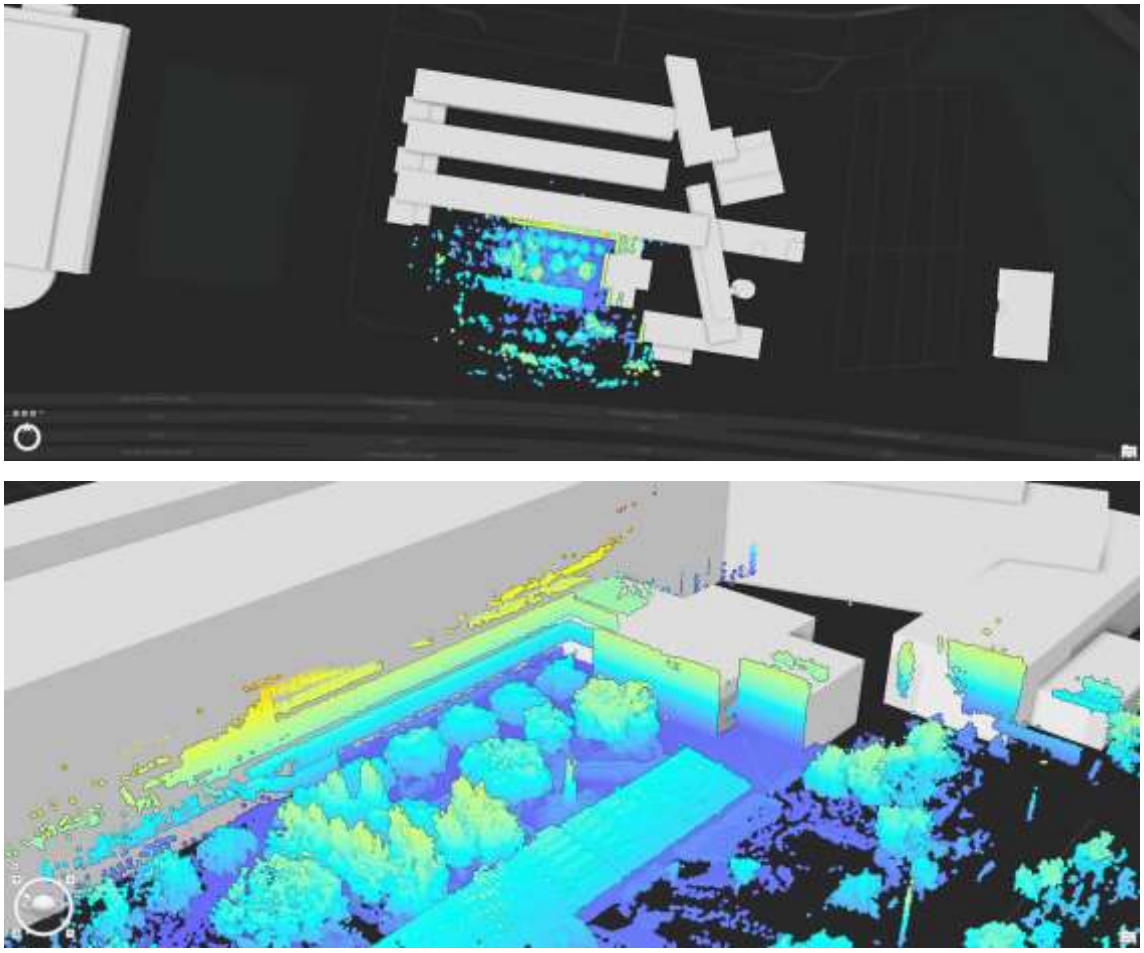


Figura 39: Vista cenital y vista oblicua del Mapa 1 junto al modelo 3D del edificio.

4.4.2. MAPA 2 – EDIFICIO COMPLETO

A continuación, se muestran las vistas correspondientes al mapa generado a partir del archivo rosbag 2, este contiene información tridimensional de todo el edificio, por lo que se puede realizar la comparación en un mayor número de zonas de interés.

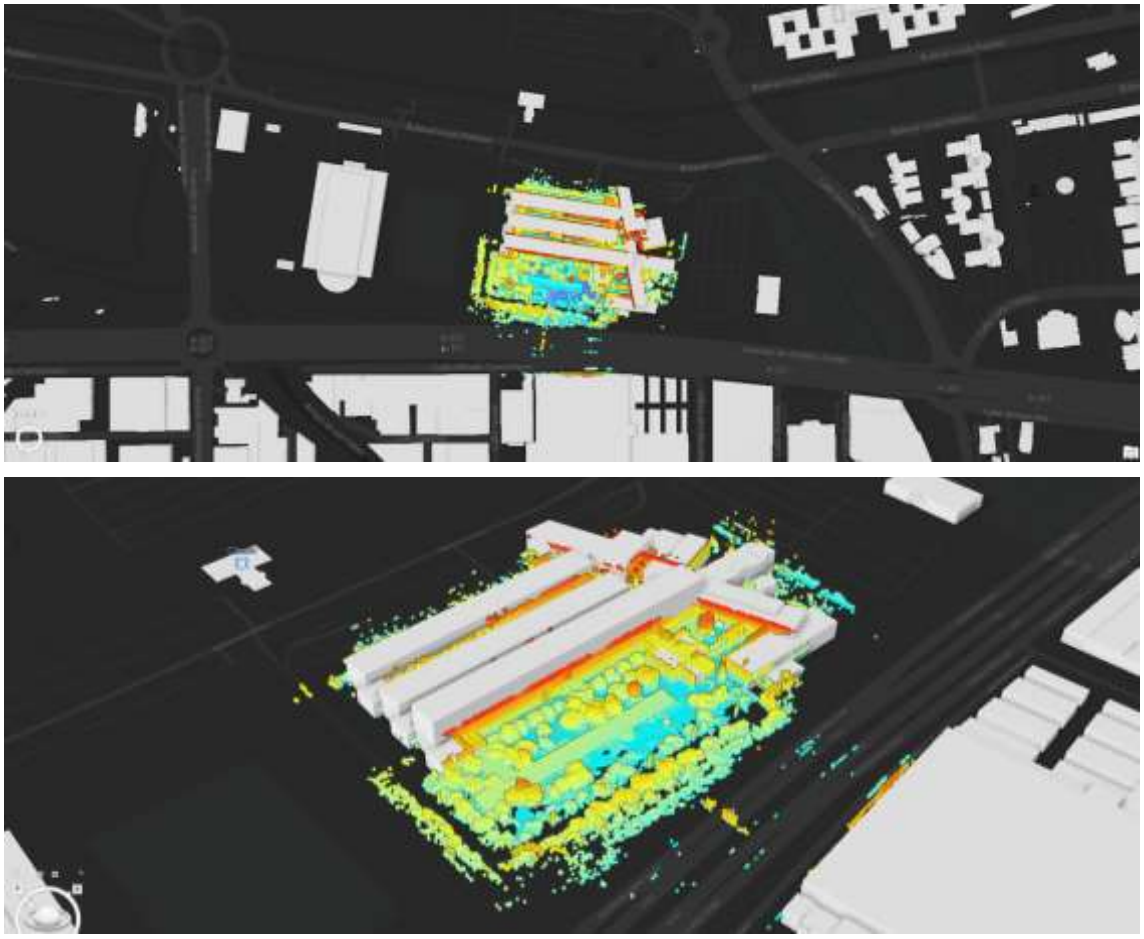


Figura 40: Vista cenital y vista oblicua del Mapa 2 junto al modelo del edificio.

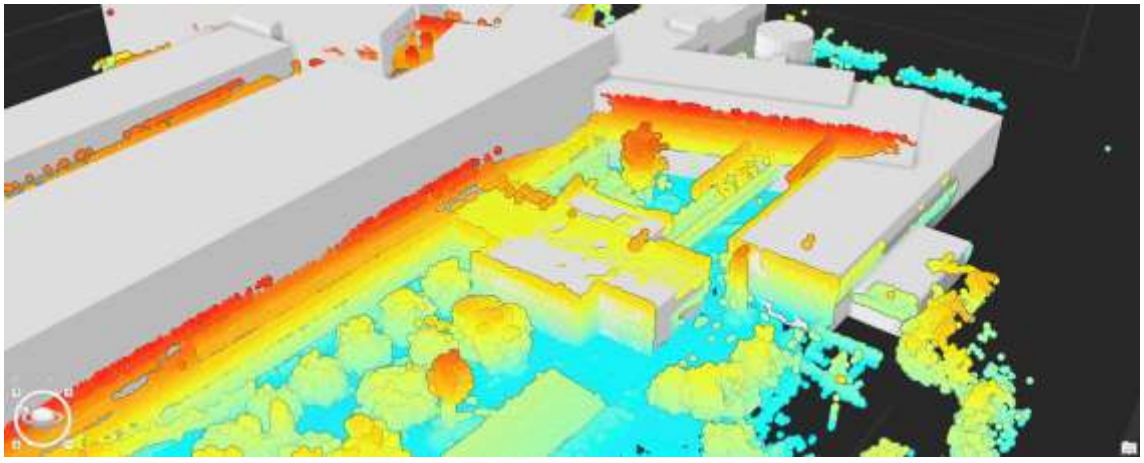


Figura 41: Detalle de coincidencia de las distintas paredes exteriores del edificio con el mapa de puntos.

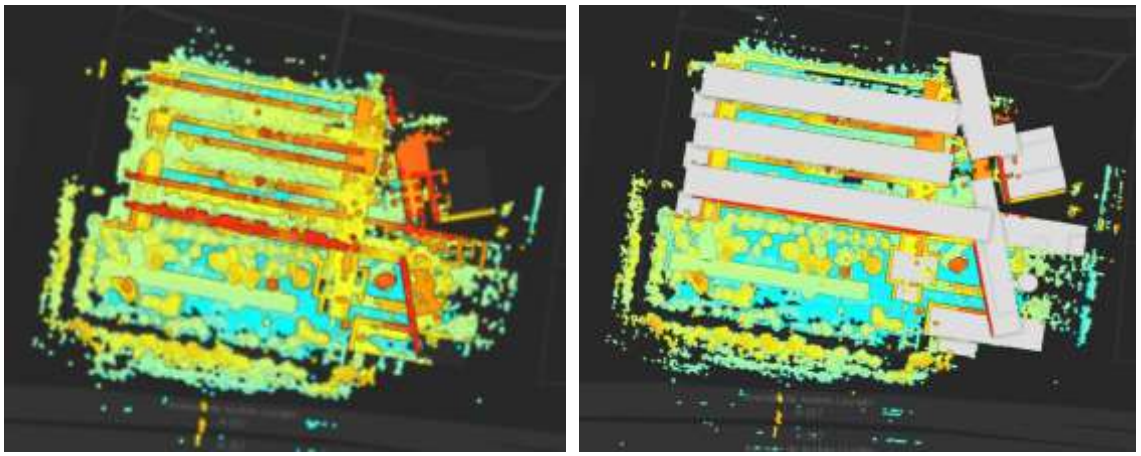


Figura 42: Comparativa del mapa sin la superposición del modelo del edificio (I) y con la superposición del modelo (II).

Otra de las visualizaciones que resulta de ayuda para la interpretación de la zona del edificio, es la visualización junto a capas de imágenes. Esto se muestra en la Figura 43.

Figura 43: Vistas de la superposición del Mapa 2 a la capa de imágenes de la ciudad de Málaga y a la capa de los modelos 3D de los edificios.

Se adjuntan más imágenes de detalle de los mapas generados en el Anexo 3.

5. PRUEBAS DE LOCALIZACIÓN Y RESULTADOS

Una vez comprobada la similitud de los mapas de puntos 3D generados con los modelos del edificio, se analiza la utilización de estos en aplicaciones relacionadas con la localización de robots móviles. Partiendo del mapa de puntos, las nubes de puntos obtenidas por el sensor LiDAR y una pose odométrica inicial, se analizan las posibles mejoras que se pueden obtener utilizando el algoritmo de registro de nubes de puntos ICP para la estimación de la posición del dispositivo en cada momento, realizándose así una corrección de la odometría inicial.

Las pruebas realizadas se basan en la utilización de la odometría generada por FAST-LIO2 durante su ejecución, a la cual se le añaden distintos tipos de ruido (tanto offsets constantes como ruidos estadísticos basados en una distribución gaussiana) de forma que se pueda analizar el error que generan tanto la pose odométrica como la estimación obtenida mediante la aplicación de ICP. La variación de los parámetros presentes tanto en el algoritmo como en la geometría propia del problema genera a su vez una variación en los resultados obtenidos, por lo que se debe analizar cómo afecta cada parámetro con el objetivo de realizar ajustes que optimicen los errores obtenidos. Entre estos parámetros destacan el número de iteraciones, el valor del error de la pose odométrica y el tamaño de voxelización de las nubes de puntos. Además, es importante estudiar el desempeño de la aplicación del algoritmo ante diferentes tipos de ruido. Para conseguir la reproducibilidad de las pruebas planificadas, se han generado archivos bag con los siguientes datos:

Nombre	Tipo	Información
/ouster/points	sensor_msgs/msg/PointCloud2	Nube de puntos capturada por el LiDAR
/Odometry	nav_msgs/msg/Odometry	Posición y orientación del robot en cada momento

Tabla 9: Topics capturados por rosbag para la realización de pruebas de mejora de estimación inicial de odometría.

De esta forma, se tiene en un mismo archivo la información que recibe el sensor LiDAR y la información de la pose del robot de forma coordinada. El topic */Odometry* contiene la información de la odometría estimada por la herramienta FAST-LIO2 durante la creación de los mapas, siendo esta odometría la que se utiliza como ground truth o trayectoria real durante las pruebas. Con este conjunto de datos, se pueden posicionar tridimensionalmente los puntos detectados por el LiDAR en cada momento, siendo posible superponer estas nubes al mapa generado, como se muestra en la Figura 44.

Una vez se tienen los archivos descritos para los tres mapas construidos, se ha procedido a la creación de un código que automatiza la generación de una pose odométrica (utilizando la odometría que genera FAST-LIO2 y añadiendo error a modo de ruido gaussiano y/o offset para cada componente). Dado que los mapas se han generado en escenarios horizontales, se estudian únicamente las componentes x e y, así como la rotación yaw (rotación medida en plano xy). Esta odometría se proporciona como estimación inicial para la ejecución de ICP. De esta forma, para cada punto de la pose odométrica se genera una pose estimada aplicando la transformación obtenida por ICP. Para el cálculo de ICP se ha utilizado la función open3d.pipelines.registration.registration_icp[2] de la librería open3d de Python.

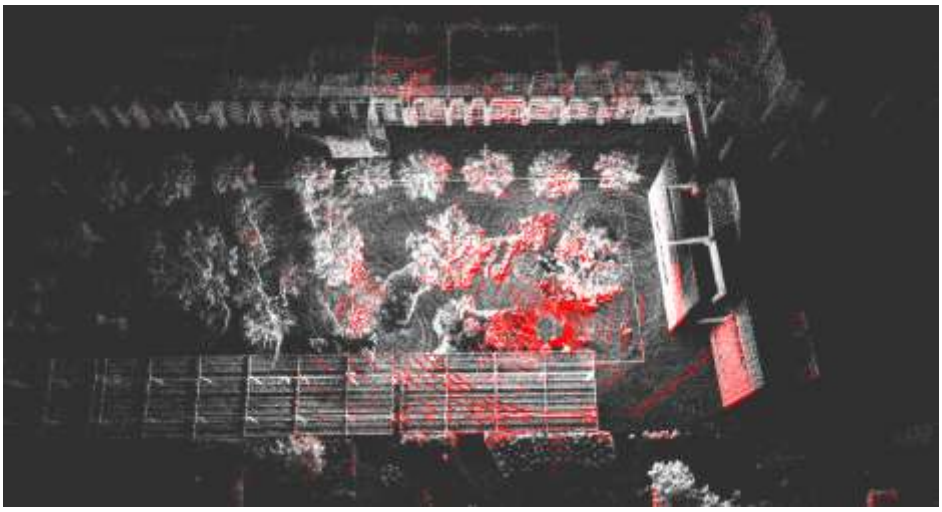


Figura 44: Superposición de la nube de puntos capturada por el sensor LiDAR al Mapa 1 en el instante inicial.

Además de estas funcionalidades, se automatiza la generación de un archivo con información sobre las transformaciones obtenidas, las distintas estimaciones de posición y sus errores para cada instante, con el objetivo de poder realizar comparaciones y analizar la influencia de los distintos parámetros presentes en la implementación. Tras este análisis se puede extraer información sobre la utilidad de algoritmos basados en ICP para la mejora de estimaciones de localización poco precisas.

Para esto, se ha diseñado un código capaz de recorrer todos los mensajes contenidos en cada archivo bag y generar una trayectoria estimada por aplicación de ICP. Es posible suprimir la reproducción del archivo para que el programa funcione de forma online, siendo relevante en esta configuración el tiempo de convergencia del algoritmo ICP, dado que un tiempo elevado puede suponer la pérdida de mensajes intermedios. En la Figura 45 se muestra el diagrama de comunicación generado por rqt para la reproducción automática del bag.

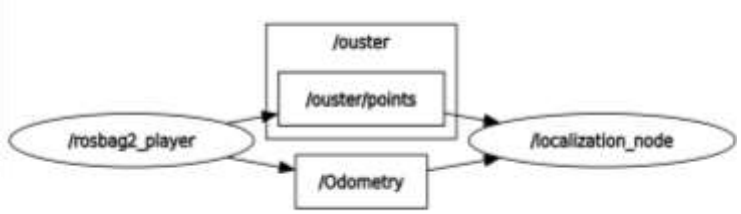


Figura 45: Diagrama de comunicación rqt_graph durante la ejecución del nodo en configuración offline.

El diagrama resultante para la configuración de funcionamiento de forma online con FAST-LIO2 se muestra en la Figura 46.

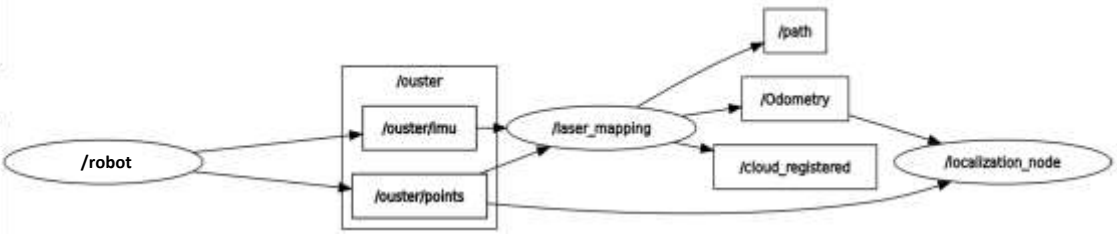


Figura 46: Diagrama de comunicación rqt_graph durante la ejecución del nodo en configuración online junto con FAST-LIO2. *FAST-LIO2 recibe de la reproducción del archivo bag los mensajes enviados por LiDAR e IMU y genera el registro de los puntos e información sobre la odometría. Por último, el nodo de localización por ICP recibe la información de las nubes de puntos capturadas por el LiDAR y de la odometría generada durante el SLAM a modo de Ground Truth.*

Se definen como parámetros los valores de offset aplicados a cada componente de la odometría, los valores sigma y mu como desviación estándar[7] y valor medio del ruido aplicado, el número de iteraciones máximas del algoritmo ICP, el tamaño de voxelización de las nubes de puntos y el umbral del algoritmo ICP para la consideración de coincidencia entre dos puntos. Tras cada ejecución del algoritmo ICP se añade una fila al archivo de datos conformada por las variables indicadas en la Tabla 10.

Ground Truth x	Componentes de la localización consideradas como valores reales.
Ground Truth y	
Pose Odométrica x	Componentes de la estimación odométrica que se proporciona al algoritmo ICP como estimación inicial.
Pose Odométrica y	
Pose Estimada x	Componentes de la estimación resultante de aplicar la transformación calculada por ICP a la Pose Odométrica.
Pose Estimada y	
Traslación ICP x	Componentes de la traslación obtenida tras la ejecución del algoritmo ICP.
Traslación ICP y	
Error Pose Odométrica x	Errores en x e y en la Pose Odométrica respecto al Ground Truth
Error Pose Odométrica y	
Error Pose Estimada x	Errores en x e y en la Pose Estimada por ICP respecto al Ground Truth
Error Pose Estimada y	
ECM Pose Odométrica	Error cuadrático medio en x e y de la Pose Odométrica
ECM Pose Estimada	Error cuadrático medio en x e y de la Pose Estimada por ICP

Tabla 10: Variables almacenadas en el archivo bag tras cada ejecución del algoritmo ICP. *Estos valores se dividen en estimaciones de la posición, traslaciones calculadas por ICP y diferentes errores.*

Se muestran a continuación los resultados obtenidos en diferentes pruebas sobre la estimación de posición del robot móvil mediante la aplicación de ICP a partir de estimaciones iniciales con diferentes tipos de ruido.

[7] Desviación estándar: Medida estadística de la dispersión de una distribución de datos. Esta se obtiene como el promedio de la variación al cuadrado de cada valor con respecto a la media:

$$\sigma = \frac{1}{N}\sum(x_i - \mu)^2$$

5.1. PRUEBAS PARA ERRORES DE TIPO OFFSET

Como primer análisis del desempeño de la estimación de posición por algoritmo ICP se realizan pruebas reproducibles de adición de offset a cada componente de la estimación inicial (x, y) de forma individual con el objetivo de identificar si el ruido afecta por igual a cada componente o si se detecta alguna variación. En la primera prueba, se aplica un offset de 1 metro en x (ECM inicial de 0.5). Para este caso, la evolución del ECM obtenido para cada ejecución se muestra en la Figura 47. Estos resultados muestran mejoras significativas para la mayoría de los cálculos, generando un error cuadrático medio menor en todos los casos. Este comportamiento es similar para la prueba realizada para la componente y.

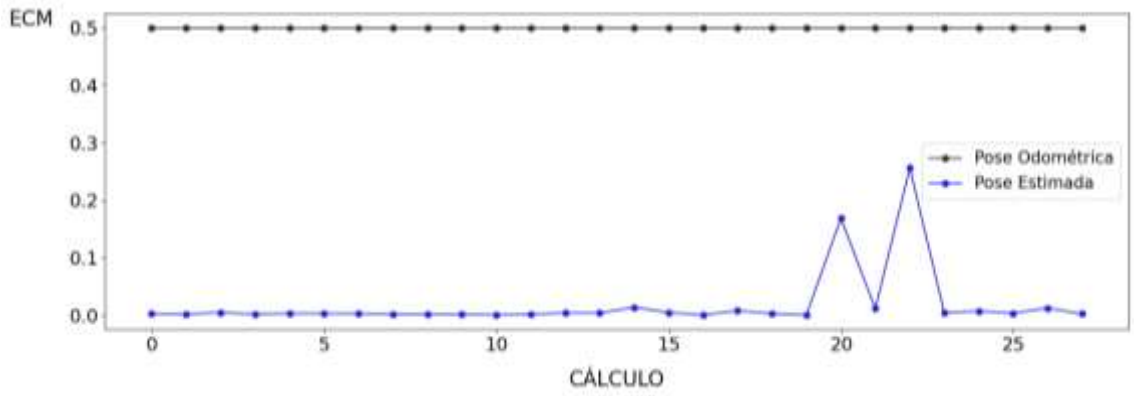


Figura 47: Evolución del error cuadrático medio de la Pose Estimada para una Pose Odométrica con error de offset de 1 metro en x y ECM constante de 0.5.

Siendo cada ejecución de ICP independiente del resto, los errores promedios obtenidos para la aplicación del offset de 1 metro de forma individual en las componentes x e y tras 100 ejecuciones han sido:

- ECM promedio en la pose estimada tras adición de offset en x: 0.01891515
- ECM promedio en la pose estimada tras adición de offset en y: 0.01904391

Tras las pruebas realizadas, se ha llegado a la conclusión de que el error de la estimación inicial afecta de manera similar tras la aplicación a cada componente, x e y, dado que los resultados producidos no presentan diferencias significativas (diferencia de error cuadrático medio menor al 1 %).

Las mejoras producidas en las estimaciones generadas tras la ejecución de ICP han mostrado resultados satisfactorios en las pruebas anteriores, por lo que es conveniente estudiar situaciones con distintos errores en la pose odométrica. En la Figura 48 se muestra la evolución del ECM para una adición de offset de 0.1 metros en la coordenada x de la pose odométrica. Como se puede observar, aunque la estimación por ICP produce mejoras en la mayoría de las ejecuciones, es posible obtener estimaciones con ECM significativamente mayor que en las estimaciones iniciales, por lo que puede resultar perjudicial.

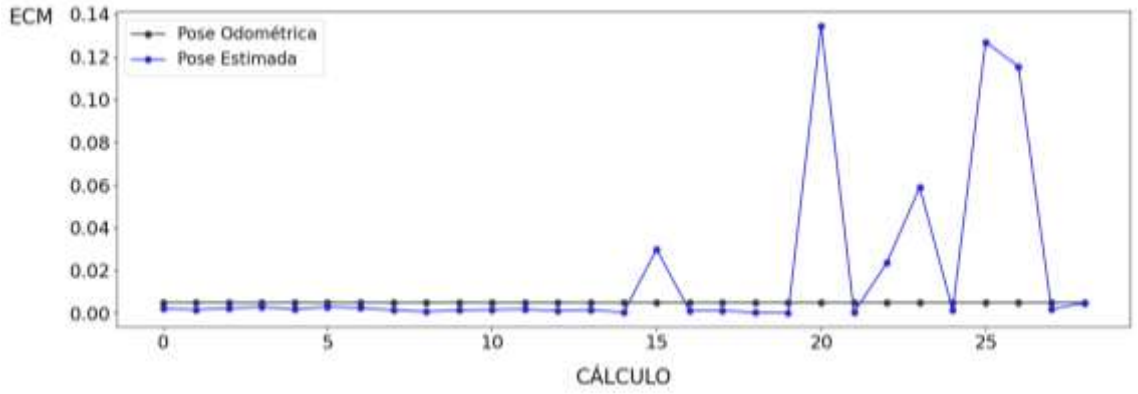


Figura 48: Evolución del error cuadrático medio de la Pose Estimada para una Pose Odométrica con error de offset de 0.1 metros y ECM constante de 0.01.

5.2. PRUEBAS PARA ERRORES DE TIPO RUIDOSO

Con el objetivo de reproducir y analizar situaciones similares a las que se dan en la realidad, se procede a la aplicación de diferentes ruidos generados mediante la variación de los parámetros definitorios de una distribución gaussiana. Este análisis proporciona una idea general sobre el desempeño del algoritmo ICP para distintos valores y escalas de ruido.

5.2.1. Prueba para ruido gaussiano con centro en 0 y desviación estándar igual a 0.5:

Para analizar el desempeño de ICP ante este tipo de ruidos se comienzan las pruebas con un límite de iteraciones relativamente poco restrictivo, 1000. Los resultados obtenidos se muestran en la siguiente figura:

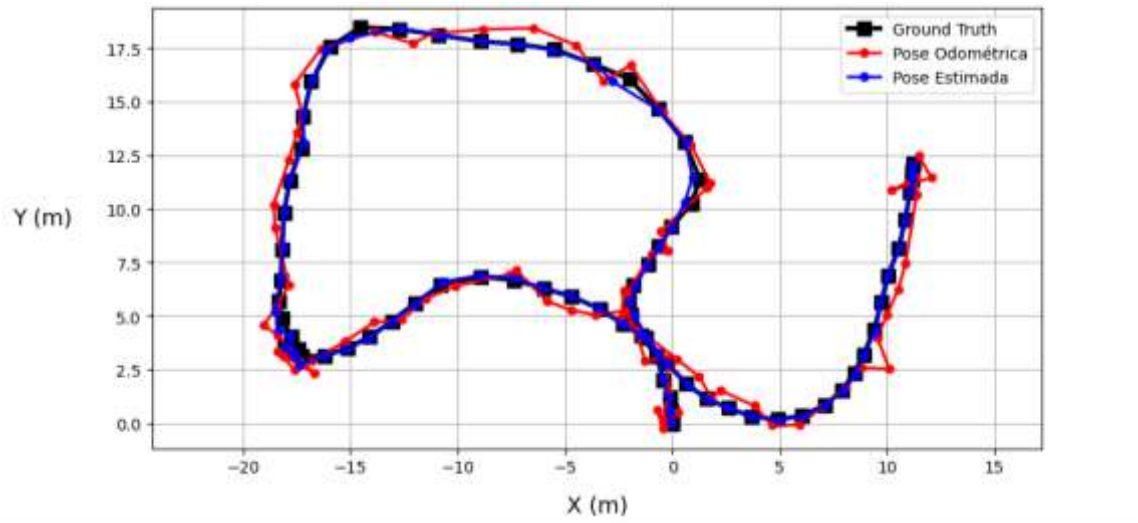


Figura 49: Comparación de trayectorias estimadas y ground truth para pose odométrica con error gaussiano con centro en 0 y sigma 0.5 con límite de 1000 iteraciones para ICP.

Se ha conseguido una mejora en el error cuadrático medio de 0.2387 a 0.0188, reduciendo su valor a un 7.86 % para 75 ejecuciones de ICP.

5.2.2. Prueba para ruido gaussiano con centro en 0 y desviación estándar igual a 1:

Las pruebas iniciales realizadas con límite de iteraciones elevado, 1000, han generado los resultados que se muestran en la Figura 50, en la que se aprecia cómo la trayectoria estimada presenta una alta concordancia con el ground truth, mientras que la trayectoria odométrica muestra varios puntos notablemente alejados.

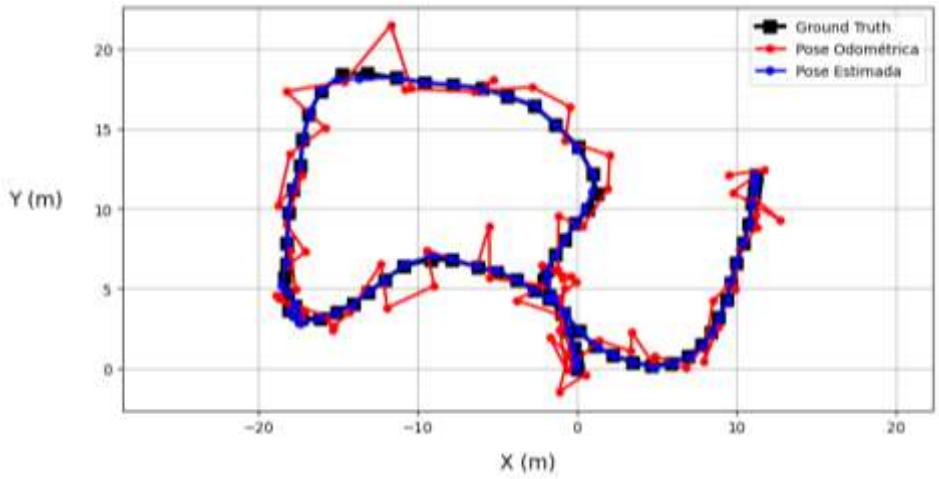


Figura 50: Comparación de trayectorias estimadas y ground truth para pose odométrica con error gaussiano con centro en 0 y sigma 1 con límite de 1000 iteraciones para ICP. *Se aprecia una trayectoria estimada por ICP con errores sustancialmente menores a los de la trayectoria odométrica.*

La evolución del ECM para la pose odométrica y la pose estimada se muestra en la Figura 51. Se puede apreciar cómo el ECM de la pose estimada es significativamente menor, por lo que esta configuración produce mejoras notables en las estimaciones.

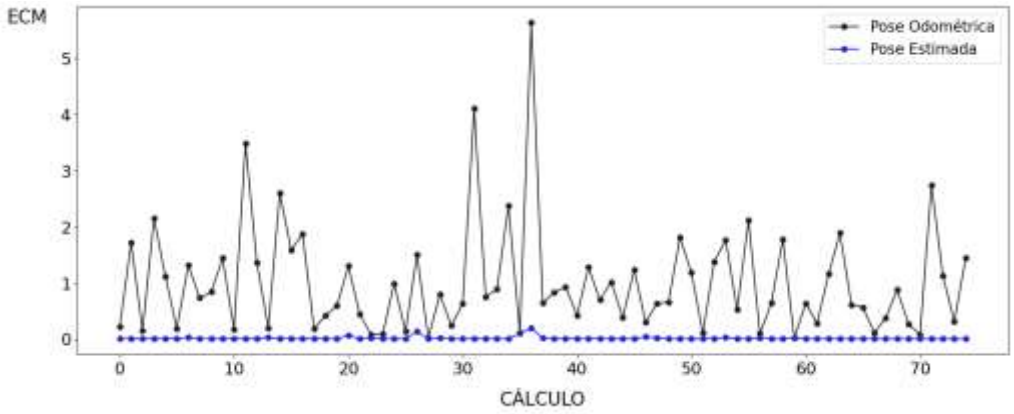


Figura 51: Evolución del error cuadrático medio de la Pose Odométrica con error gaussiano con centro en 0 y sigma 1 y la Pose Estimada con limitación de ICP a 1000 iteraciones.

Como se puede observar, los resultados de estimación de posición mediante utilización de ICP resultan considerablemente cercanos a los valores del ground truth. Por tanto, se analiza ahora el desempeño para el mismo tipo de ruido añadido en la pose odométrica para limitaciones más restrictivas del número de iteraciones máximas. Para este contexto, se espera un aumento del ECM promedio en la pose estimada conforme disminuye el valor del límite de iteraciones.

5.2.2.1. Algoritmo ICP limitado a 10 iteraciones

Mediante la limitación de iteraciones de ICP a 10, se reduce el espacio de estudio para el registro de las nubes, lo que implica que los errores elevados en la estimación inicial pueden no solventarse por completo. Las trayectorias obtenidas durante esta prueba se muestran en la Figura 52.

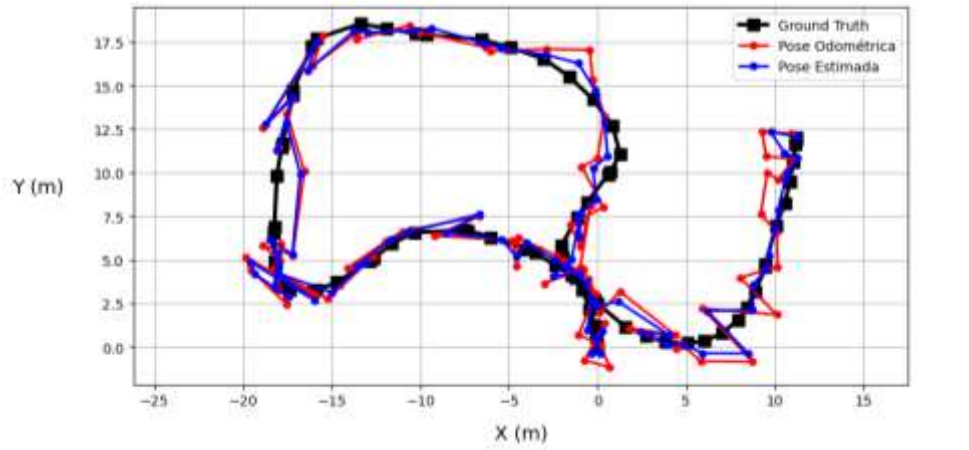


Figura 52: Comparación de trayectorias estimadas y ground truth para pose odométrica con error gaussiano con centro en 0 y sigma 1 con límite de 10 iteraciones para ICP. *Para esta configuración los resultados de la estimación mejoran la pose odométrica, aunque siguen mostrando errores considerables para los casos de estimaciones iniciales con errores relativamente grandes.*

La evolución del ECM para las diferentes ejecuciones de ICP en esta configuración se muestra en la Figura 52.

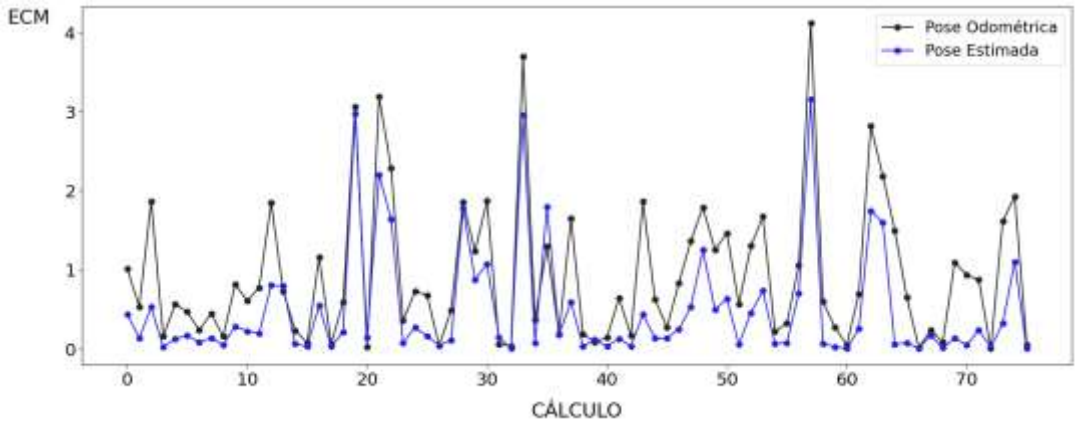


Figura 53: Evolución del error cuadrático medio de la Pose Odométrica con error gaussiano con centro en 0 y sigma 1 y la Pose Estimada con limitación de ICP a 10 iteraciones.

Para esta configuración, los resultados muestran una corrección del error aceptable para algunos casos y un error considerablemente elevado en otros, por lo que se realiza a continuación un estudio con un limite de iteraciones menor, disminuyendo aún más el espacio de estudio de ICP.

5.2.2.2. Algoritmo ICP limitado a 5 iteraciones

Por último, para un número de iteraciones muy restrictivo como es 5, se espera que, aunque ICP sea capaz de encontrar transformaciones que reduzcan el error, el poco margen de estudio no permita reducir sustancialmente el error en la mayoría de los casos. En la Figura 54, se muestran las trayectorias obtenidas para esta configuración junto al ground truth.

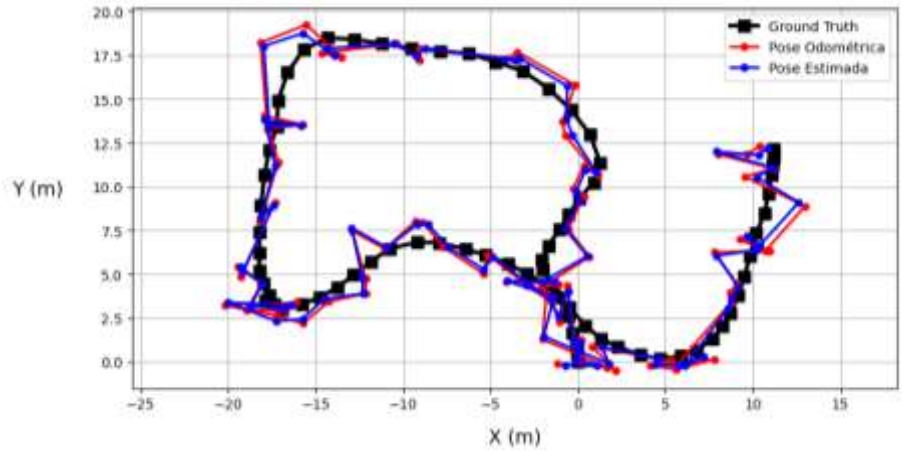


Figura 54: Comparación de trayectorias estimadas y ground truth para pose odométrica con error gaussiano con centro en 0 y sigma 1 con límite de 5 iteraciones para ICP. *Para esta configuración, los resultados de la estimación mejoran levemente la trayectoria de la pose odométrica y continúan produciendo errores considerables en la mayoría de los casos.*

Los resultados que se generan en esta configuración producen estimaciones con mejoras leves dado el acotado espacio de estudio de ICP. Es por esto que el ECM evoluciona de forma similar para ambas trayectorias (ver Figura 55).

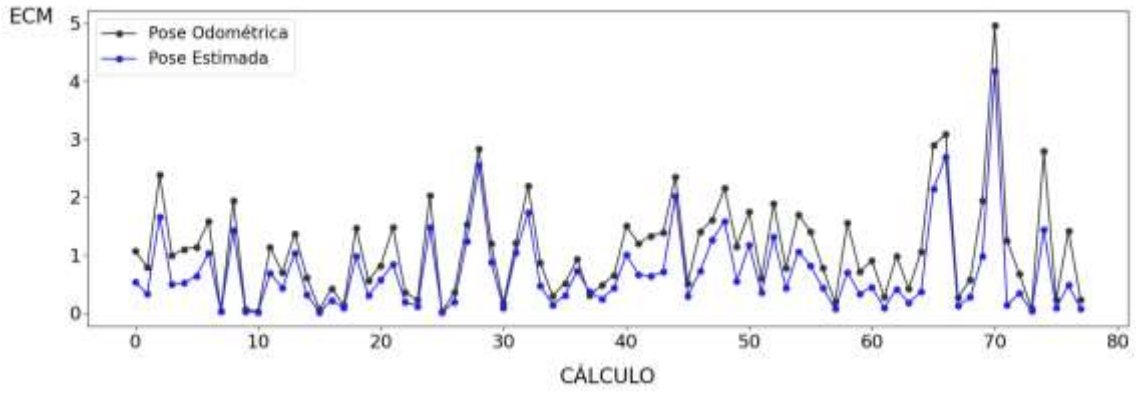


Figura 55: Evolución del error cuadrático medio de la Pose Odométrica con error gaussiano con centro en 0 y sigma 1 y la Pose Estimada con limitación de ICP a 5 iteraciones.

Como se puede apreciar en los resultados mostrados (Figura 54 y Figura 55), los errores de la pose estimada siguen siendo relativamente elevados, por lo que el método de localización resultaría impreciso en esta configuración. De esta forma, se consigue una idea general del desempeño de ICP para estimaciones con diferentes escalas y tipos de ruido para un amplio rango de iteraciones máximas permitidas.

5.3. DESEMPEÑO DE ICP EN FUNCIÓN DEL LÍMITE DE ITERACIONES

Una vez analizados los resultados producidos por ICP en función del ruido de la estimación inicial, resulta interesante analizar la influencia de otros parámetros como el límite de iteraciones de forma independiente. El algoritmo realiza un proceso iterativo, de forma que, si los resultados obtenidos tras aplicar la transformación actual no cumplen con los umbrales de error predefinidos, ICP vuelve a calcular una nueva transformación de manera recursiva. Es por esto por lo que se define un máximo de iteraciones con el objetivo de limitar el tiempo de procesamiento y así evitar posibles bloqueos dentro de la ejecución de ICP. En los casos en los que el resultado se determina por el alcance del número de iteraciones máximas y no por convergencia, se obtienen de forma general errores mayores.

Es evidente que un mayor error en la estimación inicial que se proporciona al algoritmo supone un incremento en el número de iteraciones promedio hasta la convergencia y un mayor tiempo de procesamiento en comparación a estimaciones iniciales más precisas, dado que ICP comienza por estudiar transformaciones mínimas y va incrementando tanto translación como rotación de forma gradual. Por tanto, se lleva a cabo un análisis del desempeño de ICP ante la variación del número máximo de iteraciones y una comparación de los resultados.

Tras la realización de la prueba para el recorrido del archivo rosbag 1 con un error gaussiano de centro en 0 y desviación estándar de 1 para diferentes valores de iteraciones máximas de ICP, se han obtenido los resultados que se muestran en la Figura 56. En estos resultados se puede apreciar la notable disminución del error cuadrático medio conforme aumenta el número de iteraciones máximas. A partir de 50 iteraciones los resultados no muestran diferencias sustanciales, por lo que se deduce que para límites de iteraciones mayores el algoritmo alcanza la convergencia a la misma posición en la mayoría de las ejecuciones.

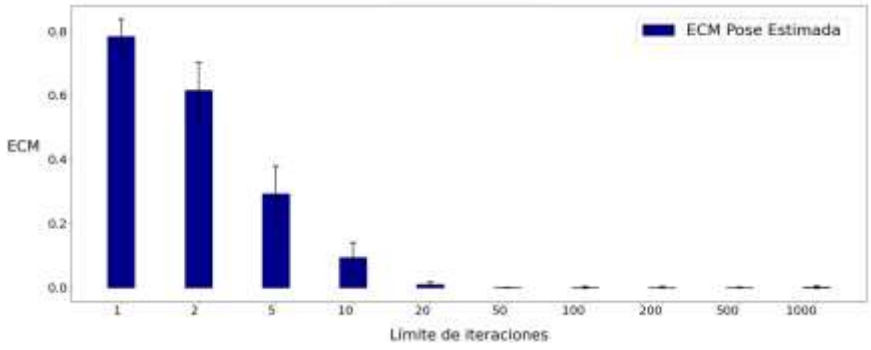


Figura 56: Variación del error cuadrático medio de la estimación de posición por ICP según el límite de iteraciones con ECM de estimación inicial de 1. *Esta prueba se ha realizado con un error de offset de 1 metro en x e y para 50 ejecuciones de cada configuración.*

5.4. DESEMPEÑO DEL ICP EN FUNCIÓN DEL TAMAÑO DE VOXELIZACIÓN

Con el objetivo de comprender la influencia del tamaño de voxelización en los resultados de la estimación de posición aplicando ICP, se realizan diferentes pruebas variando este parámetro. En primer lugar, se lleva a cabo un estudio para tamaños de voxelización de diferentes escalas con el objetivo de identificar la tendencia del error generado por la estimación. Los resultados de esta prueba se muestran en la Figura 57.

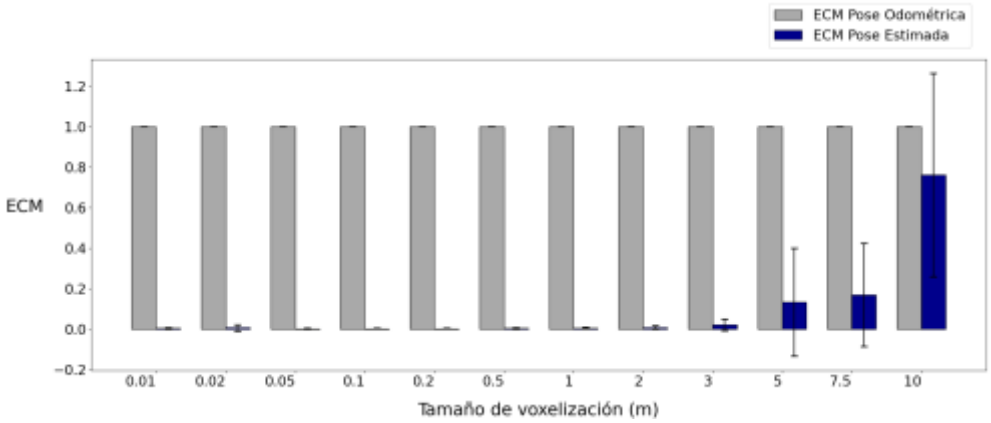


Figura 57: Variación del error cuadrático medio de la Pose Estimada en función del tamaño de voxelización con diferentes escalas. *Se observa una mejora en todas las configuraciones, aunque se puede apreciar una tendencia creciente del ECM que hace que ambos errores sean similares para el tamaño de voxelización de 10 metros.*

Dada la tendencia significativamente creciente que muestran los resultados obtenidos a partir de tamaños de voxelización del orden del metro, se realiza una prueba para valores de 1 a 15 metros con el objetivo de localizar el cruce de las líneas de tendencia de ambos errores y poder definir el rango en el que el uso de ICP resulta beneficioso y aquel en el que los resultados obtenidos empeoran con respecto a las predicciones iniciales. Los resultados de esta prueba se muestran en la Figura 58.

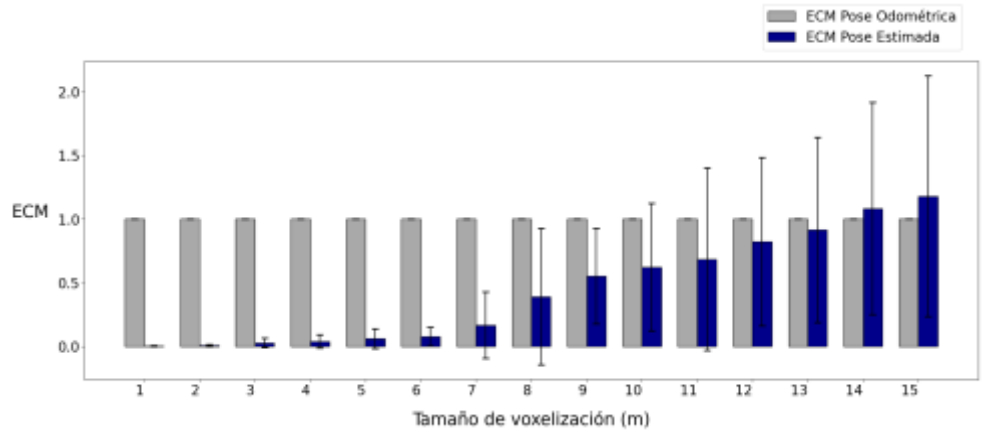


Figura 58: Variación del error cuadrático medio de la Pose Estimada en función del tamaño de voxelización para valores de 1 a 15 metros. *Prueba realizada para 50 ejecuciones de ICP en cada configuración. Se observa cómo el ECM promedio de la Pose Estimada supera al de la Pose Odométrica a partir de un tamaño de voxelización de 14 metros, por lo que esta configuración es desaconsejable.*

Dado que los resultados generados por ICP muestran una fuerte dependencia de la voxelización realizada y que este factor produce una variación en el tiempo de cálculo, es conveniente analizar su influencia para estudiar la viabilidad de la utilización de esta técnica de forma online. Un tiempo de cálculo demasiado elevado puede resultar incompatible con esta práctica en determinados escenarios. Puesto que la variación del número de iteraciones límite no supone una variación significativa del tiempo de convergencia de ICP, será el tamaño de voxelización de las nubes de puntos el parámetro clave para este estudio temporal.

En primer lugar, se realiza una prueba para tamaños de voxelización de diferentes órdenes y se grafican los resultados de ECM junto con el tiempo de convergencia promedio. Esta gráfica permite el análisis de la escala de voxelización que genera resultados interesantes en relación tanto al ECM como al tiempo de cálculo (Figura 59).

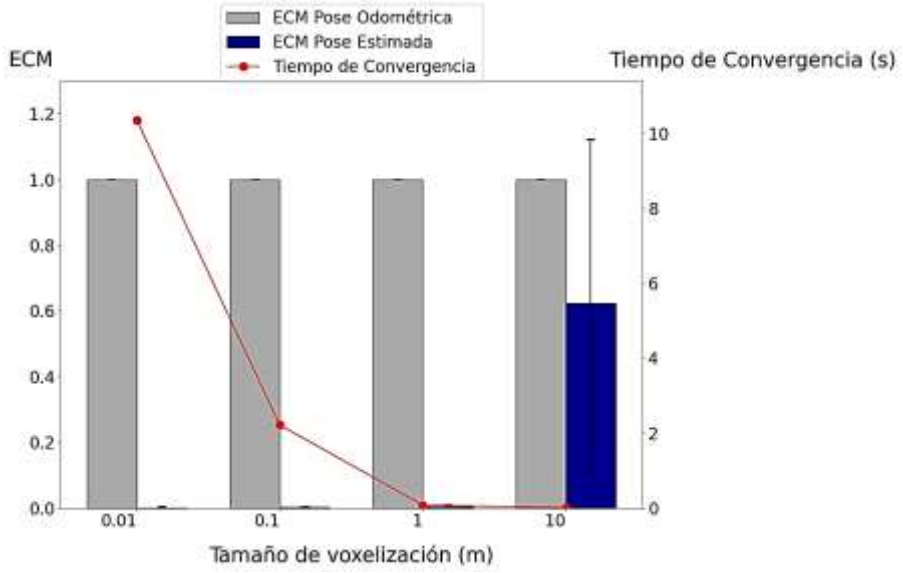


Figura 59: Evolución del error cuadrático medio y del tiempo de convergencia para diferentes escalas del tamaño de voxelización. *Se aprecia una tendencia creciente el ECM y decreciente del tiempo de convergencia a medida que aumenta el tamaño de voxelización.*

Como se puede observar, los resultados de ECM presentan una gran mejora para todos los casos excepto para la voxelización de 10 metros. En cuanto al tiempo de convergencia, el voxelizado de 1 metro requiere de un tiempo de cálculo notablemente menor a 1 segundo, por lo que este punto resulta óptimo para las dos variables analizadas. Una vez se conoce la escala adecuada para la voxelización, se realiza un análisis específico con el objetivo de comprender el comportamiento de ambas variables de una forma más precisa. Esto se muestra en la Figura 60.

Como se puede apreciar, un tamaño de voxelización de 1 metro genera resultados de gran precisión con un tiempo de convergencia de 0.0774 segundos, lo que resulta factible para la mayoría de las aplicaciones online. Este tiempo se puede decrementar si se requiere, alcanzando de igual forma mejoras sustanciales para un tamaño de voxelización de 6 metros, con un tiempo de convergencia medio de 0.0225 segundos.

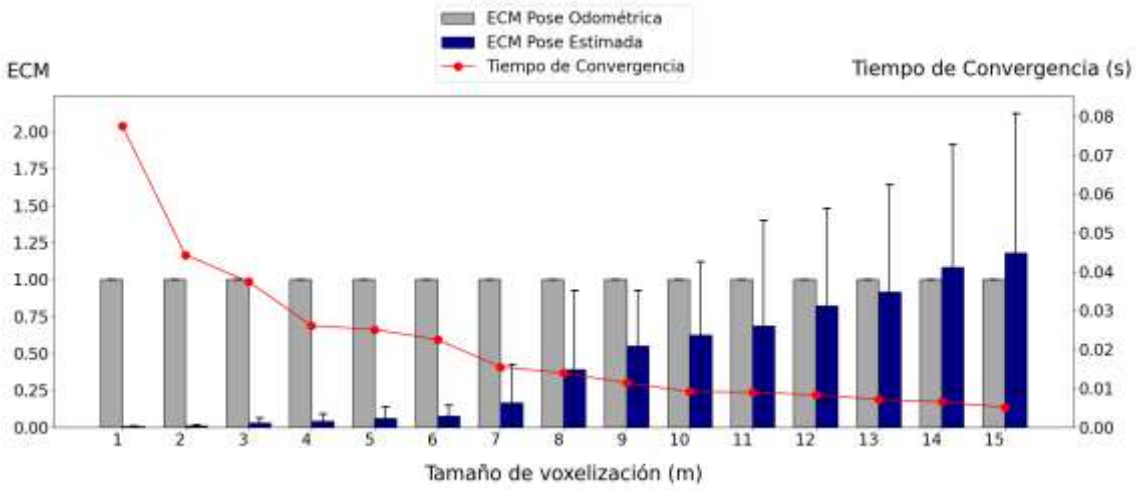


Figura 60: Evolución del error cuadrático medio y del tiempo de convergencia para tamaño de voxelización de 1 a 15 metros.

Cabe destacar que a este tiempo de convergencia del algoritmo ICP se debe sumar el tiempo de voxelización de las nubes de puntos obtenidas por el LiDAR (la voxelización del mapa se realiza una única vez, antes o al principio de la ejecución del código), el cual es del orden de los 0.02 segundos, por lo que el tiempo total de la configuración de 1000 iteraciones máximas y voxelización de 1 metro es del orden de 0.1 segundos, lo que lo hace compatible con su uso online.

6. CONCLUSIONES Y LÍNEAS FUTURAS

Durante el desarrollo de este proyecto se ha analizado la situación actual de la especialidad y se ha conseguido adquirir una idea general sobre la creación, gestión y utilización de conjuntos de datos de puntos en el espacio, y se ha alcanzado el objetivo de creación de mapas densos de puntos con gran precisión.

Además, se ha podido verificar la gran utilidad que conlleva el uso de mapas 3D en aplicaciones de localización en robótica móvil y su eficacia en mejora de estimaciones iniciales erróneas, subrayando así la importancia de la investigación en técnicas de registro punto a punto en aplicaciones de robótica. Los resultados obtenidos han demostrado un alto nivel de precisión y mejoras sustanciales ante errores y ruidos. Por tanto, resulta evidente la gran utilidad en corrección de estimaciones de pose en aplicaciones offline.

En cuanto a la utilización en aplicaciones online es necesario analizar las características de los requerimientos de funcionamiento para cada caso, dado que el tiempo de cálculo que necesita el algoritmo puede ser excesivo en tareas en las que la estimación de la posición debe utilizarse de forma inmediata. Por tanto, estas implementaciones deben estudiarse en base al número de iteraciones máximas esperadas y al tamaño de voxelización de las nubes en comparación al tiempo disponible hasta la necesidad de utilización de los resultados.

Tras la finalización de este proyecto, se abren múltiples líneas de investigación interesantes pertenecientes tanto al campo de la localización en la navegación autónoma de la robótica móvil, como al campo de la generación de mapas densos de puntos 3D.

- Mejoras en la construcción de los mapas: Adición de información sobre color, textura u otras características de posible interés.
- Construcción de los mapas de forma online.
- Integración para el uso los mapas en navegación autónoma
- Análisis de la implementación de métodos basado en ICP de forma online.
- Análisis del desempeño de la localización antes escenarios dinámicos.

7. REFERENCIAS

[1] James Teow. (2018, February 14). Understanding Kalman filters with Python. Medium. https://medium.com/@jaems33/understanding-kalman-filters-with-python-2310e87b8f48

[2] Open3D. (2023). Documentation for open3d.pipelines.registration.registration_icp. Recuperado el 9 de junio de 2024. https://www.open3d.org/docs/latest/python_api/open3d.pipelines.registration.registration_icp.html

[3] Xu, W., Cai, Y., He, D., Lin, J., & Zhang, F. (2021). FAST-LIO2: Fast Direct LiDAR-inertial Odometry. *arXiv*. https://arxiv.org/abs/2107.06829v1

[4] Xu, W., & Zhang, F. (2021). FAST-LIO: A fast, robust LiDAR-inertial odometry package by tightly-coupled iterated Kalman filter. *arXiv preprint*. https://arxiv.org/abs/2010.08196v3

[5] ROS.org, Open Robotics. (2021). ROS (Robot Operating System). Recuperado el 10 de junio de 2024. https://www.ros.org/

[6] Lee, D., Jung, M., Yang, W., & Kim, A. (2023). LiDAR Odometry Survey: Recent Advancements and Remaining Challenges. *arXiv preprint arXiv:2312.17487*. https://arxiv.org/abs/2312.17487

[7] Diab, A., Kashef, R., & Shaker, A. (2022). Deep Learning for LiDAR Point Clouds in Autonomous Driving: A Review. *arXiv*, 2005.09830. https://arxiv.org/abs/2005.09830

[8] Xu, X., & Tian, Y. (2023). A review of multi-sensor fusion SLAM systems based on 3D LiDAR. *Remote Sensing*, 15(13), 3314. https://doi.org/10.3390/rs15133314

[9] hku-mars. (2024). *ikd-Tree: Incremental k-d Tree for robotic applications*. GitHub. Recuperado el 15 de abril de 2024. https://github.com/hku-mars/ikd-Tree

[10] Kudan. (2021). 3D LiDAR SLAM: The basics. Kudan. Recuperado el 16 de junio de 2024. https://www.kudan.io/blog/3d-lidar-slam-the-basics/

[11] Denayer, M., De Winter, J., Bernardes, E., Vanderborght, B., & Verstraten, T. (2024). Comparison of point cloud registration techniques on scanned physical objects. *Sensors, 24*(7), 2142. https://doi.org/10.3390/s24072142

8. ANEXOS

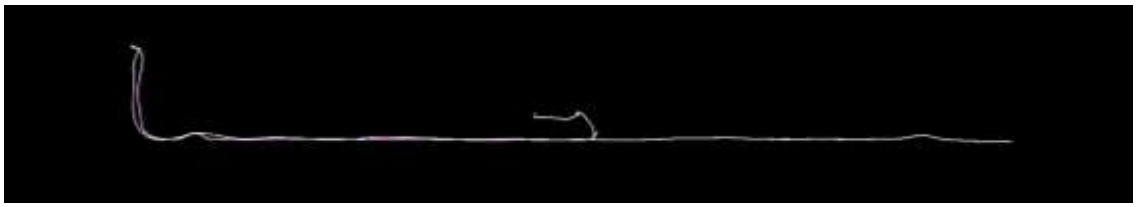
ANEXO 1: MAPA 3 - PASILLO INTERIOR

Este mapa se ha realizado mediante el recorrido de un pasillo interior del mismo edificio como prueba para la generación de mapas en interiores. El proceso de generación se puede ver en el siguiente enlace: GENERACIÓN Y VISUALIZACIÓN DEL MAPA 3

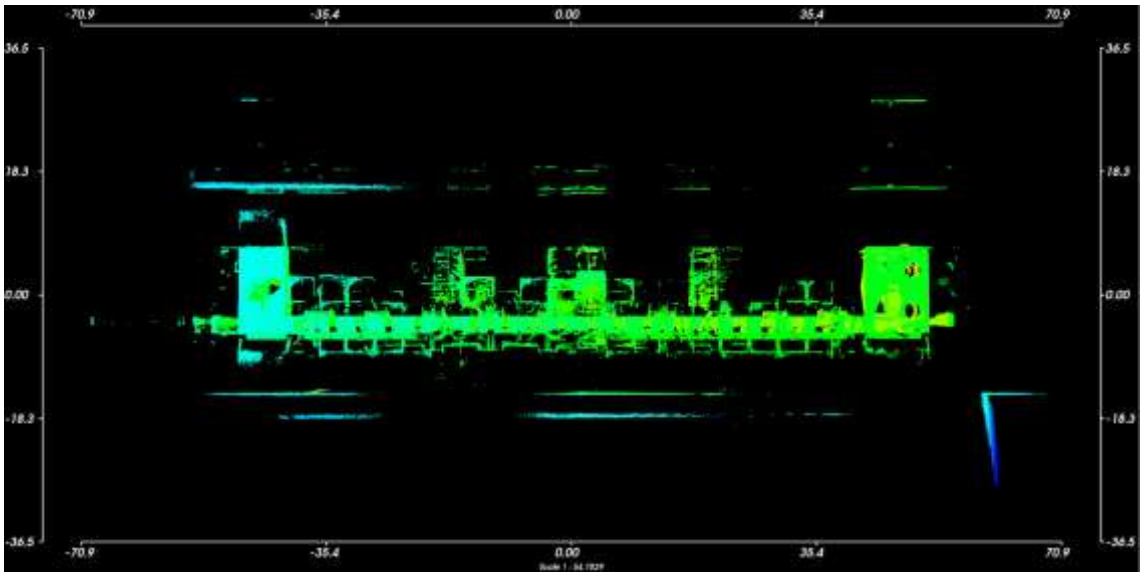
En la siguiente tabla se muestran las características principales de este mapa:

Característica	Valor
Nubes fusionadas	2817
Número de puntos procesados	92307465
Número de puntos del mapa inicial	11563887
Número de puntos tras voxelizar	1775143
Tamaño del archivo inicial	91516 KB
Tamaño del archivo tras voxelizar	40812 KB

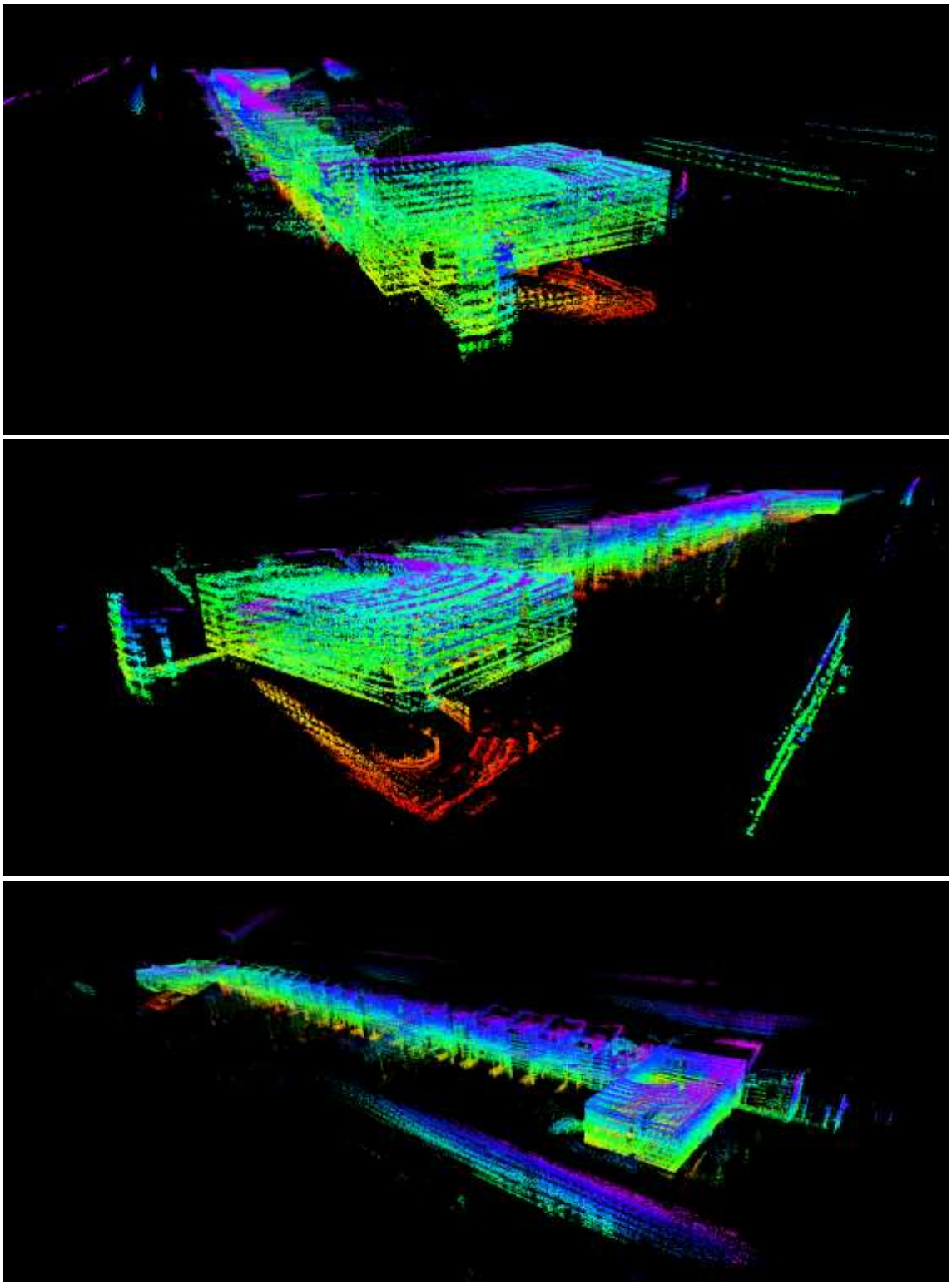
A continuación, se muestran la trayectoria seguida durante el mapeo, las medidas del mapa generado e imágenes descriptivas del mapa.



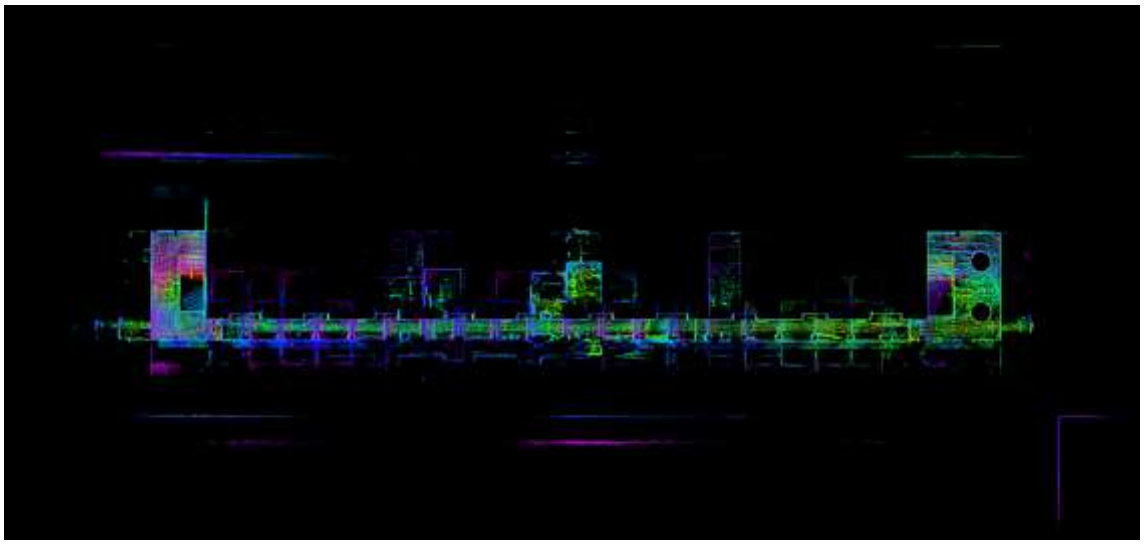
Trayectoria recorrida durante el mapeo.



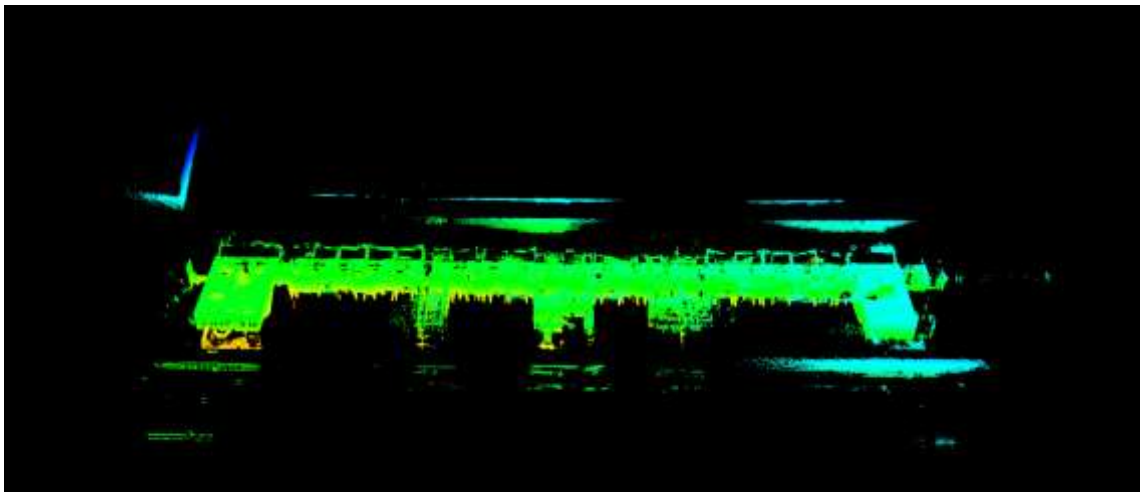
Vista cenital y dimensiones en metros del Mapa 3.



Vistas con diferentes perspectivas.



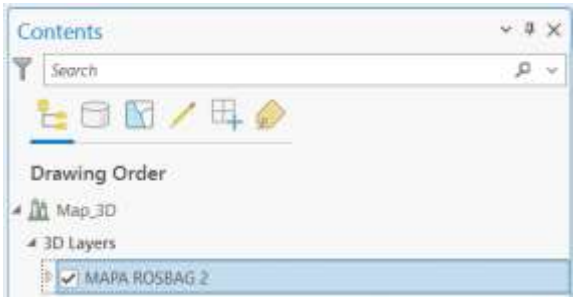
Vista cenital.



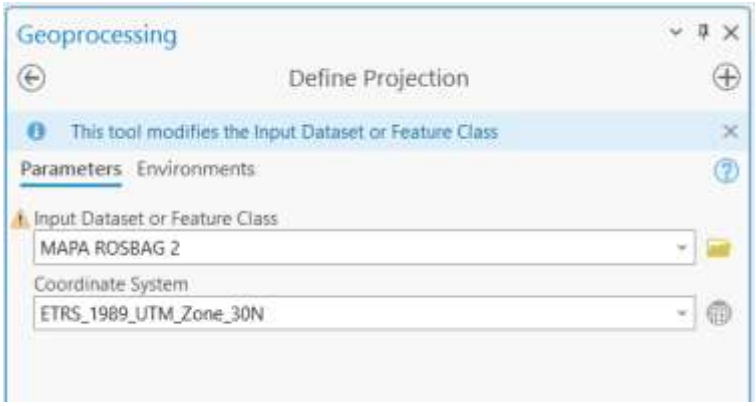
Vista oblicua.

ANEXO 2: INTEGRACIÓN DE NUBES DE PUNTOS GEORREFERENCIADAS EN ARCGIS

A continuación, se muestra el proceso por el cual se georreferencian y se visualizan los mapas de nubes de puntos. En primer lugar, se añade el archivo del mapa con extensión *las/laz* (archivos de datos LiDAR) al proyecto. Este aparece dentro de las capas 3D.



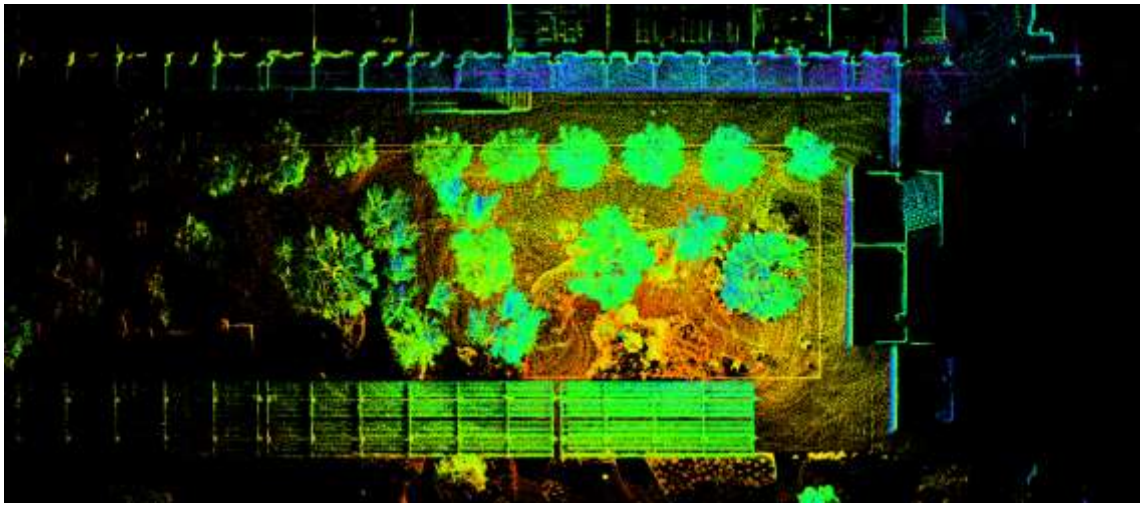
En este momento, el mapa no tiene definido el sistema de referencia, por lo que es necesario aplicar una proyección. Para esto se utiliza la herramienta Definir Proyección o Define Projection, en la que podremos seleccionar la capa del mapa y definir el sistema de coordenadas oportuno.



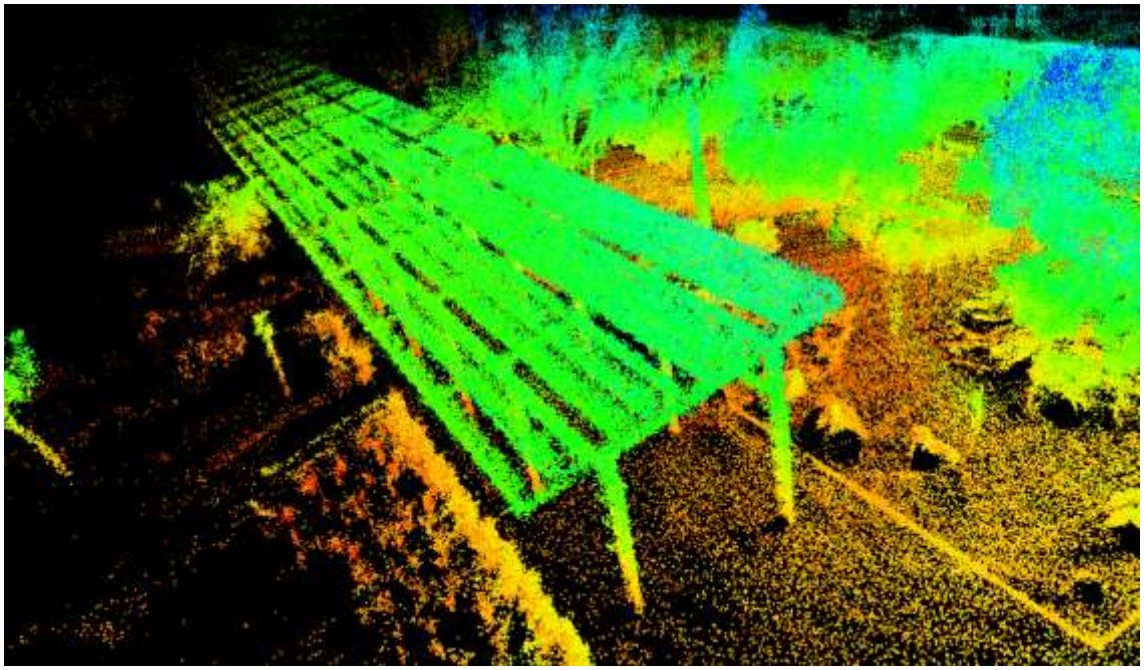
De esta forma, el mapa se sitúa en la ubicación correcta en el planeta, representando así información geoespacial. De esta forma es posible la comparación con otros datos geográficos.

ANEXO 3: GALERÍA DE IMÁGENES DE DETALLE DE LOS MAPAS GENERADOS

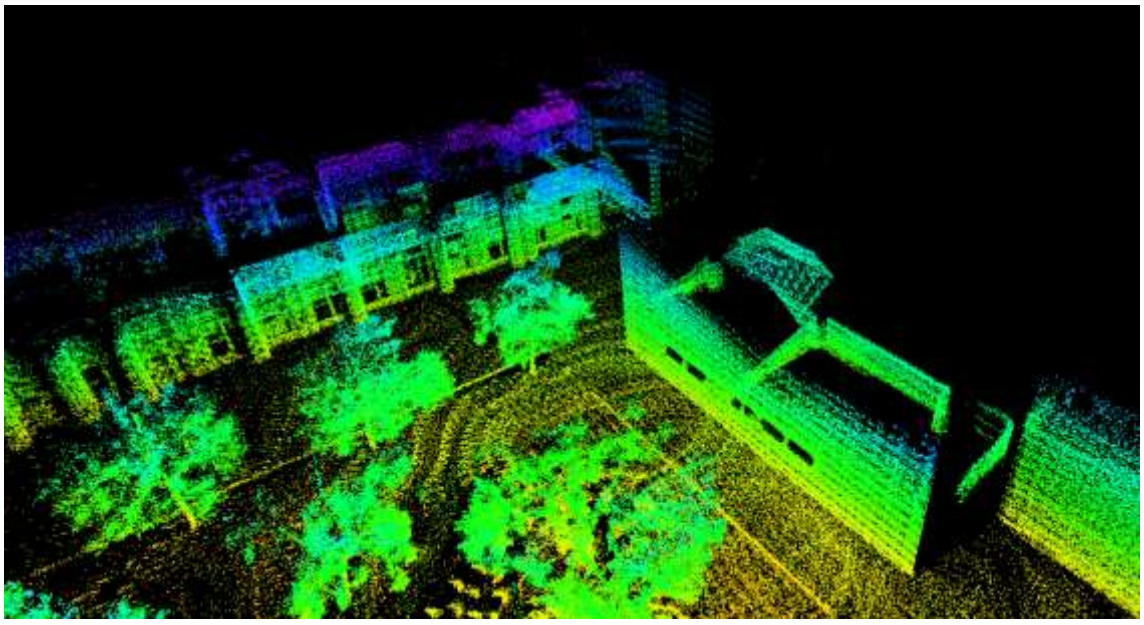
MAPA 1:



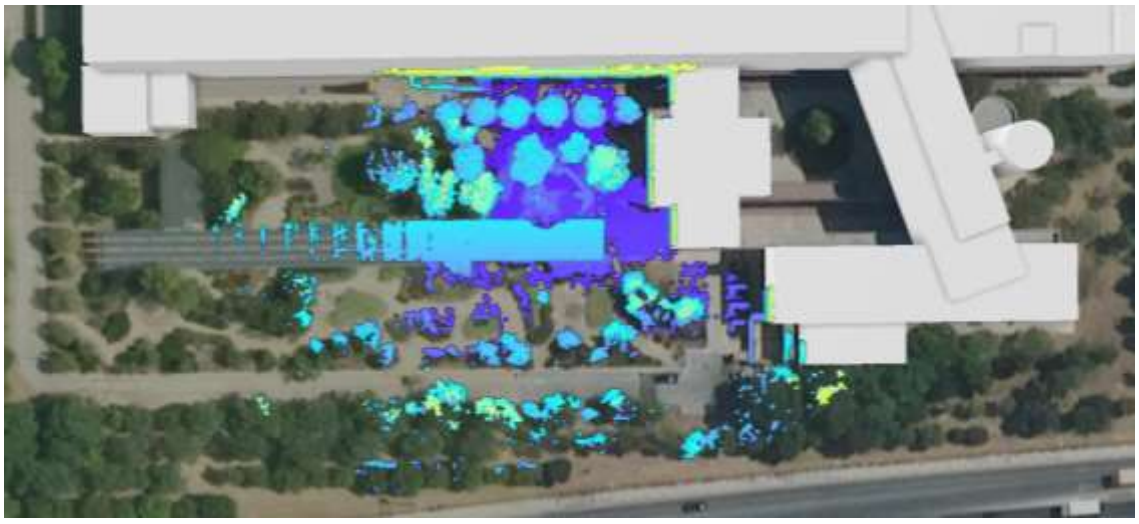
Vista cenital de la zona del patio.

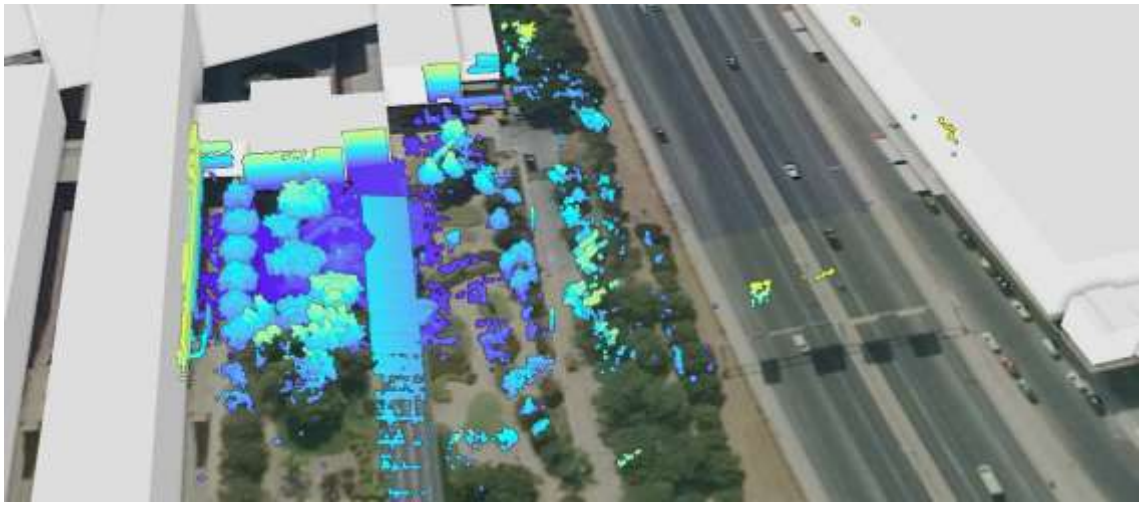


Detalle de estructura con placas solares.



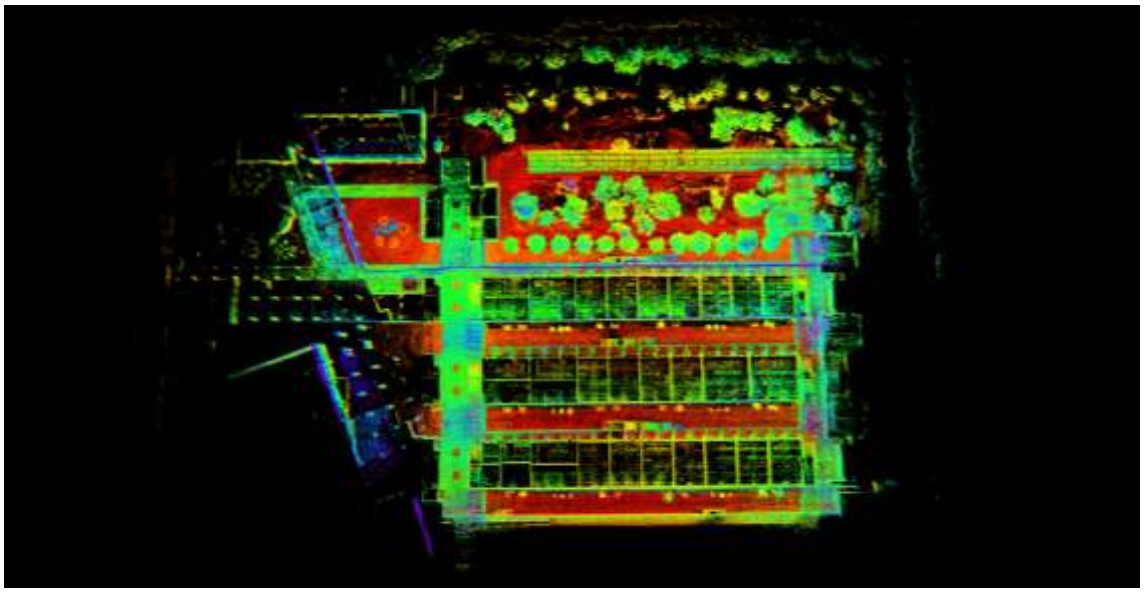
Detalle de paredes interiores capturadas a través de ventanales.



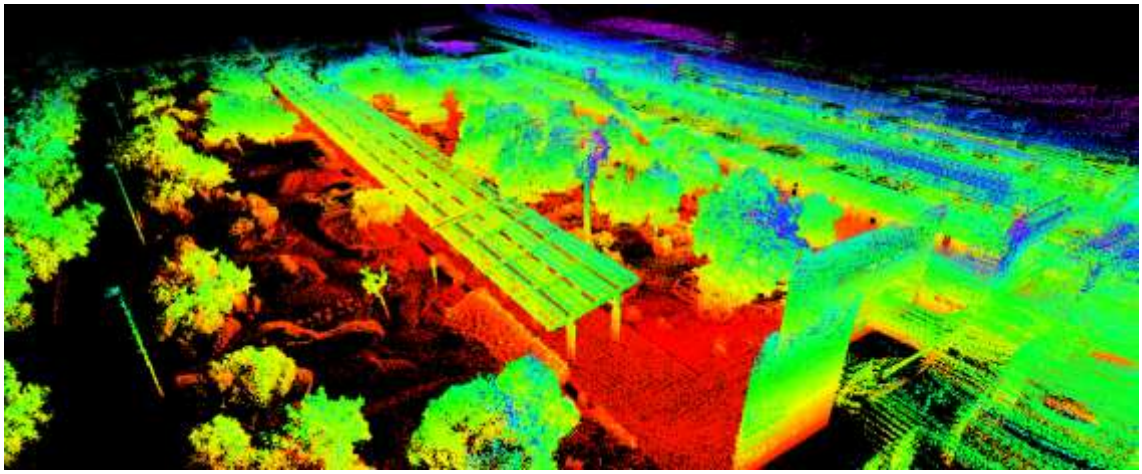


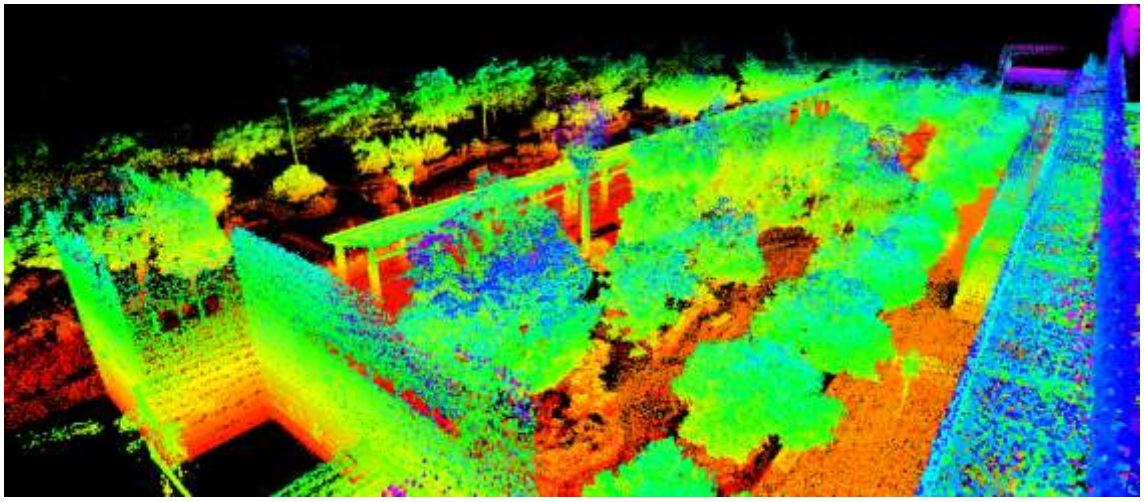
Vistas junto al mapa de imágenes.

MAPA 2:

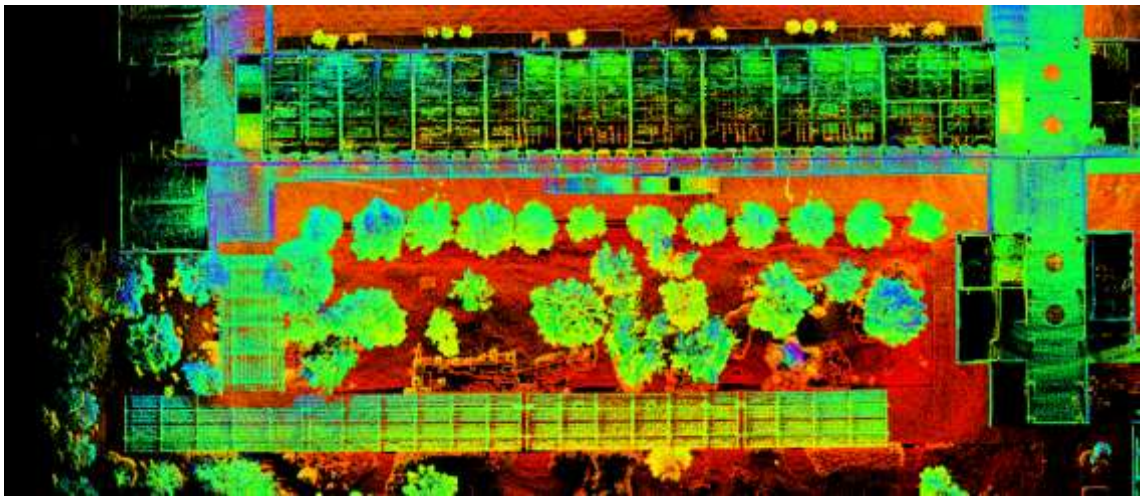


Vista cenital.

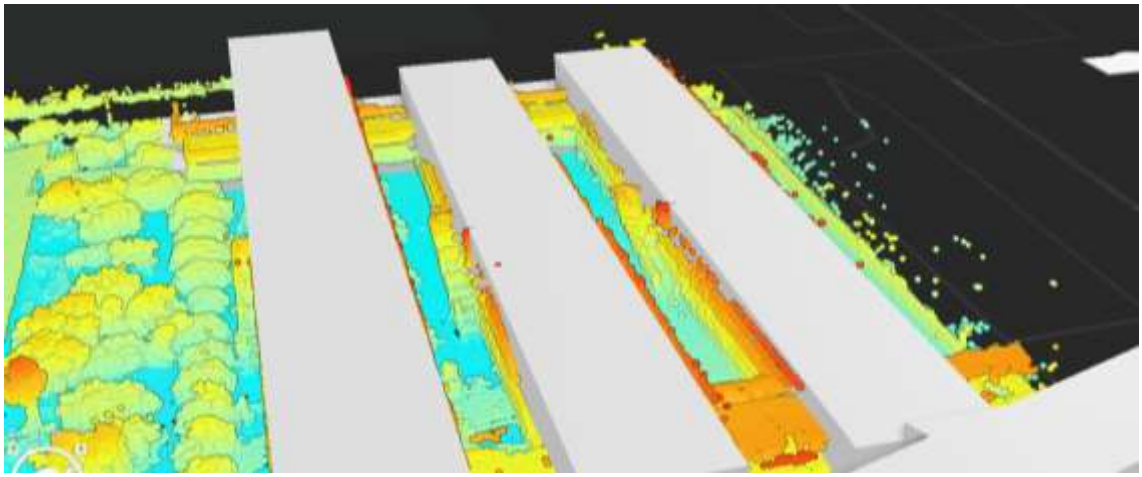




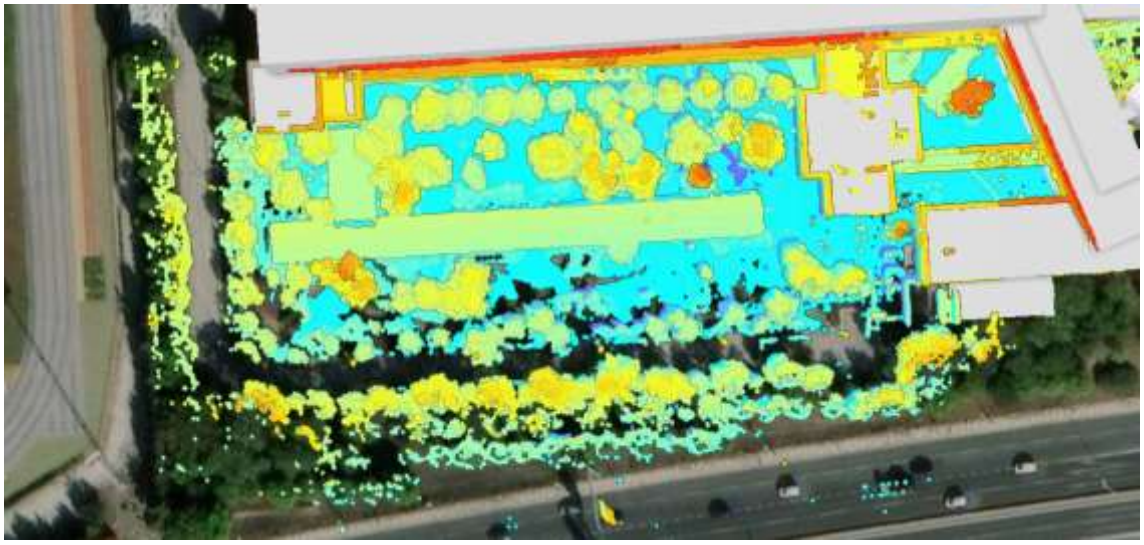
Vistas del patio exterior.



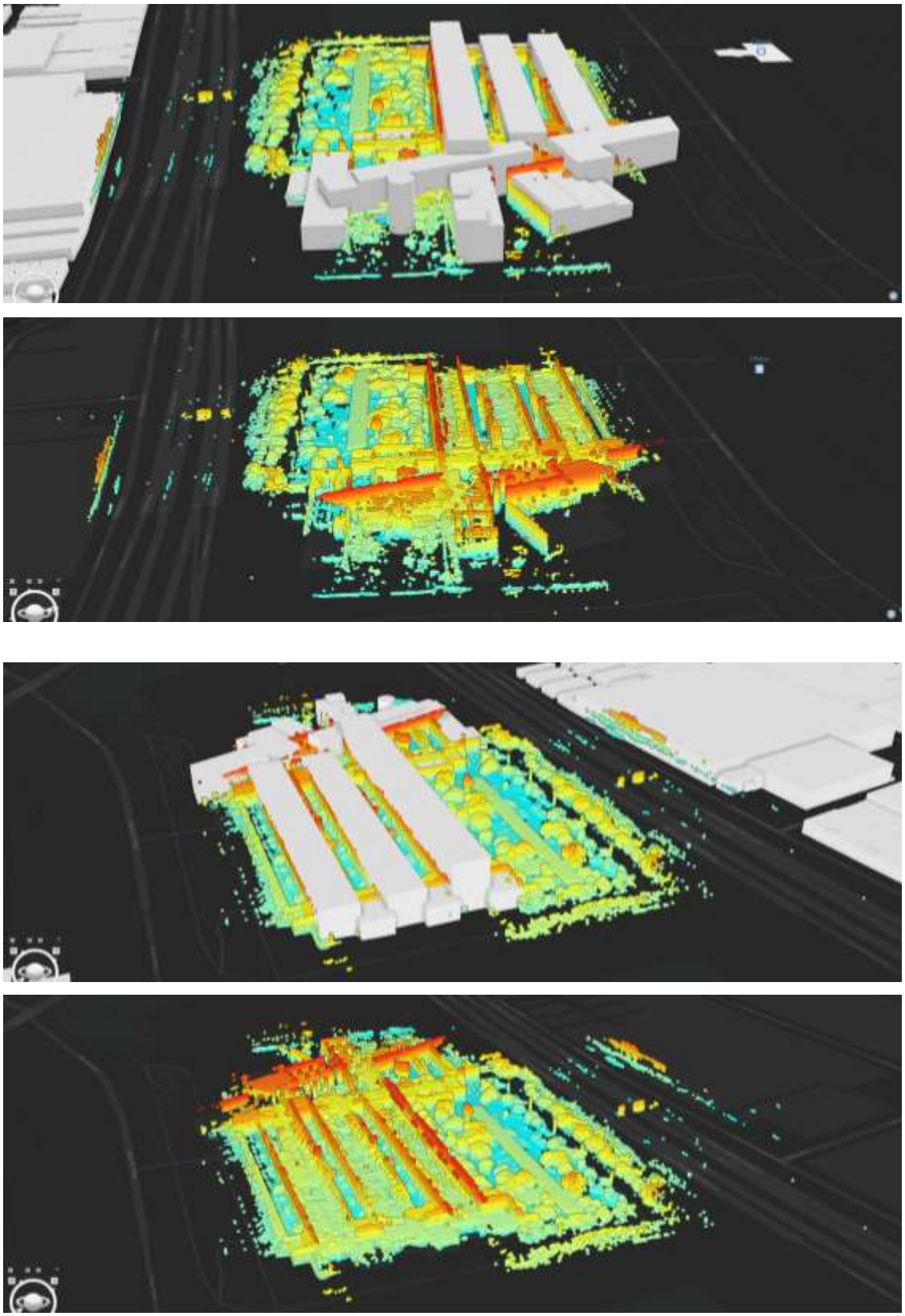
Vista cenital del patio exterior y el tercer bloque.



Detalle de las escaleras exteriores.



Detalle de la coincidencia con los árboles.



Comparativas de la nube de puntos con superposición del modelo 3D del edificio y sin superposición.

ANEXO 4: VÍDEOS Y ARCHIVOS DE INTERÉS

A continuación, se adjuntan los enlaces a los diferentes vídeos y archivos de interés generados durante el desarrollo del proyecto.

Enlaces al repositorio con los vídeos del proyecto: TFG - YouTube

- Mapa 1: GENERACIÓN Y VISUALIZACIÓN DEL MAPA 1 (youtube.com)
- Mapa 2: GENERACIÓN Y VISUALIZACIÓN DEL MAPA 2 (youtube.com)
- Mapa 3: GENERACIÓN Y VISUALIZACIÓN DEL MAPA 3 (youtube.com)

Enlaces al repositorio con los archivos de los mapas generados y los códigos desarrollados (procesamiento de nubes y automatización de localización por ICP): https://github.com/FranciscoAnayaPalacios/TFG/