



Escuela de Ingeniería
Ingeniería Civil Eléctrica

**Implementación y análisis de herramienta de mapeo de entornos frutícolas mediante sensor
LiDAR y cámara estéreo montados en plataforma robótica móvil Warthog**

Ignacio Andrés Vásquez Vásquez

Profesor guía: Rodrigo Verschae

Profesor co-guía: Robert Guamán-Rivera

Comisión evaluadora: Alfonso Ehijo, Christopher Flores

Memoria para optar al título de Ingeniero Civil Eléctrico

Rancagua, Chile

Enero, 2024

Agradecimientos

Quisiera expresar mi profunda gratitud a todas las personas que han sido fundamentales en el camino hacia la culminación de esta etapa. En primer lugar, agradezco a mi familia por su apoyo incondicional y constante aliento, pilares fundamentales en cada etapa de este proceso. A mis padres, Patricia y Claudio; hermano, Alonso; y abuelos, Héctor y Teresa, les debo mi más profundo agradecimiento por su amor inquebrantable y su constante motivación.

También quiero expresar mi más sincero agradecimiento a mi pareja, Constanza. Su amor incondicional, comprensión, y apoyo inquebrantable han sido el faro que ilumina los momentos más desafiantes de este camino académico. Su presencia ha sido más que un pilar; ha sido el sustento emocional que me ha permitido alcanzar cada meta.

A mis queridos amigos Ariel, Alex, Daniel, Eduardo, Jairo, Maximiliano y Yersson, les agradezco por su apoyo, ánimo y los momentos de distracción que me han ayudado a mantener el equilibrio durante estos 5 años.

Un agradecimiento especial a mi profesor guía, Rodrigo Verschae, cuya orientación experta y sabias sugerencias han sido invaluable en la elaboración de este trabajo, además de brindarme la oportunidad de formar parte del laboratorio de Robótica y Sistemas Inteligentes. A mi profesor co-guía, Robert Guamán, quien ha demostrado constantemente su apoyo e interés para lograr de la mejor manera este trabajo con su experiencia.

A Hernán Reyes, miembro del laboratorio de Robótica y Sistemas Inteligentes, le agradezco por su orientación adicional y sus valiosos aportes que enriquecieron significativamente este trabajo.

Por último, pero no menos importante, quiero expresar mi gratitud al profesor Ignacio Bugueño por su constante apoyo y confianza en el primer acercamiento con el laboratorio, así como por considerarme en actividades de difusión y docencia, lo cual ha sido de gran agrado y ayuda en mi desarrollo profesional.

A cada uno de ustedes, mi más profundo agradecimiento. Este logro no habría sido posible sin su apoyo, orientación y aliento constante a lo largo de este viaje académico.

Índice de Contenidos

RESUMEN	10
CAPÍTULO 1: INTRODUCCIÓN	11
1.1 PLANTEAMIENTO DEL PROBLEMA Y MOTIVACIÓN	12
1.2 OBJETIVOS	12
1.2.1 <i>Objetivo general</i>	12
1.2.2 <i>Objetivos específicos</i>	12
1.3 ESTRUCTURA DEL TRABAJO DE TÍTULO	13
CAPÍTULO 2: ESTADO DEL ARTE	14
2.1 APLICACIONES Y USOS DE PLATAFORMAS ROBÓTICAS MÓVILES	14
2.2 SENSORES Y SU IMPORTANCIA EN LA ROBÓTICA	15
2.3 MAPEO DEL ENTORNO	16
CAPÍTULO 3: MATERIALES Y MÉTODOS	18
3.1 PLATAFORMA ROBÓTICA	18
3.2 CALIBRACIÓN Y CONFIGURACIÓN DEL EQUIPO EXPERIMENTAL	21
3.3 ALGORITMO DE MAPEO	29
3.4 EVALUACIÓN DEL ALGORITMO DE MAPEO	31
3.5 HERRAMIENTA DE VISUALIZACIÓN	32
CAPÍTULO 4: RESULTADOS	33
4.1 CALIBRACIÓN Y CONFIGURACIÓN DEL EQUIPO EXPERIMENTAL	33
4.2 ADQUISICIÓN DE DATOS	40
4.3 ALGORITMO DE MAPEO	41
4.4 EVALUACIÓN DEL ALGORITMO DE MAPEO	45
4.5 HERRAMIENTA DE VISUALIZACIÓN	47
CAPÍTULO 5: CONCLUSIÓN Y TRABAJOS FUTUROS	48

5.1 CONCLUSIÓN.....	48
5.2 TRABAJOS FUTUROS.....	49
REFERENCIAS	50
ANEXOS.....	53
ANEXO 1: ODOMETRÍA	53
ANEXO 2: TRANSFORMADAS ENTRE SENSORES	55
ANEXO 3: ADQUISICIÓN DE DATOS	56
ANEXO 4: GRÁFICOS DE ODOMETRÍA UTILIZADA EN MAPEO	58
ANEXO 5: MAPAS OBTENIDOS CON ODOMETRÍA INICIAL	60
ANEXO 6: MÉTRICAS PARA EVALUACIÓN CUANTITATIVA	61

Índice de Figuras

Figura 1: Esquema de metodología de adquisición, procesamiento, implementación y visualización de un algoritmo de mapeo en aplicaciones frutícola. Fuente: elaboración propia.	18
Figura 2: Modelo 3D Plataforma robótica Warthog con restricción holonómica. Fuente: elaboración propia.	19
Figura 3: Vista frontal de la distribución del equipo de sensores montados sobre la plataforma robótica Warthog. Fuente: elaboración propia.	20
Figura 4: Emulación de funcionamiento del robot Warthog un ambiente agrícola través del software Gazebo. Fuente: elaboración propia.	21
Figura 5: Sistema de coordenadas cartesianas de referencia del robot móvil tipo anfibio conocido como Warthog. Fuente: elaboración propia.	23
Figura 6: Vista frontal de asignación de ejes de referencia entre los sensores del robot Warthog para operaciones de transformación. Fuente: elaboración propia.	25
Figura 7: Sistema de coordenadas de referencias utilizando por el paquete de calibración disponible para la cámara ZED y sensor LiDAR. Fuente: elaboración propia.	29
Figura 8: Proceso de adquisición de datos e implementación del algoritmo de mapeo basado en un sensor LiDAR y una cámara ZED de un entorno frutícola. Fuente: elaboración propia.	30
Figura 9: Diagrama correspondiente a la metodología propuesta para la evaluación del algoritmo de mapeo. Fuente: elaboración propia.	31
Figura 10: Odometría y trayectoria GPS obtenida con datos de salida a terreno fundo la Torre (13–09–2023). Fuente: elaboración propia.	33
Figura 11: Odometría y trayectoria GPS obtenida con datos de salida a terreno fundo la Torre (24–11–2023). Fuente: elaboración propia.	34
Figura 12: Odometría obtenida con datos de GPS. Fuente: elaboración propia.	34
Figura 13: Configuración inicial y modificada de distribución de sensores montados en la plataforma Warthog. Fuente: elaboración propia.	36
Figura 14: Simulación de plataforma robótica Warthog en software Gazebo con modificación implementada. Fuente: elaboración propia.	36

Figura 15: Imagen de cámara estéreo ZED 2 con nube de puntos de sensor LiDAR VLP-16 Fuente: elaboración propia.	37
Figura 16: Comando para ejecutar la calibración de la cámara. Fuente: elaboración propia.	37
Figura 17: Captura de pantalla correspondiente al algoritmo de calibración. Fuente: elaboración propia.	38
Figura 18: Imagen obtenida luego de realizar la calibración. Fuente: elaboración propia.	39
Figura 19: Comando para ejecutar la calibración entre la cámara y el sensor LiDAR. Fuente: elaboración propia.	39
Figura 20: Comando para ejecutar la calibración entre la cámara y el sensor LiDAR. Fuente: elaboración propia.	40
Figura 21: Simulación en Gazebo. Fuente: elaboración propia.....	41
Figura 22: Comando para ejecutar el algoritmo de mapeo OctoMap. Fuente: elaboración propia.	41
Figura 23: Comando para guardar el mapeo realizado. Fuente: elaboración propia.	42
Figura 24: Generación de mapa en tiempo real con OctoMap, con datos de cámara estéreo (Simulación). Fuente: elaboración propia.....	42
Figura 25: Experimentos de mapeo en un entorno frutícola, llevados a cabo mediante la utilización de datos de una cámara estéreo en un entorno de simulación. Fuente: elaboración propia.	42
Figura 26: Generación de un mapa en un entorno frutícola simulado con el algoritmo OctoMap, utilizando datos de un sensor LiDAR. Fuente: elaboración propia.	43
Figura 27: Generación de mapas utilizando información del sensor LiDAR en un entorno frutícola simulado. Fuente: elaboración propia.....	43
Figura 28: Generación de mapa frutícola a partir de datos obtenidos durante la primera salida a terreno llevado a cabo en el Fundo La Torre. Fuente: elaboración propia.	44
Figura 29: Generación de mapas de un entorno frutícola, contemplando diversas trayectorias del robot Warthog. Fuente: elaboración propia.	45

Figura 30: Comando para ejecutar la herramienta de visualización de los mapas. Fuente: Elaboración propia.	47
Figura 31: Ejecucion de herramienta de visualización habilitada. Fuente: elaboración propia..	47
Figura 32: Comportamiento de desplazamiento asociado a la odometría del robot Warthog en el software de visualización RVIZ. Fuente: elaboración propia.....	53
Figura 33: Configuración de odometría original. Fuente: elaboración propia.....	53
Figura 34: Configuración de odometría propuesta. Fuente: elaboración propia.....	54
Figura 35: Corrección de desplazamiento de odometría en el software RVIZ. Fuente: elaboración propia.	54
Figura 36: Diagrama de árbol obtenido a través del código de Python. Fuente: elaboración propia.	55
Figura 37: Comando para grabar tópicos desde archivo de Python creado. Fuente: elaboración propia.	56
Figura 38: Visualización de los datos registrados en RVIZ, correspondiente al fundo la Torre, Rengo–Chile. Fuente: elaboración propia.	57
Figura 39: Elaboración de trayectorias mediante odometría y datos GPS en un entorno simulado, con el propósito de implementarlas en un algoritmo de mapeo utilizando una cámara estéreo. Fuente: elaboración propia.....	58
Figura 40: Elaboración de trayectorias mediante odometría y datos GPS en un entorno simulado, con el propósito de implementarlas en un algoritmo de mapeo utilizando un sensor LiDAR. Fuente: elaboración propia.....	58
Figura 41: Elaboración de trayectorias mediante odometría y datos GPS en un ambiente real, con el propósito de implementarlas en un algoritmo de mapeo de un entorno frutícola. Fuente: elaboración propia.	59
Figura 42: Elaboración de mapas mediante la utilización de datos reales obtenidos en el Fundo La Torre. Fuente: elaboración propia.	60

Índice de tablas

Tabla 1: Resumen de Tecnología Sensorial implementada en Aplicaciones Robóticas y Agrícolas. Fuente: elaboración propia.....	15
Tabla 2: Especificaciones técnicas de plataforma robótica. Fuente: («Warthog Unmanned Ground Vehicle Robot – Clearpath», s. f.).....	19
<i>Tabla 3: Descripción de las principales características de los sensores de la plataforma robótica Warthog. Fuente: elaboración propia.....</i>	20
Tabla 5: Parámetros de transformadas de sensores con respecto a estructura de soporte. Fuente: elaboración propia.	35
Tabla 6: Parámetros de modificación de transformadas de sensores con respecto a estructura de soporte. Fuente: elaboración propia.....	36
Tabla 7: Métricas obtenidas al realizar mapeo con sensor LiDAR. Fuente: elaboración propia..	46
Tabla 8: Métricas obtenidas al realizar mapeo con cámara estéreo ZED 2. Fuente: elaboración propia.	46
Tabla 9: Detalle de los archivos .bag generados durante las experiencias de terreno. Fuente: elaboración propia.	57
Tabla 10: Métricas obtenidas al realizar mapeo con sensor LiDAR. Fuente: elaboración propia.	61
Tabla 11: Métricas obtenidas al realizar mapeo con cámara estéreo ZED 2. Fuente: elaboración propia.	61

Lista de abreviaturas

- **LiDAR:** *Light Detection and Ranging.*
- **GPS:** *Global Position System.*
- **AP:** Agricultura de precisión.
- **PIB:** Producto interno bruto.
- **UGV:** *Unmanned Ground Vehicle.*
- **RTK–GNSS:** *Real–Time Kinematic Global Navigation Satellite System.*
- **IMU:** *Inertial Measurement Unit.*
- **SLAM:** *Simultaneous Localization and Mapping.*
- **ROS:** *Robot Operating System.*
- **URDF:** *Unified Robot Description Format.*
- **TF:** *Transform Frames.*
- **PCD:** *Point Cloud Data.*

Resumen

El tener conocimiento del entorno agrícola representa una herramienta esencial para mejorar la gestión de los recursos, y evaluar el potencial de diversas estrategias destinadas a mejorar la eficiencia de las operaciones. A pesar de los últimos avances en la agricultura de precisión, todavía quedan varios desafíos a considerar, como lo es generar un mapa preciso de cultivos en entornos no estructurados. Dada su importancia en la economía de la región de O'Higgins, este trabajo de título se enfoca en la recopilación de datos frutícolas utilizando una plataforma robótica móvil equipada con diversos sensores, lo cual permite recolectar datos y crear mapas digitales del entorno. La metodología implementada comprende de un proceso de calibración de los diversos sensores, para garantizar un correcto funcionamiento en adquisición y preprocesamiento de la información obtenida de los mismo. Posteriormente, sobre la base a una calibración meticulosa, se implementó un algoritmo de mapeo para generar un mapa del entorno frutícola mediante el uso de un sensor LiDAR (*Light Detection and Ranging*) y una cámara estéreo. Estos mapas obtenidos, tanto en el ambiente simulado como en las salidas de terreno al Fundo La Torre en Rengo, Chile, fueron evaluados cualitativa y cuantitativamente, incluyendo una comparación visual de los mapas y análisis de rendimiento del algoritmo con porcentajes de uso de la RAM y la CPU mientras el algoritmo de mapeo se ejecutaba. Además, el algoritmo de mapeo fue validado a través de simulaciones, garantizando la confiabilidad de los mapas generados. La calibración se erige como un pilar fundamental para asegurar la fiabilidad y exactitud de los procesos de mapeo generados. Cabe destacar que los resultados del trabajo de título brindan a los productores una herramienta para comprender patrones y desarrollar estrategias para la toma de decisiones en el manejo de los cultivos.

Palabras clave: agricultura de precisión, robótica móvil, mapeo, entorno frutícola.

Capítulo 1: Introducción

A lo largo de los años, la agricultura se ha consolidado como un pilar fundamental en la sociedad y economía de muchos países, desempeñando un papel esencial en la alimentación de la población y la generación de empleo (Pawlak & Kołodziejczak, 2020). Sin embargo, el crecimiento de la población mundial y la consiguiente demanda de alimentos han ejercido una presión cada vez mayor sobre los recursos naturales, lo que ha llevado a una explotación excesiva de la tierra y el agua, y al agotamiento de los suelos. Además, el cambio climático ha generado desafíos adicionales, como la variabilidad del clima, las sequías y las inundaciones, que afectan la producción agrícola de manera significativa (Lozano-Povis et al., 2021).

La agricultura de precisión (AP) tiene como propósito maximizar el rendimiento, minimizar los costos y optimizar los recursos y producción. La AP emplea tecnologías avanzadas y datos precisos para optimizar la producción agrícola y minimizar los impactos negativos en el medio ambiente (Monteiro et al., 2021). En este contexto, el uso de plataformas robóticas, como brazos robóticos, drones y plataformas móviles, junto con algoritmos de visión por computadora, aprendizaje automático (*machine learning*) y sistemas de control, ha llevado a una transformación fundamental de la agricultura tradicional.

Estos avances tecnológicos no solo han simplificado y automatizado tareas agrícolas repetitivas, sino que también han abierto nuevas oportunidades para recopilar datos en tiempo real y realizar análisis detallados de la salud de los cultivos, las condiciones del suelo y otros parámetros críticos para la producción agrícola. Dado esto, el mapeo de entornos frutícolas juega un rol importante en la agricultura de precisión, ya que permite una evaluación exhaustiva de las condiciones de crecimiento, la distribución de cultivos y la detección de problemas potenciales.

En este contexto, con el fin de mejorar el análisis de datos en el sector agrícola, se ha equipado de sensores las diferentes plataformas robóticas con cámaras multispectrales, cámaras termográficas, GPS (*Global position System*) y LiDAR (*Light Detection and Ranging*) permitiendo realizar tareas de navegación, análisis de cultivos y generación de mapas agrícolas.

1.1 Planteamiento del problema y motivación

El sector agrícola representa una de las principales fuentes de ingresos en el Producto Interno Bruto (PIB) de Chile. Sin embargo, ha experimentado una caída del 0.6 %, principalmente debido a la actividad agropecuaria y silvícola (*Informe de Cuentas Nacionales de Chile primer trimestre de 2023 – Banco Central de Chile, s. f.*). Dada esta situación, el eficiente desarrollo de los procesos agrícolas es crucial debido al modelo exportador del país. En este contexto, la agricultura en Rancagua, la sexta región de Chile, desempeña un papel importante al generar empleo y abastecer de alimentos a nivel nacional e internacional.

La adopción de la AP se presenta como una solución valiosa para afrontar los desafíos relacionados con la escasez de agua y la necesidad de aumentar la productividad. El trabajo de título presente tiene como objetivo utilizar una plataforma robótica móvil y diversos sensores para crear un mapa del entorno frutícola que apoye la toma de decisiones orientadas a la reducción de costos y la mejora de la calidad de los cultivos. Además, se busca fomentar la innovación agrícola, impulsando la competitividad y el desarrollo sostenible de la región.

1.2 Objetivos

1.2.1 Objetivo general

- Generar un mapa del entorno frutícola mediante la recopilación de datos provenientes de sensor LiDAR y cámara estéreo integrados en una plataforma robótica móvil, en el contexto de la agricultura de precisión.

1.2.2 Objetivos específicos

- Realizar una revisión de literatura sobre plataformas robóticas móviles, el uso de sensores en robótica y el mapeo de entornos con robots.
- Realizar la configuración del equipo experimental mediante el uso de algoritmos y técnicas de calibración.
- Implementar un algoritmo de mapeo del entorno frutícola mediante la información recopilada a través del equipo experimental y uso de herramientas de mapeo.
- Evaluar el algoritmo de mapeo utilizando un software de simulación y en campo, mediante un análisis cualitativo y cuantitativo.

- Habilitar una herramienta de visualización de los datos mapeados.

1.3 Estructura del trabajo de título

La estructura del trabajo de título es la siguiente:

- **Capítulo 1: Introducción** Este capítulo presenta el contexto del trabajo de título, el planteamiento y los objetivos.
- **Capítulo 2: Estado del arte** En este capítulo se destaca la versatilidad de las plataformas robóticas móviles como el Warthog en diversas industrias, se enfatiza la importancia de los sensores en robótica y se realiza un estudio del estado del arte, abordando aplicaciones de mapeo de entornos agrícolas.
- **Capítulo 3: Materiales y métodos** En este capítulo se propone una estrategia para desarrollar un sistema de mapeo frutícola con una plataforma robótica, incluyendo la familiarización con la plataforma, calibración de sensores, implementación de algoritmos, evaluación y herramientas de visualización.
- **Capítulo 4: Resultados** En este capítulo se presentan los resultados obtenidos acorde a la metodología propuesta en el capítulo anterior.
- **Capítulo 5: Conclusión y trabajos futuros** Este capítulo presenta las conclusiones y las propuestas de futuros trabajos.

Capítulo 2: Estado del arte

La integración de robots en tareas agrícolas ha permitido desarrollar diversas investigaciones asociadas a la fusión de sensores que permiten toma decisiones en base al análisis de información relevante. Además, se destaca cómo la combinación de diferentes tipos de sensores está transformando la agricultura, posibilitando una gestión eficiente y sostenible mediante la creación de mapas detallados y la detección precisa de elementos en el entorno. Este análisis global subraya el potencial y la constante necesidad de investigación en sensores y técnicas de percepción para mejorar la aplicación de la robótica móvil en diversas áreas industriales.

2.1 Aplicaciones y usos de plataformas robóticas móviles

El uso de robot en diversas aplicaciones ha reportado la versatilidad para adaptarse en el sector industrial. Además, se ha identificado la integración de robot móviles en tareas de minería, labores de rescate y otras actividades.

En tareas de inspección autónoma de cintas transportadoras en minas subterráneas, Szerk et al. (2022) ha reportado el uso de un robot móvil. Por otro lado, en tareas de búsqueda y rescate se ha combinado un robot móvil y un tipo serpiente (Kamegawa et al., 2020). En esta línea, Wang et al (2020) propone el uso de una plataforma móvil para realizar reconstrucción tridimensional. En aplicaciones agricultura de precisión se destaca la utilización de UGV (*Unmanned Ground Vehicle*) robóticos para llevar a cabo una amplia gama de tareas agrícolas, que incluyen la siembra, plantación, deshierbe, tratamiento, poda, recolección, manipulación y cosecha (Botta et al., 2022). Estos robots también se emplean en tareas de monitoreo y sensado remoto en la agricultura. Además, Git et al. (2023) reporta el uso de plataformas robóticas móviles utilizadas tanto en la investigación como en aplicaciones comerciales en la agricultura. Se mencionan ejemplos notables de robots, como el Husky, que se especializa en la navegación autónoma en entornos agrícolas; el BoniRob y Robotanist, utilizados para el fenotipado de cultivos y plantas, y el robot Thorvald, diseñado para aplicaciones en la agricultura de precisión y la investigación agrícola. La versatilidad de los robots móviles se muestra claramente en estos ejemplos, evidenciando su habilidad para encarar diversas labores agrícolas. Este aspecto resalta el

potencial valioso de la robótica en la agricultura, ya que contribuye a optimizar procesos y mejorar la eficiencia en la agricultura de precisión.

Por otra parte, en aplicaciones de mantenimiento autónomo de vías férreas, Rahman et al. (2021) propone un Sistema Autónomo de Inspección y Reparación Robótica (RIRS) para el cuidado de las vías ferroviarias. En este trabajo se reporta la relevancia del uso de robots móviles en tareas de automatización en aplicaciones de agricultura e infraestructura ferroviaria, demostrando el amplio potencial de la robótica móvil en diversas industrias.

2.2 Sensores y su importancia en la robótica

En aplicaciones de robótica, los sensores juegan un rol fundamental al posibilitar que los robots sean capaces de percibir su entorno y tomar decisiones basadas en la información que recolectan. Alatise y Hancke (2020) destacan la importancia del uso de sensores en tareas de percepción, describiendo la relevancia de fusión de sensores en aplicaciones de robots móviles autónomos.

Los sensores son dispositivos de gran importancia que tienen la capacidad de detectar y recopilar información acerca del entorno que los rodea. Entre ellos, se tienen las siguientes categorías de sensores (ver Tabla 1):

Tabla 1: Resumen de Tecnología Sensorial implementada en Aplicaciones Robóticas y Agrícolas. Fuente: elaboración propia.

Sensor	Descripción	Aplicación	Referencia
Cámara	Capturan imágenes y vídeos mediante la captación de luz.	Medición de la altura de los cultivos y la creación mapas tridimensionales.	(Kim et al., 2021) (Rovira-Más et al., 2008)
LiDAR	Emplea láseres para medir distancias y crear nubes de puntos tridimensionales del entorno.	Detección de estructuras, digitalización árboles y plantas, y toma decisiones.	(Rivera et al., 2023)
GPS	Proporciona una ubicación precisa a través de señales satelitales.	Determinación de posición y altitud, permitiendo la navegación en tiempo real.	(Vieira et al., 2022) (Rovira-Más et al., 2008)
IMU	Registra la aceleración y orientación del dispositivo.	Se combina con GPS para determinar la posición y altitud en vehículos autónomos, mejorando la navegación.	(Vieira et al., 2022) (Rovira-Más et al., 2008)

En el campo de la agricultura, la integración de sensores desempeña un papel fundamental al revolucionar la forma en que se abordan los desafíos de la industria agrícola. Por ejemplo, Vieira et al. (2022) resalta el uso de sensores RTK-GNSS (*Real-Time Kinematic Global Navigation Satellite System*) y una IMU (*Inertial Measurement Unit*) en vehículos agrícolas autónomos para la estimación de la posición y altitud, lo que permite una navegación y guía de vehículos en tiempo real. Además, Kim et al. (2021) demuestra cómo la visión estéreo se utiliza para medir con precisión la altura de los cultivos en robots de cultivo autónomos, lo que es esencial para optimizar las prácticas agrícolas. Por otra parte, Rovira-Más et al. (2008) destaca cómo las cámaras estéreo, los sensores de localización y las IMU se combinan para crear mapas tridimensionales de terrenos en tiempo real, lo que resulta en aplicaciones valiosas para la agricultura de precisión y la robótica agrícola. En esta dirección, Rivera et al. (2023) resalta el papel esencial de la tecnología LiDAR en la detección de estructuras, la digitalización de árboles y plantas, y la toma de decisiones en la agricultura, destacando la versatilidad de los sensores en esta industria en constante evolución.

La importancia del sensorizado en la operación de robots móviles autónomos ha sido reportada por Alatisse y Hancke (2020), destacando la recolección de información sobre el entorno, incluyendo datos de localización, detección de obstáculos y navegación. Estos datos son vitales para que los robots tomen decisiones informadas y se desplacen de forma segura y eficiente en su ambiente.

2.3 Mapeo del entorno

El mapeo de entornos frutícolas con plataformas robóticas móviles y diversos sensores es una tarea fundamental para el desarrollo de robots capaces de realizar distintas tareas de manera autónoma, esto debido a que permiten obtener una percepción detallada y precisa del ambiente en el que operan.

Kang y Chen (2019) investigaron la recolección de frutas con robots en huertos no estructurados, centrándose en la percepción visual y la creación de modelos de frutas. Para lo cual emplearon modelos de aprendizaje profundo para la percepción visual, “*OctoMap*” para mapear objetos en el entorno y desarrollaron un módulo de modelado de frutas. Los resultados

han demostrado la precisión de su enfoque en la detección y localización de frutas, así como en la representación de escenarios de trabajo en entornos de huertos reales.

Por otro lado, Wang et al. (2021) reportan una estrategia de detección y mapeo de árboles en entornos forestales utilizando cámaras estéreo. En este caso, implementan un detector de objetos en 3D, entrenado específicamente para identificar árboles en este tipo de entorno. Los resultados muestran la relevancia de la detección de objetos para la creación de mapas precisos en ambientes no estructurados.

Chen et al. (2021) se centraron en la percepción tridimensional (3D) y la cartografía global en huertos no estructurados. Para lograr este objetivo, emplearon visión estéreo “*eye-in-hand*” y SLAM (*Simultaneous Localization and Mapping*) para generar un mapa detallado y global del huerto. Este enfoque de investigación ha contribuido significativamente al desarrollo de mapas de alta resolución en entornos agrícolas, una contribución fundamental en aplicaciones de robótica en agricultura.

Han et al. (2023) proponen un nuevo algoritmo conocido como “*GardenMap*” para mapear nubes de puntos estáticos en jardines. El algoritmo se centra en la creación de mapas de nubes de puntos estáticos para ayudar a la localización y navegación de robots en entornos de jardines. Este estudio resalta la importancia de desarrollar algoritmos adaptados a entornos agrícolas, lo que tiene implicaciones relevantes para la cartografía en huertos.

Xiong et al. (2023) enfocaron su investigación en la relevancia de los mapas semánticos en tareas de agricultura de precisión y presentaron un método para la creación de mapas semánticos 3D en huertos de cítricos en tiempo real. El trabajo propuesto combina SLAM VINS-*RGBD* basada en visión-inercial con el algoritmo de segmentación semántica BiSeNetV1 para generar mapas precisos y eficientes. Este trabajo subraya la importancia de la información semántica en el mapeo agrícola y su utilidad en aplicaciones en tiempo real.

Los estudios mencionados ofrecen valiosos enfoques en percepción, detección y modelización de entornos agrícolas, contribuyendo al desarrollo de robots capaces de mapear y realizar diversas tareas del rubro de manera eficiente y precisa. Estos avances desempeñan un papel fundamental en la mejora de la productividad y sostenibilidad de la industria agrícola.

Capítulo 3: Materiales y métodos

En este capítulo, se presentan y revisan los materiales para llevar a cabo el trabajo de título el cual tiene como objetivo desarrollar un sistema de mapeo de entorno frutícola utilizando una plataforma robótica móvil, en donde se contempla la plataforma robótica y los sensores a utilizar. Además, se presenta la metodología correspondiente a cada etapa necesaria, que abarca la calibración y configuración del equipo, la recolección de datos, la implementación, la evaluación y la representación visual del mapeo, la cual se muestra en la Figura 1.



Figura 1: Esquema de metodología de adquisición, procesamiento, implementación y visualización de un algoritmo de mapeo en aplicaciones frutícola. Fuente: elaboración propia.

3.1 Plataforma robótica

La plataforma robótica conocida como Warthog es ideal para el trabajo de título debido a su versatilidad y habilidad para funcionar en entornos desafiantes. Diseñada para enfrentar terrenos difíciles, puede maniobrar desde suelos blandos y vegetación densa hasta lodo espeso y pendientes pronunciadas. Warthog es un vehículo terrestre no tripulado todo terreno que incluso puede desplazarse brevemente en el agua («Warthog Unmanned Ground Vehicle Robot – Clearpath», s. f.).

Warthog es una plataforma no holonómica (ver Figura 2), es decir, un tipo de vehículo que no puede moverse en todas las direcciones de manera libre debido a sus limitaciones de movimiento, caracterizada por su tracción en las cuatro ruedas y su habilidad anfibia. Esta versatilidad le permite destacarse en una amplia gama de situaciones. Además, se destaca por su flexibilidad de personalización, gracias a placas de montaje de carga útil y puertos de comunicación y alimentación accesibles. Esto facilita la incorporación de sensores,

manipuladores y otros equipos según las necesidades específicas de aplicaciones en campos como la minería y la agricultura. Además, los detalles de hardware del robot se muestran en la Tabla 2.

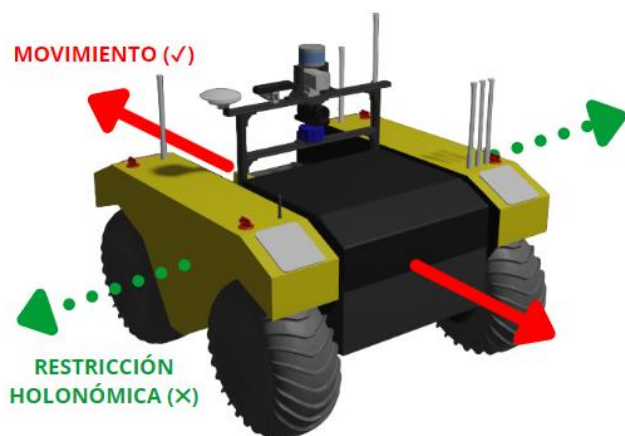


Figura 2: Modelo 3D Plataforma robótica Warthog con restricción holonómica. Fuente: elaboración propia.

Tabla 2: Especificaciones técnicas de plataforma robótica. Fuente: («Warthog Unmanned Ground Vehicle Robot – Clearpath», s. f.)

Característica	Especificación
Dimensiones	1,52 x 1,38 x 0,83 m
Peso	280 kg
Peso máximo de carga	272 kg
Velocidad máxima	18 km/h
Autonomía	3 horas

Sensores integrados

En la actualidad, la plataforma que forma parte del equipamiento del Laboratorio de Robótica y Sistemas Inteligentes (RISLab) de la Universidad de O'Higgins, está equipada con una variedad de sensores. En la Tabla 3, se presenta un detalle de los sensores: el LiDAR 3D Velodyne Puck (VLP-16), con un alcance de detección superior a 100 metros, y la cámara estéreo ZED 2 de Stereolab. La cámara ZED 2 ofrece una resolución de hasta 2.2K y una velocidad de captura de 15 FPS.

Además, es importante mencionar que, la plataforma robótica está equipada con varios dispositivos adicionales que desempeñan un papel crucial en la percepción del entorno, el movimiento y la ubicación del robot. La Tabla 3 resume las principales características de los sensores, tales como: la cámara FLIR Blackfly S, con una resolución de 6.3 MP y una frecuencia de captura de 59.6 FPS, junto con la cámara termográfica FLIR A400, que ofrece una resolución de (320 x 240) píxeles. Completando este conjunto de sensores, están la IMU UM7, que proporciona datos precisos sobre la orientación, aceleración y velocidad del robot, y el GPS

SwiftNav Duro RTK, que garantiza una localización precisa gracias a la corrección en tiempo real (RTK).

Tabla 3: Descripción de las principales características de los sensores de la plataforma robótica Warthog. Fuente: elaboración propia.

	Sensor	Resolución	FOV	FPS	Hz
Sensores utilizados en el algoritmo de mapeo	LiDAR – Velodyne Puck (VLP-16)	2.0°	360° x 30°	–	10
	Cámara estéreo – Stereolab ZED 2	1920 x 1080	110° x 70°	30	–
Sensores adicionales	Cámara – FLIR Blackfly S	3072 x 2048	–	59.6	–
	Cámara Termográfica – FLIR A400	640 x 480	42°	30	–
	IMU UM7	0.01°	–	–	20
	GPS – SwiftNav Duro RTK	1 cm	–	–	10

Para obtener una mejor noción de cómo se encuentran distribuidos los sensores descritos anteriormente, la Figura 3 muestra un esquema 2D que representa esta distribución en el marco instalado en la plataforma robótica Warthog.

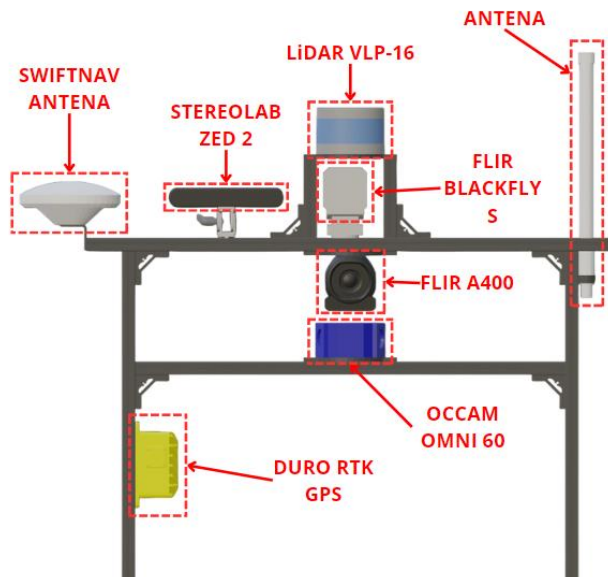


Figura 3: Vista frontal de la distribución del equipo de sensores montados sobre la plataforma robótica Warthog. Fuente: elaboración propia.

Software

Warthog está equipado con dos computadoras (una principal y una secundaria), cada una con el sistema operativo Ubuntu 20.04 y ejecuta ROS (*Robot Operating System*) versión Noetic. Este software y hardware combinados son fundamentales para las capacidades de Warthog y su capacidad de realizar tareas complejas en una variedad de aplicaciones.

Es importante destacar que esta plataforma puede ser emulada en el software Gazebo (ver Figura 4), el cual posibilita la realización de simulaciones realistas del robot en entornos virtuales antes de su despliegue en el mundo real garantizando un buen rendimiento de los algoritmos.

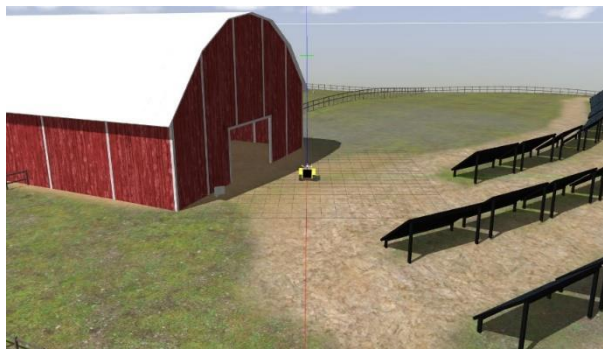


Figura 4: Emulación de funcionamiento del robot Warthog un ambiente agrícola través del software Gazebo. Fuente: elaboración propia.

3.2 Calibración y configuración del equipo experimental

Para lograr una correcta configuración y calibración del equipo en cuestión, se deben seguir una serie de pasos y consideraciones clave. En este contexto, se centrarán en tres aspectos fundamentales: la configuración de la odometría, la configuración manual del URDF (*Unified Robot Description Format*) y la calibración de la cámara-LiDAR. Estos procesos son esenciales para garantizar el funcionamiento preciso y confiable de un robot en un entorno robótico y son cruciales para tareas de navegación autónoma, percepción del entorno y toma de decisiones. A continuación, se profundizará en cada uno de estos aspectos para comprender su importancia y cómo llevar a cabo estas configuraciones de manera efectiva.

Odometría

La odometría es una técnica fundamental en robótica que se utiliza para estimar la posición y el desplazamiento de un robot en función de la información recopilada por distintos

sensores. En el contexto de ROS, la configuración de la odometría es un aspecto fundamental para lograr una navegación precisa. ROS proporciona un marco de trabajo que permite la integración de diversos sensores, como encoders en las ruedas y sensores inerciales, para estimar la posición y la orientación del robot. La odometría se basa en la recopilación y procesamiento de datos de estos sensores, lo que permite al robot calcular su movimiento y desplazamiento en un espacio tridimensional. Para mejorar la precisión de las estimaciones de posición, se recurre a la implementación de algoritmos de fusión sensorial, como los ofrecidos por el paquete *robot_localization* en ROS. Estos algoritmos permiten combinar los datos de los sensores de manera efectiva, lo que es esencial para que el robot pueda llevar a cabo tareas de navegación autónoma y mapeo en entornos desconocidos de manera eficiente y confiable.

Para configurar la odometría en ROS, se utiliza un archivo YAML (formato de serialización de datos legible por humanos que se utiliza comúnmente para configurar aplicaciones y servicios) (.yaml) que desempeña un papel crucial. Este archivo de configuración .yaml permite definir los parámetros clave que afectan la odometría del robot, como la distancia entre las ruedas, la resolución de los encoders y otros factores relevantes. Al ajustar y personalizar estos parámetros en el archivo .yaml, se puede adaptar la odometría a las especificaciones del robot, lo que resulta en estimaciones de posición más precisas y confiables. Este enfoque modular y basado en archivos de configuración facilita la adaptación de la odometría a diferentes plataformas de robots y sensores, lo que es esencial para lograr un rendimiento óptimo en una variedad de aplicaciones y entornos.

Dentro del archivo .yaml, se define la eq. (1) que especifica qué variables deben ser utilizadas de cada fuente de odometría, es decir, se especifica las posiciones, velocidades y aceleraciones angulares y lineales que se desee utilizar. A continuación, se define la estructura de esta matriz.

$$M = \begin{bmatrix} x & y & z \\ roll & pitch & yaw \\ \dot{x} & \dot{y} & \dot{z} \\ \ddot{x} & \ddot{y} & \ddot{z} \end{bmatrix} \quad (1)$$

Como se mencionó previamente, cada componente de la matriz se encuentra relacionado con el modelo del robot que describe su movimiento. En la Figura 5 se pueden identificar las componentes en las cuales “x”, “y”, y “z” representan las variables lineales (posición, velocidad y aceleración), mientras que “roll”, “pitch”, y “yaw” se refieren a las variables angulares (posición, velocidad y aceleración).

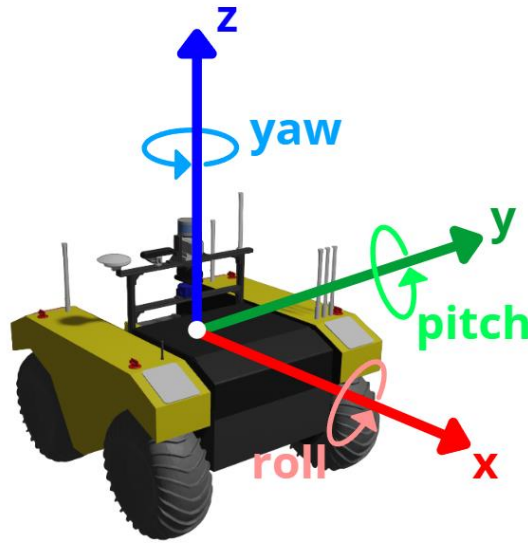


Figura 5: Sistema de coordenadas cartesianas de referencia del robot móvil tipo anfibio conocido como Warthog. Fuente: elaboración propia.

Se ha desarrollado la odometría del robot utilizando datos del GPS como base. Para lograrlo, se transformaron las coordenadas geográficas (latitud y longitud) a metros empleando las siguientes escalas de conversión definidas en la eq. (2) y eq. (3):

$$1^{\circ} \text{ latitud} = 111321 \text{ [m]} \quad (2)$$

$$1^{\circ} \text{ longitud} = \frac{40075000 * \cos(\text{latitud}[^{\circ}])}{360} \text{ [m]} \quad (3)$$

Una vez convertidas las coordenadas, se pueden calcular elementos cruciales como la velocidad y la orientación. La velocidad se obtiene al medir la diferencia entre dos puntos en momentos distintos, expresada en metros, y luego dividiendo esta diferencia por el tiempo transcurrido. Esto proporciona una visión dinámica del desplazamiento en función del tiempo.

Por otro lado, la orientación se determina al calcular el ángulo de dirección entre dos puntos, generalmente en relación con el norte geográfico. Este cálculo se realiza utilizando

funciones trigonométricas, como la tangente inversa, que tiene en cuenta las coordenadas en metros de los puntos para establecer la orientación del desplazamiento.

Estos pasos se llevaron a cabo mediante un código implementado en Python. Este código suscribe el tópico del GPS, procesa los datos de latitud y longitud obtenidos y aplica las escalas de conversión para transformarlos a metros. Luego, calcula la diferencia entre los puntos para determinar la velocidad y la orientación del desplazamiento. Una vez completadas estas operaciones, el código publica la nueva odometría, proporcionando información sobre la posición, velocidad y dirección del desplazamiento de la plataforma robótica.

Configuración “manual”

En el contexto de ROS, la calibración precisa de los sensores y la correcta correspondencia entre la descripción del robot en el archivo URDF y su representación en el mundo real son esenciales para el funcionamiento del paquete TF (*Transform Frames*) (Foote, 2013), una herramienta fundamental para la gestión de transformaciones de coordenadas de la plataforma robótica.

La información de calibración y la descripción precisa del robot en el URDF se integran en el paquete TF de ROS para garantizar que los datos recopilados por los sensores se relacionen adecuadamente con el marco de referencia global del robot. Esto permite que el robot comprenda su posición y orientación en relación con su entorno y otros objetos en ese entorno. TF de ROS se encarga de gestionar estas relaciones de manera dinámica, lo que facilita la navegación y la interacción efectiva del robot con el mundo que lo rodea. La configuración manual del URDF es un paso fundamental para asegurar que las transformaciones TF sean precisas y, como resultado, que el robot funcione de manera confiable en una variedad de aplicaciones robóticas.

Para lograr una calibración precisa de los sensores en el robot, es necesario realizar una configuración manual del URDF. Esto implica la modificación de la descripción del robot para integrar sensores que inicialmente no estaban incorporados en su diseño original. La configuración manual del URDF se convierte en un paso crucial para garantizar que los datos de los sensores se correlacionen de manera óptima con las acciones del robot en el mundo real, lo

que resulta en una calibración efectiva. Este proceso es esencial para lograr un rendimiento preciso y confiable en tareas robóticas que requieren una interacción sensible con el entorno.

Es fundamental tener en cuenta que, al realizar la configuración manual del URDF para la incorporación de sensores adicionales, se debe lograr una precisión en la traslación y rotación de estos sensores. Esto implica que la posición y orientación de los sensores en el URDF deben corresponder de manera exacta con su ubicación en el mundo real.

Es importante tener en cuenta que la plataforma adquirida por la Universidad de O'Higgins presenta modificaciones, por lo tanto, es necesario considerar la configuración del archivo XML (URDF) que permite describir al robot. Para comprender mejor cómo están relacionadas las posiciones y rotaciones de los sensores, es necesario tener una noción de las transformadas existentes entre ellos (TF). Para este propósito, se ha implementado un algoritmo que permite obtener visualmente estas transformaciones.

Este proceso garantiza que los datos recopilados por los sensores se relacionen de manera precisa con la realidad, permitiendo una calibración efectiva que mejora la capacidad del robot para interactuar y comprender su entorno. Cualquier desviación en la posición y orientación de los sensores puede llevar a errores significativos en la percepción del robot y, por lo tanto, afectar su capacidad de respuesta y toma de decisiones. En la Figura 6, se muestra la distribución de transformaciones presente entre los sensores y sus respectivos sistemas de referencia.

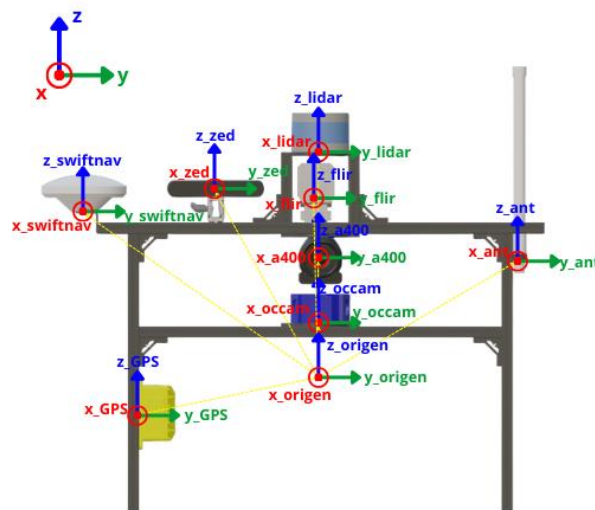


Figura 6: Vista frontal de asignación de ejes de referencia entre los sensores del robot Warthog para operaciones de transformación. Fuente: elaboración propia.

Calibración de cámara

La calibración de la cámara es un proceso fundamental que debe considerarse al trabajar con diversos sensores para lograr una percepción precisa del entorno en el que se opera. En robótica, la calibración de la cámara se lleva a cabo para garantizar la precisión de la información capturada por la cámara y su efectiva utilización en aplicaciones de visión por computadora, navegación y otros campos.

ROS proporciona un paquete llamado *camera_calibration* que simplifica el proceso de calibración de cámaras. Este paquete provee herramientas y utilidades que facilitan la calibración intrínseca de la cámara, lo que hace que el proceso sea más sencillo. Con el paquete *camera_calibration*, es posible recopilar datos de calibración a partir de imágenes capturadas por la cámara y realizar ajustes precisos en los parámetros de la cámara, lo que mejora de manera significativa la calidad de las imágenes y la precisión en la percepción del entorno en aplicaciones robóticas.

Los parámetros de calibración son esenciales en el proceso de calibración de la cámara, ya que desempeñan un papel fundamental en la percepción del entorno en el que opera el sistema robótico. Estos parámetros permanecerán constantes en todos los mensajes hasta que se realice una nueva calibración de la cámara. A continuación, se describen los componentes clave de estos parámetros:

- **Modelo de distorsión:** Este parámetro indica el modelo de distorsión utilizado. En la mayoría de las cámaras, se utiliza el modelo "plumb_bob," que es un modelo sencillo de distorsión radial y tangencial.
- **Parámetros de distorsión (D):** Los parámetros de distorsión varían según el modelo utilizado. Para el modelo "plumb_bob," se definen 5 parámetros: k_1 , k_2 , t_1 , t_2 y k_3 . Estos parámetros se utilizan para corregir las distorsiones en la imagen capturada.
- **Matriz de cámara intrínseca (K):** La eq. (4) describe los parámetros intrínsecos de la cámara para las imágenes sin procesar (distorsionadas). Incluye elementos como las distancias focales (f_x , f_y) y el centro óptico de la cámara en la imagen (c_x , c_y), que son esenciales para proyectar puntos 3D en coordenadas de píxeles 2D.

$$K = \begin{bmatrix} fx & 0 & cx \\ 0 & fy & cy \\ 0 & 0 & 1 \end{bmatrix} \quad (4)$$

- **Matriz de rectificación (R):** Esta matriz se emplea en cámaras estéreo y es una matriz de rotación que alinea el sistema de coordenadas de la cámara con el plano de imagen estéreo ideal. Esto garantiza que las líneas epipolares en ambas imágenes estéreo sean paralelas.
- **Matriz de proyección/cámara (P):** Esta matriz se utiliza para proyectar puntos 3D en el marco de coordenadas de la cámara en coordenadas de píxeles 2D. La parte izquierda de la matriz 3x3 es la matriz intrínseca de la cámara para la imagen rectificada. Los valores fx' , fy' , cx' , cy' pueden diferir de los valores en la matriz K. Esta matriz es esencial en cámaras estéreo para relacionar la posición de ambas cámaras en el sistema estéreo, donde tx , ty indican la posición del centro óptico de la segunda cámara.

$$P = \begin{bmatrix} fx' & 0 & cx' & Tx \\ 0 & fy' & cy' & Ty \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad (5)$$

Es importante señalar que estos parámetros se encuentran dentro del tópico *"/camera_info"*. Para realizar la calibración con el paquete previamente mencionado, se empleó un tablero de ajedrez de tamaño A2 con una disposición de 8x6 intersecciones entre las cuadrículas, y cada una de estas cuadrículas mide 0.058 metros. Una vez que se ha ejecutado el algoritmo de calibración, se generan nuevos parámetros que deben ser actualizados en el mencionado tópico. Para llevar a cabo este proceso, se ha desarrollado un código en Python que publica un nuevo tópico correspondiente a la imagen ya calibrada.

Calibración de cámara y sensor LiDAR

La calibración entre una cámara y un LiDAR es crucial para asegurar una correcta correspondencia entre los datos provenientes de cada sensor. Esta calibración es fundamental para lograr una localización precisa, la detección de objetos, la evasión de obstáculos y una estimación exacta de distancias. Esto, a su vez, conduce a la reducción de errores y al aumento de la confiabilidad en las mediciones y decisiones en tiempo real.

Este proceso de calibración implica ajustar los parámetros extrínsecos, que comprenden la traslación y la rotación relativa entre la cámara y el LiDAR. Estos parámetros son esenciales para establecer una posición y orientación relativa precisa entre ambos sensores, permitiendo una fusión efectiva de datos. A través de un proceso de optimización, se determinan estos parámetros extrínsecos de manera que la representación en el espacio 3D del LiDAR se alinee adecuadamente con las proyecciones de objetos en la imagen de la cámara. Esta correspondencia entre datos 2D y 3D es fundamental para lograr una percepción y localización precisas en aplicaciones como la conducción autónoma, la robótica y la visión por computadora.

Para llevar a cabo esta calibración, se empleará el trabajo propuesto *extend_lidar_camera_calib*, el cual combina dos paquetes: *livox_camera_calib* (lidar_camera_calib, 2021/2023) y *FLOAM* (Han, 2020/2023). El paquete *livox_camera_calib* permite la calibración extrínseca entre la cámara y el sensor LiDAR, utilizando una metodología que extrae bordes tanto de la nube de puntos como de la imagen de la cámara. Para cada voxel obtenido desde la nube de puntos, utiliza RANSAC repetitivamente para extraer planos, mientras que para las imágenes utiliza Canny para extraer los bordes. Luego, este paquete realiza el emparejamiento entre ambos resultados. Por otro lado, el paquete *FLOAM* permite obtener una nube de puntos mucho más densa para lograr un mejor rendimiento del algoritmo.

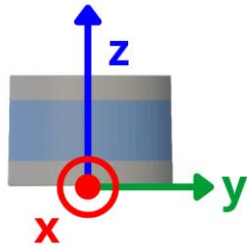
El proceso de calibración requiere seguir una serie de pasos. Se inicia ejecutando el algoritmo de obtención de la nube de puntos en archivos PCD (*Point Cloud Data*), lo cual implica capturar los datos del LiDAR y la imagen correspondiente de la cámara. Estos datos se utilizan para generar una representación tridimensional del entorno. Una vez obtenida esta información, se procede a modificar el archivo *calib.yaml* dentro del directorio designado. En este archivo, se realiza la configuración esencial, como el ajuste de la ruta de los datos para reflejar la ubicación local y la provisión de la matriz intrínseca y los coeficientes de distorsión específicos de la cámara utilizada.

El siguiente paso consiste en ejecutar el algoritmo de calibración. Este proceso es fundamental, ya que a partir de los datos recopilados y la configuración proporcionada, se determinarán los parámetros extrínsecos. En este contexto, la eq. (6) representa la traslación y

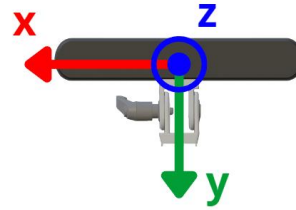
rotación relativa entre la cámara y el LiDAR. Son esenciales para lograr una alineación precisa entre los datos en 3D del LiDAR y las proyecciones de los objetos en la imagen de la cámara.

$$E = \begin{bmatrix} r_{00} & r_{01} & r_{02} & t_x \\ r_{10} & r_{11} & r_{12} & t_y \\ r_{21} & r_{22} & r_{22} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (6)$$

Es importante considerar que el sistema de referencias utilizado por el paquete de calibración para los sensores difiere de los sistemas previamente empleados para las transformadas. Al ejecutar el algoritmo, el sistema de la cámara actúa como sistema de referencia para el sensor LiDAR. Los sistemas de referencia de ambos sensores se muestran en la Figura 7.



(a) Sistema de referencia sensor LiDAR



(b) Sistema de referencia cámara ZED2

Figura 7: Sistema de coordenadas de referencias utilizando por el paquete de calibración disponible para la cámara ZED y sensor LiDAR. Fuente: elaboración propia.

Suponiendo que ambos sensores están orientados hacia el frente, es necesario establecer la siguiente matriz de parámetros extrínsecos para alinear adecuadamente ambos sensores. Esta alineación implica una rotación de 90 grados sobre el eje "x" y una rotación de -90 grados sobre el eje "y", tal como se muestra en la eq. (7).

$$E = \begin{bmatrix} 0.0 & -1.0 & 0.0 & t_x \\ 0.0 & 0.0 & -1.0 & t_y \\ 1.0 & 0.0 & 0.0 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (7)$$

3.3 Algoritmo de mapeo

Los algoritmos de mapeo son fundamentales en robótica y percepción de entornos tridimensionales. Estos algoritmos permiten a los robots explorar y comprender su entorno, lo que es esencial para una amplia gama de aplicaciones, desde la navegación autónoma en

interiores hasta la inspección de sitios peligrosos. A continuación, se describirá el algoritmo de mapeo a utilizar: OctoMap.

OctoMap es una herramienta de mapeo 3D que utiliza una representación octree (árbol octal) para dividir el espacio 3D en pequeños bloques. Esta representación es eficiente en términos de uso de memoria y tiempo de procesamiento, lo que la hace adecuada para aplicaciones de robots móviles (Hornung et al., 2013).

Un octree es una estructura de datos jerárquica que divide el espacio en ocho regiones. Cada región puede ser dividida a su vez en ocho regiones, y así sucesivamente. Esto permite representar el espacio 3D de forma eficiente, ya que solo es necesario almacenar información sobre las regiones que están ocupadas (Hornung et al., 2013).

En OctoMap, cada celda del octree almacena información de ocupación. Esta información puede ser binaria, indicando si la celda está ocupada o no, o puede ser un valor real, indicando la probabilidad de que la celda esté ocupada (Hornung et al., 2013).

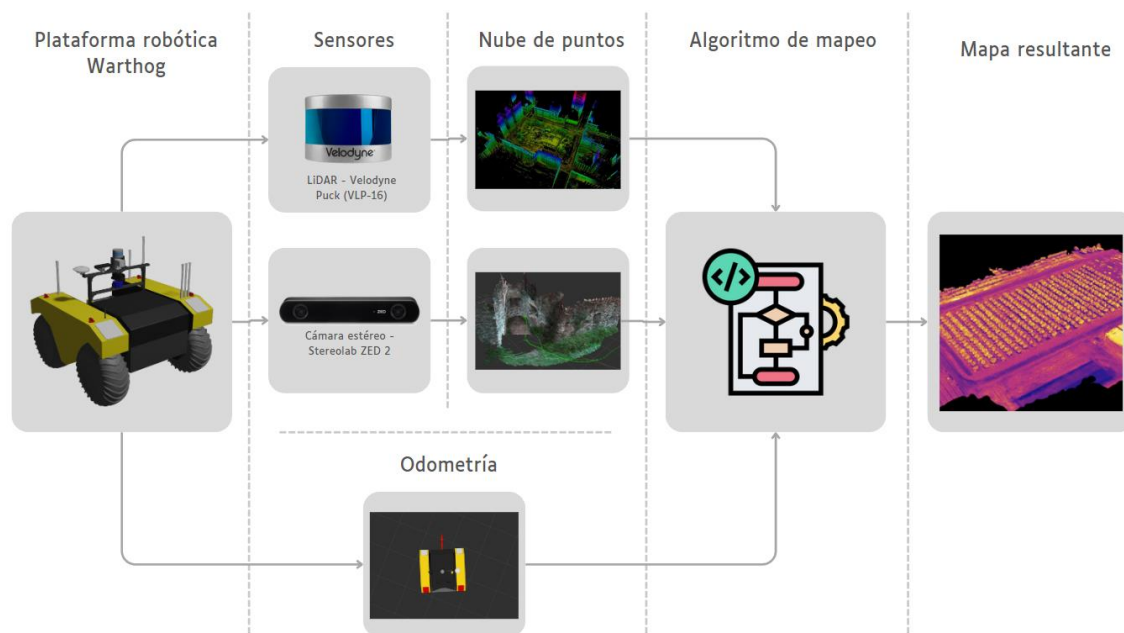


Figura 8: Proceso de adquisición de datos e implementación del algoritmo de mapeo basado en un sensor LiDAR y una cámara ZED de un entorno frutícola. Fuente: elaboración propia.

La implementación del algoritmo OctoMap se basa en la recopilación de datos de sensores que permiten al robot mapear su entorno con gran detalle. En este contexto, el robot recopila información de dos sensores clave: un LiDAR VLP-16 y una cámara RGB-D Zed 2. Específicamente,

el LIDAR VLP-16 captura una nube de puntos en 3D que representa la geometría del entorno, mientras que la cámara estéreo ZED 2 utiliza su visión estereoscópica para capturar imágenes y procesarlas, generando nubes de puntos tridimensionales que representan con precisión la estructura y los elementos del entorno. En la Figura 8 se presentan los pasos necesarios para llevar a cabo la implementación del algoritmo de mapeo.

El algoritmo OctoMap procesa la nube de puntos generada por el LiDAR VLP-16 o la cámara estéreo ZED 2 para construir mapas 3D detallados. La estructura de datos octree se llena con información de ocupación basada en la nube de puntos, permitiendo una representación precisa del entorno y la identificación de áreas ocupadas y libres. Además, el algoritmo toma como referencia el *frame* de odometría del robot, actualizando continuamente la información del mapa de acuerdo al movimiento detectado, lo que garantiza una representación en tiempo real del entorno en constante cambio.

3.4 Evaluación del algoritmo de mapeo

La evaluación del algoritmo de mapeo es crucial para garantizar su precisión, confiabilidad y aplicabilidad. Un análisis proporciona información sobre la capacidad del algoritmo para generar representaciones del entorno, así como su eficiencia en términos de recursos computacionales requeridos.



Figura 9: Diagrama correspondiente a la metodología propuesta para la evaluación del algoritmo de mapeo. Fuente: elaboración propia.

En este capítulo, se presenta el proceso utilizado para evaluar la efectividad y eficiencia del algoritmo de mapeo implementado. La evaluación se centrará en dos aspectos principales: la comparación visual para una evaluación cualitativa y el análisis de los recursos utilizados para una evaluación cuantitativa (ver Figura 9).

La comparación visual implica la comparación directa del mapa generado por el algoritmo con el entorno mapeado. Esta técnica cualitativa permite una evaluación subjetiva pero informativa de la precisión y fidelidad del mapa generado.

El análisis de recursos se centrará en medir y cuantificar los recursos computacionales empleados por el algoritmo. Se registrarán y analizarán métricas como la utilización de memoria y la carga computacional para generar y mantener el mapa. Esto proporcionará una comprensión cuantitativa de la eficiencia del algoritmo en términos de recursos utilizados.

3.5 Herramienta de visualización

La habilitación de una herramienta de visualización para los mapas generados por el algoritmo de mapeo implica una metodología específica que comprende varios pasos. En primer lugar, se realiza una configuración en RVIZ, una herramienta de visualización 3D que permite la representación de información compleja, como modelos de robots, nubes de puntos, mapas y trayectorias planificadas. Esta configuración se personaliza para representar el mapa tridimensional obtenido, lo que posibilita al usuario explorar de manera interactiva y dinámica el entorno mapeado.

Para asegurar la accesibilidad y facilidad de uso, se desarrolla un código en Python que sirve como interfaz entre el usuario y RVIZ. Este código proporciona al usuario la capacidad de seleccionar el mapa específico que desea visualizar mediante interacciones en la consola. Esto facilita la manipulación y visualización de diferentes mapas generados por el algoritmo de mapeo.

Además, para optimizar y simplificar el proceso, se creó un archivo .launch. Este archivo permite abrir automáticamente RVIZ con la configuración previamente establecida para la visualización del mapa 3D. Al mismo tiempo, ejecuta el código Python desarrollado, lo que proporciona una experiencia fluida y automatizada para la selección y visualización de mapas, eliminando la necesidad de realizar múltiples pasos manuales.

Capítulo 4: Resultados

En este capítulo, se expondrán los resultados relevantes obtenidos para cada etapa previamente mencionada y descrita en el apartado de Capítulo 3: Materiales y métodos. Esto incluye los resultados de la calibración y configuración del equipo, la recolección de datos, la implementación, la evaluación y la representación visual del mapeo.

4.1 Calibración y configuración del equipo experimental

Odometría

Se inicio configurando la odometría del robot teniendo en cuenta la metodología propuesta en la sección anterior, para mayor detalle ver Anexo 1: Odometría. Al analizar la odometría obtenida a partir de los datos recopilados en las salidas al terreno, se evidenció un gran error en comparación con el recorrido proporcionado por el GPS. En la primera salida (13-09-2023), no se consideró la configuración previamente propuesta, lo que resultó en la odometría que se muestra en la Figura 10.

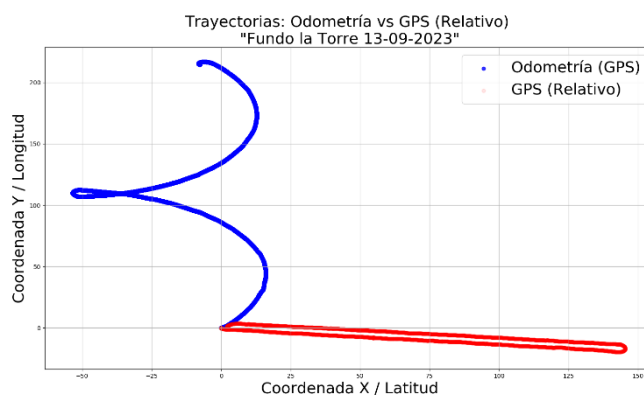
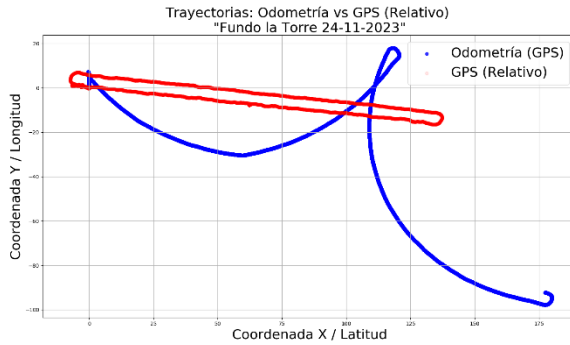


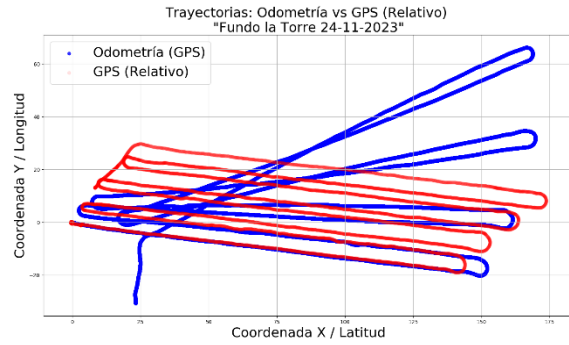
Figura 10: Odometría y trayectoria GPS obtenida con datos de salida a terreno fundo la Torre (13-09-2023). Fuente: elaboración propia.

Por otro lado, en la segunda salida (24-11-2023), se habían implementado los cambios en la configuración de la odometría, dando como resultado las siguientes representaciones (ver Figura 11 (a), (b)).

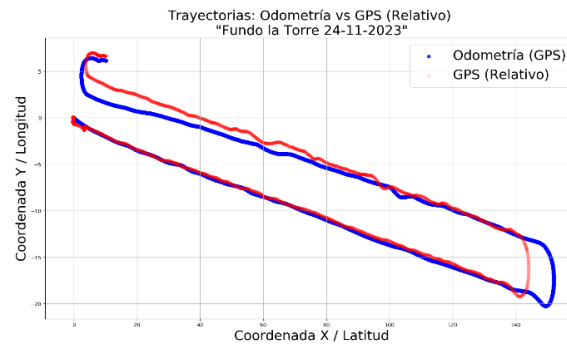
Dado que en el segundo caso (recorrido por varias hileras) la odometría inicial coincidía con el trayecto proporcionado por el GPS, se decidió segmentar esta sección para utilizarla en las futuras implementaciones de algoritmos de mapeo (ver Figura 11 (c)).



(a) Recorrido sobre una hilera



(b) Recorrido sobre múltiples hileras



(c) Extracto de odometría obtenida en mapeo sobre múltiples hileras

Figura 11: Odometría y trayectoria GPS obtenida con datos de salida a terreno fundo la Torre (24-11-2023). Fuente: elaboración propia.

Para abordar este problema, derivado de la dependencia del magnetómetro en la IMU UM7 para calcular el "yaw" y las posibles distorsiones en el campo magnético que pueden introducir errores en las estimaciones de ángulos, se tomó la decisión de implementar una nueva odometría basada exclusivamente en los datos del GPS, como se detalla en la sección de metodología. Este enfoque dio lugar a la creación de una odometría que se visualiza en RVIZ. (ver Figura 12).

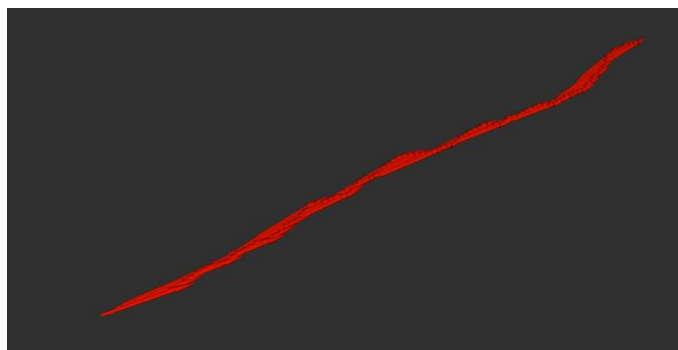


Figura 12: Odometría obtenida con datos de GPS. Fuente: elaboración propia

Configuración “manual”

Para analizar de mejor manera las relaciones y transformaciones existentes entre todos los sensores, se empleó el algoritmo de Python mencionado previamente para generar un diagrama de árbol (ver Figura 36 perteneciente al Anexo 2: Transformadas entre sensores).

Es evidente que todos los sensores están directamente relacionados con "sensor_arch", que es el arco construido de metal que sostiene a todos los sensores. En otras palabras, todos los sensores tienen como punto de origen para su posicionamiento el punto de origen del arco de metal, como se puede observar en Figura 36.

Una vez que se obtuvo una comprensión más clara de las transformaciones existentes, se procedió a analizar las traslaciones y rotaciones con respecto al arco de metal, tal como se describe en el archivo XML, y se obtuvieron los resultados presentes en la Tabla 4.

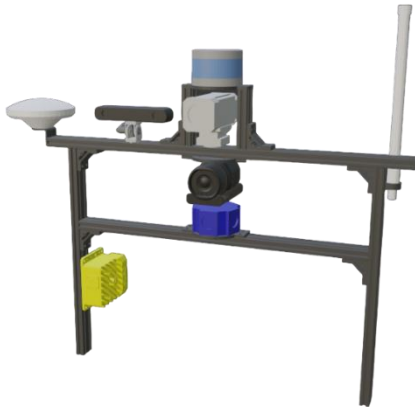
Tabla 4: Parámetros de transformadas de sensores con respecto a estructura de soporte. Fuente: elaboración propia.

Sensor	Traslación (x y z) [m]			Rotación (roll pitch yaw) [°]		
Antena	-0,05579	0,35807	0,21	0	0	0
Swiftnav Antena	-0,02	-0,425	0,29816	0	0	0
Duro RTK	-0,0525	-0,326075	-0,06471	0	0	0
Blackfly S	-0,02	-0,008	0,31809	0	0	0
ZED 2	-0,01	-0,188	0,339	0	0	0
VLP-16	-0,02	0	0,40451	0	0	0
FLIR A400	0,0038	0	0,21328	0	0	90
IMU UM7	0.227	0	-0.02	0	0	-90

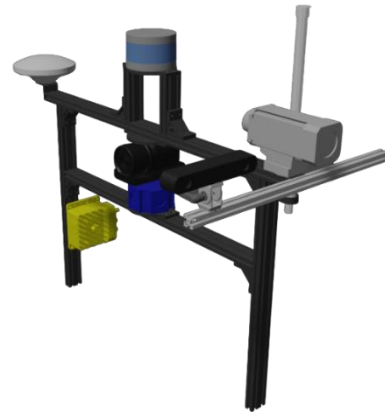
Con el fin de obtener una representación visual digital de la disposición de los sensores, se empleó el software BLENDER para importar modelos 3D de cada sensor y mostrar las traslaciones y rotaciones mencionadas anteriormente. Se llevó a cabo una modificación en el arreglo de sensores para que algunos de estos estén dirigidos hacia el lado del robot, permitiendo así la recolección directa de información de los cultivos, específicamente de la cámara ZED 2 y Blackfly S, cuyas traslaciones y rotaciones se detallan en la Tabla 5. Esta modificación conlleva la instalación de una nueva estructura que se acopla a la existente. La Figura 13 muestra tanto el arreglo original de sensores como el modificado.

*Tabla 5: Parámetros de modificación de transformadas de sensores con respecto a estructura de soporte.
Fuente: elaboración propia*

Sensor	Traslación (x y z) [m]			Rotación (roll pitch yaw) [°]		
ZED 2	0.1765	0.41011	0.27816	0	0	90
Blackfly S	-0.01	0.37191	0.27535	0	0	90



(a) Configuración inicial de sensores



(b) Configuración de sensores modificada

Figura 13: Configuración inicial y modificada de distribución de sensores montados en la plataforma Warthog. Fuente: elaboración propia.

Teniendo en cuenta esta modificación, se procedió a incorporarla en la simulación para validar el diseño y el comportamiento del robot con esta nueva configuración (ver Figura 14).



Figura 14: Simulación de plataforma robótica Warthog en software Gazebo con modificación implementada. Fuente: elaboración propia.

Es importante destacar que, al tener las traslaciones y rotaciones de los sensores en Blender, es mucho más sencillo realizar modificaciones y visualizarlas en tiempo real en comparación con la modificación del archivo XML.

Al analizar la imagen con la nube de puntos integrada en RVIZ en la Figura 15, se evidencia que la nueva configuración de sensores dificulta la obtención de información por parte del sensor LiDAR. Esto sugiere que esta configuración no es la más adecuada, lo que dificulta la aplicación de diversos algoritmos.

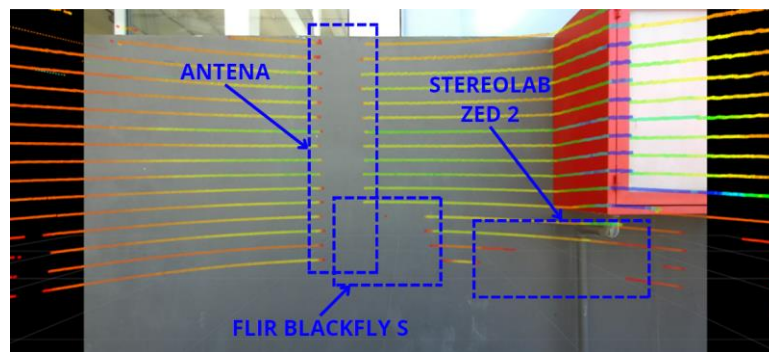


Figura 15: Imagen de cámara estéreo ZED 2 con nube de puntos de sensor LiDAR VLP-16 Fuente: elaboración propia.

Calibración de cámara

La calibración de la cámara se realizó utilizando el paquete *camera_calibration* de ROS. El tópicos utilizado para la calibración corresponde a la imagen RGB proporcionada por la cámara ZED 2, y su nombre es `"/zed2_node/rgb/image_rect_color"`. Para utilizar este paquete, se debe ejecutar el siguiente comando en la consola:

```
Terminal
user@host:~$ rosruncamera_calibration cameracalibrator.py --approximate=0.1 --size 8x6
--square 0.058 image:=/zed2_node/rgb/image_rect_color
```

Figura 16: Comando para ejecutar la calibración de la cámara. Fuente: elaboración propia.

Una vez ejecutado este comando, se abrirá una ventana que captura imágenes del tablero de ajedrez 8x6 desde diversas posiciones y ángulos hasta que se activa la opción que permite calibrar la cámara.



Figura 17: Captura de pantalla correspondiente al algoritmo de calibración. Fuente: elaboración propia.

Cuando se presiona el botón de calibración, el paquete proporcionará los nuevos parámetros de la cámara calibrada, que son los siguientes:

- *Camera matrix (K):*

$$K = \begin{bmatrix} 1085.131537 & 0.000000 & 1149.659319 \\ 0.000000 & 1081.897829 & 642.082505 \\ 0.000000 & 0.000000 & 1.000000 \end{bmatrix} \quad (8)$$

- *Distortion (D):*

$$D = [-0.006433 \quad 0.018901 \quad -0.001205 \quad 0.000886 \quad 0.000000] \quad (9)$$

- *Rectification (R):*

$$R = \begin{bmatrix} 1.000000 & 0.000000 & 0.000000 \\ 0.000000 & 1.000000 & 0.000000 \\ 0.000000 & 0.000000 & 1.000000 \end{bmatrix} \quad (10)$$

- *Projection (P):*

$$P = \begin{bmatrix} 1112.624146 & 0.000000 & 1148.601028 & 0.000000 \\ 0.000000 & 1108.668091 & 638.554862 & 0.000000 \\ 0.000000 & 0.000000 & 1.000000 & 0.000000 \end{bmatrix} \quad (11)$$

Una vez que se obtienen estos nuevos parámetros, se debe crear un nuevo tópico que corresponda a la cámara calibrada. Esto se logra utilizando el código de Python creado para este propósito. A continuación, en la Figura 18 se muestra una imagen de la cámara calibrada.



Figura 18: Imagen obtenida luego de realizar la calibración. Fuente: elaboración propia.

Calibración de cámara y sensor LiDAR

Para llevar a cabo la calibración entre la cámara y el sensor LiDAR, el primer paso es ejecutar el algoritmo para obtener la nube de puntos a través de un comando en la consola. Este procedimiento puede realizarse con el siguiente comando:

```
Terminal
user@host:~$ roslaunch floam floam_mapping.launch
```

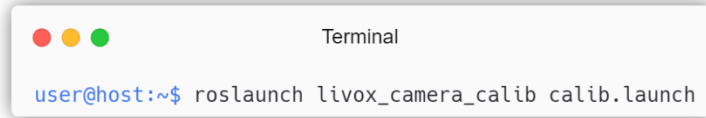
Figura 19: Comando para ejecutar la calibración entre la cámara y el sensor LiDAR. Fuente: elaboración propia.

Una vez que se haya ejecutado este algoritmo, la generación de la nube de puntos comenzará y continuará hasta que el usuario interrumpa el proceso desde la misma consola. Sin embargo, antes de interrumpirlo, es crucial asegurarse de haber guardado la imagen correspondiente a la cámara que se desea calibrar. En este caso, se trata de la cámara estéreo ZED 2. Ambos archivos se deben guardar en la carpeta PCD del mismo paquete.

Posteriormente, se procede a configurar el algoritmo de calibración extrínseca. Esto implica proporcionar la ruta de los archivos mencionados anteriormente, los parámetros de la cámara obtenidos en la sección previa y la matriz de parámetros extrínsecos que se estimaron inicialmente. Es importante destacar que el paquete asume que ambos sensores están apuntando en la misma dirección, por lo que la nube de puntos se rota 90 grados para alinearse con la dirección de la cámara. La matriz inicial es la siguiente:

$$E = \begin{bmatrix} 0.0 & -1.0 & 0.0 & -0.1965 \\ 0.0 & 0.0 & -1.0 & -0.12365 \\ 1.0 & 0.0 & 0.0 & -0.41011 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (12)$$

Una vez que todo esté configurado, se debe ejecutar el algoritmo de calibración entre la cámara y el sensor LiDAR mediante el siguiente comando en la consola:



```
Terminal
user@host:~$ roslaunch livox_camera_calib calib.launch
```

Figura 20: Comando para ejecutar la calibración entre la cámara y el sensor LiDAR. Fuente: elaboración propia.

El algoritmo realizará automáticamente todos los cálculos necesarios y abrirá una ventana de RVIZ que mostrará los voxels obtenidos de la nube de puntos, así como los planos identificados. Además, se abrirán ventanas donde se realizará el match entre la extracción de bordes de la imagen y los planos de la nube de puntos.

Finalmente, como resultado, se generará un archivo .txt que contendrá la matriz de parámetros extrínsecos resultantes:

$$E = \begin{bmatrix} -0.0698172 & -0.99756 & 0.000516412 & -0.0191322 \\ -0.15612 & 0.0104152 & -0.987683 & 0.161197 \\ 0.985268 & -0.0690379 & -0.156466 & -0.45126 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (13)$$

Es importante notar que los resultados de la rotación se expresan en una orientación de $[-156.19154358, -80.15287841, -114.0941451]$ grados, lo cual puede parecer algo inusual dada la disposición de los sensores. En cuanto a los resultados de traslación, muestran valores muy cercanos a los ya existentes, lo que facilita una correcta calibración extrínseca entre ambos sensores. Es esencial señalar que estos resultados podrían haber sido afectados por la pérdida de información de la nube de puntos, como se muestra en la Figura 15.

4.2 Adquisición de datos

En el presente capítulo, se aborda la importancia de la adquisición de datos provenientes de los sensores de la plataforma robótica para diversos propósitos y terrenos. Se detalla cómo cada sensor cumple un rol específico en la obtención de datos particulares, destacando la

relevancia de las cámaras para imágenes de alta resolución, el LiDAR para información tridimensional y el GPS para precisión de localización. Para obtener más detalles sobre las salidas realizadas y los datos recopilados, dirigirse al Anexo 3: Adquisición de datos, donde se encuentran tablas y figuras que ilustran la información obtenida en dos salidas distintas: el 13 de septiembre de 2023 y el 24 de noviembre del mismo año en Fundo La Torre, Rengo-Chile.

4.3 Algoritmo de mapeo

Inicialmente, con el propósito de evaluar el algoritmo de mapeo propuesto, se optó por su implementación en un entorno de simulación (ver Figura 21). En un principio, OctoMap se pensó exclusivamente para la nube de puntos generada por el sensor LiDAR VLP-16. Sin embargo, se decidió analizar el comportamiento de la nube de puntos proveniente de la cámara estéreo. Para llevar a cabo esto, se proporcionaron como entrada al algoritmo de mapeo la odometría del robot y los datos capturados por la cámara, además de configurar un rango de distancia mínimo de 1 metro y máximo de 5 metros para el procesamiento de los datos.

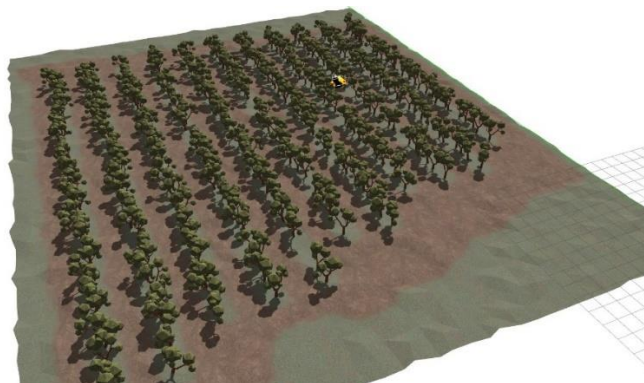


Figura 21: Simulación en Gazebo. Fuente: elaboración propia.

Una vez realizados estos ajustes y con la simulación en funcionamiento, se procedió a ejecutar el algoritmo de mapeo mediante el siguiente comando:

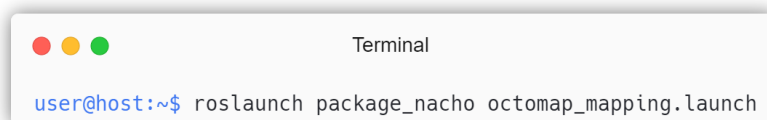


Figura 22: Comando para ejecutar el algoritmo de mapeo OctoMap. Fuente: elaboración propia.

Es importante destacar que los mapas generados tienen la capacidad de ser guardados en el espacio de trabajo mediante la ejecución del siguiente comando:

```
Terminal
user@host:~$ roslaunch octomap_server octomap_saver -f name.bt
```

Figura 23: Comando para guardar el mapeo realizado. Fuente: elaboración propia.

La ejecución del algoritmo permitió la visualización en tiempo real del proceso de mapeo a través del software RVIZ. A continuación, en la Figura 24 se presenta una captura de pantalla de la simulación y el mapa que se está generando en tiempo real.

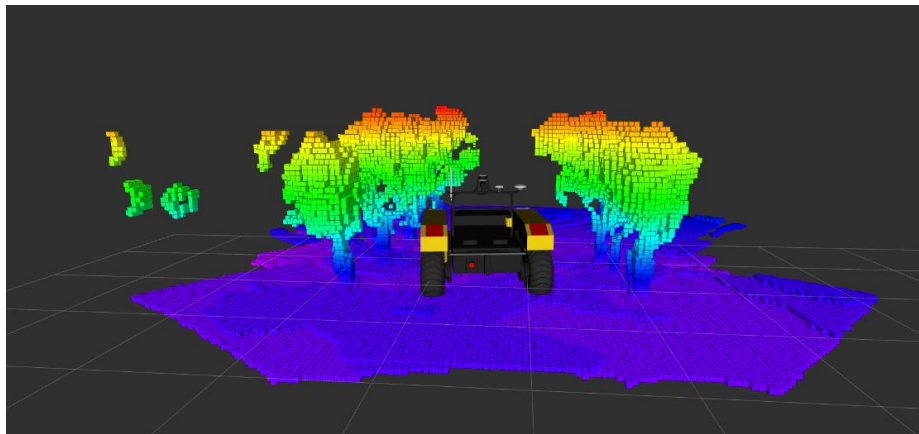
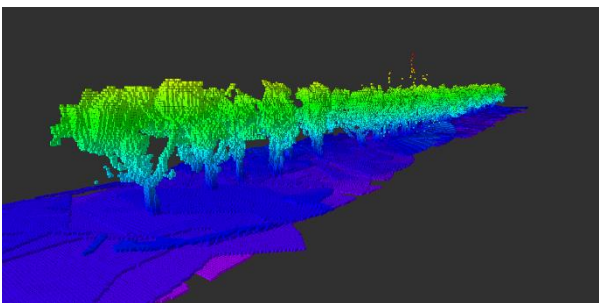
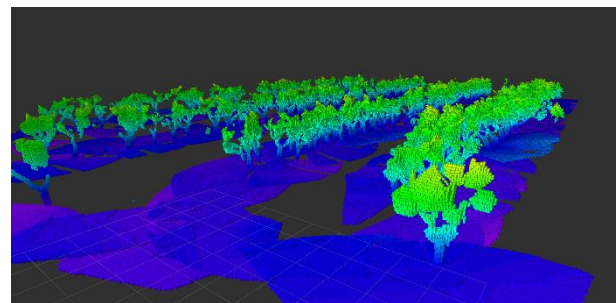


Figura 24: Generación de mapa en tiempo real con OctoMap, con datos de cámara estéreo (Simulación). Fuente: elaboración propia.

Es importante mencionar que los recorridos realizados para la generación de mapas con la cámara estéreo fueron idénticos a los llevados a cabo en las salidas a terreno (para ver la odometrías utilizadas, ver Figura 39 del Anexo 4: Gráficos de odometría utilizada en mapeo). Esto se hizo para comparar el comportamiento en simulación y con datos reales, los mapas generados pueden ser visualizados en la Figura 25.



(a) Mapeo sobre una hilera



(b) Mapeo sobre múltiples hileras

Figura 25: Experimentos de mapeo en un entorno frutícola, llevados a cabo mediante la utilización de datos de una cámara estéreo en un entorno de simulación. Fuente: elaboración propia.

Como se mencionó anteriormente, el algoritmo se pensó inicialmente para trabajar con los datos del sensor LiDAR. Dado esto, se realizó una adaptación del mismo algoritmo, utilizando la nube de puntos generada por el sensor LiDAR, sin modificar la configuración de distancias mínima y máxima para el procesamiento de los datos. Nuevamente, se ejecutó el algoritmo mediante el comando mencionado, lo que llevó a los siguientes resultados presentes en la Figura 26.

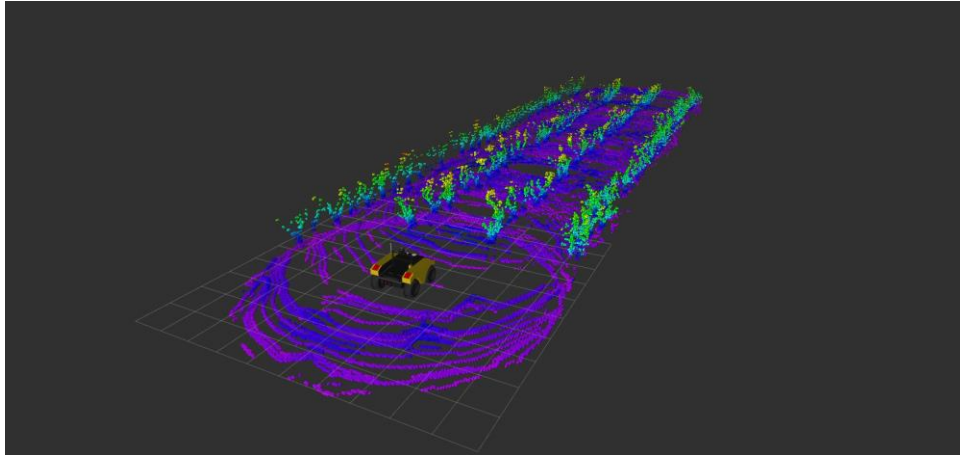
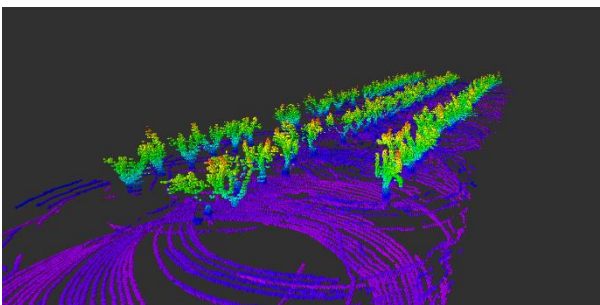
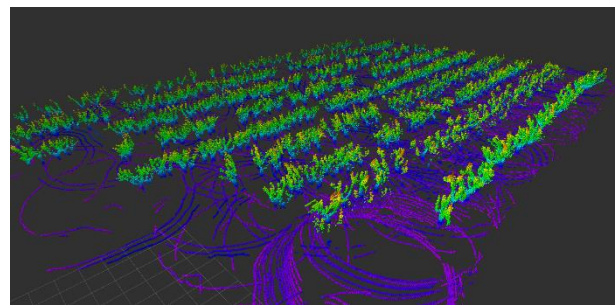


Figura 26: Generación de un mapa en un entorno frutícola simulado con el algoritmo OctoMap, utilizando datos de un sensor LiDAR. Fuente: elaboración propia.

Al igual que con la cámara estéreo, los recorridos para la generación de mapas con el sensor LiDAR fueron los mismos que los realizados en terreno (para ver la odometrías utilizadas, ver Figura 40 del Anexo 4: Gráficos de odometría utilizada en mapeo). Para visualizar los mapas generados, dirigirse a la Figura 27.



(a) Mapeo sobre una hilera



(b) Mapeo sobre múltiples hileras

Figura 27: Generación de mapas utilizando información del sensor LiDAR en un entorno frutícola simulado. Fuente: elaboración propia.

Una vez que el algoritmo funcionaba correctamente en la simulación, se procedió a ejecutar el mapeo con los datos obtenidos en las salidas a terreno. Para esto, se configuró inicialmente el paquete para utilizar la nube de puntos proveniente del tópico correspondiente, así como la odometría del robot. Sin embargo, como se mencionó anteriormente, la odometría presentaba problemas en la representación del movimiento del robot, lo que generó el mapa que se puede visualizar en la Figura 28 (para ver los demás mapas obtenidos, consultar Figura 42 del Anexo 5: Mapas obtenidos con odometría inicial).

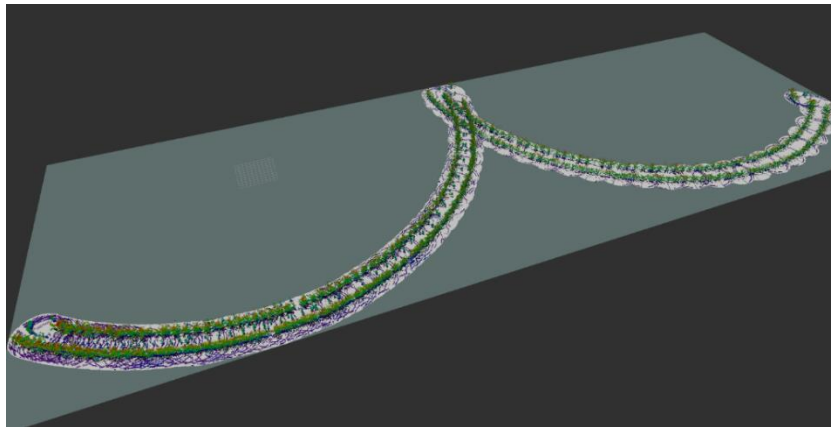
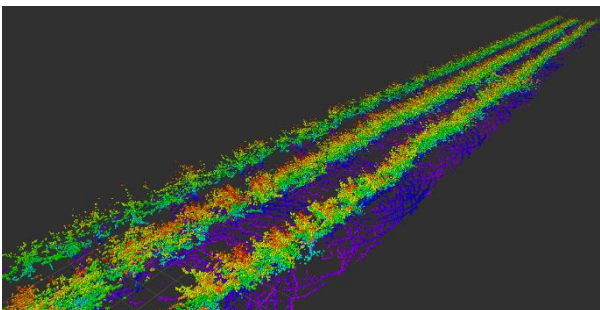
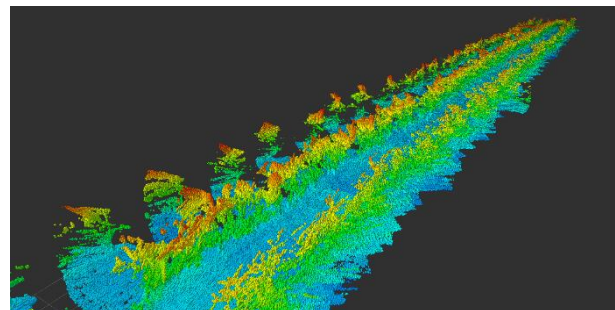


Figura 28: Generación de mapa frutícola a partir de datos obtenidos durante la primera salida a terreno llevado a cabo en el Fundo La Torre. Fuente: elaboración propia.

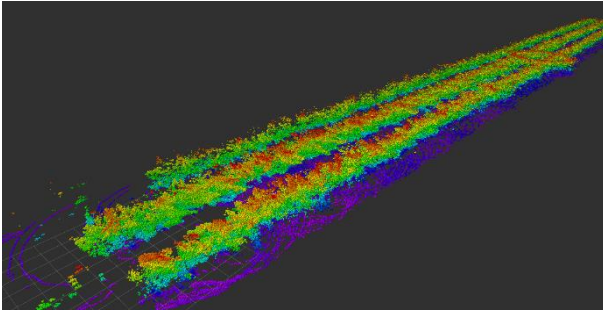
Debido a esto, se recurrió a la odometría generada a partir de los datos del GPS para implementar el algoritmo de mapeo con los datos reales. En otras palabras, el algoritmo tomó como entrada la nube de puntos proveniente de la cámara estéreo ZED 2 o del LiDAR VLP-16 junto con esta nueva odometría. Los mapas obtenidos con los datos de ambas salidas a terreno se pueden visualizar en la Figura 29 (para ver las odometías utilizadas ver Figura 41 del Anexo 4: Gráficos de odometría utilizada en mapeo).



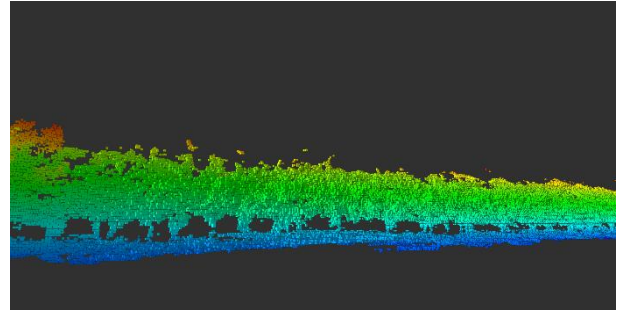
(a) Mapeo sobre una hilera con sensor LiDAR (13-09-2023)



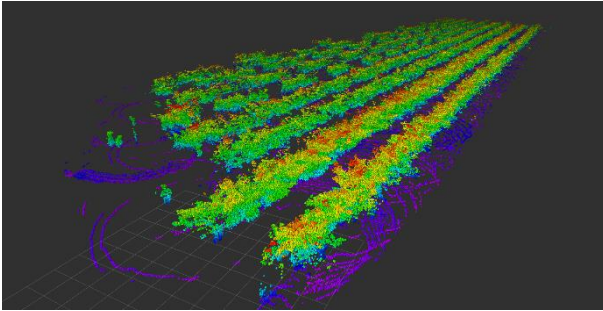
(b) Mapeo sobre una hilera con cámara ZED 2 (13-09-2023)



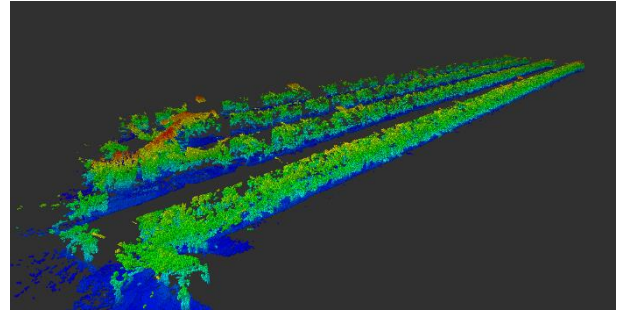
(c) Mapeo sobre una hilera con sensor LiDAR (24-11-2023)



(d) Mapeo sobre una hilera con cámara ZED 2 (24-11-2023)



(e) Mapeo sobre múltiples hileras con sensor LiDAR (24-11-2023)



(f) Mapeo sobre múltiples hileras con cámara ZED 2 (24-11-2023)

Figura 29: Generación de mapas de un entorno frutícola, contemplando diversas trayectorias del robot Warthog. Fuente: elaboración propia.

4.4 Evaluación del algoritmo de mapeo

La evaluación del algoritmo comienza con una comparación visual de los mapas generados, ya sea a partir del entorno real o simulado. Esta evaluación se segmentará en dos partes: la del sensor LiDAR y la de la cámara estéreo. Es importante tener en cuenta que los mapas resultantes están directamente influenciados por la precisión de la odometría vinculada al algoritmo de mapeo. En otras palabras, si la odometría contiene ciertos errores, los mapas no representarán de manera óptima el entorno mapeado, tal como se muestra en la Figura 28.

Los mapas generados a través del sensor LiDAR, ya sea en entornos simulados o reales, muestran una gran correspondencia entre el entorno y los mapas obtenidos, logrando una representación digital precisa del mismo.

En entornos simulados, la cámara estéreo genera una nube de puntos clara y precisa en el mapa (Figura 25). Sin embargo, en entornos reales, el movimiento variable del entorno genera un nivel considerable de ruido en la nube de puntos. Esta situación puede introducir información

falsa en el mapa, complicando la visualización precisa del entorno mapeado. Por ende, se necesita más trabajo en la captura y procesamiento de la nube de puntos de la cámara estéreo ZED 2.

Para obtener una evaluación cuantitativa del rendimiento del algoritmo de mapeo, se empleó la herramienta “Htop”, que permite monitorear los recursos del sistema. Se extrajeron los porcentajes de uso de la RAM y la CPU mientras el algoritmo de mapeo se ejecutaba, dividiendo estos datos en dos tablas: una para el sensor LiDAR (ver Tabla 6) y otra para la cámara estéreo (ver Tabla 7). (Es importante destacar que estos datos se obtienen a partir del promedio de los datos presentados en la Tabla 9 y Tabla 10 del Anexo 6: Métricas para evaluación cuantitativa).

Tabla 6: Métricas obtenidas al realizar mapeo con sensor LiDAR. Fuente: elaboración propia.

Datos reales						Simulación			
TOMA 1 (13-09-2023)		TOMA 1 (24-11-2023)		TOMA 2 (24-11-2023)		TOMA 1		TOMA 2	
RAM	CPU	RAM	CPU	RAM	CPU	RAM	CPU	RAM	CPU
6,1	100,8	4,6	100,8	7,3	101,3	2,1	102,1	4,6	102,6

Tabla 7: Métricas obtenidas al realizar mapeo con cámara estéreo ZED 2. Fuente: elaboración propia.

Datos reales						Simulación			
TOMA 1 (13-09-2023)		TOMA 1 (24-11-2023)		TOMA 2 (24-11-2023)		TOMA 1		TOMA 2	
RAM	CPU	RAM	CPU	RAM	CPU	RAM	CPU	RAM	CPU
4,0	100,8	3,3	97,3	3,8	98,1	2,0	113,3	3,2	111,2

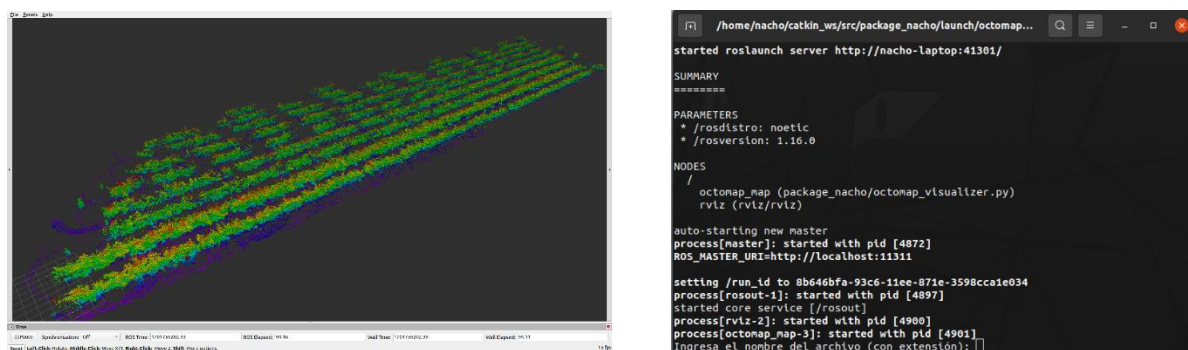
Al realizar de estos datos se obtienen diferencias en el consumo de recursos entre los dos tipos de sensores utilizados: el LiDAR y la cámara estéreo. Con el LiDAR, se observa que, en promedio, el consumo de RAM es menor en datos simulados (3,3%) en comparación con datos reales (6%), mientras que la CPU tiende a tener un uso mayor con datos reales (100,9%) que en simulaciones (102,4%).

En el caso de la cámara estéreo, los resultados muestran un consumo promedio de RAM más bajo en ambos entornos, con un 2,6% en simulaciones y un 3,7% en datos reales. Sin embargo, la CPU tiende a tener un uso más alto en simulaciones (112,2%) que en datos reales (98,7%).

El uso de CPU que excede el 100% se debe a que la medición se realiza en relación con un solo núcleo del procesador. Por lo tanto, cuando un proceso utiliza más del 100% de un núcleo, esto se refleja en un porcentaje de uso mayor al 100% en la monitorización de la CPU.

Una vez que los mapas generados mediante el algoritmo de mapeo han sido almacenados, es posible visualizarlos utilizando el archivo .launch creado. Para ejecutar este archivo desde la consola, se emplea el siguiente comando:

Figura 30: Comando para ejecutar la herramienta de visualización de los mapas. Fuente: Elaboración propia.



(a) Ventana de RVIZ con mapa cargado

Figura 31: Ejecucion de herramienta de visualización habilitada. Fuente: elaboración propia.

Capítulo 5: Conclusión y trabajos futuros

5.1 Conclusión

En este trabajo de título, se ha realizado la implementación de un algoritmo para mapear entornos frutícolas utilizando la plataforma robótica móvil Warthog junto con una variedad de sensores integrados. Para lograrlo, se llevaron a cabo una serie de procedimientos fundamentales que han contribuido a los resultados obtenidos, siendo la calibración un paso crucial.

Como se mencionó previamente, la odometría desempeña un papel fundamental en la generación de mapas, ya que constituye una entrada crucial para el algoritmo. La discrepancia identificada entre la odometría visualizada y el movimiento real destaca la necesidad de ajustar los parámetros de la odometría para obtener resultados confiables en el mapeo frutícola con robots. Por ello, se implementó una nueva odometría basada exclusivamente en datos del GPS.

La disposición y la relación entre los sensores tienen un impacto significativo en la aplicación de diversos algoritmos, lo que subraya la importancia de integrar estas modificaciones con precisión en los archivos descriptivos del robot para lograr una calibración extrínseca correcta entre sensores. Además, es crucial realizar un análisis detallado al modificar esta disposición para evitar afectar futuras aplicaciones.

La calibración de la cámara es fundamental para garantizar la precisión de las imágenes, lo que permite obtener datos visuales precisos y confiables en entornos de investigación y desarrollo tecnológico. De igual manera, la calibración entre la cámara y el sensor LiDAR ha sido un paso significativo para integrar coherentemente estos dispositivos, aunque se destaca la necesidad de abordar posibles pérdidas de información que podrían afectar la precisión en análisis futuros.

La adquisición de datos de los sensores de la plataforma robótica ha sido fundamental para trabajar con estos datos en la implementación *offline* del algoritmo de mapeo. Además, estos datos quedan disponibles como base para posibles trabajos futuros.

Finalmente, las secciones dedicadas al algoritmo OctoMap han demostrado su capacidad para generar mapas en entornos frutícolas, tanto en simulaciones como en situaciones reales. Estos mapas son esenciales para proporcionar una representación visual detallada de los campos,

siendo herramientas fundamentales para que los productores comprendan patrones y tomen decisiones estratégicas en la gestión de cultivos. Sin embargo, se destaca la importancia de abordar la precisión de la odometría para obtener representaciones más precisas del entorno. Lograr un seguimiento preciso del movimiento y realizar un procesamiento más exhaustivo de los datos de la cámara estéreo en entornos reales representan desafíos cruciales en la generación de estos mapas. La complejidad de las condiciones del mundo real subraya la necesidad de una calibración cuidadosa y ajuste de los algoritmos para garantizar resultados confiables. Este enfoque meticuloso es esencial para maximizar la utilidad de los mapas generados en entornos frutícolas, contribuyendo al avance continuo de la agricultura de precisión.

5.2 Trabajos futuros

En base a los resultados obtenidos en el presente trabajo de título, se proponen las siguientes áreas para posibles investigaciones futuras:

- Realizar un estudio detallado de la configuración de la odometría en la plataforma robótica móvil Warthog para obtener una representación precisa del movimiento, especialmente en entornos techados y considerando las limitaciones del GPS.
- Investigar estrategias para mejorar la disposición y alineación de los sensores en la plataforma robótica, con el objetivo de reducir pérdidas de información durante la calibración de la cámara y el sensor LiDAR y mejorar la captura y precisión de datos.
- Implementar algoritmos de mapeo que utilicen datos RGB para generar mapas visuales del entorno mapeado, permitiendo un análisis más detallado y profundo, complementando la información tridimensional con datos visuales.

Estas propuestas de trabajos futuros tienen como objetivo expandir y mejorar las capacidades del presente trabajo. Se enfocan en áreas específicas como la configuración de la odometría, la optimización en la disposición de sensores y la implementación de algoritmos de mapeo más completos, los cuales incluyan información visual del entorno.

Cabe destacar que los repositorios de trabajo utilizados tanto en la simulación como en la aplicación real se encuentran disponibles en el siguiente enlace: [Documentación](#).

Referencias

- Alatise, M. B., & Hancke, G. P. (2020). A Review on Challenges of Autonomous Mobile Robot and Sensor Fusion Methods. En *IEEE Access* (Vol. 8, pp. 39830–39846). Institute of Electrical and Electronics Engineers Inc. <https://doi.org/10.1109/ACCESS.2020.2975643>
- Botta, A., Cavallone, P., Baglieri, L., Colucci, G., Tagliavini, L., & Quaglia, G. (2022). A Review of Robots, Perception, and Tasks in Precision Agriculture. En *Applied Mechanics* (Vol. 3, Número 3, pp. 830–854). MDPI. <https://doi.org/10.3390/applmech3030049>
- Chen, M., Tang, Y., Zou, X., Huang, Z., Zhou, H., & Chen, S. (2021). 3D global mapping of large-scale unstructured orchard integrating eye-in-hand stereo vision and SLAM. *Computers and Electronics in Agriculture*, 187. <https://doi.org/10.1016/j.compag.2021.106237>
- Foote, T. (s. f.). *tf: The Transform Library*.
- Gil, G., Casagrande, D. E., Cortés, L. P., & Verschae, R. (2023). Why the low adoption of robotics in the farms? Challenges for the establishment of commercial agricultural robots. *Smart Agricultural Technology*, 3. <https://doi.org/10.1016/j.atech.2022.100069>
- Han, B., Wei, J., Zhang, J., Meng, Y., Dong, Z., & Liu, H. (2023). GardenMap: Static point cloud mapping for Garden environment. *Computers and Electronics in Agriculture*, 204. <https://doi.org/10.1016/j.compag.2022.107548>
- Han, Wang (2023). *FLOAM* [C++]. <https://github.com/wh200720041/floam> (Obra original publicada en 2020)
- Hornung, A., Wurm, K. M., Bennewitz, M., Stachniss, C., & Burgard, W. (2013). OctoMap: An efficient probabilistic 3D mapping framework based on octrees. *Autonomous Robots*, 34(3), 189–206. <https://doi.org/10.1007/s10514-012-9321-0>
- Informe de Cuentas Nacionales de Chile primer trimestre de 2023—Banco Central de Chile*. (s. f.). Recuperado 23 de septiembre de 2023, de <https://www.bcentral.cl/contenido/-/detalle/informe-de-cuentas-nacionales-de-chile-primer-trimestre-de-2023>
- Kamegawa, T., Akiyama, T., Sakai, S., Fujii, K., Une, K., Ou, E., Matsumura, Y., Kishutani, T., Nose, E., Yoshizaki, Y., & Gofuku, A. (2020). Development of a separable search-and-rescue robot

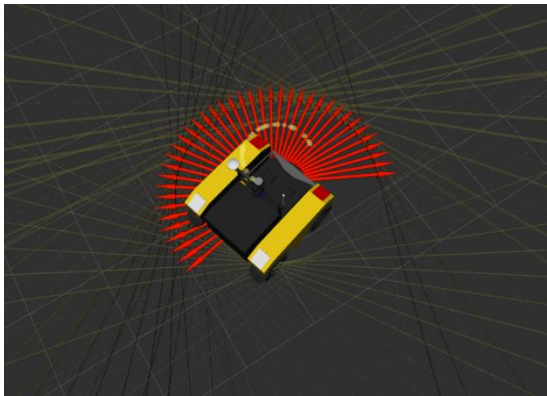
- composed of a mobile robot and a snake robot. *Advanced Robotics*, 34(2), 132–139. <https://doi.org/10.1080/01691864.2019.1691941>
- Kang, H., & Chen, C. (2019). *Visual Perception and Modelling in Unstructured Orchard for Apple Harvesting Robots*. <https://doi.org/10.1109/ACCESS.2020.2984556>
- Kim, W. S., Lee, D. H., Kim, Y. J., Kim, T., Lee, W. S., & Choi, C. H. (2021). Stereo-vision-based crop height estimation for agricultural robots. *Computers and Electronics in Agriculture*, 181. <https://doi.org/10.1016/j.compag.2020.105937>
- HKU-Mars-Lab. (2023). *Lidar_camera_calib* [C++]. https://github.com/hku-mars/livox_camera_calib (Obra original publicada en 2021)
- Lozano-Povis, A., Alvarez-Montalván, C. E., & Moggiano, N. (2021). Climate change in the Andes and its impact on agriculture: a systematic review. En *Scientia Agropecuaria* (Vol. 12, Número 1, pp. 101–108). Universidad Nacional de Trujillo. <https://doi.org/10.17268/SCI.AGROPECU.2021.012>
- Monteiro, A., Santos, S., & Gonçalves, P. (2021). Precision agriculture for crop and livestock farming—Brief review. En *Animals* (Vol. 11, Número 8). MDPI AG. <https://doi.org/10.3390/ani11082345>
- Pawlak, K., & Kołodziejczak, M. (2020). The role of agriculture in ensuring food security in developing countries: Considerations in the context of the problem of sustainable food production. *Sustainability (Switzerland)*, 12(13). <https://doi.org/10.3390/su12135488>
- Rahman, M., Liu, H., Cardenas, I. D., Starr, A., Hall, A., & Anderson, R. (2021). Towards an Autonomous RIRS: Design, Structure Investigation and Framework. *2021 7th International Conference on Mechatronics and Robotics Engineering, ICMRE 2021*, 164–168. <https://doi.org/10.1109/ICMRE51691.2021.9384846>
- Rivera, G., Porras, R., Florencia, R., & Sánchez-Solís, J. P. (2023). LiDAR applications in precision agriculture for cultivating crops: A review of recent advances. En *Computers and Electronics in Agriculture* (Vol. 207). Elsevier B.V. <https://doi.org/10.1016/j.compag.2023.107737>

- Rovira-Más, F., Zhang, Q., & Reid, J. F. (2008). Stereo vision three-dimensional terrain maps for precision agriculture. *Computers and Electronics in Agriculture*, 60(2), 133–143. <https://doi.org/10.1016/j.compag.2007.07.007>
- Szrek, J., Jakubiak, J., & Zimroz, R. (2022). A Mobile Robot-Based System for Automatic Inspection of Belt Conveyors in Mining Industry. *Energies*, 15(1). <https://doi.org/10.3390/en15010327>
- Vieira, D., Orjuela, R., Spisser, M., & Basset, M. (2022). Positioning and Attitude determination for Precision Agriculture Robots based on IMU and Two RTK GPSs Sensor Fusion. *IFAC-PapersOnLine*, 55(32), 60–65. <https://doi.org/10.1016/j.ifacol.2022.11.115>
- Wang, B. H., Diaz-Ruiz, C., Banfi, J., & Campbell, M. (2021). Detecting and Mapping Trees in Unstructured Environments with a Stereo Camera and Pseudo-Lidar. *Proceedings – IEEE International Conference on Robotics and Automation*, 2021–May, 14120–14126. <https://doi.org/10.1109/ICRA48506.2021.9562056>
- Wang, H., Zhang, C., Song, Y., Pang, B., & Zhang, G. (2020). Three-Dimensional Reconstruction Based on Visual SLAM of Mobile Robot in Search and Rescue Disaster Scenarios. *Robotica*, 38(2), 350–373. <https://doi.org/10.1017/S0263574719000675>
- Warthog Unmanned Ground Vehicle Robot—Clearpath. (s. f.). *Clearpath Robotics*. Recuperado 19 de noviembre de 2023, de <https://clearpathrobotics.com/warthog-unmanned-ground-vehicle-robot/>
- Xiong, J., Liang, J., Zhuang, Y., Hong, D., Zheng, Z., Liao, S., Hu, W., & Yang, Z. (2023). Real-time localization and 3D semantic map reconstruction for unstructured citrus orchards. *Computers and Electronics in Agriculture*, 213. <https://doi.org/10.1016/j.compag.2023.108217>

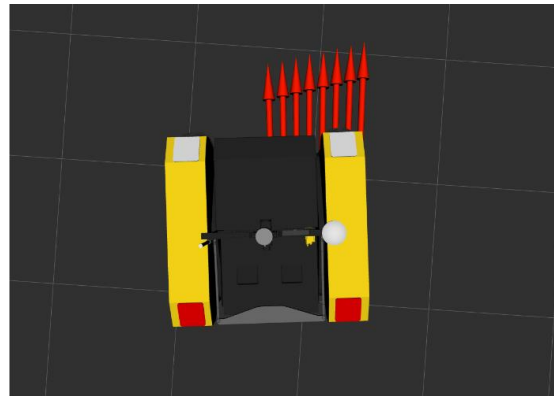
Anexos

Anexo 1: Odometría

Al analizar el movimiento de la plataforma robótica en comparación con la odometría visualizada en RVIZ, se observó una rotación en torno al eje z, lo que implica una variación en la variable angular “yaw”. También se notó una translación en el eje “y” (Ver Figura 32).



(a) Rotación en torno a “z”



(b) Traslación en “y”

Figura 32: Comportamiento de desplazamiento asociado a la odometría del robot Warthog en el software de visualización RVIZ. Fuente: elaboración propia.

A raíz de estos hallazgos, se consideró necesario revisar la configuración predeterminada de fábrica que se encuentra en el ordenador de la plataforma robótica. A continuación, en la Figura 33 se detalla esta configuración:

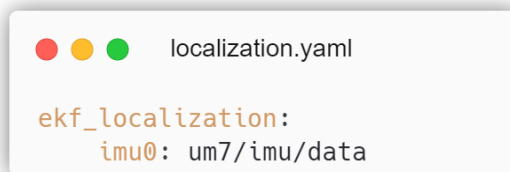


Figura 33: Configuración de odometría original. Fuente: elaboración propia.

Como se puede apreciar, la configuración inicial solo toma en cuenta la información de la IMU UM7, pero no especifica qué parámetros utilizar de esta información. Por tanto, se requiere actualizar esta configuración para incluir los parámetros relevantes para la odometría del robot. Para lograrlo, se seleccionaron los parámetros de posición, velocidad angular “yaw” y aceleración en “x”. Además, se integró la información del tópico correspondiente a la odometría interna del

robot, que se obtiene a través del modelo cinemático del robot. Para ello, se configuraron los parámetros de manera que se consideren las velocidades lineales “x” e “y”, así como la velocidad angular “yaw”. A continuación, en la Figura 34 se muestra la configuración propuesta:

```
localization.yaml

ekf_localization:
  imu0: um7/imu/data

  #Configuración de odometría actualizada (RISLAB)
  imu0_config: [false, false, false,
                false, false, true,
                false, false, false,
                false, false, true,
                true, false, false]

  imu0_remove_gravitational_acceleration: true

  odom0: warthog_velocity_controller/odom
  odom0_config: [false, false, false,
                 false, false, false,
                 true, true, false,
                 false, false, true,
                 false, false, false]
```

Figura 34: Configuración de odometría propuesta. Fuente: elaboración propia.

Una vez más, al analizar el movimiento del robot real en comparación con la odometría visualizada en RVIZ, se puede observar que el problema mencionado anteriormente no se presenta. Se logra una representación precisa del movimiento real en el software de visualización (ver Figura 35).

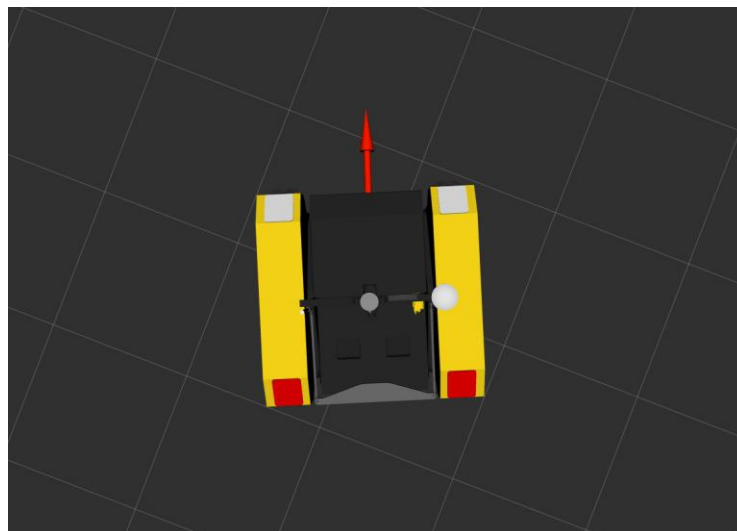


Figura 35: Corrección de desplazamiento de odometría en el software RVIZ. Fuente: elaboración propia.

Anexo 2: Transformadas entre sensores

La Figura 36 del presente anexo, representa el diagrama de árbol correspondiente a las transformadas existentes entre los sensores.

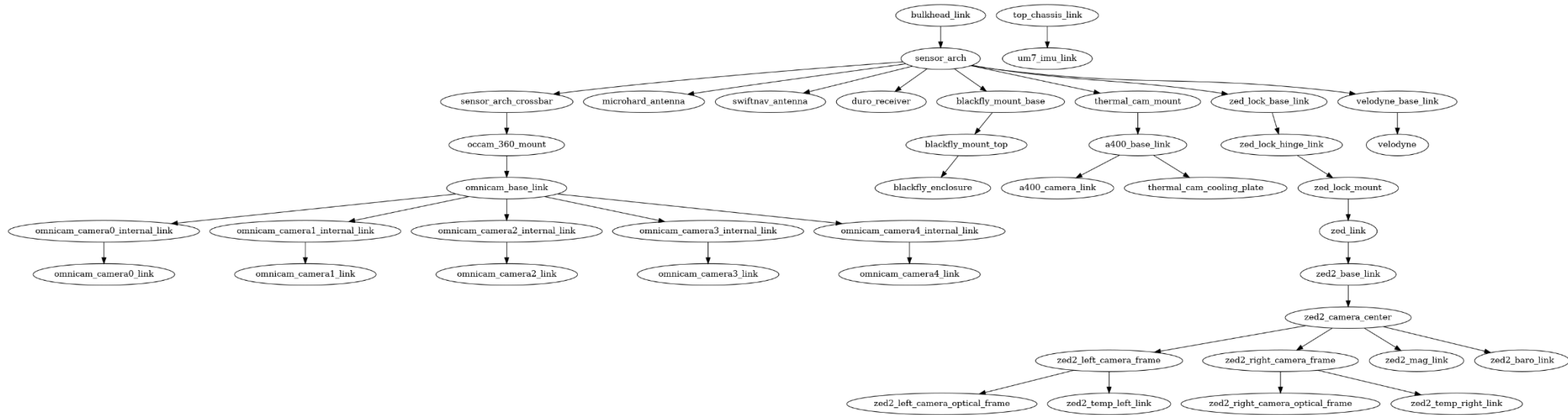


Figura 36: Diagrama de árbol obtenido a través del código de Python. Fuente: elaboración propia.

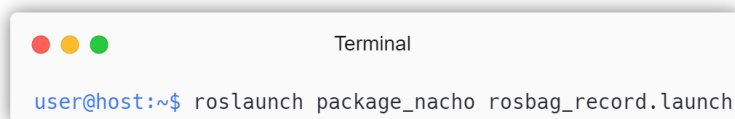
Anexo 3: Adquisición de datos

La adquisición de datos proporcionados por los sensores de la plataforma robótica es un proceso fundamental a la hora de trabajar en distintos terrenos. Cada sensor debe ser seleccionado en función de los datos específicos necesarios y de los objetivos de la tarea. Por ejemplo, las cámaras pueden usarse para capturar imágenes de alta resolución, mientras que el LiDAR es valioso para obtener información tridimensional detallada del entorno. El GPS, por su parte, proporciona datos precisos de localización.

El 13 de septiembre del año 2023, se llevó a cabo la primera salida a terreno de la plataforma robótica en Fundo La Torre en Rengo, Chile. Específicamente, el recorrido se enfocó en la hilera 66, la cual alberga árboles cerezos y tiene una longitud de 140 metros aproximadamente. Dado que esta fue la primera salida del robot, se enfocó en observar su comportamiento en el terreno y recopilar datos de todos los sensores. Para llevar a cabo esta tarea, se utilizó el paquete rosbag de ROS, el cual proporciona una amplia variedad de herramientas para registrar la información obtenida por los diferentes sensores montados en el robot.

La segunda salida de la plataforma robótica se realizó en el mismo lugar el 24 de noviembre del mismo año, con el propósito de realizar diversas pruebas y recopilar datos de todos los sensores para establecer una "base de datos" del mismo sector en diferentes fechas (ver Tabla 8).

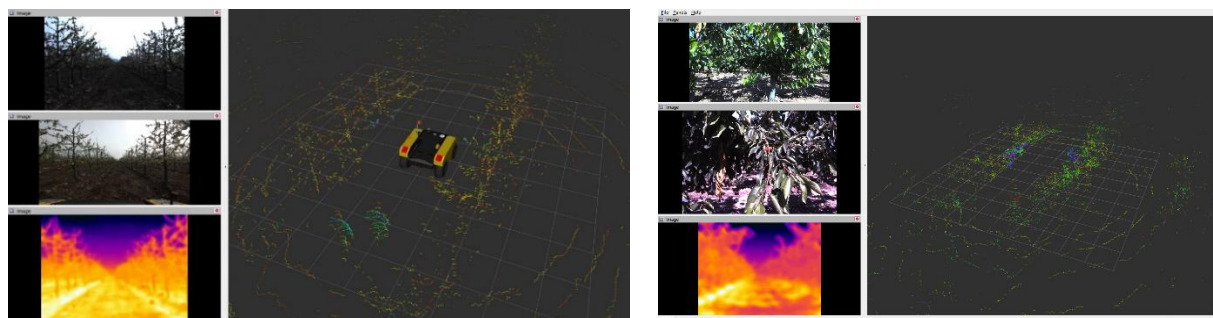
Para registrar la información de los sensores, se emplearon algoritmos implementados en Python, los cuales se ejecutan desde un archivo .launch y permiten monitorear el almacenamiento en tiempo real y una interacción más amigable al grabar los tópicos. Este archivo se inicia desde la consola con el siguiente comando:

A screenshot of a terminal window titled "Terminal". The window has three colored window control buttons (red, yellow, green) in the top-left corner. The terminal text shows a prompt "user@host:~\$" followed by the command "roslaunch package_nacho rosbag_record.launch".

```
user@host:~$ roslaunch package_nacho rosbag_record.launch
```

Figura 37: Comando para grabar tópicos desde archivo de Python creado. Fuente: elaboración propia.

A continuación, en la Figura 38 se presentan algunos de los datos recopilados en ambas salidas, en el software RVIZ. Estos datos pertenecen a las imágenes de la cámara FLIR Blackfly S, cámara Stereolab ZED 2, cámara FLIR A400 y la nube de puntos generada por el sensor LiDAR Velodyne Puck (VLP-16).



(a) Primera salida: 13-09-2023

(b) Segunda salida: 24-11-2023

Figura 38: Visualización de los datos registrados en RVIZ, correspondiente al fundo la Torre, Rengo-Chile. Fuente: elaboración propia.

Tabla 8: Detalle de los archivos .bag generados durante las experiencias de terreno. Fuente: elaboración propia.

Lugar / Fecha	Nombre del archivo	Tamaño	Descripción
Fundo la Torre, Rengo-Chile / 13- 09-2023	toma1.bag	50,2 GB	Ida y vuelta por hilera 66 desde el este.
	toma2_camara.bag	64,1 GB	Ida y vuelta por hilera 66 desde el este.
	giro.bag	17,3 GB	Giro sobre propio eje de Warthog.
Fundo la Torre, Rengo-Chile / 24- 11-2023	toma1.bag	91,0 GB	Ida y vuelta por hilera 66 desde el este. 0.32 m/s a 1,3 m de distancia.
	toma2.bag	121,0 GB	Recorrido por hileras 66 a 73. 0.5 m/s aproximadamente a 1,5 m de distancia.

Anexo 4: Gráficos de odometría utilizada en mapeo

La Figura 39, Figura 40 y Figura 41 representan las distintas odometría utilizadas en la implementación del algoritmo de mapeo.

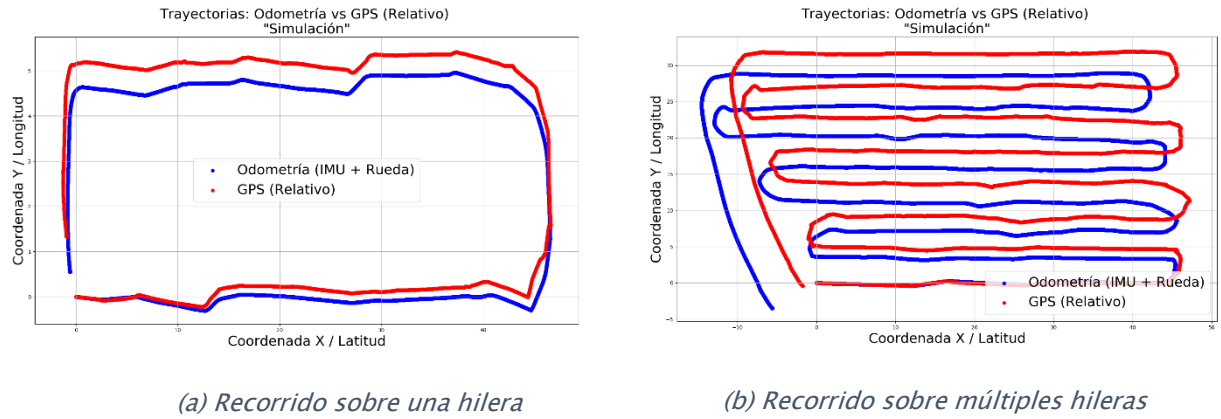


Figura 39: Elaboración de trayectorias mediante odometría y datos GPS en un entorno simulado, con el propósito de implementarlas en un algoritmo de mapeo utilizando una cámara estéreo. Fuente: elaboración propia.

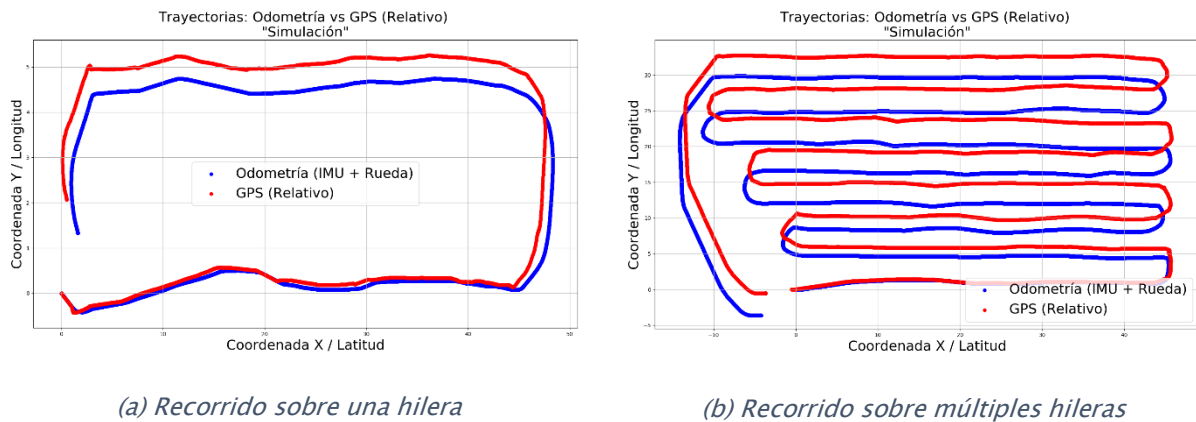
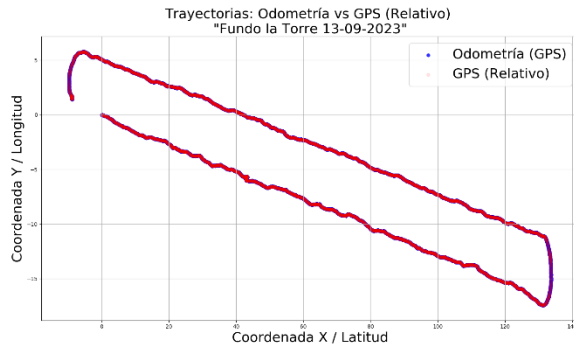
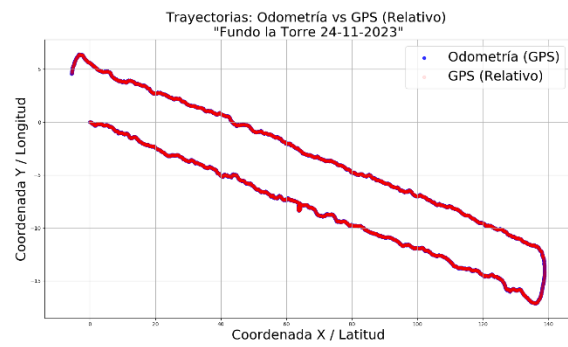


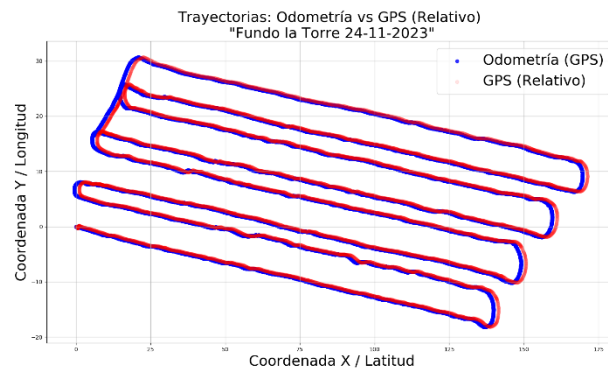
Figura 40: Elaboración de trayectorias mediante odometría y datos GPS en un entorno simulado, con el propósito de implementarlas en un algoritmo de mapeo utilizando un sensor LiDAR. Fuente: elaboración propia.



(a) Recorrido sobre una hilera (13-09-2023)



(b) Recorrido sobre una hilera (24-11-2023)

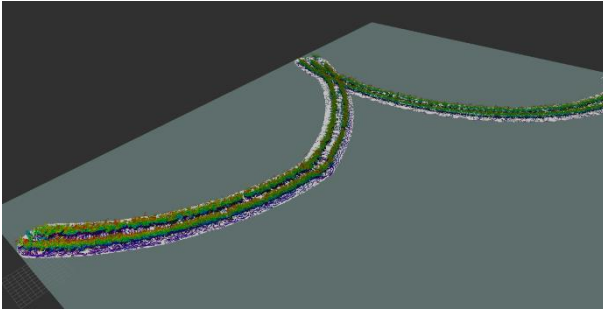


(c) Recorrido sobre múltiples hileras (24-11-2023)

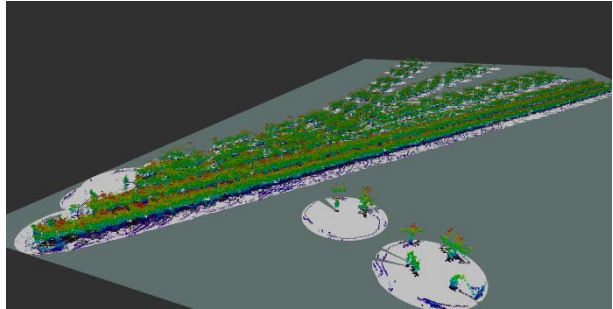
Figura 41: Elaboración de trayectorias mediante odometría y datos GPS en un ambiente real, con el propósito de implementarlas en un algoritmo de mapeo de un entorno frutícola. Fuente: elaboración propia.

Anexo 5: Mapas obtenidos con odometría inicial

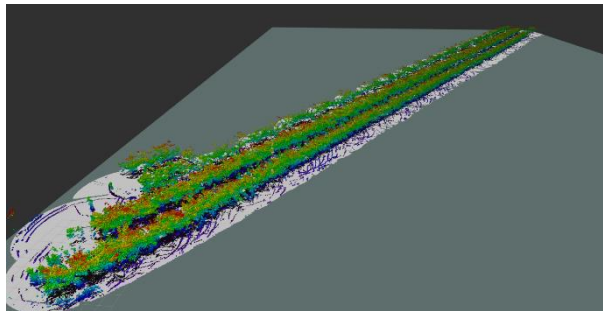
La Figura 42 deja en evidencia los mapas obtenidos con los datos de las salidas a terreno, sin considerar la utilización de la nueva odometría creada con datos GPS.



(a) Primera salida: 13-09-2023



(b) Segunda salida: 24-11-2023



(c) Segunda salida: 24-11-2023

*Figura 42: Elaboración de mapas mediante la utilización de datos reales obtenidos en el Fundo La Torre.
Fuente: elaboración propia.*

Anexo 6: Métricas para evaluación cuantitativa

La Tabla 9 y Tabla 10 recopilan las métricas obtenidas al momento de ejecutar o realizar el mapeo en simulación o datos reales.

Tabla 9: Métricas obtenidas al realizar mapeo con sensor LiDAR. Fuente: elaboración propia.

Datos reales						Simulación			
TOMA 1 (13-09-2023)		TOMA 1 (24-11-2023)		TOMA 2 (24-11-2023)		TOMA 1		TOMA 2	
RAM	CPU	RAM	CPU	RAM	CPU	RAM	CPU	RAM	CPU
3.1	99	0.8	100	1.0	100	0.6	99.1	0.5	101
4.6	100	1.8	98.7	2.4	103	0.8	103	1.4	102
5.1	101	2.3	102	4.1	100	1.1	100	2.4	100
5.4	102	3.0	101	5.8	103	1.5	101	3.1	106
5.9	101	3.8	101	6.7	101	2.0	100	4.5	104
6.3	101	4.2	100	7.3	102	2.5	103	3.6	105
7.1	100	5.5	103	8.6	100	2.8	98.9	5.2	104
6.9	102	6.3	101	9.0	102	3.1	104	6.4	102
7.0	101	6.9	99.6	10.7	100	2.8	107	7.2	104
8.0	102	7.6	100	13.1	103	3.1	104	8.0	101
7.5	100	8.4	102	11.7	100	3.2	103	7.8	100

Tabla 10: Métricas obtenidas al realizar mapeo con cámara estéreo ZED 2. Fuente: elaboración propia.

Datos reales						Simulación			
TOMA 1 (13-09-2023)		TOMA 1 (24-11-2023)		TOMA 2 (24-11-2023)		TOMA 1		TOMA 2	
RAM	CPU	RAM	CPU	RAM	CPU	RAM	CPU	RAM	CPU
0.9	100	0.9	84.5	0.9	99	1.1	112	1.0	113
1.6	101	1.2	78.9	0.9	78	1.3	119	1.1	118
3.0	101	2.6	100	1.8	100	1.5	117	1.8	113
3.9	102	2.9	102	2.2	101	1.8	115	2.5	112
4.4	99.7	3.6	101	3.2	101	2.2	112	3.1	107
4.5	102	2.8	102	4.2	100	1.9	114	3.9	110
5.0	100	4.3	100	4.7	99.2	2.3	109	4.2	103
5.0	101	3.9	99	5.3	102	2.7	113	3.9	109
5.2	99	4.6	100	6.0	100	2.5	111	4.4	115
5.8	102	4.7	101	5.9	100	2.4	114	4.8	113
4.7	101	4.4	102	6.8	99	2.3	110	4.7	110