



UNIVERSIDAD TÉCNICA DEL NORTE

FACULTAD DE ANGENIERÍA EN CIENCIAS APLICADAS

CARRERA DE INGENIERÍA EN MECATRÓNICA

TRABAJO DE GRADO PREVIO A LA OBTENCIÓN DEL TÍTULO  
DE INGENIERO EN MECATRÓNICA

TEMA:

“ANÁLISIS COMPARATIVO DE DIFERENTES TECNOLOGÍAS DE  
SENSORES PARA SLAM SOBRE ROBOTS MÓVILES ”

AUTOR: JESSICA ESTEFANIA QUIMBIA LOYO

DIRECTOR: CARLOS XAVIER ROSERO CHANDI

IBARRA-ECUADOR  
2023



**UNIVERSIDAD TÉCNICA DEL NORTE**  
**BIBLIOTECA UNIVERSITARIA**  
**AUTORIZACIÓN DE USO Y PUBLICACIÓN A FAVOR DE LA**  
**UNIVERSIDAD TÉCNICA DEL NORTE**  
**IDENTIFICACIÓN DE LA OBRA**

En cumplimiento del Art. 144 de la Ley de Educación Superior, hago la entrega del presente trabajo a la Universidad Técnica del Norte para que sea publicado en el Repositorio Digital Institucional, para lo cual pongo a disposición la siguiente información:

<b>DATOS DEL AUTOR</b>			
CÉDULA DE IDENTIDAD	1003743323		
APELLIDOS Y NOMBRES	QUIMBIA LOYO JESSICA ESTEFANIA		
DIRECCIÓN EMAIL	Ibarra jequimbial@utn.edu.ec		
TELÉFONO FIJO		TELÉFONO MÓVIL	0994857010
<b>DATOS DE LA OBRA</b>			
TÍTULO	"ANÁLISIS COMPARATIVO DE DIFERNTES TECNOLOGÍAS DE SENSORES PARA SLAM SOBRE ROBOTS MÓVILES"		
AUTOR	JESSICA ESTEFANIA QUIMBIA LOYO		
FECHA	06/06/2023		
PROGRAMA	PREGRADO		
TÍTULO POR EL QUE OPTA	INGENIERÍA EN MECATRÓNICA		
DIRECTOR	CARLOS XAVIER ROSERO CHANDI		



UNIVERSIDAD TÉCNICA DEL NORTE  
FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS  
CONSTANCIA

El autor manifiesta que la obra objeto de la presente autorización es original y se la desarrolló sin violar derechos de autor de terceros, por lo tanto la obra es original, y que es el titular de los derechos patrimoniales, por lo que asume la responsabilidad sobre el contenido de la misma y saldrá en defensa de la Universidad en caso de reclamación por parte de terceros.

Ibarra, a los 6 días del mes de junio de 2023

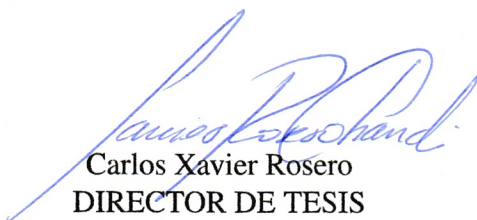
Jessica Estefania Quimbia Loyo  
C.I.: 1003743323



**UNIVERSIDAD TÉCNICA DEL NORTE**  
**FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS**  
**CERTIFICACIÓN**

En calidad de director del trabajo de grado “ANÁLISIS COMPARATIVO DE DIFERENTES TECNOLOGÍAS DE SENSORES PARA SLAM SOBRE ROBOTS MÓVILES”, presentado por la egresada Jessica Estefania Quimbia Loyo, para optar por el título de Ingeniera en Mecatrónica, certifico que el mencionado proyecto fue realizado bajo mi dirección.

Ibarra, a los 6 días del mes de junio de 2023

  
Carlos Xavier Rosero  
DIRECTOR DE TESIS

## **Agradecimiento**

En primer lugar, quiero agradecer a mis padres y hermanos por la motivación y apoyo en mi desarrollo personal y profesional.

A mi director de tesis Carlos Xavier Rosero quiero expresar mi sincero agradecimiento por su conocimiento, paciencia y enseñanza durante este proceso. También por ser quien confió en mí desde el principio.

*Jessica Quimbia Loyo*

## **Dedicatoria**

Este trabajo está dedicado a mi familia. Principalmente, a mi madre Rosa y mi padre Gonzalo por todo el apoyo incondicional.

*Jessica Quimbia Loyo*

## Resumen

En la actualidad en la robótica móvil se utilizan diferentes tipos de sensores para aplicaciones de Localización y Mapeo Simultáneos (SLAM, Simultaneous Location and Mapping). Estos sensores poseen sus propias ventajas y desventajas. Sin embargo, pocos estudios han comparado el rendimiento de los sensores en aplicaciones similares. El propósito de este trabajo es realizar un análisis comparativo de rendimiento de SLAM de un sensor láser de distancia frente a la cámara de profundidad, utilizando el robot Kobuki Turtlebot 2. Para realizar la comparativa primero se obtiene el mapa del entorno. Se lleva a cabo SLAM con el algoritmo Gmapping. A continuación, se tiene información de las trayectorias con ayuda del paquete Adaptativo de Localización de Monte Carlo (AMCL, Adaptative Monte Carlo Localization) y con una métrica de error se evalúa el rendimiento de los sensores. Las pruebas en interiores emiten que el error de localización de los sensores es inferior al 5%. No obstante, el sensor láser de distancia es más eficiente en función a los criterios de rendimiento del sistema SLAM.

**Palabras clave:** SLAM, Kinect, Rplidar, Comparativa

## **Abstract**

A variety of sensors are currently used in mobile robotics for Simultaneous Location and Mapping (SLAM) applications. Each of these sensors has advantages and disadvantages. Although few studies have compared the performance of sensors in similar conditions. The aim of this work is to implement a comparative analysis of the SLAM performance between laser sensor to the depth camera, using the Kobuki Turtlebot 2 robot. First you get a map of the indoor environment to compare. SLAM is carried out with the Gmapping algorithm. Then, trajectory information is obtained from the Adaptive Monte Carlo Localization (AMCL) package and an error metric evaluates the performance of the sensors. Indoor tests show that the localization error of the sensors is less than 5%. However, according to the performance evaluation of SLAM systems, the laser sensor is more efficient.

**Keywords:** SLAM, Kinect, Rplidar, Competitive



# Índice general

<b>1. Introducción</b>	<b>1</b>
1.1. Descripción del problema . . . . .	1
1.2. Objetivos . . . . .	2
1.2.1. Objetivo General . . . . .	2
1.2.2. Objetivos específicos . . . . .	2
1.3. Alcance . . . . .	2
1.4. Justificación . . . . .	3
<b>2. Análisis literario</b>	<b>4</b>
2.1. Estado del arte . . . . .	4
2.2. Marco teórico . . . . .	6
2.2.1. SLAM . . . . .	6
2.2.2. Cámara de profundidad en SLAM . . . . .	6
2.2.3. Sensor láser de distancia en SLAM . . . . .	9
2.2.4. Criterios de rendimiento para la comparación de métodos SLAM . . . . .	10
<b>3. Aplicación de localización y mapeo simultáneos</b>	<b>12</b>
3.1. Implementación de SLAM . . . . .	12
3.1.1. Obtención de datos del sensor . . . . .	13
3.1.2. Odometría del robot . . . . .	13
3.1.3. Filtro de partículas . . . . .	14

3.1.4.	Información de localización . . . . .	14
3.2.	Implementación de tecnologías . . . . .	14
3.2.1.	Puesta en marcha del robot . . . . .	15
3.2.2.	Primera prueba de SLAM con la cámara de profundidad . . . . .	16
3.2.3.	Primera prueba de SLAM con el sensor láser de distancia . . . . .	19
<b>4.</b>	<b>Resultados y análisis</b>	<b>21</b>
4.1.	Métrica de evaluación de rendimiento . . . . .	21
4.2.	Descripción de resultados . . . . .	22
4.2.1.	Resultados del tiempo de construcción de los mapas . . . . .	22
4.2.2.	Resultados de mapas con la cámara de profundidad . . . . .	23
4.2.2.1.	Escenario de habitación . . . . .	23
4.2.2.2.	Escenario de pasillo . . . . .	23
4.2.3.	Resultados de mapas con el sensor láser de distancia . . . . .	24
4.2.3.1.	Escenario de habitación . . . . .	24
4.2.3.2.	Escenario de pasillo . . . . .	25
4.2.4.	Resultados del error de localización . . . . .	25
4.3.	Análisis de los resultados . . . . .	29
<b>5.</b>	<b>Conclusiones y recomendaciones</b>	<b>31</b>
5.1.	Conclusiones . . . . .	31
5.2.	Recomendaciones . . . . .	32
5.3.	Trabajo a futuro . . . . .	32
<b>Anexos</b>		<b>37</b>
<b>A.</b>	<b>Construcción de un mapa (SLAM)</b>	<b>38</b>
<b>B.</b>	<b>Navegación autónoma</b>	<b>41</b>

# Índice de figuras

2.1. El marco típico del algoritmo visual SLAM [3] . . . . .	8
3.1. Proceso de mapeo con el algoritmo SLAM. . . . .	13
3.2. Robot móvil Turtlebot 2. . . . .	14
3.3. En [25], se implementa los siguientes comandos para el control manual del robot. . . . .	16
3.4. Comandos para la construcción del mapa del entorno con cámara de profundidad . . . . .	17
3.5. Comandos para evaluar el error relativo con la cámara de profundidad. . . . .	18
3.6. Comandos para la construcción del mapa del entorno con el sensor láser de distancia. . . . .	19
3.7. Comandos para evaluar el error relativo con el sensor laser de distancia. . . . .	20
4.1. Mapa de la habitación vacía usando la cámara de profundidad. . . . .	23
4.2. Mapa del pasillo con la cámara de profundidad. . . . .	24
4.3. Mapa del escenario de habitación con sensor láser de distancia. . . . .	24
4.4. Mapa del pasillo con sensor láser de distancia. . . . .	25
4.5. Rutas seleccionados en el mapa de la habitación vacía.(a) Trayectoria AB de la habitación vacía. (b) Trayectoria AC de la habitación vacía. . . . .	26
4.6. Trayectorias seleccionados en el mapa del pasillo de la casa.(a) Trayectoria AB del pasillo de una casa. (b) Trayectoria AC del pasillo de una casa. . . . .	27
A.1. Launch de mapeo con cámara Kinect. . . . .	38
A.2. Mapeo de la habitación vacía con cámara Kinect. . . . .	39

A.3. Mapeo del pasillo de una casa con el sensor LiDAR . . . . .	39
A.4. Mapa del pasillo de una casa con la cámara de profundidad. Tiempo de mapeo de 263 segundos. . . . .	40
A.5. Mapa del pasillo de una casa con sensor láser de distancia. Tiempo de mapeo de 150 segundos. . . . .	40
B.1. Launch de la navegación con el sensor LiDAR. . . . .	41

# Índice de tablas

2.1. Especificaciones técnicas de la cámara de profundidad en comparación con un sensor láser de distancia en SLAM [17]. . . . .	7
2.2. Descripción de algoritmo Visual SLAM [7]. . . . .	8
2.3. Características de 2D-LiDAR SLAM [8]. . . . .	9
4.1. Tiempo de generación del mapa 2D de la habitación vacía. . . . .	22
4.2. Tiempo de construcción del mapa global 2D del pasillo de una casa. . . . .	23
4.3. Resultados de la precisión de localización en el entorno de prueba de habitación vacía en la trayectoria AB. . . . .	26
4.4. Resultados de la precisión de localización en la trayectoria AC en el entorno de prueba de habitación vacía. . . . .	27
4.5. Resultados del error de localización en la trayectoria AB del pasillo de la casa. .	28
4.6. Resultados del error de localización en el pasillo de la casa en la trayectoria AC.	28
4.7. Resultados del análisis comparativo del rendimiento de SLAM bajo las tecnologías de cámara de profundidad y de sensor láser de distancia. . . . .	29

# Capítulo 1

## Introducción

### 1.1. Descripción del problema

Los robots móviles con navegación autónoma son cada vez más utilizados tanto en la vida cotidiana como en la industria. A causa de su versatilidad para acoplarse a diferentes aplicaciones [1]. El uso de mapas del entorno, la capacidad de explorar de forma independiente y obtener su propia pose en el mismo mapa, es esencial para que un robot móvil realice una tarea determinada evitando colisiones con obstáculos. Todo esto, sin necesidad de ser constantemente guiados y controlados por operadores humanos [2], [3].

En [4] se presentó un robot móvil para interiores, implementado usando el Sistema Operativo de Robots (ROS, Robot Operating System). Este desarrollo se centró en la exploración de fronteras basada en la técnica de navegación de Localización y Mapeo Simultáneos (SLAM, Simultaneous Location and Mapping), usando una cámara de profundidad como sensor principal. Sin embargo, el sensor láser de distancia también es útil en aplicaciones similares [5]. Éste proporciona una localización sólida gracias al uso de información espacial 3D escaneada mediante una nube de puntos láser [6].

Es necesario extender el trabajo presentado en [4] a través de la implementación de un sensor láser de distancia sobre el mismo robot y para similares escenarios de prueba. Esto permitirá realizar una comparativa del rendimiento de SLAM tanto para la cámara de profundidad como para el sensor láser de distancia; se podrán determinar los puntos fuertes y débiles de la integración de las dos tecnologías. Los resultados del análisis comparativo podrían ser útiles en la

academia y en la industria.

## **1.2. Objetivos**

### **1.2.1. Objetivo General**

Realizar una comparativa de rendimiento de SLAM considerando las tecnologías de cámara de profundidad y sensor láser de distancia.

### **1.2.2. Objetivos específicos**

- Determinar los criterios de rendimiento a considerarse dentro de la comparativa SLAM usando las dos tecnologías.
- Implementar las tecnologías bajo comparación sobre un robot móvil real.
- Evaluar el comportamiento del robot móvil desde la información obtenida de trayectorias y tiempos en distintos escenarios de prueba.

## **1.3. Alcance**

En el presente trabajo se presentará un análisis comparativo del rendimiento de SLAM bajo las tecnologías de cámara de profundidad y de sensor láser de distancia.

Luego de adaptar el robot móvil Kobuki Turtlebot 2 en [4] que trabaja usando ROS, para su funcionamiento con un sensor láser de distancia, se evaluarán tiempos y trayectorias en la exploración de fronteras, bajo diferentes escenarios. Estos resultados serán contrastados con aquellos obtenidos previamente al implementar el mismo robot usando una cámara de profundidad.

## **1.4. Justificación**

En robótica móvil es necesario que el robot use un mapa del entorno para obtener su ubicación exacta dentro de este y para predecir una ruta de un punto a otro, lo que se conoce como SLAM. Dentro del contexto mencionado, el presente trabajo proveerá información de posibles mejoras a implementar en el robot Kobuki Turtlebot 2 en [4] al considerar la implementación de una nueva tecnología de sensores. A la vez, será útil para que profesionales que trabajen con esta problemática puedan seleccionar soluciones de SLAM adecuadas para la navegación práctica en interiores de acuerdo con sus propios entornos [6].



# Capítulo 2

## Análisis literario

En este capítulo primero se expone brevemente sobre los trabajos relacionados con la comparativa de tecnologías de sensores en SLAM. Luego, se presenta la tipología de SLAM de acuerdo con la aplicación del sensor. A continuación, se muestra los principales sensores utilizados dentro de la técnica SLAM.

### 2.1. Estado del arte

La aplicación de robots móviles para la navegación en interiores es estudiada por técnicas de SLAM basadas en ROS. Para ello, se utilizan variedad de sensores [6]. La información que se consigue de los sensores ayuda a seleccionar soluciones de SLAM adecuadas para el entorno de trabajo.

Las técnicas de SLAM se clasifican en visual y LiDAR SLAM. Visual SLAM usa la cámara como sensor principal [7]. Emplea la imagen para identificar y mapear las posiciones del robot. Mientras que, LiDAR SLAM usa un sensor láser de distancia para escanear el entorno. Calcula la distancia del obstáculo con un pulso de láser y construye el mapa [8].

La investigación en [7] se enfoca en visual SLAM con cámaras RGB-D. Las cámaras RGB-D se pueden dividir en Microsoft Xbox 360 Kinect y cámara Microsoft Xbox One Kinect. Con el propósito de evaluar el rendimiento del sistema visual SLAM se usó dos métodos para la

medición del error de localización. El error absoluto de trayectoria (ATE, Absolute Trajectory Error) y el error relativo de pose (RPE, Relative Pose Error). El SLAM basado en Kinect ofrece a los robots móviles aplicaciones en interiores.

Por otro lado, en [9] se presenta una comparativa de sensores de radar y LiDAR en SLAM con el método de error relativo (RE, Relative Error). Para evaluar el rendimiento de los sensores en SLAM se usa la información de las trayectorias de la navegación. Para cada trayectoria, se calcula la distancia euclidiana entre poses del robot en el mapa creado. El robot se localiza gracias al paquete AMCL [10]. En otra comparativa en [8] el sensor LiDAR muestra un mayor rendimiento en comparación con las cámaras. Los sensores 3D LiDAR son más precisos y tienen mejor resolución a las versiones 2D. No obstante, LiDAR 2D es adecuado para un ambiente interior simple [11].

Asimismo, en [12] compara sensores en sistemas SLAM (como sensores Kinect y LiDAR) en un escenario creado en un laboratorio. La comparativa mide el rendimiento de los sensores en SLAM en tiempo real. Los resultados indican que el sensor Kinect tiene inconvenientes con la resolución de los mapas en tiempo real, mientras que el sensor LiDAR adquiere mapas más cercanos con la realidad.

Comparar tipologías de SLAM Visual y LiDAR SLAM detectando puntos débiles y fuertes que ratifica el nivel de precisión de los diferentes métodos y así implementa posibles mejoras en el sistema estudiado [13].

Este trabajo presenta una comparación de rendimiento de SLAM con una cámara de profundidad y un sensor láser de distancia. Ambas tecnologías de sensores se pueden apreciar en la Fig 3.2 en la plataforma del robot móvil Turtlebot 2. Se realiza SLAM con el algoritmo Gmapping [11], [12] de interiores. Además, para la localización del robot se usa el paquete AMCL [10]. Con las pruebas experimentales en escenarios diferentes se obtiene información

de tiempos de mapeo y distancias [9] de trayectorias que permiten la evaluación del rendimiento de SLAM de las tecnologías. Los resultados de la comparativa pueden ser adecuados para trabajos de la academia y la industria.

## **2.2. Marco teórico**

### **2.2.1. SLAM**

La Localización y Mapeo Simultáneos es una técnica que se usa en robots móviles [14]. La principal característica del SLAM es construir un mapa 2D de un entorno y conocer, a su vez, la ubicación del robot con la autoexploración de este.

Los algoritmos en SLAM ayudan a reunir la información obtenida de los sensores. El algoritmo Gmapping es uno de los más usado en SLAM 2D, para la implementación del algoritmo especialmente se usa sensores láser. Gmapping emplea los datos de odometría y concurrencia de escaneo, para evaluar el movimiento de la tecnología (sensor) y la localización del robot en el entorno (mapa) [15].

### **2.2.2. Cámara de profundidad en SLAM**

RGB-D SLAM es un método del SLAM visual que usa la cámara RGB-D para lograr SLAM. La cámara RGB-D conocida como cámara de profundidad se aplica como sensor. La cámara facilita una imagen de color RGB y una imagen de profundidad del espacio escaneado simultáneamente por píxel. Además, mediante medición física puede obtener referencia de profundidad [7].

La cámara de profundidad Microsoft Xbox 360 Kinect se basa en el principio de detector de luz infrarroja, que emite un haz de luz estructurada al objetivo escaneado y posteriormente calcula la distancia entre el objeto y la cámara [16].

En condiciones de ambiente cambiante o inestable, la sensibilidad de la cámara a la luz, así como la eficiencia de extraer información 3D no es conveniente como el sensor láser de distancia [11].

Con referencia a la Tabla 2.1 en función al alcance, el sensor láser es más eficiente que la cámara de profundidad. También, el sensor posee un ángulo máximo de visión de hasta 360 grados. Por otro lado, el sensor Kinect, es más preciso en la navegación por la dimensionalidad que este brinda.

Tabla 2.1: Especificaciones técnicas de la cámara de profundidad en comparación con un sensor láser de distancia en SLAM [17].

<b>Especificaciones técnicas de tecnologías de sensores</b>		
Componentes	Kinect	Escáner láser 2D
Rango de operación (m)	0.8 a 4.0	0.16 a 6.0
Ángulo horizontal (°)	57	360
Ángulo vertical (°)	43	-
N° puntos de medición	640x480=307200	hasta 6000

En la Fig 2.1 se muestra el marco algoritmo típico usado en visual SLAM. El algoritmo investiga la ubicación en el mapa construido en tiempo real.

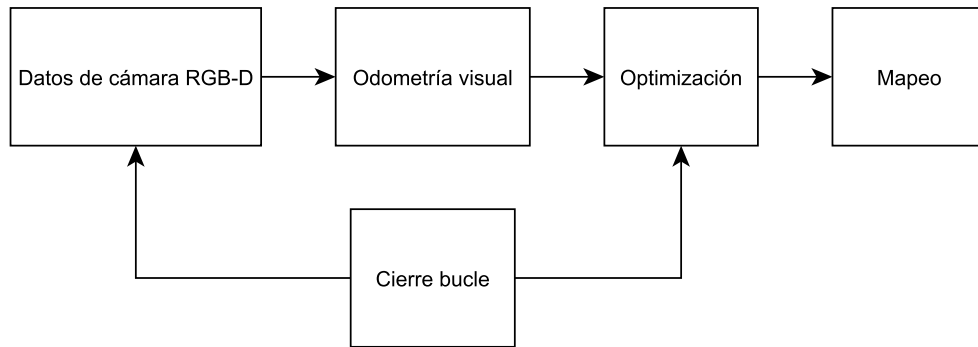


Figura 2.1: El marco típico del algoritmo visual SLAM [3]

La Tabla 2.2 la aplicación de algoritmos de visual SLAM en robot móviles autónomos, se han realizado estudios que describen ciertas características: odometría visual, optimización, cierre de bucle y mapeo.

Tabla 2.2: Descripción de algoritmo Visual SLAM [7].

<b>Trabajo del algoritmo de Visual SLAM</b>	
Descripción	
Odometría visual (como front-end)	Proporciona información del movimiento de la cámara medida en diferentes momentos y la visualización del mapa.
Optimización (como back-end)	Reduce ruido en sistema SLAM. Optimiza trayectorias y obtiene mapas precisos una vez que adquiere la información de la pose medida por el front-end.
Lazo cerrado	Rectifica la pose estimada y borra la desviación con los datos similares que recopila la cámara al pasar por el mismo lugar.
Mapeo	Es el proceso de crear una presentación del entorno local.

### 2.2.3. Sensor láser de distancia en SLAM

LiDAR SLAM proporciona un mayor rendimiento de localización en interiores. LiDAR SLAM utiliza un LiDAR como sensor para escanear el entorno. Los sistemas de LiDAR SLAM se basan en técnicas de concurrencia de escaneo [5], [13].

El sensor LiDAR usando una fuente de luz constante, logra mediciones de mayor precisión y obtener información más precisa del entorno independientemente del cambio de luz. El sensor LiDAR, además, puede trabajar de manera estable en condiciones de movimiento inestable [8].

Las principales características de LiDAR SLAM se representan con el mapeo, localización y navegación, como se muestra en la Tabla 2.3.

Tabla 2.3: Características de 2D-LiDAR SLAM [8].

<b>Característica de 2D-LiDAR SLAM</b>	
Descripción	
Mapeo	Permite a los robots construir un mapa del entorno.
Localización	El robot puede conocer su posición en el mapa generado. El posicionamiento ayuda al robot a evaluar la trayectoria para evitar un choque cuando un objeto está cerca.
Navegación	Combina características de navegación y de posicionamiento. La navegación facilita al robot optar por una trayectoria adecuada según los datos obtenidos.

El sensor LiDAR en la localización en interiores con pasillos largos tiene problemas. A diferencia de las trayectorias con rotaciones [18].

#### 2.2.4. Criterios de rendimiento para la comparación de métodos SLAM

La eficiencia y precisión en la localización y el mapeo en sistemas SLAM en interiores dependerá de varios factores, como el algoritmo utilizado, tipo de aplicación, el entorno, la precisión requerida, el rango de los sensores y las limitaciones del hardware implementado. En una aplicación de tiempo real, el sensor láser de distancia presenta una eficiencia temporal considerablemente mayor que la cámara de profundidad, llegando incluso a triplicarla [12]. Asimismo, en sistemas SLAM, cuando la tasa de error oscila entre 1% y 5%, se considera aceptable [5], [19], [20], [21], [22]. Esto implica que el sistema puede alcanzar del 95% al 99% de precisión.

Los criterios de rendimiento del tiempo en una aplicación real [12] y error de localización [17] son relevantes para evaluar la eficiencia y precisión de los sistemas SLAM de la siguiente manera: en el experimento, el robot se controló de manera remota. Se registró en tiempo real todos los datos de la lectura del sensor del entorno con el paquete **rosvbag**, por otro lado, para comprobar la idoneidad de localización en términos de distancia entre el cálculo del sensor en el sistema SLAM y la medición del mundo real, se usó el **error relativo**. Los criterios a continuación:

Tiempo de mapeo: es la cantidad de segundos que el robot tarda para la generación del mapa 2D del entorno de prueba.

Distancia de trayectoria: es la distancia euclidiana en metros que el robot recorre desde una posición inicial hasta una posición final.

##### **Distancia euclidiana**

La distancia euclidiana es el cálculo entre dos puntos [23]. Para calcular la distancia euclidiana  $D_e$  se usa

$$D_e = \sqrt{(x_B - x_A)^2 + (y_B - y_A)^2}, \quad (2.1)$$

donde  $D_e$  = representa la distancia euclidiana entre punto A y B,  $x_A$  = es la coordenada x del punto A,  $x_B$  = es la coordenada x del punto B,  $y_A$  = es la coordenada y del punto A,  $y_B$  = es la coordenada y del punto B.



## Capítulo 3

# Aplicación de localización y mapeo simultáneos

Este capítulo proporciona información a seguir para la configuración experimental. Desde la puesta en marcha del robot, la implementación del launch de mapeo con el algoritmo SLAM en el robot móvil Turtlebot 2 con los diferentes sensores hasta los pasos de evaluación de rendimiento de las tecnologías.

### 3.1. Implementación de SLAM

En la Figura 3.1 se observa el flujograma del proceso del algoritmo de SLAM.

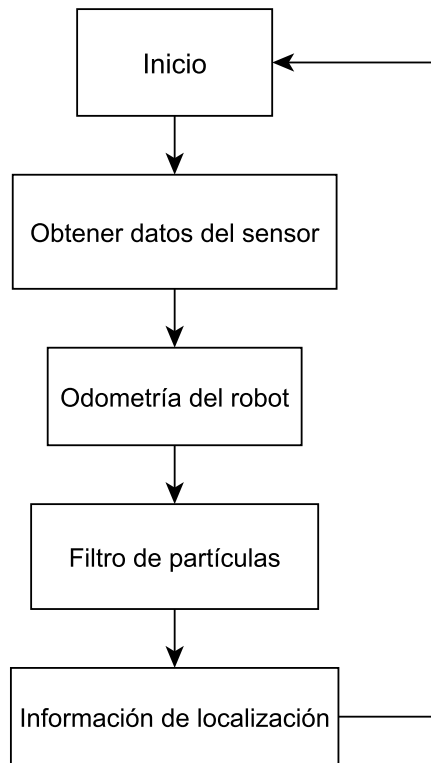


Figura 3.1: Proceso de mapeo con el algoritmo SLAM.

### 3.1.1. Obtención de datos del sensor

La función principal del sensor es proporcionar información con la que se puede determinar la ubicación y la posición del robot en el entorno interior.

### 3.1.2. Odometría del robot

Estima el movimiento del robot en el mapa local que se actualiza constantemente [7]. Sin embargo, alberga un error.

### 3.1.3. Filtro de partículas

El filtro de partículas ayudan a reducir el error de la odometría. Además, al igual que el filtro de Kalman este también predice y actualiza la información [24].

### 3.1.4. Información de localización

Usando la información de ubicación recopilada por el sensor del robot, se construye un mapa bidimensional del área mapeada.

## 3.2. Implementación de tecnologías

En solución a la necesidad, en primer lugar, se adapta el robot móvil Kobuki trabajado en [4] para su operación con el sensor láser de distancia, como se nota en la Figura 3.2. Luego, con el software que comanda el proceso y la tecnología de sensores implementada, se realizan pruebas funcionales con el fin de validar su funcionamiento y obtener resultados.



Figura 3.2: Robot móvil Turtlebot 2.

La plataforma robótica, conocida como Turtlebot 2, se ha emplea en estudios preliminares, ofrece una estructura fácil de comprender para conectar diferentes tipos de sensores. Algunas de las especificaciones y características de este trabajo son:

- Implementación de ROS, un software libre compatible para el robot móvil. ROS es un entorno de trabajo (framework) que se adapta con sencillez, con diversas librerías y paquetes para el funcionamiento del robot.
- El sistema es adaptable que permite fácilmente agregar sensores. En este caso, sensor láser de distancia.
- Las pruebas de funcionamiento se desarrollan en ambientes interiores.

### **3.2.1. Puesta en marcha del robot**

Para revisar que exista la comunicación de la base Kobuki con la netbook ASUS, se hace un ejercicio de teleoperación que permite el control manual del robot Turtlebot 2, como se refleja en la Figura 3.3.

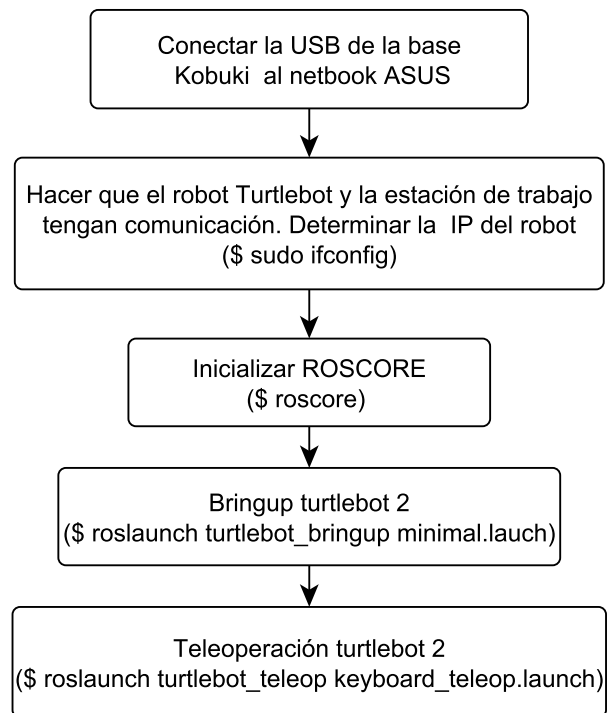


Figura 3.3: En [25], se implementa los siguientes comandos para el control manual del robot.

### 3.2.2. Primera prueba de SLAM con la cámara de profundidad

Una vez instalado los controladores en ROS para la cámara Kinect, se prueba en la aplicación de SLAM ejecutando los siguientes comandos que se encuentran en [26].

Para la construcción del mapa 2D del entorno, se usa el algoritmo Gmapping basado en el Anexo A con la teleoperación del robot móvil. Se sigue una serie de comandos descritos en orden como se indica en la Figura 3.4. También, se guardó los datos de mapeo con el comando de ROS 'roslaunch record -a'.

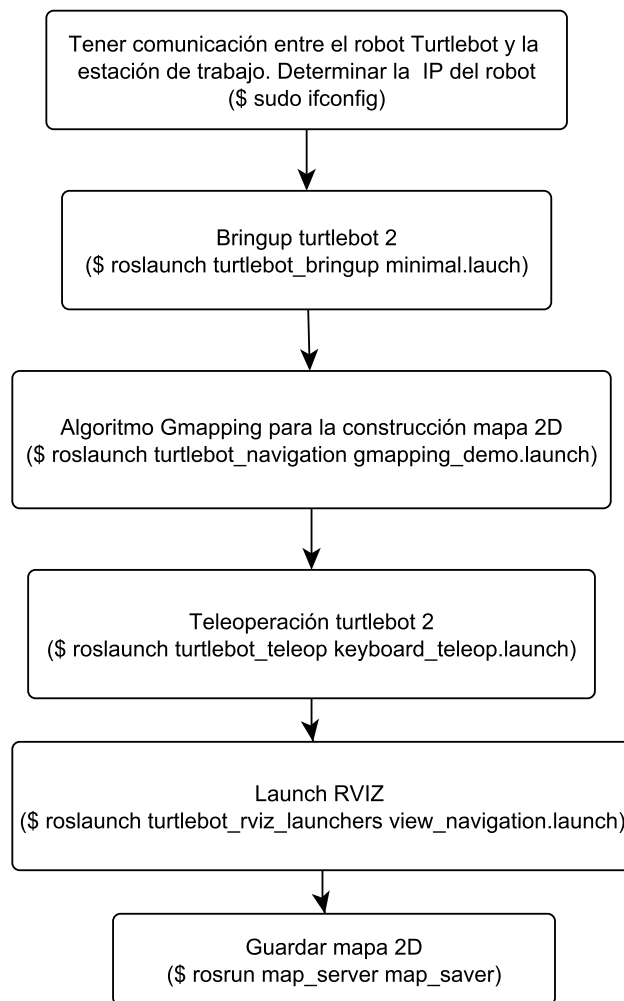


Figura 3.4: Comandos para la construcción del mapa del entorno con cámara de profundidad

La Figura 3.5 muestra la secuencia de pasos para evaluar a la cámara de profundidad en términos de localización. Se usa el paquete AMCL para encontrar la ubicación del robot en un mapa previamente guardado y obtenga las poses de las trayectorias, que luego se usan para calcular la distancia euclidiana.

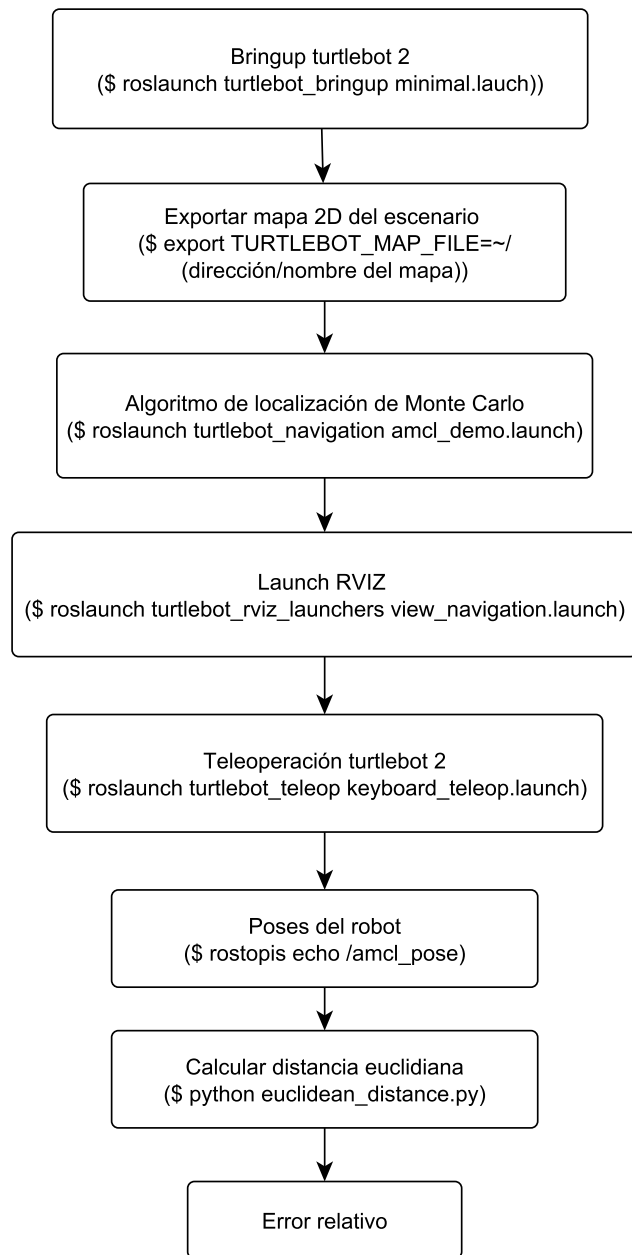


Figura 3.5: Comandos para evaluar el error relativo con la cámara de profundidad.

### 3.2.3. Primera prueba de SLAM con el sensor láser de distancia

Previamente a la prueba, se instala el paquete **rplidar ROS** que se halla en [27], compatible para el sensor rplidar A2. La Figura 3.6 tiene los comandos para el mapeo del escenario con el sensor láser de distancia supliendo a la cámara de profundidad. Para obtener un mapa 2D del área de prueba.

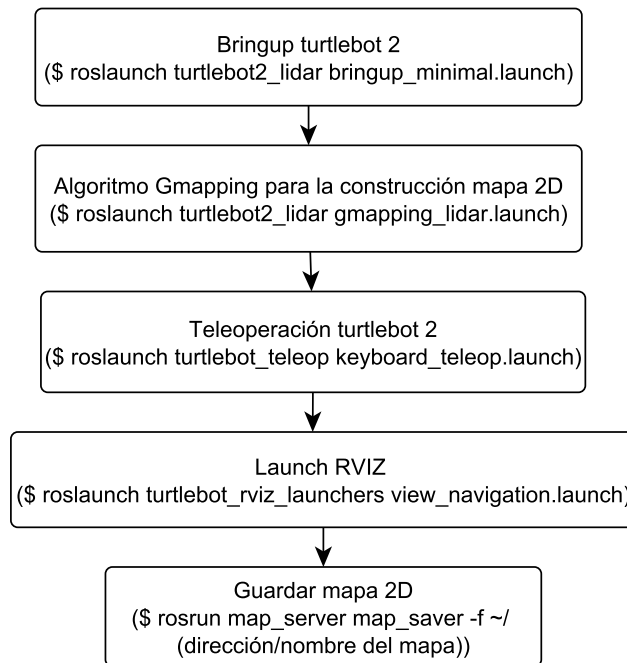


Figura 3.6: Comandos para la construcción del mapa del entorno con el sensor láser de distancia.

Como indica la Figura 3.7 el rendimiento del sensor de distancia láser se evalúa utilizando la información de la trayectoria seleccionada. Para comprender la ubicación del robot, se necesita el paquete AMCL con adaptaciones para recibir datos del sensor LiDAR (véase, el Anexo B) y el mapa 2D creado por el algoritmo Gmapping. Finalmente, se calcula el error relativo entre la distancia calculada y la distancia real del robot.



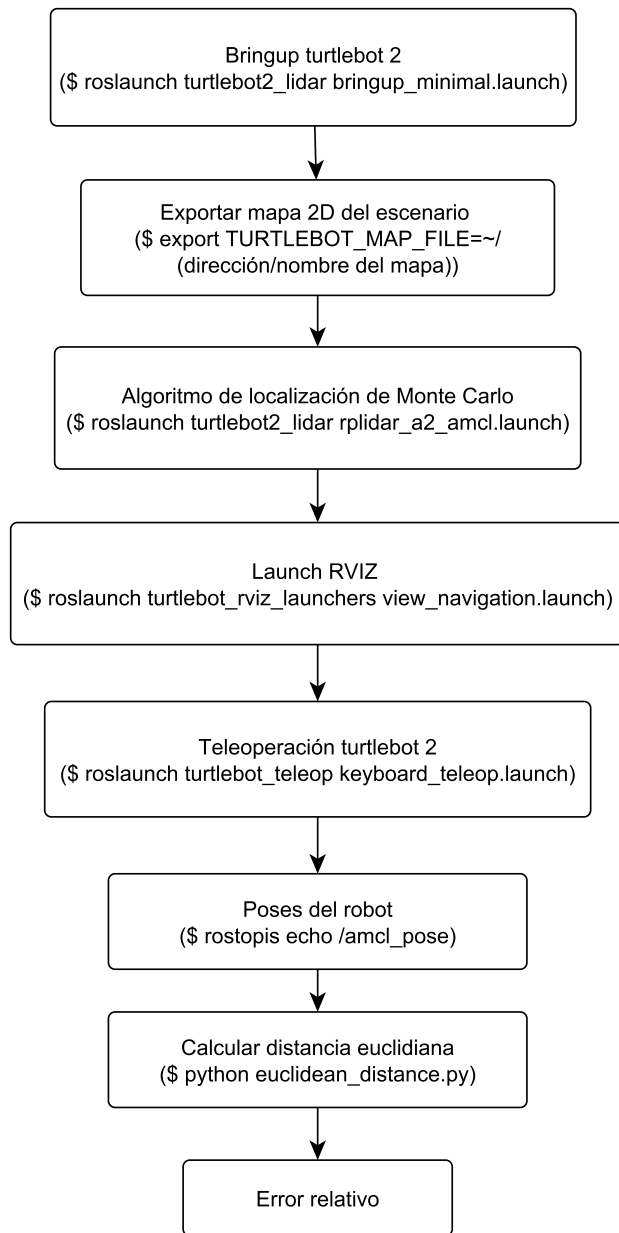


Figura 3.7: Comandos para evaluar el error relativo con el sensor laser de distancia.

# Capítulo 4

## Resultados y análisis

En esta sección del trabajo primero se expresa la información adquirida de las pruebas. Posteriormente, se estudia los resultados con el fin de comparar y analizar el performance de los sensores en la aplicación de SLAM.

### 4.1. Métrica de evaluación de rendimiento

Para el análisis comparativo de tecnologías de sensores se realiza pruebas en escenarios diferentes con el propósito de medir y comparar los tiempos en la construcción del mapa 2D y distancias de trayectorias en la exploración del robot. El primer escenario es una habitación de  $16 \text{ m}^2$  con una pared en el medio; el segundo escenario es un pasillo de una casa de aproximadamente  $16 \text{ m}^2$ . Ambos escenarios sin obstáculos.

#### **Error relativo**

La principal idea detrás del error relativo es que, en el sistema de odometría visual (VO, Visual Odometry) / odometría visual-inercial (VIO, Odometry Visual-Inercial) no posee posición global, la efectividad de la estimación se evalúa calculando la relación relativa entre poses en diferentes momentos.

Dadas las poses del robot, se mide la distancia euclidiana. Para comparar la medición

calculada y la real. El RE se calcula como

$$RE = \frac{|D_e - D_r|}{D_r}, \quad (4.1)$$

donde  $D_e$  = representa la distancia euclidiana entre poses del robot en el mapa 2D,  $D_r$  = es la distancia real del robot.

## 4.2. Descripción de resultados

A continuación se muestran los resultados de las pruebas del tiempo de mapeo de cada escenario de prueba y el cálculo del error relativo entre distancia euclideana de poses del robot en el mapa y la distancia real.

### 4.2.1. Resultados del tiempo de construcción de los mapas

En el entorno de prueba de la habitación vacía con una pared en el medio, el tiempo de mapeo con los sensores se refleja en la Tabla 4.1.

Tabla 4.1: Tiempo de generación del mapa 2D de la habitación vacía.

Sensor	Escenario	Tiempo (s)
Cámara de profundidad	Habitación vacía	256
Sensor láser de distancia	Habitación vacía	142

Los resultados del tiempo requerido en la construcción del mapa 2D del corredor de prueba usando las tecnologías de sensores se pueden observar en la Tabla 4.2. Además, se evidencian directamente los resultados del tiempo real en las Figuras A.4 y A.5 del Anexo A.

Tabla 4.2: Tiempo de construcción del mapa global 2D del pasillo de una casa.

<b>Sensor</b>	<b>Escenario</b>	<b>Tiempo (s)</b>
Cámara de profundidad	Pasillo de la casa	263
Sensor láser de distancia	Pasillo de la casa	150

## 4.2.2. Resultados de mapas con la cámara de profundidad

### 4.2.2.1. Escenario de habitación

El mapa de la Figura 4.1 representa el resultado de la aplicación de mapeo con la cámara de profundidad en la habitación vacía.

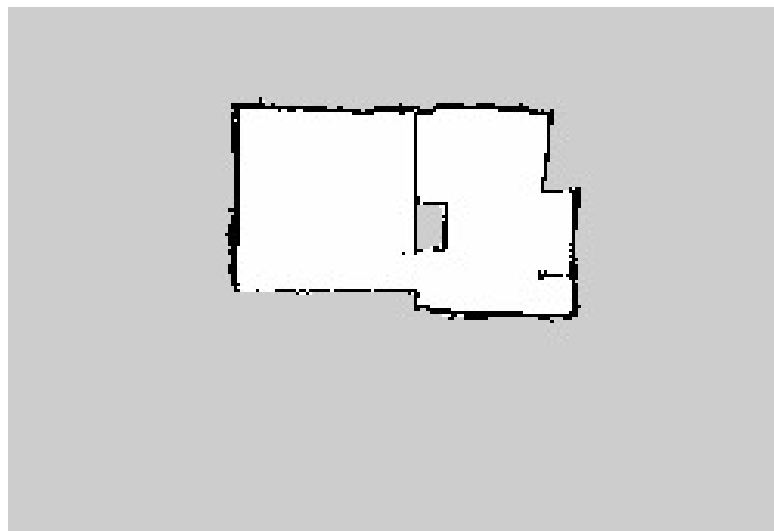


Figura 4.1: Mapa de la habitación vacía usando la cámara de profundidad.

### 4.2.2.2. Escenario de pasillo

La Figura 4.2 muestra el mapa construido con la cámara de profundidad del pasillo de una casa.

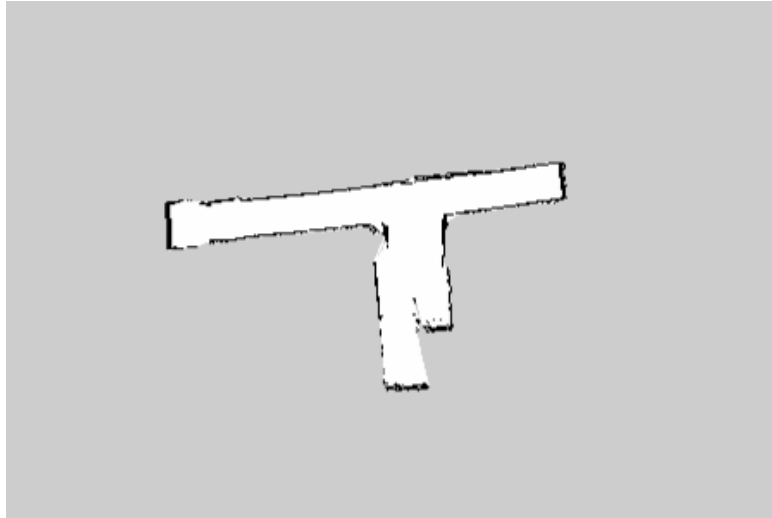


Figura 4.2: Mapa del pasillo con la cámara de profundidad.

### **4.2.3. Resultados de mapas con el sensor láser de distancia**

#### **4.2.3.1. Escenario de habitación**

El mapeo en este caso utilizando un sensor láser de distancia se muestra en la Figura 4.3.

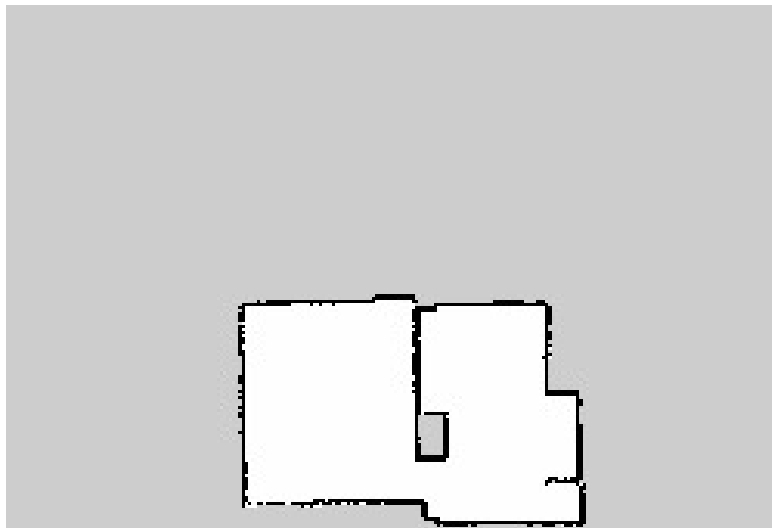


Figura 4.3: Mapa del escenario de habitación con sensor láser de distancia.

#### 4.2.3.2. Escenario de pasillo

El mapa del pasillo de la casa generado por el sensor láser de distancia cuenta con más exactitud con el ascenario real (véase la Figura 4.4).

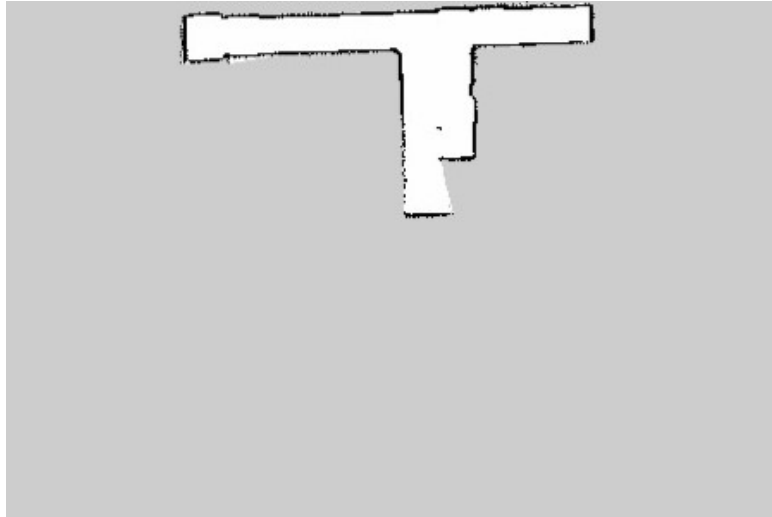


Figura 4.4: Mapa del pasillo con sensor láser de distancia.

#### 4.2.4. Resultados del error de localización

La Figura 4.5 representa el ejemplo visual de las trayectorias seleccionadas en el mapa de la habitación para evaluar el rendimiento de los sensores. (a) Visualización de la trayectoria AB en el mapa 2D desde rviz. (b) Imagen que indica la trayectoria de prueba AC.

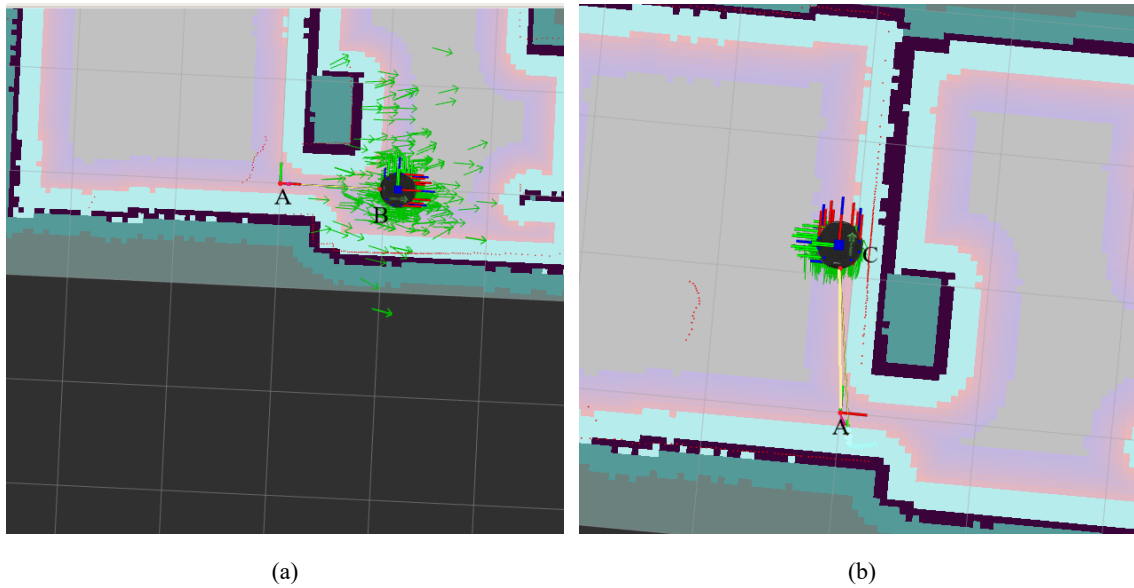


Figura 4.5: Rutas seleccionados en el mapa de la habitación vacía.(a) Trayectoria AB de la habitación vacía. (b) Trayectoria AC de la habitación vacía.

La Tabla 4.3 resume el desempeño de la cámara de profundidad y el sensor láser de distancia en la prueba de la Figura 4.5 (a). Se han calculado la distancia euclidiana y el error relativo a través de 2.1 y 4.1.

Tabla 4.3: Resultados de la precisión de localización en el entorno de prueba de habitación vacía en la trayectoria AB.

Sensor	Escenario	Trayectoria	$D_r(m)$	$D_e(m)$	Error relativo (%)
Cámara de profundidad	Habitación vacía	AB	1.00	0.979	2.10
Sensor láser de distancia	Habitación vacía	AB	1.00	1.01	1.01

Las pruebas del sensor en la trayectoria AC de la habitación vacía evidencian un error en la precisión de localización menor al 2.1 % (véase la Tabla 4.4).

Tabla 4.4: Resultados de la precisión de localización en la trayectoria AC en el entorno de prueba de habitación vacía.

Sensor	Escenario	Trayectoria	$D_r(m)$	$D_e(m)$	Error relativo (%)
Cámara de profundidad	Habitación vacía	AC	1.20	1.25	4.01
Sensor láser de distancia	Habitación vacía	AC	1.20	1.23	2.27

La Figura 4.6 señala las trayectorias en el escenario de prueba para los sensores en comparación. Se usa el pasillo de la casa visto desde el visualizador 3D rviz. (a) Una de las trayectorias de prueba del robot está representada por puntos de referencia (poses del robot en el mapa), la trayectoria AB. (b) La visualización de la trayectoria de la posición A a la C.

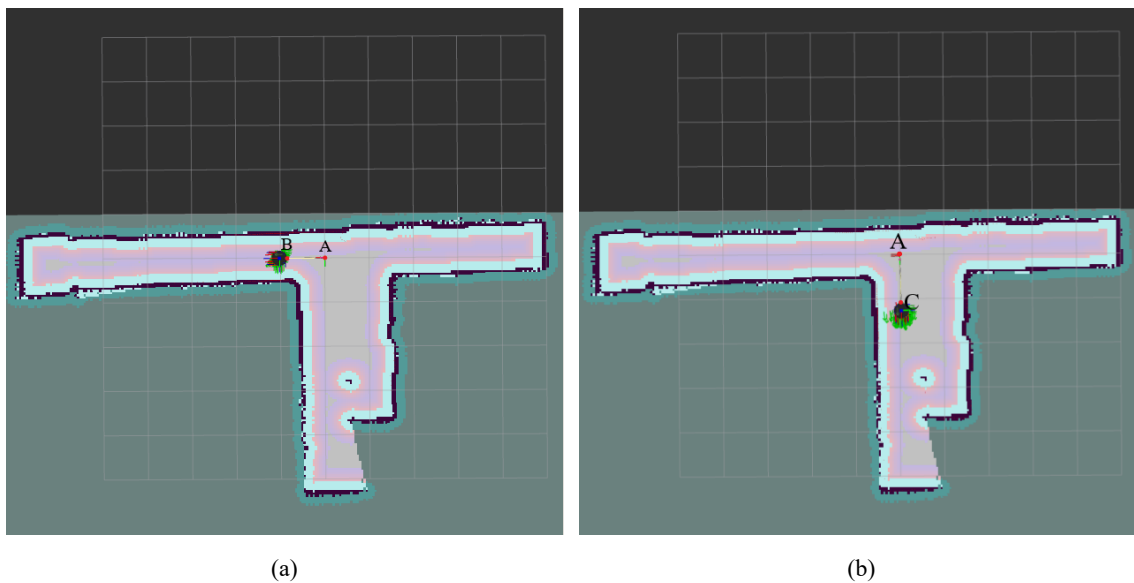


Figura 4.6: Trayectorias seleccionados en el mapa del pasillo de la casa. (a) Trayectoria AB del pasillo de una casa. (b) Trayectoria AC del pasillo de una casa.

La Tabla 4.5 muestran los resultados de las pruebas de localización al implementar tanto la cámara de profundidad como el sensor láser de distancia en las mismas condiciones. La evaluación de la ruta AB en el pasillo de una casa.



Tabla 4.5: Resultados del error de localización en la trayectoria AB del pasillo de la casa.

<b>Sensor</b>	<b>Escenario</b>	<b>Trayectoria</b>	$D_r(m)$	$D_e(m)$	<b>Error relativo (%)</b>
Cámara de profundidad	Pasillo de una casa	AB	1.00	0.978	2.19
Sensor láser de distancia	Pasillo de una casa	AB	1.00	1.01	1.43

Los sensores se pusieron a prueba en el trayectoria AC del pasillo de una casa como se mira en la Figura 4.6 (b), los resultados se resume en la Tabla 4.6

Tabla 4.6: Resultados del error de localización en el pasillo de la casa en la trayectoria AC.

<b>Sensor</b>	<b>Escenario</b>	<b>Trayectoria</b>	$D_r(m)$	$D_e(m)$	<b>Error relativo (%)</b>
Cámara de profundidad	Pasillo de una casa	AC	1.20	1.25	4.19
Sensor láser de distancia	Pasillo de una casa	AC	1.20	1.16	3.36

La Tabla 4.7 resume los resultados de las pruebas experimentales de la eficiencia temporal y la precisión de la cámara de profundidad y el sensor láser de distancia.

Tabla 4.7: Resultados del análisis comparativo del rendimiento de SLAM bajo las tecnologías de cámara de profundidad y de sensor láser de distancia.

<b>Rendimiento de tecnologías de sensores</b>			
<b>Criterio</b>	<b>Sensor</b>	<b>Tiempo de mapeo (s)</b>	<b>Tiempo de mapeo (s) de otros trabajos</b>
Tiempo de mapeo	Cámara de profundidad	260	hasta 3 veces más eficiente que la cámara
	Sensor láser de distancia	156	
<b>Criterio</b>	<b>Sensor</b>	<b>Error relativo (%)</b>	<b>Error (%) de otros trabajos</b>
Error de localización	Cámara de profundidad	3.12	1-5
	Sensor láser de distancia	2.02	

### 4.3. Análisis de los resultados

Este trabajo ha presentado un análisis comparativo de tecnologías de sensores que se utilizan en SLAM. El rendimiento de los sensores en SLAM se evalúa a partir del mapeo en tiempo real y de la precisión de la localización desde una perspectiva de distancia. Las pruebas se realizaron en dos escenarios interiores.

Cuando se examina la Tabla 4.7 se evidencia que el sensor láser de distancia logró una eficiencia de tiempo de mapeo de hasta el doble en comparación con la cámara de profundidad. Es decir, el sensor láser de distancia tiene un mayor rendimiento (menor tiempo) en el proceso de construcción del mapa 2D del escenario de prueba. Por el contrario, el rendimiento de la cámara de profundidad en el mapeo en tiempo real es menor (mayor tiempo). Además, hay algunos errores en los mapas y espacios no explorados (color blanco) que indican la influencia del ángulo reducido de visión de la cámara Kinect.

Según los resultados de las pruebas, el rendimiento de los sensores en términos de error de

localización no superan el 5%. Sin embargo, el sensor láser de distancia es más preciso (menor error relativo). El sensor láser de distancia tiene una precisión del 97.98%. Genera trayectorias más ciertas a las mediciones reales debido a que el sensor LiDAR tiene un escaneo de 360 grados y un rango operativo menor a 0.2 metros. Por otro lado, los errores de localización son más comunes en ambientes interiores cuando se utiliza la cámara de profundidad. En este contexto, la cámara de profundidad tiene una precisión del 96.88%.

# Capítulo 5

## Conclusiones y recomendaciones

### 5.1. Conclusiones

- Se realizó una comparativa cuantitativa de la precisión de localización y mapeo simultáneos de la cámara de profundidad frente al sensor láser de distancia en interiores como escenarios experimentales.
- El rendimiento de los sistemas *visual SLAM* y *LiDAR SLAM* se evaluaron mediante criterios de tiempo de mapeo y las mediciones de error de las distancias estimadas frente a las distancias reales con base en los criterios mencionados en el capítulo 2 del presente trabajo. Un sistema de SLAM eficiente y preciso genera mapas rápidos y estimaciones precisas de las trayectorias del robot.
- La plataforma robótica Turtlebot 2 es un sistema adaptable para agregar más de un sensor. En este contexto, se agregó al sistema SLAM un sensor láser de distancia. Además, el robot tiene su repositorio en ROS lo que permite configurar el algoritmo Gmapping y el paquete AMCL para el mapeo y localización del robot con el sensor láser de distancia.
- Los resultados de las pruebas comparativas de los sensores en sistemas SLAM, han demostrado que el sensor láser de distancia es más eficiente en la aplicación de mapeo en tiempo real en comparación con la cámara de profundidad. Es decir, el tiempo que emplea el sensor Kinect en el proceso de mapeo es de casi el doble en relación con el sensor LiDAR. Asimismo, el desempeño de los sensores permite la localización del robot en el

mapa 2D. No obstante, el sensor LiDAR con una precisión del 97,98 % estima trayectorias con menos errores en relación con las mediciones reales.

## **5.2. Recomendaciones**

- Adaptar los paquetes ROS utilizados con la cámara de profundidad para que funcione con el sensor láser de distancia. Se agrega la transformación de TF estática a los archivos del robot.
- Antes de usar los diferentes paquetes de ROS con el sensor láser de distancia verifique el permiso del puerto serial del sensor.

## **5.3. Trabajo a futuro**

- El análisis comparativo de diferentes tecnologías de sensores para SLAM se puede mejorar al implementar el uso de inteligencia artificial para estudiar y examinar automáticamente la información de las trayectorias del robot. Además, realizar pruebas experimentales en escenarios con obstáculos.

# Bibliografía

- [1] R. Francisco, V. Francisco, and Carlos Llopis-Albert, “A Review of mobile robots: Concepts, methods, theoretical framework, and applications. International Journal of Advanced Robotic Systems,” *Int J Adv Robot Syst*, Apr. 2019, doi: 10.1177/1729881419839596.
- [2] E. Uslu, F. Cakmak, M. Balcilar, A. Akinci, M. F. Amasyali, and S. Yavuz, “Implementation of frontier-based exploration algorithm for an autonomous robot,” in 2015 International Symposium on Innovations in Intelligent Systems and Applications (INISTA), Sep. 2015, pp. 1–7. doi: 10.1109/INISTA.2015.7276723.
- [3] A. Gaona, D. Loza, B. Cortez, and L. Segura, “Diseño y Construcción de una plataforma de robot móvil teleoperada a bajo costo para interiores,” Jun. 2018. doi: 10.24133/cctes-pe.v13i1.794.
- [4] J. Alvarado, “Exploración de fronteras para robots móviles en interiores,” Trabajo de fin de grado, Universidad Técnica del Norte, Ibarra, 2021. Accessed: Nov. 06, 2022. [Online]. Available: <http://repositorio.utn.edu.ec/handle/123456789/11288>
- [5] Q. Zou, Q. Sun, L. Chen, B. Nie, and Q. Li, “A Comparative Analysis of LiDAR SLAM-Based Indoor Navigation for Autonomous Vehicles,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 7, 2022, doi: 10.1109/TITS.2021.3063477.
- [6] I. Z. Ibragimov and I. M. Afanasyev, “Comparison of ROS-based visual SLAM methods in homogeneous indoor environment,” in 2017 14th Workshop on Positioning, Navigation and Communications (WPNC), Oct. 2017, pp. 1–6. doi: 10.1109/WPNC.2017.8250081.

- [7] Q. Jin, Y. Liu, Y. Man, and F. Li, “Visual SLAM with RGB-D Cameras,” in 2019 Chinese Control Conference (CCC), Jul. 2019, pp. 4072–4077. doi: 10.23919/ChiCC.2019.8865270.
- [8] M. U. Khan, S. A. A. Zaidi, A. Ishtiaq, S. U. R. Bukhari, S. Samer, and A. Farman, “A Comparative Survey of LiDAR-SLAM and LiDAR based Sensor Technologies,” in 2021 Mohammad Ali Jinnah University International Conference on Computing (MAJICC), Jul. 2021, pp. 1–8. doi: 10.1109/MAJICC53071.2021.9526266.
- [9] M. Mielle, M. Magnusson, and A. J. Lilienthal, “A comparative analysis of radar and lidar sensing for localization and mapping,” in 2019 European Conference on Mobile Robots (ECMR), Sep. 2019, pp. 1–6. doi: 10.1109/ECMR.2019.8870345.
- [10] S.-H. Chan, P.-T. Wu, and L.-C. Fu, “Robust 2D Indoor Localization Through Laser SLAM and Visual SLAM Fusion,” in 2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Oct. 2018, pp. 1263–1268. doi: 10.1109/SMC.2018.00221
- [11] L. Huang, “Review on LiDAR-based SLAM Techniques,” in 2021 International Conference on Signal Processing and Machine Learning (CONF-SPML), Nov. 2021, pp. 163–168. doi: 10.1109/CONF-SPML54095.2021.00040.
- [12] M. Korkmaz, A. Durdu, and Y. E. Tusun, “Sensor Comparison for a Real-Time SLAM Application,” *International Journal of Information and Electronics Engineering*, vol. 8, no. 1, pp. 1–4, Mar. 2018, doi: 10.18178/IJIEE.2018.8.1.684.
- [13] A. Escalante, “Análisis del método de odometría Lidar LOAM,” Trabajo de fin de grado, Universidad Técnica Superior de Ingeniería Universidad de Sevilla, Sevilla, 2016.
- [14] S. Gobhinath, K. Anandapoorani, K. Anitha, D. D. Sri, and R. DivyaDharshini, “Simultaneous Localization and Mapping [SLAM] of Robotic Operating System for Mobile Robots,” in 2021 7th International Conference on Advanced Computing and Communication Systems (ICACCS), Mar. 2021, pp. 577–580. doi: 10.1109/ICACCS51430.2021.9441758.

- [15] B. M. F. da Silva, R. S. Xavier, T. P. do Nascimento, and L. M. G. Goncalves, “Experimental evaluation of ROS compatible SLAM algorithms for RGB-D sensors,” in 2017 Latin American Robotics Symposium (LARS) and 2017 Brazilian Symposium on Robotics (SBR), Nov. 2017, pp. 1–6. doi: 10.1109/SBR-LARS-R.2017.8215331.
- [16] İ. Ünal, S. Sözer, and Ö. Kabaş, “Usability of Kinect Camera on Outdoor Conditions for Agricultural Mobile Robots,” in ICENS 4th International Conference on Engineering and Natural Science, Nov. 2020.
- [17] K. Kamarudin, S. Mamduh, A. Shakaff, and A. Zakaria, “Performance Analysis of the Microsoft Kinect Sensor for 2D Simultaneous Localization and Mapping (SLAM) Techniques,” *Sensors*, vol. 14, no. 12, pp. 23365–23387, Dec. 2014, doi: 10.3390/s141223365.
- [18] A. Zhang and M. Maher Atia, “Comparison of 2D Localization Using Radar and LiDAR in Long Corridors,” in 2020 IEEE SENSORS, Oct. 2020, pp. 1–4. doi: 10.1109/SENSORS47125.2020.9278684.
- [19] Y.-S. Shin, Y. S. Park, and A. Kim, “Direct Visual SLAM Using Sparse Depth for Camera-LiDAR System,” in 2018 IEEE International Conference on Robotics and Automation (ICRA), IEEE, May 2018, pp. 5144–5151. doi: 10.1109/ICRA.2018.8461102.
- [20] N. Zarrabi, R. Fesharakifard, and M. B. Menhaj, “Robot localization performance using different SLAM approaches in a homogeneous indoor environment,” in 2019 7th International Conference on Robotics and Mechatronics (ICRoM), IEEE, Nov. 2019, pp. 338–344. doi: 10.1109/ICRoM48714.2019.9071902.
- [21] X. Kang, J. Li, X. Fan, and W. Wan, “Real-Time RGB-D Simultaneous Localization and Mapping Guided by Terrestrial LiDAR Point Cloud for Indoor 3-D Reconstruction and Camera Pose Estimation,” *Applied Sciences*, vol. 9, no. 16, p. 3264, Aug. 2019, doi: 10.3390/app9163264.



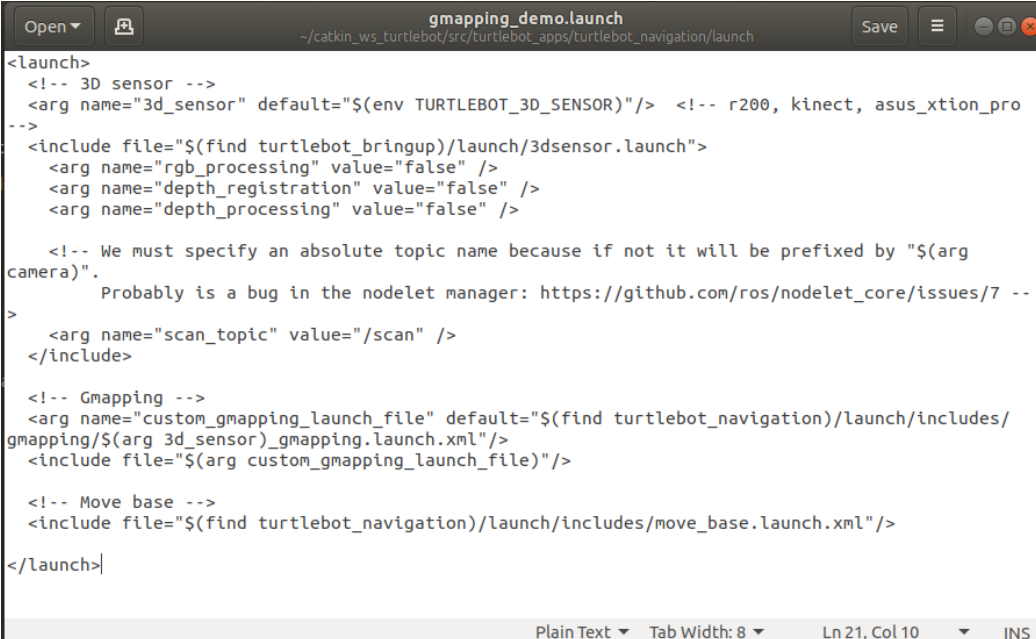
- [22] M. S. A. Khan, D. Hussain, K. Naveed, U. S. Khan, I. Q. Mundial, and A. Bin Aqeel, "Investigation of Widely Used SLAM Sensors Using Analytical Hierarchy Process," *J Sens*, vol. 2022, pp. 1–15, Jan. 2022, doi: 10.1155/2022/5428097.
- [23] G. C. Cardarilli, L. Di Nunzio, R. Fazzolari, A. Nannarelli, M. Re and S. Spanò, "N - Dimensional Approximation of Euclidean Distance," in *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 67, no. 3, pp. 565-569, March 2020, doi: 10.1109/TC-SII.2019.2919545.
- [24] F. Pineda Torres, "Técnicas de slam con filtros probabilísticos; caracterización y resultados en robots móviles," 2019. Accessed: Mar. 04, 2023. [Online]. Available: <https://www.fesc.edu.co/Revistas/OJS/index.php/mundofesc/article/view/408>
- [25] Turtlebot, "Learn Turtlebot with ROS. ," 2015. [Online]. Available: <http://learn.turtlebot.com/2015/02/01/9/> (accessed Jan. 09, 2023).
- [26] Dabit Industries, "Turtlebot Tutorials," 2018. [Online]. Available: <https://github.com/dabit-industries/turtlebot2-tutorials> (accessed Jan. 09, 2023).
- [27] Shanghai Slamtec, Rplidar ros. Zhangjiang. 2015. [Online]. Available: [https://github.com/Slamtec/rplidar\\_ros.git](https://github.com/Slamtec/rplidar_ros.git)
- [28] Z. Zhang and D. Scaramuzza, "A Tutorial on Quantitative Trajectory Evaluation for Visual(-Inertial) Odometry," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct. 2018, pp. 7244–7251. doi: 10.1109/IROS.2018.8593941.

# **Anexos**

# Anexo A

## Construcción de un mapa (SLAM)

Para mapear se utiliza el archivo launch igual a la Figura A.1 que por defecto se usa con la cámara Kinect con el paquete “gmapping” de ROS. El mismo archivo se editó para ser usado con el sensor LiDAR, denominado “gmapping\_lidar.launch”.



```
<launch>
  <!-- 3D sensor -->
  <arg name="3d_sensor" default="$(env TURTLEBOT_3D_SENSOR)"/> <!-- r200, kinect, asus_xtion_pro -->
  <!-->
  <include file="$(find turtlebot_bringup)/launch/3dsensor.launch">
    <arg name="rgb_processing" value="false" />
    <arg name="depth_registration" value="false" />
    <arg name="depth_processing" value="false" />

    <!-- We must specify an absolute topic name because if not it will be prefixed by "$(arg
camera)".
Probably is a bug in the nodelet manager: https://github.com/ros/nodelet_core/issues/7 --
  >
  <arg name="scan_topic" value="/scan" />
  </include>

  <!-- Gmapping -->
  <arg name="custom_gmapping_launch_file" default="$(find turtlebot_navigation)/launch/includes/
gmapping/$(arg 3d_sensor)_gmapping.launch.xml"/>
  <include file="$(arg custom_gmapping_launch_file)"/>

  <!-- Move base -->
  <include file="$(find turtlebot_navigation)/launch/includes/move_base.launch.xml"/>
</launch>
```

Figura A.1: Launch de mapeo con cámara Kinect.

En la Figura A.2 se puede observar la construcción del mapa 2D de la habitación experimental

desde el visualizador rviz. Una vez terminado el mapeo, se guarda el mapa. De igual manera, la Figura A.3 representa el proceso del mapeo del pasillo con el sensor láser de distancia.

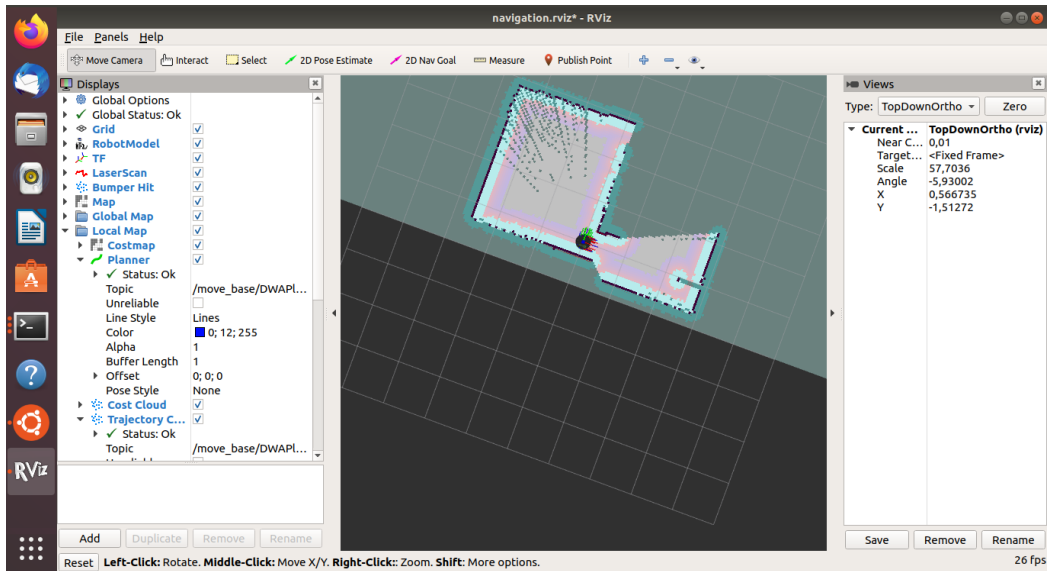


Figura A.2: Mapeo de la habitación vacía con cámara Kinect.

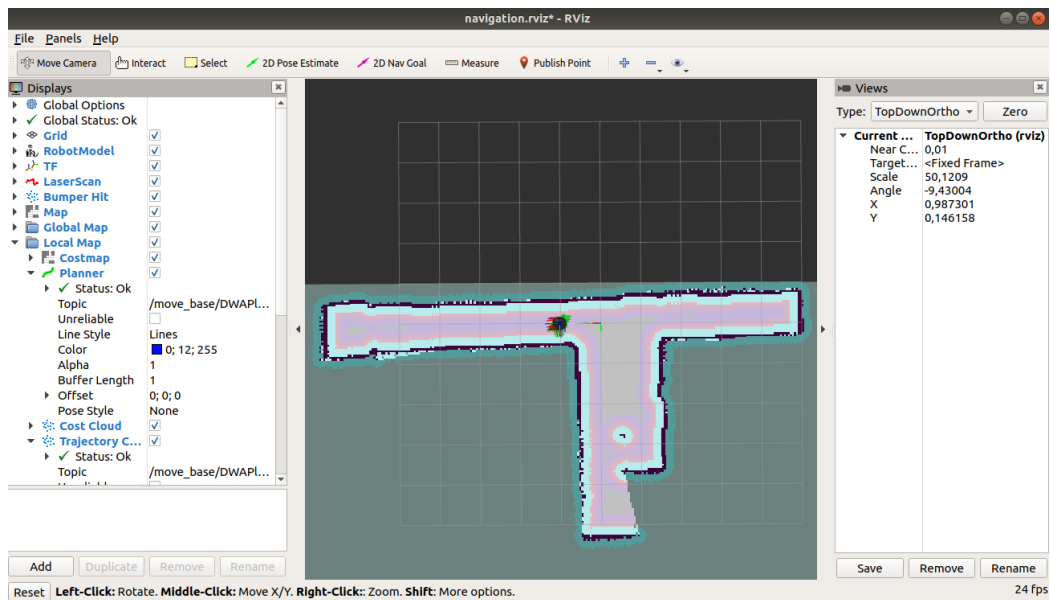


Figura A.3: Mapeo del pasillo de una casa con el sensor LiDAR .

Las Figura A.4 y A.5 señalan el tiempo de mapeo en el pasillo de prueba con los diferentes

sensores en comparación.

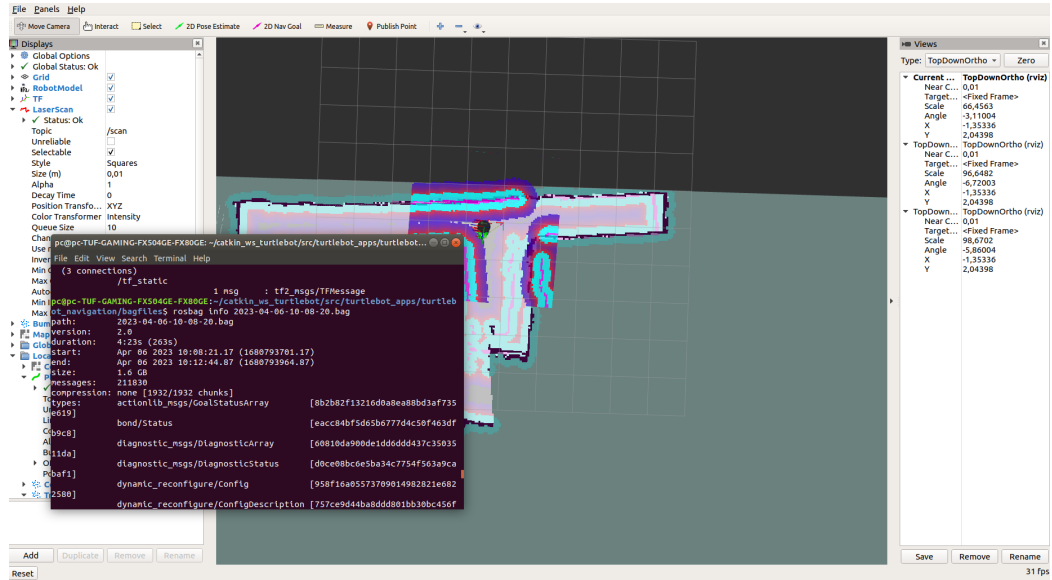


Figura A.4: Mapa del pasillo de una casa con la cámara de profundidad. Tiempo de mapeo de 263 segundos.

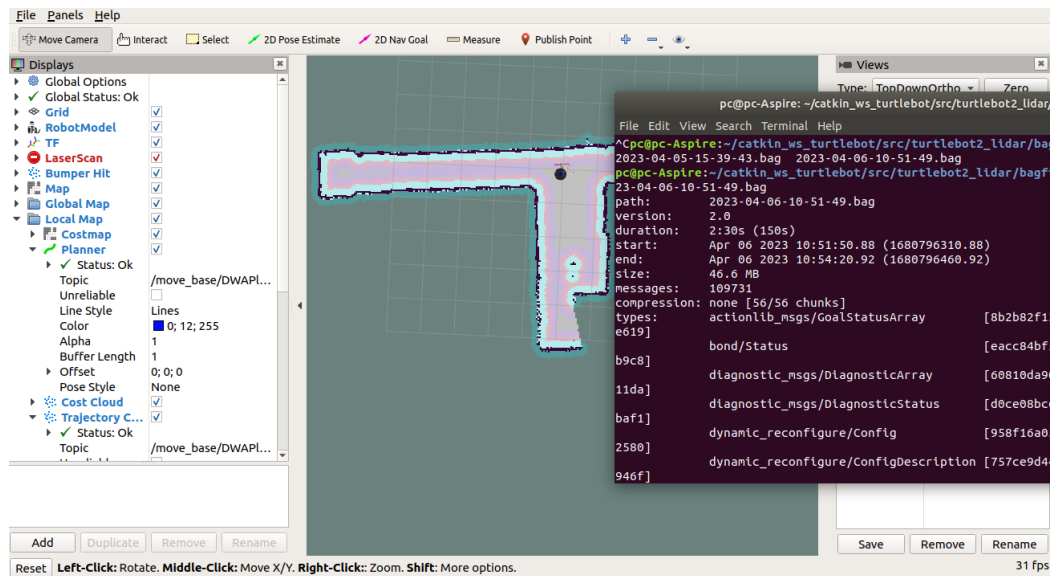


Figura A.5: Mapa del pasillo de una casa con sensor láser de distancia. Tiempo de mapeo de 150 segundos.

## Anexo B

# Navegación autónoma

Luego de que el mapa es generado, es posible la navegación autónoma en el escenario mapeado. Para ello, el archivo launch se adaptado al sensor láser de distancia, como en la Figura B.1



```
<launch>
  <!-- sensor -->
  <arg name="lidar_sensor" default="$(env TURTLEBOT_3D_SENSOR)"/>
  <include file="$(find rplidar_ros)/launch/rplidar.launch"/>

  <!-- Map server -->
  <arg name="map_file" default="$(env TURTLEBOT_MAP_FILE)"/>
  <node name="map_server" pkg="map_server" type="map_server" args="$(arg map_file)" >
    <param name="frame_id" value="map" />
  </node>

  <!-- AMCL -->
  <arg name="custom_amcl_launch_file" default="$(find turtlebot_navigation)/launch/includes/amcl/
amcl.launch.xml"/>
  <arg name="initial_pose_x" default="0.0"/> <!-- Use 17.0 for willow's map in simulation -->
  <arg name="initial_pose_y" default="0.0"/> <!-- Use 17.0 for willow's map in simulation -->
  <arg name="initial_pose_a" default="0.0"/>
  <include file="$(arg custom_amcl_launch_file)">
    <arg name="initial_pose_x" value="$(arg initial_pose_x)"/>
    <arg name="initial_pose_y" value="$(arg initial_pose_y)"/>
    <arg name="initial_pose_a" value="$(arg initial_pose_a)"/>
  </include>

  <!-- Move base -->
  <arg name="custom_param_file" default="$(find turtlebot_navigation)/param/$(arg
lidar_sensor)_costmap_params.yaml"/>
  <include file="$(find turtlebot_navigation)/launch/includes/move_base.launch.xml">
    <arg name="custom_param_file" value="$(arg custom_param_file)"/>
  </include>

  <node pkg="tf" type="static_transform_publisher" name="base_laser_link_to_laser" args="0 0 0 0
0 base_laser_link laser 100"/>
</launch>
```

Figura B.1: Launch de la navegación con el sensor LiDAR.