



UNIVERSIDAD DE LA REPÚBLICA
FACULTAD DE INGENIERÍA



Dispositivo de asistencia ante el congelamiento de la marcha en pacientes con Parkinson

Guillermo Trinidad

Tesis final para obtener el título de Ingeniero en Computación

TUTORES

Gonzalo Tejera..... Universidad de la República
Mercedes Marzoa Universidad de la República

TRIBUNAL

Ewelina Bakala Universidad de la República
Eduardo Fernandez Universidad de la República
Federico Andrade Universidad de la República

Montevideo
jueves 10 diciembre, 2020

Dispositivo de asistencia ante el congelamiento de la marcha en pacientes con Parkinson, Guillermo Trinidad Barnech.

Esta tesis fue preparada en L^AT_EX usando la clase iietesis (v1.1).
Compilada el jueves 10 diciembre, 2020.

Resumen

El Parkinson es un trastorno cerebral caracterizado por el temblor y la dificultad en el movimiento, la marcha y la coordinación que afecta a un 2-3% de la población mayor de 65 años. Los pacientes entre otros síntomas padecen del llamado “congelamiento de la marcha”, caracterizado por episodios de detenimiento abrupto del movimiento. A la fecha no se cuenta con un tratamiento efectivo contra el congelamiento, pero en los últimos años se ha demostrado que una manera de disminuir la cantidad y duración de los episodios es la utilización de dispositivos que provean al paciente información espacial y/o temporal que guíen la marcha. Las razones detrás de la efectividad de estas técnicas no son claras, pero suele atribuirse a que estos llevan la atención de los pacientes a la marcha, evitando que estos realicen una doble tarea, que es un disparador conocido de congelamiento.

Los estímulos presentados por los dispositivos actuales no toman en cuenta el entorno en que se encuentran, generando que sea el paciente quien deba decidir los cambios de dirección necesarios tanto para dirigirse hacia un lugar específico como para evitar los obstáculos que se presenten en el camino. Es en estos momentos de toma de decisión que se genera una doble tarea y los episodios de congelamiento pueden suceder.

En este proyecto se presenta un dispositivo de señalización capaz de guiar a los pacientes hacia destinos de su elección por trayectorias que toman en cuenta los obstáculos presentes en el camino y los cambios de dirección necesarios. Utilizando técnicas de mapeo, localización y planificación tomadas de la robótica autónoma, el sistema presenta líneas en el suelo a través de lentes de realidad aumentada. Una vez que los usuarios seleccionan un destino estos solo deben pisar sobre las líneas generadas para dirigirse hacia allí. Con estos estímulos se busca minimizar las situaciones donde la atención sea desviada de la marcha y así evitar los episodios de congelamiento.

Tabla de contenidos

Resumen	I
1. Introducción	1
1.1. Presentación del problema y contexto	1
1.2. Estructura del documento	3
2. Estado del arte	5
2.1. Introducción	5
2.2. Estímulo continuo utilizando bastón con láser	6
2.3. Estímulo continuo utilizando Google Glass	7
2.4. Estímulo no continuo utilizando Moverio Bt-200	8
3. Marco teórico	11
3.1. SLAM	11
3.1.1. Grillas de ocupación	12
3.1.2. SLAM visual	12
3.1.3. RTAB-Map	14
3.2. Robot Operating System	16
3.2.1. Stack de navegación de ROS	16
3.2.2. Transform	17
4. Solución propuesta	19
4.1. Diseño de los estímulos	19
4.2. Mapeo del entorno	20
4.2.1. Cámara binocular ZED	21
4.2.2. RTAB-Map	21
4.3. Planificación de trayectorias	22
4.4. Generación de estímulos	24
4.4.1. Construcción	24
4.4.2. Renderizado	24
4.5. Interfaz	28
4.5.1. Proyección de las líneas	28
4.5.2. Guardado de puntos	29
4.5.3. Elección de punto destino	29
4.6. Despliegue	31

Tabla de contenidos

5. Experimentación	35
5.1. Prueba de mapeo y localización	35
5.1.1. Mapeo	36
5.1.2. Localización	37
5.1.3. Análisis de resultados	43
5.2. Prueba de integración	44
6. Conclusiones y trabajos a futuro	47
6.1. Conclusiones	47
6.2. Trabajos a futuro	48
Referencias	49
Glosario	55
Índice de tablas	57
Índice de figuras	58

Capítulo 1

Introducción

1.1. Presentación del problema y contexto

El Parkinson es un trastorno cerebral caracterizado por el temblor y la dificultad en el movimiento, la marcha y la coordinación que afecta a un 2-3% de la población mayor de 65 años [1]. Los pacientes entre otros síntomas padecen del llamado “congelamiento de la marcha”, caracterizado por episodios de detención abrupto del movimiento, trayendo aparejado también la dificultad para iniciarlo [2]. Quienes lo padecen suelen describirlo como “tener los pies pegados al suelo” [3]. Una de sus más graves consecuencias son las caídas, que generan temor, falta de confianza y la necesidad continua de auxilio. De esta manera, en el paciente con Parkinson se va gestando un mecanismo de autoexclusión, deteriorando así su salud y su vida social.

El congelamiento de la marcha **FoG** (por su sigla en inglés *Freezing of Gait*) se presenta en un 70% de los pacientes en etapas avanzadas [4,5] y en un 26% de los pacientes en etapas tempranas [6].

A la fecha, la fisiopatología detrás de los episodios de FoG sigue sin ser clara, sin embargo existen algunas opciones para su tratamiento. Una forma de categorizar los tipos de FoG es según su respuesta a la medicación dopaminérgica [7]:

1. FoG sensible a la dopamina: el síntoma se presenta sólo cuando el paciente no está bajo medicación
2. FoG resistente a la dopamina: el síntoma se presenta sin importar si el paciente está o no medicado
3. FoG inducida por la dopamina: el síntoma se presenta sólo cuando el paciente está bajo medicación

En los casos 1 y 3, se suele recomendar un tratamiento que varíe la dosis de dopamina (aumentando en 1 y disminuyendo en 3 [8]) pero esto acarrea complicaciones con respecto a los demás síntomas de la enfermedad [9]. Por esta razón suelen

Capítulo 1. Introducción

ser recomendados tratamientos quirúrgicos [10]. Si bien estos no atacan directamente el FoG, reducen otros síntomas y permiten de esta manera más libertad en la dosis de medicación dopaminérgica a administrar. Sin embargo, estas cirugías suelen ser invasivas y sumamente costosas (en el orden de las decenas de miles de dólares [11]). Esta situación motiva la búsqueda de formas alternativas de tratar el FoG, como pueden ser los dispositivos de asistencia.

Durante los últimos años se ha demostrado que los dispositivos de señalización ayudan a mejorar la caminata y reducir la cantidad de episodios de FoG [12]. La señalización puede definirse como la utilización de estímulos externos que provean información temporal o espacial para facilitar el inicio y la continuidad de la marcha [13]. Existen tres modalidades de señalización ampliamente reportadas en la literatura actual: visual [14,15], auditiva [16,17] y somatosensorial [18,19]. A su vez existen estudios que utilizan combinaciones de diferentes tipos de estímulos [20].

Las razones detrás de la efectividad de estas técnicas para aliviar el FoG siguen sin estar claras, pero estudios anteriores sugieren lo siguiente:

1. La señalización puede compensar el defectuoso funcionamiento del generador rítmico interno de los ganglios basales, afectando la coordinación y ejecución del movimiento [21, 22]. De esta manera el paciente puede acoplar su movimiento a las señales provistas.
2. Las señales visuales pueden brindar información espacial para ayudar a controlar el largo del paso y guiar el movimiento en la marcha [23].
3. Los dispositivos de asistencia pueden dirigir la atención de los pacientes a la acción de caminar, siendo esto lo que aminora la presencia del síntoma [24].
4. Los estímulos visuales y somatosensoriales pueden estimular al paciente a tomar consciencia sobre el posicionamiento de sus extremidades (entradas propioceptivas) durante la caminata, ayudando así a aliviar el síntoma [25].

Ciertos estudios señalan que el FoG ocurre con mayor frecuencia cuando existen (I) condicionantes ambientales que exijan cambios en la velocidad y sentido de la marcha y/o (II) variaciones en el patrón del piso o paredes [26].

Este fenómeno podría deberse a que en el primer caso el paciente debe enfocar su atención al ajuste de su trayectoria/velocidad, dejando de lado la tarea de la caminata en sí misma mientras en el segundo existen estímulos que pueden actuar como distracciones.

Las situaciones mencionadas motivan la construcción de dispositivos de señalización que minimicen la cantidad de conflictos a resolver por el paciente, pudiendo este concentrarse únicamente en la acción de caminar.

1.2. Estructura del documento

En 2019 se aprueba el proyecto CSIC “Diseño de un dispositivo para fomentar la continuidad de la marcha en pacientes con Parkinson”, el cual busca la construcción de un sistema de orientación espacial que promueva la continuidad en la marcha de pacientes con Parkinson, brindando estímulos generados teniendo en cuenta el entorno donde este se encuentra. Se busca con este dispositivo que durante la marcha el paciente no deba distraer su atención a esquivar obstáculos o cambiar de dirección.

Enmarcado en esta iniciativa, este proyecto busca diseñar y evaluar la efectividad de un dispositivo que genere estímulos capaces de guiar a un paciente dentro de entornos cambiantes y con presencia de obstáculos. Para lograrlo, este debe identificar los elementos a su alrededor, calcular trayectorias que los tomen en cuenta y generar estímulos que permitan seguirlos. El objetivo a cumplir es lograr que el paciente pueda concentrarse únicamente en las señales brindadas durante la marcha, evitando distraer su atención.

1.2. Estructura del documento

Este documento se compone de seis capítulos. Los primeros tres capítulos están relacionados a la etapa de investigación y diseño de la solución y los últimos tres a la implementación del proyecto.

El resto del documento queda organizado de la siguiente forma, en el capítulo dos se presenta el estado del arte, donde se describen las principales soluciones actuales de dispositivos de asistencia al síntoma del congelamiento en la marcha. El capítulo tres presenta los aspectos teóricos que enmarcan el trabajo, en particular se presentan las herramientas más importantes utilizadas, junto con un pequeño análisis de su funcionamiento.

En el cuarto capítulo se presenta la solución propuesta, donde se describe el trabajo realizado junto con la justificación de las decisiones tomadas. Este se divide en varias secciones que representan cada uno de los componentes desarrollados a lo largo del proyecto.

En el quinto capítulo se presentan los experimentos realizados, donde se describe el caso de estudio utilizado para la validación del sistema. Por último, en el capítulo seis se presenta la discusión final sobre los resultados obtenidos y las principales líneas de investigación a desarrollar a partir de lo aquí realizado.

Capítulo 2

Estado del arte

2.1. Introducción

Este capítulo es una síntesis del documento de estado del arte realizado en el marco de este proyecto [27], donde se presentan los artículos más relevantes en la temática a abordar, realizando un análisis sobre sus resultados y conclusiones.

Los tres tipos de señalización presentados en trabajos previos (visual, auditiva y somatosensorial) han demostrado ser efectivos para aliviar o disminuir la cantidad de episodios de FoG, siendo todos buenos candidatos para el desarrollo de esta investigación. En este trabajo se busca generar estímulos que guíen al usuario para desplazarse y aseguren que al seguirlos este no tenga que preocuparse por evitar obstáculos o considerar que cambios de dirección debe realizar. Para esto, es necesario indicar una trayectoria que ya tome en cuenta los cambios de dirección necesarios y los muestre al usuario. Una técnica conocida para guiar el paso de las personas con Parkinson consiste en colocar líneas en el suelo, donde este tenga que pisar y así logre desplazarse. Combinando esta idea con la necesidad de mostrar una trayectoria se opta por utilizar estímulos visuales (mostrando líneas perpendiculares a la trayectoria calculada), considerando esta la manera más simple de codificar la dirección en los estímulos.

El proceso de selección de artículos está basado fuertemente en el trabajo de Sweeney *et al.* [28] el cual revisa diferentes dispositivos de señalización que tengan la posibilidad de ayudar al tratamiento del FoG. Si bien este artículo releva representantes de las tres categorías de señalización (visual, auditiva y somatosensorial), teniendo en cuenta la decisión de utilizar señalización visual, se estudian solamente trabajos sobre este tipo de estímulos.

A continuación se presentan los principales trabajos.

2.2. Estímulo continuo utilizando bastón con láser

En 2010, Bryant *et al.* [29] presentaron un bastón que proyecta una línea en el suelo mediante el uso de un láser (figura 2.1). Esto imita el uso de líneas paralelas colocadas en el piso, extendiendo esta técnica a cualquier entorno.

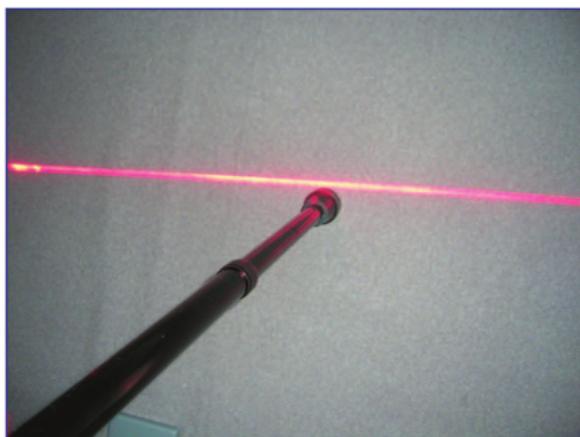


Figura 2.1: Bastón proyectando una línea roja utilizada en el trabajo desarrollado por Bryant *et al.* (Imagen tomada de [29]).

En este trabajo se compara la efectividad del láser según el color que proyecta, experimentando con rojo y verde. Para esto diseñaron dos pruebas:

1. Caminar 7.62m, girar 180° y caminar 7.62m más.
2. Realizar un giro de 360°.

Ambas pruebas fueron realizadas con un bastón que no emitía luz, uno con luz roja y otro con luz verde. A su vez repitieron las pruebas con los pacientes estando bajo medicación y sin estarlo. Se midió la cantidad de episodios de FoG (*On-FoG* estando bajo medicación y *Off-Fog* en caso contrario). En la tabla 2.1 se presentan los resultados para cada experimento.

	Sin medicación			Bajo medicación		
	Sin luz	Luz roja	Luz verde	Sin luz	Luz roja	Luz verde
Caminata	0.67 (0.52) ⁺	0.83 (0.75)*	0.00 (0.00)^{+*}	0.29 (0.49)	0.14 (0.38)	0.00 (0.00)
Giro	2.14 (1.35) ⁺	1.43 (0.98)*	0.29 (0.49)⁺	1.00 (1.00)	0.57 (0.79)	0.57 (0.79)

Tabla 2.1: Numero de episodios de FoG utilizando un bastón normal vs. uno que proyecta una línea verde vs. uno con luz roja en las tareas de (1) realizar una caminata de 15.24m con un giro de 180° y (2) realizar un giro de 360° en el lugar [29]. Se muestran los resultados en formato **promedio (desviación estándar)**. Con * se marcan los casos de luz roja vs luz verde con p valor < 0.05 y con ⁺ los casos sin luz vs luz verde con p valor < 0.05

2.3. Estímulo continuo utilizando Google Glass

Se puede observar que el número de episodios disminuye siempre que se utiliza estimulación visual (excepto en el caso de los Off-FoG con la luz roja) y que utilizando el láser de color verde se obtienen resultados significativamente mejores. Los autores atribuyen el mejor desempeño del láser verde a que este color suele asociarse a la señal de avanzar, mientras el rojo a la de detenerse (como en los semáforos) o a que en el ambiente de laboratorio este aparecía con más brillo que su competidor.

Si bien estos resultados son positivos, los experimentos contaron solamente con seis sujetos de prueba, por lo que resulta riesgoso generalizarlos.

2.3. Estímulo continuo utilizando Google Glass

Dentro de los dispositivos que proveen señalización visual, existe otra línea que en lugar de utilizar láser se basa en realidad aumentada para generar los estímulos. Allí se enmarca el trabajo de Zhao *et al.* [30] quienes desarrollaron una aplicación Android para realizar el estímulo utilizando los lentes de realidad aumentada *Google Glass*.

En esta aplicación se configuraron tres tipos de estímulo:

1. Auditivo: un metrónomo sin estímulo visual.
2. Rítmico: la pantalla se prende y se apaga a un ritmo previamente fijado.
3. Flujo óptico: Se generan dos líneas verticales a los lados de la pantalla que se aproximan al usuario a un ritmo previamente fijado.

Dentro de estos estímulos nos interesan los últimos dos, por su naturaleza de estímulos visuales. Se representan en la figura 2.2.

Realizaron diferentes tareas de caminata que incluían giros de 180° y 360° para testear la efectividad de este dispositivo en pacientes cuya última dosis de medicación ya no tenía efecto (fue tomada hace mucho tiempo). Los autores no registraron ninguna mejoría significativa con ninguno de los dos métodos de interés.

En una encuesta, Zhao *et al.* consultaron a los usuarios sobre su experiencia con los *Google Glass*, y si bien muchos reportaron que estos eran fáciles y cómodos de usar, pareciera que no son un buen método para feedback visual. La posición de la pantalla (arriba a la derecha) quita los estímulos del campo visual del usuario, requiriendo que este se concentre en mirar en una dirección particular (la pantalla) y generando una posible tarea múltiple.

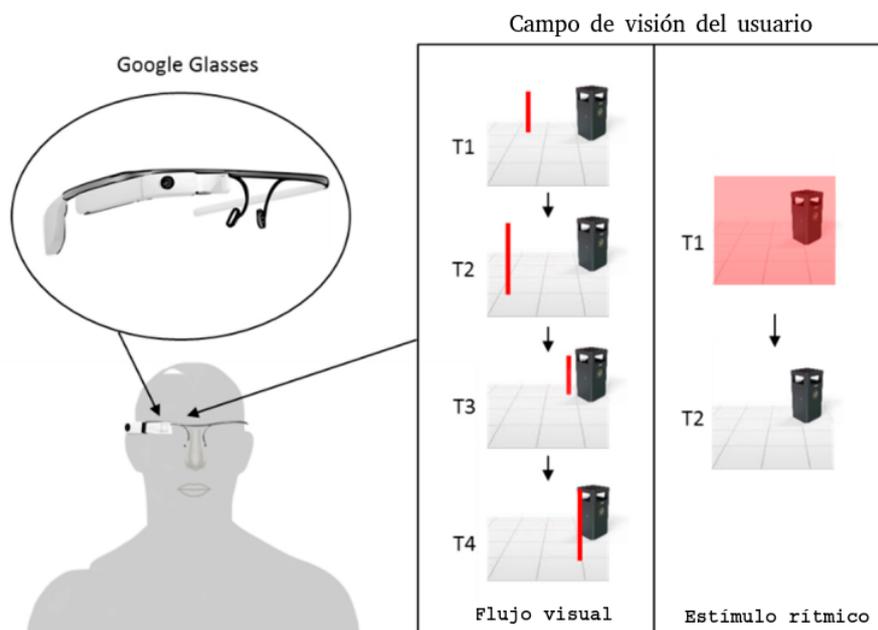


Figura 2.2: Estímulos en *Google Glass* por Zhao *et al.* Cada T representa un paso discreto en el tiempo, a la izquierda vemos las líneas dibujadas que se aproximan por la izquierda y luego la derecha del campo de visión del usuario (Flujo visual) y a la derecha el estímulo rítmico donde la pantalla se prende y apaga a un ritmo previamente fijado y en color rojo (imagen tomada de [28])

2.4. Estímulo no continuo utilizando Moverio Bt-200

Respondiendo a las sugerencias de trabajos anteriores sobre el posicionamiento de la pantalla y los problemas de la señalización continua (presencia de estímulos todo el tiempo), Ahn *et al.* [31] desarrollaron un algoritmo de detección de FoG específicamente para smart glasses y lo implementaron en los *Moverio BT-200* de Epson. La ventaja de estos lentes es que proyectan de manera binocular, es decir, directamente sobre el campo visual del usuario.

Cuando se detecta un episodio de FoG, se muestran líneas azules en el suelo, imitando las clásicas cintas usadas para estimulación visual. Para lograr este efecto los lentes (1) ajustan la distancia entre las líneas según el ángulo en el que el usuario esté mirando (Figura 2.3), (2) aproximan las líneas según el avance del usuario y (3) detectan cuando este pisa sobre una línea, haciéndola desaparecer.

Para la evaluación, se utilizan dos pruebas: (1) *TUG test*, donde el paciente debe pararse de una silla, caminar tres metros en línea recta, girar y volver a la silla y (2) una caminata de diez metros en línea recta.

Los autores no presentan resultados en función de episodios de FoG, pero si reportan una amplia mejoría en otros indicadores de la caminata. A continuación

2.4. Estímulo no continuo utilizando Moverio Bt-200

se presentan los resultados, promediando ambas pruebas:

1. Cantidad de pasos dados (se reducen un 22,6 %).
2. Velocidad promedio de los pacientes (aumentada en 37,2 %).
3. Longitud de zancada (aumentada en 31,7 %)

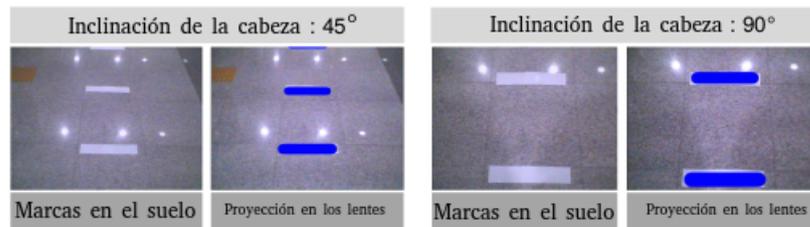


Figura 2.3: Cambio de la proyección en los lentes según ángulo de la cabeza por Ahn *et al.* alineado con marcas fijas en el suelo para demostrar la correctitud del ajuste. (Imagen tomada de [31], traducida)

Ahn *et al.* [31] proponen como trabajo a futuro agregar a los lentes la capacidad de detectar el suelo y/o obstáculos. Sugieren que la capacidad de cómputo de los lentes no es suficiente para esta tarea y como otro problema proponen la limitada capacidad de su batería, con la cual actualmente no se puede utilizar su sistema durante todo el día sin recargarla.

Capítulo 3

Marco teórico

Para el diseño de un dispositivo que genere estímulos capaces de guiar a un paciente dentro de entornos cambiantes y con presencia de obstáculos, este debe identificar los elementos a su alrededor, calcular trayectorias que los tomen en cuenta y generar estímulos que permitan seguirlos.

Estos requisitos presentan similitudes con algunos de los principales desafíos de la robótica autónoma. Si reemplazamos el robot con un usuario del dispositivo, se comparte la necesidad de:

- Crear un mapa del entorno de manera dinámica.
- Localizar al usuario dentro del mapa.
- Planificar una trayectoria a seguir hasta un destino marcado.
- Representar tridimensionalmente la pose del usuario.

Esto permite utilizar herramientas originalmente pensadas para desarrollos en robótica prácticamente sin necesidad de modificarlas.

En este capítulo se describen las herramientas utilizadas para el desarrollo del prototipo.

3.1. SLAM

Uno de los problemas a solucionar en el diseño de cualquier sistema de robótica autónoma que requiera desplazarse es el mapeo y localización simultáneos o SLAM (del inglés *simultaneous localization and mapping*). Este consiste en construir una representación de un espacio al mismo tiempo que se estima la posición del robot con respecto al punto donde este inicia su ejecución. Esta estimación es llamada **odometría**.

Capítulo 3. Marco teórico

Existen variadas formas de realizar SLAM, dependiendo de los sensores que se quiera utilizar, las características del entorno donde se despliegue y/o el tipo de robot.

3.1.1. Grillas de ocupación

Antes de poder localizarse en un entorno se necesita contar con una representación de este. Las representaciones más utilizadas son las grillas de ocupación. Estas consisten de una matriz bidimensional donde cada celda representa una porción de un plano y su valor indica la probabilidad de que ésta se encuentre ocupada. En la figura 3.1 se presenta un ejemplo de estas grillas. En negro se indican las celdas que están ocupadas, en gris claro las libres y en gris oscuro aquellas que no han sido mapeadas aún.

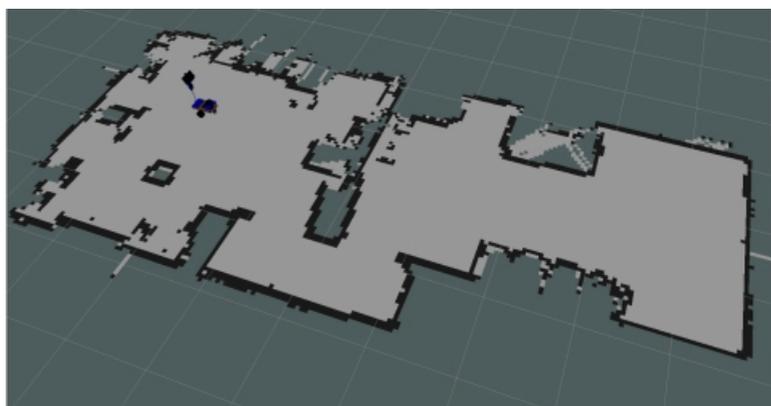


Figura 3.1: Ejemplo de una grilla de ocupación. En negro se indican las celdas que están ocupadas, en gris claro las libres y en gris oscuro aquellas que no han sido mapeadas aún. (Imagen tomada de [32])

3.1.2. SLAM visual

Dentro de los enfoques existentes para SLAM se encuentra el llamado SLAM visual (VSLAM). Este refiere a la utilización de cámaras como fuente de información para realizar el mapeo y la localización.

Existen dos subcategorías dentro del VSLAM que difieren en si se utiliza una única cámara (VSLAM monocular) o dos (VSLAM estéreo). La ventaja de la segunda, es que al obtener dos fotogramas simultáneos conociendo la distancia entre los sensores que los tomaron, se puede generar una imagen de profundidad (figura 3.2). Una imagen de profundidad contiene la distancia y el ángulo entre la cámara y cada uno de los objetos de la imagen. Con estas podemos construir una representación en tres dimensiones de lo que los sensores están observando.

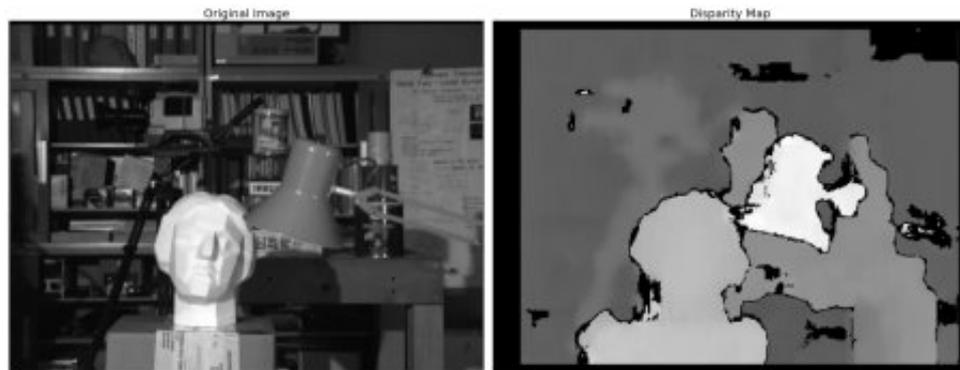


Figura 3.2: Ejemplo de imagen de profundidad. La intensidad de los píxeles representa la distancia a la que se encuentran del sensor, siendo más claros los más cercanos. (Imagen tomada de [33])

Para recrear la estructura de un espacio, es necesario mover los sensores y obtener imágenes de profundidad en varias direcciones, cubriendo la totalidad del mismo. Además, se debe comprender las conexiones espaciales entre estas imágenes para poder unir la información que nos brindan. Para lograr esto, los sistemas de VSLAM seleccionan un conjunto de puntos característicos en cada fotograma (el criterio de selección varía entre implementaciones) y buscándolos en el fotograma consecutivo deducen el movimiento de las cámaras en el espacio. De esta manera se puede obtener la odometría y reconstruir la escena a medida que esta se recorre. La figura 3.3 muestra un ejemplo de este proceso.



Figura 3.3: Ejemplo de odometría visual. Se observa la correspondencia de los puntos seleccionados entre dos fotogramas consecutivos y la reconstrucción de la escena, con la posición de las cámaras cuando tomaron cada fotograma. (Imagen tomada de [34])

Las estimaciones de la odometría pueden no ser precisas a largo plazo por acumulación de errores, generando problemas cuando se revisita un lugar. Es necesario entonces que los sistemas de SLAM cuenten con mecanismos para reconocer cuando la escena actual ya fue mapeada y poder de esa manera corregir las estimaciones y evitar la sobrescritura de información previamente obtenida. Este proceso es llamado **cierre de ciclos**.

3.1.3. RTAB-Map

RTAB-Map [35] (*Real-Time Appearance-Based Mapping*) es una implementación de VSLAM que utiliza grafos para almacenar los fotogramas representativos de cada zona y se basa en la apariencia para el cierre de ciclos.

Cuando se mapea un entorno amplio, la cantidad de puntos a almacenar escala rápidamente y con ellos la capacidad de cómputo necesaria para realizar el proceso de cierre de ciclos (ya que se debe contrastar el fotograma actual con los anteriores). RTAB-Map implementa un sistema de manejo de memoria a corto y largo plazo para limitar la cantidad de puntos a contrastar, logrando de esa manera funcionar en tiempo real incluso en entornos grandes. Es por esto que es una de las implementaciones de VSLAM más utilizadas en la robótica autónoma.

En la figura 3.4 se presenta el diagrama de funcionamiento de RTAB-Map.

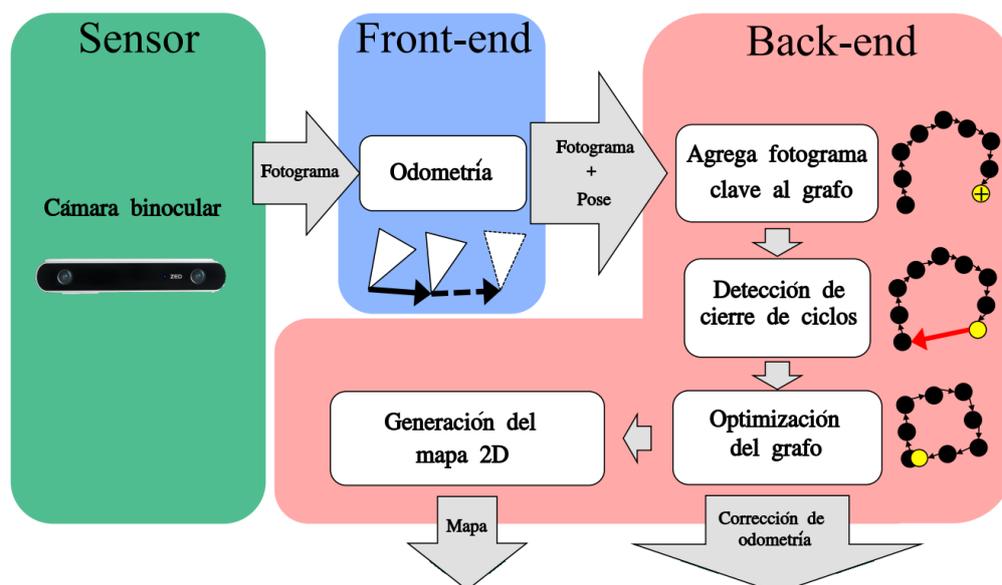


Figura 3.4: Diagrama de funcionamiento de RTAB-Map. (Imagen tomada de [34], traducida)

Sensor

Se necesita una cámara estéreo que se encuentre calibrada para minimizar la deformación causada por sus lentes en las imágenes obtenidas y envíe los fotogramas a la siguiente etapa.

Front-end

Este módulo recibe como entrada los fotogramas tomados y es quien se encarga de computar la imagen de profundidad y la odometría, siguiendo los procedimientos

descritos en la parte anterior. A la salida de este módulo tenemos la pose de la cámara en el espacio y los fotogramas recibidos.

Back-end

Una vez obtenida la información de la pose, es este módulo quien se encarga de la construcción del mapa en su totalidad. Para esto se extraen las características más representativas del frame utilizando *Speeded-Up Robust Features* [36] y se agregan al grafo de lugares.

El siguiente paso es comparar las características obtenidas con las de otros frames candidatos a representar el mismo lugar (almacenados en la memoria de trabajo de RTAB-Map). En caso de encontrar una coincidencia, se realiza el cierre de ciclo correspondiente y se optimiza el grafo con las correcciones que este implique.

Para obtener la representación tridimensional se utiliza la información de profundidad. Se calcula con ésta la ubicación en el espacio de cada pixel, generando una nube de puntos en tres dimensiones. En la figura 3.5 (a) se presenta una nube de puntos a modo de ejemplo. Usualmente los sistemas no requieren de una reconstrucción en tres dimensiones de su entorno para resolver las tareas de navegación, es por esto que RTAB-Map también proyecta esta nube en dos dimensiones y genera una grilla de ocupación (figura 3.5 (b)).

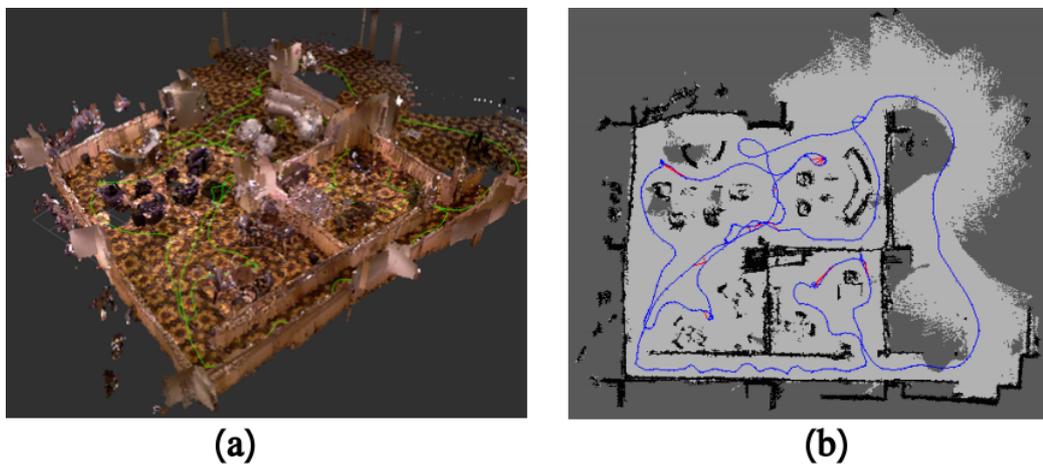


Figura 3.5: (a) muestra una nube de puntos creada por RTAB-Map al recorrer la trayectoria presentada en verde y (b) la proyección de la nube en un plano, en forma de grilla de ocupación. En azul se marca la trayectoria seguida y en rojo los cierres de ciclos detectados. (Imágenes de [34], unidas)

Este módulo también se encarga de corregir la odometría calculada en el módulo de front-end para reflejar los ajustes realizados al mapa.

3.2. Robot Operating System

Robot Operating System (ROS) [37] es un meta-sistema operativo diseñado para sistemas robóticos. Este provee servicios tales como la abstracción del hardware, el control de dispositivos de bajo nivel, la implementación de funcionalidad de uso común, el paso de mensajes entre procesos y el mantenimiento de paquetes, siendo estos últimos implementaciones *open source* de algoritmos y funcionalidades comunes en robótica. Está basado en una arquitectura de grafos donde el procesamiento toma lugar en los nodos que pueden recibir, mandar y multiplexar mensajes de sensores, control, estados, planificaciones y actuadores, entre otros. Estos nodos se comunican a través de tópicos, donde los mensajes son almacenados para su lectura en un modelo de cola.

3.2.1. Stack de navegación de ROS

Para el cálculo de trayectorias se utiliza el paquete de ROS *Navigation* [38], el cual toma las lecturas de los distintos sensores del sistema, el mapa del entorno, la posición del robot e información sobre las características físicas de este (como pueden ser sus dimensiones) y se encarga de planificar trayectorias seguras para el mismo. Esto implica encontrar caminos lo suficientemente espaciosos para el robot y tomar en cuenta la aparición de obstáculos nuevos (no presentes en el mapa) para recalcularlos.

El diagrama completo de componentes del paquete se presenta en la figura 3.6:

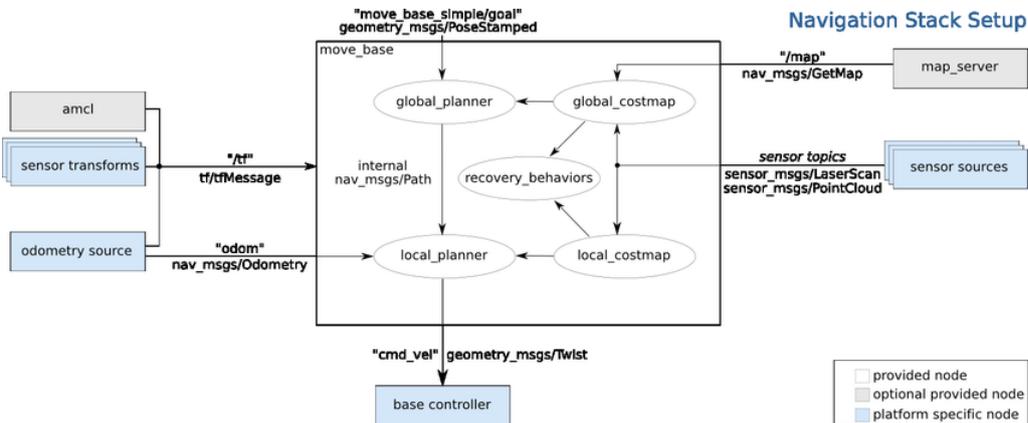


Figura 3.6: Diagrama del stack de navegación ROS. (Imagen tomada de [39])

Los nodos provistos por el paquete son:

- **global_costmap**: toma como entrada el mapa del entorno y un archivo de configuración. Se encarga de transformar este mapa de ocupación en uno de costos, donde además de marcar los lugares donde existen obstáculos, podemos expandirlos y marcar ciertas zonas como más costosas de navegar.

3.2. Robot Operating System

- **local_costmap:** toma como entrada el mapa de costos global y junto con las lecturas de los sensores agrega obstáculos y genera un mapa local, solo existente en el entorno más cercano al robot.
- **global_planner:** este nodo, una vez que recibe el destino, es quien planifica la trayectoria hasta allí, utilizando la posición actual del robot y la información del mapa de costos global.
- **local_planner:** tomando la posición actual del robot y la trayectoria global a seguir, este nodo planifica una trayectoria a corto plazo, capaz de lograr que se siga la global de la mejor manera. Toma para esto la información del mapa de costos local.
- **recovery_behaviours:** este nodo es quien realiza distintas maniobras cuando se detecta que el robot es incapaz de seguir la trayectoria planificada hacia el destino, estos comportamientos de recuperación están diseñados para actualizar los mapas de costos en búsqueda de nuevos espacios por los que navegar.

3.2.2. Transform

El paquete *Transform* [40] de ROS es utilizado para mantener información sobre los marcos de referencia durante el tiempo. Es especialmente útil en el área de la robótica para representar la posición y orientación de varias partes conectadas entre sí. Estos marcos de referencia se relacionan mediante una estructura arborescente, conectándose a través de transformaciones espaciales (traslaciones y rotaciones).

En la figura 3.7 se muestra un ejemplo donde se utilizan los marcos de referencia para conocer el estado de las distintas partes del robot.

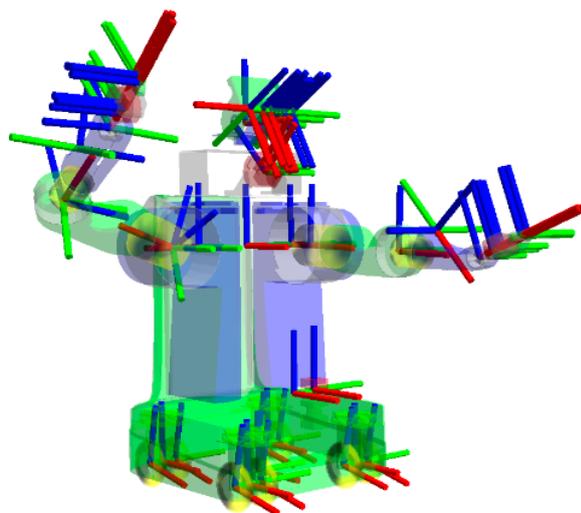


Figura 3.7: Ejemplo de uso de transform. (Imagen tomada de [40])

Capítulo 3. Marco teórico

Una de las mayores ventajas en la utilización de este paquete es la posibilidad de transformar puntos y vectores entre distintos marcos de referencia de manera sencilla. Con este se puede obtener rápidamente información del estilo “¿Dónde se encuentra la base del robot en el mapa?” o “¿Cuál es el ángulo entre la mano del robot y un objeto?”.

Capítulo 4

Solución propuesta

Para el diseño de la solución se parte de los requerimientos recabados por el proyecto CSIC “Diseño de un dispositivo para fomentar la continuidad de la marcha en pacientes con Parkinson”. En este se plantea un diseño basado en la usabilidad, donde la simplicidad de la interfaz es central.

Con esto en mente, se construye un dispositivo controlado por comandos de voz que es capaz de automáticamente generar mapas de su entorno, ubicarse en el mismo y planificar trayectorias seguras hacia el lugar que el usuario indique. Utilizando lentes de realidad aumentada se dibujan líneas en el suelo perpendiculares a la trayectoria calculada, de modo que el usuario solo deba pisarlas para desplazarse.

En búsqueda de validar la solución propuesta se realizaron consultas con doctores del hospital de clínicas que tratan pacientes con Parkinson.

4.1. Diseño de los estímulos

A la hora de diseñar un sistema con la capacidad de guiar la caminata de un usuario se toman en cuenta las experiencias exitosas obtenidas por los estímulos visuales que proyectan líneas en el suelo, donde los usuarios deben pisarlas para desplazarse. Estos dispositivos muestran efectividad en el alivio del congelamiento en la marcha, aunque los episodios siguen sucediendo cuando el usuario debe hacer un giro o encuentra un obstáculo que debe sortear.

Una posible explicación a la complicación en los cambios de dirección, es que los dispositivos de señalización como el bastón láser o los *Laser Shoes* [15] (figura 4.1) proyectan las líneas frente al usuario siempre de manera paralela a sus hombros. Con estos estímulos, un cambio de dirección involucra ajustarlos (mediante una acción voluntaria de re-posicionamiento) o ignorarlos. Esta situación puede generar estrés y/o confusión en el usuario, despertando un episodio de congelamiento.

Capítulo 4. Solución propuesta

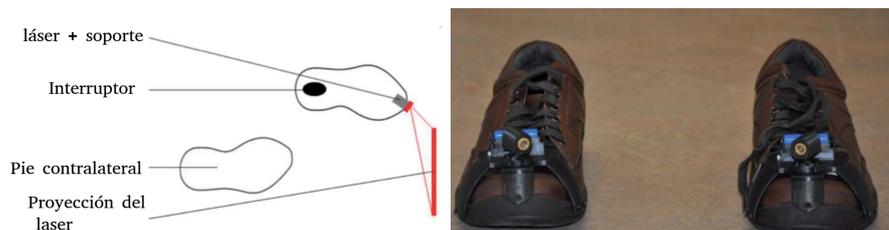


Figura 4.1: Los Laser Shoes utilizan un interruptor en el talón del usuario que cuando es presionado proyecta una línea delante del otro pie, generando un estímulo visual donde los usuarios deben pisar para continuar la marcha. (Imágenes tomadas de [15], unidas y traducidas)

Considerando esta problemática se construye una solución capaz de mostrar líneas que respondan a la trayectoria real que debe realizar el usuario. En la figura 4.2 se compara los estímulos generados por sistemas como el bastón láser o los *Laser Shoes* con los propuestos en este trabajo.

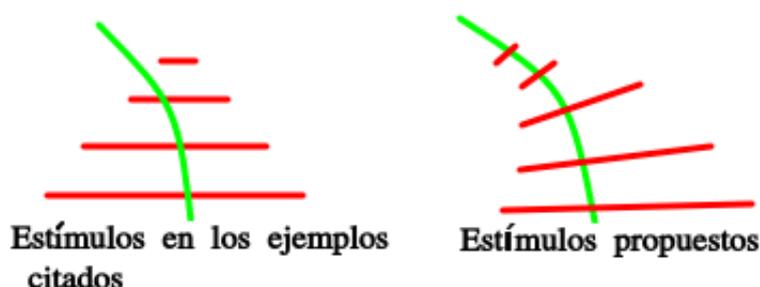


Figura 4.2: Comparación entre los estímulos brindados por los dispositivos actuales y los propuestos en el trabajo. La línea verde representa la trayectoria que el usuario debería seguir y las rojas los estímulos brindados por los dispositivos.

4.2. Mapeo del entorno

Los estímulos descritos en la parte anterior responden a una trayectoria calculada entre la posición del usuario y su destino, que tome en cuenta los obstáculos que pueda haber entre esos dos puntos. Para obtener esa información es necesario contar con una representación del entorno, la ubicación del usuario en la misma y el punto destino. Como se mencionó anteriormente este es uno de los desafíos fundamentales en la robótica autónoma, donde para que un robot pueda navegar es necesario crear un mapa del espacio donde se encuentra y su posición.

Uno de los factores decisivos al elegir qué tipo de SLAM utilizar es el tipo de robot con el que se cuenta. En nuestro caso, se busca sustituir al mismo por un ser humano, siendo la única alternativa comparable los robots humanoides (aquellos construidos con características físicas similares a los humanos).

4.2. Mapeo del entorno

Los robots tradicionales (con ruedas) tienen menos grados de libertad que los humanoides, dado que estos no cambian la altura a la cual se encuentran sus sensores ni los ángulos entre estos y el suelo, facilitando el cálculo de las distancias en el entorno cuando se utilizan sensores del estilo sonar. Esta diferencia junto con la necesidad de minimizar la cantidad de hardware montado sobre los humanoides (y en el caso de este proyecto los humanos) motiva el uso de SLAM visual [41], donde solo se necesitan un par de cámaras y es posible calcular los movimientos directamente de la entrada sensorial.

4.2.1. Cámara binocular ZED

El sensor utilizado para realizar VSLAM es el ZED (figura 4.3) debido a su bajo costo, que este cuenta con un paquete ROS para su utilización [42] y la disponibilidad de este en el laboratorio de robótica de facultad.



Figura 4.3: Sensor de cámara binocular ZED. (Imagen tomada de [43])

Es este paquete quien se encarga de (I) obtener las imágenes del sensor, rectificarlas utilizando los parámetros de deformación característicos de sus lentes y publicarlas en sus respectivos tópicos y (II) publicar la transformación espacial entre los centros de las cámaras, necesaria para obtener la información de profundidad a partir de las imágenes obtenidas.

4.2.2. RTAB-Map

Una vez que se cuenta con las imágenes rectificadas y la transformación espacial entre ellas, resta reconstruir la escena capturada por las mismas tridimensionalmente, con el objetivo de crear un mapa del entorno. El paquete ROS RTAB-Map es el encargado de realizar esta tarea.

Este consiste de dos nodos:

- ***RTABMap odometry*** calcula la pose de los sensores en función de los desplazamientos encontrados en los píxeles de imágenes consecutivas.
- ***RTABMap map*** toma como entrada las imágenes rectificadas, la transformación entre estas y la odometría calculada por el nodo RTABMap odometry para construir una nube de puntos tridimensional. Luego proyecta la misma sobre un plano y publica un mapa del entorno en forma de grilla de ocupación.

Capítulo 4. Solución propuesta

Para el correcto funcionamiento de RTAB-Map, este debe construir una imagen de disparidad a partir de lo captado por las cámaras. Dependiendo de las situaciones a las que el sistema se enfrente, varios parámetros deben ser configurados. Para nuestro caso de uso (hogares), se ajustaron los siguientes:

- **Max disparity:** para encontrar las coincidencias de píxeles entre las dos imágenes, se barre la derecha buscando los píxeles observados en la izquierda. Este parámetro indica qué tanto recorrer en búsqueda de coincidencias. Cuanto más grande, el sistema es capaz de detectar mejor objetos grandes o cercanos a la cámara, pero el tiempo de cómputo también crece.
- **Speckle Size:** si un conjunto de píxeles conexos y con un mismo valor de distancia (*Speckle*) tiene un tamaño menor a este parámetro, es considerado ruido y se descarta.
- **Speckle Range:** diferencia de valor entre conjuntos de píxeles conexos para ser considerados parte de un mismo grupo (*Speckle*).

En la figura 4.4 se presenta un diagrama de lo descrito en esta sección. A la salida de este módulo se encuentra el mapa construido del entorno y la ubicación del usuario en el mismo.

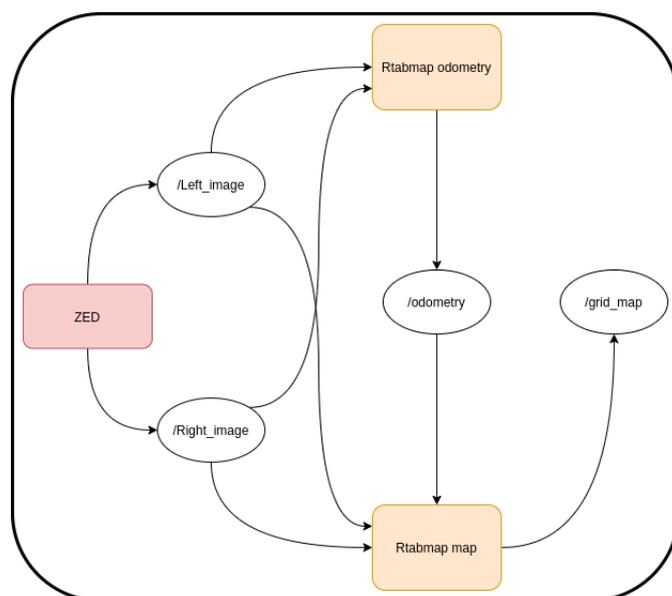


Figura 4.4: Diagrama de componentes VSLAM

4.3. Planificación de trayectorias

Una vez que se cuenta con un mapa y la ubicación de usuario en el mismo, se tiene toda la información necesaria para planificar las trayectorias a seguir para

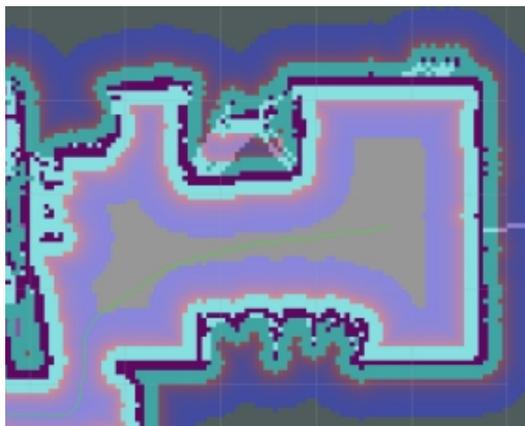


Figura 4.6: Ejemplo de mapa de costos. Las celdas ocupadas tienen costo infinito (representado en violeta) y a su alrededor existen tres niveles de costo (celeste, rojo y azul) descendientes en valor a medida que se alejan del obstáculo. (Imagen tomada de [32])

Para calcular una trayectoria entre dos puntos en el mapa de costos, se aplica el algoritmo de Dijkstra, buscando el camino entre ellos que minimice el costo acumulado.

4.4. Generación de estímulos

4.4.1. Construcción

El nodo *Navigation* publica la trayectoria calculada en forma de línea. Para transformar esto en los estímulos buscados en este trabajo, se implementa un nodo llamado *Line generator* que toma la trayectoria, genera líneas perpendiculares a esta en intervalos regulares (coincidentes con el largo de paso buscado) y las publica. La figura 4.7 presenta un ejemplo donde se muestran las líneas generadas sobre una trayectoria.

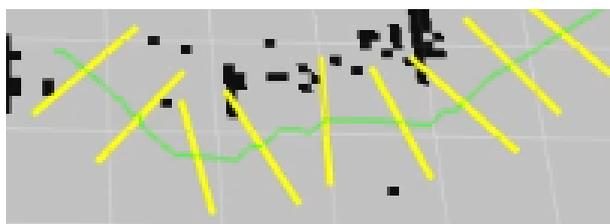


Figura 4.7: Ejemplo de estímulos sobre una trayectoria.

4.4.2. Renderizado

Una vez obtenidas las líneas que guiarán al usuario, es necesario mostrarlas. Esto puede realizarse con un sistema de proyección, que al integrarse con la repre-

4.4. Generación de estímulos

sentación virtual del entorno ubique y muestre las líneas en el suelo. Esto implica que para utilizar el dispositivo de asistencia en un lugar, este deberá ser previamente equipado y configurado. En búsqueda de mayor flexibilidad, se opta por utilizar lentes de realidad aumentada. Con estos es posible generar la ilusión de que las líneas están en el suelo sin la necesidad de montar hardware en el entorno.

A la hora de elegir los lentes a utilizar en este proyecto se investigaron las principales opciones disponibles en el mercado que cumplieran con los estándares mínimos en relación a las recomendaciones descritas en el trabajo de Zhao *et al.* [45], quienes realizaron una encuesta a nivel nacional en Países Bajos para investigar la opinión de los pacientes con Parkinson sobre la opción de utilizar lentes de realidad aumentada en su vida diaria. En el documento de estado del arte [27] se presenta el relevamiento completo de las opciones, que resultada en la elección de los Moverio BT-300 como mejores candidatos.

Otra ventaja en la utilización de lentes de realidad aumentada, es que estos cuentan con cámaras integradas, haciendo posible el despliegue de el sistema de VSLAM con la información proveniente de estos. En este trabajo se utiliza los lentes Moverio BT-300, que cuentan con una única cámara, por lo que de todas formas se necesita utilizar el sensor ZED. El prototipo desarrollado en el marco de este proyecto de grado coloca al sensor ZED montado sobre los BT-300 (figura 4.8), generando que el sistema sea aplicable directamente a lentes que si cuentan con más de una cámara.



Figura 4.8: Montaje del sensor ZED sobre los Moverio BT-300.

Contar con las cámaras montadas en la cabeza del usuario implica que el nodo de localización reporte la posición y postura de esta en el mapa, lo que es de gran utilidad, dado que para mostrar las líneas en los lentes y generar el efecto de que se encuentran en el suelo, es necesario saber hacia dónde mira el usuario.

Capítulo 4. Solución propuesta

A partir de la odometría publicada por el nodo de RTAB-Map, se realizan dos transformaciones:

(I) Para que la planificación de trayectorias parta desde la posición de los pies, se crea el nodo **ZED projector** que proyecta la posición reportada por RTAB-Map en el plano del suelo (figura 4.9 A) y se configura esta como el punto de partida para el planificador.

(II) Para contar con un marco de referencia centrado en los ojos del usuario, se utiliza el nodo **Static Transform Publisher** y se publica una transformación espacial que traslada el marco de referencia de las cámaras hacia abajo y atrás. Este marco es llamado **head_position** (figura 4.9 B).

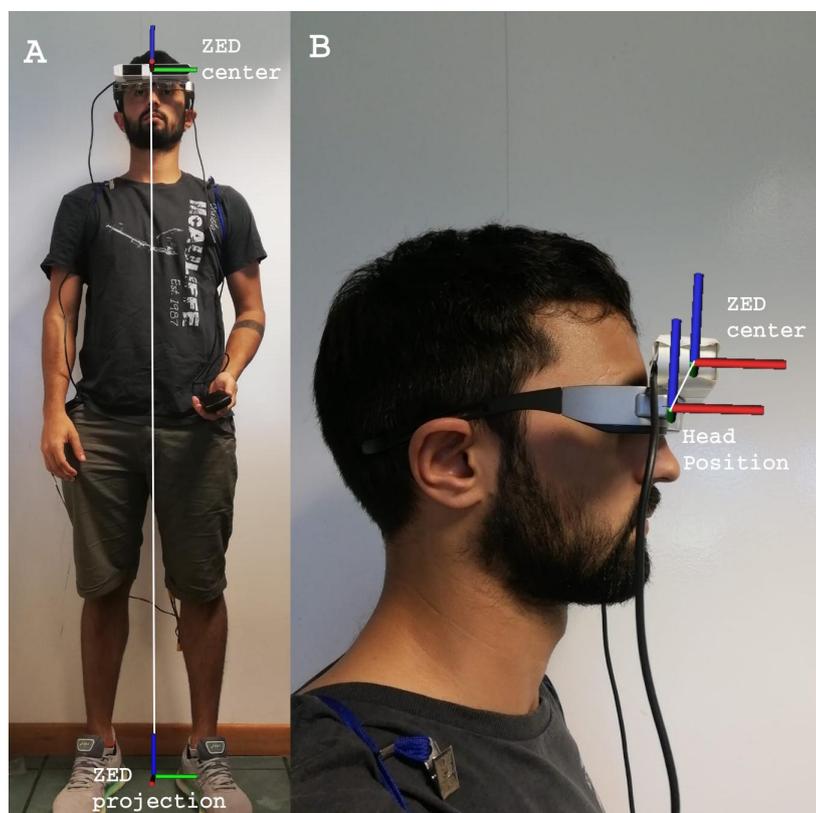


Figura 4.9: Diagrama de transformaciones y marcos de referencia.

Con la representación tridimensional de la situación actual, el siguiente paso es calcular la proyección de las líneas con respecto a los ojos del usuario, para luego renderizar esta información en los lentes. Para esto se implementa el nodo **Line transformer** que se encarga de:

- Obtener las coordenadas de los extremos de las líneas creadas en el marco de referencia *head_position*, utilizando el paquete **Transform**.
- Simular una cámara centrada en *head_position* y proyectar los puntos obtenidos en el paso anterior en el plano imagen de esta (figura 4.10), utilizando

el paquete *Image Geometry* [46].

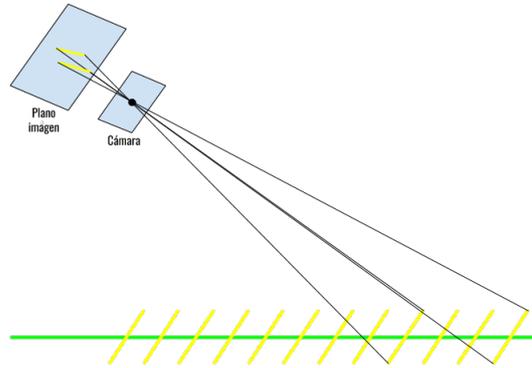


Figura 4.10: Proceso de proyección de las líneas en una cámara.

Al utilizar una resolución del plano imagen igual a la de la pantalla de los lentes y parámetros de deformación de la cámara que simulan el comportamiento del ojo humano, simplemente se debe mostrar lo obtenido en el plano imagen (invertido) en la pantalla de los lentes.

En la figura 4.11 se presenta el diagrama de componentes del módulo descrito. Este parte de la odometría generada por el módulo de VSLAM y genera, a la salida del nodo *Line Transformer*, los puntos a renderizar en los lentes de realidad aumentada.

La salida de este módulo (extremos de los puntos de las líneas) es enviada a los lentes para ser mostrada. En una primer instancia se buscó incorporar los lentes al entorno ROS, pero debido a la limitada capacidad de cómputo y la versión de Android que estos tienen se descartó esta posibilidad. La información entonces es enviada a través de una red wifi local, en un paquete UDP, que será procesado por una aplicación desarrollada de forma nativa para Android.

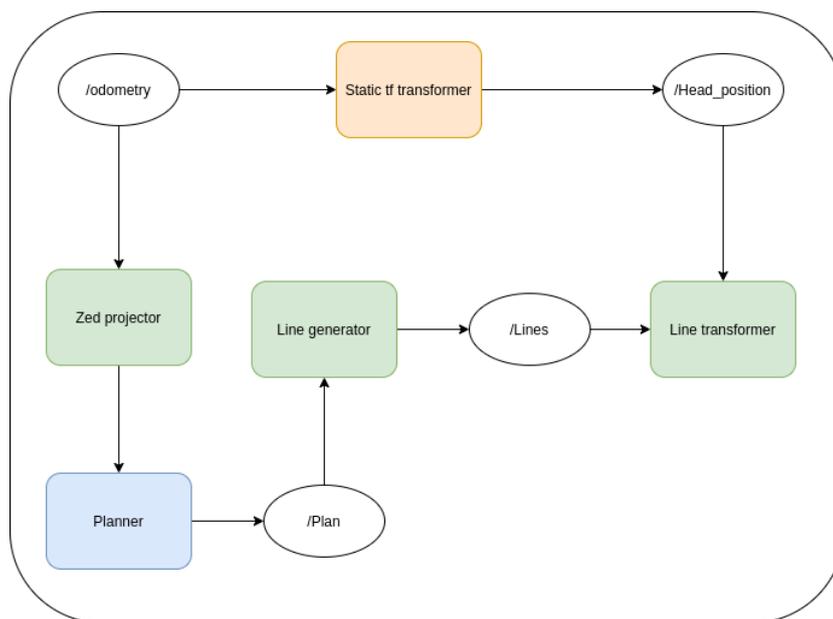


Figura 4.11: Diagrama de componentes del módulo de generación de estímulos.

4.5. Interfaz

Para conectar las funcionalidades descritas anteriormente con el usuario, se diseña una aplicación nativa en Android, que ejecuta en los lentes de realidad aumentada. Todos los procesos anteriormente descritos son transparentes al usuario, quien interactúa directa y únicamente con los lentes.

A continuación se describen las tres funcionalidades principales implementadas.

4.5.1. Proyección de las líneas

Una vez que el sistema de señalización obtiene las proyecciones de los extremos de las líneas a mostrar, estos son enviados a los lentes utilizando el protocolo UDP. Esta decisión se toma en búsqueda de obtener la menor latencia posible y con la consideración de que la pérdida de paquetes no es un problema en el pasaje de esta información, ya que se prioriza mostrar la proyección más reciente sobre considerar todas las calculadas.

La aplicación cuenta con un hilo de ejecución que recibe estos paquetes y dibuja en pantalla las líneas que estos puntos representan. Estas se muestran en color verde, teniendo en cuenta lo reportado por Bryant *et al.* [29]. El resto de la pantalla se muestra negra dado que esto se aprecia como transparente en los lentes.

4.5.2. Guardado de puntos

Mientras el usuario recorre el entorno donde desea utilizar el sistema, este puede guardar ciertos puntos de su interés, para contar con la posibilidad de navegar hacia estos en un futuro.

Para guardar un lugar, el usuario debe dirigirse hacia este una vez que se encuentra allí pulsar el botón derecho del control asociado a los Moverio BT-300 (figura 4.13). En ese momento, la aplicación presenta una ventana indicando al usuario que debe elegir un nombre para ese punto.

El ingreso del nombre se realiza con la API de reconocimiento de voz de Google [47]. Se despliega una ventana y el usuario dice en voz alta el nombre que desea utilizar para guardar el punto donde se encuentra parado.

La aplicación envía la orden de guardar el punto, junto con el nombre elegido al sistema de localización utilizando el protocolo TCP. Esta decisión se toma considerando que este paso requiere de una comunicación sin pérdida de información y no es relevante la latencia que pueda tener. Para las comunicaciones TCP, se implementa un nodo ROS llamado *Places Handler*, que recibe esta orden, obtiene la posición actual del usuario en el mapa utilizando el paquete *Transform* y la almacena en un archivo permanente en memoria. Al terminar, se envía una señal de éxito desde el nodo a la aplicación y se muestra en pantalla un cartel que lo indica.

En la figura 4.12 se presenta un diagrama de secuencia de esta acción.

4.5.3. Elección de punto destino

En cualquier momento durante el uso del dispositivo, el usuario puede indicar que desea dirigirse a alguno de los puntos que guardó previamente.

Al pulsar el botón izquierdo del control asociado a los Moverio BT-300 (figura 4.13), la aplicación envía utilizando TCP una petición de los sitios que han sido guardados. Es el nodo *Places Handler* quien atiende la misma y responde con una lista de nombres obtenida desde el archivo de lugares en memoria. Esta lista es desplegada en una ventana al usuario, indicando que a continuación podrá nombrar su destino.

Una vez más se utiliza la API de Google para reconocer lo dicho por el usuario. Luego, se busca dentro de los nombres, el de mayor similitud con lo reconocido por la misma (utilizando la distancia de *Levenshtein*) para compensar por diferencias durante la detección. Es ese nombre el que se envía al nodo *Places Handler*, que se encarga de obtener las coordenadas asociadas y enviarlas al nodo de planificación de trayectorias, indicando el destino e iniciando la navegación.

Capítulo 4. Solución propuesta

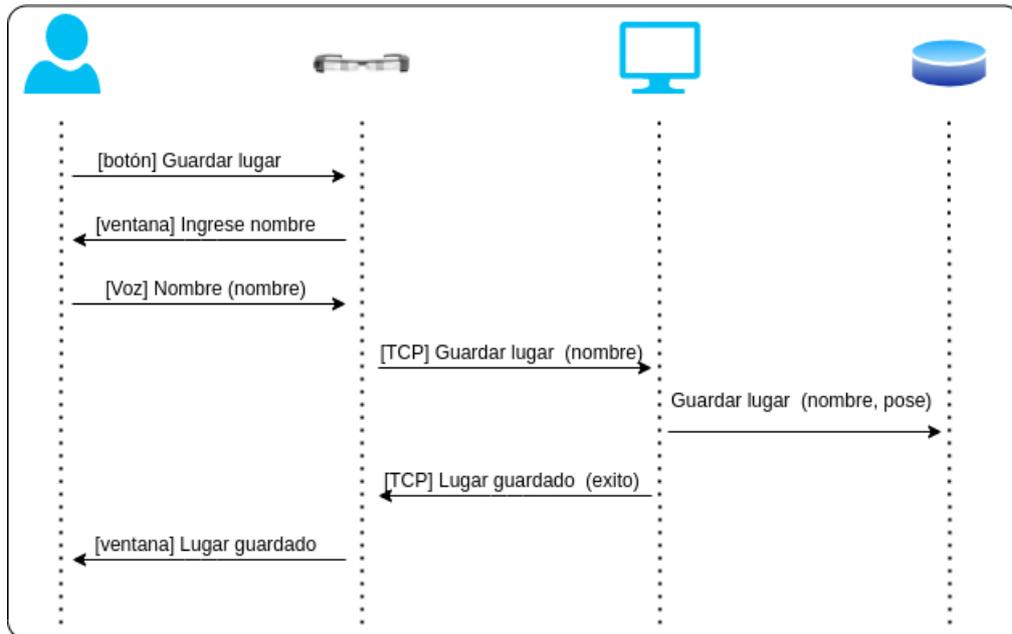


Figura 4.12: Diagrama de secuencia para el guardado de un punto. Las comunicaciones se presentan en el formato **[Interfaz] acción (información enviada)**

En la figura 4.14 se presenta un diagrama de lo descrito en esta sección, separando el sistema desplegado en el entorno ROS, la aplicación Android y lo almacenado en memoria.



Figura 4.13: Control de los lentes Moverio BT-300. (Imagen tomada de [48])

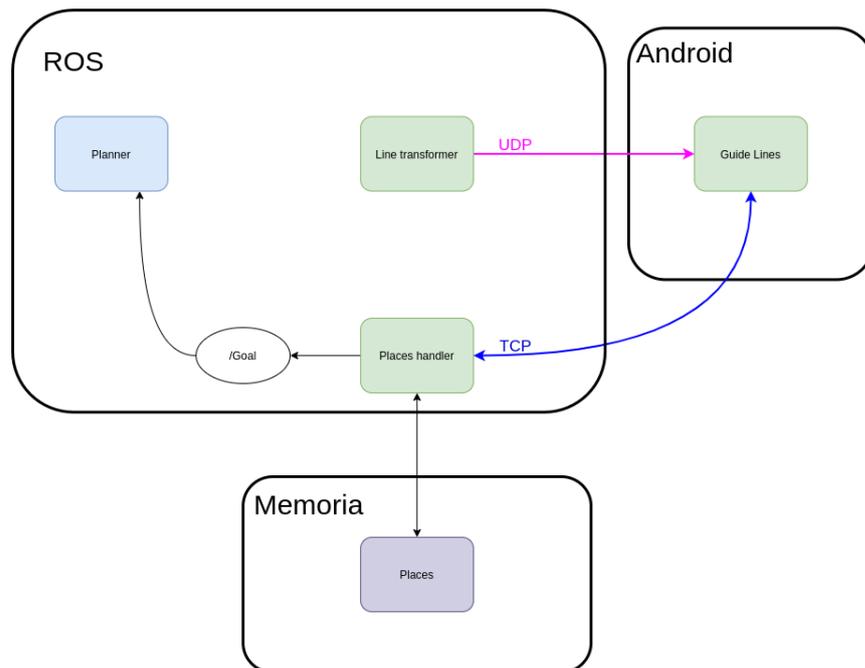


Figura 4.14: Diagrama de comunicación de la interfaz

4.6. Despliegue

En la figura 4.15 se presenta el esquema del sistema, separado en tres módulos:

- **ROS:** allí se listan los nodos (rectángulos) y tópicos (óvalos) involucrados en la solución. En verde los implementados completamente para este desarrollo, en rojo el nodo de comunicación con el sensor ZED, en amarillo los perte-

4.6. Despliegue



Figura 4.16: Componentes de hardware utilizados durante el desarrollo

Capítulo 5

Experimentación

Una vez desplegado el sistema en su totalidad, se realizan pruebas sobre el mismo para validar su correcto funcionamiento. Se separa la experimentación en dos partes donde primero se pone a prueba el sistema de mapeo y localización, probando la robustez del mismo y luego se realiza una prueba de integración entre todas las partes mediante la ejecución de un caso de uso típico.

5.1. Prueba de mapeo y localización

La primer prueba realizada se enfoca en validar el correcto funcionamiento del módulo de mapeo y localización RTAB-Map una vez realizada la configuración de parámetros. Teniendo en cuenta que el uso esperado del dispositivo es en entornos domésticos (casas de los usuarios) y buscando probarlo sobre uno más desafiante, las pruebas se realizaron en el piso 0 del Instituto de Computación de Facultad de Ingeniería.

El entorno (mostrado en la figura 5.1) tiene **74m** de largo por **10m** de ancho y cuenta con tres halls (izquierdo, central y derecho). Se considera que el tamaño promedio de un hogar es significativamente menor, siendo esta prueba exigente en términos de dimensiones. Como se utiliza VSLAM, es importante considerar las diferencias visuales entre distintos lugares del entorno, ya que serán utilizadas para ubicar el usuario dentro del mapa. Probar el dispositivo en un edificio de oficinas es exigente también desde esta perspectiva, ya que suelen ser más monótonos y repetitivos que los hogares. A pesar de estas diferencias entre la zona elegida y los hogares promedio, estos comparten una estructura general de habitaciones continuas y pasillos que las conectan. También es similar la iluminación, teniendo ventanas donde entra luz directa del sol, pero también zonas donde solo existe iluminación artificial.

Capítulo 5. Experimentación

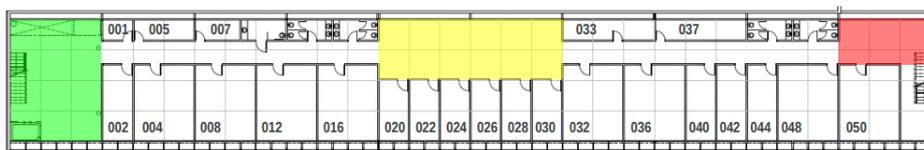


Figura 5.1: Plano del Instituto de Computación. Se resaltan en verde, amarillo y rojo los halls izquierdo, central y derecho respectivamente.

5.1.1. Mapeo

La primer parte de la prueba consiste en la construcción de un mapa del entorno. Para esto se recorre el edificio de una punta a la otra, ida y vuelta, una única vez. No se ingresa a las salas mostradas en el plano (por motivos de disponibilidad), por lo que se espera una representación que solo tome en cuenta el pasillo. Con esta única instancia el resultado construido por el sistema es presentado en la figura 5.2.

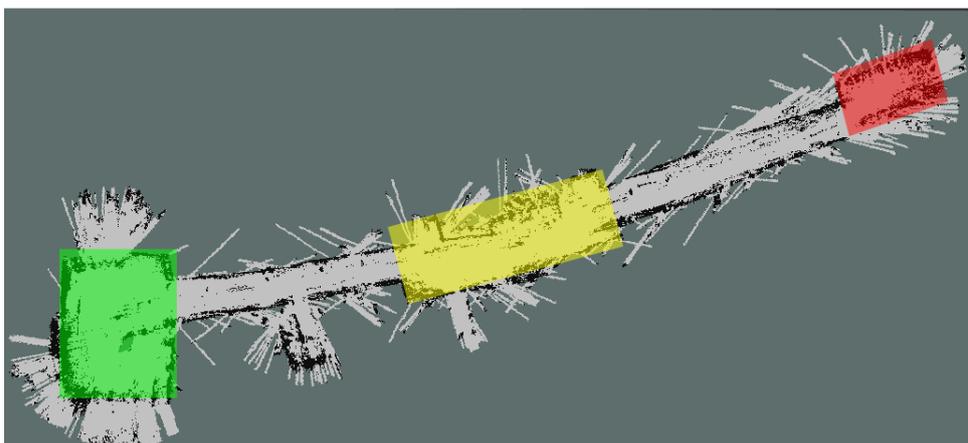


Figura 5.2: Grilla de ocupación construida del Instituto de Computación. Se resaltan en verde, amarillo y rojo los halls izquierdo, central y derecho respectivamente.

La grilla de ocupación construida representa de manera fiel la realidad. En esta se pueden distinguir los tres grandes espacios libres presentados en el plano (halls) y los pasillos que los unen. Es importante notar que actualmente la parte central contiene mobiliario, disminuyendo el espacio libre en el mapa. En algunas de las paredes se observan haces de espacio libre, estos responden a puntos aislados que fueron ubicados donde el haz termina. Dependiendo el caso, esto sucede por una de dos razones: (I) las puertas que delimitan las salas son de vidrio (o pueden incluso estar abiertas), por lo que el sensor puede encontrar puntos que se encuentren dentro de las mismas, o (II) puntos cuya distancia fue mal calculada por RTAB-Map.

El mapeo inicial se realizó por la mañana, lo que implica una iluminación mayormente de luz natural entrando por las ventanas. En las siguientes pruebas

5.1. Prueba de mapeo y localización

se somete al sistema a cambios en la iluminación y se estudia la robustez ante los mismos.

5.1.2. Localización

Una vez que se cuenta con el mapa del entorno, se prueba la capacidad del sistema para localizar al usuario en el mismo. Esto se hace iniciando el sistema en el extremo izquierdo del edificio, caminando hasta el derecho y volviendo. Primero se realiza solamente reportando la posición detectada sobre el mapa estático y luego permitiendo al sistema actualizar el mismo. Se realizan dos experimentos, uno en la mañana y otro en la noche, para comprobar la robustez ante los cambios de tipo (artificial o natural) y cantidad de luz en el entorno.

Para ubicarse en el mapa, RTAB-Map toma puntos característicos encontrados en cada frame y busca coincidencias con su base de datos. Los puntos que se toman como coincidencias son llamados *Inliers* y para validar una hipótesis de ubicación se requiere una cantidad mayor a un umbral predefinido (*minInliers*), cuyo valor por defecto es 20. En caso que no se cuente con la cantidad necesaria, se utiliza odometría visual para calcular el desplazamiento desde la última hipótesis aceptada. Durante cada prueba se registra la cantidad de *Inliers* encontrados como medida de confianza en la posición reportada.

Resultados en el día utilizando únicamente localización (dia-loc)

Se presenta en la figura 5.3 la cantidad de *Inliers* reportados por RTAB-Map durante la recorrida y una línea constante que marca el mínimo necesario para validar las hipótesis de localización. Las líneas rojas punteadas representan los pasajes entre cada hall (identificados por la barra en la parte superior) y el siguiente pasillo. El máximo de características encontradas es de **324**, el promedio **72.9** y la desviación estándar **73.25**. El sistema reporta más de *minInliers* en un **76 %** de los casos.

En la figura 5.4 se presenta la posición del usuario (publicada por RTAB-Map) en el plano para cada uno de los reportes. La recorrida es desde el hall izquierdo al derecho ida y vuelta, por claridad se presentan ida y vuelta de forma independiente. Los puntos en verde indican una cantidad de *Inliers* por encima de la media, en amarillo entre la media y el mínimo necesario y en rojo por debajo del mínimo.

En esta prueba las condiciones lumínicas son similares a las presentes durante la creación del mapa: luz natural en gran medida. El sistema fue capaz de encontrar coincidencias entre sus observaciones y el mapa en todas las partes del mapa (halls y pasillos), teniendo que valerse únicamente de la odometría visual solo por unos metros en el trayecto desde el hall central al derecho.

Capítulo 5. Experimentación

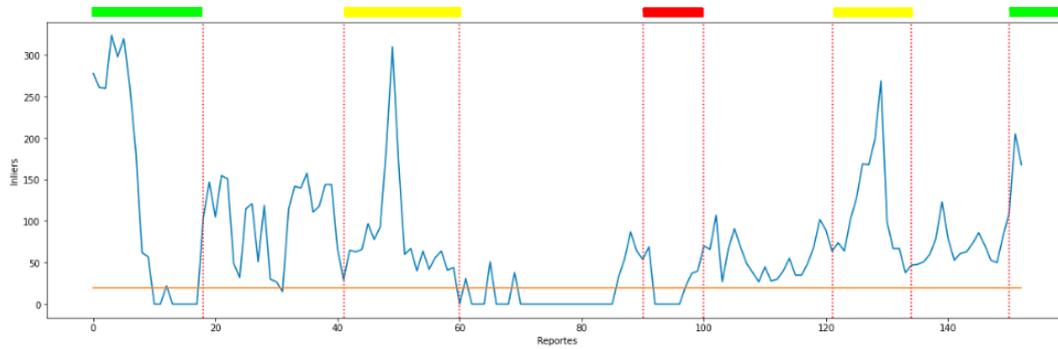


Figura 5.3: Cantidad de *Inliers* por reporte durante el día utilizando solo localización. Se resaltan en verde, amarillo y rojo las zonas correspondientes a los halls izquierdo, central y derecho respectivamente.

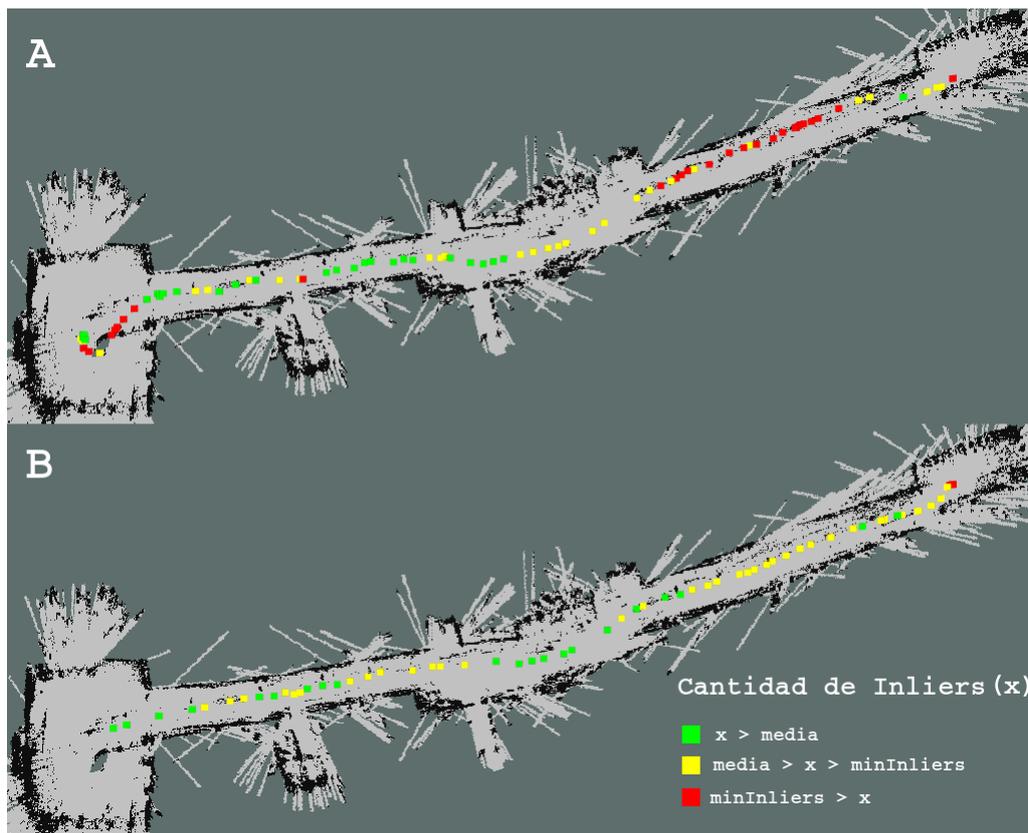


Figura 5.4: Resultados durante el día utilizando únicamente localización. **A.** *Inliers* reportados recorriendo desde el hall izquierdo al derecho, **B.** *Inliers* reportados recorriendo desde el hall derecho al izquierdo.

5.1. Prueba de mapeo y localización

Resultados en el día mapeando (dia-map)

En esta prueba el máximo de características encontradas es de **315**, el promedio **69.9** y la desviación estándar **73.0**. El sistema reporta más de *minInliers* en un **61 %** de los casos.

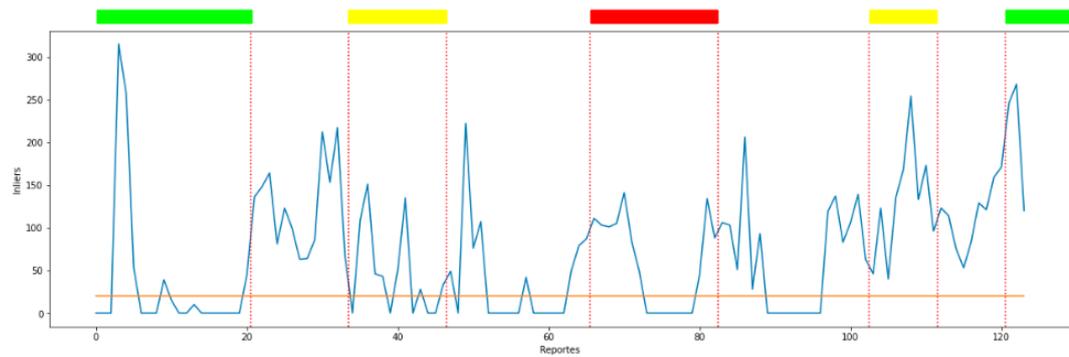


Figura 5.5: Cantidad de *Inliers* por reporte durante el día mapeando. Se resaltan en verde, amarillo y rojo las zonas correspondientes a los halls izquierdo, central y derecho respectivamente.

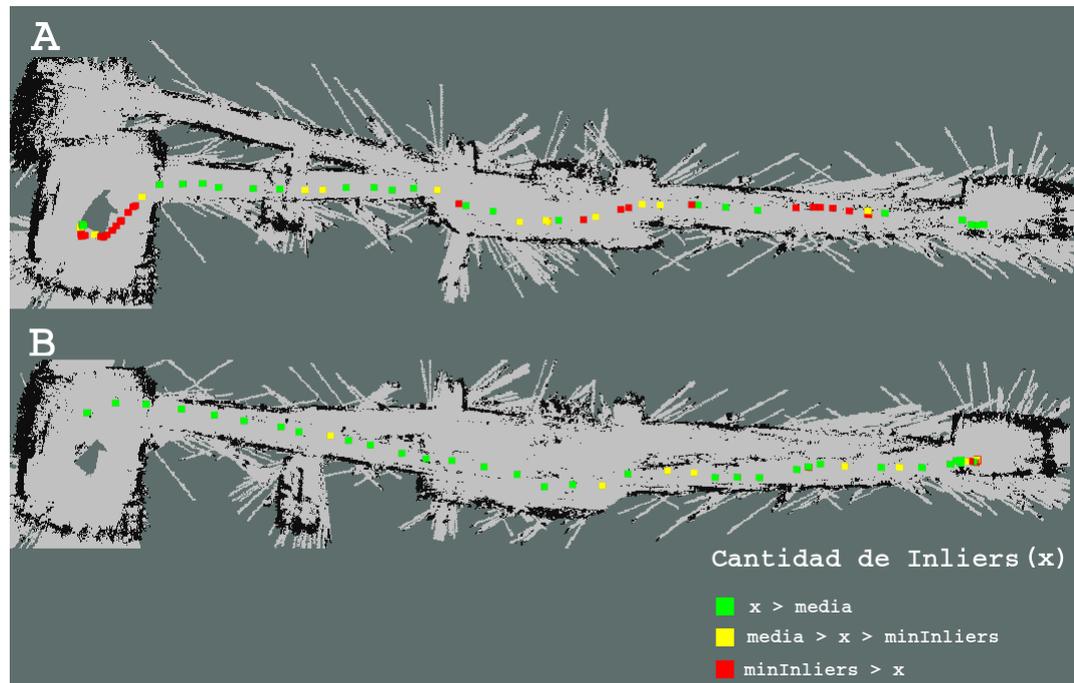


Figura 5.6: Resultados durante el día mapeando. **A.** *Inliers* reportados recorriendo desde el hall izquierdo al derecho, **B.** *Inliers* reportados recorriendo desde el hall derecho al izquierdo.

En la figura 5.6 (A) se observa que al finalizar el recorrido hacia el hall derecho, el mapa se encuentra en un estado incorrecto. Al ingresar al hall central el sistema

Capítulo 5. Experimentación

no encontró suficiente evidencia que respalde su (correcta) hipótesis de posición por lo que modifica el mapa, interpretando que llegó a ese lugar desde un pasillo diferente. Luego de esa errónea modificación, se encontraron coincidencias en todas las partes del mapa. Durante la recorrida de vuelta, el sistema logra ubicarse con gran número de coincidencias y con esto ajustar otra vez el mapa, unificando los dos pasillos antes creados en uno. El resultado final es un plano que respeta la realidad, ajustando los ángulos entre los halls y los pasillos, que en el mapa original distaban más de los 90 grados reales, aunque las dimensiones de los pasillos y el hall central se encuentran distorsionadas. La distorsión de los pasillos se debe al momento de unir los dos creados durante la ida; si bien el sistema logra juntarlos al encontrar cierres de ciclos en otras zonas, no es capaz de hacerlo dentro de los pasillos, por lo que aparece una suerte de duplicado de estos.

Resultados en la noche utilizando únicamente localización (noche-loc)

En esta prueba el máximo de características encontradas es de **235**, el promedio **48.4** y la desviación estándar **56.8**. El sistema reporta más de *minInliers* en un **52 %** de los casos.

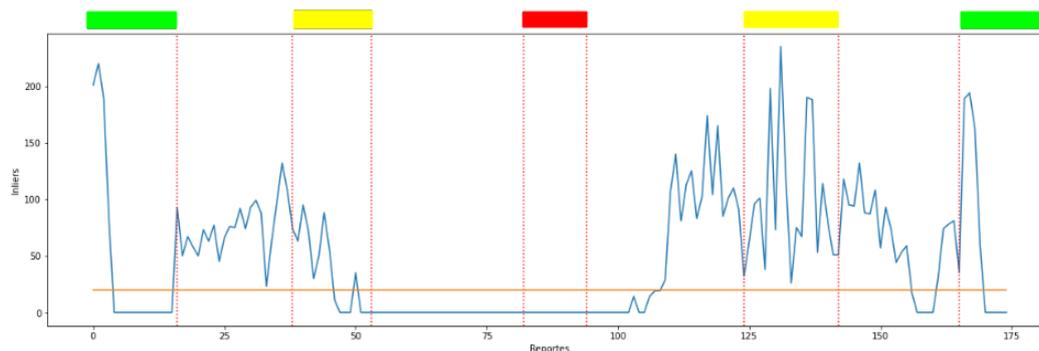


Figura 5.7: Cantidad de *Inliers* por reporte durante la noche utilizando solo localización. Se resaltan en verde, amarillo y rojo las zonas correspondientes a los halls izquierdo, central y derecho respectivamente.

En la figura 5.8 se presenta la posición del usuario (publicada por RTAB-Map) en el plano para cada uno de los reportes.

La información presentada en la figura 5.8 (A) muestra como al salir del hall central, el sistema es incapaz de encontrar una cantidad de coincidencias mínimas y debe recurrir a la odometría. Utilizando esta es capaz de correctamente identificar que se realiza un movimiento en línea recta (recorriendo el pasillo hacia el

5.1. Prueba de mapeo y localización

hall derecho), pero debido a una pobre detección de la dirección en el comienzo del mismo, se reportan posiciones por fuera del pasillo. Luego de dar la vuelta, en la figura B se muestra el momento en que el sistema logra localizarse otra vez, identificando correctamente su posición dentro del pasillo y continuando desde allí con una correcta localización por el resto del recorrido.

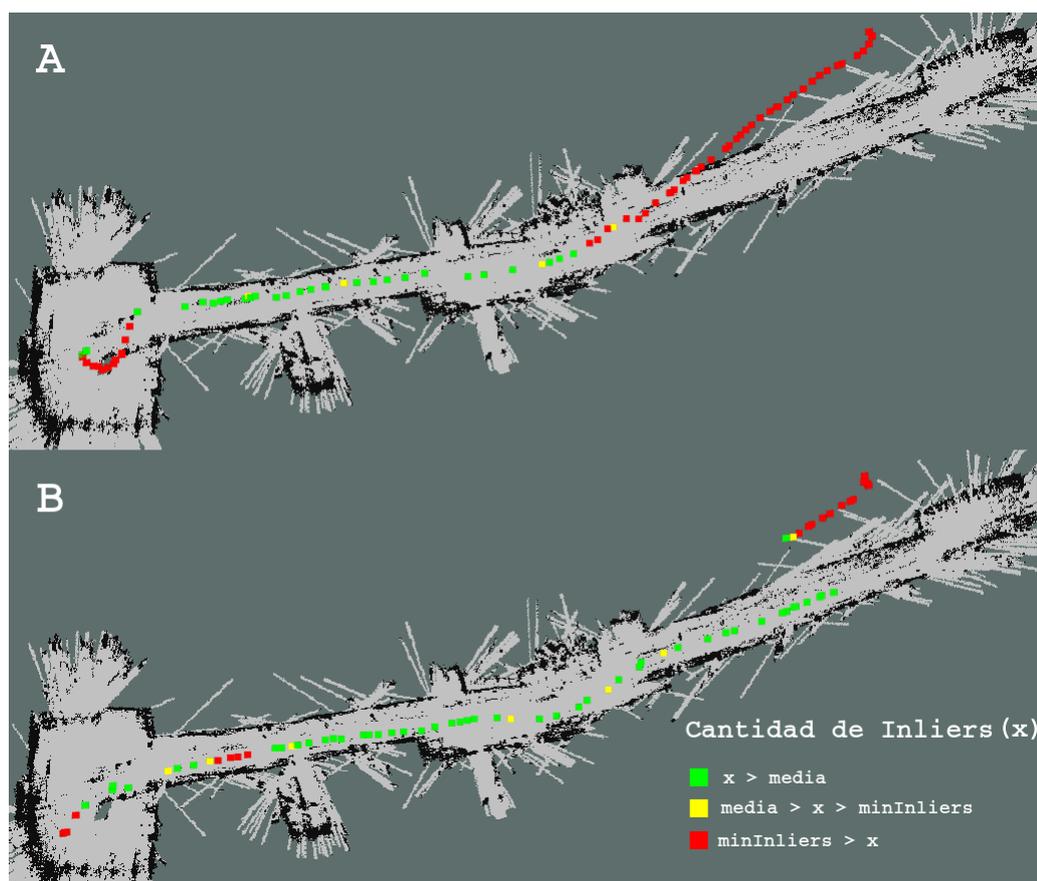


Figura 5.8: Resultados durante la noche utilizando únicamente localización. **A.** *Inliers* reportados recorriendo desde el hall izquierdo al derecho, **B.** *Inliers* reportados recorriendo desde el hall derecho al izquierdo.

Resultados en la noche mapeando (noche-map)

En esta prueba el máximo de características encontradas es de **264**, el promedio **45.0** y la desviación estándar **59.2**. El sistema reporta más de *minInliers* en un **47%** de los casos.

En la figura 5.10 se presenta la posición del usuario (publicada por RTAB-Map) en el plano para cada uno de los reportes.

Capítulo 5. Experimentación

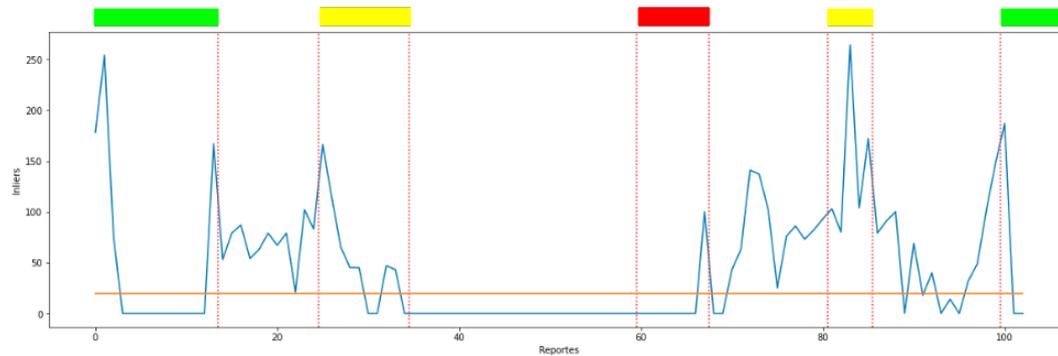


Figura 5.9: Cantidad de *Inliers* por reporte durante la noche mapeando. Se resaltan en verde, amarillo y rojo las zonas correspondientes a los halls izquierdo, central y derecho respectivamente.

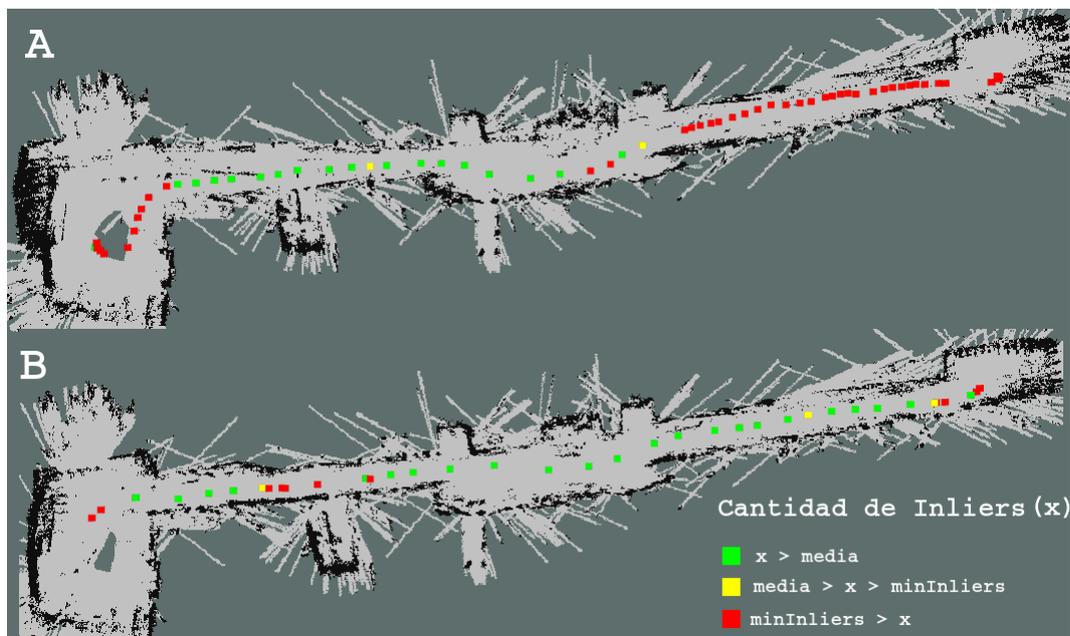


Figura 5.10: Resultados durante la noche mapeando. **A.** *Inliers* reportados recorriendo desde el hall izquierdo al derecho, **B.** *Inliers* reportados recorriendo desde el hall derecho al izquierdo.

Al igual que en el experimento anterior, se observa en la figura 5.10 (A) cómo el sistema se basa puramente en odometría durante todo el trayecto desde el hall central al derecho. La diferencia y razón por la cuál las posiciones reportadas son más ajustadas a las reales es que en este caso si se detectó correctamente la pose del usuario al entrar al pasillo. Los ajustes realizados al mapa acercan a este a la realidad, ajustando los ángulos entre pasillos y halls y agregando detalle en paredes y el hall central.

5.1.3. Análisis de resultados

Con las pruebas realizadas se buscó validar el sistema desarrollado en cuanto a su capacidad de construir mapas del entorno y localizarse en estos. El primer mapa construido, resultado de una única recorrida del lugar es considerado correcto, dado que este contiene una buena representación de la estructura del edificio en términos de los espacios libres identificados y las paredes que los delimitan. Esta información es suficiente para representar las salas de una casa y lograr navegar entre estas.

En cuanto a la localización, se evalúa al sistema iniciando con el mapa creado previamente y recorriendo el espacio bajo distintas iluminaciones. Para las pruebas realizadas durante el día, las posiciones reportadas coinciden con la real, mientras que durante la noche se registran algunos errores para estimar la ubicación. Es importante resaltar que en una situación de uso, el mapa debería ser construido realizando varias recorridas bajo distintas condiciones lumínicas, acumulando información con mayor variedad sobre cada punto del mismo.

En la tabla 5.1 se presenta el promedio de cantidad de *Inliers* reportados en cada zona (halls y pasillos) durante las recorridas realizadas. Allí se observa por que en el día las coincidencias son más abundantes en casi todos los casos, lo que explica el mejor desempeño del sistema. Resulta interesante notar que al recorrer el pasillo desde el hall central al derecho, se reporta una cantidad muy baja de *Inliers* sin importar la prueba. Esto puede deberse a que existen pocas señales visuales únicas en ese trayecto y que al final (en el hall derecho) se encuentra una ventana por donde ingresa una gran cantidad de luz natural, lo que dificulta la tarea de localización.

	día-loc	día-map	noche-loc	noche-map
Hall izquierdo (ida)	129.1 (132.8)	34.9 (83.5)	43.2 (79.6)	48.1 (83.4)
Pasillo	102.5 (45.7)	124.0 (51.3)	75.5 (23.4)	69.7 (20.7)
Hall central(ida)	86.2 (66.3)	45.8 (51.1)	38.3 (34.0)	52.5 (50.3)
Pasillo	11.9 (23.3)	34.7 (56.6)	0.0 (0.0)	0.0 (0.0)
Hall derecho	22.1 (24.9)	58.2 (51.5)	0.0 (0.0)	12.5 (33.1)
Pasillo	56.5 (24.5)	61.7 (61.0)	55.9 (57.1)	70.8 (43.1)
Hall central(vuelta)	115.9 (64.6)	129.8 (62.4)	99.8 (61.0)	144.6 (67.2)
Pasillo	67.4 (19.1)	114.3 (36.2)	62.3 (38.6)	53.0 (44.6)
Hall izquierdo(vuelta)	161.0 (39.1)	211.3 (65.2)	64.0 (79.8)	62.3 (88.2)

Tabla 5.1: Promedio (desviación estándar) de cantidad de *Inliers* reportados en cada zona (halls y pasillos) por experimento

En los casos donde se continuó construyendo el mapa mientras se recorría, se aprecia una mejoría en este. Si bien en puntos intermedios de las pruebas los mapas pueden encontrarse en estados erróneos, en cuanto se vuelve a visitar un espacio característico el sistema es capaz de unificar la información recibida y ge-

Capítulo 5. Experimentación

nerar un mapa más preciso. Esto refuerza la noción de que un mapa construido durante varias experiencias en el mismo edificio se acercará a la realidad [50], una característica deseable en un algoritmo de SLAM. El buen rendimiento de esta variante es de gran importancia, dado que a la hora de utilizar el sistema en un hogar, es probable que la estructura de este cambie continuamente y si el sistema se encuentra siempre construyendo el mapa, debería ser capaz de agregar estos nuevos obstáculos al mismo sin la necesidad de volver a construirlo desde cero.

La cantidad de pruebas no es suficiente para concluir sobre el desempeño del sistema a largo plazo, hace falta más experimentación para validar la robustez en el tiempo de los mapas y la precisión en la localización de los usuarios. De todos modos, estas experiencias resultan prometedoras y sugieren que la configuración e integración propuesta logra resolver la tarea de SLAM.

5.2. Prueba de integración

En búsqueda de validar la capacidad del sistema para funcionar en hogares y probar cada una de las partes del mismo, se realiza una experiencia de uso completa en una casa. El plano de esta es presentado en la figura 5.11, formada por una sala de estar (6mx3.5m), una cocina (3mx2.5m) y un patio (6mx3m).

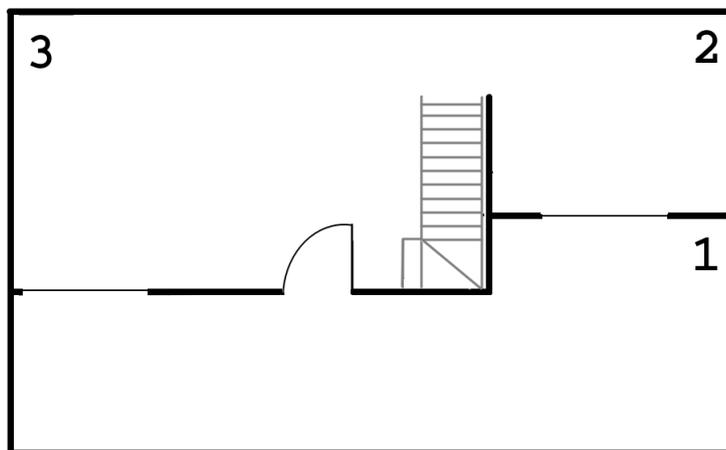


Figura 5.11: Plano de la casa. 1 representa el patio, 2 la cocina y 3 la sala de estar.

La prueba consiste en iniciar el sistema, sin ningún mapa construido y con el usuario parado en el centro de la sala de estar. Recorrer la casa una vez yendo desde allí a la cocina, saliendo al patio y volviendo a ingresar a la sala de estar (el orden, respetando lo mostrado en el plano sería 3, 2, 1, 3). Esto resultó en la grilla de ocupación presentada en la figura 5.12.

5.2. Prueba de integración

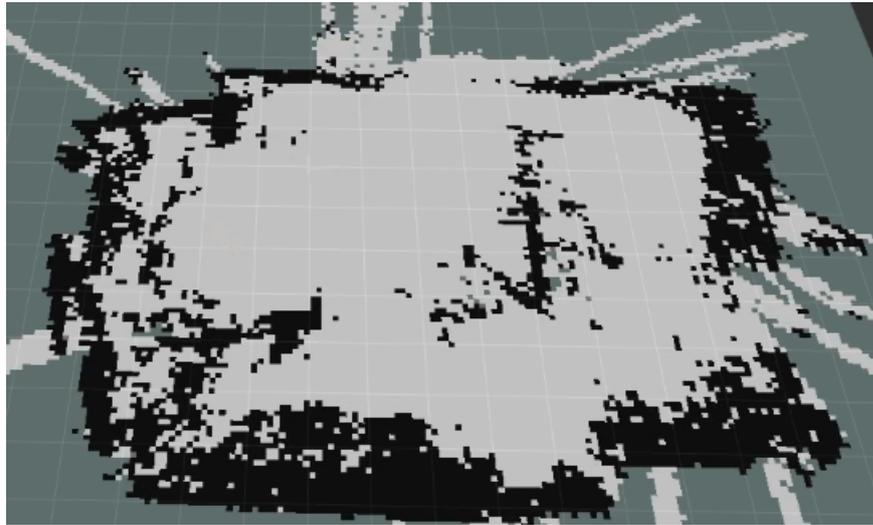


Figura 5.12: Grilla de ocupación de la casa generada por el sistema.

Durante la recorrida, se guardaron (utilizando el sistema de reconocimiento de voz) dos puntos: (I) "heladera", ubicado en la esquina inferior izquierda de la cocina y (II) "puerta principal", justo al frente de la puerta de entrada a la sala de estar. Luego se inicia el sistema con el usuario parado en la sala de estar, este se dirige a la salida desde la cocina al patio y se le solicita al sistema que nos guíe hacia el punto conocido como "puerta principal". La trayectoria calculada, junto con las líneas dibujadas se presentan en la figura 5.13. Allí se puede observar también la posición del usuario y la pose de su cabeza en forma de marcos de referencia.

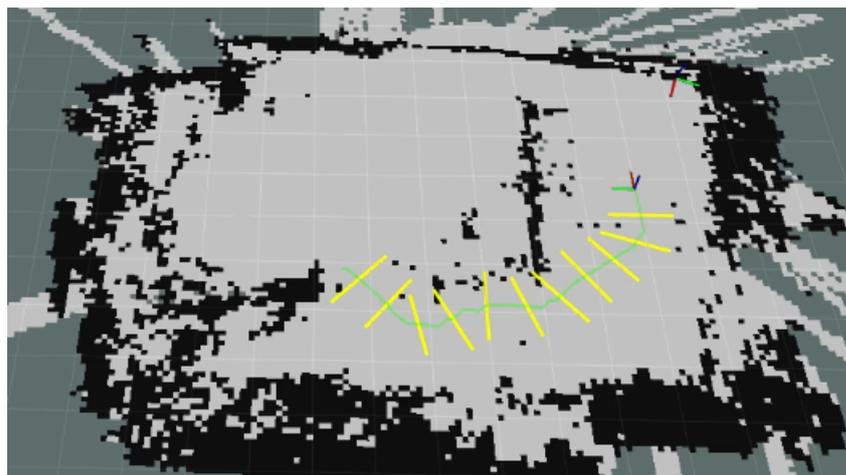


Figura 5.13: Trayectoria calculada desde la cocina hacia la puerta principal.

El resultado de esta prueba fue positivo. El sistema fue capaz de guiar al usuario hacia el destino elegido, demostrando para esto la capacidad de almacenar

Capítulo 5. Experimentación

correctamente los puntos pedidos y de proyectar los estímulos de forma precisa. La experiencia fue registrada en forma de video y es presentada en la demo del proyecto [51].

Capítulo 6

Conclusiones y trabajos a futuro

6.1. Conclusiones

En este proyecto, se buscó desarrollar un prototipo de dispositivo de señalización capaz de guiar al usuario dentro de entornos complejos con presencia de obstáculos. El objetivo principal es que el usuario pueda navegar desde su posición hacia un destino elegido, siguiendo los estímulos brindados, sin distraer su atención.

Un sistema con dichas características debe ser capaz de crear una representación del entorno en el que se encuentra y localizar al usuario en ella. Para esto se utilizó el paquete RTAB-Map, junto con un sensor ZED. La configuración de VSLAM implementada mostró ser robusta ante cambios de iluminación incluso en entornos monótonos, donde las características visuales únicas son escasas. Tanto en las pruebas dentro del Instituto de Computación como en la casa, el mapa construido representa fielmente la realidad y el sistema es capaz de ubicarse en él.

Con el objetivo de que los estímulos tomen en cuenta los obstáculos entre la posición actual y el destino elegido, se integró a la solución el paquete *Navigation*. Este se encarga de calcular trayectorias que pasen a una distancia segura de los obstáculos presentes en el mapa construido. Para lograr que los estímulos guíen al usuario a seguir la trayectoria, se utilizaron lentes de realidad aumentada y se implementó una aplicación nativa capaz de mostrar líneas perpendiculares a la trayectoria que generan la ilusión de estar pintadas en el suelo.

Durante la prueba de integración, se comprobó que al seguir los estímulos brindados por el sistema, se puede transitar hacia un destino elegido previamente. Esta experiencia no es suficiente para evaluar la usabilidad y se espera poder validarla con pacientes mediante encuestas de satisfacción dada su naturaleza cualitativa.

Por último, se considera que la interfaz mediante botones y comandos de voz es intuitiva y fácil de utilizar. Una vez más, se buscará validar esto con un grupo mayor de usuarios, pero es una tecnología ampliamente aceptada al día de hoy [52].

Capítulo 6. Conclusiones y trabajos a futuro

En términos generales, se considera que el prototipo implementado cumple con los requisitos relevados por el proyecto CSIC que lo enmarca y lo propuesto por los trabajos relevados. Se diseñó e implementó una interfaz simple e intuitiva que presenta estímulos capaces de guiar un usuario sin la necesidad de que este desvíe su atención. Se prevee que el dispositivo propuesto logre aliviar uno de los síntomas más debilitantes de la enfermedad de Parkinson, mejorando la calidad de vida de quienes lo utilicen.

6.2. Trabajos a futuro

Como se mencionó anteriormente, este proyecto de grado se encuentra enmarcado en un proyecto CSIC, el cuál continuará por un año más. Tanto para la continuación de dicho proyecto como para otras iniciativas que busquen impulsar este prototipo se proponen los siguientes trabajos a futuro:

- Validar lo implementado con un grupo de pacientes en un entorno controlado (previsto con la colaboración de los doctores del hospital de clínicas).
- Utilizar algún sistema de suavizado de trayectorias, teniendo en cuenta parámetros específicos de los usuarios como puede ser el largo del paso, velocidad de la marcha/giros.
- Evaluar otras formas de presentar estímulos en los lentes de realidad aumentada, como puede ser una flecha que indique hacia donde dirigirse (esto puede implementarse directamente sobre lo desarrollado).
- Considerar en el cálculo de trayectorias otros aspectos disparadores de FOG, como pueden ser los patrones de las paredes o el piso y aumentar el costo de atravesar estos lugares.
- Investigar sobre la posibilidad de solo presentar los estímulos cuando el usuario está transitando un episodio de congelamiento (podría utilizarse el algoritmo propuesto por Ahn *et al.* [31])
- Aprovechar la capacidad sensorial del dispositivo (los lentes cuentan con acelerómetro y giroscopio) para detectar posibles problemas como caídas y reportarlos.

Referencias

- [1] Werner Poewe, Klaus Seppi, Caroline M Tanner, Glenda M Halliday, Patrik Brundin, Jens Volkman, Anette-Eleonore Schrag, and Anthony E Lang. Parkinson disease. *Nature reviews Disease primers*, 3(1):1–21, 2017.
- [2] J. J. Rodríguez Castiblanco. Congelación de la marcha en pacientes institucionalizados y diagnosticados de la enfermedad de parkinson, 2017.
- [3] Rita Moretti, Paola Torre, Rodolfo M Antonello, Francesca Esposito, and Giuseppe Bellini. The on-freezing phenomenon: cognitive and behavioral aspects. *Parkinson's disease*, 2011, 2011.
- [4] Michael Macht, Yvonne Kaussner, Jens Carsten Möller, Karin Stiasny-Kolster, Karla Maria Eggert, Hans-Peter Krüger, and Heiner Ellgring. Predictors of freezing in parkinson's disease: a survey of 6,620 patients. *Movement Disorders*, 22(7):953–956, 2007.
- [5] Iria Cabo López, Pedro J García Ruiz, Silvia Vázquez Fernández del Pozo, and Vicenta Sánchez Bernardos. Motor complications in parkinson's disease: Ten year follow-up study. *Movement disorders*, 25(16):2735–2739, 2010.
- [6] N Giladi, MP McDermott, S Fahn, S Przedborski, J Jankovic, M Stern, C Tanner, Parkinson Study Group, et al. Freezing of gait in pd: prospective assessment in the datatop cohort. *Neurology*, 56(12):1712–1721, 2001.
- [7] Li-Li Zhang, S Duff Canning, and Xiao-Ping Wang. Freezing of gait in parkinsonism and its potential drug treatment. *Current neuropharmacology*, 14(4):302, 2016.
- [8] Elke Heremans, Alice Nieuwboer, and Sarah Vercruyse. Freezing of gait in parkinson's disease: where are we now? *Current neurology and neuroscience reports*, 13(6):350, 2013.
- [9] Alice Manson, Paola Stirpe, and Anette Schrag. Levodopa-induced-dyskinesias clinical features, incidence, risk factors, management and impact on quality of life. *Journal of Parkinson's disease*, 2(3):189–198, 2012.
- [10] Jorik Nonnekes, Anke H Snijders, John G Nutt, Günter Deuschl, Nir Giladi, and Bastiaan R Bloem. Freezing of gait: a practical approach to management. *The Lancet Neurology*, 14(7):768–778, 2015.

Referencias

- [11] Pharmacoeconomic review report. <https://www.ncbi.nlm.nih.gov/books/NBK539446/>.
- [12] Tamar C Rubinstein, Nir Giladi, and Jeffrey M Hausdorff. The power of cueing to circumvent dopamine deficits: a review of physical therapy treatment of gait disturbances in parkinson's disease. *Movement disorders: official journal of the Movement Disorder Society*, 17(6):1148–1160, 2002.
- [13] Alice Nieuwboer, Gert Kwakkel, Lynn Rochester, Diana Jones, Erwin van Wegen, Anne Marie Willems, Fabienne Chavret, Victoria Hetherington, Katherine Baker, and Inge Lim. Cueing training in the home improves gait-related mobility in parkinson's disease: the rescue trial. *Journal of Neurology, Neurosurgery & Psychiatry*, 78(2):134–140, 2007.
- [14] Lisette Bunting-Perry, Meredith Spindler, Keith M Robinson, Joseph Noorigian, Heather J Cianci, and John E Duda. Laser light visual cueing for freezing of gait in parkinson disease: a pilot study with male participants. *J Rehabil Res Dev*, 50(2):223–30, 2013.
- [15] Claudia Barthel, Jorik Nonnekes, Milou Van Helvert, Renée Haan, Arno Jansen, Arnaud Delval, Vivian Weerdesteijn, Bettina Debû, Richard Van Wezel, Bastiaan R Bloem, et al. The laser shoes: A new ambulatory device to alleviate freezing of gait in parkinson disease. *Neurology*, 90(2):e164–e171, 2018.
- [16] Pablo Arias and Javier Cudeiro. Effect of rhythmic auditory stimulation on gait in parkinsonian patients with and without freezing of gait. *PloS one*, 5(3), 2010.
- [17] Sinziana Mazilu, Ulf Blanke, Moran Dorfman, Eran Gazit, Anat Mirelman, Jeffrey M. Hausdorff, and Gerhard Tröster. A wearable assistant for gait training for parkinson's disease with freezing of gait in out-of-the-lab environments. *ACM Transactions on Interactive Intelligent Systems (TiiS)*, 5(1):1–31, 2015.
- [18] Lois Rosenthal, Dean Sweeney, Anne-Louise Cunnington, Leo R Quinlan, and Gearoid O'Laighin. Sensory electrical stimulation cueing may reduce freezing of gait episodes in parkinson's disease. *Journal of Healthcare Engineering*, 2018, 2018.
- [19] Martina Mancini, Katrijn Smulders, Graham Harker, Samuel Stuart, and John G Nutt. Assessment of the ability of open-and closed-loop cueing to improve turning and freezing in people with parkinson's disease. *Scientific reports*, 8(1):1–9, 2018.
- [20] Alberto J Espay, Yoram Baram, Alok Kumar Dwivedi, Rakesh Shukla, Maureen Gartner, Laura Gaines, Andrew P Duker, and Fredy J Revilla. At-home training with closed-loop augmented-reality cueing device for improving gait in patients with parkinson disease. *J Rehabil Res Dev*, 47(6):573–581, 2010.

- [21] Jean-Philippe Azulay, Serge Mesure, Bernard Amblard, Olivier Blin, Iban Sangla, and Jean Pouget. Visual control of locomotion in parkinson's disease. *Brain*, 122(1):111–120, 1999.
- [22] Anne Burleigh-Jacobs, Fay B Horak, John G Nutt, and Jose A Obeso. Step initiation in parkinson's disease: influence of levodopa and external sensory triggers. *Movement disorders*, 12(2):206–215, 1997.
- [23] Mitchell Glickstein and John Stein. Paradoxical movement in parkinson's disease. *Trends in neurosciences*, 14(11):480–482, 1991.
- [24] Alice Nieuwboer. Cueing for freezing of gait in patients with parkinson's disease: a rehabilitation perspective. *Movement disorders: official journal of the Movement Disorder Society*, 23(S2):S475–S481, 2008.
- [25] Marcelo P Pereira, Lilian TB Gobbi, and Quincy J Almeida. Freezing of gait in parkinson's disease: Evidence of sensory rather than attentional mechanisms through muscle vibration. *Parkinsonism & related disorders*, 29:78–82, 2016.
- [26] S. Fernández del Saz. Estrategias de movimiento compensadoras en el parkinson, uso de pistas sensoriales y cognitivas. <https://www.efisioterapia.net/articulos/estrategias-movimiento-compensadoras-el-parkinson-uso-pistas-sensoriales-y-cognitivas>, 2011.
- [27] Análisis de estado del arte. https://gitlab.fing.edu.uy/gtrinidad/proyecto-de-grado/-/blob/master/estado_del_arte/EstadoDelArte.pdf. Accessed: 2020-11-6.
- [28] Dean Sweeney, Leo R Quinlan, Patrick Browne, Margaret Richardson, Pauline Meskell, and Gearóid ÓLaighin. A technological review of wearable cueing devices addressing freezing of gait in parkinson's disease. *Sensors*, 19(6):1277, 2019.
- [29] Mon S Bryant, Diana H Rintala, Eugene C Lai, and Elizabeth J Protas. A pilot study: influence of visual cue color on freezing of gait in persons with parkinson's disease. *Disability and Rehabilitation: Assistive Technology*, 5(6):456–461, 2010.
- [30] Yan Zhao, Jorik Nonnekes, Erik JM Storcken, Sabine Janssen, Erwin EH van Wegen, Bastiaan R Bloem, Lucille DA Dorresteijn, Jeroen PP van Vugt, Tjitske Heida, and Richard JA van Wezel. Feasibility of external rhythmic cueing with the google glass for improving gait in people with parkinson's disease. *Journal of neurology*, 263(6):1156–1165, 2016.
- [31] DaeHan Ahn, Hyerim Chung, Ho-Won Lee, Kyunghun Kang, Pan-Woo Ko, Nam Sung Kim, and Taejoon Park. Smart gait-aid glasses for parkinson's disease patients. *IEEE Transactions on Biomedical Engineering*, 64(10):2394–2402, 2017.

Referencias

- [32] Rodney - a long time coming autonomous robot. <https://www.codeproject.com/Articles/2465567/Rodney-A-Long-Time-Coming-Autonomous-Robot-Part-8>.
- [33] Opencv depth map from stereo images. https://docs.opencv.org/master/dd/d53/tutorial_py_depthmap.html.
- [34] A summary of rtab-map as a rgbd-slam approach. <https://introlab.3it.usherbrooke.ca/mediawiki-introlab/images/3/31/Labbe2015ULaval.pdf>.
- [35] Mathieu Labbé and François Michaud. Memory management for real-time appearance-based loop closure detection. In *2011 IEEE/RSJ international conference on intelligent robots and systems*, pages 1271–1276. IEEE, 2011.
- [36] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-up robust features (surf). *Computer vision and image understanding*, 110(3):346–359, 2008.
- [37] Robot operating system. <http://wiki.ros.org/ROS/Introduction>.
- [38] Ros navigation. <http://wiki.ros.org/navigation>.
- [39] Setup and configuration of the navigation stack on a robot. <http://wiki.ros.org/navigation/Tutorials/RobotSetup>.
- [40] Transform. <http://wiki.ros.org/tf>. Accessed: 2020-11-6.
- [41] Walaa Gouda, Walid Gomaa, and Tetsuji Ogawa. Vision based slam for humanoid robots: A survey. In *2013 Second International Japan-Egypt Conference on Electronics, Communications and Computers (JEC-ECC)*, pages 170–175. IEEE, 2013.
- [42] Zed ros wrapper. <http://wiki.ros.org/zed-ros-wrapper>. Accessed: 2020-11-11.
- [43] The camera that senses space and motion. <https://www.stereolabs.com/zed/>.
- [44] Costmap 2d. http://wiki.ros.org/costmap_2d. Accessed: 2020-11-7.
- [45] Yan Zhao, Tjitske Heida, Erwin EH van Wegen, Bastiaan R Bloem, and Richard JA van Wezel. E-health support in people with parkinson’s disease with smart glasses: a survey of user requirements and expectations in the netherlands. *Journal of Parkinson’s disease*, 5(2):369–378, 2015.
- [46] Image geometry. http://wiki.ros.org/image_geometry. Accessed: 2020-11-6.
- [47] Api de reconocimiento de voz de google. <https://developer.android.com/reference/android/speech/SpeechRecognizer>. Accessed: 2020-11-6.

- [48] Moverio bt-300. <https://www.epson.es/products/see-through-mobile-viewer/gafas-moverio-bt-300>.
- [49] Plataforma jetson tx2 de nvidia. <https://www.nvidia.com/en-us/autonomous-machines/embedded-systems/jetson-tx2/>. Accessed: 2020-11-6.
- [50] Mathieu Labbe and François Michaud. Online global loop closure detection for large-scale multi-session graph-based slam. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2661–2666. IEEE, 2014.
- [51] Demo del proyecto. https://www.youtube.com/watch?v=j_CZte8wYjk&ab_channel=guillermotrinidad. Accessed: 2020-12-05.
- [52] François Portet, Michel Vacher, Caroline Golanski, Camille Roux, and Brigitte Meillon. Design and evaluation of a smart home voice interface for the elderly: acceptability and objection aspects. *Personal and Ubiquitous Computing*, 17(1):127–144, 2013.

Glosario

Fisiopatología: descripción de los mecanismos de producción de las enfermedades/síntomas.

FoG: del inglés *Freezing of Gait*, refiere al síntoma de congelamiento de la marcha.

On-FoG: manifestación de FoG cuando el paciente se encuentra bajo el efecto de medicación.

Off-FoG: manifestación de FoG cuando el paciente no se encuentra bajo el efecto de medicación.

Medicación dopaminérgica: refiere a los tratamientos farmacológicos que regulan la actividad de la dopamina.

Estímulos somatosensoriales: utilizado para referirse a estímulos sobre el tacto, en general en forma de vibraciones.

Entradas propioceptivas: información recibida por el cerebro sobre el posicionamiento y movimiento de las partes del cuerpo.

Distancia de levenshtein: es el número mínimo de operaciones (inserción, eliminación o sustitución) requeridas para transformar una cadena de caracteres en otra.

Odometría: la estimación de la posición del robot con respecto al punto donde este inicia su ejecución.

SLAM: del inglés *Simultaneous Localization and Mapping*, refiere a un conjunto de técnicas utilizada por los sistemas autónomos para construir un mapa del entorno a la vez que lo recorren.

VSLAM: del inglés *Visual Simultaneous Localization and Mapping*, refiere a las técnicas de SLAM que utilizan cámaras como fuente sensorial.

Capítulo 6. Glosario

Paquete: implementaciones *open source* de algoritmos y funcionalidades comunes.

Nodo: proceso ejecutado dentro de ROS.

Tópico: cola de mensajes utilizada para la comunicación entre nodos de ROS.

Índice de tablas

2.1. Evaluación del color del láser proyectado desde un bastón, Bryant <i>et al.</i>	6
5.1. Promedio (desviación estándar) de cantidad de <i>Inliers</i> reportados en cada zona (halls y pasillos) por experimento	43

Índice de figuras

2.1. Bastón proyectando una línea roja utilizada en el trabajo desarrollado por Bryant <i>et al.</i> (Imagen tomada de [29]).	6
2.2. Señalización en Google Glass por Zhao <i>et al.</i>	8
2.3. Ajuste de proyección según el ángulo de la cabeza, Ahn <i>et al.</i>	9
3.1. Ejemplo de una grilla de ocupación.	12
3.2. Ejemplo de imagen de profundidad	13
3.3. Ejemplo de odometría visual	13
3.4. Diagrama de funcionamiento de RTAB-Map	14
3.5. Ejemplo de mapas creados en RTAB-Map	15
3.6. Diagrama del stack de navegación ROS	16
3.7. Ejemplo de uso de transform	17
4.1. Estímulos de los <i>Laser Shoes</i>	20
4.2. Estímulos propuestos	20
4.3. Sensor de cámara binocular ZED	21
4.4. Diagrama de componentes VSLAM	22
4.5. Diagrama de componentes de <i>Navigation</i> para la solución planteada	23
4.6. Ejemplo de mapa de costos.	24
4.7. Ejemplo de estímulos sobre una trayectoria.	24
4.8. Montaje del sensor ZED sobre los Moverio BT-300.	25
4.9. Diagrama de transformaciones y marcos de referencia.	26
4.10. Proceso de proyección de las líneas en una cámara.	27
4.11. Diagrama de componentes del módulo de generación de estímulos.	28
4.12. Diagrama de secuencia para el guardado de un punto.	30
4.13. Control de los lentes Moverio BT-300	31
4.14. Diagrama de comunicación de la interfaz	31
4.15. Diagrama del sistema	32
4.16. Componentes de hardware utilizados durante el desarrollo	33
5.1. Plano del Instituto de Computación	36
5.2. Grilla de ocupación construida del Instituto de Computación	36
5.3. Cantidad de <i>Inliers</i> por reporte durante el día utilizando solo localización	38
5.4. Resultados durante el día utilizando únicamente localización	38

Índice de figuras

5.5. Cantidad de <i>Inliers</i> por reporte durante el día mapeando	39
5.6. Resultados durante el día mapeando	39
5.7. Cantidad de <i>Inliers</i> por reporte durante la noche utilizando solo localización	40
5.8. Resultados durante la noche utilizando únicamente localización . .	41
5.9. Cantidad de <i>Inliers</i> por reporte durante la noche mapeando	42
5.10. Resultados durante la noche mapeando	42
5.11. Plano de la casa	44
5.12. Grilla de ocupación de la casa generada por el sistema.	45
5.13. Trayectoria calculada desde la cocina hacia la puerta principal. . .	45