



Universidad de Valladolid

**ESCUELA TÉCNICA SUPERIOR DE INGENIEROS
INDUSTRIALES**

**DEPARTAMENTO DE INGENIERÍA DE SISTEMAS Y
AUTOMÁTICA**

TESIS DOCTORAL:

**EXPLORACION Y RECONSTRUCCION
TRIDIMENSIONAL DE ENTORNOS EXTERIORES
MEDIANTE ROBOTS MOVILES**

Autor:

JAIME PULIDO FENTANES

Ingeniero Mecatrónico

Dirigida por los profesores:

Dr. D. EDUARDO ZALAMA CASANOVA

Dr. D. JAIME GÓMEZ GARCÍA-BERMEJO

Valladolid, 2012

Agradecimientos

Quiero agradecer a todas las personas que han colaborado en la realización de esta tesis doctoral, bien sea aportando su ayuda y conocimientos o dándome su apoyo.

En primer lugar quiero agradecer a mis directores de tesis los profesores D. Eduardo Zalama Casanova y D. Jaime Gómez García-Bermejo, directores también de la división de Robótica y Visión Artificial de CARTIF, por su apoyo y sugerencias para la realización de esta tesis.

Igualmente, quiero agradecer a mis compañeros de división en CARTIF por su apoyo y colaboración, en especial a Salva y Sergio sin quienes la construcción de “Rodolfo” hubiese sido imposible, a Raúl que soporto conmigo el terrible frío de Enero durante los experimentos en “El Pinaron”, a Samuel quien siempre me dio sabios consejos y apoyo, a Roberto Pinillos y Álvaro porque siempre estuvieron ahí para ayudarme y finalmente a José Llamas y Pedro por su invaluable colaboración en todos los aspectos del 3D.

Igualmente quiero agradecer de manera muy especial a mi familia quienes siempre me han apoyado, dado ánimos y han sido mi mayor fuente de motivación.

TRIBUNAL:

Resumen

El presente trabajo de tesis presenta un método de exploración de entornos exteriores con el objetivo de generar modelos tridimensionales de los mismos mediante escaneado láser. La metodología propuesta está orientada a optimizar el proceso de exploración maximizando la calidad del modelo y reduciendo el número de escaneos necesarios para crear un modelo 3D con una calidad mínima requerida.

Para lograr este propósito el método propuesto en este trabajo utiliza la información capturada por un robot móvil todoterreno dotado con un escáner láser tridimensional. Esta información es procesada y analizada para extraer características del entorno que son relevantes para el desarrollo de su misión (como obstáculos, superficies navegables o la calidad de modelo en una zona del entorno), las cuales son proyectadas a un mapa de información 2D que tiene el propósito de facilitar los cálculos de evaluación de distintas posiciones candidatas a ser el próximo punto de recogida de datos dentro del entorno.

Para la evaluación de estas posiciones candidatas, se usan como criterios la ganancia esperada de información, la mejora esperada de la calidad del modelo, la cobertura de zonas marcadas como de especial interés y el coste estimado de la trayectoria hasta dicho punto. La posición seleccionada será la que obtenga una puntuación más alta de acuerdo con estos criterios.

Para la verificación de esta metodología y dentro del marco de esta investigación se ha construido una plataforma robótica todoterreno orientada a la reconstrucción tridimensional de entornos exteriores. Para ello se ha dotado a la plataforma con un escáner láser tridimensional diseñado especialmente para esta aplicación y compuesto por un escáner láser planar instalado sobre una plataforma giratoria que permite una adquisición de información del entorno rápida y completa. El proceso de construcción de esta plataforma así como la estructura mecánica del robot, su arquitectura hardware software y de comunicaciones y los sistemas perceptivos embarcados también forman parte de este trabajo. Asimismo, se ha diseñado una herramienta de simulación muy útil para el desarrollo y las pruebas iniciales cuya descripción igualmente se encuentra consignada en este trabajo.

Finalmente, se presentan resultados obtenidos con la plataforma robótica y el simulador en diferentes tipos de entornos exteriores, los cuales demuestran la utilidad de esta metodología para la generación de modelos tridimensionales de diferentes tipos de entorno.

Tabla de Contenidos

CAPÍTULO 1. INTRODUCCIÓN.....	1
1.1. Introducción	1
1.2. Objetivos	6
1.3. Estructura y Contenidos	6
CAPÍTULO 2. MARCO CONCEPTUAL.....	9
2.1. Introducción	9
2.1.1. Localización, Mapeado y SLAM	11
2.1.2. Mapeado de Exteriores	16
2.1.3. Exploración	18
2.2. Trabajos Relacionados	25
2.3. Metodología Propuesta	35
2.4. Conclusiones del Capítulo	37
CAPÍTULO 3. ANÁLISIS TRIDIMENSIONAL	39
3.1. Tratamiento Previo Del Modelo	40
3.1.1. Construcción de la Malla	40
3.1.2. Alineamiento y Rotación Del Modelo	41
3.2. Análisis De Las Características Geométricas Del Modelo	45
3.3. Extracción De Información Del Modelo	47
3.3.1. Extracción De Obstáculos Y Superficies Navegables	47
3.3.2. Análisis de la Calidad del Modelo	54
3.3.3. Extracción de Zonas de Interés	58
3.4. Conclusiones del Capítulo	60
CAPÍTULO 4. EVALUACIÓN DE CANDIDATOS	61
4.1. Generación Del Mapa De Información 2D	62
4.1.1. Extracción del Mapa de Navegabilidad	62
4.1.2. Evaluación de la Calidad del Modelo	63
4.2. Evaluación de Posiciones Candidatas	64
4.2.1. Estimación del Coste de la Trayectoria	65
4.2.2. Ganancia Esperada de Información	67

4.2.3.	Ganancia Esperada de Calidad del Modelo	68
4.2.4.	Análisis de la cobertura de Zonas de Interés	69
4.3.	Selección de NBV y Selección del criterio de Parada	69
4.4.	Conclusiones del Capítulo	70
CAPÍTULO 5. MARCO EXPERIMENTAL.....		71
5.1.	Motivación	71
5.2.	Robot Todoterreno	77
5.2.1.	Características generales de la plataforma móvil	78
5.2.2.	Arquitectura de Control	80
5.2.3.	Sistemas Perceptivos	86
5.2.4.	Sistema de comunicaciones	90
5.3.	Escáner 3D	94
5.4.	Simulador	97
5.4.1.	Características del Simulador	98
5.5.	Conclusiones del Capítulo	104
CAPÍTULO 6. RESULTADOS.....		107
6.1.	Ajuste de los Parámetros	107
6.2.	Entornos Simulados	109
6.2.1.	Entorno Simple	110
6.2.2.	Entorno Estructurado	114
6.2.3.	Entorno Desestructurado	117
6.3.	Entornos Reales	119
6.3.1.	Edificio de la Fundación CARTIF	120
6.3.2.	El “Pinarón”	123
6.3.3.	Análisis de los Tiempos de Ejecución	128
6.4.	Conclusiones del Capítulo	129
CAPÍTULO 7. CONCLUSIONES.....		131
7.1.	Contribuciones de la Tesis	133
7.2.	Trabajo Futuro	134
PUBLICACIONES		136
REFERENCIAS.....		138

Capítulo 1.

INTRODUCCIÓN

1.1. Introducción

El modelado tridimensional de entornos de exterior está cobrando una importancia creciente. Día a día se percibe un incremento de su aplicación tanto en el campo de la robótica como en otras disciplinas y áreas profesionales de gran interés práctico tales como el urbanismo, las obras públicas, la arquitectura inversa o la arqueología.

Este aumento en la utilización de los modelos 3D para la descripción de entornos se debe en gran medida a que se ha descubierto su enorme valor como herramienta descriptiva de un entorno. Adicionalmente, en este posicionamiento han concurrido otros factores como puede ser la aparición de nuevos medios de soporte y visualización de la información tridimensional, la constante reducción de precios en diferentes sensores que pueden facilitar la generación de modelos 3D (cámaras, GPS, inclinómetros, escáneres láser, etc.) y un mayor conocimiento sobre las técnicas a emplear para obtener mapas de entorno cada vez más ajustados a la realidad y por lo tanto más confiables.

Inicialmente la estereoscopía ha sido la técnica más utilizada para recoger información tridimensional, en la que la información de profundidad se puede obtener a partir de dos imágenes capturadas desde posiciones ligeramente desplazadas. Esta técnica si bien se sigue empleando en la actualidad y particularmente en el ámbito de la construcción. Sin embargo la aparición de los sistemas de escaneado láser ha permitido obtener modelos del entorno con mayor precisión especialmente en la medida de profundidad.

Entre los sistemas de escaneado láser destacan los que se basan en tecnología de diferencia de fase y los que se basan en tiempo de vuelo. Los primeros utilizan la emisión de trenes de onda de diferente amplitud que al reflejarse en un objeto permiten computar de forma precisa la

distancia hasta las superficies reflectantes, utilizando el cambio de fase de la señal. No obstante esta técnica se ve muy influenciada por la luz ambiental, lo que puede restringir trabajos en exteriores. En el caso de los escáneres de tiempo de vuelo se mide el tiempo transcurrido entre la emisión y la recepción de un haz láser. Aprovechando que se conoce su velocidad de propagación, es posible calcular directamente la distancia al objeto. Así, mediante el barrido automático del emplazamiento a estudio con dicho haz se obtienen las coordenadas geométricas (X, Y, Z) de cada punto por donde discurre, respecto a la posición del escáner. Este proceso permite la obtención de una nube de puntos cuya densidad esta directamente relacionada con el número de datos adquiridos.

Dicho esto y a pesar del aumento en la demanda de este tipo de modelos, la recogida de información que puede permitir el desarrollo de una descripción de entorno 3D plantea grandes retos. Entre estas dificultades se destaca por encima de todas la recogida de información de un entorno real. Por una parte los equipos utilizados para este propósito son costosos y requieren un manejo especializado. Por otra parte, puesto que no es posible conseguir toda esta información desde un único punto de vista, es necesario situar el escáner en diferentes posiciones respecto al objeto para obtener las nubes de puntos correspondientes a las vistas parciales correspondientes a cada una de esas posiciones. Esto aumenta los requerimientos de tiempo y de recursos humanos debido a la necesidad de transportar los equipos de medición de un punto a otro y al necesario pos-procesamiento de la información para el alineamiento de los datos de las distintas tomas (Figura 1-1).



Figura 1-1: Vistas alineadas de la iglesia de Valberzoso (Palencia) adquiridas con un escáner Leica hds3000

Con el fin de obviar en todo lo posible estos inconvenientes, autores como (Blaer & Allen, 2007) y (Nüchter, Lingemann, Hertzberg, & Surmann, 2007) han propuesto el uso de robots móviles para facilitar el proceso de reconstrucción de entornos exteriores. Al aprovechar los recursos que ofrece el robot como son movilidad, energía, localización y sistema informático

los procesos de recolección de información y elaboración de modelos 3D se agilizan y aumentan su fiabilidad.

Ahora bien, el uso de robots móviles para capturar la información tridimensional de un entorno, no es un valor añadido sino más bien una condición necesaria para que un robot puedan desarrollar una tarea de forma autónoma. Así pues, la relación entre sistemas de recogida de información y robots móviles es una relación complementaria que puede aportar grandes beneficios para ambos campos.

En el contexto de la construcción de modelos tridimensionales del entorno, el problema de la toma de decisiones puede ser planteado en términos de la elección entre las diferentes vistas que puede tomar el robot. De entre estas, el robot elegirá aquella que maximice la utilidad de la nueva información sensorial y conserve el esfuerzo necesario para alcanzar la posición dentro de límites razonables, de cara a optimizar el tiempo necesario para explorar el entorno completamente, orientando así la planificación y la ejecución del movimiento hacia descubrir las porciones del entorno que son desconocidas.

A pesar de que existen numerosas aplicaciones en las que los robots han sido diseñados para explorar el entorno autónomamente, tales como exploración planetaria, misiones de búsqueda y rescate o tareas de vigilancia o supervisión de ecosistemas, la realización de esta exploración autónoma aún debe mejorar sus parámetros de ejecución, pues, en la mayoría de los casos existe un supervisor humano que define el recorrido a realizar el robot o bien interviene en la definición de este recorrido de una u otra forma. Para este propósito se requiere el desarrollo de una solución que permita al robot realizar una reconstrucción tridimensional del entorno de operación atendiendo a criterios como calidad del modelo y movimientos a realizar.

La selección del proceso de exploración está fundamentada sobre el proceso de modelado tridimensional de un entorno, lo cual es una labor ardua y dispendiosa. Aunque los sistemas de captura de información láser permiten el escaneado del entorno a una resolución deseada, la instalación de una estación de medición típica requiere de tiempo y una cierta planificación ya que estos dispositivos son pesados y exigen la instalación de equipos auxiliares (trípode, baterías, GPS y ordenador). Lo anterior constituye un gran inconveniente especialmente en el caso de grandes estructuras exteriores, en las cuales el acceso o la presencia de humanos puede ser difícil o incluso peligrosa.

Un inconveniente adicional lo constituye el hecho de que, por lo general la reconstrucción completa de un escenario exige su medición desde distintos puntos de vista. De esta forma el operario debe seleccionar diversas posiciones en base a su experiencia y a la tipología del entorno. Esta selección ha de realizarse sin poder revisar a fondo el modelo resultante entre toma y toma, debido a la precariedad de las condiciones de trabajo (falta de ergonomía, meteorología adversa, poca visibilidad por la incidencia de rayos solares sobre pantalla de visualización, etc.). Todo esto limita la capacidad del operador de evaluar las mejores ubicaciones de captura en términos de calidad del modelo, menor número de tomas, distancia recorrida y nivel de detalle.

De forma adicional, dentro de la robótica móvil, la robótica de exteriores es la que presenta los mayores desafíos en lo que a movilidad y seguridad se refiere, y es tal vez la que ha recibido menor atención por parte de la comunidad investigadora. Los entornos exteriores representan un ámbito de aplicación paradigmático para la robótica de servicios por su interés práctico, incluso cuando los seres humanos están incluidos en el lazo de control (Birk & Kenn, 2002). De hecho encontramos aplicaciones en las que robots teleoperados o semi-autónomos tienen una gran utilidad, por ejemplo en labores de intervención y rescate en situaciones de riesgo (Amano H. , 2002) (Ko & Lau, 2009). En este tipo de situaciones, es necesario disponer de la mayor cantidad posible de información sobre las condiciones del entorno, antes de planear una intervención. Sin embargo contar con un conocimiento detallado y confiable entraña enormes dificultades a cuya superación puede contribuir de forma decisiva la construcción de modelos 3D.

En los escenarios de emergencias, los recursos humanos siempre son limitados y por tanto, la utilidad real de un sistema es directamente proporcional a su autonomía (Bruemmer, Dudenhoeffer, & Marble, 2002) (Goodrich, Olsen Jr., Crandall , & Palmer, 2001). Por tanto, la respuesta adecuada a estas necesidades pasará por desarrollar robots que tengan la capacidad de trabajar con un alto grado de autonomía, aún en condiciones imprevistas y variables, y que interactúen con entornos cambiantes y no conocidos de antemano, sin alterar la efectividad de sus acciones ni la fiabilidad de la información que suministren.

Por lo general, las aplicaciones de la robótica de exteriores deben realizarse en entornos más extensos, menos estructurados, menos controlados y menos predecibles que los entornos interiores. Esto ha hecho necesario realizar avances en el desempeño de los robots móviles y de forma especial en su grado de autonomía. Aun así, lograr que un robot pueda trabajar autónomamente en estas condiciones es todavía una labor de gran nivel de complejidad.

En concreto, es un requisito imprescindible realizar, desde la concepción misma del proyecto, un cuidadoso análisis de las operaciones a las que va a ser destinado prioritariamente el robot y de los entornos en los que estas funciones se desarrollarán. Esta reflexión previa permitirá determinar claramente los aspectos que requieran una atención prioritaria, tanto en lo estructural como en lo operativo.

En este contexto, la primera necesidad identificada en la presente investigación es la planificación de acciones para el modelado del terreno. La adquisición del mapa del entorno es una fase crucial para poder navegar con seguridad. Sin ella, resulta muy difícil que los robots puedan llevar a cabo cualquier tarea que requiera desplazamientos sea cual sea el tipo de entorno, ya que la capacidad del robot para cumplir adecuadamente una misión sin la colaboración de un operador humano supone que esté dotado con la habilidad de desplazarse de forma rápida y segura sobre el entorno y generalmente, esta capacidad se soporta sobre la adquisición y el procesamiento de la información de este entorno.

La construcción de mapas del terreno no es una tarea sencilla, ya que para generar mapas fiables es necesaria una buena estimación de la posición desde la cual se están capturando estos datos; y para tener una buena estimación es importante, a su vez, contar con un mapa que ofrezca total fiabilidad. Este problema es bien conocido en el mundo de la robótica móvil y

la forma de resolverlo constituye prácticamente una disciplina dentro de esta, donde es conocido como SLAM (Simultaneous Localization and Mapping).

El SLAM para entornos de exteriores o SLAM-6D¹ exige el manejo de información en tres dimensiones, ya que los desplazamientos del robot se pueden dar sobre seis grados de libertad. Además, para el desplazamiento del robot es necesario considerar todo el entorno y no solamente los obstáculos que se encuentren a una altura determinada, como sucede cuando se trabaja en entornos interiores.

A pesar de los importantes resultados obtenidos con la aplicación de las técnicas del SLAM-6D, aún persisten problemas como los derivados de la ubicación óptima de los sensores para la reconstrucción tridimensional de un entorno. Además, el problema se agudiza cuando se trata de operar el robot en entornos exteriores, debido a que estos entornos son con frecuencia muy desestructurados, no presentan suelos totalmente planos sino con desniveles u obstáculos, no están necesariamente restringidos por límites físicos y generalmente contienen múltiples elementos no previstos pero muy relevantes, que obstaculizan o sirven como referentes para la navegación del robot (como árboles, rocas, coches, etc.).

La operación en entornos exteriores también conlleva dificultades respecto a la elaboración de modelos tridimensionales debido a las múltiples oclusiones que generan los elementos que se encuentran en el área de trabajo. En contraposición, el problema de la localización se ha ido simplificando gracias al uso de dispositivos como los DGPS (Sistemas de Posicionamiento Global Diferencial) y las brújulas que proveen un posicionamiento razonablemente preciso.

Por otra parte, la exploración del entorno es una cuestión que sorprendentemente, ha sido mucho menos estudiada que el SLAM, a pesar de ser un tema crucial a la hora de desplegar robots autónomos en entornos desconocidos. De hecho, para que el robot sea realmente autónomo, tiene que poder decidir por sí mismo los movimientos necesarios para construir el modelo. De aquí se desprende que todo robot que desempeñe misiones en entornos desconocidos debe contar con esta habilidad básica pues de otra manera le resultará imposible alcanzar su objetivo. Además, el robot debe tomar decisiones sobre su propia operación, a partir de una información que no ha sido previamente incorporada en su sistema sino que él mismo se encarga de recolectar y procesar. Esta capacidad está íntimamente relacionada con el incremento de la autonomía del robot, y la adopción de determinaciones debe estar basada en la información del terreno, en la información disponible sobre el objetivo y en una serie de reglas que establezcan la relación entre ambas.

¹ La expresión 6D se refiere al número de grados de libertad al considerar que son seis las tres coordenadas cartesianas {x, y, z} y los giros respecto a cada una de estas {yaw, roll y pitch}

1.2. Objetivos

En este marco general la investigación plasmada en la presente tesis tiene como objetivo central el desarrollo de una metodología que permita la obtención de modelos tridimensionales de entornos de exterior mediante un robot móvil, en forma eficiente, estimando las siguientes mejores posiciones para una captura de datos a partir de la información recogida por el mismo sistema.

Para alcanzar este objetivo se debe desarrollar un procedimiento de exploración basado en la información tridimensional del terreno adquirida por el robot. La metodología propuesta busca optimizar el proceso de exploración y reconstrucción de entornos exteriores permitiendo a un robot predecir qué movimientos le aportarán una mayor cantidad de información sobre el terreno, reduciendo el tiempo y el consumo de recursos y mejorando la calidad final del modelo. Dicha optimización se centra en la reducción del número de capturas de información y la distancia a recorrer por el robot, y en la maximización de la calidad del modelo final. El uso de estos criterios permite la reducción del consumo energético y el tiempo de operación.

Manteniendo como eje central el diseño de un procedimiento de exploración, el presente trabajo incluye también la descripción del robot que se ha desarrollado para validar la investigación en condiciones reales de experimentación. Este robot cuenta con una arquitectura y un conjunto de herramientas que lo hacen especialmente indicado para las labores exploración autónoma. Asimismo, se propone el diseño de un sistema para la adquisición de la información tridimensional del entorno de trabajo del robot.

Dentro de esta investigación se propone también la construcción de una herramienta de simulación tridimensional que permite el ensayo y la validación de los procedimientos de exploración de entornos previamente a su incorporación robots reales, lo cual es sumamente útil durante la etapa de desarrollo de los distintos algoritmos de exploración ensayados a lo largo de la investigación.

1.3. Estructura y Contenidos

En el Capítulo 2 se presenta una revisión del marco conceptual y trabajos relacionados en el ámbito de la navegación y reconstrucción tridimensional, incluyendo, una introducción a los conceptos necesarios para entender las técnicas de reconstrucción de modelos tridimensionales de exteriores usando robots móviles. Finalmente en este mismo capítulo se encuentra una introducción a la metodología propuesta para la exploración de exteriores.

El Capítulo 3 describe la metodología que hemos desarrollado para el análisis de los datos tridimensionales capturados por el robot. En este proceso se extraen las superficies

navegables y los obstáculos, se cuantifica la calidad por punto del modelo y se determinan las zonas de interés para futuros escaneos.

El Capítulo 4 contiene una pormenorizada descripción de la solución empleada para la generación de un mapa de información 2D. Este mapa se emplea para valorar la navegabilidad sobre el entorno, generar posiciones candidatas a ser la próxima vista y para la evaluación de las mismas.

En el Capítulo 5 se detallan los sistemas experimentales desarrollados durante esa investigación, para la validación en condiciones reales de experimentación. Por una parte se presenta el robot todoterreno diseñado para la reconstrucción de exteriores, junto con las características de sus sistemas y componentes. Forman parte de este capítulo las distintas consideraciones hechas durante la etapa de diseño del sistema, así como los detalles constructivos del robot. El contenido de este capítulo se complementa con la presentación de una herramienta de simulación con el objeto de analizar el desempeño de robots móviles en entornos exteriores y geoméricamente complejos. Esta herramienta ha demostrado ser sumamente útil para el análisis en profundidad de las prestaciones del algoritmo de exploración propuesto y su puesta a punto.

En el Capítulo 6 se muestran los resultados obtenidos en las diversas etapas que forman parte de esta investigación, tanto en condiciones de simulación como en condiciones reales de funcionamiento. En este capítulo se describen resultados obtenidos en cinco diferentes escenarios, tres de ellos simulados y dos de naturaleza real. Estos cinco escenarios representan diferentes casos de aplicación de la metodología propuesta.

Finalmente, en el Capítulo 7 se presentan las conclusiones alcanzadas y se proponen algunas líneas futuras de investigación.

Capítulo 2.

MARCO CONCEPTUAL

2.1. Introducción

El objetivo de la robótica móvil apunta a independizar al máximo al robot del operador, de forma que este desempeñe tan solo la función de supervisar el normal comportamiento del robot y en caso necesario, a efectuar los ajustes o correcciones imprescindibles dando un mínimo de instrucciones. Esto supone que el robot debe ser diseñado de forma que cuente con las capacidades y destrezas necesarias para identificar, por sí solo, el entorno en el cual va a trabajar y tomar las decisiones oportunas para que su desempeño responda los criterios de calidad y eficacia esperados. En la práctica, la consecución de este objetivo requiere la presencia y conjunción de una serie de factores para lograr el funcionamiento del sistema en la forma deseada.

Un robot móvil autónomo se caracteriza por una conexión inteligente entre las operaciones de percepción y acción. El grado de autonomía depende en gran medida de la facultad del robot para abstraer el entorno y convertir la información obtenida en órdenes, de tal modo que, aplicadas sobre los actuadores de su sistema de locomoción, garanticen la realización eficaz de la tarea encomendada. La capacidad de percepción y la de decisión son definitivas a la hora de diferenciar a los robots móviles de cualquier otro vehículo o dispositivo móvil con el que pudiera ser comparado.

Durante la evolución de la búsqueda del objetivo de alcanzar la mayor independencia posible en la operación de los robots, se fueron presentando dificultades que, paralelamente, encontraron su propia solución. Así, los primeros robots móviles, requerían para su adecuado desempeño conocer totalmente el entorno donde iban a trabajar, es decir, las dimensiones del entorno y la posición y forma exacta de cada obstáculo; además estaban diseñados para seguir únicamente una serie de tareas preprogramadas.

A medida que fueron avanzando las investigaciones en robótica, se estableció como regla general que para que un robot móvil pueda navegar de forma exitosa debe ser capaz de responder a tres preguntas: **¿Dónde estoy?**, **¿Hacia dónde voy?** y **¿Cómo puedo llegar allí?** (Borenstein, Everet, & Liqiang, 1996). De estas tres preguntas la más importante, o al menos la que ha suscitado el mayor interés de los investigadores, es la primera. La respuesta adecuada a la tercera pregunta ha sido también ampliamente analizada y en consecuencia, los algoritmos de planificación de trayectorias y control reactivo han alcanzado un nivel de madurez elevado. Finalmente, la segunda pregunta es la que ha merecido una menor atención por parte de la comunidad investigadora, dejándose generalmente al arbitrio del operador o de la aplicación específica la elaboración de la respuesta adecuada. No obstante, a pesar de ser la menos estudiada, es precisamente la búsqueda de una respuesta a este segundo interrogante la que proporciona el marco teórico y experimental a la exploración robótica para dar una respuesta satisfactoria al conjunto de los tres interrogantes.

La investigación en robótica móvil se suele dividir en varios campos que confluyen en la búsqueda de un objetivo común: el desarrollo de un sistema autónomo. En la Figura 2-1 se puede observar de una manera gráfica la estrecha interrelación que existe entre los distintos campos de investigación en robótica y hasta qué punto el desarrollo de todos y cada uno de ellos es necesario para dotar de autonomía a un robot.



Figura 2-1: Tareas que debe realizar un robot para la navegación autónoma (Makarenko, Williams, Bourgault, & Durrant-Whyte, 2002)

Esta diversificación se fundamenta en la necesidad de prestar atención específica al desarrollo de las tres áreas que son esenciales para la navegación de un robot y que podríamos concretar en la siguientes: mantener información sobre la posición o el estado actual en el que se encuentra, disponer de información sobre el objetivo que debe alcanzar y establecer la relación entre el estado actual y ese objetivo.

Con base en lo anterior, podemos afirmar que la navegación robótica se fundamenta en un modelo centralizado del mundo que, o bien se le proporciona, o bien aprende el robot. Ese modelo centralizado se denomina mapa.

2.1.1. Localización, Mapeado y SLAM

Para que un robot móvil pueda afrontar satisfactoriamente tareas como generar trayectorias, evitar obstáculos o monitorizar la ejecución, se requiere que sea capaz de determinar su ubicación (posición y orientación) con respecto a un sistema de referencia absoluto.

De forma general, determinar la ubicación de un robot móvil en entornos exteriores equivale a encontrar las coordenadas de posición (x , y , z) y de rotación (Θ_x , Θ_y , Θ_z) del sistema de coordenadas del robot con respecto a un sistema absoluto. Sin embargo, la localización por sí sola no es suficiente, también se necesita contar con una representación del entorno sobre la cual ubicar el sistema.

Uno de los primeros trabajos en abordar la representación del entorno para robots móviles fue el de (Chatila & Laumond, 1985). En él se diferencian tres niveles posibles de representación del entorno: métricas, topológicas y semánticas. Un entorno idealmente representado debería contar con los tres niveles de representación. Sin embargo, por distintos motivos, la mayoría de propuestas trabajan solo con un nivel.

Generalmente las representaciones métricas son las más usadas para localizar un robot móvil dentro de un entorno, debido principalmente a su capacidad de representar un terreno con gran exactitud. Otros factores que las hacen prioritariamente elegibles consisten en que su construcción es mucho más fácil y permiten una navegación robusta del robot. Los otros dos niveles se suelen calcular a partir del nivel métrico y se usan principalmente para tareas de planificación.

Las representaciones métricas contienen un modelo geométrico en el que se representan las coordenadas y las propiedades de los objetos del entorno. Existen dos formas de realizar esta representación: una consiste en almacenar todos los objetos del entorno con sus propiedades, mientras que la otra representa la ocupación del espacio disponible dividiéndolo y marcando cada división como ocupada o no.

Esta última forma de representar el mapa es conocida como **mapas de ocupación** (Moravec & Elfes, 1985) (Ulrich & Borenstein, 1998) y es el tipo de representación más utilizada para la reconstrucción del entorno ya que ofrece múltiples ventajas. Entre ellas destaca, por ejemplo, la posibilidad de considerar toda la información sensorial sin segmentar de forma que es posible mantener una función de probabilidad de ocupación del entorno. Otra de las ventajas estriba en el hecho de que al no descartar ninguna información sensorial, surge la posibilidad de generar un mapa de estas características incluso cuando se utilizan sensores muy ruidosos

como los ultrasonidos o cuando el mapeado se realiza en entornos con alta presencia de formas irregulares.

La localización sobre los mapas de ocupación generalmente se enfrenta como un problema de seguimiento de la posición. En estos casos, una hipótesis de la posición del sistema es actualizada secuencialmente a medida que se va recogiendo información desde los sensores. De esta manera, se utiliza un modelo probabilístico tanto para la posición del robot como para la ocupación de las celdas en la etapa de construcción del mapa. Estos modelos probabilísticos constituyen hoy en día la técnica dominante a la hora de representar el entorno (Thrun, Gutmann, Fox, Burgard, & Kuipers, 1998).

A pesar de todas las consideraciones realizadas anteriormente, se puede decir, sin temor a equivocarse, que el mayor problema al que se enfrenta un robot es la construcción y actualización del mapa del entorno ya que si la estimación de la posición no es correcta, la información recogida del entorno en este punto estará mal referenciada con respecto a su posición real en el mundo y como consecuencia, las conclusiones a las que pueda llegarse a partir de esta información serán también incorrectas, al menos parcialmente. El sistema de percepción de un robot debe permitirle realizar tres tareas fundamentales: estimar su posición y orientación, mantener actualizado el mapa del entorno y detectar los posibles obstáculos.

Para llevar a cabo estas operaciones el robot debe valerse de una serie de sensores que pueden clasificarse en dos grupos: los **sensores de estado interno**, los cuales miden las variables inherentes a la estructura mecánica del robot como la velocidad o la posición odométrica; y los **sensores del entorno** que recogen toda la información de lo que rodea al robot para determinar la situación del mismo en función del entorno externo.

Estos dos tipos de sensores deben usarse porque la estimación de la posición del robot no es una tarea fácil. La información provista por los sistemas odométricos está expuesta a pequeños errores acumulativos que, con el transcurso del tiempo, generan grandes fallos en el cálculo de la posición estimada. Para prevenir estos problemas, deben usarse sistemas perceptivos del entorno que permitan efectuar la corrección de los errores estableciendo de forma efectiva las variaciones efectuadas en la situación del robot dentro del área de desplazamiento, con respecto a un instante anterior.

Este problema, conocido como SLAM, ha recibido una considerable atención por parte de numerosos investigadores (Smith, Self and Cheeseman 1988), (Durrant-Whyte, Majumder, Thrun, de Battista, & Scheduling, 2003), (Castellanos, Neira, & Tardós, 2001), (Newman, 2000), (Nüchter, Lingemann, Hertzberg, & Surmann, 2007). Un buen resumen de las contribuciones de varios autores puede encontrarse en (Thrun, 2002), proponiéndose una gran variedad de técnicas para la corrección de los fallos en la estimación. Estas técnicas varían significativamente en función del tipo de entorno en el cual el robot móvil ha de navegar, del conocimiento que se tenga de dicho entorno, de la tarea a realizar y del tipo de sensor con el que el robot va equipado.

El objetivo del SLAM es usar el entorno del robot para actualizar la posición de este. Teniendo en cuenta que la odometría de los robots no suele ser acertada, es imposible confiar

completamente en ella. Así pues, es necesario contar con la información del entorno para corregir los errores que puedan presentarse. Esta actualización de la información se logra extrayendo características del terreno y comparando las posiciones de estas a medida que el robot se va moviendo, de forma que a través de un proceso probabilístico es posible deducir dónde está el robot. En la Figura 2-2 se puede observar un proceso típico de SLAM.

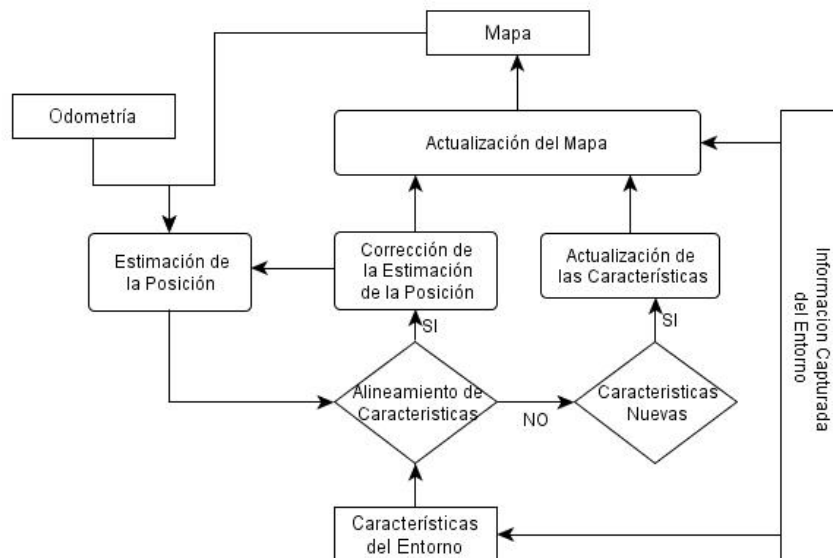


Figura 2-2: El proceso del SLAM

Los esfuerzos realizados en el ámbito del SLAM con el objetivo de percibir y modelar el entorno del robot pueden ser clasificados en dos categorías: estimación (en el sentido de predicción y corrección) y percepción. La función de los algoritmos de percepción, es el manejo de las características que van a servir para la localización del robot, y están clasificados en dos grupos según la tarea que realiza cada algoritmo que son: extracción y definición de características y asociación de datos.

Las características son marcas que pueden ser fácilmente observadas e identificadas en el entorno y son usadas por el robot para localizarse a sí mismo. El tipo de características que utiliza un robot depende del tipo de entorno en el cual este se desenvuelve. Las características deben ser estacionarias y re-observables, de forma que puedan ser vistas desde distintas posiciones y ángulos. Igualmente deben ser diferenciables unas de otras, bien sea por que tienen propiedades diferentes o porque están suficientemente alejadas entre sí. Tiene que haber suficientes características en el entorno, para que el robot no navegue por este durante mucho tiempo sin encontrar una característica que le permita localizarse a sí mismo.

Una vez se ha escogido el tipo de características que se deben utilizar, es necesario extraerlas a partir de las entradas de los sensores del robot. Hay múltiples métodos para realizar este proceso y su elección depende del tipo de características que se pueden identificar y del tipo de sensores usados.

Tras extraer las características del entorno, es necesario pasarlas por un proceso de asociación de datos. Esta tarea consiste en la identificación de las características del terreno reconociendo las detectadas con anterioridad y asociándolas con la información actual, así

como definir como nuevas aquellas características no detectadas previamente. Esta tarea es fundamental, ya que una asociación errónea de una característica puede arruinar un mapa por completo al superponer dos posiciones diferentes. Obviamente la dificultad a la hora de asociar estos datos se deriva de varios factores, tales como las características, los sensores y el entorno. Al encontrar y asociar las diferentes características es posible estimar la posición del robot a partir de esta información y la odometría del sistema, lo que se conoce como SLAM.

La evolución de los algoritmos de SLAM puede ser dividida en tres fases: En la **primera fase** la formulación matemática del problema estaba aún abierta y no había una estructura estándar que permitiese modelar las incertidumbres. Las primeras aproximaciones a la construcción de mapas se basaban en sistemas de celdas en las cuales se almacenaba la evidencia de que hubiese un obstáculo en ella de acuerdo con un número de cero a uno (Moravec and Elfes 1985). Estas representaciones son útiles para integrar la información provista por los sensores Sin embargo, no permiten representar las incertidumbres en la posición del robot. Otras aproximaciones estaban basadas en características en las cuales se representaba un mapa como un gráfico de relaciones entre objetos, en el que se podían incorporar las incertidumbres en la posición del robot.

La **segunda fase** del desarrollo del SLAM fue iniciada por la publicación de (Smith, Self and Cheeseman 1988), en la cual se planteaba el SLAM como un problema de estimación probabilística y se proponía una representación en la que se representan las posiciones de las características y las del robot en un vector de estado combinado con una matriz de covarianza. Esta representación es conocida como mapa estocástico, y es básicamente un filtro probabilístico fundamentado en el teorema de Bayes.

Todos los algoritmos de este tipo que han alcanzado cierta relevancia representan la posición del robot y el entorno probabilísticamente. Se puede asumir que en todo sistema robótico las medidas del entorno (z_t) y las de odometría (u_t), llegan secuencialmente de forma que el estado del robot x_t en el instante t dependerá de la posición del robot S_t y las mediciones del entorno m_t expresado como una función de probabilidad, se obtiene utilizando la ecuación 2.1:

$$p(x_t|z^t, u^t) = \eta p(z_t|s_t, m_t, z^{t-1}, u^t) p(s_t, m_t|z^{t-1}, u^t) \quad 2.1$$

donde η es el factor de normalización de la función de probabilidad, z^t y u^t son las lecturas acumuladas hasta el instante t . Utilizando la hipótesis de Markov y considerando que el movimiento del robot no depende del mapa, se puede establecer que:

$$p(x_t, m|z^t, u^t) = \eta p(z_t|s_t, m_t) \int p(s_t|s_{t-1}, m_t) p(s_{t-1}, m_{t-1}|z^{t-1}, u^{t-1}) ds_{t-1} \quad 2.2$$

La ecuación 2.2 es conocida como el filtro de Bayes para SLAM (Thrun, 2002). Resolver esta expresión requiere considerar los modelos de medida ($p(z_t|s_t, m_t)$) y movimiento ($p(s_t|u_t, s_{t-1})$) que sean invariantes en el tiempo, por lo cual es necesario hacer simplificaciones, suposiciones o hipótesis añadidas para su solución.

Aunque esta formulación ha sido ampliamente utilizada y ampliada por muchos autores, presenta un grave problema que ha limitado su adopción generalizada y consiste en la gran cantidad de procesamiento que requiere la actualización de la matriz de covarianzas. En la práctica, esto significa que esta metodología es recomendable únicamente cuando se trata de mapeado de entornos pequeños.

La presencia de esta limitación provocó un aumento de interés en la búsqueda de procedimientos más eficientes de SLAM, originando así la tercera fase en el desarrollo de los algoritmos de percepción, centrada en la búsqueda de nuevos y más eficientes algoritmos.

En esta **tercera fase**, una de las soluciones más difundidas y aplicadas es la basada en el Filtro de Kalman que plantea la metodología para la construcción de mapas de manera incremental mediante el uso del EKF¹ (Mountarlier & Chatila, 1989). En esta aproximación se supone que el estado del robot puede ser modelado por una distribución gaussiana del ruido de los sensores caracterizados por matrices de covarianza y de valor esperado las cuales deben ser linealizadas.

El SLAM-EKF tiene como ventaja su capacidad para mantener la estimación del estado del robot de forma incremental. Esta posibilidad facilita que se pueda cerrar el lazo correctamente y de forma adicional, lo convierte en un proceso lo suficientemente eficiente como para ser utilizado con buenos resultados en entornos de tamaño medio (aproximadamente 500 características extraídas del terreno). Los métodos de SLAM-EKF que alcanzan mayor eficiencia en requerimientos de tiempo pueden ser utilizados en entornos de hasta 1000 características, pero en estos casos es posible que, bajo ciertas circunstancias, se presenten limitaciones o disminución en la calidad del mapa.

La mayoría de estas propuestas de SLAM-EKF se basan en el hecho de que los sensores tienen campo de cobertura limitado. Esta limitación produce como consecuencia que sea cual sea el punto del entorno donde se sitúe el robot solo podrá observar una parte de las características y no la totalidad de las mismas. La aceptación de este postulado vuelve innecesaria la actualización permanente de valores para todas las características, y por lo tanto disminuye el tiempo de procesamiento del algoritmo.

En el caso del SLAM en dos dimensiones, en el cual se han centrado la mayoría de los esfuerzos, se puede afirmar que el problema de la construcción de mapas ha sido prácticamente resuelto cuando se utilizan sensores de proximidad bidimensionales (escáners láser o sonares) y mapas de dimensiones pequeñas o medianas. Aplicando modelos probabilísticos basados en el Filtro Extendido de Kalman (EKF).

Considerando la limitación en tamaño de estos métodos, muchos artículos se han enfocado en el problema de mejorar la eficiencia a la hora de estimar mapas grandes, lo cual es muy importante a la hora de cerrar grandes lazos. Sin embargo, a pesar de estos esfuerzos aún no existe un único método estándar que permita el manejo de esta cantidad de información en cualquier circunstancia, como sí lo hay en el caso en el que se manejan mapas más pequeños. En el campo de la percepción también se están haciendo grandes avances en SLAM y se puede

¹ Las siglas EKF se refieren al Filtro Extendido de Kalman

encontrar bastante literatura que propone soluciones para éste usando sistemas de visión artificial.

2.1.2. Mapeado de Exteriores

La mayoría de algoritmos de SLAM existentes trabajan con la hipótesis de un mundo bidimensional. Aunque esta hipótesis es una simplificación considerable, la configuración de los espacios interiores la hace factible en estos casos.

Por el contrario, los entornos exteriores son menos estructurados, menos controlados y menos predecibles que los entornos interiores, por lo tanto la hipótesis del mundo bidimensional es menos válida en estos casos ya que el terreno no es completamente plano y debido a la naturaleza de este tipo de entornos los obstáculos no pueden ser representados por celdas de ocupación. Por lo tanto es indispensable proponer una nueva filosofía en cuanto a navegación, planificación de trayectorias y modelado. Además se hace necesario trabajar con sistemas perceptivos que capturen información tridimensional del entorno y de la situación del robot.

Curiosamente, mientras lo anteriormente descrito sucede en los procesos de análisis de entorno y construcción de mapas, las acciones correspondientes a la determinación de la localización se han visto simplificadas en los entornos exteriores gracias al uso de sistemas GPS que proporcionan la posición (x, y, z) de cualquier entidad en cualquier lugar del mundo. Sin embargo, estos sistemas deben ser complementados con la utilización de sensores que permitan conocer la orientación del robot respecto a los ejes de referencia $(\Theta_x, \Theta_y, \Theta_z)$. Con este propósito la orientación del robot se puede calcular usando brújulas digitales y sensores de inclinación y giroscópos.

La generación de modelos tridimensionales (o mapeado en tres dimensiones) hace necesario tener en cuenta un número mucho mayor de datos y la implementación de sistemas perceptivos específicos capaces de recoger este tipo de información. Como es fácilmente comprensible, la presencia de estas nuevas necesidades conlleva, paralelamente, el incremento de las investigaciones que buscan darles adecuada respuesta.

En el caso del SLAM en tres dimensiones o SLAM 6D, un gran número de estas investigaciones se enfocan en los primeros pasos del procesamiento de información: alinear escaneos 3D, encontrar características confiables, asociación de características, filtrado de datos e imágenes entre otros.

Aquellas investigaciones sobre el SLAM 6D que han alcanzado un mayor éxito tienen en común el haber seguido un mismo proceso basado en el algoritmo ICP (Besl & McKay, 1992) (Iterative Closest Point, en Inglés) como medio principal para la localización global del sistema apoyándose en la alineación de datos para corregir la posición del robot durante el recorrido.

El proceso en referencia tiene como objeto disponer de información tridimensional relativa al terreno de la que puedan desprenderse los datos necesarios para el análisis del mismo. Hasta ahora, en los proyectos adelantados en este campo, normalmente esta información se recolecta empleando escáneres láser, montados sobre robots móviles, que recogen toda la información 3D mientras los robots están estacionados en diferentes puntos del espacio de trabajo (Wulf, Nüchter, Hertzberg, & Wagner, 2008).

Cada una de estas tomas se ubica en un mismo marco de referencia aprovechando la información sobre la posición provista por los sensores de posición (GPS, brújula, inclinómetros y odometría), corregida de acuerdo con la matriz de alineación obtenida al alinear los escaneos anteriores utilizando el algoritmo ICP.

Una vez los datos han sido capturados se utiliza un algoritmo ICP para alinear los puntos de los distintos escaneos que se encuentren a una distancia ligeramente mayor que la cobertura del láser para obtener así una nueva matriz de alineación. De igual manera, la tarea de cerrar el lazo ya no resulta tan dificultosa, pues el uso de sistemas de posicionamiento global permite estimar con mayor facilidad superposiciones entre los datos nuevos datos capturados con anterioridad, y cuya alineación permite corregir la posición y realinear los escaneos intermedios.

Como se anotaba, la generación de modelos tridimensionales hace necesario implementar más sistemas perceptivos. Las investigaciones adelantadas en el campo de la percepción han permitido desarrollar varias técnicas para adquirir datos tridimensionales con escáneres láser rotatorios instalados en robots móviles (John, y otros, 2008) (Urmsen, y otros, 2008). Igualmente, fruto de similares investigaciones, se han propuesto sistemas que permiten resolver el SLAM-6D, basados en visión artificial utilizando cámaras 3D o estereoscópicas (Paz, Piniés, Tardós, & Neira, 2008), (May, Dröschel, Holz, Fuchs, & Nüchter, 2009). Otras técnicas han tenido buena acogida y gran impacto son aquellas que utilizan para la recogida de información uno o varios láser embarcados en un vehículo el cual ha sido dotado, a su vez con otros sensores tales como GPS, inclinómetros y sistemas odométricos.

A pesar de todos estos desarrollos logrados por diferentes investigaciones, hasta el momento ninguna de ellas ha tenido como resultado un algoritmo especializado en la extracción de características que sean especialmente útiles en SLAM 6D.

Generación de Modelos 3D de exteriores

El proceso para dar lugar a un modelo digital 3D con información de color puede resumirse esquemáticamente en dos pasos: Trabajo de campo para toma de datos y Procesamiento de los datos para la creación del modelo poligonal.

La medición tridimensional de un entorno implica la necesidad de trasladarse hasta este con el equipamiento necesario para escanearlo. Además, como se ha mencionado en la introducción este equipamiento debe ser trasladado a diferentes posiciones dentro del entorno para

obtener las nubes de puntos correspondientes a las vistas parciales consideradas por cada una de esas posiciones.

Adicionalmente debe planificarse con anterioridad una estrategia a seguir para cubrir la mayor cantidad del entorno con el menor número de escaneos, garantizando un solapamiento mínimo del 10% entre las tomas realizadas en los diferentes puntos seleccionados para que el alineamiento de las tomas sea posible. Esta estrategia suele requerir el análisis previo del entorno por parte de un experto.

Una vez seleccionadas las vistas a escanear se debe proceder a la instalación del equipo de escaneo, que suele estar compuesto por una estación de escaneo láser y sus baterías, un ordenador y un sistema GPS, y al posterior escaneo desde cada uno de los puntos seleccionados. Este proceso puede llegar a tardar varias horas dependiendo del tamaño del entorno y de la cantidad de puntos de escaneo seleccionados.

Obtenidas las nubes de puntos, estas se importan y depuran en un software de tratamiento y edición de datos 3D (fundamentalmente para realizar una limpieza y filtrado de elementos que no forman parte del emplazamiento) y se procede a su alineamiento. Para esta operación se necesitan dos condiciones: Solapamiento mínimo del 10% entre escaneos adyacentes; y en que en dichos solapamientos exista al menos un cambio de forma reconocible que permita hallar puntos en común entre las nubes de puntos. Una vez cumplidas estas condiciones, generalmente se procede a alinear las nubes de puntos entre si mediante el algoritmo ICP (Iterative Closest Point) (Besl & McKay, 1992) o alguna de sus variantes (Rusinkiewicz S. , 2005).

Una vez alineadas las diferentes nubes de puntos es posible georreferenciar el modelo resultante, para lo cual es necesario hacer corresponder tres puntos del modelo con puntos GPS medidos sobre el terreno. Además, dado que los modelos poligonales suelen tener una apariencia más continua que las nubes de puntos, se genera una malla de triángulos a partir de ellas, lo que además es necesario para la extracción de las características y los contornos de los objetos 3D del entorno. Dicha malla debe ser editada para rellenar agujeros y para realizar un suavizado y una reducción de puntos del modelo.

2.1.3.Exploración

Con creciente frecuencia, los sistemas robóticos deben realizar tareas en ambientes que son parcial o totalmente desconocidos. Esta situación plantea innumerables problemas que pueden resumirse substancialmente en uno solo, a saber, la necesidad de realizar operaciones de detección en múltiples localizaciones e integrar los datos capturados en una representación única del entorno. Los robots autónomos que están diseñados para crear mapas de su entorno necesitan la capacidad de cubrirlo de forma eficaz.

Las técnicas para construcción automática de modelos del entorno han sido estudiadas extensivamente a lo largo de los últimos años con el objetivo de estimar la trayectoria y a la vez construir un mapa del entorno. A pesar de esto, la investigación se ha centrado mayoritariamente en la extracción de características relevantes a partir de los datos de los sensores (por ejemplo, bordes o esquinas), y en integración de estas características en un modelo del área de trabajo. Sin embargo, el SLAM no considera el problema de la ubicación de los sensores dentro del proceso de construcción del mapa.

Hay diversas aplicaciones en las que los robots han sido diseñados para explorar el entorno, tales como exploración planetaria o en misiones de búsqueda y rescate. Sin embargo aunque estas misiones implican que el robot navegue de forma autónoma a través de un entorno complejo, en la gran mayoría de los casos la decisión de los puntos de destino del robot son responsabilidad de un operador humano.

¿Hacia dónde debe moverse el robot? Este es un problema que sorprendentemente ha sido menos estudiado que el SLAM, a pesar de ser crucial a la hora de desplegar robots autónomos para realizar labores de mapeado en entornos desconocidos. De acuerdo con (Stachniss, 2009) la exploración robótica puede ser definida como la tarea de guiar a un vehículo durante el mapeado de forma que cubra todo el entorno con sus sensores. El guiado de un robot móvil representa la selección de acciones que este debe realizar durante el desarrollo de su misión. A su vez, el problema de la selección de acciones en el proceso de mapeado puede ser explicado como el proceso de elegir vistas que maximicen la utilidad de la nueva información sensorial para minimizar el tiempo necesario para explorar el entorno completamente. A lo largo de los años se han implementado diferentes estrategias de exploración que pueden ser clasificadas en dos grandes categorías (Basilico & Amigoni, 2008): Estrategias Predeterminadas o Trayectorias Fijas, y Estrategias Basadas en la Información.

Estrategias Predeterminadas o Trayectorias Fijas

Dentro de esta categoría se encuentran todas las estrategias que utilizan información conocida previamente sobre el entorno, para determinar la trayectoria a seguir para cubrirlo completamente.

Algunas estrategias de exploración hacen que un robot se mueva aleatoriamente de un punto a otro sin evaluar la utilidad de estos puntos. Estas estrategias no están implementadas habitualmente pero sí se utilizan como base de pruebas para evaluar la efectividad de cualquier algoritmo. Su función en estos casos es actuar como referente de “el peor de los casos” o resultado que no debería darse, y sirve para comprobar si un algoritmo es útil o no dependiendo de la mejora que supone con respecto a una estrategia de selección de acciones aleatoria.

Otros sistemas utilizan estrategias que siguen trayectorias predefinidas basadas en información previa del entorno. Así, por ejemplo en algunos sistemas el robot debe trazar círculos concéntricos hasta recorrer completamente la escena. Otro ejemplo de este caso

puede ser los algoritmos de seguimiento de pared. Otro más particular es el presentado por (Edlinger & von Puttkamer, 1994), en el cuál se presenta un algoritmo diseñado específicamente para explorar entornos de oficina. En este caso el robot debe recoger toda la información de un espacio antes de atravesar una puerta o pasillo.

Dentro de esta categoría también encajan los métodos orientados a misiones de inspección o vigilancia. Aunque en estos casos el mapa del entorno es conocido, el objetivo sigue siendo cubrir todo el entorno con los sensores del robot. Generalmente, se recurre a planificar con anterioridad la trayectoria más eficiente (Choset, 2001) para cubrir el entorno completamente. Otro ejemplo interesante de estos métodos es el presentado por (Danner & Kavraki, 2000) en el cual se replanifica una nueva trayectoria de forma aleatoria tras cada recorrido, para garantizar que un robot de vigilancia no sea predecible en sus movimientos.

Otro método interesante es el presentado por (Bourque & Dudek, 1998) cuyo método está basado en la forma como un turista recorre un museo, de manera que el robot debe detectar puntos de interés visibles desde su última posición y navegar hacia el más cercano repitiendo el proceso hasta que se hallan visitado todos los puntos de interés. Este método se incluye en esta categoría, ya que está diseñado específicamente pensando en la reconstrucción de espacios turísticos de interiores.

Este grupo de estrategias presenta varios problemas, el más importante de los cuales es la falta de adaptabilidad a entornos desconocidos en los cuales pueden dejar grandes áreas sin explorar o ser muy ineficientes. Por este motivo no es un grupo muy interesante para la investigación en mapeado de entornos, a diferencia de lo que sucede con las estrategias basadas en información que constituyen el segundo gran grupo de esta clasificación y que ha recibido mucha más atención.

Estrategias Basadas en la Información

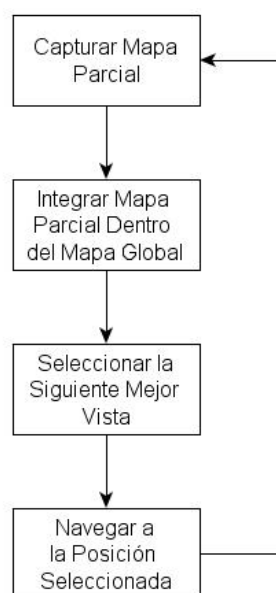


Figura 2-3: Proceso de exploración seguido por los métodos *nbv* (Basilico & Amigoni, 2008)

Se denominan así porque utilizan la información que se ha recabado del entorno para escoger la mejor posición para la siguiente toma de datos, estas estrategias también son conocidas como métodos de la siguiente mejor vista (*nbv*) y a diferencia del primer grupo, se adaptan mucho mejor a diferentes tipos de entornos. Dentro de esta categoría existen varias propuestas, pero como regla general todas ellas siguen el proceso mostrado en la Figura 2-3.

Las distintas propuestas se diferencian en las estrategias utilizadas para seleccionar la siguiente posición. En concreto, difieren en la forma de seleccionar las posiciones a evaluar y en los criterios utilizados para evaluar las posiciones candidatas.

Los **Métodos Frontera** (Yamauchi, 1997) son métodos en los se generan posiciones candidatas en las fronteras entre el área explorada y el área sin explorar. Para definir el punto de la frontera hacia el cual se debe desplazar el robot, se calcula la dificultad que este puede tener para llegar hasta cada punto de la frontera, y se escoge la trayectoria de menor costo (Yamauchi, Schultz, Adams, & Graves, 1998). Como función de costo se escoge la distancia de navegación la posición candidata.

En la Figura 2-4 se muestra un ejemplo de exploración de un entorno usando un método basado en fronteras. En cada figura las posiciones candidatas están marcadas con cruces, el robot con un círculo. En este caso se puede ver que aunque los métodos frontera garantizan la cobertura total del entorno de trabajo, no garantizan la trayectoria más eficiente; aunque sin conocimiento previo del entorno esto es muy complejo (Choset, 2001)).

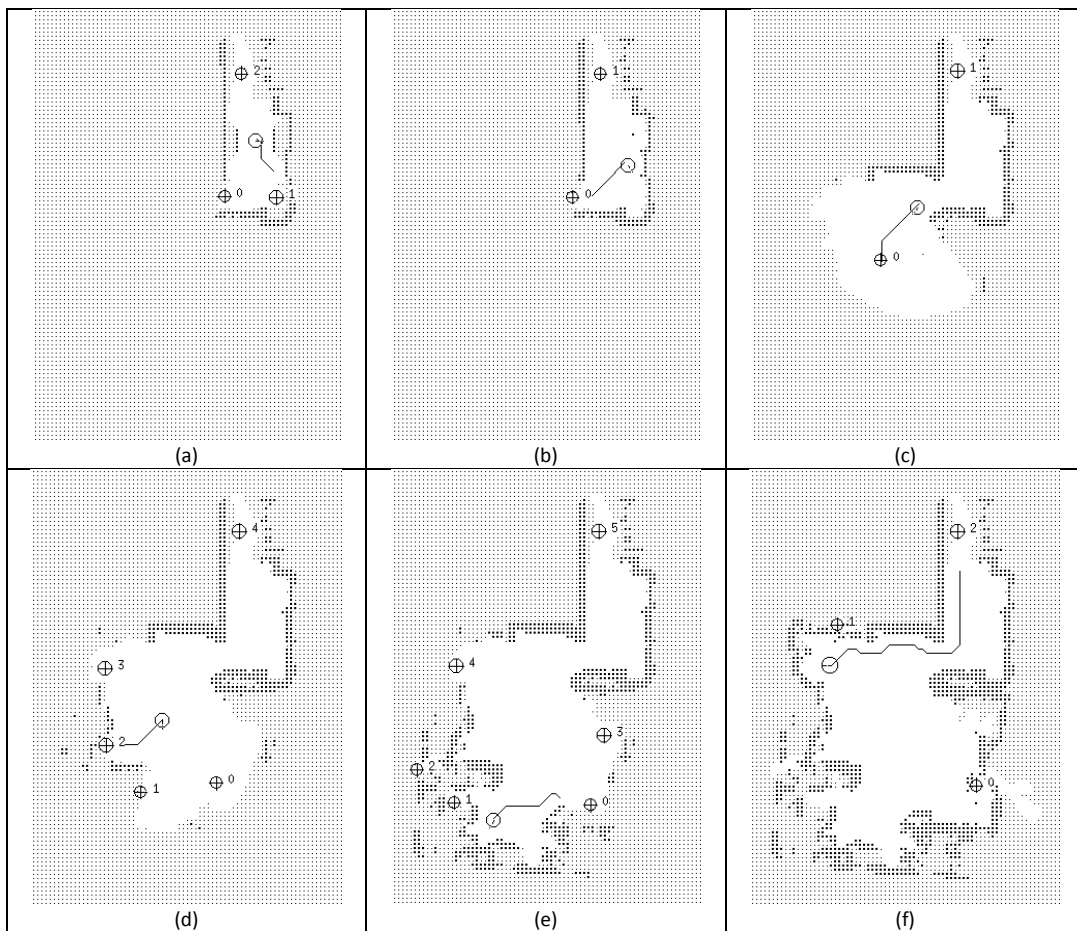


Figura 2-4: Ejemplo de exploración de un entorno de interiores usando un método frontera (Yamauchi, 1997)

Las **estrategias tipo "greedy"** (avaras, en español) (Albers & Henzinger, 1997); (Koenig, Tovey, & Halliburton, 2001), son algoritmos que buscan llevar al robot desde su posición actual hacia la posición más cercana que pueda proveer algún tipo de información. El objetivo es adquirir la información nueva disponible al menor costo posible. Para lograr este objetivo se eligen posiciones cercanas siguiendo cuatro criterios distintos (Tovey y Koenig 2003):

No visitado más cercano: En este caso el robot siempre va por el camino más corto desde su posición actual a la posición no visitada más próxima. Una vez allí repite el proceso, hasta que ha visitado todas las posiciones posibles. Este criterio se utiliza únicamente cuando se asume que el robot solo puede recoger información de su posición actual.

No escaneado más cercano: Similar al caso anterior pero aquí, el robot viaja hacia la posición que no ha escaneado más cercana a su posición actual.

No escaneado más cercano con replanificación: Este caso es igual al anterior pero se considera que el robot puede escanear una posición sin necesidad de llegar a ella, de forma que replanifica su ruta al escanear esa posición.

Información más cercana: El robot se desplazara hasta la posición más cercana que pueda aportarle información nueva.

(Tovey y Koenig 2003) demostraron que el uso de estas estrategia mantiene la distancia de viaje relativamente corta al compararla con la distancia recorrida en la trayectoria mínima necesaria para cubrir todo el entorno. El problema de estas estrategias, sin embargo, es que no consideran ni la calidad ni la cantidad de información que puede llegar a ser recogida en cada punto de escaneo.

Otras estrategias evalúan la **utilidad de un objetivo** para el robot. Por ejemplo (Gonzalez-Baños & Latombe, 2002) define la utilidad de un objetivo como una ganancia esperada de información. En este objetivo se combina la ganancia esperada de información con la distancia a recorrer, de forma que la siguiente vista maximice la información recogida y reduzca el recorrido realizado por el robot. Ello se logra analizando varias ubicaciones seleccionadas como candidatas aleatoriamente de acuerdo con su cercanía a otros puntos de observación no visitados y la cantidad de información nueva que se calcula que puede ser extraída del nuevo objetivo.

La Figura 2-5 muestra un ejemplo de selección de **nbv** usando el método mencionado anteriormente. Como se puede observar, en este caso se generan posiciones candidatas entre la posición del robot (Figura 2-5b) y las fronteras entre el espacio conocido y no conocido. Cada posición candidata es evaluada en función del área no conocida visible desde cada candidato (Figura 2-5c) y la distancia a recorrer hasta la misma.

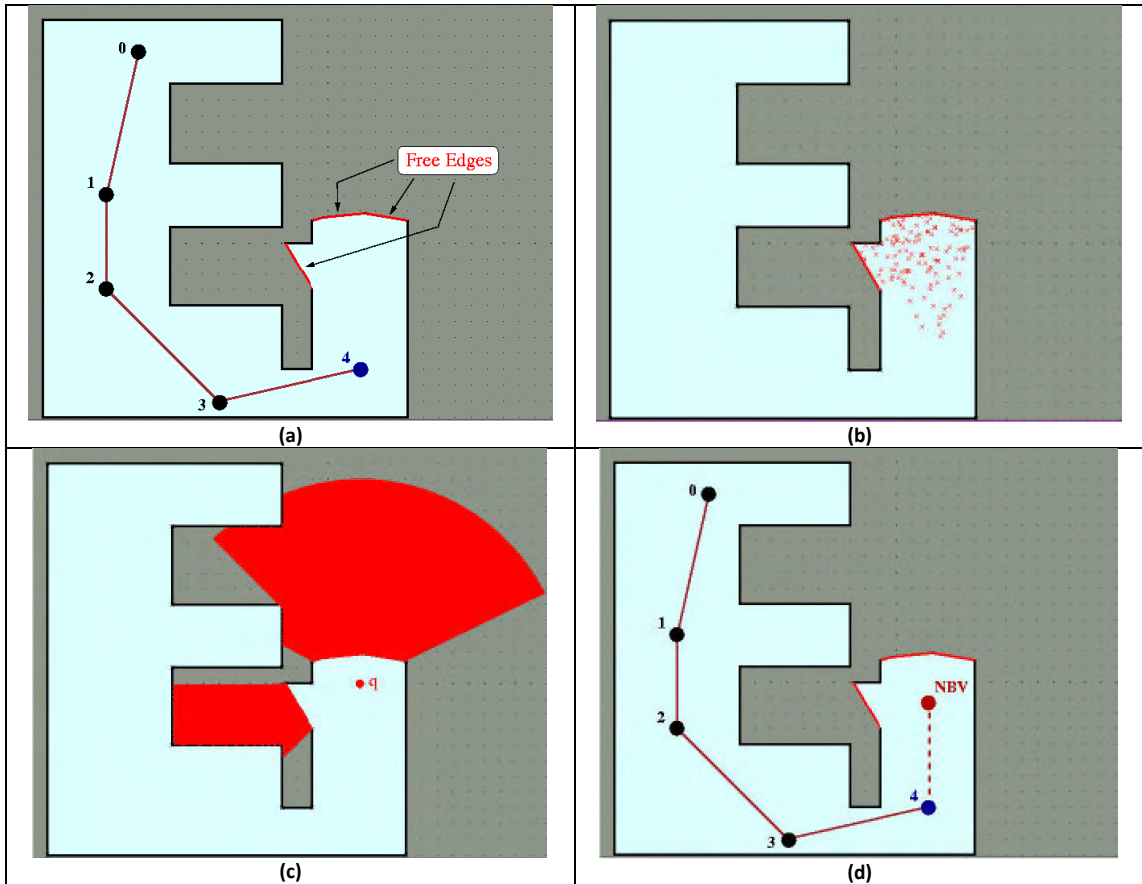


Figura 2-5: ejemplo de selección de *nbv* mediante la evaluación de la utilidad de las posiciones candidatas (Gonzalez-Baños & Latombe, 2002)

Otras estrategias usan características relevantes para el SLAM como criterio de evaluación de la utilidad de cada posición candidata. Por ejemplo, (Grabowski, Khosla, & Choset, 2003) marcan las fronteras y las zonas donde se pueden ver las características utilizadas para el SLAM como **zonas de interés**, de forma que las posiciones candidatas desde las que se ven estas zonas con un ángulo diferente obtienen unas puntuaciones más altas, lo cual mejora los resultados del proceso de mapeado.

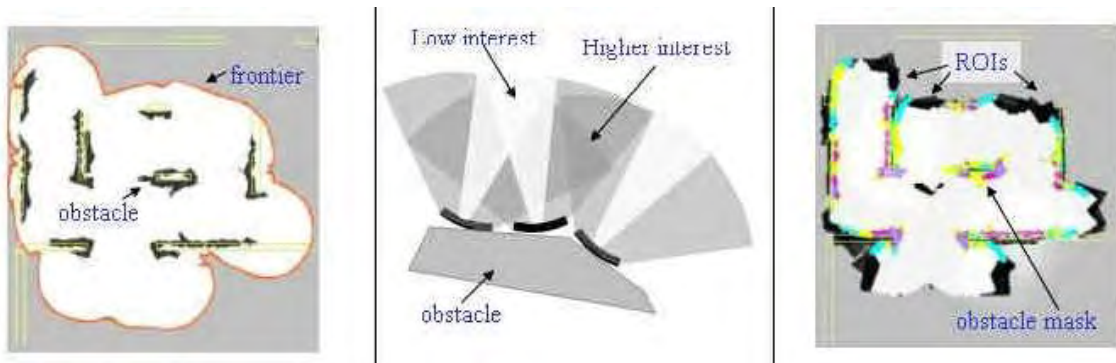


Figura 2-6: Zonas de interés extraídas de un entorno (Grabowski, Khosla, & Choset, 2003)

En (Newman, Bosse, & Leonard, 2003) se utiliza más de un criterio para evaluar las posiciones candidatas, de forma que además de la información nueva que se espera recoger en cada punto, también se utiliza como criterio la cantidad de características visibles, las regiones con

múltiples características y la fiabilidad de cada característica para la localización del robot. La posición elegida es la que cumpla más adecuadamente con estos requerimientos.

Otros autores también han incluido criterios diferentes a los relativos a la ganancia de información y al mapeado. Por ejemplo, (Visser & Slamet, 2008) incluyen como criterio para la exploración la probabilidad de mantener comunicación con una estación de control en su próximo punto de captura. Siguiendo esta idea (Basilico & Amigoni, 2009) han presentado una metodología para escoger puntos de vista de acuerdo con múltiples criterios correlativos.

La mayoría de los trabajos se basan en representaciones de celdillas de ocupación o topológicas. Sin embargo, algunos autores como (Stachniss & Burgard, 2003) han propuesto el uso de una representación llamada mapas de cobertura que es una ampliación de los mapas de celdas de ocupación en las que se incluye un valor de cobertura para cada celda del mapa. La exploración en este tipo de representación es una función que optimiza el valor de la cobertura en el mapa con respecto a la distancia recorrida.

SLAM activo

Las denominadas **técnicas de SLAM activo** son otra solución propuesta para el mapeado durante la exploración. El objetivo principal es el cálculo de trayectorias que mejoren la localización del sistema y por ende el mapa del entorno.

Las técnicas SLAM activo no pueden ser consideradas como técnicas de *nbv* ya que en estos casos se asume que existe una incertidumbre en la posición, por lo que no se conoce tampoco la posición real de la siguiente mejor vista. Para abordar el problema (Murphy, 2000) y (Doucet, de Freitas, Murphy, & Russell, 2000) propusieron el uso de filtros de partículas tipo Rao-Blackwell para crear una estimación de la trayectoria realizada por el robot, la cual a su vez es utilizada para crear una predicción del mapa del entorno. Esta predicción facilita la detección de posiciones desde las cuales se pueda cerrar el lazo del mapa de manera que el mapa es más fiable.

Sin embargo, el uso de este tipo de filtros no aborda el problema de la selección de acciones. En estos casos, ya que la posición es incierta, no se puede planificar una trayectoria hacia un punto específico del mapa. Por este motivo se debe escoger una estrategia de selección de acciones durante el proceso de mapeado que garanticen una fiabilidad mínima del mapa mientras cubren nuevas zonas dentro del mapa. Estas técnicas son conocidas como SLAM activo porque la selección de acciones se hace durante el proceso de navegación.

Autores como (Makarenko, Williams, Bourgault, & Durrant-Whyte, 2002) y (Bourgault, Makarenko, Williams, Grocholsky, & Durrant-Whyte, 2002) evalúan las acciones a realizar en términos de utilidad de la nueva información y disminución de la incertidumbre de la posición usando funciones de evaluación similares a las utilizadas en los métodos de *nbv*. Igualmente (Guez & Pineau, 2010) proponen evaluar estas mismas condiciones además de algunos criterios que dependen directamente de la misión a realizar por el robot por medio de un árbol de decisión markoviano (POMDP).

(Sim & Dudek, 2004) Proponen una solución completamente diferente. En este caso el robot debe realizar una trayectoria en espiral pero controlando que la incertidumbre en la posición no rebase cierto nivel, de forma que el robot seguirá siempre trayectorias que aumenten la información disponible sobre el entorno sin perder calidad en el posicionamiento.

Sin embargo, estas estrategias requieren de la capacidad de identificar características dentro del mapa para garantizar un nivel mínimo de incertidumbre. Por este motivo (Blanco, Fernández-Madrigal, & Gonzalez, 2008) y (Stachniss, 2009) proponen una metodología basada en la detección de oportunidades para cerrar el lazo, de forma que el robot debe tener dos comportamientos diferentes. El primero es el de aumento de la información del mapa en la cual el robot explora zonas desconocidas del entorno, pero evalúa constantemente la utilidad de reentrar en zonas previamente mapeadas. Cuando se detecta una posibilidad de cerrar el lazo, el robot cambia de comportamiento y se dirige hacia el punto donde esto puede suceder creando así mapas más consistentes.

Los métodos presentados hasta ahora están orientados a la construcción de mapas en 2D a diferencia de la metodología propuesta en este documento, la cual está orientada a la generación de modelos tridimensionales de entornos exteriores. En la siguiente sección se presentan algunos trabajos que están orientados a este propósito. Algunos de estos trabajos abordan la exploración, pero sin embargo, ninguno ha desarrollado una metodología específica para este propósito.

2.2. Trabajos Relacionados

En la actualidad es posible encontrar un número significativo de trabajos que guardan algún tipo de relación con el objeto de la presente investigación, pues están orientados bien sea a la creación de modelos tridimensionales de grandes entornos exteriores o a la exploración de este tipo de entornos con objetivos diferentes a la reconstrucción.

Muchos de estos trabajos están orientados a la generación de modelos a gran escala de ciudades. En este grupo podemos hacer mención de los denominados “3D city” (Früh & Zakhor, 2003), “City Scanning Project” (Teller, 1997), “4D Cities” (Dellaert, 2005) o Google Street View (Williams, 2010) que son ampliamente conocidos. Sin embargo, en estos casos, las plataformas de captura de datos son equipos de medición conducidos por humanos que simplemente recogen la información tridimensional del entorno, la cual es procesada posteriormente, como se puede ver en la Figura 2-7.

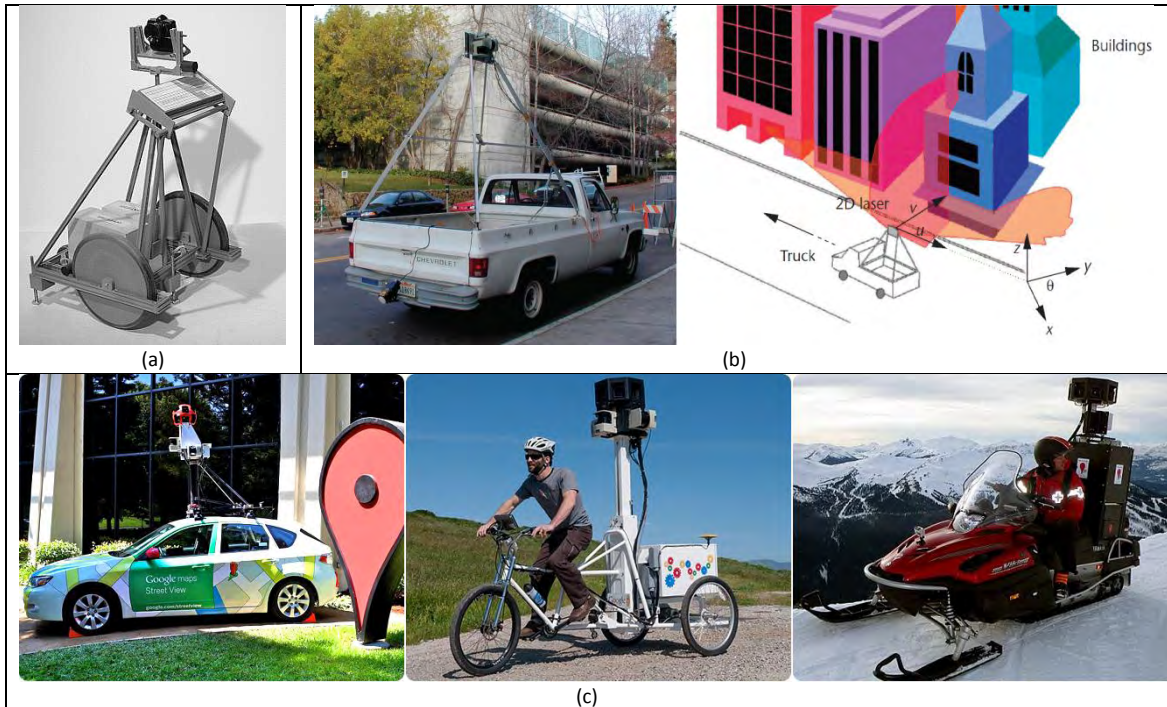


Figura 2-7: (a) Plataforma usada en “City Scanning Project” (Teller, 1997); (b) Sistema de reconstrucción de “3D city” (Früh & Zakhor, 2003); (c) Vehículos utilizados por “Google Street View” (Williams, 2010).

Algunos de las investigaciones coincidentes con el presente trabajo se orientan principalmente a dar solución a los problemas planteados por la navegación, reconstrucción y exploración de entornos exteriores. Uno de los primeros que se enfrentó a estos problemas es el descrito por (Thrun, Burgard and Fox 2000), cuyo objetivo es la consecución de un algoritmo que permitiese obtener un mapa tridimensional del entorno en tiempo real. Esta investigación proponía un modelo incremental para resolver el problema de la localización que, a pesar de ser incapaz de manejar entornos cíclicos y de no poder corregir las posiciones obtenidas con anterioridad, es bastante popular debido a su simplicidad.

En este caso se diseñaron diferentes robots móviles (ver Figura 2-8) equipados con dos escáners láser bidimensionales situados de modo perpendicular, de manera tal que uno pudiera recoger los datos de forma horizontal, mientras que el otro se encargaba de recoger los datos tridimensionales. El escáner que recoge la información en forma horizontal es el que sirve para la navegación del robot.



Figura 2-8: Robots utilizados en (Thrun, Burgard and Fox 2000) para la adquisición de datos; a) Robot urbano para la exploración de exteriores e interiores. b) Robot tipo pioneer utilizado para mapeado 3D.

El objetivo de este trabajo es la consecución de un algoritmo que permita la obtención de mapas tridimensionales con robots móviles en tiempo real. Con este propósito se implementó

un algoritmo que permitía la alineación rápida de los diferentes escaneos. Al combinar este algoritmo con un método de localización basado en probabilidades de localización a partir de la información del entorno y de la odometría, se hacía posible realizar la localización y el mapeado en robots equipados con escáners láser obteniendo mejores resultados. Como resultado de ese proyecto se ha desarrollado, igualmente, un algoritmo que permite reconstruir entornos sin información odométrica.

Este algoritmo tiene además la particularidad de que puede ser usado con robots diseñados para moverse por entornos exteriores. La importancia de este trabajo radica en dos aspectos. En primer lugar, fue uno de los primeros trabajos que abordó el uso de robots móviles para la reconstrucción de modelos tridimensionales y la experiencia recopilada durante en esta investigación permitió llegar a la conclusión de que al igual que en el mapeado en 2D en los datos 3D también existen inconsistencias que deben ser resueltas mediante proceso de alineamiento de los datos que considere todos los datos capturados y no solo los escaneos sucesivos.



Figura 2-9: Robots utilizados por (Hähnel, Burgard, & Thrun, 2003). Interiores (izq) y Exteriores (der).

Posteriormente (Hähnel, Burgard, & Thrun, 2003), presentaron un método para la reconstrucción de entornos exteriores en interiores utilizando robots móviles (ver Figura 2-9), basado en la metodología expuesta anteriormente. La característica principal de su propuesta la constituye la fundamentación del método, el cual se basa en un algoritmo que inicialmente estima la posición del robot utilizando algoritmos de alineamiento de datos 2D y 3D. Posteriormente la información tridimensional obtenida se simplificaba para facilitar la identificación de planos.

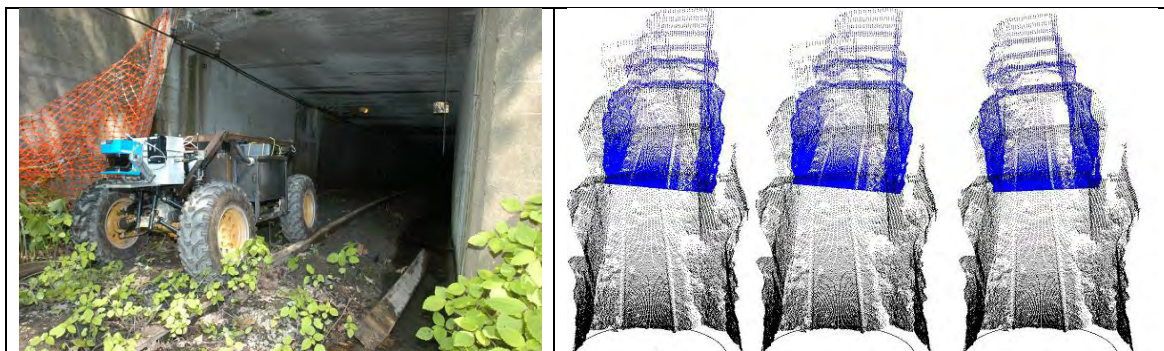


Figura 2-10: Robot utilizado para el escaneo de minas (izquierda); Datos capturados y a la derecha tras algunas alineaciones el resultado final (derecha). (Nüchter, Surmann, Lingemann, Hertzberg, & Thrun, 2004)

Una ampliación de este algoritmo para el mapeado de minas ha sido propuesta por (Nüchter, Surmann, Lingemann, Hertzberg, & Thrun, 2004). En este trabajo un robot especialmente diseñado (Figura 2-10) para recoger información tridimensional de las galerías de una mina va recogiendo datos que se van alineando con los datos capturados anteriormente por medio del algoritmo ICP (Besl & McKay, 1992). Al igual que en los casos anteriores el sistema no maneja lazos cerrados, pero por la naturaleza de las minas este problema no es excesivamente limitante de cara a la calidad del mapa. Además, se ha implementado un algoritmo de reducción de datos para optimizar el rendimiento del algoritmo.

Otros de los trabajos relacionados destacables es la investigación llevada a cabo por (Sequeira, Ng, Wolfart, Gonçalves, & Hogg, 1999), en la cual se plantea una aproximación para la construcción de modelos tridimensionales con texturas, a partir de datos tomados por escáneres láser y videos del entorno. Para esta propuesta se diseñó un prototipo para la reconstrucción tridimensional, llamado EST (siglas en inglés para sensor ambiental para la telepresencia), que puede estar implementado en una plataforma móvil autónoma (AEST) o en un carro que sea empujado por personas (EST), las cuales se pueden ver en la Figura 2-11 .

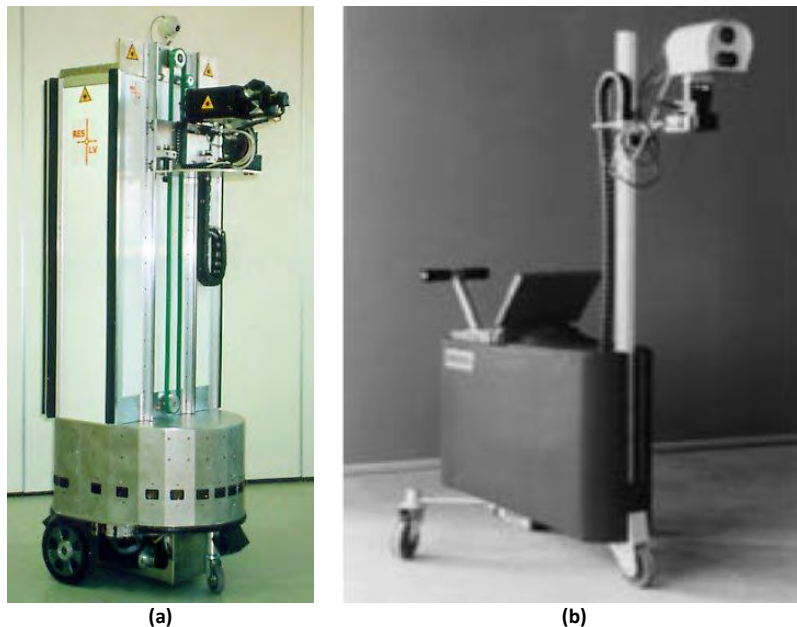


Figura 2-11: Prototipos utilizados por (Sequeira, Ng, Wolfart, Gonçalves, & Hogg, 1999). (a) AEST, (b) EST.

Sobre estas plataformas está dispuesto un escáner láser 3D RIEGL, y una cámara de video la cual se utiliza para capturar las texturas que serán pegadas sobre el modelo. Una de las características de este proyecto es su capacidad de reconstruir modelos tridimensionales al mismo tiempo que va localizando la plataforma. Para realizar ambas tareas simultáneamente, emplea un algoritmo ICP para alinear los puntos tridimensionales obtenidos en diferentes posiciones, aplicando a la vez un método de extracción de características para el pegado de texturas desde el video. Es necesario mencionar sin embargo, que el proceso de obtención del modelo 3D se hace *offline* y requiere una gran cantidad de recursos computacionales. La importancia de esta propuesta radica en el uso del algoritmo ICP para la corrección de la posición odométrica o directamente para la localización de la plataforma. Sin embargo, el uso de este algoritmo hace imposible la reconstrucción del modelo en tiempo de ejecución, este debe ser obtenido en una etapa posterior de procesamiento de datos.

Para esta misma plataforma (Klein & Sequeira, 2000) propusieron un algoritmo basado en las zonas no exploradas del entorno. En este algoritmo se separan dos áreas de trabajo (Figura 2-12), una que cubre el área conocida, y otra que representa la frontera entre las regiones conocidas y no conocidas. Posteriormente se calcula una tasa de costo/beneficio para cada punto de la región que representa la frontera y se selecciona el punto con la mejor relación costo/beneficio, pero desde el cual al menos el 20% de la información vista sea del entorno conocido. Esta es una de las primeras propuestas de exploración de entornos basadas en información tridimensional.

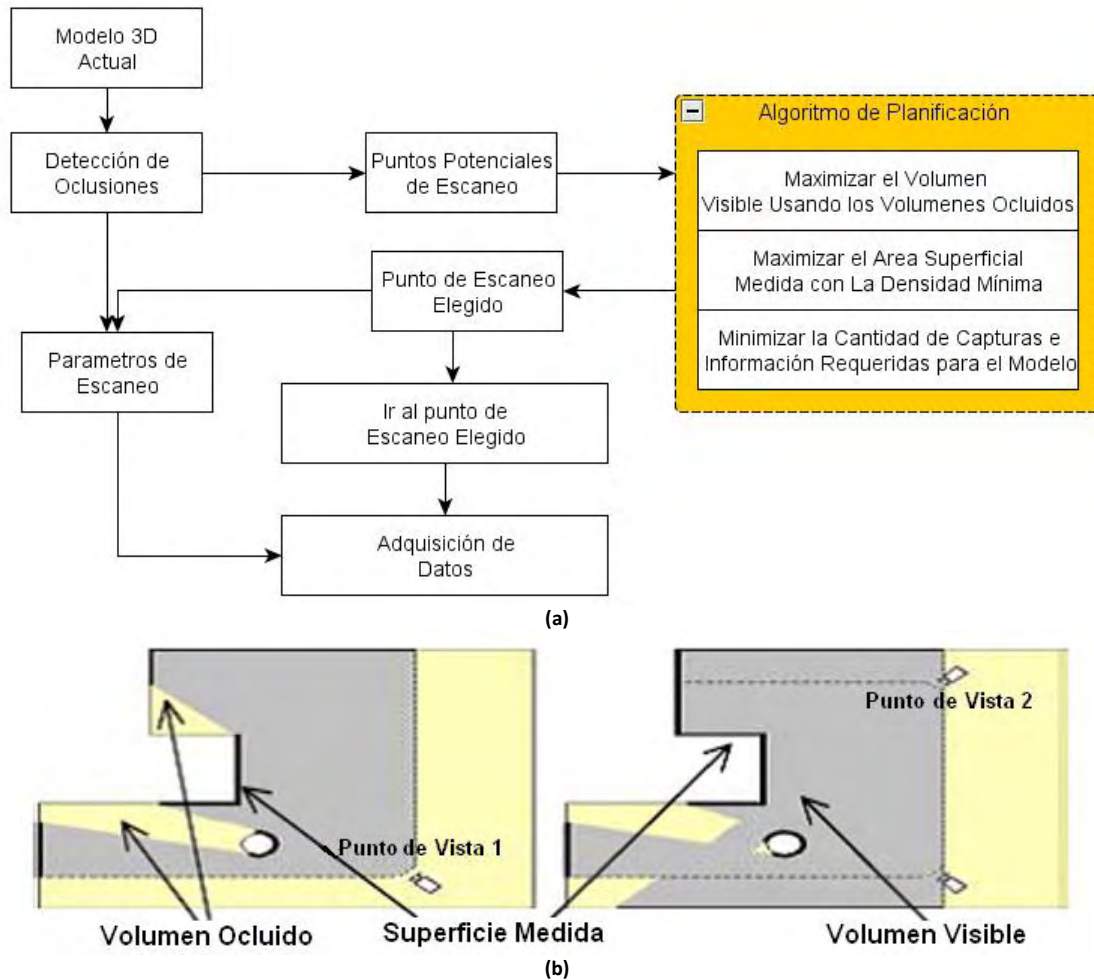


Figura 2-12: (a) Diagrama de flujo del algoritmo de planificación propuesto por (Puig, Sequeira, Ruiz, Gonçalves, & Mellado, 2005), (b) ejemplo del proceso de reconstrucción del mismo.

El mismo grupo de investigación presenta una mejora de este algoritmo (Puig, Sequeira, Ruiz, Gonçalves, & Mellado, 2005) incluyendo la densidad deseada de la nube de puntos como un parámetro de entrada al sistema. Además, también se propone un sistema para reconstrucción de entornos urbanos basado en el método de reconstrucción offline presentado por (Bostrom, Fiocco, Puig, Rossini, Goncalves, & Sequeira, 2004), utilizando un sistema de adquisición de datos dispuesto sobre un vehículo convencional (Figura 2-13).



Figura 2-13: Vehículo para reconstrucción de entornos urbanos (Puig, Sequeira, Ruiz, Gonçalves, & Mellado, 2005)

Dentro de los proyectos relacionados merece también ser destacado el “Smarter Team” (Jensen, Weingarten, Kolski, & Siegwart, 2005) y (Nüchter, Lingemann, Hertzberg, & Surmann, 2007), que está orientado a la construcción de un coche autónomo para la reconstrucción de entornos urbanos. En este caso el robot se encarga de la navegación y reconstrucción del entorno.

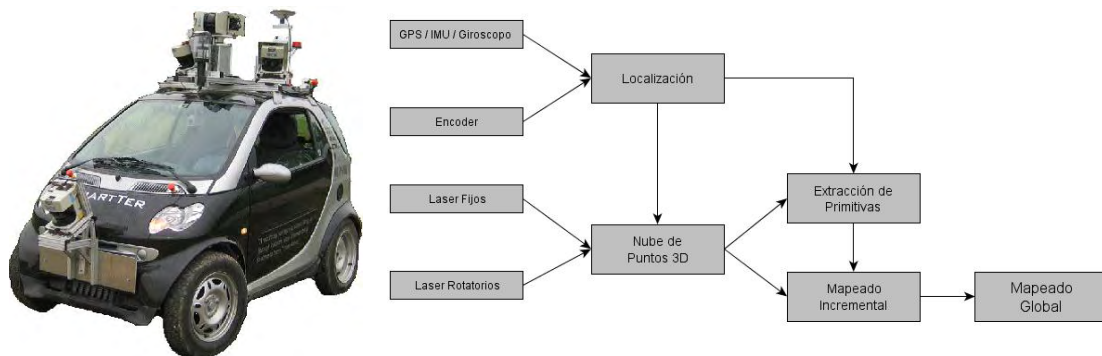


Figura 2-14: (Izq.) Robot SMARTER, (der) arquitectura software utilizada para el mapeado. (Jensen, Weingarten, Kolski, & Siegwart, 2005)

En este proyecto se ha diseñado un filtro que se encarga de recoger los datos provistos por todos los sensores e integrarlos para proveer una posición final de acuerdo a los datos y a la confiabilidad de estos. Igualmente, en su desarrollo, se implementaron algoritmos de alineamiento rápido de escaneos para cerrar el lazo una vez el robot visitaba una posición en la que había estado anteriormente, basados en la disminución del error mínimo cuadrático. Adicionalmente este proyecto propone una representación por mapas de elevación que disminuye la cantidad de información que se debe manejar. En lo concerniente a la exploración, ya que este sistema es para el mapeado de ciudades, los autores suponen que al robot se le dará un recorrido de modo que el sistema conozca previamente los puntos por los que debe pasar y navegue hacia éstos secuencialmente.

El proyecto Kurt 3D (Nüchter, Lingemann, Hertzberg, & Surmann, 2007) tiene como principal objetivo el desarrollo de un sistema muy integrado y compacto que permite la generación tridimensional de escenas tanto interiores como exteriores. Con esta finalidad se ha desarrollado una plataforma de exteriores con un escáner de tres dimensiones basado en un sistema pan-tilt y un escáner láser lineal, sobre una plataforma móvil. La finalidad de este proyecto es el desarrollo de técnicas para la localización y mapeado simultáneo (SLAM) en tres

dimensiones lo cual consiguen a través del uso de un algoritmo ICP sobre los datos capturados para corregir la posición del robot.

En lo referente a exteriores este proyecto está basado en un modelo de plataforma tipo KURT2 (Figura 2-15). Sobre la plataforma está dispuesto un sistema de reconstrucción 3D, el cual es un escáner láser bidimensional montado sobre un servo motor estándar para darle al escáner la capacidad de capturar datos en tres dimensiones. Dispone, igualmente, de dos cámaras web con un sistema de posicionamiento propio para el texturizado del modelo.

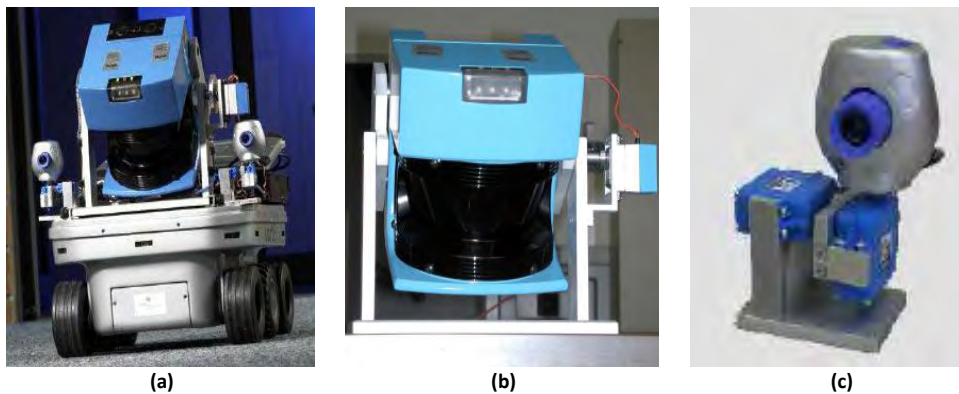


Figura 2-15: Componentes del sistema de escaneo KURT 3D, a) sistema de escaneo montado sobre plataforma Kurt2, b) Sistema de escaneo 3D AIS, c) Cámara web y sistema de posicionamiento para esta

Para abordar la exploración se ha implementado un algoritmo basado en los planos ocluidos del entorno (Nüchter, Surmann, & Hertzberg, 2003). Con este fin, a un escaneo a nivel de suelo del entorno se le añaden “líneas invisibles” que señalan los planos ocluidos. Posteriormente se generan puntos al azar en el espacio libre y se evalúa cuanto espacio entre las “líneas invisibles” se puede ver desde cada posición (Figura 2-16).

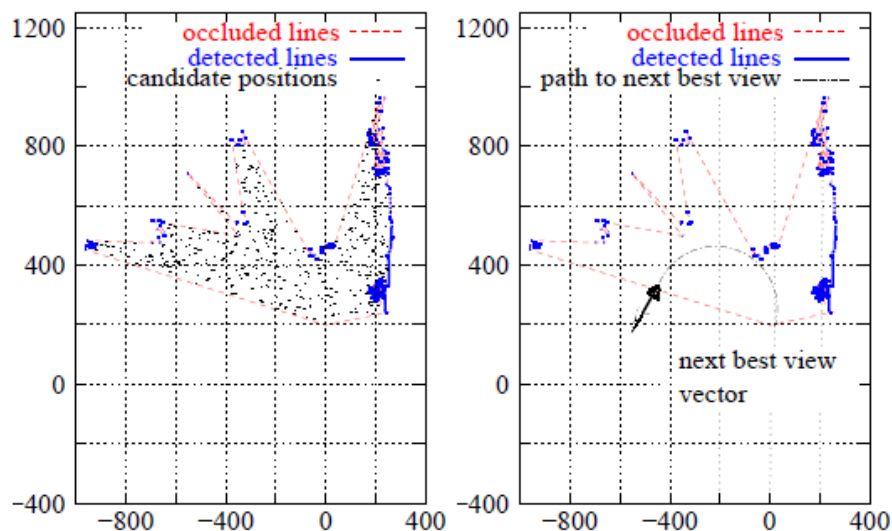


Figura 2-16: Algoritmo de exploración propuesto por (Nüchter, Surmann, & Hertzberg, 2003)

Sin embargo esta propuesta solo es válida para ambientes interiores. Con el fin de hacer posible su implementación en entornos exteriores, este proyecto ha recurrido al concepto de “SPLAM” (simultaneous planning localization and mapping) (Nüchter, Lingemann, & Hertzberg, 2006), para extraer superficies válidas para el desplazamiento del robot en entornos

desestructurados. Sin embargo, en el trabajo no se abordó ninguna metodología para evaluar el siguiente punto al que el robot deba desplazarse.

El proyecto AVENUE (Blaer & Allen, 2009) se propuso construir un sistema para la reconstrucción tridimensional de entornos urbanos. Para ello se implementó un robot comercial con un escáner láser tipo Cyrax y un sistema de localización basado en GPS diferencial. Igualmente se dispuso una cámara sobre una unidad de Pan-Tilt para el mapeado de texturas. La Figura 2-17 muestra el robot utilizado y su arquitectura software.

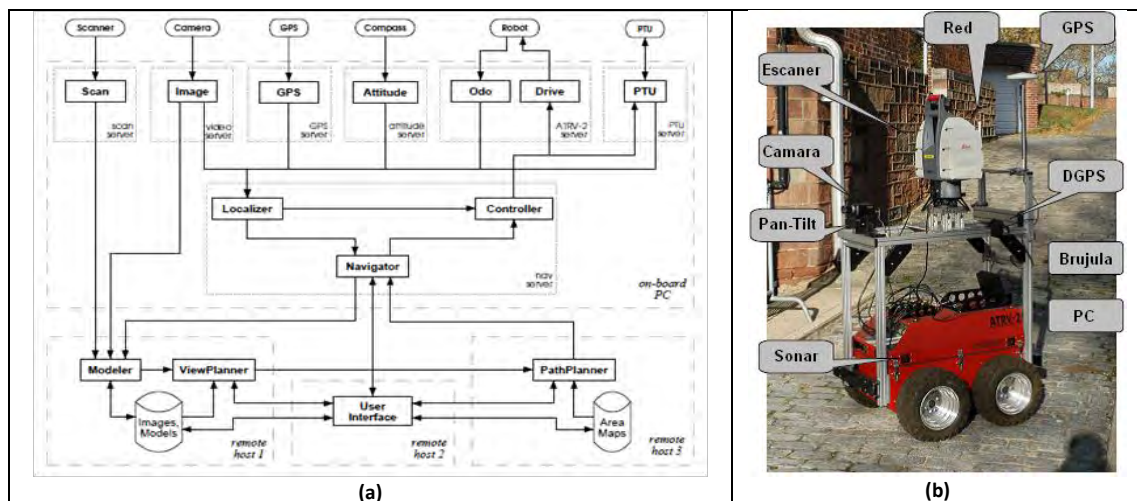


Figura 2-17: Arquitectura Software (a), y robot desarrollado (b) en el proyecto AVENUE (Blaer & Allen, 2009).

Para la generación automática de modelos se planteó un algoritmo de alineación de datos que consiste en la segmentación de las distintas nubes de puntos capturadas en el proceso, con el objetivo de buscar puntos coincidentes a lo largo de escaneos anteriores, para luego alinearlos entre sí utilizando un algoritmo ICP. Para el mapeado de texturas se extraen líneas y planos a partir de las imágenes de la cámara y se alinean con las líneas extraídas de los planos segmentados como muestra la Figura 2-18.

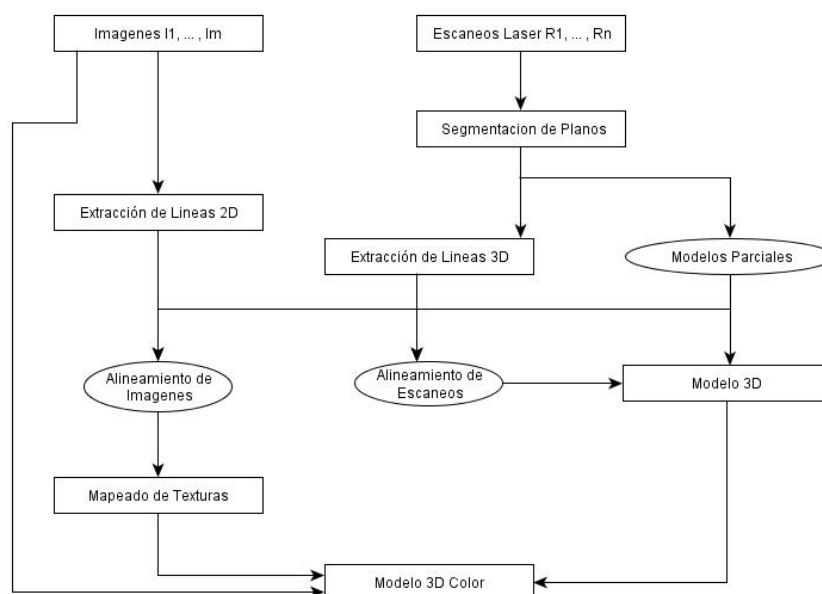


Figura 2-18: Algoritmo de alineación de escaneos propuesto en Avenue

Además en este trabajo se ha desarrollado un algoritmo de planificación de vistas basado en un mapa 2D del entorno, que genera una serie de puntos al azar sobre el espacio libre. Con estos datos se selecciona la serie de puntos de escaneo que cubra la mayor cantidad de espacio libre en menos tomas. Una vez escaneado el entorno en todos los puntos preseleccionados, se analiza el modelo 3D resultante y se extraen las zonas ocluidas del modelo de modo que se pueda replanificar una serie de puntos de vista que permita cubrir estas zonas.

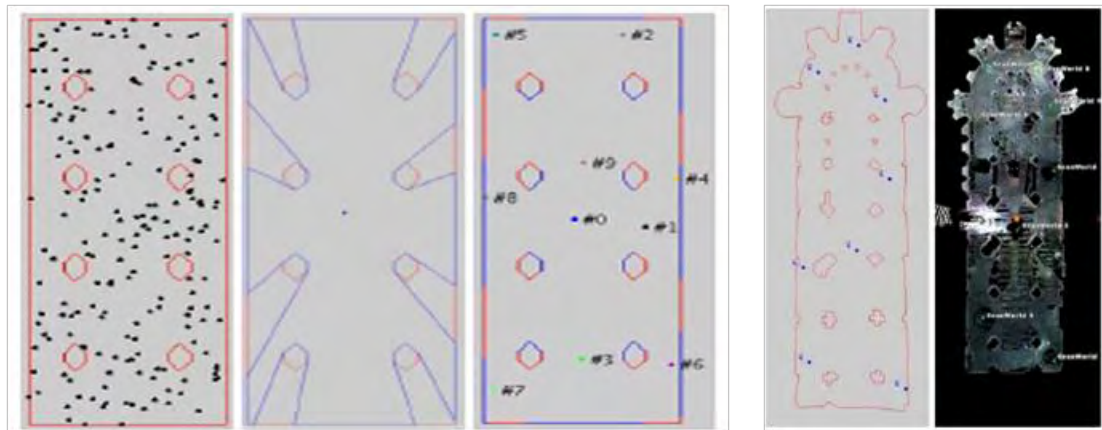


Figura 2-19: Algoritmo de NBV desarrollado en el proyecto AVENUE (Blaer & Allen, 2009)

Previamente al desarrollo de este trabajo, también merece la pena destacar el desarrollo de un robot para el modelado de interiores (Martín Nevado, Gómez García-Bermejo , & Zalama Casanova, 2004) (Pulido Fentanes, Marcos Pablos, Dominguez Quijada, Zalama Casanova, Gómez García-Bermejo , & Perán, 2009), en el cual se puede supervisar la información capturada desde un estación de supervisión remota en línea con el proceso de captura (ver Figura 2-20). Este robot tiene una arquitectura hardware y software similar a la arquitectura del robot construido en esta investigación y que será descrito en el Capítulo 5.

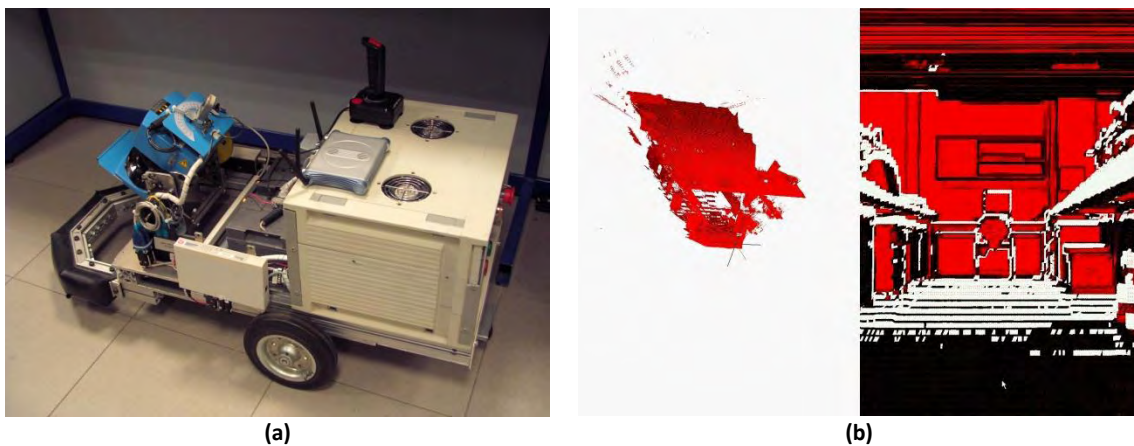


Figura 2-20: (a) Robot desarrollado en (Pulido Fentanes, Marcos Pablos, Dominguez Quijada, Zalama Casanova, Gómez García-Bermejo , & Perán, 2009), (b) Interfaz de visualización de datos 3D online.

Igualmente se han hecho esfuerzos importantes para modelar este tipo de entornos dentro del campo de la Robótica de emergencias (SSRR), para la cual este tipo de aplicación tiene un valor añadido muy grande. Por ejemplo, (Ohno, Tadokoro, Nagatani, Koyanagi, & Yoshida, 2009) desarrollaron un método para crear mapas tridimensionales usando odometría y alineamiento de escaneos basado en ICP, durante el simulacro de un atentado terrorista en un centro

comercial y una estación de metro. Para mejorar la fiabilidad del modelo, los autores proponen una ampliación del algoritmo de alineación de escaneos usando datos giroscópicos para filtrar la odometría reduciendo los deslizamientos.

Otro ejemplo interesante es el método desarrollado por (Pathak, Birk, Vaskevicius, Pflingstorn, Schwertfeger, & Poppinga, 2010) que describe una herramienta para la generación de modelos tridimensionales probada en diferentes escenarios de emergencia como un aparcamiento derrumbado y una inundación (ver Figura 2-21). La metodología propuesta por estos autores se diferencia de otras en que no utiliza un algoritmo de alineamiento de datos 3D, sino que se basa en la extracción de planos de la información 3D para usarlos como características para el SLAM.



Figura 2-21: Plataforma utilizada por (Pathak, Birk, Vaskevicius, Pflingstorn, Schwertfeger, & Poppinga, 2010)

En lo referente a exploración en escenarios de emergencia, aparte de las técnicas mencionadas en el apartado anterior, (Calisi, Farinelli, Iocchi, & Nardi, 2007) proponen un algoritmo de exploración basado en redes de Petri. Los autores presentan varias redes para que los robots cumplan con distintas misiones, por ejemplo, para hallar a las víctimas primero o el foco de un incendio. Esta implementación es sencilla y permite gran flexibilidad, sin embargo, su utilización requiere de un sistema perceptivo capaz de detectar múltiples situaciones.



Figura 2-22: Robot Participante en el DARPA Urban Challenge (Montemerlo, y otros, 2008)

Finalmente, en esta misma línea cabe también destacar los distintos proyectos desarrollados para participar en el DARPA challenge los cuales han implementado complejos y completos sistemas para la navegación y captura de información del terreno, sobre vehículos comerciales que han de cumplir autónomamente con una serie de misiones en diferentes entornos (ver

Figura 2-22). Es necesario señalar que en estos casos, la generación de un modelo tridimensional no es importante y los datos 3D son utilizados únicamente para percibir el entorno y navegar por él.

2.3. Metodología Propuesta

El objetivo principal de este trabajo es el desarrollo de un método de exploración que se base en la información capturada con anterioridad para hacer el proceso de reconstrucción mejor y más eficiente.

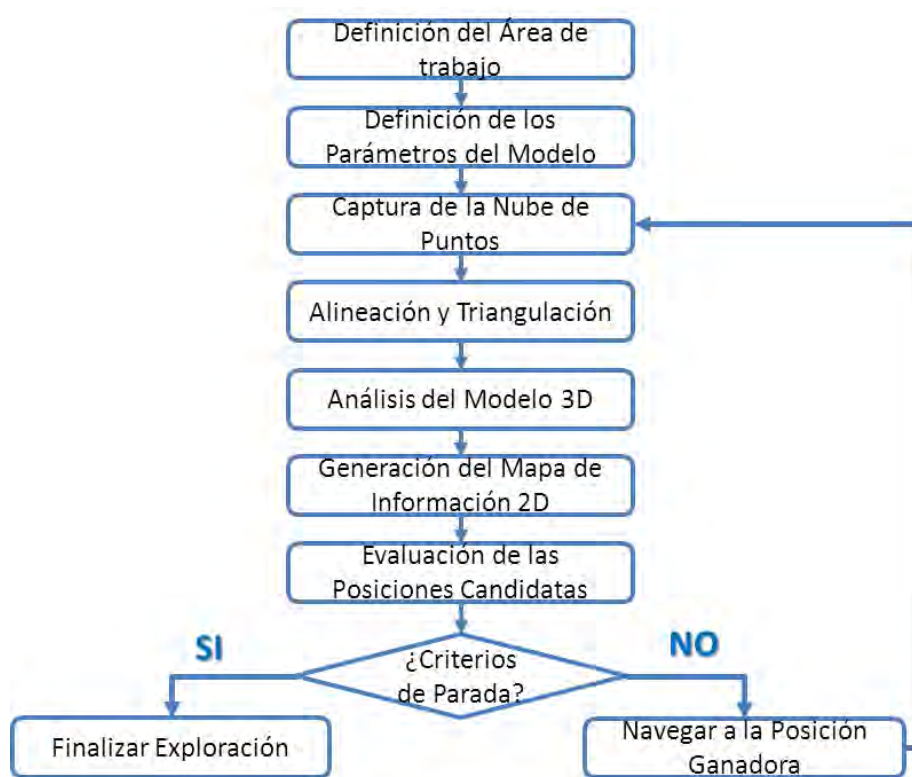


Figura 2-23: Proceso de exploración del entorno

Como se ha visto hasta el momento, existen múltiples metodologías para el modelado tridimensional de entornos exteriores. El método aquí propuesto se basa en una plataforma móvil todoterreno dotada de un escáner 3D capaz de capturar información tridimensional del entorno y en un algoritmo que utiliza la información tridimensional recogida para seleccionar la siguiente vista. Esto es una diferencia considerable con los sistemas empleados hasta el momento, ya que ninguno utiliza únicamente la información 3D recogida por el mismo para explorar el entorno.

Para lograr modelar el entorno por completo, el robot deberá desplazarse de un punto a otro tomando escaneos de forma estacionaria en cada lugar (Figura 2-23). Después de cada captura el robot debe usar su sistema de localización para calcular una posición estimada la cual será

utilizada para obtener el alineamiento de las nubes de puntos usando un algoritmo de SLAM6D. Una vez finalizado el alineamiento, se procede a seleccionar el siguiente punto de captura utilizando la información disponible para establecer los puntos que optimicen la consecución de modelos, aplicando para ello criterios de calidad, cantidad de información nueva y dificultad de navegación.

El proceso de selección de vista o exploración (Figura 2-24), está basado en tres sub-procesos principales, el primero es el análisis de la información 3D en el cual se determinan las zonas de navegación segura en el terreno la calidad del modelo resultante y se extraen algunas zonas de interés.

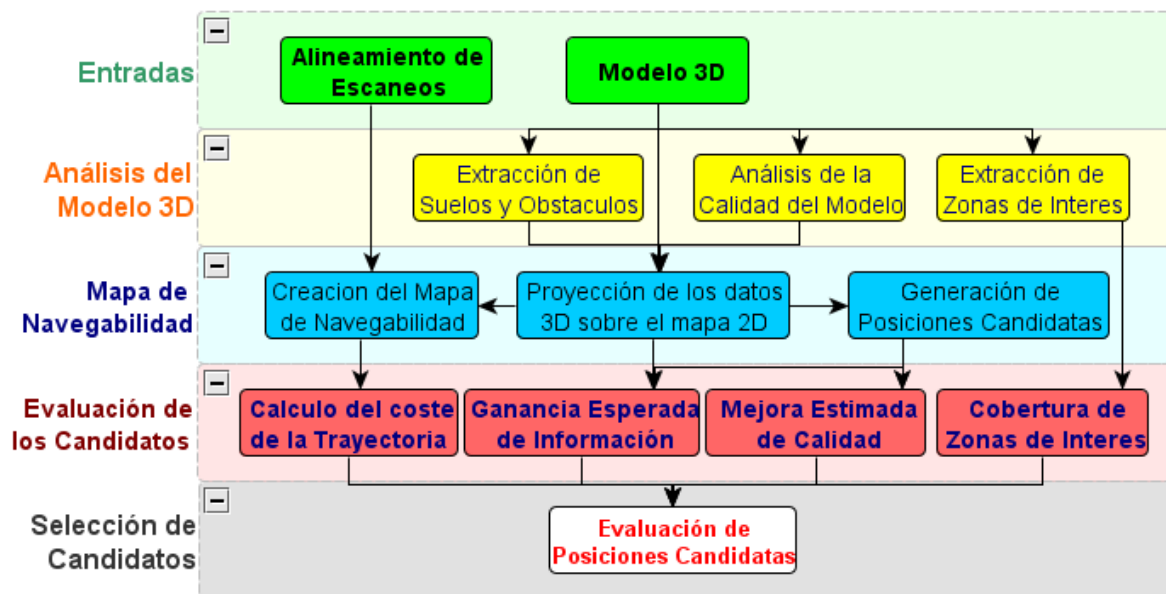


Figura 2-24: Proceso para la selección de vistas

El segundo sub-proceso es la proyección de esta información a un mapa 2D sobre el cual además se crearan una serie de posiciones candidatas sobre zonas de navegación segura de forma aleatoria. Cada una de estas posiciones es evaluada en el tercer paso, en el cual a cada candidata se le otorga una puntuación de acuerdo con una función de evaluación $u(c)$. La candidata con la puntuación más alta será la posición seleccionada para el siguiente escaneo.

La función de evaluación es una aplicación de los métodos de exploración *nbv* con múltiples criterios (Pulido Fentanes, Zalama, & Gómez-García-Bermejo, 2011), donde se considera la información tridimensional para evaluar cada uno de los criterios por los que se selecciona la siguiente vista. En este caso se consideran la ganancia esperada de información, la mejora en la calidad del modelo, las zonas ocluidas en escaneos anteriores que se cubren desde cada vista y la dificultad de alcanzar cada objetivo evaluado.

$$u(c) = \frac{w_A A(c) + w_Q Q(c) + w_O O(c)}{w_c C(c)} \quad 2.3$$

En la ecuación 2.3, $u(c)$ es la función de evaluación de cada posición candidata c , $A(c)$ es un valor normalizado que representa la cantidad de información nueva que se espera recoger desde cada posición c , $Q(c)$ es la mejoría esperada en la calidad del modelo, $O(c)$ es un valor

que representa la cantidad y calidad con la que las zonas de interés son cubiertas desde cada candidata, $C(c)$ es una estimación de la dificultad de navegar hasta c y finalmente w_A , w_Q , w_O y w_C son parámetros que indican la influencia de cada factor en la evaluación.

Estos criterios fueron seleccionados con el propósito de obtener un modelo que no solo contenga información del entorno de trabajo, sino que además la información recogida cumpla con un requerimiento de calidad mínimo. Se busca adicionalmente que el proceso de reconstrucción se haga de una forma eficiente reduciendo la cantidad de escaneos, el consumo energético y el tiempo necesario para crear un modelo del entorno que cumpla con unos estándares de calidad.

2.4. Conclusiones del Capítulo

En este capítulo se han presentado algunos conceptos básicos para entender la forma en que la presente investigación encaja dentro del ámbito de la robótica móvil y la generación de modelos tridimensionales. En el primer apartado se ha explicado la importancia y las dificultades de la adquisición de mapas para un robot móvil. Especialmente se ha hecho una introducción al problema del SLAM, el cual es el principal obstáculo para el mapeado en robots móviles. Posteriormente se ha explicado porque este problema es menos crítico en el caso del mapeado de exteriores donde a su vez la necesidad de utilizar información 3D hace más complejo el análisis de la información capturada por el robot.

Igualmente se han analizado los métodos de exploración de entornos más representativos. Como se ha observado, estos métodos están generalmente orientados al mapeado en entornos bidimensionales dejando a un lado los entornos exteriores. Esta ha sido una de las motivaciones principales de esta tesis. Por este motivo se ha propuesto una metodología de exploración basada en información tridimensional, la cual ha sido presentada en el apartado final de este capítulo.

Capítulo 3.

ANÁLISIS TRIDIMENSIONAL

Este capítulo presenta el tratamiento que se realiza sobre la información tridimensional para extraer, analizar y cuantificar las características del modelo capturado que son útiles para el proceso de exploración. El primer paso para la selección del siguiente punto de escaneo dentro de la metodología propuesta, es el análisis de la información capturada en el punto de escaneo.

Este proceso de análisis tridimensional se realiza sobre todos los puntos tridimensionales de la malla y su objetivo es extraer las superficies navegables del entorno y estimar la calidad del modelo en cada punto del mismo. El análisis es ejecutado tras cada adquisición de datos del terreno.

Una vez el robot ha capturado la información tridimensional en la posición en que se encuentra situado en ese momento, debe proceder a decidir cuál será la siguiente posición a la que debe dirigirse. Para ello es imperativo analizar la información adquirida y extraer los datos más relevantes para el proceso de exploración. En el análisis tridimensional se enmarcan todos los procesos que se realizan sobre la nube de puntos adquirida en la última posición, con el propósito de facilitar la evaluación de las características del entorno utilizadas para el proceso de exploración.

La información requerida por este proceso comprende la nube de puntos y una hipótesis de la posición de captura de la misma. Con esta información se ejecutan tres tareas distintas: el **pre-procesamiento de la nube de puntos**, donde se construye la malla y se alinea la nube de puntos con el resto del modelo del entorno; la **extracción de las características geométricas de los puntos** donde se extraen la información básica de la malla tridimensional indispensable para el siguiente paso; y finalmente el **análisis y evaluación de las características del entorno** donde se extrae del modelo la información relevante para la evaluación de las posiciones objetivo.

El resultado esperado tras este proceso es un modelo tridimensional con una calificación para cada una de las características elegidas en cada punto del mismo, además de la matriz de alineación del último escaneo con respecto al sistema de coordenadas de referencia.

3.1. Tratamiento Previo Del Modelo

En esta etapa se prepara la información tridimensional para que pueda ser útil para el análisis de las características del entorno a partir de la información 3D. Este preprocesamiento de la información es vital ya que la alineación de la nube de puntos es la que permite referenciar la información capturada con respecto al entorno de trabajo. Sin esta referenciación es imposible seleccionar vistas de exploración tras haber realizado más de una toma del entorno. Además sería imposible obtener un modelo adecuado del entorno.

La construcción de la malla es otro proceso muy importante, que se realiza únicamente sobre la última nube de puntos capturada y que facilita la extracción de información del entorno. Por ejemplo, la calidad de la información tridimensional de una zona depende directamente de la densidad de puntos sobre esta zona y esta densidad de puntos es inversamente proporcional al área de los triángulos de la malla.

3.1.1. Construcción de la Malla

La malla se construye aprovechando la forma ordenada en la que el escáner 3D captura la nube de puntos (ver sección 5.3). Este proceso se realiza generalmente en los escáneres láser que dan los datos de forma ordenada. Debe realizarse de tal manera que resulte posible crear los triángulos entre puntos vecinos (por ejemplo, entre los dos primeros puntos de una lectura láser, y el primer punto de la lectura siguiente). Cuando se trata de un escáner que no provee la información de manera ordenada, se pueden utilizar algoritmos de triangulación conocidos, como los métodos que utilizan triangulación de Delaunay (Su & Drysdale, 1996) o Marching Cubes (Chernyaev, 1995).

El primer paso para triangular es organizar los puntos en una rejilla esférica con la misma resolución angular del escáner, de forma que cada celda contenga un punto tridimensional. Una vez creada esta rejilla, se generan triángulos entre todos los puntos de las celdas vecinas omitiendo las celdas que no contengan puntos (es decir que los puntos estén fuera del rango del escáner).

Una vez que se han creado todos los triángulos, estos se evalúan independientemente usando la ley del seno para establecer una relación entre el largo de los lados y el seno de sus ángulos, tal como se expresa en

$$\gamma = \frac{\sin(\theta_1)}{\sin(\theta_2)} \quad 3.1$$

donde γ es la relación entre ángulos y lados del triángulo expresada por la ley del seno, θ_1 y θ_2 son los ángulos del triángulo que se puede ver en la Figura 3-1.

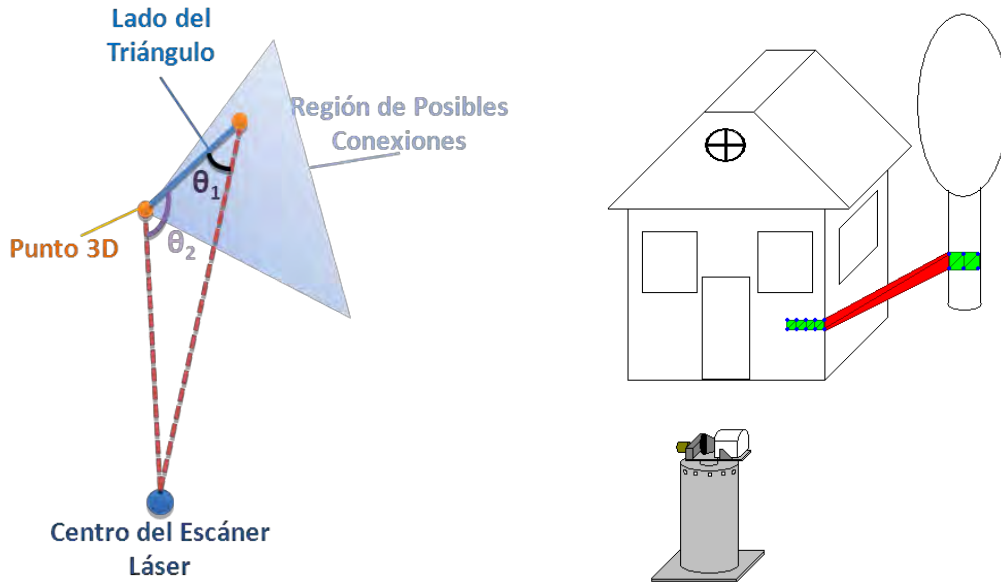


Figura 3-1: Metodología para el descarte de triángulos

A continuación, se procede a descartar aquellos puntos para los que esta relación está fuera de ciertos límites, γ_{\min} y γ_{\max} , ya que los triángulos que están por fuera de estos límites suelen tener lados muy desiguales, lo que generalmente significa que están compuestos por puntos correspondientes a dos superficies diferentes como se puede observar en la Figura 3-1.

Los triángulos descartados se almacenan en una lista aparte que puede ser utilizada para buscar los planos de oclusión.

El modelo triangulado es muy útil para la evaluación de la calidad de los datos capturados así como para la extracción de superficies del entorno (tales como terreno navegable u obstáculos).

3.1.2. Alineamiento y Rotación Del Modelo

Alineamiento del Modelo

El alineamiento del modelo se realiza utilizando un algoritmo de SLAM6D (Nüchter, Lingemann, Hertzberg, & Surmann, 2007). Esta solución está basada en el registro de escaneos sucesivos alineándolos por medio del algoritmo ICP (Besl & McKay, 1992). Ahora bien, el alineamiento sucesivo de escaneos con ICP no es suficiente para crear mapas consistentes de

forma global. Por este motivo para que el mapa obtenido tenga un número mínimo de errores y pueda lograrse de la forma más rápida posible, deberán realizarse los siguientes pasos:

1. **Calcular la Posición del Robot en 6 grados de libertad:** para este propósito se utiliza el sistema de localización del robot, el cual provee tanto la posición (x, y, z) utilizando la información de GPS, acelerómetros y odometría, como la orientación del robot (θ_x , θ_y , θ_z) utilizando sensores inerciales y de campo magnético.

En este paso se extraen los valores para los seis grados de libertad, los cuales se representan usando una matriz P_{odo} para la posición del sistema, que incluye el vector de traslación t y la matriz de rotación R la cual se complementa de la siguiente forma:

$$P_{odo} = \begin{pmatrix} R & 0 \\ t & 1 \end{pmatrix} = \begin{pmatrix} \cos\theta_y \cos\theta_z & \sin\theta_x \sin\theta_y \cos\theta_z + \cos\theta_x \sin\theta_z & -\cos\theta_x \sin\theta_y \cos\theta_z + \sin\theta_x \sin\theta_z & 0 \\ -\cos\theta_y \sin\theta_z & -\sin\theta_x \sin\theta_y \sin\theta_z + \cos\theta_x \cos\theta_z & \cos\theta_x \sin\theta_y \sin\theta_z + \sin\theta_x \cos\theta_z & 0 \\ \sin\theta_y & -\sin\theta_x \cos\theta_y & \cos\theta_x \sin\theta_y & 0 \\ x & y & z & 1 \end{pmatrix} \quad 3.2$$

2. **Calcular las estimaciones iniciales para la alineación ICP basándose en este cálculo:**

Este es un proceso de alineación de escaneos preliminar que permite mejorar la estimación inicial de la posición del robot. Se basa en la división del ultimo escaneo y el escaneo inmediatamente anterior en arboles tipo “*octree*”, los cuales permiten encontrar coincidencias entre datos tridimensionales.

Estos árboles se construyen dividiendo el espacio sucesivamente en octantes mientras estos contengan puntos (ver Figura 3-2). Cada nivel de profundidad del árbol se corresponde con un volumen de cubo diferente y gracias a esta estructura es posible reducir la cantidad de datos a manejar para encontrar el mejor alineamiento de los datos.

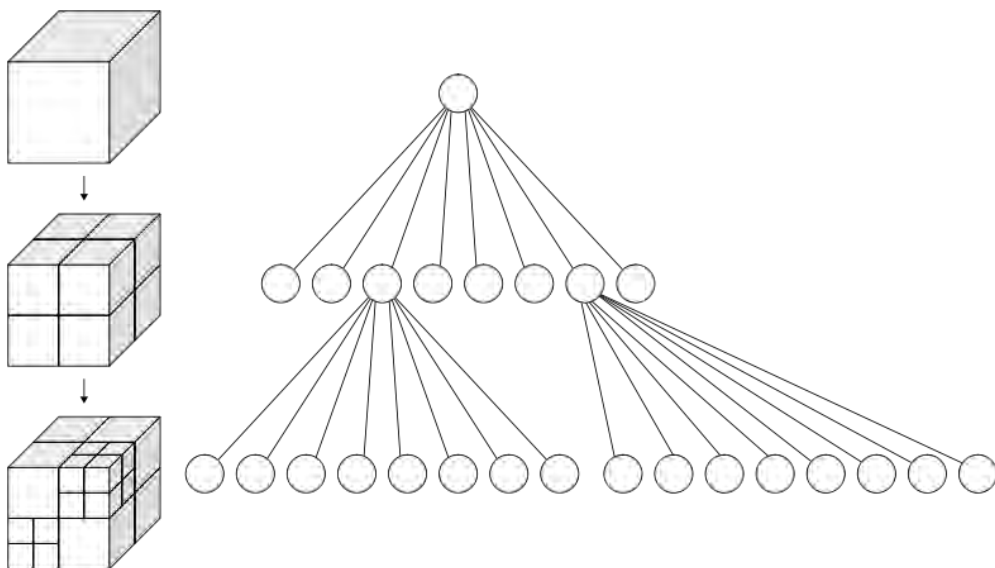


Figura 3-2: Representación gráfica de los “*octree*”.

Usando estas representaciones se busca la matriz de desplazamiento sobre la cual se superpongan más cubos de los dos modelos. Esta búsqueda se realiza entre dos niveles de profundidad S_{inicio} y S_{fin} calculando para cada nivel de profundidad la mejor transformación ΔP_{best} .

En cada nivel de profundidad se define una distancia máxima de desplazamiento ΔP_{max} en metros y grados para cada uno de los seis grados de libertad $(x, y, z, \theta_x, \theta_y, \theta_z)$ y se prueba la superposición entre los modelos para todas las combinaciones discretas entre $[-\Delta P_{max} + \Delta P_{best}, \Delta P_{max} + \Delta P_{best}]$ siendo $\Delta P_{best} = 0$ para el primer nivel.

ΔP_{best} es la transformación para la que existen más cubos superpuestos entre sí en el proceso. De esta forma, la matriz de posición preliminar resultante P_{ini} resulta:

$$P_{ini} = P_{odo} + \Delta P_{best} \quad 3.3$$

- 3. Registrar los escaneos 3D en un sistema de coordenadas único usando ICP:** Usando la posición estimada anteriormente, se alinea el último escaneo con el escaneo anterior usando el método ICP que ha sido modificado para que no se pueda desplazar la matriz de transformación más allá de una distancia máxima d_{max} .

La alineación entre dos conjuntos de datos tridimensionales \hat{M} y \hat{D} consiste en encontrar una matriz de rotación R y un vector traslación t que minimice el error estimado por la siguiente ecuación:

$$E(R, t) = \sum_{i=1}^{|\hat{M}|} \sum_{j=1}^{|\hat{D}|} \|\hat{m}_i - (R\hat{d}_j + t)\|^2 \quad 3.4$$

El algoritmo ICP calcula de forma iterativa las correspondencias para los puntos. En cada iteración, el algoritmo selecciona los puntos más cercanos entre sí como correspondencias y calcula la transformación (R, t) que minimice la ecuación 3.4. En la última iteración se supone que la transformación es correcta.

Sin embargo, hay que tener en cuenta que en este caso, los escaneos no se superponen completamente. Ello puede originar que la estimación de la última iteración no sea totalmente correcta, sino que contenga errores o desviaciones. Para compensar esta situación es necesario almacenar las estimaciones de cada iteración y al finalizar el proceso, seleccionar la que parezca más adecuada para ser utilizada como válida.

- 4. Cerrar el lazo y redistribuir el error:** Dado que el alineamiento de escaneos se realiza por parejas, el error global es acumulativo de forma que se pueden producir grandes errores en la posición estimada para los últimos escaneos.

Esto se soluciona alineando el último escaneo con los escaneos anteriores que estén a su alcance, basándose en el alcance máximo del escáner y la posición hipotética del robot. Esta hipótesis se verifica utilizando el algoritmo descrito en el paso 2.

Si el alineamiento entre el último escaneo y un escaneo anterior es posible, se detecta un lazo cerrado y se alinean estos dos escaneos. La matriz de transformación obtenida tras esta alineación es propagada de forma proporcional a la distancia a todos los escaneos que se encuentren entre los dos escaneos alineados de esta forma:

$$c_i = \frac{\text{distancia desde el inicio del lazo hasta el escaneo } i}{\text{distancia total del lazo}} \quad 3.5$$

$$t_i = c_i t \quad 3.6$$

$$R_i = c_i R \quad 3.7$$

- 5. Relajación Global:** Una vez han sido tomados todos los escaneos del entorno, se puede realizar este proceso para refinar el modelo. Este algoritmo alinea todos los escaneos entre sí tomando cada escaneo y alineándolo con todos los demás a la vez, hasta que la matriz de traslación resultante implique un desplazamiento menor a una distancia dada.

Rotación Del Modelo 3D

Una vez obtenidas las matrices de alineación para cada escaneo, se multiplica cada punto de la nube de puntos por la componente de rotación de la nube. El objetivo de este proceso es compensar cualquier inclinación del robot al momento de capturar la escena de forma que el eje **Z** real del entorno y el del modelo coincidan.

Esto facilita especialmente el proceso de extracción de información del modelo, ya que por ejemplo la inclinación de un plano en la nube de puntos coincidirá con la inclinación que debe sortear el robot a la hora de navegar por el entorno. Por otra parte, no se traslada la nube de puntos ya que la posición relativa de un punto a la hora de escanearlo es útil para calcular otras variables como el ángulo entre la superficie a la que pertenece y el escáner en momento de la captura, lo cual como se verá más adelante es importante para estimar la calidad del modelo.

Una vez calculada la matriz de alineamiento para el escaneo, ésta se usa para rotar todos los puntos del último escaneo de forma que el eje Z del modelo sea perpendicular al suelo del entorno. De esta manera se facilita enormemente la extracción de las características del entorno usadas para la exploración.

Para una nube de puntos **M**, se extrae de su matriz de alineación **T_M**, la componente de rotación **R**, la cual es una matriz de 3x3 de la siguiente forma:

$$\mathbf{R} = \begin{bmatrix} \mathbf{T}_{M_{11}} & \mathbf{T}_{M_{12}} & \mathbf{T}_{M_{13}} \\ \mathbf{T}_{M_{21}} & \mathbf{T}_{M_{22}} & \mathbf{T}_{M_{23}} \\ \mathbf{T}_{M_{31}} & \mathbf{T}_{M_{32}} & \mathbf{T}_{M_{33}} \end{bmatrix} \quad 3.8$$

Esta matriz se multiplica por el vector \vec{P} que representa las coordenadas (x,y,z) de cada punto de la nube de puntos \mathbf{M} , de forma que:

$$\mathbf{T}_{M+1} = \vec{P}_M \cdot \mathbf{R} \quad 3.9$$

$$\mathbf{T}_{M+1} = [P_x \quad P_y \quad P_z] \cdot \begin{bmatrix} \mathbf{R}_{11} & \mathbf{R}_{12} & \mathbf{R}_{13} \\ \mathbf{R}_{21} & \mathbf{R}_{22} & \mathbf{R}_{23} \\ \mathbf{R}_{31} & \mathbf{R}_{32} & \mathbf{R}_{33} \end{bmatrix} \quad 3.10$$

Al aplicar este proceso la matriz de alineación \mathbf{T}_M ya no es válida puesto que utilizarla significaría realizar dos veces la misma transformación. Por ello, tras este proceso se debe utilizar la matriz \mathbf{T}_{M+1} para la nube de puntos, obtenida como:

$$\mathbf{T}_{M+1} = \begin{pmatrix} \mathbf{I} & 0 \\ \vec{p} & 1 \end{pmatrix} \quad 3.11$$

$$\mathbf{T}_{M+1} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ x & y & z & 1 \end{pmatrix}$$

La nube de puntos rotada se usa para extraer las características del entorno utilizadas por el algoritmo de exploración. La matriz de alineación se usa para proyectar cada uno de los puntos sobre el mapa de información 2D

3.2. Análisis De Las Características Geométricas Del Modelo

Para facilitar la extracción de la información de navegabilidad y calidad del modelo es conveniente hacer un preprocesamiento de la malla. En concreto, se busca calcular algunas características del modelo de manera que el manejo ulterior de la información sea más rápido. En el presente trabajo, para cada punto y triangulo del modelo se calcula y almacena la siguiente información:

Área por punto $P_{ar}(p)$: El área representada por cada punto es la suma de la tercera parte¹ de las áreas de los triángulos a los que pertenece el punto (ver Figura 3-3). Esta información es especialmente útil para evaluar la calidad del modelo.

$$P_{ar}(p) = \frac{\sum_{i \in \psi(p)} A_{Ti}}{3} \quad 3.12$$

donde $\psi(p)$ es el conjunto de triángulos a los que pertenece el punto p , y A_T es el área del triángulo.

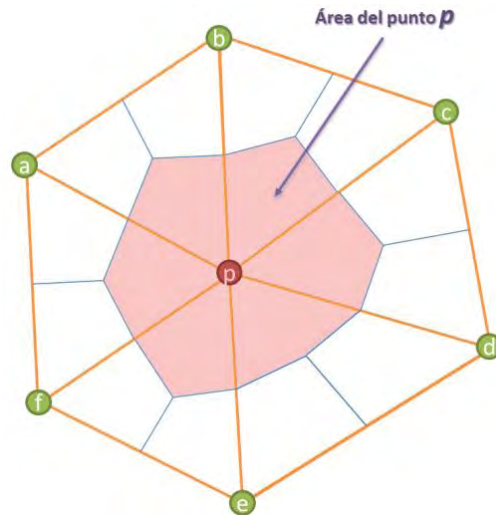


Figura 3-3: Representación del área por punto para el punto p

Vectores normales del punto $\vec{N}(p)$: Se calculan haciendo una suma ponderada de las normales de cada triángulo al que pertenece el punto y asignando a dichas normales un peso proporcional al área del triángulo al que pertenecen, como muestra la ecuación:

$$\vec{N}(p) = \frac{\sum_{i \in \psi(p)} A_{Ti} \vec{N}_{Ti}}{\sum_{i \in \psi(p)} A_{Ti}} \quad 3.13$$

donde $\psi(p)$ es el conjunto de triángulos a los que pertenece el punto p , A_T es el área del triángulo y \vec{N}_T es el vector normal al triángulo.

Triángulos y puntos vecinos: Para cada punto se almacena la lista de índices de los triángulos a los que pertenece y otra lista con los puntos vecinos del mismo (puntos con los que conforma triángulos), para aumentar la rapidez del algoritmo.

Triángulos vecinos: Se denominan de esta forma los triángulos que comparten lados entre sí. Para cada triángulo se almacena una lista con los triángulos vecinos al mismo, lo cual simplifica la extracción de zonas de interés.

¹ El área total de cada triángulo dividida entre el número de vértices que lo conforman, que siempre será tres.

3.3. Extracción De Información Del Modelo

3.3.1. Extracción De Obstáculos Y Superficies Navegables

Los objetos del entorno pueden ser clasificados en superficies navegables, obstáculos u objetos inalcanzables (Nüchter & Hertzberg, 2008). La identificación de las dos primeras es crucial para estimar las trayectorias seguras sobre un entorno.

Extracción De Superficies Navegables

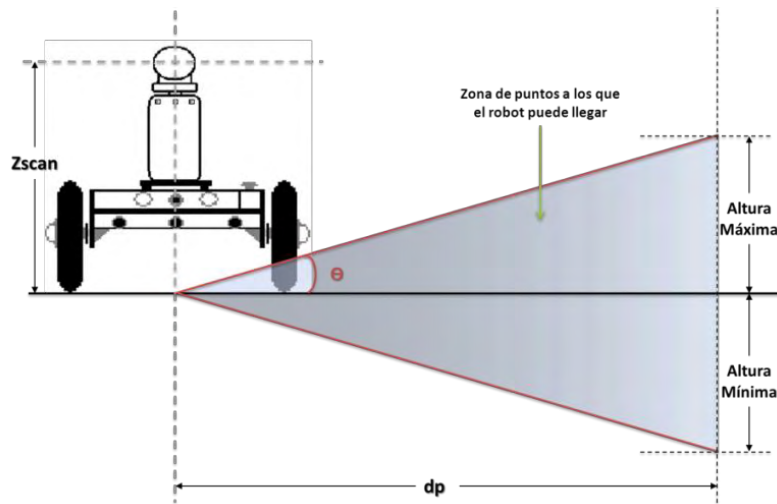


Figura 3-4: Zona considerada para la extracción de áreas de navegación segura

La extracción de áreas de navegación seguras se hace calculando una puntuación según la posibilidad de pertenencia a estas zonas para cada punto de la malla. Así, si un punto está en un ángulo alcanzable para el robot (es decir, que el robot no tiene que escalar más allá de sus posibilidades para alcanzarlo, ver Figura 3-4) y su vector normal tiene una proyección alta sobre el eje Z, entonces se considera que este punto tiene una alta probabilidad de pertenecer a una zona de navegación segura y por ende se le otorga una puntuación alta.

Por ejemplo, en la Figura 3-5 se puede observar como los puntos cuya normal cuenta con una componente alta sobre el eje Z (que como se vio en la sección 3.1.2, coincide con el eje Z del mundo) se corresponde generalmente con la superficie navegable del entorno.

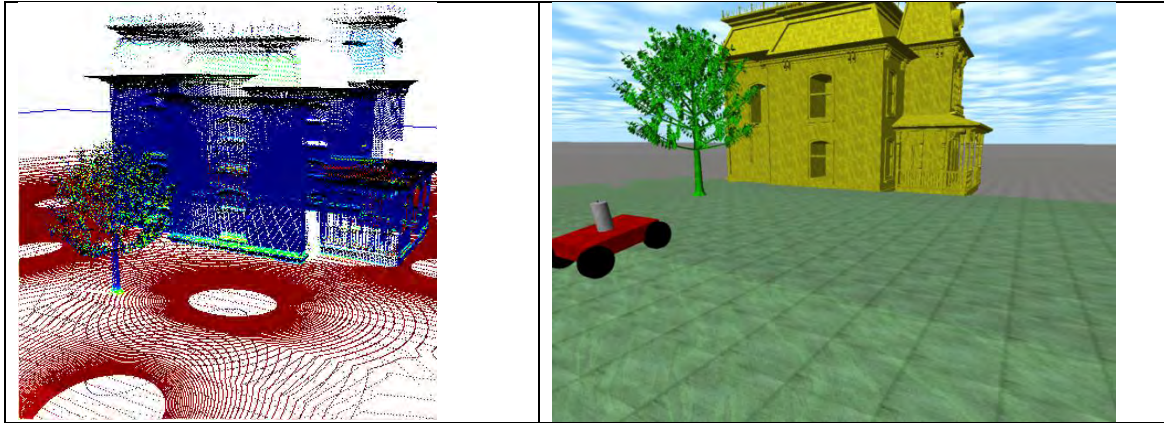


Figura 3-5: Magnitud de la componente Z de las normales de cada punto (izquierda), los puntos en rojo representan puntos con valores elevados para la componente Z del vector normal a cada punto, escaneo realizado sobre un entorno simulado (derecha, ver Sección 5.4).

La magnitud de la componente Z de la normal de cada punto y su área son utilizadas para calcular una puntuación *local* $F_{pl}(p)$ que es un valor real entre 0 y 1, y se calcula únicamente para los puntos que estén al alcance del robot (ver Figura 3-4). Dicha puntuación se obtiene utilizando la ecuación 3.14.

$$F_{pl}(p) = S(p) \cdot H(p) \quad 3.14$$

donde $S(p)$ es una función que otorga una puntuación de navegabilidad según la proyección de la normal del triángulo sobre el eje Z y la capacidad de escalada del robot, en la forma:

$$S(p) = \frac{1}{1 + e^{\frac{6 \cdot (N_z(p) - \cos(\theta))}{(1 - \cos(\theta))}}} \quad 3.15$$

En esta expresión, $N_z(p)$ es la magnitud del componente Z del vector normal al punto y θ es el ángulo de inclinación máxima que el robot puede escalar. Esta función premia a los puntos que pertenezcan a superficies con inclinación menor que θ con puntuaciones mucho más altas que los puntos con inclinaciones mayores.

El objetivo de la función $S(p)$ es diferenciar claramente las puntuaciones para los puntos que pueden pertenecer al suelo, de las puntuaciones de los puntos que pertenecen a superficies demasiado inclinadas para el robot pueda atravesarlas, por estar más allá de su capacidad de escalada.

La Figura 3-6 muestra el comportamiento de esta función para distintos valores de θ . Se puede ver como las puntuaciones $S(p)$ son siempre altas para los valores de $N_z(p)$ cercanos a uno, y decrecen a medida que va disminuyendo el valor de $N_z(p)$ lo que significa que el plano es más inclinado respecto al suelo. En concreto, el valor de $S(p)$ disminuye hasta alcanzar la puntuación media (0.5) en el valor de $N_z(p)$ que se corresponde con la inclinación máxima que el robot puede escalar (línea roja en los gráficos de la Figura 3-6), y luego desciende hasta cero en la misma proporción.

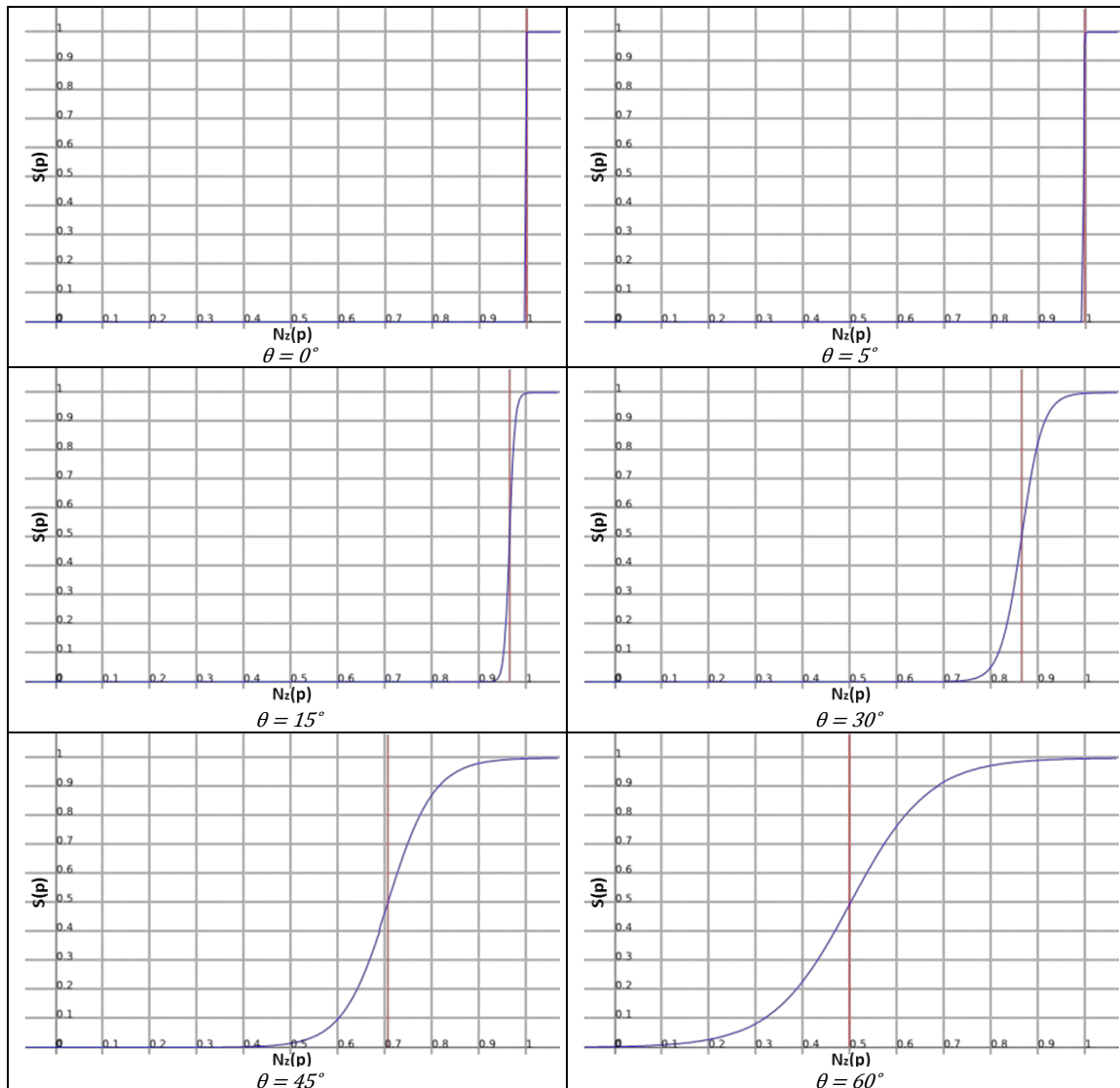


Figura 3-6: Función $S(p)$ según la capacidad de escala del robot definida por θ , la línea roja marca la frontera entre los puntos que están dentro de la capacidad de escalada del robot y los que no.

Este tipo de representación fue seleccionada porque permite evaluar más claramente los esfuerzos que debe realizar un robot escalando un plano inclinado. Por ejemplo, a un robot con capacidad de escalar un plano de 45° no le costará mucho escalar un plano de 15° de inclinación. Sin embargo un robot con una capacidad de escalada de cinco grados como máximo será extremadamente sensible a cualquier inclinación del terreno. También se permite que inclinaciones superiores a la capacidad de escalada del robot obtengan puntuaciones mayores que 0. Esto tiene como propósito principal que pequeños escollos no afecten demasiado la puntuación general de sus vecinos.

$H(p)$ es una función que otorga una puntuación de navegabilidad a cada punto según su altura con respecto a la posición con la que fue escaneada:

$$H(p) = e^{-\frac{P_z(p)^2}{2d(p)\sin(\theta)}} \quad 3.16$$

Aquí $Pz(p)$ es la diferencia de altura del punto p con respecto a la base del robot, $d(p)$ es su distancia hasta el escáner sobre el plano XY y θ es el ángulo de inclinación máxima que el robot puede escalar. Al igual que con la función $S(p)$, en este caso se intenta dar más relieve a los puntos que mejor cumplan las condiciones para ser considerados alcanzables.

La Figura 3-7 muestra cómo se comporta la función para puntos a diferentes distancias y con diferentes valores de θ . Se puede observar que los puntos que están a la misma altura del robot siempre obtienen puntuaciones altas y a medida que la diferencia de altura entre el punto y el robot va aumentando, los puntos son penalizados en función de la capacidad de escalada del robot y de la distancia entre este y el punto.

De esta manera un punto alejado del robot y con una diferencia de altura elevada recibirá una doble penalización. Esto se ha diseñado así para considerar igualmente el esfuerzo necesario para alcanzar un punto, ya que por ejemplo alcanzar un punto a diez metros de distancia y a diez metros de altura requiere un gran esfuerzo incluso cuando la capacidad de escalada del robot permitiría alcanzar dicho punto. De esta forma se premian indirectamente las posiciones que son fácilmente alcanzables.

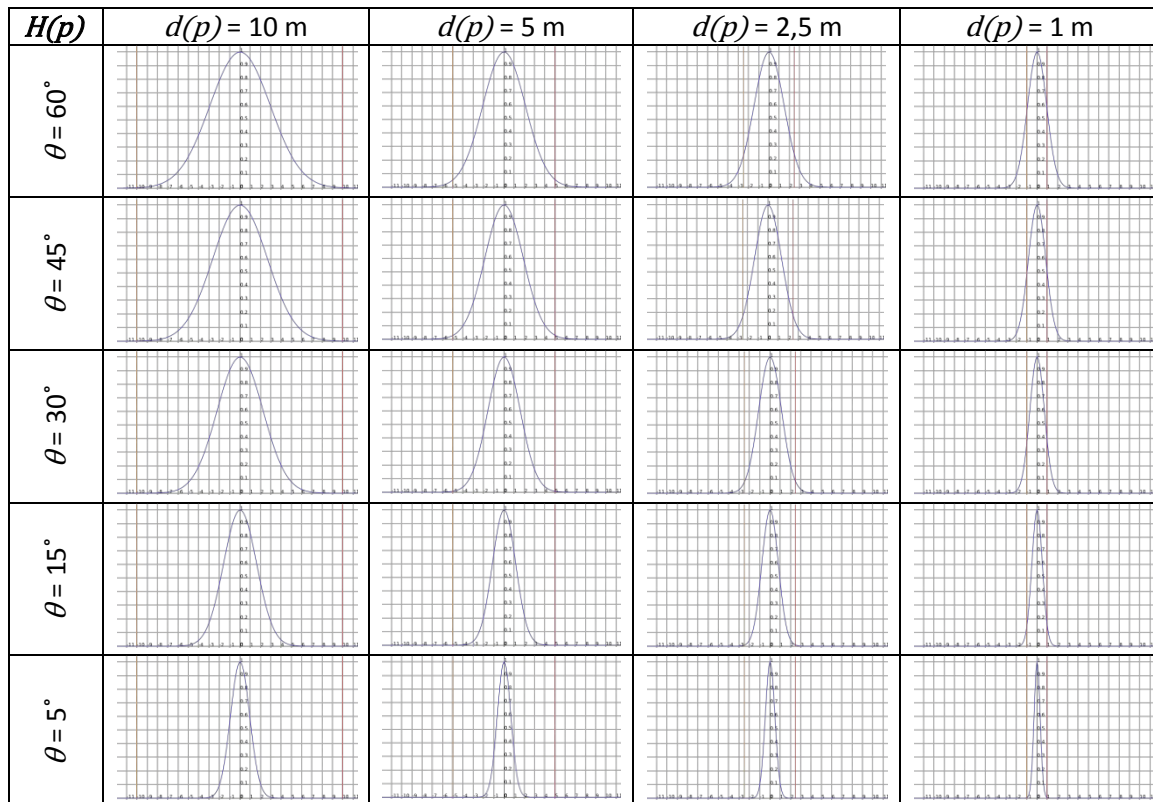


Figura 3-7: Función $H(p)$ con respecto a la altura del punto desde la base del robot. Según la capacidad de escalada del robot (θ) y la distancia hasta el punto.

En las ecuaciones 3.15 y 3.16 los términos exponenciales penalizan en mayor medida a los puntos con peores características de forma que la puntuación en estos casos sea cercana a cero, eliminando así la posibilidad de que sean considerados parte de una superficie navegable.

Hasta ahora esta puntuación solo evalúa la situación de un punto y no de su entorno, así que un punto ubicado en una superficie elevada puede obtener un puntaje alto aunque sus vecinos

tengan puntajes muy bajos. Por este motivo se calcula un segundo valor que representa esta misma condición dependiendo del entorno de cada punto. Esta puntuación $F_{pg}(p)$ es el promedio de las puntuaciones *locales* de los vecinos de cada punto p , como se puede ver en la siguiente ecuación:

$$F_{pg}(p) = \frac{1}{n} \sum_{i \in \gamma(p)} F_{pl}(i) \quad 3.17$$

donde $\gamma(p)$ es el conjunto de puntos vecinos al punto p y n es el número de estos. Una vez computados estos dos valores se obtiene una puntuación final haciendo una suma ponderada de estos valores utilizando dos parámetros w_l y w_g para asignar un peso a cada puntaje.

$$F(p) = w_l F_{pl}(p) + w_g F_{pg}(p) \quad 3.18$$

Ajuste de los Parámetros

El parámetro w_l pondera la influencia que tiene la situación local del punto sobre el puntaje $F(p)$. Si este parámetro representase el 100 % del valor de la puntuación, todos los puntos tendrían una puntuación que se correspondería únicamente a su situación local. El parámetro w_g por el contrario permite la eliminación de zonas apartadas y la reducción de los puntos fronterizos.

En resumen, aumentar w_l permite dar una puntuación más alta a una mayor cantidad de puntos, mientras que incrementar w_g reducirá la puntuación para las zonas aisladas y los bordes del suelo.

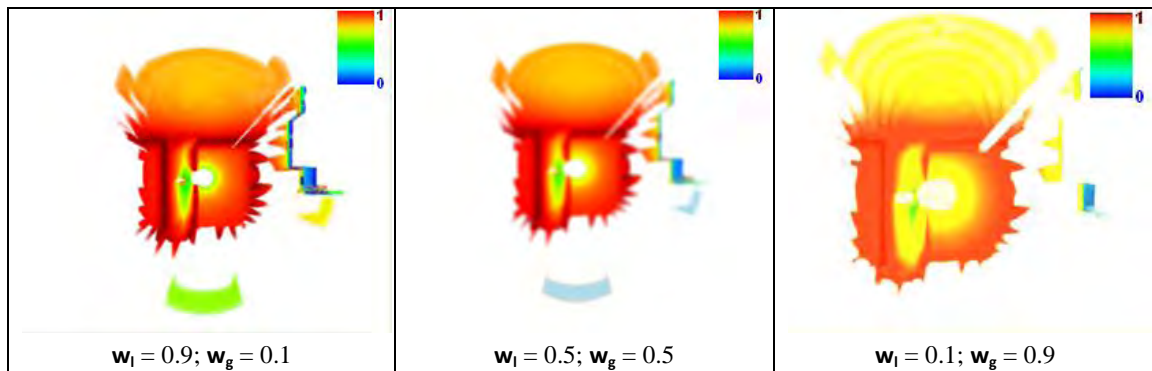


Figura 3-8: Efecto de los Parámetros de Ajuste Sobre la Extracción de Superficies Navegables, los valores altos para $F(p)$ están en rojo. Los puntos con valores de 0 no están representados.

La Figura 3-8 muestra cómo los parámetros afectan a la extracción de superficies seguras sobre un mismo escaneo. En la imagen de la izquierda la situación local del punto es la dominante y por eso se aprecia cómo existen diferencias claras entre las distintas zonas capturadas. Igualmente se puede ver que hay algunas zonas con puntuaciones bajas y medias. En la imagen central se puede ver que comparativamente con la imagen anterior se distinguen menos cambios puntuación y en las zonas aisladas las puntuaciones han bajado.

Finalmente, en la imagen de la derecha donde la situación global es la dominante se puede ver cómo las puntuaciones se han uniformado a lo largo del entorno. Además, las zonas aisladas han desaparecido. En otras palabras, se puede apreciar cómo el incremento del parámetro w_g va disminuyendo la cantidad de zonas aisladas y distribuye el valor de la puntuación de suelo a lo largo del entorno.

Vale la pena destacar que estos parámetros influyen principalmente en la creación del mapa de navegabilidad (apartado 4.1), en cuya descripción se puede ver más claramente la influencia que tienen en la extracción de las zonas de navegación segura. Igualmente, el uso de estos parámetros otorga flexibilidad al sistema para manejar entornos con características especiales como rizados o muy planos.

Extracción De Obstáculos

La detección de obstáculos se realiza usando las mismas premisas, si bien en este caso sólo se consideran los puntos que según su posición, puedan afectar la navegación del robot. Al igual que en la sección anterior, para esta evaluación solo se consideran también los puntos que están a una cierta altura por encima de la zona considerada (ver Figura 3-4). Dicha altura corresponde a la altura del robot con el escáner (1 metro en nuestro caso). La Figura 3-9 muestra los puntos que se consideran en un escaneo.

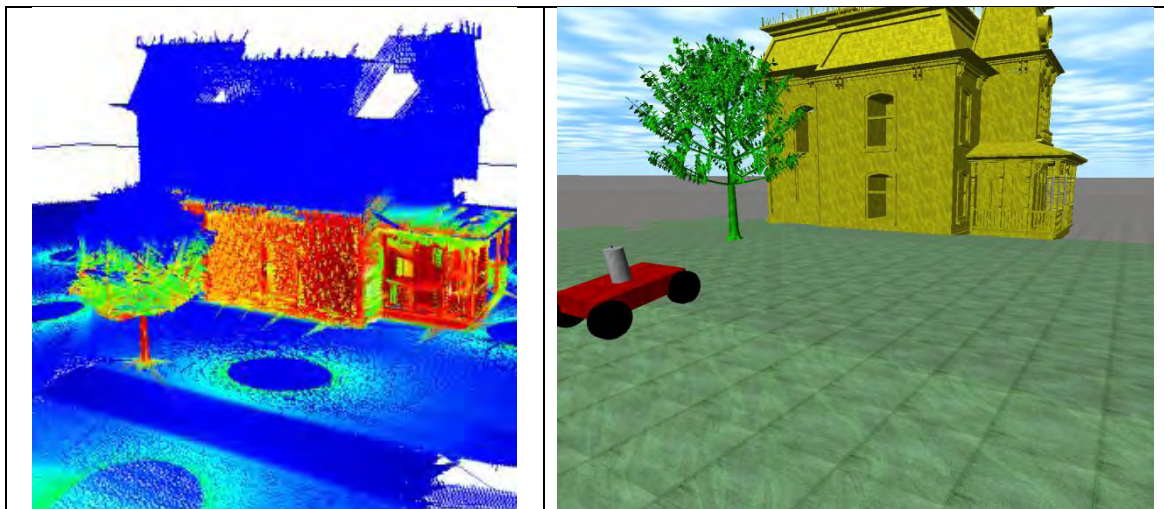


Figura 3-9: Puntos considerados para la extracción de obstáculos (en rojo). Entorno simulado de ejemplo.

Un punto obtendrá un valor alto en su probabilidad de pertenencia a un obstáculo si, además de estar en una posición alcanzable por el robot, el plano al que pertenece es perpendicular al suelo. Estos factores se usan para calcular una puntuación *local* $B_l(p)$. Los puntos vecinos también son considerados y se utilizan para calcular una puntuación desde el punto de vista *global* $B_g(p)$.

$$B_l(p) = \left(\frac{N_{xy}(p)Ar(p)}{A_{max}} \right) \left(\frac{\vec{P}}{|\vec{P}|} \cdot \vec{N}_p \right) \quad 3.19$$

$$B_g(p) = \frac{1}{n} \sum_{i \in \gamma(p)} B_l(i) \quad 3.20$$

En estas expresiones, $|N_{xy}|$ es la magnitud de la suma vectorial de las componentes de la normal del punto en los ejes X e Y, y \vec{P} es un vector desde el centro del escáner láser hasta el punto p y \vec{N}_p es vector normal para este punto. Una vez calculados $B_l(p)$ y $B_g(p)$ se obtiene una puntuación final haciendo una suma ponderada con dos parámetros w_{bl} y w_{bg} para asignar un peso a cada valor:

$$B(p) = w_{bl}B_l(p) + w_{bg}B_g(p) \quad 3.21$$

Ajuste de los Parámetros

El parámetro w_{bl} representa la influencia que tiene la situación local del punto sobre el puntaje, el parámetro w_{bg} por el contrario permite la eliminación de pequeños objetos y la reducción de las puntuaciones en los puntos fronterizos.

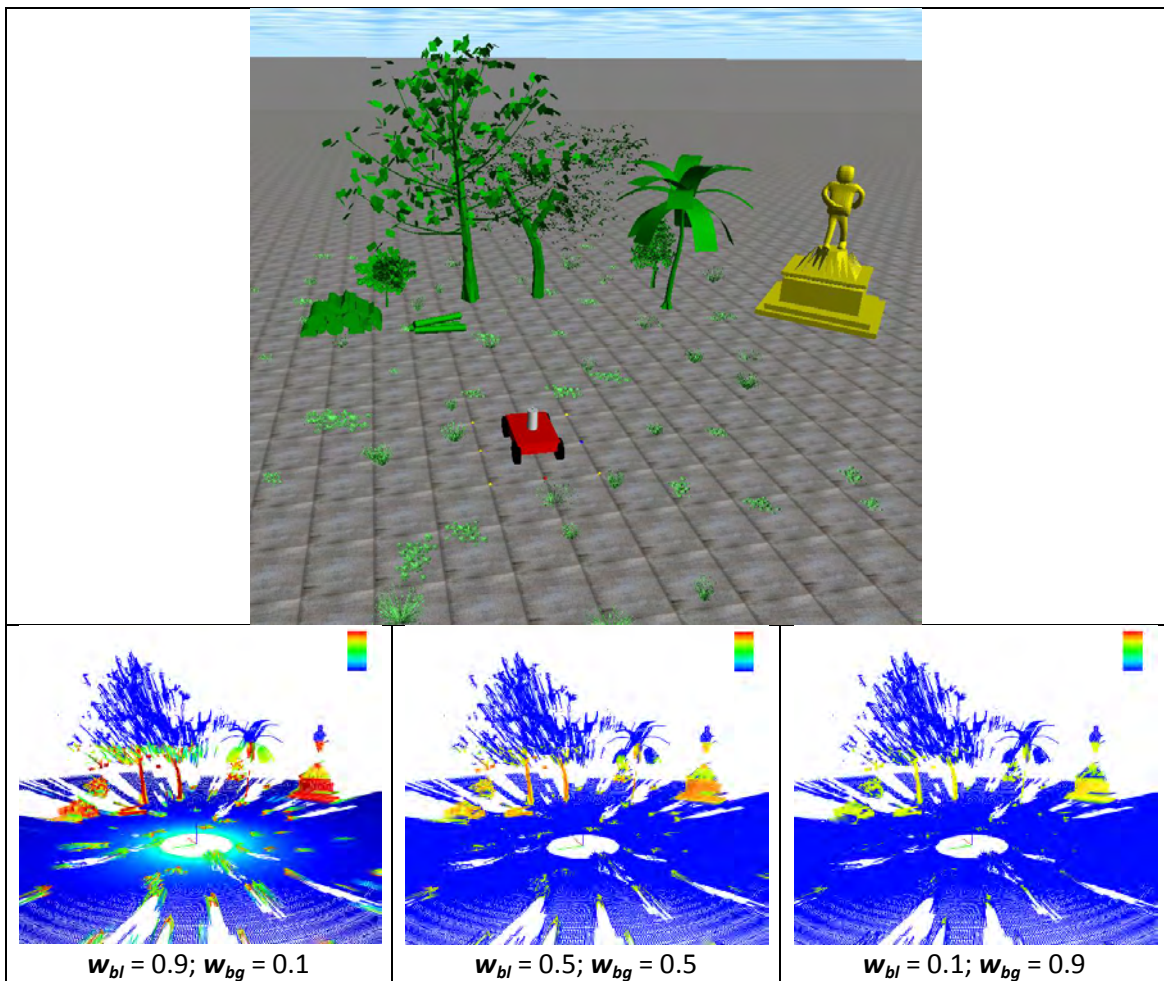


Figura 3-10: (Arriba) Entorno simulado sobre el que se ha realizado la extracción de obstáculos. (Abajo) Efecto de los Parámetros de ajuste Sobre la Extracción de Obstáculos.

La Figura 3-10 muestra el efecto de los parámetros sobre la clasificación de obstáculos. Por ejemplo, se puede observar cómo utilizar w_{bf} como parámetro dominante dará puntuaciones más altas a todos los puntos que se encuentren al alcance del robot, mientras que el incremento de w_{bg} hará que la puntuación sea más homogénea y se reducirá para los puntos que pertenezcan a objetos pequeños, hasta el punto donde dejan de considerarse como obstáculos.

Vale la pena destacar que estos parámetros, al igual que en el caso anterior, influyen principalmente en la creación del mapa de navegabilidad (apartado 4.1). El uso de los parámetros otorga cierta flexibilidad al sistema, la cual es buena para manejar entornos donde pueden existir muchos objetos pequeños (como hierba) y reducir su influencia sobre la navegación.

3.3.2. Análisis de la Calidad del Modelo

La calidad del modelo debe ser analizada punto a punto, ya que no es homogénea debido a que está influenciada por dos factores principales: la resolución del modelo y el ángulo de incidencia del rayo sobre la superficie escaneada. El método propuesto para cuantificar la calidad del modelo considera estos dos factores que, además, pueden ser calculados de una forma ágil y sencilla en cada punto del modelo para establecer una puntuación $D(p)$ entre 0 y 1 (ver ecuación 3.27).

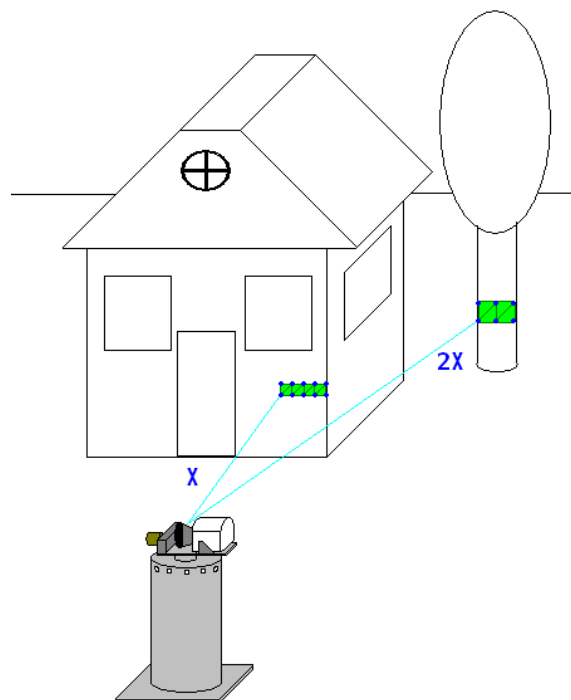


Figura 3-11: Área del triángulo según la distancia

La resolución del modelo es un factor que cuantifica la cantidad de información del mundo representada por cada punto del mismo. Este valor varía proporcionalmente a la distancia de

cada punto al escáner ya que este hace barridos angulares del entorno y la distancia entre los puntos aumenta al escanear objetos más lejanos (ver Figura 3-11 y Figura 3-11).

El área por punto del modelo también depende de las resoluciones verticales y horizontales del escáner (res_p y res_t). La Figura 3-12 muestra la importancia de estos factores, en esta figura se puede ver como en una misma escena capturada dos resoluciones diferentes la densidad de puntos varia ostensiblemente. Estos factores se utilizan indirectamente ya que sirven para calcular un área máxima por punto deseada como se puede ver en la ecuación 3.24.

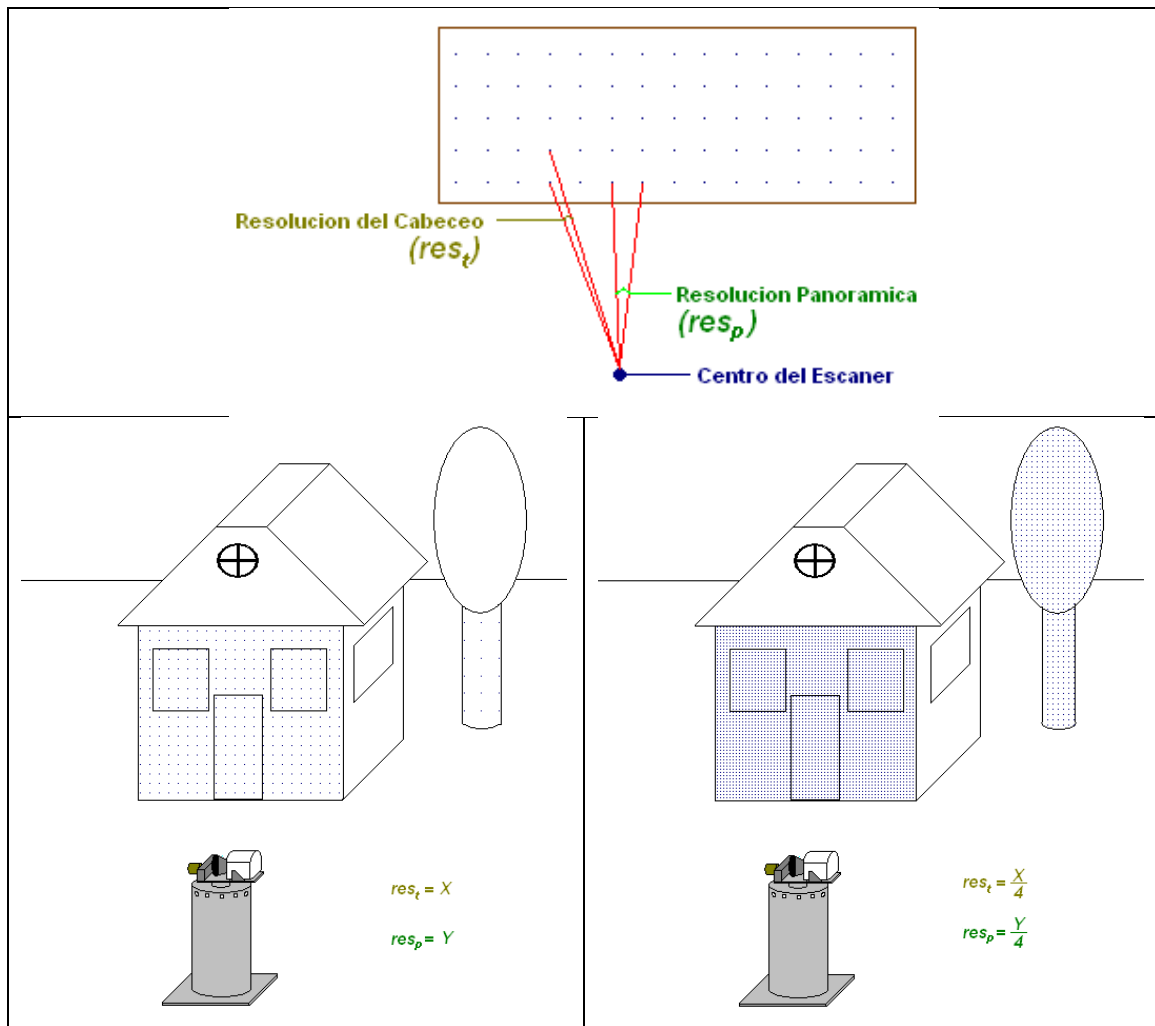


Figura 3-12: (Arriba) Definición de los factores res_p y res_t , (abajo) ejemplo de dos escaneos con diferentes resoluciones del escaner

Para cuantificar esta característica se utiliza el área por punto $P_{Ar}(p)$ definida en la ecuación 3.22. Una vez computada el área, se calcula una puntuación $D_A(p)$ que depende también del área máxima por punto deseada para el escaneo (A_{max}).

$$D_A(p) = \begin{cases} \left(\frac{A_{max}}{P_{Ar}(p)}\right), & P_{Ar}(p) > A_{max} \\ 1, & P_{Ar}(p) \leq A_{max} \end{cases} \quad 3.23$$

El área máxima A_{max} se establece según las necesidades de la exploración. Sin embargo, se puede calcular un valor aproximado para este parámetro utilizando las resoluciones del escáner y definiendo una distancia máxima deseada d_{obj} hasta los objetos a escanear:

$$A_{max} = \frac{d_{obj}^2 (\sin(res_p) * \sin(res_t))}{2} \quad 3.24$$

El ángulo de incidencia del rayo también es utilizado para estimar la calidad del modelo. Este factor permite saber si un objeto dentro del entorno ha sido escaneado de manera frontal. Esto es importante ya que en un objeto escaneado lateralmente se pueden perder múltiples detalles. Igualmente cualquier protuberancia en la superficie puede crear sombras en el modelo y además, la probabilidad de tener lecturas espurias por parte del escáner aumenta ya que el haz láser no se refleja con la misma intensidad sobre superficies frontales que oblicuas.

Para cuantificar este factor se le otorga a cada punto una puntuación $D_I(p)$ que depende directamente del producto escalar entre el vector normal al punto $\vec{N}(p)$ (ecuación 3.25) y el vector del rayo \vec{CP} (un vector desde el centro del escáner c hasta el punto p).

$$D_I(p) = \left(\frac{\vec{CP}}{|\vec{CP}|} \cdot \vec{N}(p) \right) \quad 3.26$$

En la Figura 3-13 se puede apreciar cómo la puntuación es más baja en las zonas que han sido escaneadas de forma más lateral, mientras que las zonas que han sido escaneadas más frontalmente obtienen puntuaciones más altas. Usando estas dos puntuaciones se puede calcular una calificación de calidad para cada punto $D(p)$.

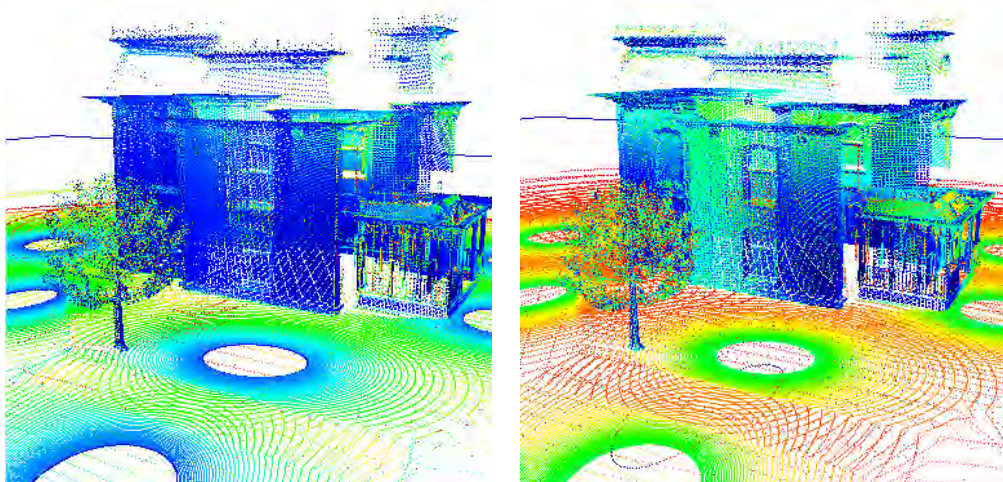


Figura 3-13: Puntuación de calidad según área del punto (izquierda) e incidencia del rayo (derecha). Los colores más fríos indican mejor puntuación de calidad.

A diferencia de los dos casos anteriores, estos parámetros no se utilizan para calcular una media ponderada de las calificaciones de calidad sino que esta puntuación se obtiene utilizando un método basado en la comparación de la hipotenusa de dos triángulos.

Se ha elegido este criterio ya que se ha observado experimentalmente que la ponderación normal penaliza demasiado a las puntuaciones de los puntos pertenecientes a superficies capturadas con un valor angular medio. La ecuación 3.27) muestra el procedimiento para calcular este valor, donde ω_a y ω_n son los pesos que permiten darle mayor relevancia a uno u otro factor.

$$D(p) = \frac{\sqrt{[\omega_a D_A(p)]^2 + [\omega_n D_I(p)]^2}}{\sqrt{(\omega_a^2 + \omega_n^2)}} \quad 3.27$$

Ajuste de los Parámetros

Para ajustar los parámetros hay que tener en cuenta que tal y como se indica en la ecuación 3.27, la calificación final depende de la relación entre las hipotenusas de dos triángulos. El primero de estos triángulos es el triángulo modelo, el cual se toma como referencia para ajustar la calidad.

Para el segundo triángulo las calificaciones de calidad son multiplicadas por los parámetros de calidad y se crea un triángulo cuya hipotenusa siempre es menor o igual a la hipotenusa del triángulo deseado (ya que los valores para las calificaciones de calidad siempre están entre 0 y 1). La calificación final se obtiene calculando la relación entre estas dos hipotenusas (ecuación 3.27).

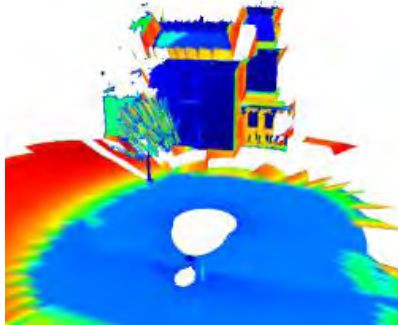
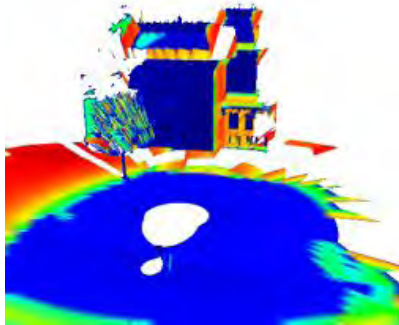
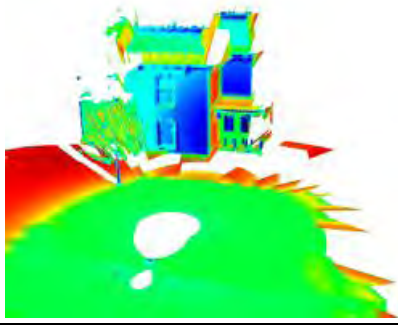
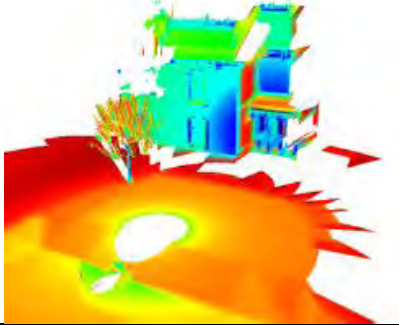
El parámetro w_a representa la influencia que tiene el área del triángulo sobre la puntuación de calidad de cada punto y la de incidencia del rayo. Generalmente el valor de w_a debe ser más alto puesto que es el factor más relevante. Sin embargo, el valor de w_n debe ser suficiente para que los triángulos capturados con un ángulo de incidencia demasiado abierto sean penalizados. En resumen, disminuir el valor de w_a y aumentar el de w_n tendrá como efecto el aumento de la necesidad de capturar los puntos de la forma más frontal posible.

La elección de este tipo de ponderación sobre la utilizada en los casos anteriores se basa principalmente en que los valores de la calificación más relevante tienen mucho más peso en el valor final de la calificación, lo cual es de suma importancia en este caso donde hay un factor que debe ser muy dominante. Además, utilizando este método también es posible cuantificar la penalización a la que son sometidos los puntos que tienen el valor menos relevante y a la vez garantizar que una calificación alta requiera de buenas puntuaciones en los dos criterios de evaluación.

La Tabla 3-1 muestra los efectos de modificar los valores de los parámetros sobre las puntuaciones de calidad del modelo (los colores más fríos muestran mayor calidad). El **recuadro I** de la tabla muestra una configuración de parámetros buena que premia el área como criterio de evaluación de calidad. Esto es deseable ya que solo los puntos tomados con un ángulo muy abierto deben ser penalizados. Sin embargo la disminución de la influencia del ángulo de captura produce un efecto casi de todo o nada como se puede ver en el **recuadro II**,

lo cual tampoco es deseable. Los recuadros III y IV muestran porque es importante que el criterio dominante sea el área por punto.

Tabla 3-1 Efecto de los Parámetros de Ajuste Sobre la Evaluación de Calidad

Exp.	Resultado		Exp.	Resultado	
I			II		
	$w_a = 0.8$	$w_n = 0.2$		$w_a = 0.9$	$w_n = 0.1$
III			IV		
	$w_a = 0.5$	$w_n = 0.5$		$w_a = 0.2$	$w_n = 0.8$

3.3.3.Extracción de Zonas de Interés

Las zonas de interés son aquellas que han sido ocluidas en algún escaneo anterior, por lo que no se dispone de información suficiente sobre ellas (ver Figura 3-14). Sin embargo, es posible que se haya recogido bastante información en los alrededores del plano ocluido lo cual, en el análisis de objetivos, disminuiría las posibilidades de volver a escanear este plano. Por este motivo se ha incorporado al algoritmo de exploración un criterio que incluye los planos ocluidos dentro de la evaluación de los objetivos con el fin de garantizar que el modelo sea lo más completo posible.

Para la identificación de estos planos ocluidos se utilizan los triángulos descartados en el proceso de triangulación, ya que estos indican los espacios sin cubrir del modelo. En el caso de los triángulos o grupos de triángulos (triángulos que comparten lados entre sí) que superen un área predefinida se calcula un vector normal que apunta al centro del triángulo (o grupo) y cuya magnitud es proporcional al área ocluida.

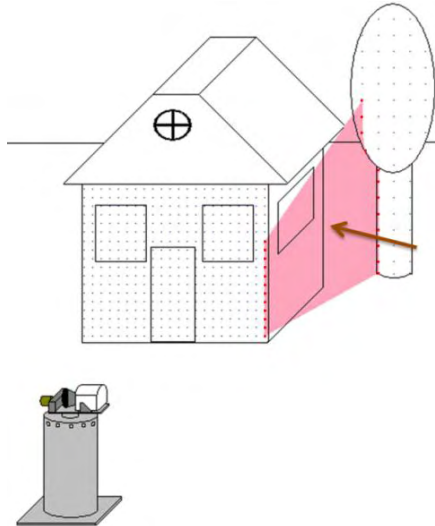


Figura 3-14: Ejemplo gráfico de un plano ocluido

Seguidamente, se calcula un punto central para el área ocluida p_{co} promediando las componentes x , y , z de los puntos que conforman el plano ocluido (los puntos de los triángulos descartados que conforman cada plano, en rojo en la Figura 3-14) de la siguiente forma:

$$p_{co}(x, y, z) = \frac{1}{n} \sum_{i \in \vartheta} p_i(x, y, z) \quad 3.28$$

donde n es el número de puntos que conforman la zona ocluida, p_i es cada uno de estos puntos y ϑ es el conjunto de los mismos. Una vez hecho esto, se calcula el vector normal a la zona ocluida utilizando la siguiente ecuación:

$$\vec{V} = \frac{\sum_{i \in \omega} A_{T_i} \vec{N}_{T_i}}{n_t A_{max}} \quad 3.29$$

donde ω es el conjunto de triángulos que conforman la zona ocluida, \vec{V} es el vector normal a la misma, A_T es el área de cada triángulo y \vec{N}_T es el vector normal de los mismos. Este vector y el punto intermedio de la zona se almacenan en una lista que luego será utilizada para estimar la cantidad de zonas de este tipo cubiertas desde cada punto del modelo, en la evaluación de candidatos (Capítulo IV). La Figura 3-15 muestra los planos ocluidos extraídos de un escaneo.

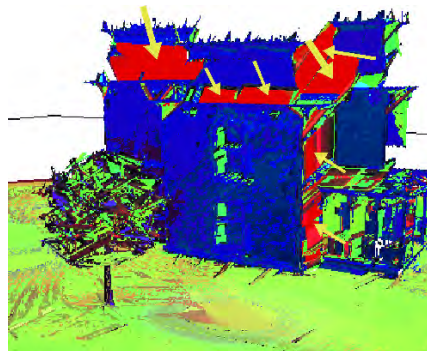


Figura 3-15: En rojo los planos ocluidos extraídos de un escaneo. Las flechas apuntan al punto central de la zona ocluida.

3.4. Conclusiones del Capítulo

En este capítulo se ha presentado la metodología utilizada para el análisis y extracción de información sobre las características del entorno. Al principio del capítulo se ha presentado la metodología utilizada para la creación de la malla y para el alineamiento de las nubes de puntos, procesos sobre los que se fundamenta el análisis tridimensional y la creación del mapa de información 2D. La creación de la malla tridimensional facilita la extracción de información del entorno al simplificar los cálculos para la detección de superficies y la evaluación de los puntos del entorno. El proceso de alineamiento o SLAM6D está basado en el algoritmo ICP y soportado sobre el sistema de posicionamiento global del robot. Esta metodología además de rápida y efectiva, refina la alineación del modelo al encontrar lazos cerrados.

También se ha analizado cómo se puede extraer información relevante para la navegación de un robot móvil a partir de la información geométrica de una malla tridimensional. Asimismo, se ha mostrado cómo aprovechando las características de los triángulos que componen la malla es posible evaluar la calidad de la información en cada punto del modelo y aún más, extraer información sobre zonas que son de interés especial para la adquisición de nueva información. Adicionalmente, se ha estudiado cómo los parámetros utilizados en este apartado pueden ser utilizados para ajustar el comportamiento del proceso según el tipo de terreno donde se lleve a cabo la exploración.

El resultado de este proceso es la obtención de un modelo tridimensional correctamente alineado con los escaneos anteriores, al que además se le han añadido puntuaciones que permiten establecer la calidad del modelo y las posibilidades de pertenecer a un obstáculo o a una superficie de navegación segura, para cada punto. Adicionalmente, se ha generado una lista con las zonas de interés para futuros escaneos extraídas en el proceso. Las puntuaciones obtenidas durante el análisis tridimensional permiten evaluar las posiciones candidatas del robot para el proceso de exploración, que se describirá en el Capítulo 4.

Capítulo 4.

EVALUACIÓN DE CANDIDATOS

La evaluación de posiciones candidatas utilizando información 3D es un proceso que puede consumir muchos recursos y tiempo. Sin embargo, los entornos exteriores suelen estar compuestos de un único nivel de suelo y la planificación del movimiento puede plantearse como una planificación de movimiento bidimensional donde es necesario tomar en consideración la capacidad del robot y el grado de dificultad para atravesar un espacio, no únicamente la ocupación del mismo.

Al igual que la planificación del movimiento se puede realizar en dos dimensiones haciendo estas consideraciones, también se puede estimar la cantidad y calidad de la información a percibir desde una posición determinada usando una rejilla de información.

Con este propósito se propone utilizar un mapa de información 2D en el cual cada celda almacena toda la información 3D capturada sobre un área del entorno. Esto supone que cada vez que se captura una nube de puntos 3D, se proyecta la información de cada punto junto con las puntuaciones obtenidas para el mismo (Capítulo 3), sobre la celda del mapa de información que corresponda la posición de ese punto en el entorno. Esto permite reducir el procesamiento de datos ya que facilita la estimación de zonas cobertura y la extracción de las características geométricas del terreno.

4.1. Generación Del Mapa De Información 2D

4.1.1. Extracción del Mapa de Navegabilidad

El mapa de navegabilidad es una representación del mapa de información 2D que se propone para facilitar el cálculo de trayectorias de navegación en entornos exteriores, tomando como principio el concepto de navegabilidad introducido en (Hertzberg, Lingemann, Lörken, Nüchter, & Stiene, 2008). En apariencia es un mapa muy similar a los mapas de celdas de ocupación utilizados en los entornos 2D. Sin embargo, en este caso, los valores contenidos en cada celda representan la probabilidad de que esta pueda ser o no atravesada. La Figura 4-1 muestra un ejemplo de esta representación.

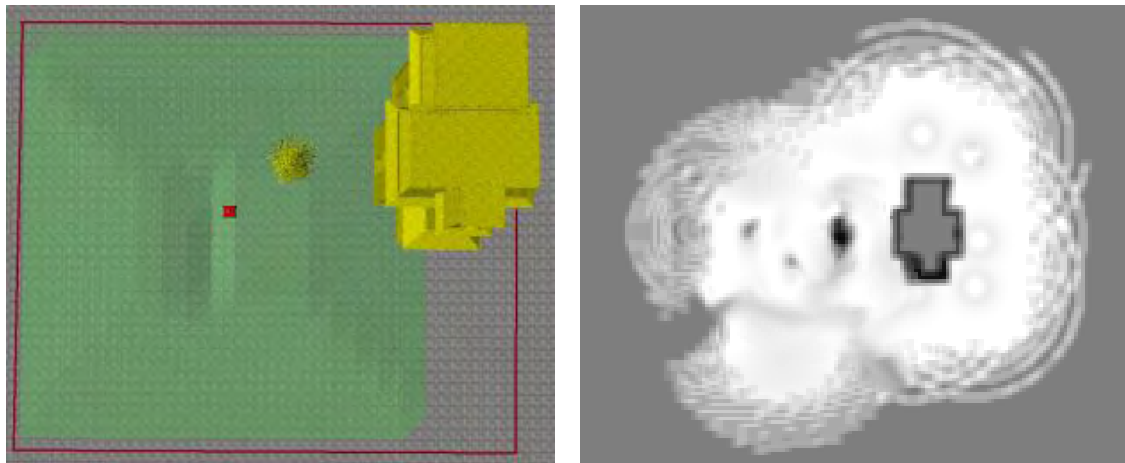


Figura 4-1: Entorno simulado (izq) y Mapa de Navegabilidad extraído del mismo (der)

Utilizando la información extraída sobre las superficies navegables y los obstáculos del entorno para cada punto (ver sección 3.3.1), se calcula una puntuación para cada celda que representa la navegabilidad de la misma. Cada vez que se añade a la celda un punto nuevo con una calificación alta para $F(p)$ (ecuación 3.16), la probabilidad de que la celda sea atravesable aumenta, ocasionando que la puntuación otorgada a la celda aumente igualmente. Si por el contrario el punto tiene una calificación alta para $B(p)$ (ecuación 3.21), la navegabilidad de la celda decrece.

Esta puntuación de navegabilidad de cada celda c , $C_{nc}(c)$ puede tomar valores entre 0 y 1 (donde 0 implica que la celda no se puede cruzar y 1 que es completamente atravesable). Para facilitar la diferenciación entre celdas navegables y no navegables solo se consideran los puntos con valores de $F(p)$ mayores que un umbral ε_f y los puntos con valores para $B(p)$ que superen el umbral ε_o en el cálculo de $C_{nc}(c)$.

Los puntos que superan el umbral ε_f conforman el conjunto φ y sus valores se promedian con el fin de obtener una puntuación representativa del conjunto de puntos navegables $F_c(c)$ en la celda c , el cual se calcula con la ecuación 4.1.

$$F_c(c) = \frac{1}{n_{pf}} \sum_{i \in \varphi} F(i) \quad 4.1$$

donde n_{pf} es el número de puntos que conforman el conjunto φ .

Se sigue un proceso similar con los puntos que superan el umbral ε_o los cuales conforman el conjunto α y cuyos valores se usan para obtener una puntuación representativa del conjunto de puntos obstaculizantes $B_c(c)$ en la celda c , según muestra la ecuación 4.2.

$$B_c(c) = \frac{1}{n_{po}} \sum_{i \in \alpha} B(i) \quad 4.2$$

Aquí, n_{po} es el número de puntos que conforman el conjunto α .

Una vez obtenidos estos valores se calcula una relación r_{fo} entre la cantidad de puntos del conjunto φ y los del conjunto α de la siguiente forma:

$$r_{fo} = \frac{n_{pf}}{n_{pf} + n_{po}} \quad 4.3$$

Entonces la puntuación de navegabilidad $C_{nc}(c)$ se calcula ponderando $F_c(c)$ y $B_c(c)$ mediante:

$$C_{nc}(c) = 0.5 + \left(r_{fo} \frac{F_c(c)}{2} \right) - \left((1 - r_{fo}) \frac{B_c(c)}{2} \right) \quad 4.4$$

En resumen, todas las celdas del mapa son inicializadas en un valor intermedio del intervalo de ponderación, es decir 0.5, y a medida que se añaden puntos con un marcador alto para la puntuación de zonas seguras de navegación, este valor aumenta. Por el contrario cada punto con un marcador alto en la extracción de obstáculos disminuye esta puntuación. La cantidad de variación de $C_{nc}(c)$ depende del número de puntos con unas y otras características.

4.1.2. Evaluación de la Calidad del Modelo

El mapa de información 2D también se utiliza para evaluar la calidad del modelo. Esta evaluación es necesaria por dos motivos principales: la evaluación de la mejora de calidad desde las posiciones candidatas y la evaluación de los criterios de parada de la exploración (el proceso no debe detenerse si no se ha alcanzado un nivel mínimo de calidad global del modelo).

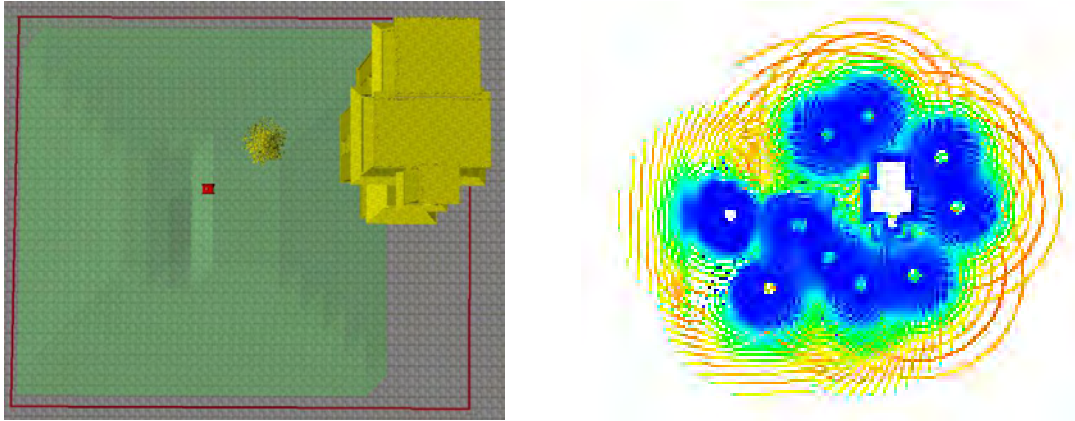


Figura 4-2: Mapa de Calidad (der) extraído de un entorno simulado sencillo (izq)

Para evaluar la calidad del modelo se utiliza un mapa de calidad que es una representación que se crea de forma similar al mapa de navegabilidad. Sin embargo, en este caso cada celda almacena el promedio del valor de calidad de los puntos proyectados sobre ésta, $Q_a(c)$. Esto se hace mediante:

$$Q_a(c) = \frac{1}{np(c)} \sum_{p \in c} D(p) \quad 4.5$$

donde $np(c)$ es el número de puntos en cada celda y $D(p)$ es el valor por punto calculado en la ecuación 3.27. La Figura 4-2 muestra un ejemplo de un mapa de calidad extraído de un entorno simple.

4.2. Evaluación de Posiciones Candidatas

Sobre el mapa de información se crea una serie de posiciones a evaluar, las cuales son generadas aleatoriamente sobre las celdas con un coeficiente de navegabilidad superior a 0,7 siempre separadas entre sí más de tres metros, hasta un máximo de 1/50 de las celdas con $C_{nc}(c) > 0,7$.

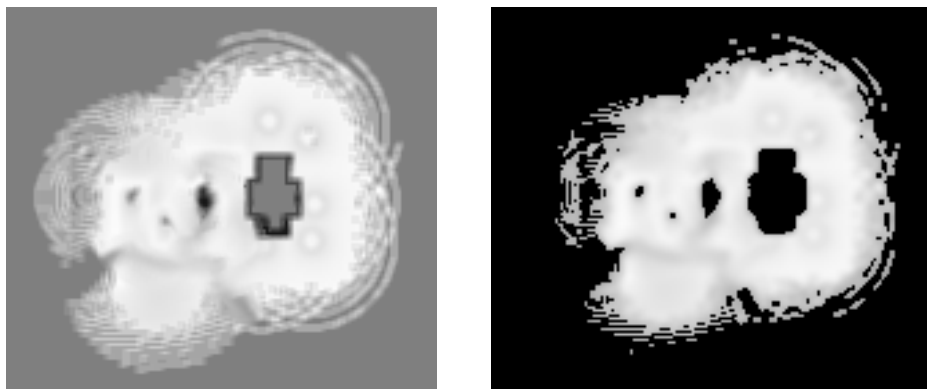


Figura 4-3: Mapa de Navegabilidad de un entorno (der). Celdas excluidas de la generación de candidatos (izq).

Cada celda seleccionada como candidata, es evaluada utilizando la ecuación 2.3. Para esta evaluación se computan los datos calculados en los apartados del 4.2.1 al 4.2.4.

4.2.1. Estimación del Coste de la Trayectoria

Este proceso se hace calculando una trayectoria desde la posición actual del robot hasta cada una de las posiciones candidatas sobre el mapa de navegabilidad, que cruce únicamente las celdas con un valor de navegabilidad $C_{nc}(c)$ mayor que 0,7. Este proceso se realiza antes que los demás ya que si no se encuentra ninguna trayectoria hasta la posición candidata, esta es descartada.

La estimación de la trayectoria entre la posición actual del sistema y cada posición candidata se realiza creando un “árbol de navegación”, que permite estimar el costo de alcanzar cualquier celda navegable en el mapa. El árbol de navegación es una estructura de información en la cual cada celda del mapa de información 2D es un nodo del árbol. La estructura de información se va construyendo a partir de la celda de la posición del robot, expandiendo los cuatro nodos adyacentes a los nodos expandidos en la iteración anterior.

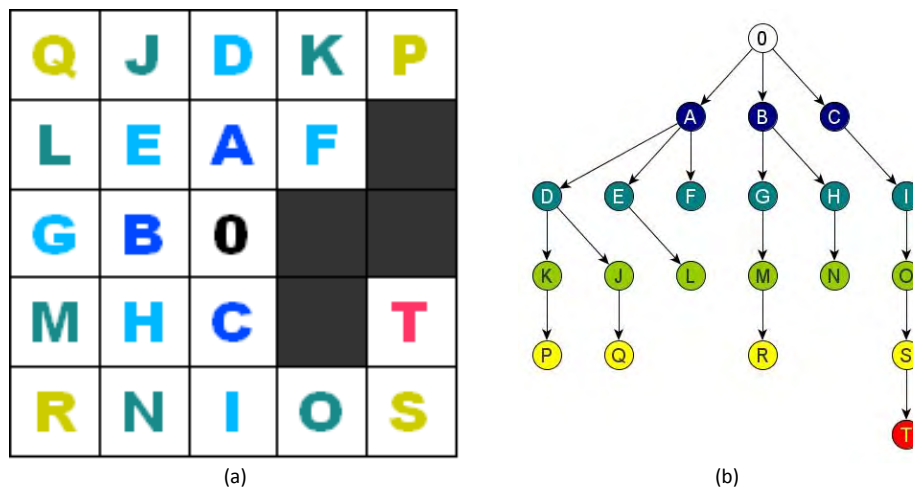


Figura 4-4: (a) Ejemplo de árbol de navegación. Mapa de Información 2D. (b) Árbol de Navegación creado a partir del mapa. La celda 0 representa la posición del robot en ambos casos.

Ya que se puede llegar a una misma celda siguiendo múltiples trayectorias, es necesario introducir una función de coste para escoger la ruta más efectiva. Esta función de coste tiene la siguiente forma:

$$C(c) = \sum_{t \in \tau(a,c)} d_{ec}(c, c-1) \cdot \frac{1}{C_{nc}(c)} \tag{4.6}$$

donde $\tau(a,c)$ es el conjunto de celdas en la trayectoria desde la celda actual a hasta cada posición evaluada c y $d_{ec}(c,c-1)$ es la distancia entre cada celda y la siguiente en la trayectoria.

Cada vez que se expande un nuevo nodo, si uno de sus nodos hijos es ya hijo de un nodo previamente expandido, se calcula la ruta con el coste más bajo usando la ecuación 4.6. Si el nuevo padre tiene un coste más bajo, el hijo se retira de la lista de hijos del nodo padre anterior y se deja el valor de $C(n)$ más bajo. Si el nodo hijo ha sido expandido con anterioridad, se debe recalcular el valor de $C(n)$ para todos los descendientes del mismo.

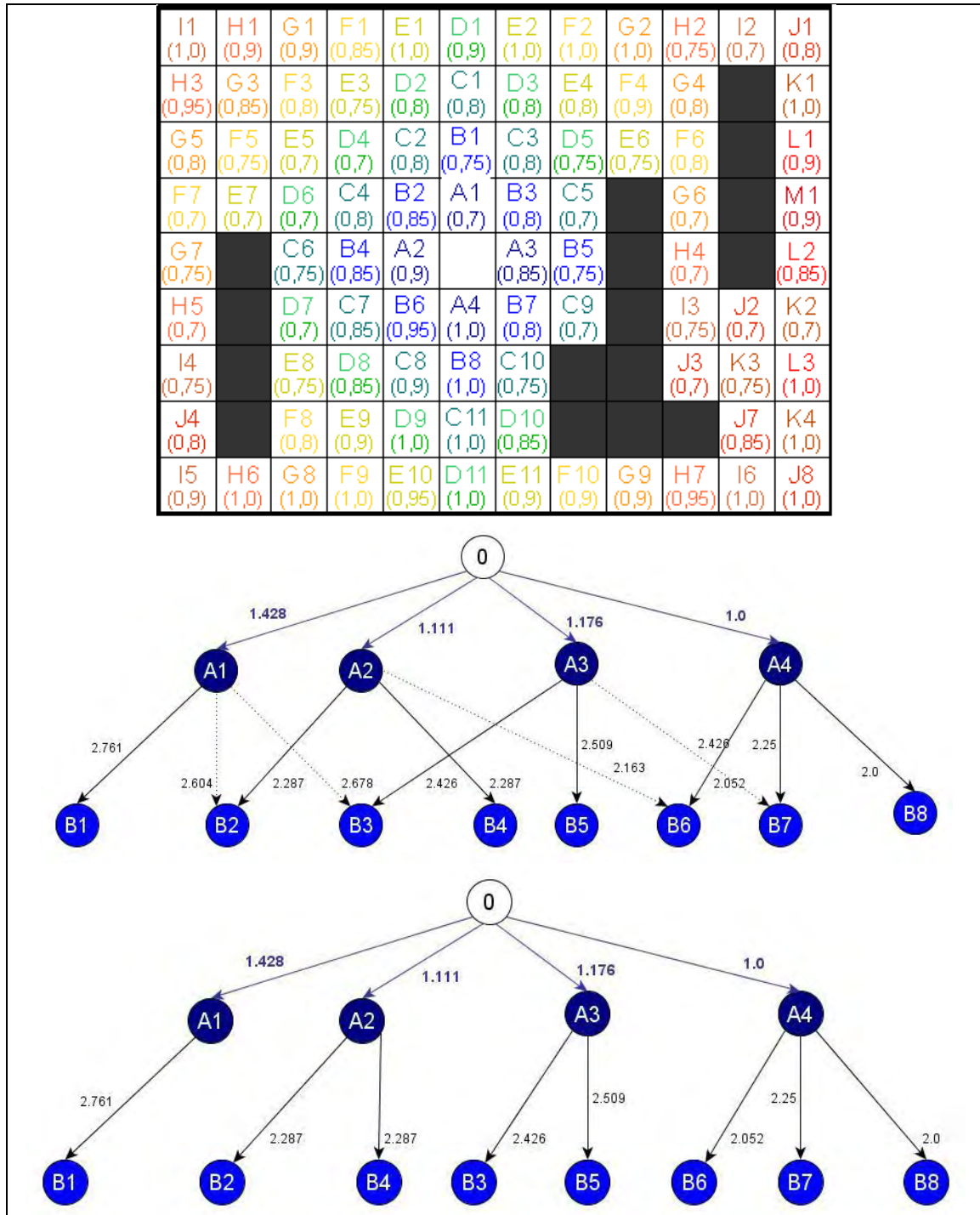


Figura 4-5: Creación del árbol de costos de navegación.

La Figura 4-5 muestra el proceso de expansión de los nodos del árbol de navegación y la Figura 4-6 muestra un mapa de las celdas alcanzables con la puntuación de navegación para cada una y un ejemplo de la trayectoria estimada a un set de posiciones candidatas.

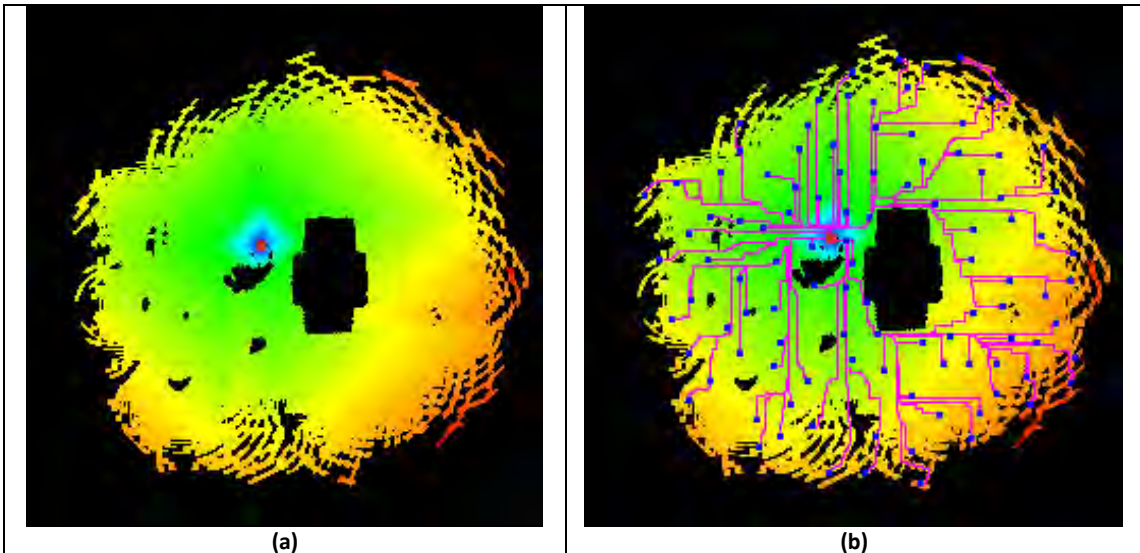


Figura 4-6: (a) Representación del costo de la navegación, los colores más cálidos representan mayor costo de navegación, la posición del robot se encuentra marcada en magenta. (b) Ejemplo de la trayectoria estimada a un set de posiciones candidatas.

Las posiciones candidatas para las que no se encuentra ruta valida son descartadas. Una vez creado el árbol de navegación, solo hace falta acceder al nodo que representa la celda objetivo y tomar el costo almacenado en este nodo como estimación del costo de la trayectoria. Sin embargo, la extracción de los puntos que conforman la trayectoria es sencilla ya que basta con hacer una búsqueda inversa, partiendo de la posición candidata y retrocediendo hasta el nodo de origen a través de los padres de cada nodo.

4.2.2. Ganancia Esperada de Información

La ganancia esperada de información se evalúa calculando el área inexplorada vista desde cada una de las celdas. Con este fin se utiliza el mapa de navegabilidad y se cuentan las celdas no exploradas que estén dentro del área de cobertura del escáner, menos las celdas ocluidas o que están fuera del área de trabajo.

$$A(c) = \frac{Ac(C_{se} - n_o)}{A_{ce}} \quad 4.7$$

En esta expresión, Ac es el área representada por cada celda, C_{se} el número de celdas sin explorar vistas desde cada posición c , A_{ce} es el área de cobertura total del escáner y n_o es el número de celdas ocluidas desde c . Para el cálculo de dichas celdas se traza una línea imaginaria desde la posición evaluada, hasta cada una de las celdas sin explorar. Si el valor de navegabilidad ($C_{nc}(c)$, ver ecuación 4.4) de cualquiera de las celdas por las que atraviesa esta línea es menor que un umbral, la posición se considera ocluida.

4.2.3. Ganancia Esperada de Calidad del Modelo

En cada posición candidata se calcula la calidad esperada para cada celda en el rango del escáner, usando los mismos criterios que para la evaluación de la calidad del modelo, en otras palabras el área por punto esperada y la incidencia del rayo sobre los planos del entorno.

El primer factor se obtiene usando la resolución del escaneo y la distancia desde la celda analizada c hasta cada una de las celdas que están a su alcance e para estimar el tamaño de los triángulos, usando la ecuación 4.8:

$$E_A(c, e) = \frac{A_{max}}{r(c, e)^2(\sin(res_p) \sin(res_t))} \quad 4.8$$

En esta expresión $r(c, e)$ es la distancia desde la posición evaluada hasta una celda e que está en su radio de cobertura, res_p es la resolución panorámica del escáner y res_t es la resolución de cabeceo del escaneo tridimensional en términos angulares. A_{max} es el área máxima deseada para cada triángulo.

Para calcular la mejora en la calidad por la incidencia del rayo sobre los planos, se calcula la proyección del rayo teórico sobre la normal local, generando un vector normalizado sobre cada punto desde la posición candidata. Cuanta más alta es la proyección, más frontalmente se suponen capturada la superficie desde cada posición. Este valor se promedia para cada celda del mapa de calidad, de forma que es equiparable con los otros dos valores usados para esta estimación. La ecuación siguiente muestra la forma de calcular este valor para cada celda.

$$E_I(e) = \frac{1}{np(e)} \sum_{k \in e} \left(\frac{\vec{R}_k}{|\vec{R}_k|} \cdot \vec{N}_k \right) \quad 4.9$$

\vec{R}_k es un vector desde la posición evaluada c hasta cada punto (k) de la celda e , \vec{N}_k es un vector unitario normal a cada uno de los puntos y $np(e)$ es el número de puntos de la celda e . De esta manera la mejora en la calidad $Q(c)$ para cada una de las posiciones evaluadas se calcula usando,

$$Q(c) = \sum_{e \in \sigma} \left(\frac{\sqrt{[\omega_a E_A(c, e)]^2 + [\omega_n E_I(e)]^2}}{\sqrt{(\omega_a^2 + \omega_n^2)}} - Q_a(e) \right) \quad 4.10$$

donde σ es el conjunto de las celdas en el rango del escáner desde la posición evaluada c , y ω_a y ω_n son los valores utilizados para el análisis de la calidad del modelo.

4.2.4. Análisis de la cobertura de Zonas de Interés

Las zonas de interés son los planos ocluidos del modelo, extraídos en el proceso de triangulación. Estos planos son almacenados con su vector normal. Se crea un vector desde cada posición evaluada hasta el centro de cada plano ocluido que esté en el rango de cobertura del escáner, y utilizando la ecuación 4.11 se obtiene una puntuación para estimar la cobertura de estas zonas,

$$O_n(c) = \sum_{i \in \beta} |\vec{r}_{ci} \cdot \vec{V}_i| \quad 4.11$$

Aquí β representa el conjunto de planos ocluidos visibles desde la posición candidata c , \vec{r}_{ci} es un vector desde la celda evaluada hasta el centro del plano ocluido y \vec{V} es el vector normal a cada plano ocluido calculado en la ecuación 3.29. Cada plano ocluido escaneado es removido de la lista.

4.3. Selección de NBV y Selección del criterio de Parada

A partir de los valores obtenidos en los pasos anteriores, se procede a evaluar cada posición candidata, de acuerdo con la ecuación 2.3 y los parámetros de exploración del experimento (ver Sección 6.1), de modo que introduciendo los valores obtenidos en las ecuaciones 4.6, 4.7, 4.10 y 4.11 en la ecuación 2.3 tenemos una puntuación de utilidad para la celda c .

Las ecuaciones, 4.7, 4.10 y 4.11 generalmente darán valores comprendidos entre 0 y 1. Además, los parámetros de exploración (w_A , w_O , w_D y w_C) utilizados en la ecuación 2.3 son valores comprendidos entre 0 y 1.

Teniendo estas consideraciones en cuenta se puede establecer un criterio de parada usando igualmente la ecuación 2.3, suponiendo la distancia máxima que se quiere que el robot recorra para un punto de utilidad media. Este criterio, C_p , es la magnitud mínima que debe tener el valor $u(c)$ de la posición ganadora y se puede calcular usando la siguiente ecuación:

$$C_p = \frac{0.5}{w_c \cdot d_{max}} \quad (4.12)$$

donde w_c es el parámetro que pondera el peso del costo de la trayectoria en la ecuación, y d_{max} es la distancia máxima que se quiere que el robot recorra hasta un objetivo de utilidad media.

4.4. Conclusiones del Capítulo

En este capítulo se ha presentado la metodología utilizada para evaluar las posiciones candidatas a ser elegidas como la siguiente vista dentro del proceso de reconstrucción del entorno. Primero se ha presentado el mapa de información 2D sobre el cual se proyectan las puntuaciones extraídas en el proceso de análisis tridimensional. A partir de este mapa se pueden obtener dos representaciones del entorno: el mapa de navegabilidad y el mapa de calidad del modelo.

Estas representaciones permiten agilizar el cálculo de los criterios utilizados para evaluar las posiciones candidatas a ser el siguiente punto de escaneo. El primer paso para la evaluación de las posiciones candidatas es la evaluación de la dificultad de la trayectoria, el siguiente paso es la estimación de la ganancia de información y la calidad del modelo, y por último se estima la cobertura de las zonas de interés elegidas.

Como se ha visto a lo largo de este capítulo, es importante notar que la implementación del mapa de información 2D no significa una reducción de la información tridimensional capturada, sino una simplificación que facilita y agiliza los cálculos para la evaluación de las distintas vistas. Por ejemplo, el uso del mapa de navegabilidad facilita enormemente los cálculos de estimación del costo de la trayectoria que de otra forma debería realizarse sobre el modelo 3D, con la consecuente complejidad.

Capítulo 5.

MARCO EXPERIMENTAL

Este capítulo está dedicado a la descripción pormenorizada del diseño de una plataforma todoterreno para la reconstrucción tridimensional de entornos de exterior. La plataforma está compuesta por un cuerpo robusto, con un sistema de tracción de seis motores independientes y un sistema informático en el cual se incorporan los algoritmos propuestos en la presente investigación y, especialmente, el algoritmo de exploración basado en la información tridimensional del entorno. Este algoritmo tiene como propósito reconstruir de forma autónoma el mapa tridimensional del entorno sobre el que se desplaza el robot, tal como se ha descrito en los capítulos precedentes.

El desarrollo y la construcción del robot pueden separarse en dos partes fundamentales. Por un lado, se ha implementado una plataforma móvil todoterreno, ideada para su utilización en la mayor cantidad posible de aplicaciones robóticas en exteriores. Por otro lado, se ha desarrollado un sistema de reconstrucción 3D que es un dispositivo independiente y permite la captura de información tridimensional del entorno circundante. La combinación de estos dos elementos da como resultado un sistema que hace posible realizar mediciones tridimensionales de entornos complejos de forma rápida, y con un esfuerzo mucho menor del que se requiere con los sistemas estacionarios. En las siguientes subsecciones se describen sus principales características.

5.1. Motivación

La motivación para la construcción de un robot de propio viene de la escasez de plataformas robotizadas todoterreno de coste moderado, aptas para la investigación y el desarrollo de tareas en entornos exteriores. El propósito es lograr una plataforma asequible y altamente

configurable que pueda ser utilizada para una amplia gama de tareas en entornos exteriores, como:

- Misiones de búsqueda, rescate en situaciones de alto riesgo para el hombre como incendios, derrumbes, desactivación de explosivos, trabajo en presencia de sustancias peligrosas o misiones policiales.
- Labores en entornos de difícil accesibilidad para el hombre como cuevas, túneles, galerías subterráneas o tuberías.
- Investigación en diferentes áreas, por ejemplo en algoritmos de navegación, reconstrucción 3D en exteriores, cooperación entre robots, o integración de equipos diversos de sensorización.

El desarrollo y construcción del robot fue realizado en el marco del proyecto ROBAUCO (Robots Autónomos Cooperantes FIT-170200-2007-1), financiado por el Ministerio de Industria Comercio y Turismo, cuyo objetivo principal ha sido el desarrollo de tecnologías para para el desempeño de tareas complejas que exijan un elevado grado de autonomía y capacidad de colaboración en presencia de seres humanos.

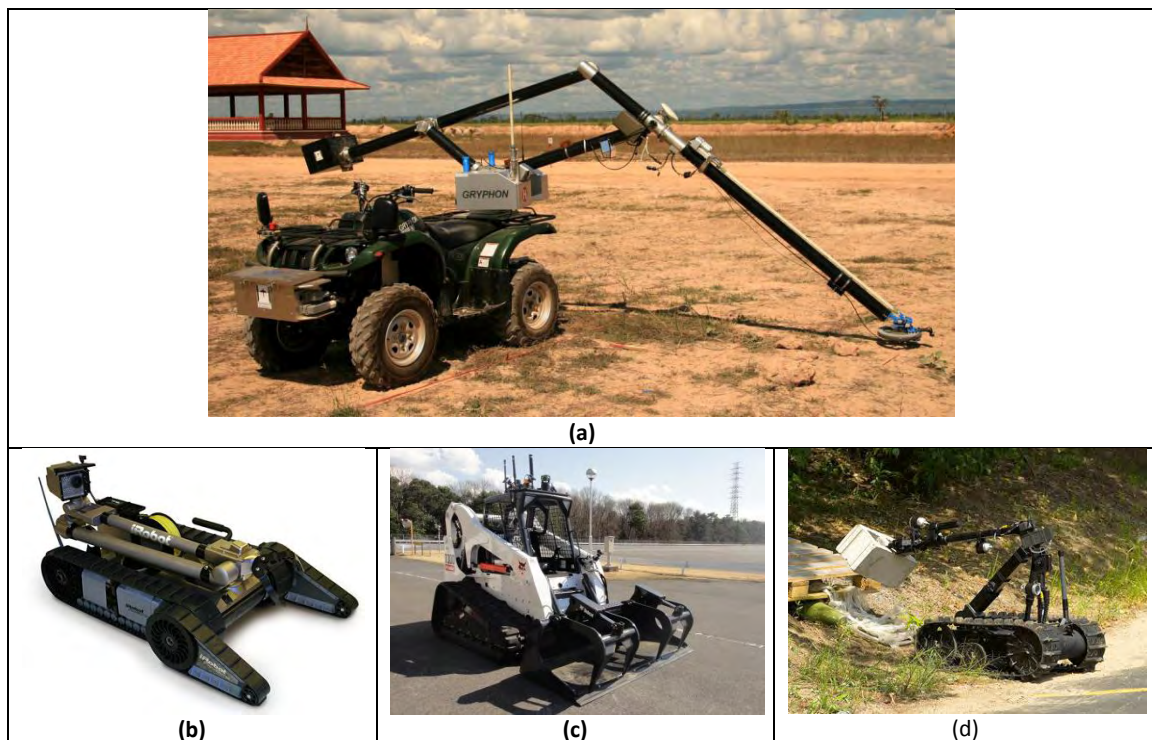


Figura 5-1: Robots para aplicaciones de exteriores. Arriba para tareas de desminado (a) Robot “Gryphon” (Freese, y otros, 2007), Abajo robots utilizados durante el desastre de Fukushima (b) Robot “Packbot” de iRobot, (c) Robot “Bobcat” de Qinetiq, (d) Robot “Talon” de Qinetiq (Murphy R. , 2011).

Existen robots comerciales, caracterizados por lo general por un coste elevado y un carácter altamente especializado (ver Figura 5-1). Tenemos, por ejemplo, robots desactivación de explosivos o desminado (Debenest, Fukushima, Tojo, & Hirose, 2005), (Freese, y otros, 2007), robots para aplicaciones militares o que manejan y transportan sustancias peligrosas o radioactivas, como los utilizados durante el accidente nuclear de Fukushima (Murphy R. , 2011).

También existen algunas plataformas más genéricas desarrolladas con propósitos de investigación, como por ejemplo los vehículos autónomos realizados para el DARPA Challenge en sus dos versiones (2005 y 2007), o el Smarter (ver Figura 5-2). Sin embargo no son comerciales y su precio es muy elevado.

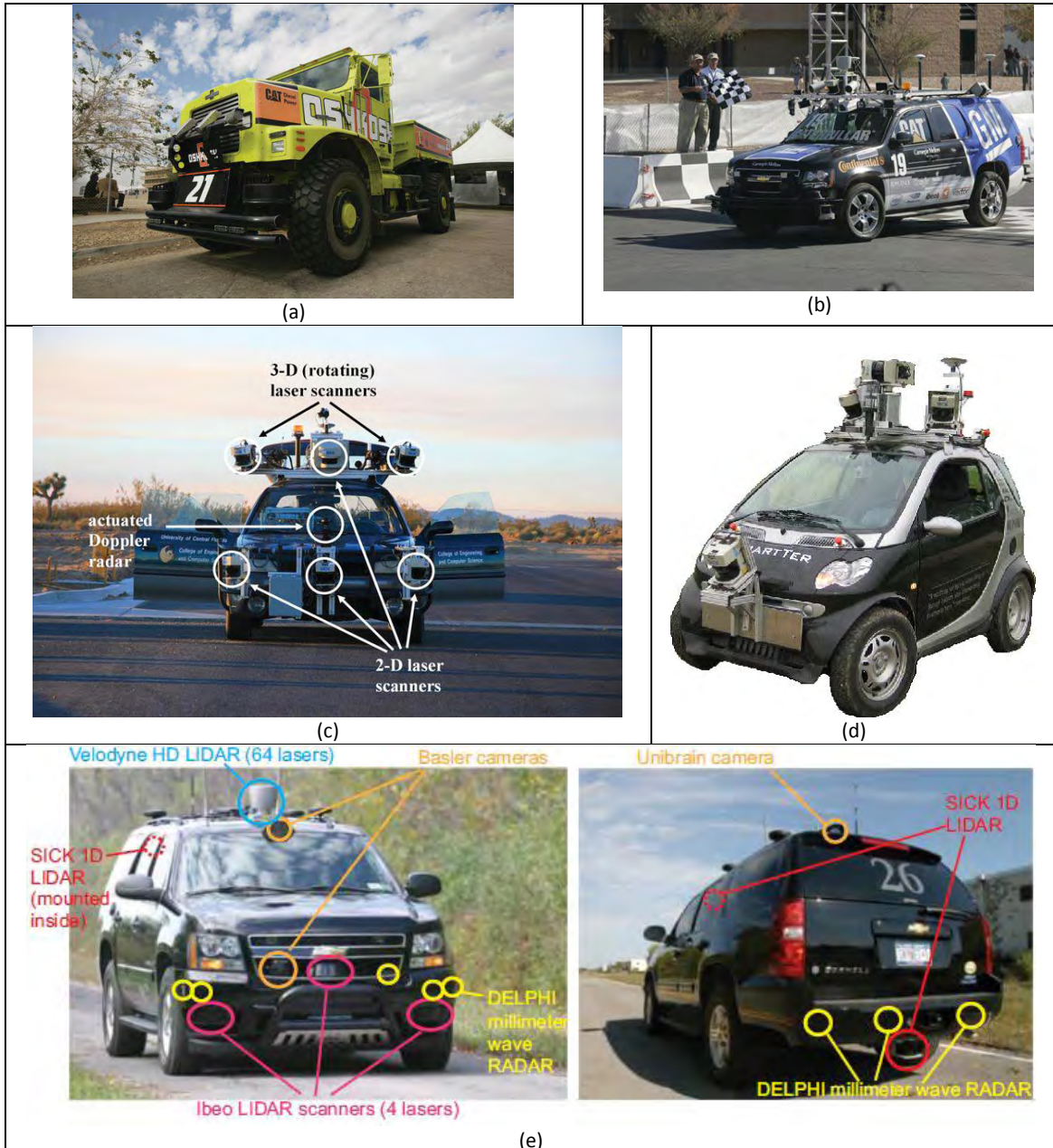


Figura 5-2: Robots Competidores en el Darpa Urban Challenge. (a) Robot TerraMax (Chen, y otros, 2008), (b) Robot Boss (Urmsion, y otros, 2008), (c) Robot Knight Rider (Patz, Papelis, Pillat, Stein, & Harper, 2008), (d) Robot Smarter (Pfaff, Triebel, Stachniss, Lamon, Burgard, & Siegwart, 2007) (e) Robot SkyNet (Miller, y otros, 2008)

También se destacan algunas plataformas comerciales como el Pioneer 3-AT o el robot Kurt 3D las cuales están muy limitadas en su movilidad por sus características físicas. Además, en el caso del Pioneer 3-AT su costo ronda los 25,000€. Finalmente hay algunas plataformas de investigación comerciales dotadas de gran movilidad en exteriores, como las plataformas Seekur y Seekur Jr de Adept Mobile Robots, cuyo precio ronda los 80,000€ y los 50,000€ respectivamente. En la Figura 5-3 se pueden ver estos robots.

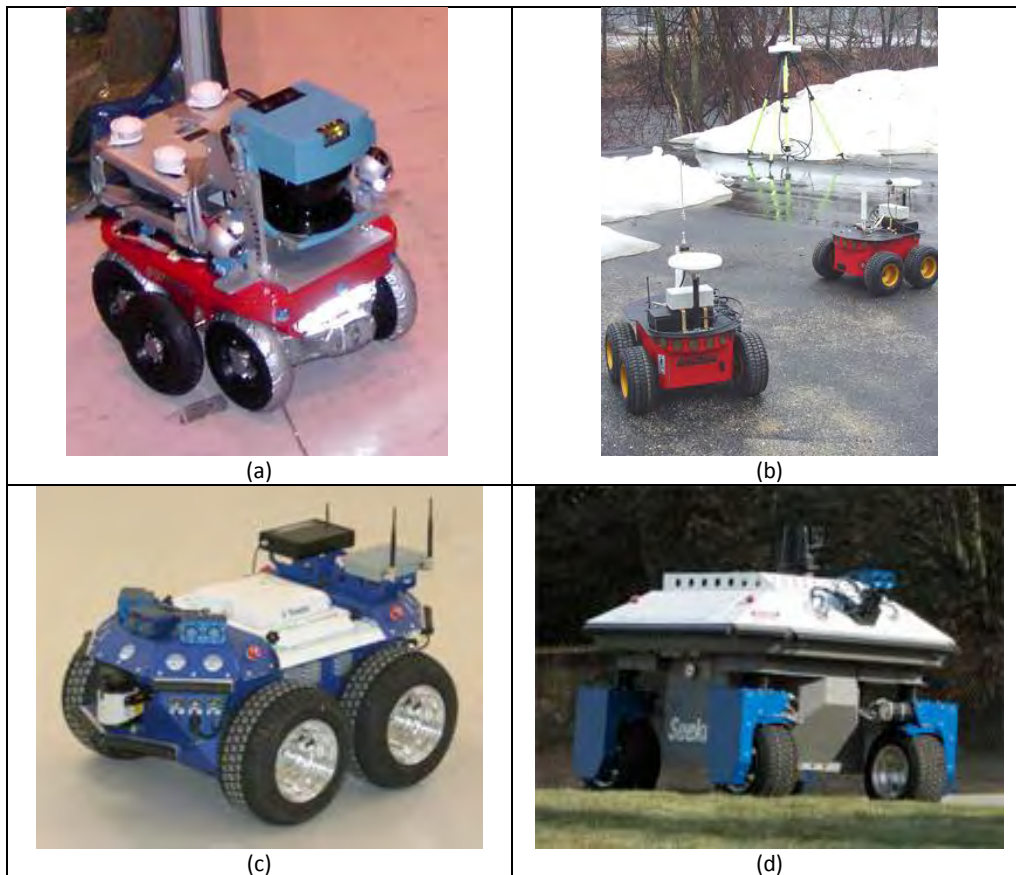


Figura 5-3: (a) Robot Kurt 3D (Nüchter, y otros, 2005); (b) Pioneer 3-AT, (c) Seekur Jr, (d) Seekur (Adept Technology Inc, 2012)

Esta gama de opciones tan limitada motivó el desarrollo de una plataforma propia más adecuada a las necesidades del proyecto ROBAUCO, pero que sirviese también para otros propósitos de investigación.

Respecto al desarrollo de un escáner 3D, la motivación es similar, aunque en este caso las opciones no son tan limitadas. Existen diversos métodos para la reconstrucción tridimensional de un entorno. En la presente investigación se ha optado por una tecnología láser para garantizar una calidad mínimo del modelo generado y un alcance suficiente. Además, se ha optado por una tecnología de tiempo de vuelo, dado que por lo general los escáneres de diferencia de fase presentan niveles de ruido considerables para entornos de exterior.

Existen diversos sistemas de reconstrucción tridimensional que se ajustan a estas características. Sin embargo, la mayoría son costosos y delicados. Específicamente las vibraciones pueden desajustar los espejos y dejar al escáner inservible. Por ello, solo se pueden desplazar de un punto a otro con el dispositivo apagado y no sobre un robot móvil como se propone la presente investigación. Además las tomas de información de estos dispositivos son generalmente lentas (aunque de gran calidad; ver apartado 6.3.3).

Para el uso conjunto con robots móviles existe un escáner tridimensional de gran calidad, el Velodyne HDL-64E (ver Figura 5-4), utilizado por muchos equipos participantes en el DARPA Urban Challenge. Sin embargo, su precio oscila entre los 80.000€ y los 100.000€. Por ello, en el presente trabajo se ha optado por una solución alternativa: el uso de un escáner Láser

bidimensional Sick LMS-111 y una plataforma giratoria. De esta forma se ha conseguido un sistema de adquisición de datos robusto, rápido y de coste notablemente inferior.



Figura 5-4: Escáner tridimensional Velodyne HDL-64E (Velodyne Lidar, 2010)

El presente capítulo incluye, igualmente, la descripción del simulador robótico 3D de entornos tridimensionales diseñado para el desarrollo, prueba y validación de los algoritmos propuestos en esta investigación. Este simulador ha facilitado enormemente la puesta a punto de dichos algoritmos y ha permitido un ahorro de tiempo considerable, al ser utilizado como paso previo a la validación en el robot real. La motivación para el diseño e implementación de este simulador robótico radica básicamente en las necesidades específicas para simular nuestras condiciones de exploración.

Existen diversos simuladores robóticos con capacidad de manejar información tridimensional. Destacan básicamente tres: UsarSim, Gazebo y Microsoft Robotics Studio.



Figura 5-5: Capturas de pantalla del simulador UsarSim (Geeknet, Inc., 2012)

El simulador UsarSim (ver Figura 5-5) (Geeknet, Inc., 2012) tiene algunos inconvenientes, siendo el principal de ellos que está basado en el motor de simulación física de un juego de 2003, el cual fue completamente remplazado a partir de 2004. Por ello, conseguir copias del

mismo fue muy complicado durante mucho tiempo, hasta que a mediados de 2010 la compañía del juego lo ofrece libremente para UsarSim. Esto retrasó el desarrollo del simulador hasta llevarlo prácticamente al abandono, lo cual lo condenó al uso en sistemas Linux antiguos por la falta de actualización. En 2010 se reactivó el proyecto debido al interés de los equipos participantes en la RoboCup del cual es el simulador oficial. Sin embargo, además del inconveniente citado, el uso e instalación de este simulador es realmente complejo y tiene una curva de aprendizaje muy lenta. Igualmente la lista de sensores incluido es muy limitada.

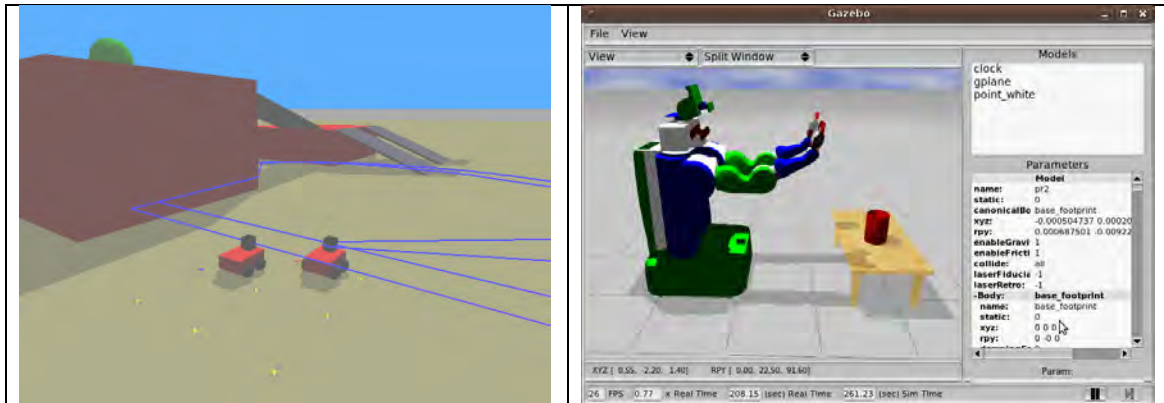


Figura 5-6: Capturas de pantalla del simulador Gazebo (Koenig & Hsu, 2012)

Por otra parte, Gazebo (Koenig & Hsu, 2012) es un simulador muy conocido (ver Figura 5-6), está correctamente mantenido y actualizado y perfectamente integrado en ROS (Willow Garage, 2009) y en Player/Stage (Gerkey, 2010). Sin embargo, hasta ahora ha estado orientado a la simulación de robots complejos o poblaciones de robots en entornos simples. Esto quiere decir que incluir objetos de geometría compleja como parte del entorno no es posible. Además, aunque es posible incluir terrenos de diferentes alturas, la forma de incluirlos no es sencilla ni fiable.



Figura 5-7: Capturas de pantalla del simulador de Microsoft Robotics Studio (Microsoft, 2006).

El simulador de Microsoft Robotics Studio (Microsoft, 2006) está en una situación similar a Gazebo, aunque en este simulador sí es posible introducir objetos con geometrías complejas. Sin embargo, la interacción de los sensores del robot se realiza exclusivamente con los *bounding boxes* de estos objetos (ver Figura 5-7), de forma que la información 3D capturada sería poco útil para simular la metodología propuesta en la presente investigación. Además, este simulador solo está disponible bajo entornos Windows, lo que representa un inconveniente adicional de cara a su uso en sistemas donde los recursos computacionales deben estar optimizados.

5.2. Robot Todoterreno

La plataforma desarrollada en la presente investigación ha sido diseñada con el propósito principal de desarrollar e implementar los algoritmos explicados a lo largo de este documento. De manera adicional, y sin perder de vista este objetivo en su diseño e implementación, se ha buscado, también, dotarla de una gran versatilidad que permita su utilización en la mayor cantidad posible de tareas que se desarrollen en entornos exteriores. Respondiendo a este propósito la plataforma es capaz de moverse autónomamente por entornos desestructurados y de realizar múltiples y muy variadas funciones gracias a su diseño modular, que permite ampliar o variar su funcionalidad mediante la instalación de diferentes dispositivos. En la Figura 5-8 puede verse su aspecto final, con el sistema de escaneo 3D instalado en su zona central.



Figura 5-8: Robot Todoterreno desarrollado en la presente investigación

Con el fin de que la plataforma robotizada cuente con la flexibilidad necesaria para realizar diferentes tipos de misiones en exteriores, se han tenido que considerar dos factores principales. El primero es el diseño de una mecánica robusta que permita el desplazamiento por diferentes tipos de terreno en los que el robot puede llegar a desempeñar misiones. El segundo y más importantes es el desarrollo de un sistema perceptivo y de comunicaciones que permita un grado de autonomía ajustable en función del tipo de misión y de los dispositivos instalados en cada caso. De esta forma, el robot puede realizar misiones con diferente nivel de intervención humana.

Los niveles de autonomía han sido discutidos por distintos autores (Bruemmer, Dudenhoeffer, & Marble, 2002); (Goodrich, Olsen Jr., Crandall , & Palmer, 2001). En concreto, se ha relacionado directamente la autonomía de un robot con la realización de las tareas, de forma que algunas de estas se realizan más eficazmente con un mayor grado de autonomía. La Figura 5-9 presenta estos niveles junto con las tareas para las que presentan más eficacia.

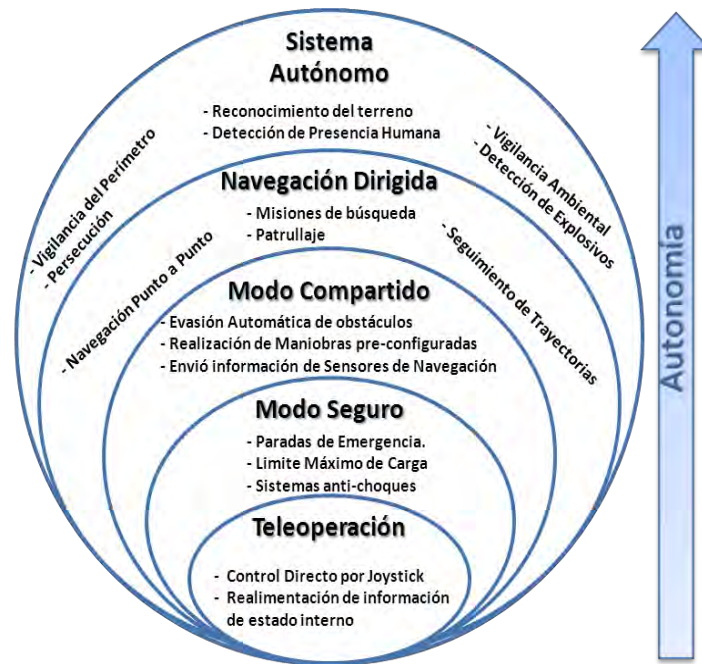


Figura 5-9: Niveles de Autonomía para un robot móvil

A partir de esta figura se puede observar como los requerimientos perceptivos de un sistema robotizado aumentan según aumenta la autonomía requerida para una misión. Por otro lado, se puede ver que a medida que disminuye la necesidad de intervención humana en el desempeño de una misión, la fiabilidad de una conexión es un requerimiento menos crítico. Sin embargo, el ancho de banda de la comunicación y el alcance del mismo es independiente de la autonomía del sistema y está más relacionado con la misión que debe desempeñar y con las características propias del robot. Así por ejemplo, un robot teleoperado tiene el mismo nivel de autonomía cuando es operado *in situ* que cuando es controlado a través de internet, pero el ancho de banda necesario en ambos caso es totalmente diferente.

Para que la plataforma pueda alcanzar todos estos grados de autonomía debe contar con un sistema perceptivo completo que le permita navegar por el entorno (ver sección 5.2.3), además de con los dispositivos específicos para cada misión. Por otro lado, el sistema de comunicaciones (ver sección 5.2.4) depende directamente del tipo de misión que esté desempeñando el robot y de la cantidad de información que debe fluir entre este y el operador.

5.2.1. Características generales de la plataforma móvil

La plataforma móvil es el componente encargado de ofrecer todo el soporte necesario para la reconstrucción tridimensional, desde la movilidad del sistema que permite adquirir las diversas tomas de datos, hasta la propia alimentación del mismo. Uno de los grandes retos planteados por su diseño ha sido lograr una movilidad idónea en entornos de diversa naturaleza, con firmes de distinta consistencia y con capacidad de acceso a zonas como cuevas, túneles, galerías subterráneas o tuberías. Para obtener esta versatilidad ha sido necesario prestar

especial atención a dos características del diseño: las dimensiones del robot y su sistema de tracción.

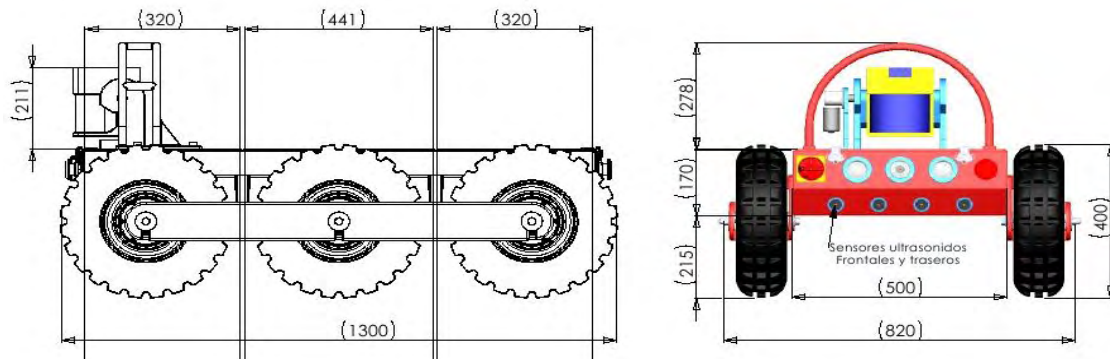


Figura 5-10: Dimensiones Físicas del Robot

Las dimensiones del robot (104 cm de largo x 85 cm de ancho, ver Figura 5-10) le permiten moverse por terrenos rugosos, introducirse en espacios angostos o de difícil accesibilidad, y ser transportado y manejado cómodamente. Además, la altura de la parte inferior del cuerpo del es de 21 cm para hacer más fácil el sobrepaso de pequeños obstáculos, mientras que la superficie superior se encuentra a 38cm (por debajo de la altura de las ruedas). Esta altura reducida facilita que el robot se pueda introducir en espacios tales como cuevas o u otros lugares con poca altura.

La plataforma es totalmente simétrica respecto a su eje transversal lo cual implica que posee la misma facilidad para el desplazamiento en cualquiera de los dos sentidos. Las ruedas sobresalen del cuerpo del vehículo en ambos extremos, lo que facilita por ejemplo la escalada de obstáculos. Además se ha dotado a los bajos de la plataforma de una protección en plano inclinado con el fin de pueda superar con facilidad distintos tipos de obstáculos sin sufrir daños.

Con el propósito de lograr una plataforma flexible que se pueda adaptar a las distintas circunstancias previstas para el robot, se ha diseñado un sistema de tracción basado en dos trenes de ruedas situados en los laterales, creando una configuración diferencial. Cada módulo de tracción cuenta con 3 ruedas con motor eléctrico integrado.

Igualmente la plataforma ha sido dotada de una mecánica robusta, ideada para que el robot pueda transportar distintos equipos seleccionados en función de la aplicación, a través de diversos tipos de terrenos. También alberga un amplio sistema de baterías compuesto por 3 baterías ion-litio de 24V/20Ah que proveen al robot de una autonomía suficiente para abordar sin dificultad misiones de varias horas de duración.

Para garantizar la correcta captación de imágenes por las cámaras de navegación, aún en situaciones de escasa luminosidad, se ha equipado el robot con dos pares de lámparas en cada uno de sus extremos.

Finalmente, debido a la multifuncionalidad que se espera del robot, se han instalado dos conectores en los laterales para facilitar la adición de diferentes dispositivos tales como un cabezal de escaneo 3D, un brazo robotizado o cualquier otro elemento destinado a misiones

específicas. Esto es posible ya que se ha dispuesto una configuración de pines que consta de alimentación y comunicación USB con el dispositivo conectado.

5.2.2. Arquitectura de Control

La arquitectura propuesta para el sistema de control de la plataforma está dividida en dos niveles de gestión. Por un lado está la gestión de las tareas de bajo nivel del robot que controlan las funciones más básicas relacionadas directamente con la parte *hardware*, como el control del movimiento o la monitorización de los estados de emergencia. Por otro lado se encuentra la gestión de tareas de alto nivel donde se ejecutan los procesos más complejos computacionalmente y que requieren la capacidad de un ordenador para su ejecución.

Esta subdivisión resulta muy ventajosa ya que permite la intervención en el sistema a cualquier nivel, permitiendo así el control de las tareas de bajo nivel. La Figura 5-11 muestra cómo se interconectan los diferentes componentes del sistema y en que nivel de gestión son controlados.

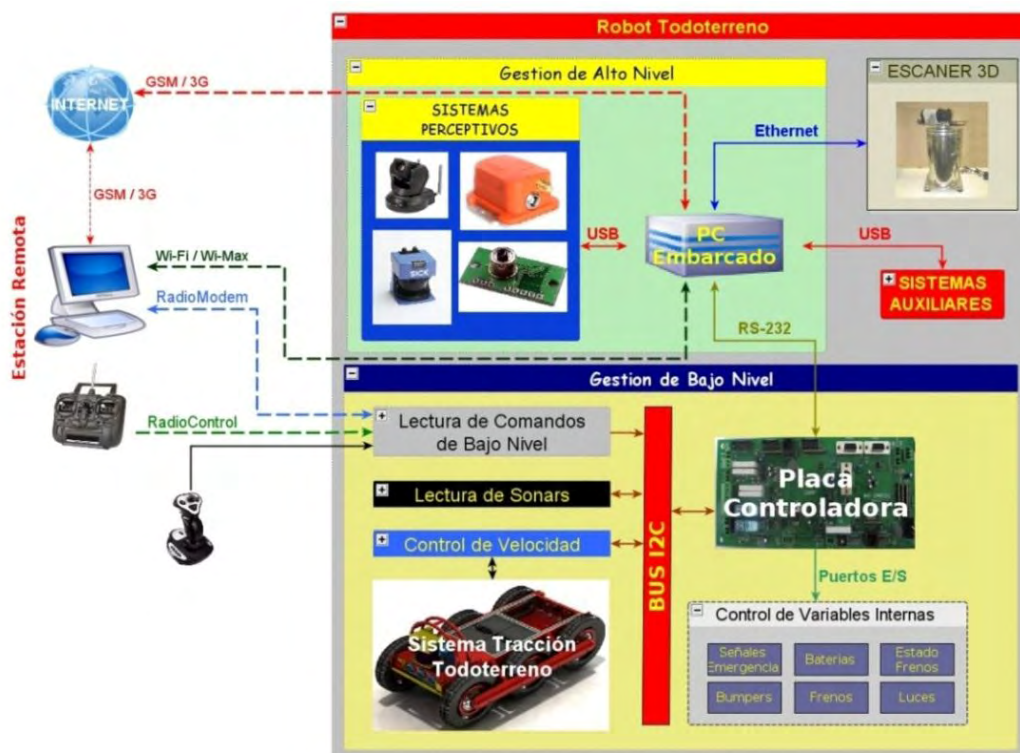


Figura 5-11: Arquitectura Hardware del Sistema

Gestión de bajo nivel

Para la gestión de bajo nivel se ha diseñado una placa basada en microcontroladores tipo dsPIC que hemos denominado GPRMC (“*General Purpose Robot Motion Controller*”), la cual le da una

gran versatilidad al sistema puesto que permite el uso de la plataforma aún sin un sistema informático (por ejemplo cuando está siendo teleoperada). En esta capa se gestionan las funcionalidades imprescindibles y más relevantes para la operación y la seguridad del sistema.

Placa Controladora GPMRC

Esta placa (ver Figura 5-12) ha sido diseñada con el objeto de controlar diferentes tipologías de robots móviles, pero además incluye algunas particularidades propias de control de la plataforma todoterreno. En concreto, se encarga de ejecutar las tareas más básicas del robot tales como gestionar el control de velocidad de hasta seis motores a la vez, proveyendo un lazo de control individual para cada motor a partir de la información de los *encoders* así como el cálculo de la odometría del robot. El control de velocidad de los motores puede ser hecho a partir de las referencias de velocidad lineal y angular para el sistema o de referencias individuales para cada motor.



Figura 5-12: Placa de control genérico de Robot GPMRC

Las referencias de movimiento pueden provenir de diversas fuentes, las cuales son gestionadas por la GPMRC mediante un sistema de jerarquías para establecer cuál es la referencia prioritaria. En este caso, por ejemplo, el robot puede recibir órdenes desde un *joystick* conectado directamente a la placa, desde un mando de radio control, desde un *radiomodem* o desde el PC. Para evitar conflictos la placa tiene establecidas un conjunto de reglas que determinan cuál debe ser la referencia de control prioritaria en cada caso, garantizando así que no haya órdenes contradictorias, que en caso de fallo siempre se pueda tomar el control del robot de alguna forma y finalmente, que el robot se detenga al no encontrar ninguna fuente de referencias válida.

Otra funcionalidad de la placa es identificar cualquier emergencia interna del robot y decidir el comportamiento del mismo en estas situaciones (setas de emergencia, batería baja, pérdida de comunicación). Además, la placa provee recogida de información sobre el estado interno del robot y la transmisión de esta hacia las capas de control de más alto nivel.

La placa también cuenta con cuatro salidas de relé para activar actuadores u otro tipo de dispositivos que requieran de potencia media para su activación. Además dispone de buses CAN e I2C para la conexión de dispositivos que provean funcionalidades adicionales como los sonares I2C con los que cuenta la plataforma.

Gestión de alto nivel

Para las tareas de más alto nivel, la plataforma dispone de un ordenador embarcado que se encarga de las operaciones que requieren mayor capacidad computacional, tales como como gestionar las cámaras web y la conexión de red del robot o leer los sensores de alto nivel (GPS, IMU, etc), ente otras. El ordenador embarcado también ejecuta los algoritmos de localización, exploración y navegación integrados en el sistema.

Ordenador Embarcado

La plataforma cuenta con un ordenador embarcado que se encarga de la gestión de las tareas de alto nivel (Figura 5-13). Este ordenador contiene todo el software de control del robot y en él se gestionan todas las tareas que el robot debe realizar de forma automática. En particular, gestiona los distintos sistemas de comunicación con los que cuenta el sistema, ya que centraliza toda la información. Es el encargado de hacer que los datos sean accesibles desde el exterior del sistema, sin importar desde dónde se esté accediendo al mismo. También tiene la responsabilidad de gestionar la comunicación con todos los dispositivos del sistema a través de los puertos serie o USB, según el caso, conteniendo el software de control para cada uno de estos además de los módulos necesarios para el funcionamiento del sistema. En este ordenador se encuentran los módulos que tienen control directo sobre algún dispositivo del sistema, encargándose de gestionar todas las comunicaciones entre este y el resto de aplicaciones.

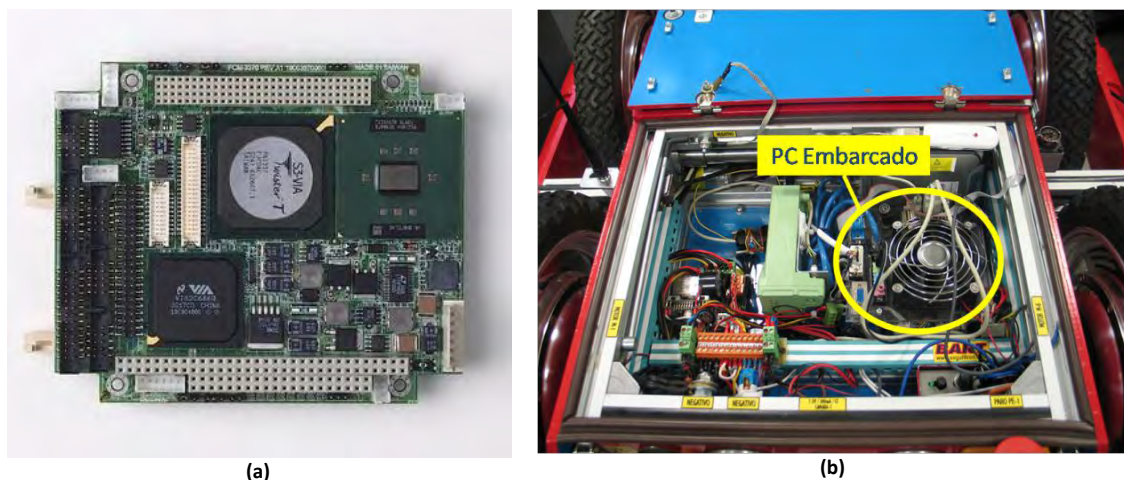


Figura 5-13: (a) Ordenador embarcado del sistema. (b) Instalado en el robot

Capas de Software

En este nivel de gestión los módulos *software* se pueden clasificar en varias capas según su función dentro del sistema. La capa más importante es la capa de gestión de la información, que es donde se centraliza toda la información del sistema asumiendo la responsabilidad de que cada módulo reciba los datos necesarios para el desarrollo de sus funciones.

La capa de control de dispositivos contiene los programas que se comunican con los dispositivos de percepción, la placa controladora del robot, el escáner láser y los dispositivos auxiliares en caso de haberlos. Estos programas controlan los datos provenientes de los dispositivos y traducen esta información para que los módulos de control del robot y de supervisión y teleoperación puedan hacer uso de ella.

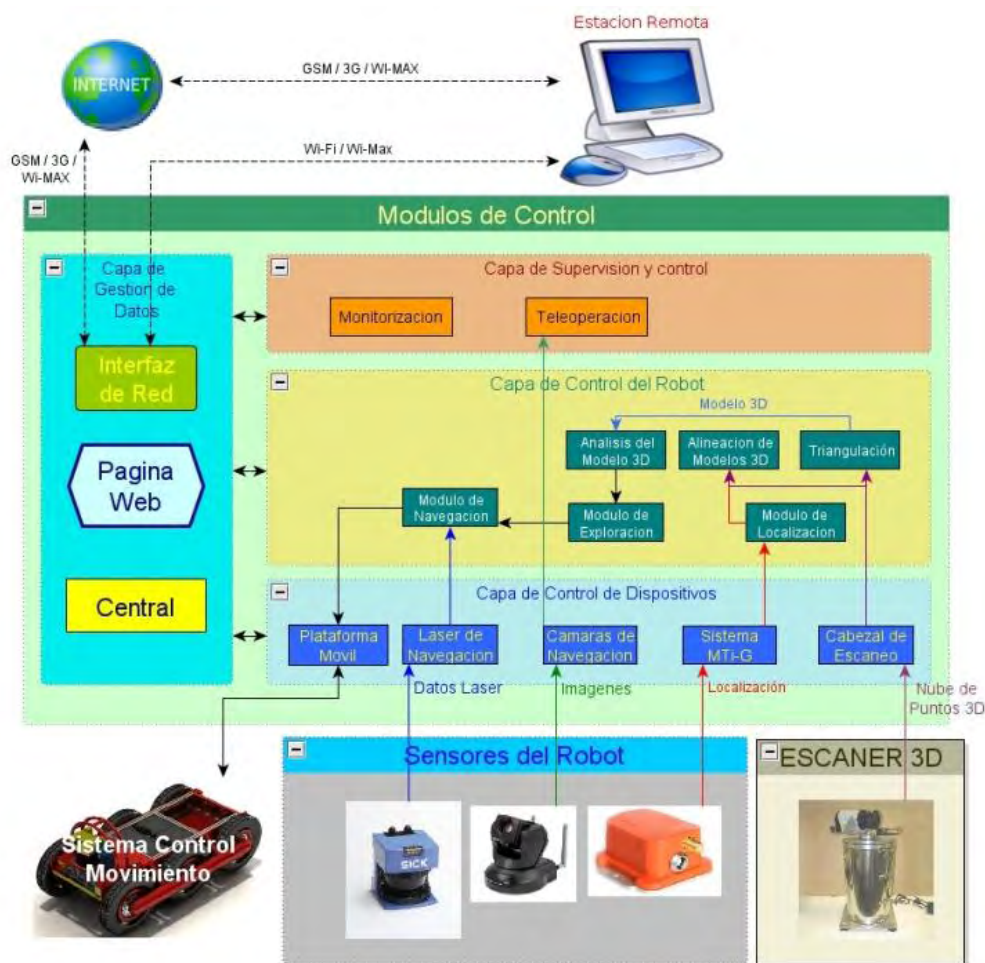


Figura 5-14: Módulos Software del sistema

Los módulos de control del robot son los que se encargan de realizar las tareas para las que ha sido programado el sistema (por ejemplo la exploración de un entorno) incluyendo todos los procesos auxiliares (localización, navegación, etc.) que se puedan requerir para la realización de dichas tareas. Finalmente esta la capa de supervisión y control que contiene los módulos que reciben las ordenes de un operador humano cuando el robot está siendo teleoperado y que a su vez transmiten la información sobre el estado del robot al operador. La Figura 5-14 muestra los módulos de software utilizados para la exploración de un entorno.

La arquitectura software está basada en las librerías de comunicaciones IPC (Simmons, 2009), las cuales son un *software* flexible y eficiente que provee la capacidad de intercambiar mensajes entre procesos. Está basado en un protocolo de transmisión del tipo “publicación/suscripción” y cuenta con una aplicación llamada *central* a la cual se conectan los distintos programas que intercambian mensajes dentro del sistema. Además, su utilización hace innecesario que todos los programas corran en el mismo ordenador ya que estas librerías permiten el intercambio de mensajes utilizando una interfaz de red, lo cual hace posible utilizar distintos ordenadores para controlar el sistema.

La elección de este paquete de comunicaciones se basó principalmente en las razones que se han venido exponiendo pero además, también se ha tenido en cuenta que estas librerías están disponible tanto en Linux como en Windows lo cual facilita el desarrollo de aplicaciones de supervisión por parte de otros desarrolladores.

El servidor central es una aplicación que utilizan las librerías IPC para centralizar toda la información y garantizar que cada módulo reciba la información a la que está suscrito. Este servidor recibe todos los mensajes publicados por los distintos módulos y los retransmite hacia los módulos que se han suscrito al tipo de mensaje que este ha recibido. Adicionalmente el servidor central gestiona el *buffer* de transmisión para evitar que se pierdan datos entre los procesos.

Librerías de Control del Robot

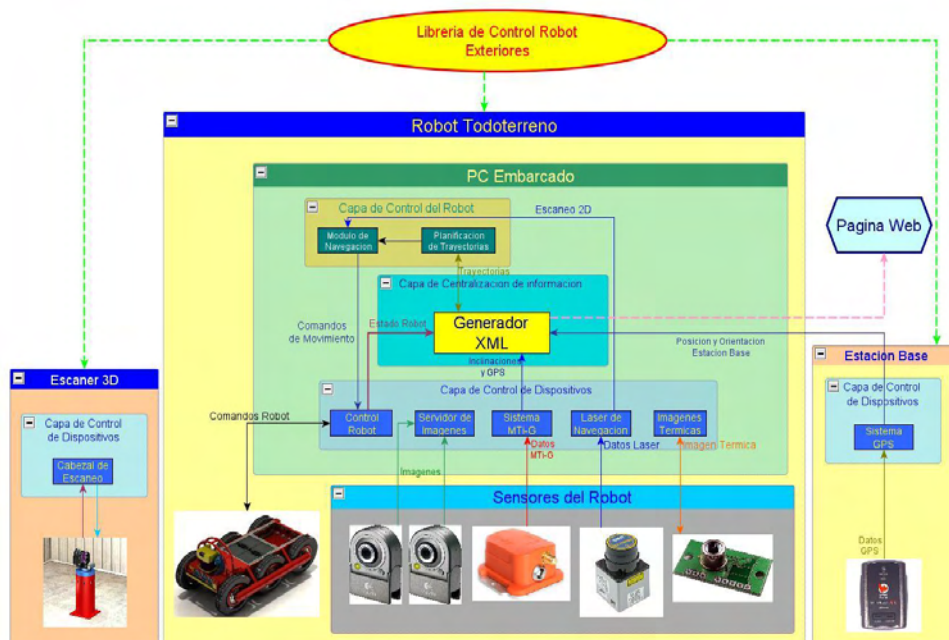


Figura 5-15: Esquema de la arquitectura software del sistema

Adicionalmente, con el propósito de facilitar el desarrollo de las distintas aplicaciones de control del robot a lo largo de la investigación aquí descrita se ha desarrollado una librería que permite el desarrollo rápido de estas aplicaciones. Este componente ha sido una parte crucial

del desarrollo de los distintos programas del robot y provee funciones para controlar y recoger datos de cada uno de los módulos del sistema.

Cada funcionalidad prevista en la librería tiene un objetivo específico, y está dirigida bien sea a enviar una orden a un módulo, o a recibir la información de este. Todas las funciones verifican que el módulo al que va dirigido el comando o la petición esté conectado y posteriormente envían el mensaje. En caso de que este mensaje sea un comando se envía un mensaje específico para dicho comando mientras que si se trata de una petición se envía un mensaje del tipo *query/reply* el cual transmite una orden de recogida de datos al módulo objetivo y recibe los datos capturados por los módulos de control.

Capa de Control de Dispositivos

En este nivel se encuentran todos los módulos que ejercen control directo sobre cualquier dispositivo instalado en el robot. Algunos módulos contenidos en esta capa son los siguientes.

Control de la Plataforma Móvil: Este módulo es el encargado de la comunicación con la placa controladora del robot. Recibe (a través del puerto serie) toda la información recogida por el controlador y ensambla un mensaje IPC con la información sobre el estado del robot para publicarlo hacia los módulos que requieran de esta información. Igualmente, recibe las órdenes de movimiento para el robot y los transmite hacia el mismo.

Módulo de Captura de imágenes: Captura las imágenes de las cámaras web de navegación, y les aplica un post-procesamiento para acercar o alejar las imágenes, modificar el brillo y rotarlas de acuerdo a los mensajes que recibe desde el módulo de visualización y monitorización, que habrán sido emitidos por el usuario del sistema.

Módulo de Control del MTi-G: Recibe los datos del sensor MTi-G (Orientación 3D, Posición y Velocidad, Datos GPS, los datos puros de los sensores, aceleración y tasa de giro en 3D, Campo magnético de la Tierra en 3D y temperatura; véase la sección 5.1.3). A partir de estos datos, ensambla y publica los mensajes IPC conteniendo esta información, la cual es muy útil para la localización del robot y la referenciación de la nube de puntos tridimensional.

Control del Robot

Módulo de Localización: Este módulo recoge la información proveniente del citado sensor MTi-G y del odómetro del robot para calcular la posición más probable del mismo. Una vez hecho esto, publica esa posición junto con la inclinación del robot capturada por dicho sensor del MTi-G.

Módulo De Extracción De Información 3D: Este módulo recibe la nube de puntos 3D y a partir de ella crea un mapa de dificultad para el desplazamiento del robot.

Módulo de Planificación del Movimiento: Calcula la trayectoria con menos obstáculos entre dos posiciones de acuerdo al mapa de dificultad establecido por el módulo de extracción de información 3D.

Módulo de navegación: Calcula las velocidades de las ruedas para seguir la trayectoria fijada para el robot. Se encarga, igualmente, de buscar obstáculos insalvables a partir de la información de los sonares y evitarlos sin perder la trayectoria.

Supervisión y Monitorización

Ciente de Monitorización y Control: Se trata de una interfaz gráfica que puede ser ejecutada de forma remota. A través de este cliente el usuario puede ver todos los datos transmitidos por el robot e igualmente puede comandar velocidades al robot, ordenarle que vaya a una posición específica, ver las imágenes de las cámaras web, los datos de los sonares y la posición GPS.

Generador XML: Esta es una aplicación que recoge toda la información proveniente de los módulos de control de dispositivos y genera un mensaje a partir del cual es posible extraer el estado del robot. Igualmente este módulo actualiza periódicamente un fichero *xml* con esta misma información, el cual es publicado utilizando el servidor web del robot. Este fichero es leído por una página web desarrollada utilizando el API de *googlemaps* y en la cual se puede ver la posición del robot y de la estación de supervisión en el mundo, además de la información más relevante sobre el estado del sistema.

5.2.3. Sistemas Perceptivos

Los sistemas perceptivos incorporados a la plataforma están orientados a dos tareas principales: la localización del robot y la percepción del entorno.

Los sensores orientados a la localización del robot, tienen como objetivo calcular la posición, orientación e inclinación del robot en todo momento. La ejecución de estas funciones tiene una importancia transcendental, no sólo para que la navegación pueda ser exitosa, sino también, para que la información capturada del entorno esté referenciada correctamente en el mundo.

Para la localización y navegación se utiliza como sensor principal un MTi-G de Xsens (ver Figura 5-16) cuyas principales cualidades son las de ser un sistema pequeño, liviano y de muy bajo consumo, ideal para la navegación y control de sistemas no tripulados. Esta clase de sensores están compuestos por una unidad de medición tipo IMU (sistema de medición inercial), un sistema GPS, un sensor de presión atmosférica, un sensor de campo magnético en 3D y un procesador digital interno que ejecuta algoritmos de fusión de datos en tiempo real con una tasa de muestreo muy superior a la de otros sistemas GPS comerciales.

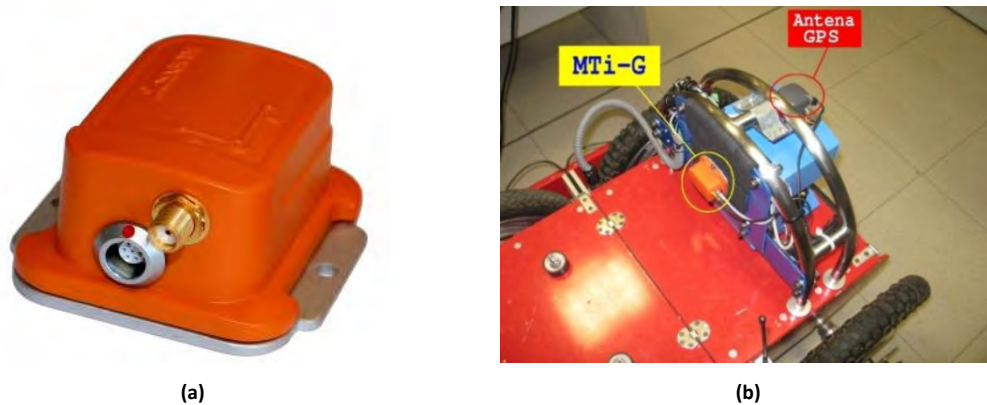


Figura 5-16: Dispositivo MTi-G (a) y su Posición en el Robot Todoterreno (b)

Para la localización del sistema se usa la posición odométrica del robot, proporcionada por la placa controladora, aunque en este caso se usa simplemente como sistema de apoyo ya que por la configuración del robot y las características propias de los sistemas odométricos, es poco fiable. Por este motivo se utiliza la información del MTi-G que es mucho más fiable en exteriores, para integrarla con la odometría mediante el uso de un filtro probabilístico que provee la posición en la que existe una mayor probabilidad de que se encuentre el robot.

Los demás sistemas perceptivos permiten al robot capturar información relevante acerca del entorno. La percepción es una tarea fundamental para desempeñar cualquier trabajo en entornos dinámicos. En esta línea, hay que distinguir entre los sistemas perceptivos que forman parte del propio robot y los sistemas perceptivos instrumentales, que se instalan según la misión que deba realizar y que se conectan dependiendo de los requerimientos de cada tarea. En este apartado se incluye únicamente, la descripción de los sistemas perceptivos que forman parte del robot. Los instrumentales, utilizados para la reconstrucción tridimensional, son descritos en la sección 5.1.

Para la navegación autónoma se ha dispuesto en la parte superior del robot (ver Figura 5-17) un escáner láser tipo SICK LMS 200 el cual hace un barrido desde 0° hasta 180°, cada 0.25°, 0.5° o 1° según la configuración dada, y cubre hasta 32 m de distancia.

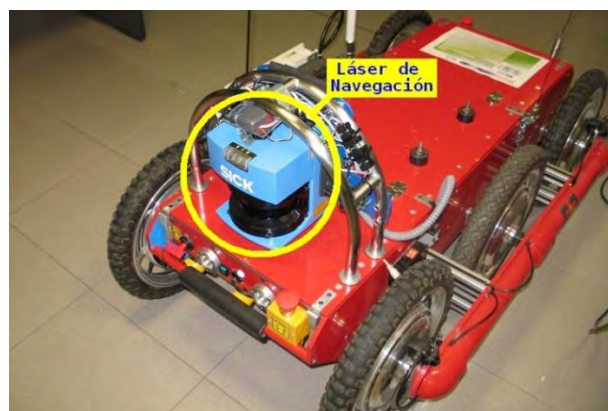


Figura 5-17: Escáner Láser Tipo Sick LMS200 y montaje sobre el robot.

El escáner esta acoplado a la parte frontal del robot, puede ser fácilmente instalado o removido, e indica dónde se encuentra un obstáculo que debe evitar el robot, información

indispensable para la navegación reactiva. La altura del plano del escáner es adecuada para detectar obstáculos que sean insalvables como árboles, piedras, muros, personas, etc.

Adicionalmente, con el fin de obtener la mayor información posible para una navegación segura, se han instalado tres sensores de ultrasonidos en cada extremo del robot con un ángulo ajustable, orientado hacia el suelo (ver Figura 5-18). Estos sensores proporcionan datos que permiten detectar irregularidades en el terreno sobre el que se desplaza el robot y sobre el que tiene frente a él. De manera simultánea facilitan conocimientos relativos a la distancia que lo separa de los factores de riesgo detectados y que pueden ser muy variados: piedras, zanjas, escalones, objetos, etc. Con estos datos, el sistema de control reactivo podrá adoptar la decisión más adecuada evitando que el robot sufra daños físicos o bloqueos en el terreno.

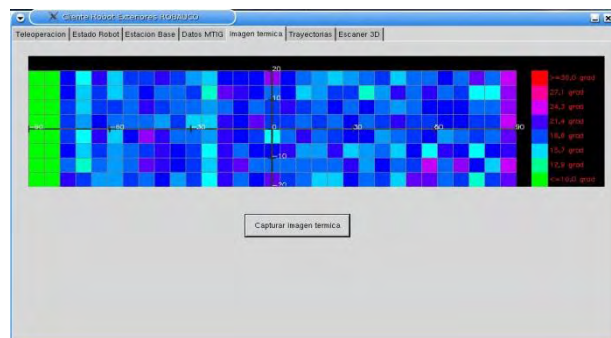


Figura 5-18: Sensores de Ultrasonidos del Robot

El robot dispone también de un pequeño sensor térmico multipunto (Figura 5-19) que permite localizar fuentes de calor en el área de trabajo. Este dispositivo está formado por una matriz de 8 sensores colocados linealmente de manera que puede medir 8 puntos adyacentes simultáneamente. Esto hace posible medir la temperatura de un objeto sin estar en contacto con él y entregar la información térmica en forma de 8 píxeles ubicados linealmente. A diferencia de los sensores utilizados en sistemas de alarmas y detectores para encender luces, este dispositivo no necesita que haya movimiento para detectar el calor. La información suministrada por este sensor es leída por la placa controladora del robot la cual controla igualmente un servo motor que se ha dispuesto para hacer un barrido horizontal y tomar 32 mediciones diferentes, con la finalidad de obtener un mapa térmico de 180 grados.



(a)



(b)

Figura 5-19: (a) Sensor Térmico Instalado en el Robot Todoterreno. (b) Captura de la representación gráfica de los datos capturados por el sensor

El sensor de temperatura es un módulo térmico de detección por infrarrojos en un rango de $2\mu\text{m}$ a $22\mu\text{m}$ (que es la longitud de onda del calor radiante). Se trata de un sensor tipo termopila, como los que se utilizan en los termómetros por infrarrojos sin contacto. Cuenta con un ángulo de detección o campo de visión (FOV) muy amplio, de aproximadamente 100° . Por ello se incorpora una lente que le permite obtener un campo de visión reducido de aproximadamente 6° por pixel. De esta manera, el campo de visión del sensor térmico resulta de 48° por 6° aproximadamente. Esto sumado al barrido del servo al que esta acoplado el sensor permite capturar hasta 192° en el plano horizontal.

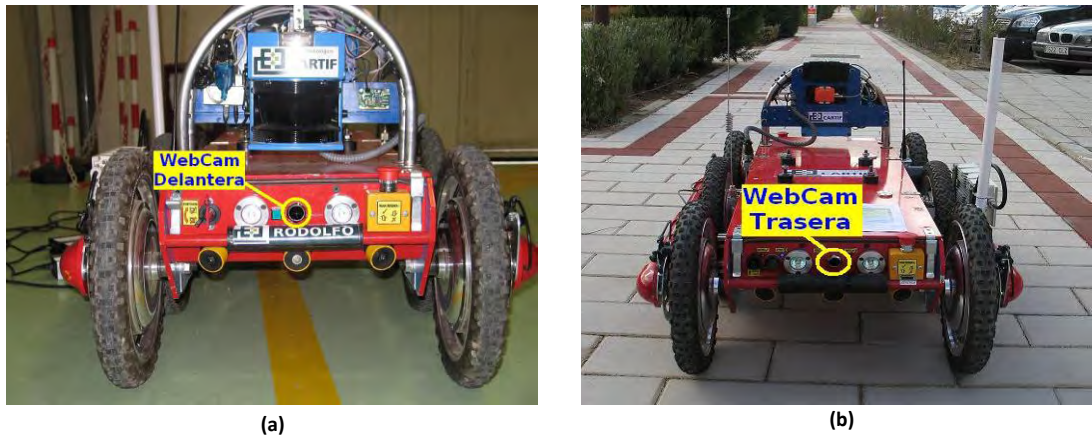


Figura 5-20: WebCams instaladas en el Robot, (a) delantera, (b) Trasera

Para la teleoperación y para la supervisión de la navegación, se utilizan dos cámaras tipo webcam instaladas en la parte anterior y posterior del robot (ver Figura 5-20). A través de estas cámaras el PC embarcado captura imágenes en color que pueden ser visualizadas en el ordenador de teleoperación y en otros ordenadores conectados a Internet. Esto permite además que una persona conectada a la página web pueda ver la situación del robot y del entorno en cualquier momento.

Para poder disponer de una adecuada iluminación en el campo de cobertura visual de las cámaras y lograr la mejor calidad en las tomas producidas por ellas cuando trabajan en ambientes con poca luz, se han instalado en ambos extremos del robot dos focos halógenos de 30W cada uno.

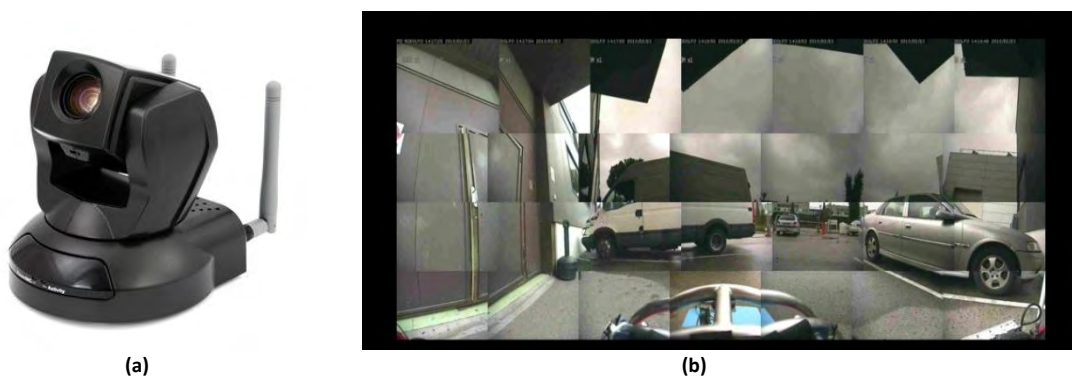


Figura 5-21: Cámara PTZ (a); Reconstrucción del entorno utilizando la cámara PTZ (b)

Finalmente la plataforma cuenta con una cámara pan-tilt con un zoom óptico de 10X (ver Figura 5-21), la cual facilita la teleoperación del sistema y resulta sumamente útil para misiones de vigilancia de entornos. Su incorporación como parte del robot responde a un doble propósito: por un lado la teleoperación, ya que gracias a su sistema orientable y a su zoom el operador puede enfocar un punto del entorno para analizar algún detalle específico. Por otra parte, puede ser utilizada para hacer un mosaico de imágenes del área de trabajo lo cual es útil para hacerse una idea del estado general del entorno. Esta cámara se controla y proporciona toda su información a través de una interfaz de red.

5.2.4. Sistema de comunicaciones

El sistema de comunicaciones en un robot de exteriores es extremadamente importante ya que los puntos de control y supervisión suelen estar alejados del robot y es deseable que en caso necesario, un operador pueda entrar en cualquier punto del lazo de control del sistema. Las comunicaciones han de ser robustas y es esencial contar con diferentes sistemas que permitan tener una amplia gama de opciones de comunicación, reforzando con unos sistemas los puntos débiles que puedan presentarse en otros en cuanto a alcance, velocidad de transmisión, robustez o interferencias.

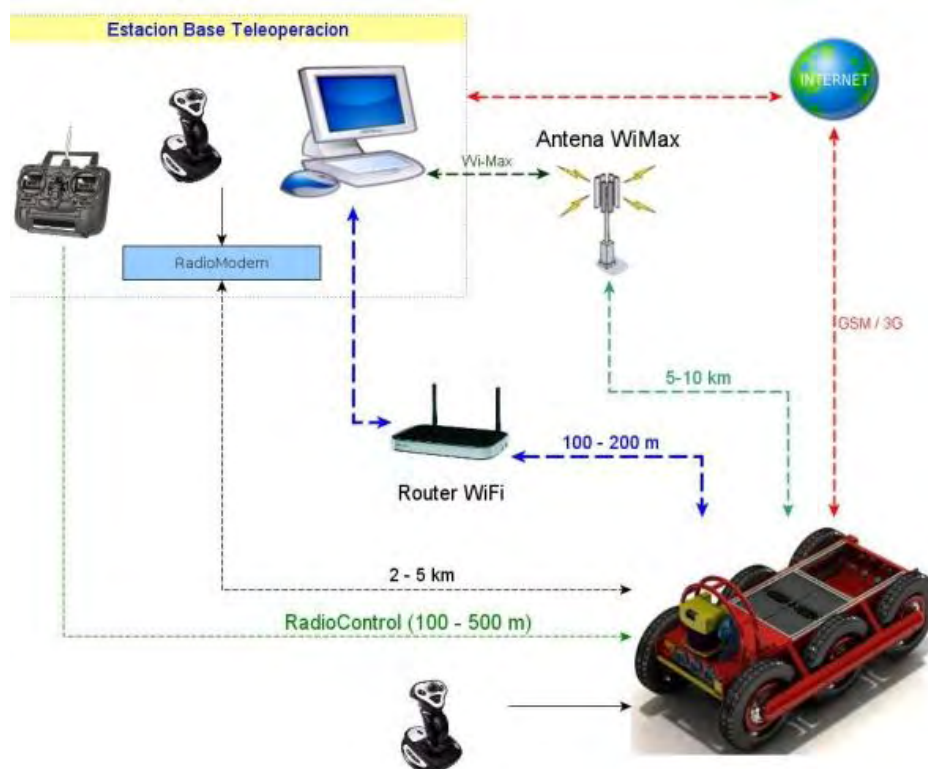


Figura 5-22: Sistemas de Comunicación del Robot

Con este propósito se han dispuesto en el robot los siguientes sistemas de comunicación: Radiocontrol, radiomódem, WiFi, WiMax y 3G/GPRS (ver Figura 5-22). El objetivo es mantener siempre un enlace de comunicación disponible para poder comunicarse con el robot.

Radiocontrol

Para la teleoperación por radiocontrol (ver Figura 5-23) se ha escogido un sistema emisor-receptor comercial. Estos sistemas funcionan transmitiendo comandos de referencias al receptor embarcado, el cual genera una serie de señales PWM en función de dichas referencias. Los mandos de la emisora se pueden configurar de múltiples formas para modificar el comportamiento de las señales PWM. El alcance de estos es bastante limitado, ya que tienen una zona de trabajo aproximada de máximo 1km lo cual lo hace un sistema recomendable únicamente para teleoperación con visualización directa del robot.



Figura 5-23: Teleoperación del robot terrestre por radiocontrol

Se han evaluado las prestaciones de este equipo mediante pruebas realizadas en un entorno boscoso con suelo liso, cuyo resultado ha permitido comprobar que el sistema presenta una buena respuesta para una distancia de hasta unos 150m en este tipo de entorno; a una distancia mayor se producen retrasos en la ejecución de los comandos enviados lo cual es inadecuado para un correcto control del robot.

En conclusión, este sistema de comunicación es útil solo para casos en los que se quiera controlar el movimiento del robot de forma directa desde una distancia corta, pero es un sistema inadecuado para el control del robot a más alto nivel o a mayor distancia.

Radiomodem

Otra posibilidad es controlar el robot mediante radiomodems. Este modo de comunicación presenta un alcance superior y proporciona un enlace de transmisión de datos digitales desde la consola de operación al robot. Esto posibilita el envío de órdenes mediante un PC externo, así como la recepción de información proveniente del robot.

El módem se encuentra conectado a una antena flexible y la comunicación con el PC y el microcontrolador de control del robot es vía RS232 (ver Figura 5-24). La comunicación a través de radiomodems tiene un alcance de 5 km pero un ancho de banda limitado que solo permite la transmisión de comandos de control y de estado del robot, no de otra información útil para la teleoperación como imágenes de las cámaras web.

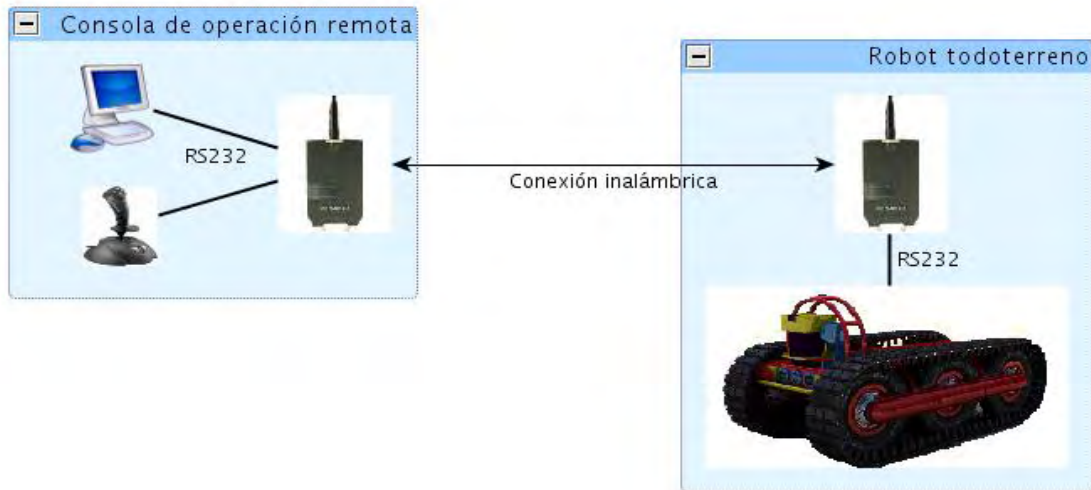


Figura 5-24: Teleoperación del robot terrestre por radiomodem

WiFi

Un tercer sistema de comunicación con el robot es vía WiFi (ver Figura 5-25). Aunque el alcance de estos equipos es bastante limitado (decenas de metros) la velocidad de transmisión es muy alta con lo que es ideal para la fase de desarrollo de aplicaciones así como para la transferencia de ficheros de datos de gran tamaño. Gracias a su velocidad y ancho de banda, permite la teleoperación del robot a media distancia mediante un ordenador portátil, así como disponer de realimentación visual y de acceso al sistema de control de alto nivel del robot.

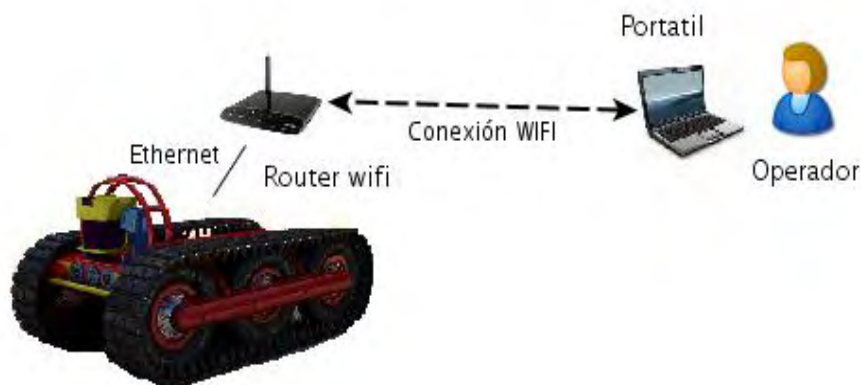


Figura 5-25: Control del robot terrestre por WiFi

WiMax

Para situaciones en las que es necesaria una mayor cobertura se ha provisto al robot de comunicación WiMax, un concepto similar a WiFi pero con mayor cobertura y ancho de banda, lo que le posibilita la teleoperación hasta una distancia de 5-6 km. WiMax son las siglas de

'Worldwide Interoperability for Microwave Access' y es la marca que certifica que un producto está conforme con los estándares de acceso inalámbrico 'IEEE 802.16'. En definitiva, ese trata de un Wi-Fi de alta potencia lo que le posibilita llegar sin cables a grandes distancias. De hecho, originalmente se proyectó como una alternativa inalámbrica al acceso de banda ancha ADSL y cable, así como una forma de conectar nodos Wifi en una red de área metropolitana.

La implementación en el sistema requiere la instalación de una estación base WiMax con la antena orientada hacia la zona de trabajo del robot, y de un adaptador WiMax con antena omnidireccional que va instalado a bordo. La estación base con su antena es transportable y se instala en una zona lo más elevada posible, desde donde se disponga de una buena perspectiva del entorno de trabajo.

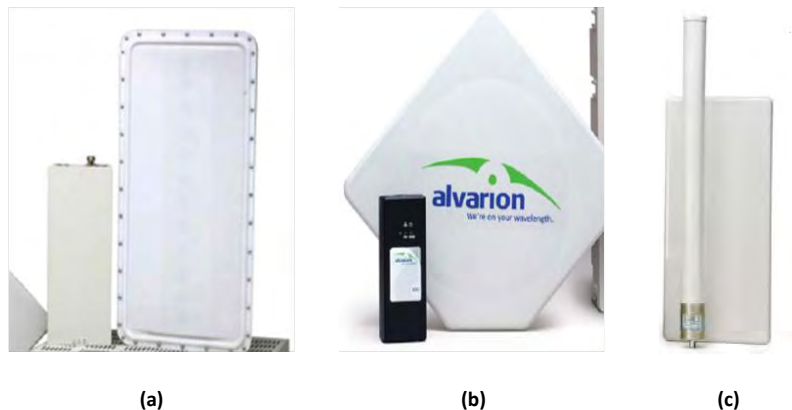


Figura 5-26: Unidad de acceso WiMax (a); Antena plana WiMax (b); Unidad de Acceso móvil Wimax (c)

En concreto, para el robot todoterreno desarrollado se han utilizado los siguientes equipos:

- Estación base compuesta por una unidad de acceso de la marca Alvarión modelo AUS-3-SA-5.4-VL que consta de una unidad interna (IDU) y una externa (ODU) (ver Figura 5-26a), además de una antena plana direccional de 15dB (ver Figura 5-26b).
- Estación móvil instalada en el robot y que consta de una unidad de suscripción de la marca Alvarion modelo SU-E-5.4-3-VL y una antena omnidireccional de 8 dB (ver Figura 5-26c).

3G/GPRS

Considerando la gran cobertura de la telefonía celular y las características favorables de esta tecnología, se ha implementado un sistema de conexión a Internet a través de la red de telefonía móvil usando un módem USB de datos de tercera generación (3G). Este módem se conecta a uno de los puertos USB del PC embarcado y funciona como un adaptador de conexión a Internet, con lo que se puede controlar el robot desde cualquier PC que disponga de conexión a la red (siempre que en la zona de trabajo del robot haya cobertura 3G o GPRS). Aprovechando esto, en el ordenador embarcado se encuentra un servidor web que publica

una página en la que es posible ver la ubicación, el recorrido y el estado del robot en tiempo real, a través de una interfaz desarrollada utilizando el API de *googlemaps* (ver Figura 5-27).

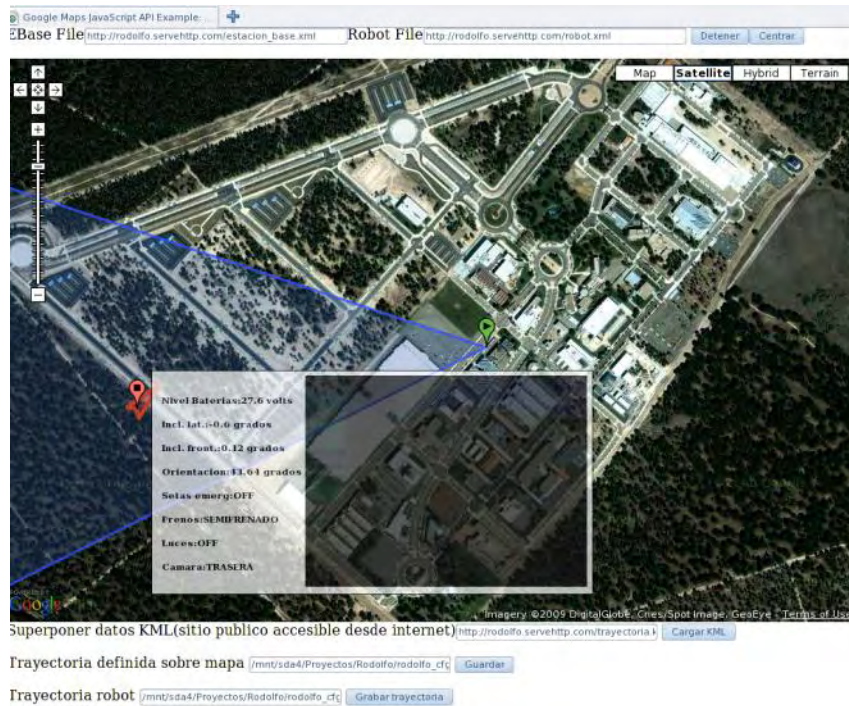


Figura 5-27: Interfaz de Supervisión por web

5.3. Escáner 3D



Figura 5-28: Cabecal de Escaneo 3D instalado sobre el Robot

Este sensor instrumental (Figura 5-28) es el encargado de la toma de datos 3D para la reconstrucción del entorno. Se trata de un cabezal giratorio desarrollado en CARTIF, que está

integrado por un escáner láser *SICK LMS111* instalado sobre una plataforma giratoria compuesta por un motor DC, un encoder que sirve para conocer la posición angular del escáner láser, y un conector giratorio a través del cual pasan las alimentaciones y los datos del escáner.

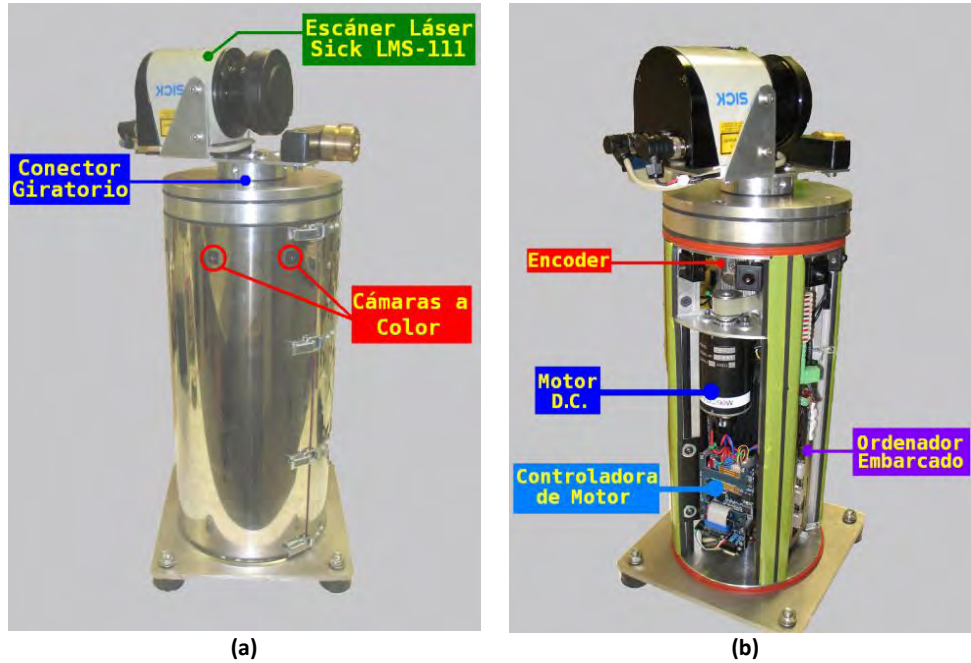


Figura 5-29: Componentes del Escáner 3D

Esta plataforma está montada sobre una estructura (ver Figura 5-29) en la cual se encuentran instaladas 8 cámaras fijas para el pegado de texturas sobre la información 3D. Dentro de esta estructura se encuentra el controlador para el motor DC, el microcontrolador y un ordenador para la captura de las imágenes de las cámaras. Este ordenador sirve asimismo para la sincronización de los datos de posición angular del escáner láser y las mediciones que proporciona. Una vez generada la nube de puntos, el ordenador la transmite al robot utilizando una comunicación FTP. La velocidad de giro de la plataforma (hasta 5 rpm) permite obtener un escaneo completo del entorno en menos de 20 segundos para una resolución angular de 0,5 grados.

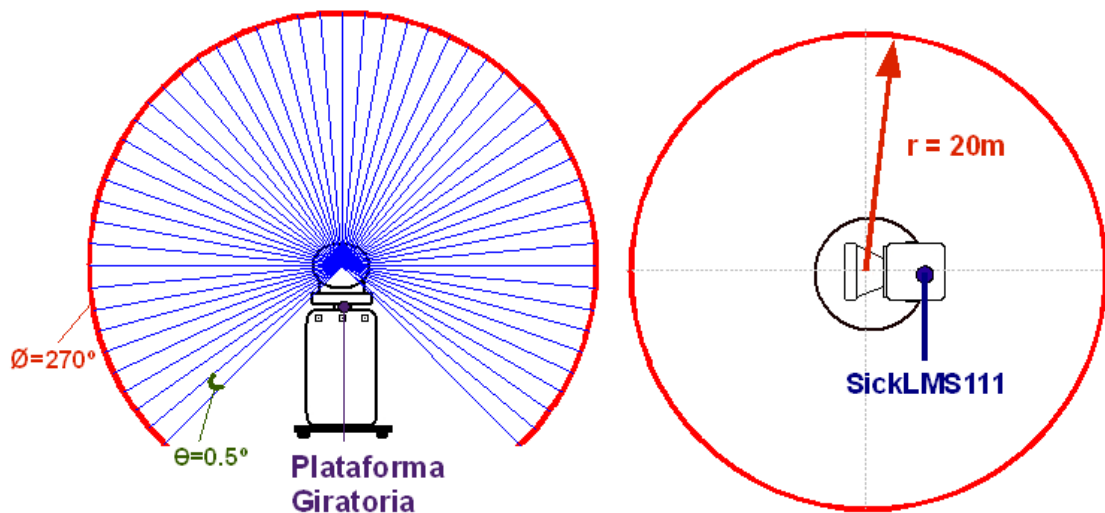


Figura 5-30: Representación del Escáner 3D

El principio de funcionamiento del escáner tridimensional es el siguiente: a medida que el cabezal va girando, el escáner láser va haciendo barridos verticales de la escena. La combinación de los dos movimientos angulares permite cubrir toda la escena en un radio de cincuenta metros (ver Figura 5-30), que es el alcance máximo del escáner láser.

El microcontrolador es el encargado de la sincronización de los datos. Este espera una orden proveniente del ordenador para poner a cero un temporizador interno, y una vez detectado el final de carrera que indica que el cabezal ha pasado por cero grados por primera vez, el empieza a transmitir cada 10 milisegundos la última posición del encoder y la velocidad del cabezal. Con esta información, el modulo del cabezal puede calcular por medio de una interpolación la posición angular de los datos del escáner láser, el cual envía los datos capturados cada 20 milisegundos.

El PC embarcado dentro del cabezal contiene el módulo de reconstrucción tridimensional, que se encarga de sincronizar los datos provenientes del escáner láser con las posiciones angulares leídas por el microcontrolador para generar el modelo tridimensional del entorno. Este módulo comanda la velocidad de giro del cabezal y una vez alcanzada esta velocidad, obtiene la información láser y la posición capturada por el microcontrolador instalado en el cabezal, para generar la nube de puntos (ver Figura 5-31). También obtiene una serie de fotos para el texturizado, utilizando, las cámaras ubicadas en el cabezal.

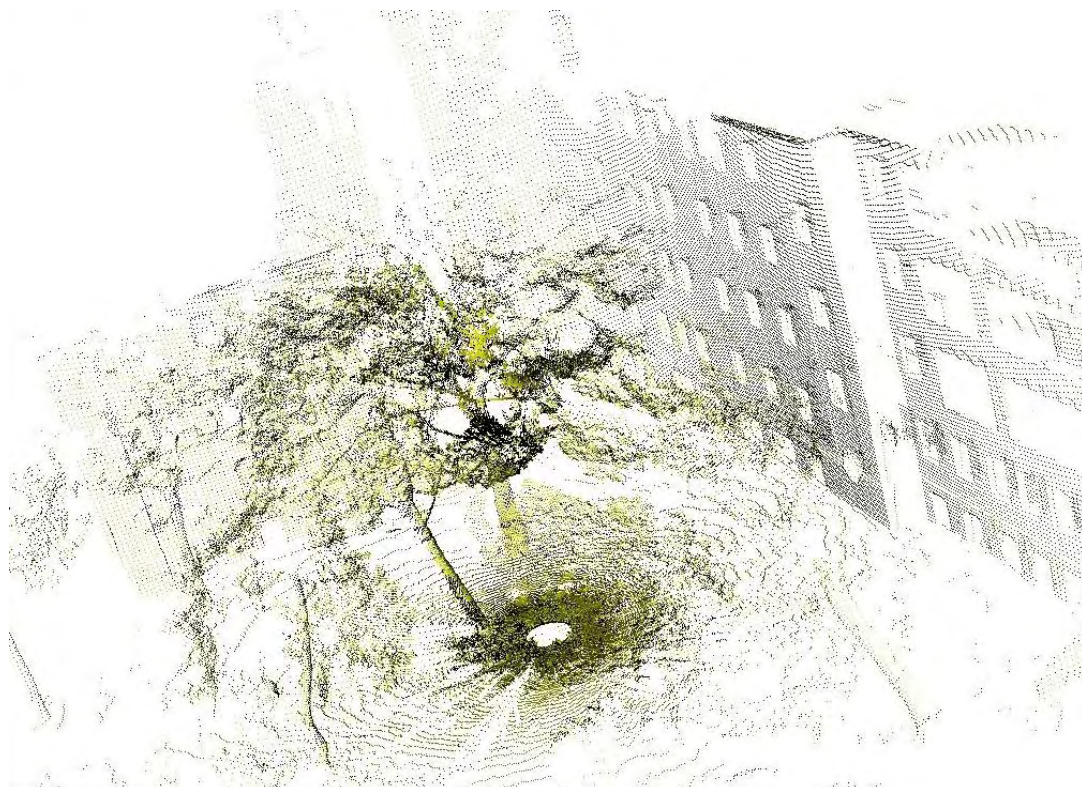


Figura 5-31: Nube de Puntos Capturada con el Escáner 3D

Una vez se ha capturado una nube de puntos usando el escáner 3D, se triangula el modelo para facilitar el análisis de la información tridimensional por parte de otras aplicaciones del sistema (como por ejemplo, por el módulo de exploración autónoma).

Cabe destacar que si bien el robot es capaz de capturar datos en movimiento, el procesamiento de estos datos resulta considerablemente más complejo que si se adquieren con el robot en posición estacionaria, puesto que se requiere extrapolar la posición del escáner (en los seis grados de libertad) para cada toma de datos. En consecuencia este sistema de medición incrementa los requerimientos computacionales y el error de medida, y complica el proceso de triangulación de la nube de puntos. Por todo ello resulta más recomendable la toma de datos en estacionario. Otro factor adicional que avala la elección de la medición en modo estacionario es que gracias a la rapidez con la que el escáner puede reconstruir el entorno, la diferencia en términos de tiempos entre la captura estacionaria y la captura con el robot en movimiento es escasa.

5.4. Simulador

El desarrollo de un algoritmo de exploración requiere múltiples pruebas en diferentes tipos de entornos, para garantizar que sea suficientemente válido y flexible para ser utilizado en escenarios diversos. Sin embargo, la realización de estas pruebas no es en modo alguno sencilla. Se requiere, en primer lugar, contar con una amplia variedad de escenarios disponibles. Además, es necesario dedicar una gran cantidad de tiempo para realizar diferentes experimentos, con el robot en estos entornos. Contar con estas dos condiciones resulta costoso y por tanto, es posible afirmar que la valoración del algoritmo de exploración en diferentes entornos resulta una de las fases de la investigación más costosa y difícil de afrontar.

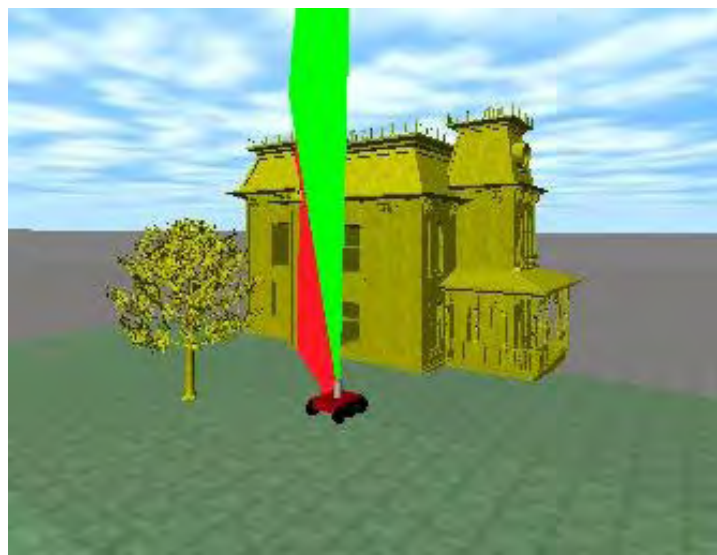


Figura 5-32: Captura de Pantalla del Simulador desarrollado

Los problemas citados pueden evitarse (o al menos reducirse) en las etapas previas mediante la utilización de simuladores. La simulación se convierte entonces en una herramienta esencial, pues permite al investigador disponer de muy variados entornos para la validación del

algoritmo, al tiempo que se reduce al mínimo las dificultades que pueden presentarse en ambientes exteriores, manteniendo la confiabilidad de los resultados. De esta manera, la validación en ambientes exteriores se deja para una última etapa de verificación del algoritmo, evitando así tener que hacer múltiples repeticiones y pruebas con distintas configuraciones de parámetros. Sin embargo, el simulador por sí solo no es una herramienta útil para la validación final del algoritmo, ya que aunque ha sido diseñado con gran cantidad de detalle, la información capturada sobre un entorno real está sometida a diferentes niveles de ruido y los entornos reales tienen más elementos que pueden generar sombras sobre la información 3D.

Por estos motivos, como parte de la investigación se ha desarrollado un simulador robótico tridimensional sobre el cual se pueden realizar múltiples pruebas a un costo muy reducido (ver Figura 5-32).

El desarrollo ha sido realizado en C++ usando las librerías ODE (Smith, 2007) para la simulación de variables físicas y el cálculo de los puntos escaneados desde diferentes vistas. La carga y tratamiento de datos 3D se realiza utilizando las librerías Trimesh2 (Rusinkiewicz). Igualmente, para facilitar el paso de los algoritmos desarrollados desde un entorno simulado hacia un sistema real, se han utilizado las librerías IPC “Inter Process Communication libraries” (Simmons, 2009) que permiten integrar el simulador con otros programas de control.

5.4.1. Características del Simulador

El simulador está diseñado en principio el caso de un robot desplazándose en un entorno tridimensionalmente complejo, pero en caso necesario también es posible simular más de un robot en forma simultánea. En relación con el desarrollo de actividades simuladas, es muy importante destacar la existencia de cuatro elementos principales que deben ser definidos y situados sobre un mundo simulado vacío (Figura 5-33). Estos son: el terreno, el robot, los sensores y los elementos tridimensionales del entorno.

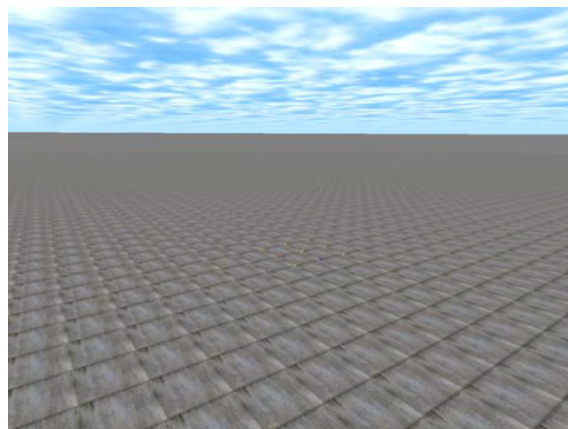


Figura 5-33: Mundo Simulado vacío

Simulación de Terrenos

El terreno en el simulador es el suelo sobre el cual se desplaza el robot. Se define mediante un mapa de elevaciones de forma cuadriculada que puede ser ajustado a la resolución necesaria, y el cual se carga en el simulador desde un fichero de configuración. Un ejemplo de esta característica puede ser visto en la Figura 5-34



Figura 5-34: Terreno Simulado

Simulación de Elementos 3D

El simulador tiene la posibilidad de incluir en el escenario elegido cualquier elemento estático, ya que puede cargar diferentes tipos de ficheros de modelos 3D usados en múltiples aplicaciones. Estos elementos son útiles para representar objetos inamovibles dentro del entorno (por ejemplo edificaciones) y además pueden contar con un elevado nivel de detalle.

Por ejemplo, para cargar un fichero 3D (que puede ser de tipo *ply*, *dae*, *off*, 3DS u *obj* creados con Wavefront) tan solo hace falta definir un objeto en el fichero de configuración del entorno con la ruta al fichero 3D, las coordenadas de posición del objeto dentro del entorno, la rotación del mismo con respecto a sus coordenadas originales, el color RGB deseado para representar la malla y la escala del modelo dentro del entorno. A continuación se puede ver la forma de definir un elemento tridimensional en el fichero de configuración del entorno:

Campo de definición de un elemento 3D en el fichero de configuración del entorno simulado
<pre>3D Model: File: /Ruta_al_Fichero/Modelo.ply Position: coordX[m]; coordY[m]; coordZ[m] Rotation: angX[rad]; angY[rad]; angZ[rad] Color: R[0,1]; G[0,1]; B[0,1] Scale: 1 end</pre>

Esta forma de definir la manera de cargar los modelos 3D resulta útil porque permite utilizar cualquier modelo fácilmente en cualquier parte del entorno sin necesidad de modificarlo.

Además, la posibilidad de cambiar la escala y rotación de un mismo modelo permite darle mayor variedad al entorno. Un ejemplo de esto se muestra en la Figura 5-35 donde un mismo modelo ha sido cargado tres veces con escalas diferentes de manera que parecen objetos diferentes.

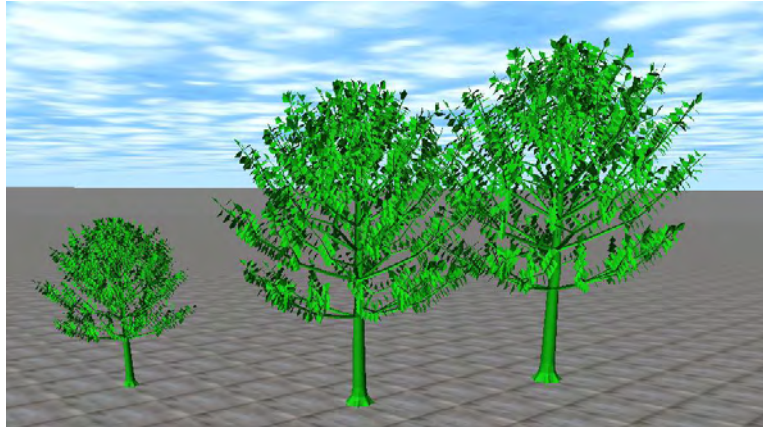


Figura 5-35: Modelo 3D de un árbol reproducido en simulador usando escalas diferentes

De esta forma, se pueden definir muchos tipos de objetos como se puede ver en la Figura 5-36.

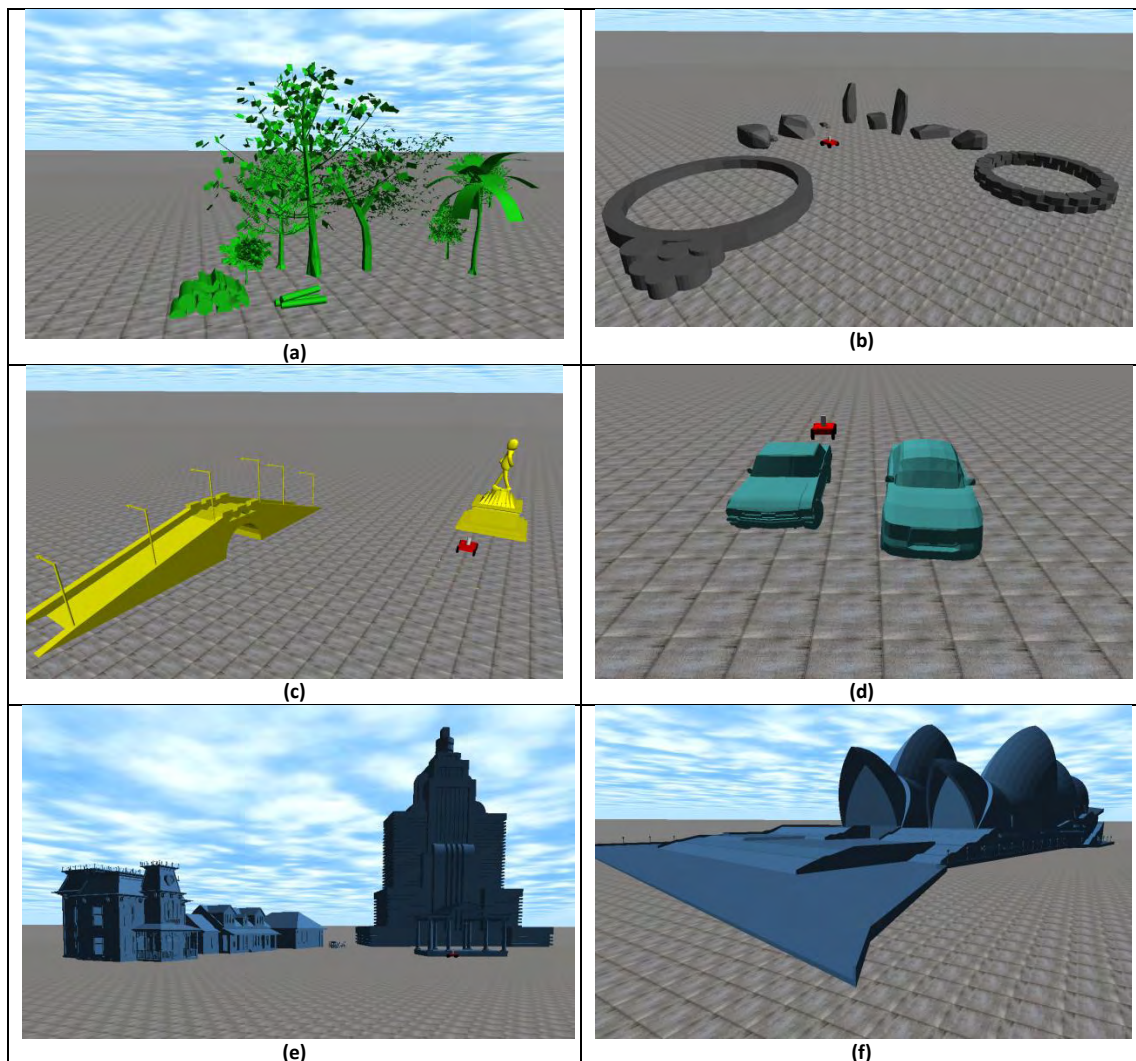


Figura 5-36: Ejemplos de distintos tipos de elementos que se pueden cargar en el simulador.

Simulación del Robot y Sensores Embarcados

El robot se puede definir en un fichero en el cual se especifican las dimensiones y la geometría del cuerpo, el número de ruedas, la posición y el color de cada uno de estos elementos. Igualmente pueden incluirse en este fichero los elementos añadidos como equipos de medición o manipuladores que vayan integrados en el robot.

Para incluir un robot en el simulador se deben definir tres campos: **Body**, **Wheels** y **Position**. También es posible incluir un cuarto campo llamado **Equipment** en el cual se define el equipamiento del robot y la posición de anclaje sobre el mismo.

Campo de definición de un Robot en el entorno simulado
<pre>Robot: Body: # Definición del cuerpo del robot end Wheels: # Definición de las ruedas del robot end Position: coordX[m]; coordY[m]; coordZ[m]; angX[rad]; angY[rad]; angZ[rad] Equipment: # Definición de los equipos del robot end end</pre>

Definición del cuerpo del robot

El cuerpo del robot se define creando una lista de vértices y caras que lo conforman, las cuales se consignan en los campos de **Vertex** y **Faces**. Además se puede definir el color para el cuerpo en el campo **Color**, por ejemplo:

Definición del Cuerpo del Robot
<pre>Vertex: X1, Y1, Z1 X2, Y2, Z2 X3, Y3, Z3 Faces: V1, V2, V3 Color: R, G, B</pre>

Definición de las ruedas del robot

Las ruedas del robot se definen independientemente, especificando los valores para los campos **Radius**, **Width**, **Type**, **Orientation**, **Connection_points** y **Color**. Las ruedas son cilindros

cuyo punto de origen (0,0,0) se encuentra en el centro. Esto es importante ya que para definir el campo **Connection_points** se necesita definir un punto relativo al cuerpo del robot y un punto sobre la rueda¹ que servirán de conexión entre ambos elementos.

Para el campo **Type** existen tres tipos de ruedas, cada uno definido por un número con el siguiente significado:

Tipo 1: Son ruedas locas, que carecen de tracción propia y giran de acuerdo al movimiento del robot únicamente sobre el eje X de las mismas.

Tipo 2: Son ruedas locas, que carecen de tracción propia y giran de acuerdo al movimiento del robot sobre los ejes X y Z de las mismas.

Tipo 3: Son ruedas con control de tracción propia y giran únicamente sobre su eje X. Este tipo de ruedas son las utilizadas en un robot de tracción diferencial.

Tipo 4: Son ruedas con control de tracción controlando la velocidad de giro sobre su eje X y con control de dirección a través del ángulo de giro de la rueda sobre su eje Z.

El número de ruedas del robot depende de la cantidad de campos para ruedas (**Wheel**) que se definen dentro del espacio para la definición de ruedas (**Wheels**), de manera que el número de ruedas es variable (ver Figura 5-37).

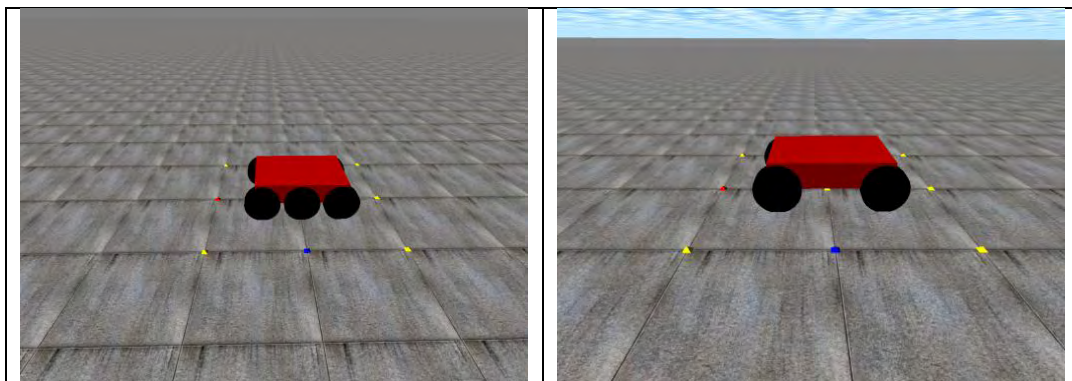


Figura 5-37: Cuerpo del robot con seis (izquierda) o cuatro (derecha) ruedas

A continuación se muestra un ejemplo de definición de ruedas:

Definición de las Ruedas del Robot	
Wheels:	
Wheel:	# Definición de una rueda
Radius:	radio[m] # Radio de la rueda
Width:	Ancho[m] # Ancho de la rueda
Type:	Tipo
Orientation:	

¹ El punto definido sobre la rueda siempre debe estar sobre el eje X ya que es el eje que se corresponde con el eje de giro de la rueda.


```

    angX[rad]; angY[rad]; angZ[rad]
    Connection_Points:
    coordX[m]; coordY[m]; coordZ[m]; # Punto de conexión sobre el
                                     # cuerpo del robot
    coordX[m]; coordY[m]; coordZ[m]; # Punto de conexión sobre la
                                     # rueda

    Color:
    R, G, B
end
Wheel:
# Definición de otras ruedas del robot
end
end

```

Definición de los sensores del Robot

Igualmente se pueden definir sensores embarcados en el robot. Por el momento se han previsto cuatro tipos de sensores: *cameras*, *Laser1D*, *Laser2D*, *Laser3D*). Los sensores tipo *Laser1D* son un rayo que parte desde el punto identificado y retorna la distancia al primer objeto que alcanza. Análogamente operan los sensores tipo *Laser2D*, pero en este caso se crean una serie de rayos que parten desde un punto central predefinido. El numero de rayos depende del campo de cobertura definido o *FOV* ("field of view") y de la resolución. Así por ejemplo para un sensor con un *FOV* de 180° y resolución de medio grado se crearán 360 rayos.

El sensor tipo *Laser3D* es en realidad un sensor de tipo *Laser2D* que gira sobre el eje Z del sensor, entre dos ángulos previamente definidos. Finalmente el sensor tipo camera es un sensor que guarda una imagen desde un punto de vista del robot.

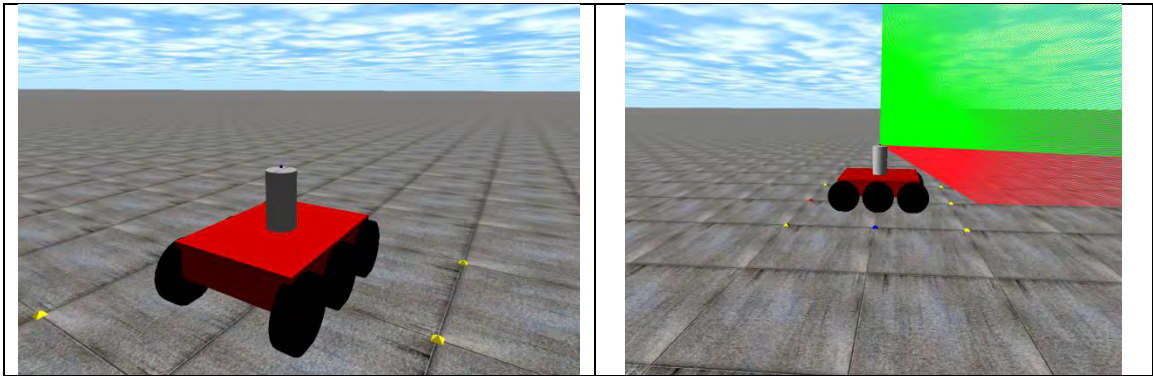


Figura 5-38: Cuerpo de un Laser3D simulado

Para todos los sensores se puede definir un cuerpo como en el caso del robot (ver Figura 5-38), sobre el cual se incorporan al mismo. El cuerpo se define creando los campos *Vertex*, *Faces*, *Color*, *Orientation* y *Connection_points*. Tras definir el campo *Body*, se definen cada uno de los sensores como muestra el siguiente ejemplo:

Definición de los sensores del Robot
Equipment: Body: Vertex:

```

X1, Y1, Z1
X2, Y2, Z2
X3, Y3, Z3
Faces:
  V1, V2, V3
Color:
  R, G, B
Orientation:
  angX[rad]; angY[rad]; angZ[rad]
Connection_Points:
  coordX[m]; coordY[m]; coordZ[m]; # Punto de conexión sobre el
                                   # cuerpo del robot
  coordX[m]; coordY[m]; coordZ[m]; # Punto de conexión sobre el
                                   # cuerpo de los sensores
end
Camera:
  Position: # Posición con respecto al cuerpo del sensor
            coordX[m]; coordY[m]; coordZ[m]; angX[rad]; angY[rad]; angZ[rad]
end
Laser1D:
  Position: # Posición con respecto al cuerpo del sensor
            coordX[m]; coordY[m]; coordZ[m]; angX[rad]; angY[rad]; angZ[rad]
  Range:
    Alcance[m] # Alcance máximo del rayo
end
Laser2D:
  Position: # Posición con respecto al cuerpo del sensor
            coordX[m]; coordY[m]; coordZ[m]; angX[rad]; angY[rad]; angZ[rad]
  Range:
    Alcance[m] # Alcance máximo del rayo
  Resolution:
    Resolución [rad] # Resolución del escáner
  FOV:
    Cobertura [rad] # Apertura del escáner 2D
end
Laser3D:
  Position: # Posición con respecto al cuerpo del sensor
            coordX[m]; coordY[m]; coordZ[m]; angX[rad]; angY[rad]; angZ[rad]
  Range:
    Alcance[m] # Alcance máximo del rayo
  Resolution:
    Resolución [rad] # Resolución del escáner
  FOV:
    Cobertura [rad] # Apertura del escáner 2D
  Angles:
    Angmin[rad]; AngMax[rad] # Angulos de inicio y final del giro
                             # del escáner 2D sobre el eje Z
end
end
end

```

5.5. Conclusiones del Capítulo

En este capítulo se han presentado las herramientas desarrolladas para la validación del algoritmo junto con las motivaciones para su desarrollo, surgidas del análisis de las soluciones equivalentes existentes.

En concreto, se ha desarrollado un robot todoterreno dotado de múltiples sistemas perceptivos orientados a la teleoperación, localización y navegación del sistema. A lo largo del capítulo se han detallado sus características físicas y mecánicas, los equipos de comunicaciones y los distintos sensores integrados en el robot. Igualmente se ha presentado la arquitectura hardware/software desarrollada para el control del robot. El robot en sí resulta de utilidad para tareas de rescate en zonas de difícil accesibilidad o peligrosas que permitan operación autónoma o teleoperada. Igualmente, en situaciones de emergencia el robot puede resultar de gran utilidad gracias a su rapidez y a su capacidad para ser supervisado a grandes distancias.

Asimismo se ha desarrollado un sistema de escaneado 3D, que es la herramienta principal para la generación de modelos del entorno. Este sistema ha demostrado ser fiable y capaz de generar modelos a un bajo coste en comparación con otras alternativas. Además, vale la pena destacar que ha sido diseñado específicamente para que pueda funcionar independientemente de cualquier otro sistema, aunque en el presente trabajo se ha utilizado siempre instalado sobre el robot todoterreno.

La tercera herramienta desarrollada, el simulador tridimensional, ha demostrado ser sumamente útil para la validación de la metodología desarrollada. Una diferencia muy importante con respecto a los simuladores disponibles en la actualidad es la capacidad de cargar y simular terrenos complejos de múltiples elevaciones, además de que permite al robot simulado interactuar con elementos tridimensionales complejos.

Capítulo 6.

RESULTADOS

En este capítulo se presentan y discuten los resultados de la presente investigación sobre la obtención de modelos tridimensionales de entornos de exterior mediante un robot móvil.

En primer lugar, se exponen algunas consideraciones sobre el ajuste de los parámetros del algoritmo propuesto para la exploración del entorno, así como sobre los criterios de parada de la exploración. A continuación, se presentan los resultados correspondientes a distintos niveles de ejecución. La experimentación se ha realizado con entornos simulados, mediante el

Se contemplan dos grandes grupos de experimentos: sobre entornos simulados y sobre entornos reales. En el primer caso se ha utilizado el simulador desarrollado dentro de la presente tesis y que fue presentado en el capítulo 5. Para la experimentación en entornos reales se ha utilizado el robot todoterreno descrito en el mismo capítulo, junto con el correspondiente escáner 3D,

6.1. Ajuste de los Parámetros

La Tabla 6-1 presenta los parámetros utilizados por el algoritmo de exploración, las etapas del mismo afectadas por cada parámetro, los valores típicos para cada parámetro y el efecto que tiene su modificación sobre el comportamiento del algoritmo. En la última columna de esta tabla se pueden ver los valores utilizados en los experimentos aquí presentados. Los valores asignados para los parámetros de ajuste de la ecuación 2.3 no se incluyen ya que han sido modificados según el experimento.

Tabla 6-1: Parámetros Usados en los Experimentos

Etapa del Algoritmo	Parámetro	Valores Típicos	Proceso Controlado	Efecto	Valor Experimental
Parámetros del Escáner 3D	res_p	0,25° - 1°	Adquisición del Modelo 3D	↓ Aumentara la Calidad del Modelo.	0,5°
	res_t	0,25° - 1°		↓ Aumentara la Calidad del Modelo	0,5°
Parámetros del Análisis Tridimensional	w_l	0 - 1	Extracción de Superficies navegables. Ver Ecuación (3.18)	↓ Reducirá los outliers en el proceso de extracción de superficies navegables	0,66
	w_g	0 - 1		↑ Reducirá los bordes y las zonas aisladas de la zona de navegación segura	0,33
	w_{bl}	0 - 1	Extracción de Obstáculos. Ver Ecuación (3.21)	↓ Reducirá los outliers en el proceso de extracción de obstáculos.	0,55
	w_{bg}	0 - 1		↑ Filtrara los obstáculos pequeños	0,45
	ω_a	0 - 1	Análisis de la Calidad del Modelo. Ver Ecuación 3.27)	↓ Aumentara los requerimientos de calidad del modelo.	0,93
	ω_n	0 - 1		↑ Filtrara los obstáculos pequeños	0,07
	A_{max}	0,0001 m ² - 0,02 m ²		↓ Incrementara los requerimiento de calidad del modelo	0,0025m ²
	ϵ_f	0 - 1	Mapa de Navegación 2D	Valor del Offset para considerar un punto como zona de navegación segura en la creación del mapa de navegabilidad	0,65
ϵ_o	0 - 1	Valor del Offset para considerar un punto como obstaculo en la creación del mapa de navegabilidad		0,83	
Mapa de Información 2D	Tamaño de las celdas	0,01 m ² - 1 m ²	Creación del Mapa de Información 2D	↑ Reduce el detalle del modelo, mejora la absorción de los fallos de alineación en la creación del mapa de información 2D.	0,3m x 0,3m
Evaluación de Candidatos	w_A	0 - 1	Ganancia Esperada de la Información. Ecuación (2.3)	↑ Aumenta la importancia de la ganancia esperada de información en la evaluación de candidatos.	Según Experimento

w_Q	0 – 1	Ganancia Esperada de la Calidad del Modelo. Ecuación (2.3)	↑ Aumenta la importancia de la ganancia esperada de calidad en la evaluación de candidatos.	Según Experimento
w_O	0 – 1	Cobertura de las Zonas de Interés. Ecuación (2.3)	↑ Aumenta la importancia de la cobertura de zonas de interés.	Según Experimento
w_C	0 – 1	Estimación de los costos de Navegación. Ecuación (2.3)	↑ Los candidatos a mayores distancias son penalizados.	Según Experimento

Los parámetros más relevantes son los utilizados para la evaluación de candidatos (ecuación 2.3) ya que afectan al comportamiento del algoritmo. Los parámetros w_A , w_Q y w_O pueden tener valores entre 0 y 1 proporcionalmente a la influencia deseada de cada término.

Asignar valores superiores a 0,25 para w_O no es recomendable ya que se ha observado que el algoritmo puede comportarse de forma inestable. Esto es debido a que cuando las zonas ocultas tienen demasiado peso en la elección de un punto de captura se pueden generar bucles en la trayectoria, ya que en todos los escaneos se extraen zonas de interés y en las zonas con estructuras complejas, se puede mantener al robot concentrado en una única zona del mapa.

El parámetro w_c también puede tomar cualquier valor. Sin embargo, una forma fácil de calcular un valor para este parámetro es tomar la inversa de la distancia ideal (por ejemplo la distancia de calidad ideal del escáner utilizada para calcular A_{max} ecuación 3.24 hasta el siguiente escaneo; por ejemplo, 0,2 para tomas cada cinco metros).

6.2. Entornos Simulados

Se ha validado el algoritmo de exploración en tres entornos simulados. El primero de ellos es un entorno relativamente simple que cuenta con una casa y un árbol dispuestos sobre una colina ligeramente irregular. Este escenario ha sido utilizado para el desarrollo del algoritmo, con el propósito específico de evaluar cómo los parámetros pueden afectar al comportamiento del algoritmo y en qué forma las pendientes afectan al análisis de navegabilidad y la estimación del coste de la trayectoria.

Los otros dos entornos son notablemente más complejos. En concreto, el presentado en segundo lugar es un entorno estructurado de tipo urbano, diseñado para ensayar como se movería el robot en un entorno donde la complejidad de navegación no se encuentra en la inclinación del terreno sino en la gran cantidad de objetos presentes en el mismo. Igualmente,

este escenario es ideal para comprobar el comportamiento del algoritmo cuando es posible realizar una gran cantidad diferente de trayectorias.

El tercer entorno es el que presenta un mayor grado de complejidad. Se trata de un escenario desestructurado, con múltiples obstáculos y grandes inclinaciones que complican considerablemente la navegación. Adicionalmente, esta profusión de obstáculos y de elementos en el terreno ocasiona que se generen muchas sombras y oclusiones, dificultando la adquisición de datos 3D. El objetivo buscado en el ensayo con de este tipo de escenario es verificar si el algoritmo puede escoger posiciones candidatas seguras y que garanticen la mejor toma de datos posible según el peso de los criterios seleccionados.

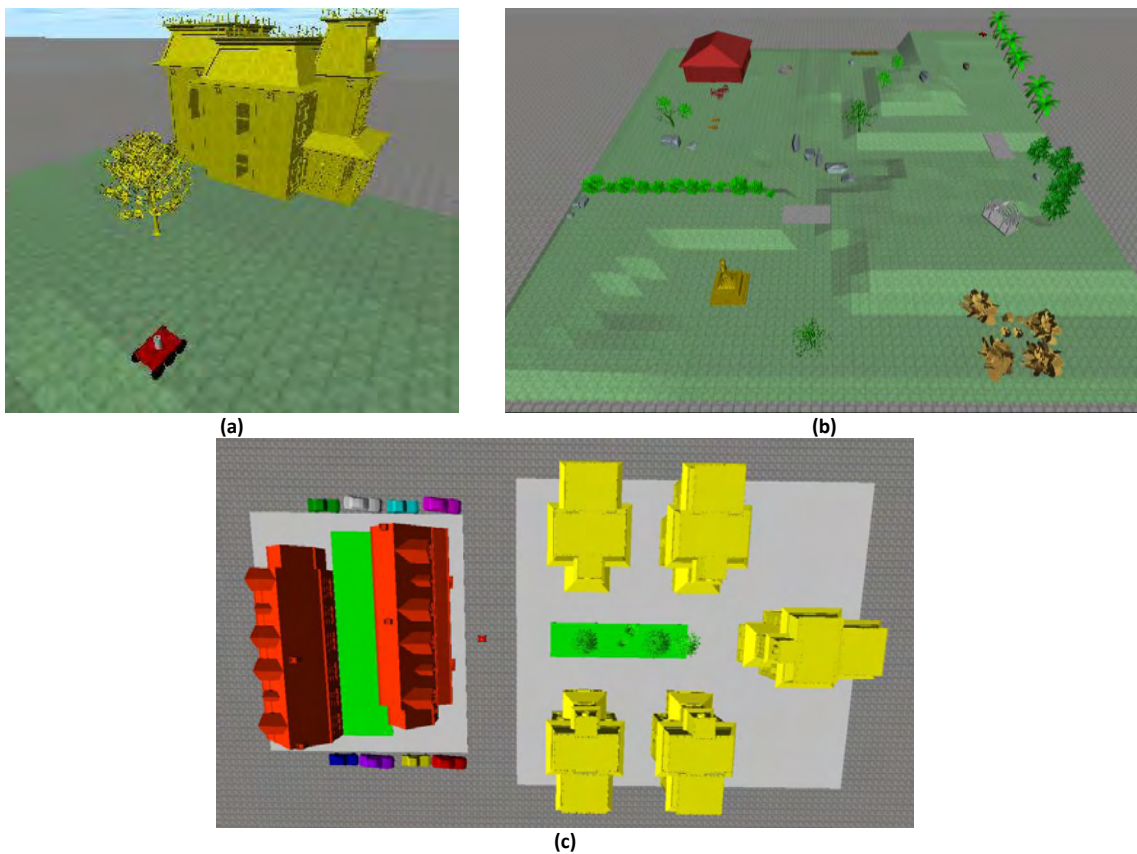


Figura 6-1: Entornos simulados; (a) Entorno Simple; (b) Entorno Desestructurado; (c) Entorno Estructurado.

6.2.1. Entorno Simple

A continuación se muestran tres experimentos diferentes realizados en este entorno (ver Figura 6-2). En cada uno de ellos se ha utilizado una configuración diferente para los parámetros de la ecuación 2.3 con el propósito de conseguir distintos objetivos.

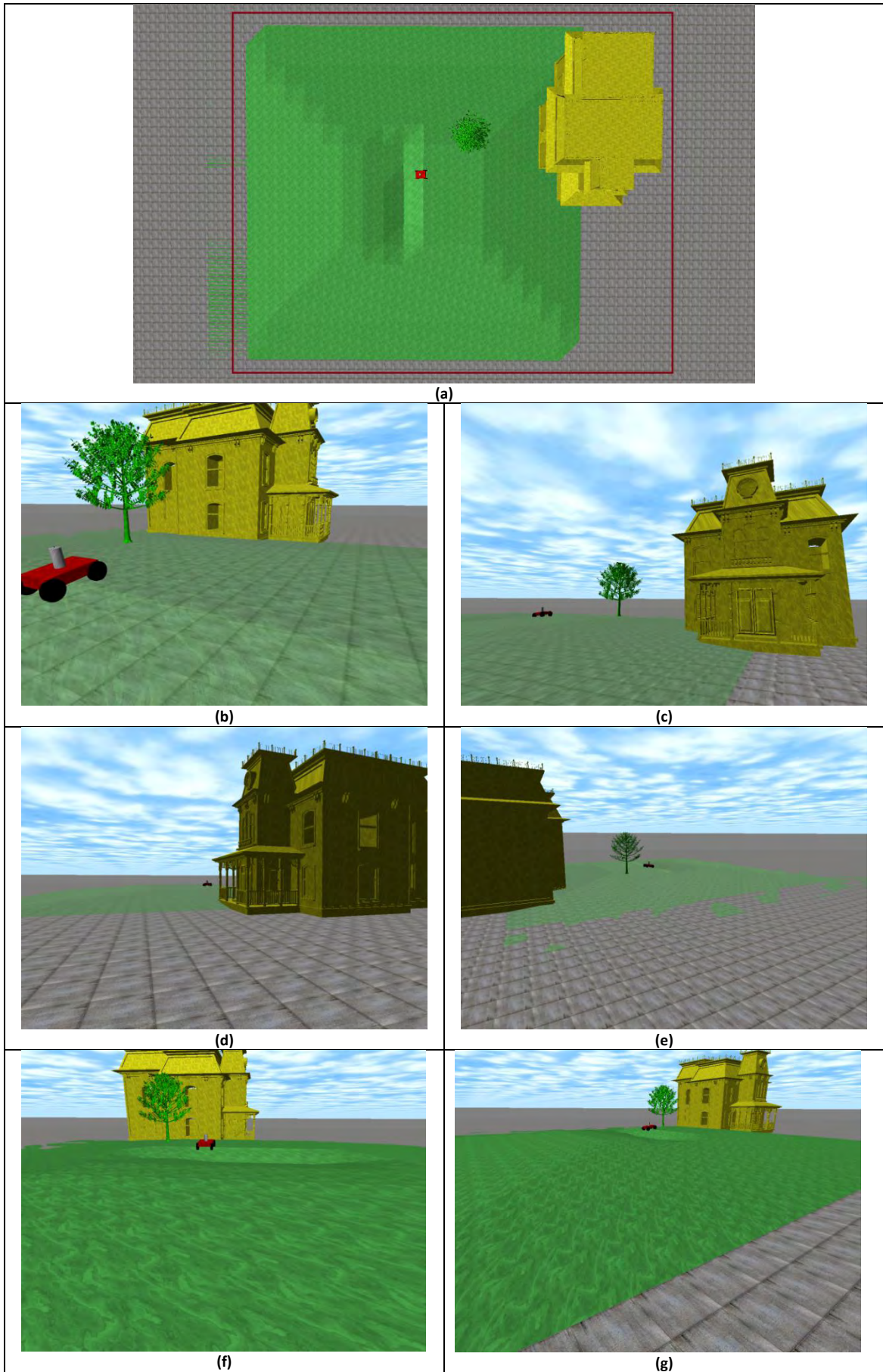


Figura 6-2: Capturas del entorno simple simulado, el área de exploración está limitada por el rectángulo rojo.

En el primer experimento, los parámetros se han configurado de forma equilibrada para que ningún criterio se imponga por encima de los otros. En el segundo experimento, la ganancia de información fue el criterio dominante mientras que en el tercer experimento se consideró como criterio dominante la calidad del modelo.

Para ningún experimento se modificó el parámetro w_c , con objeto de demostrar la capacidad del algoritmo para evaluar las posiciones candidatas de acuerdo a su utilidad y seguir los criterios dominantes según la elección de parámetros. Los valores concretos de los parámetros utilizados en cada experimento se muestran en la Tabla 6-2.

Tabla 6-2: Parámetros Usados en los Experimentos en el Entorno Simple

	w_A	w_Q	w_O	w_C
Experimento 6.2.1.I	0,43	0,35	0,22	0,09
Experimento 6.2.1.II	0,65	0,25	0,10	0,09
Experimento 6.2.1.III	0,36	0,5	0,14	0,09

Las trayectorias resultantes para los experimentos se muestran en la Figura 6-3. En dicha figura, se puede ver la trayectoria realizada en cada experimento (en magenta) superpuesta sobre el mapa de calidad extraído del entorno.

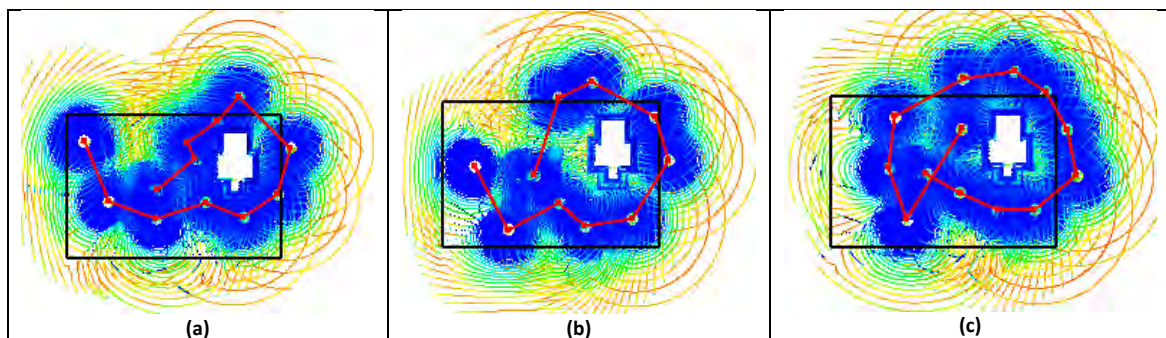


Figura 6-3: Trayectorias resultantes para el Experimento 6.2.1.I (a) parámetros equilibrados entre ganancia de información y calidad del modelo; Experimento 6.2.1.II (b) se prima la ganancia de información; Experimento 6.2.1.III (c) se prima la calidad del modelo. Los colores más fríos representan mejores calidades por celda.

Igualmente se ha realizado un cuarto experimento en el cual los puntos de escaneo son seleccionados por un algoritmo del tipo “greedy mapping” como el de (Koenig, Tovey , & Halliburton, 2001), seleccionando la posición que mayor cantidad de información nueva pueda proveer sobre el mapa de información 2D. La trayectoria resultante se muestra en la Figura 6-4.

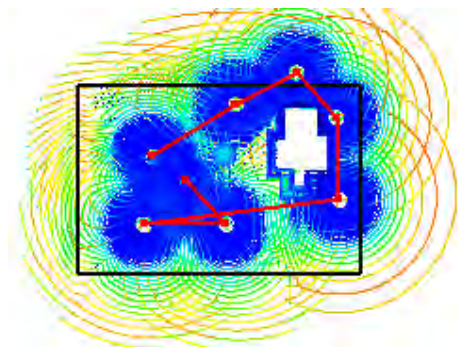


Figura 6-4: Trayectoria resultante para el algoritmo de greedy mapping Experimento 6.2.1.IV. Los colores más fríos representan mejores calidades por celda.

La Tabla 6-3 muestra los resultados obtenidos para los cuatro experimentos en términos de distancia recorrida, número de escaneos, porcentaje de celdas alcanzables escaneadas y calidad por celda promedio.

Tabla 6-3: Resultados Obtenidos en los Experimentos

	Distancia Recorrida	Puntuación de Calidad	Cobertura	Numero de Escaneos
Experimento 6.2.1.I	106 m	0,6846	92,5 %	12
Experimento 6.2.1.II	107 m	0,6904	94,1 %	10
Experimento 6.2.1.III	128 m	0,7503	93,6 %	13
Experimento 6.2.1.IV	116 m	0,5013	91,4 %	8

A partir de estos resultados se observa que el Experimento 6.2.1.II es la solución más eficiente, ya que ha cubierto el área de trabajo usando solo diez escaneos, viajando tan solo un metro más que la trayectoria más corta. El Experimento 6.2.1.I obtuvo un resultado muy similar pero usando un número de escaneos mayor. Por último, el Experimento 6.2.1.III necesitó trece escaneos para cubrir todo el entorno; sin embargo la calidad del modelo resultante fue muy elevada.

En comparación con el algoritmo de “*greedy mapping*” (Experimento 6.2.1.IV), el método propuesto requiere más escaneos para alcanzar el mismo nivel de cobertura. Sin embargo, alcanza una puntuación de calidad mucho más alta.

La diferencia en términos de distancia recorrida no es significativa, pero la trayectoria realizada por el método de “*greedy mapping*” es menos eficiente ya que requiere giros más bruscos para alcanzar los objetivos y recorridos muy largos entre puntos de escaneo. En relación con la calidad del modelo, el método propuesto ofrece mejores resultados que los obtenidos empleando el algoritmo de “*greedy mapping*” lo que se evidencia en los parámetros de calidad que logran sostenerse de una manera regular a lo largo del mapa.

La Figura 6-5 muestra una captura del modelo tridimensional obtenido en el Experimento 6.2.1.I.

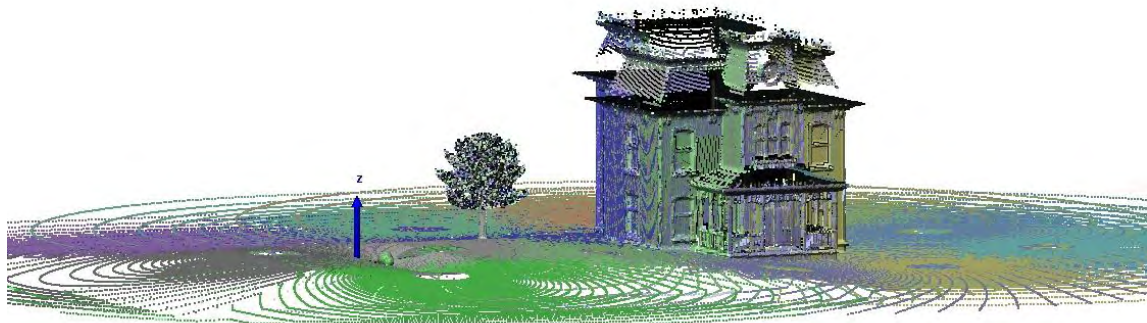


Figura 6-5: Captura de la reconstrucción del modelo.

6.2.2. Entorno Estructurado

La Figura 6-6 muestra el entorno estructurado simulado compuesto por edificaciones, calles con coches aparcados y zonas ajardinadas.

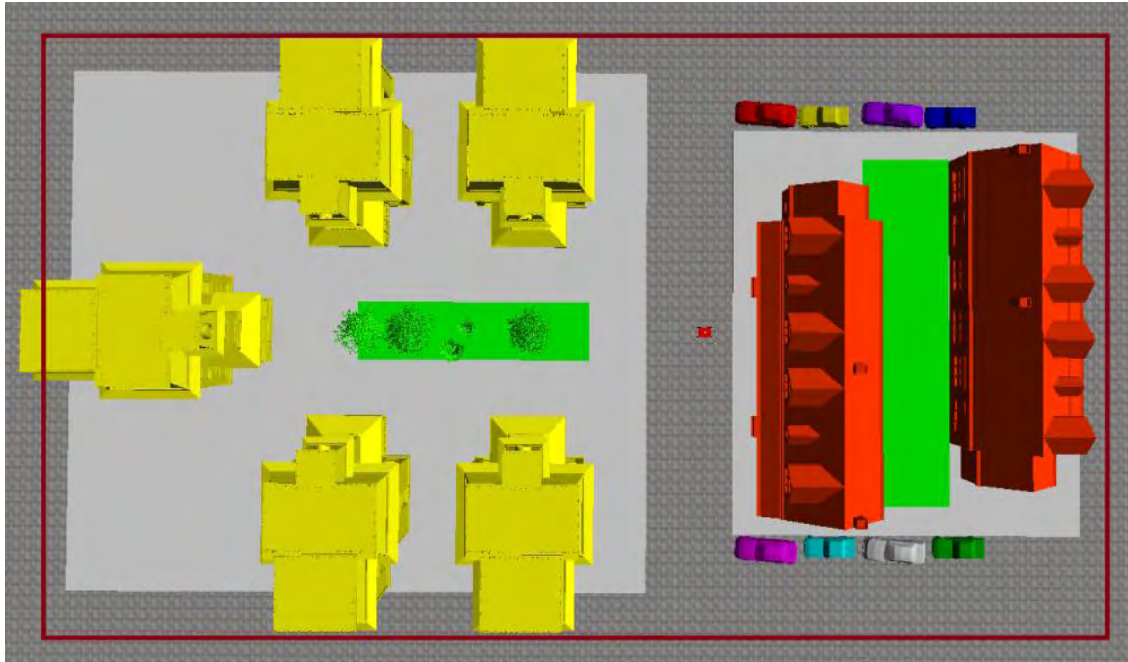


Figura 6-6: Vista aérea del Entorno Estructurado

En los diferentes recuadros de la Figura 6-7 se puede observar detalladamente las características del entorno. Lo primero destacable es que en este caso, el terreno es plano y no existen alturas destacables más allá de los bordes de los arcones.

Otro aspecto destacable es que existen rutas claras de navegación y que hay múltiples elementos de gran tamaño. El objetivo de simular este entorno es verificar la capacidad de esta metodología de manejar este tipo de entornos y especialmente, ver si el algoritmo seleccionaría trayectorias que permitieran escanear los pasos cubiertos por grandes elementos (como los espacios entre casas), motivo por el cual se integró el criterio de las zonas de interés dentro del procedimiento.

Para este escenario se muestran dos experimentos que siguen dos estrategias distintas. En el primer experimento, las zonas de interés y la calidad del mapa son los criterios dominantes; en el segundo, la ganancia esperada de información es el criterio dominante. Igualmente en este segundo experimento se ha reducido el coste de navegación ligeramente, para estimular recorridos más largos.

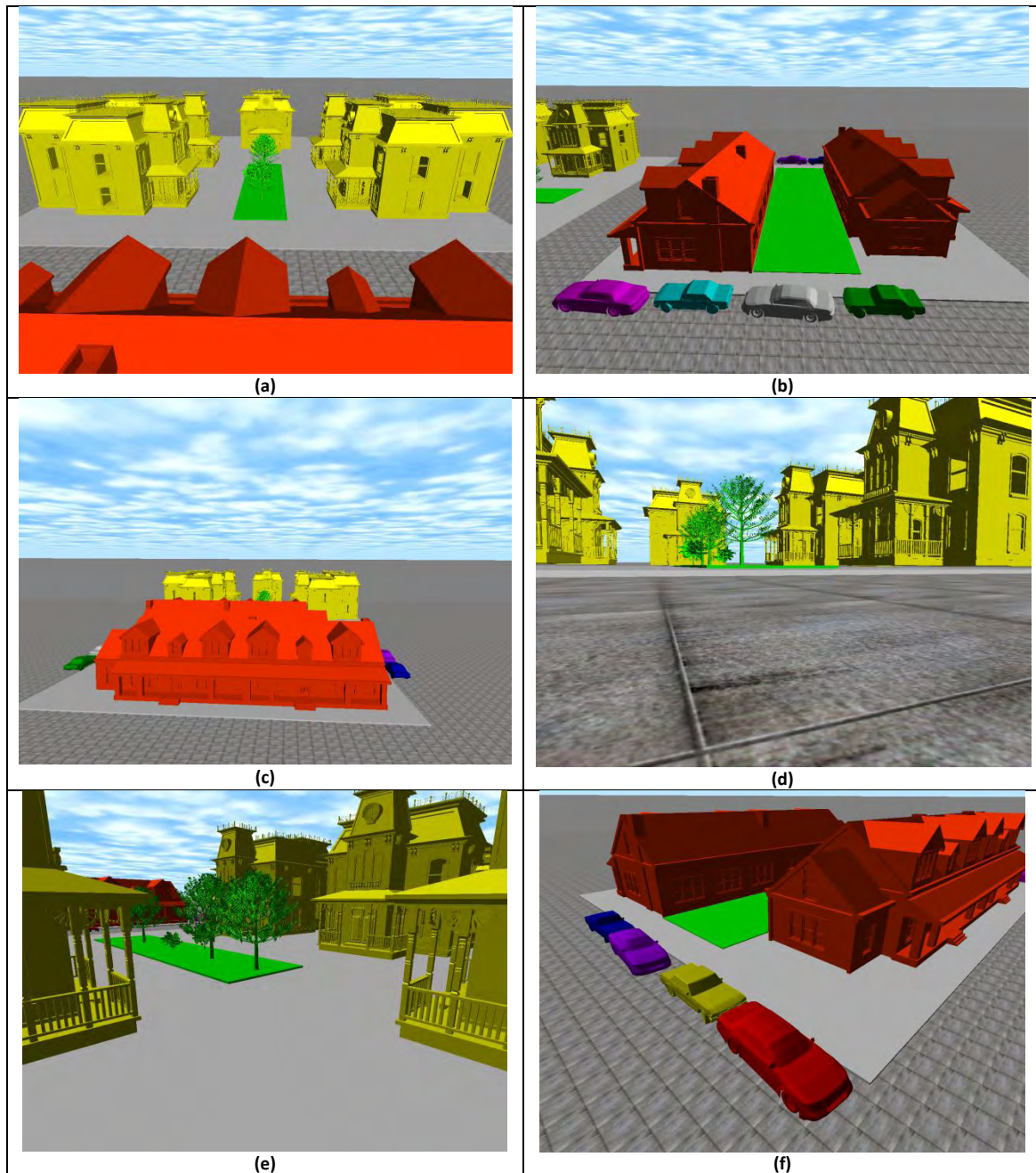


Figura 6-7: Vistas varias del Entorno Estructurado

La Tabla 6-4 muestra los parámetros utilizados en los experimentos y la Figura 6-8 muestra las trayectorias obtenidas en ambos casos.

Tabla 6-4: Parámetros Usados en los Experimentos

	W_A	W_Q	W_O	W_C
Experimento 6.2.2.I	0,2	0,55	0,25	0,09
Experimento 6.2.2.II	0,65	0,25	0,10	0,08

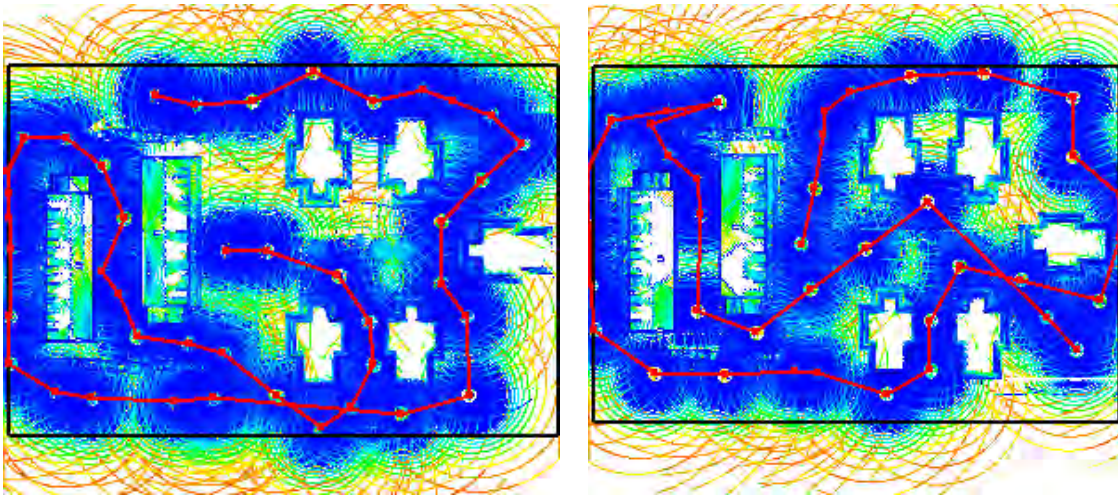


Figura 6-8: Trayectorias Obtenidas en el Experimento 6.2.2.I (Izquierda) y el Experimento 6.2.2.II (Derecha). Los colores más fríos representan mejores calidades por celda.

La Tabla 6-5 muestra los resultados obtenidos para los experimentos en términos de distancia recorrida, número de escaneos, porcentaje de celdas alcanzables escaneadas y el promedio de calidad por celda.

Tabla 6-5: Resultados Obtenidos en los Experimentos

	Distancia Recorrida	Puntuación de Calidad	Cobertura	Numero de Escaneos
Experimento 6.2.2.I	352 m	0,6813	95 %	43
Experimento 6.2.2.II	397 m	0,726	94,8 %	39

Estos resultados demuestran que el algoritmo puede ser usado eficazmente en entornos estructurados. Se puede observar que ambas estrategias tienen resultados similares en términos de cobertura y puntuaciones de calidad. La diferencia en términos de distancia recorrida en el Experimento 6.2.2.II es compensada por un menor número de escaneos.

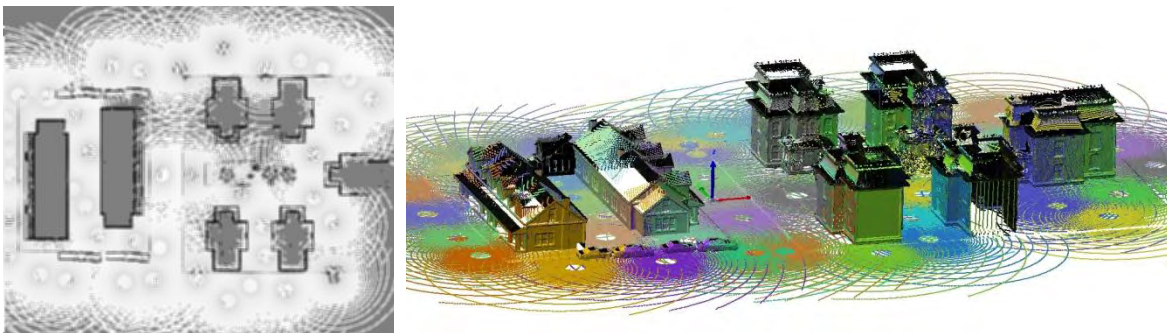


Figura 6-9: Mapa de Navegabilidad (Experimento 6.2.2.I, Izquierda). Modelo 3D (Experimento 6.2.2.II, Derecha)

Es importante mencionar que aunque en el Experimento 6.2.2.I la calidad era el criterio dominante, el Experimento 6.2.2.II tuvo una puntuación de calidad ligeramente más alta. Esto se debe principalmente a que las zonas de interés y la mejora de la calidad generan puntuaciones similares en muchas zonas del mapa (lo que se debe probablemente a la alta estructuración del entorno) de forma que el criterio dominante es la ganancia de información esperada. Esto, como se puede ver en la próxima sección, no es aplicable para entornos menos estructurados. La Figura 6-9 muestra el mapa de navegabilidad y el modelo 3D resultante.

6.2.3. Entorno Desestructurado

El entorno desestructurado puede ser visto en la Figura 6-10. El terreno en este caso tiene pocos pasos seguros y la complejidad de navegación que ofrece es muy alta.



Figura 6-10: Captura de Pantalla del Entorno Desestructurado Simulado.

El entorno está diseñado de tal forma que el robot tiene que atravesar necesariamente sobre y bajo puentes (en gris en la figura), de manera que se puede probar la habilidad del algoritmo para manejar este tipo de elementos, así como desfiladeros y estructuras elevadas. El objetivo de este escenario ha sido probar la capacidad del algoritmo de elegir posiciones que son seguras y de manejar eficientemente entornos complejos.

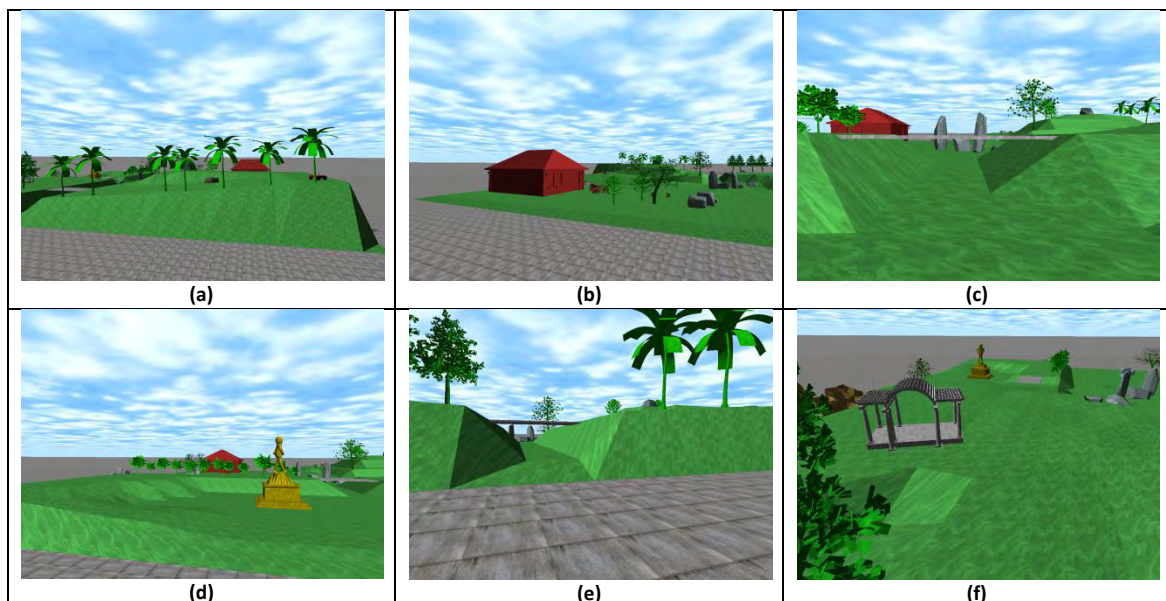


Figura 6-11: Capturas del Entorno Desestructurado

En este entorno se presentan dos experimentos. El primero cuenta con un coste de navegación elevado y una configuración equilibrada para los parámetros de exploración. El segundo

experimento tiene un costo de navegación más bajo y se ha establecido la ganancia de información como criterio de exploración dominante.

Tabla 6-6: Parámetros Usados en los Experimentos

	W_A	W_Q	W_O	W_C
Experimento 6.2.3.I	0,4	0,35	0,25	0,125
Experimento 6.2.3.II	0,65	0,3	0,05	0,08

Los parámetros elegidos se pueden ver en la Tabla 6-6. La posición inicial del robot en ambos casos se puede ver en la esquina superior derecha (ver Figura 6-10), desde donde el robot debe encontrar una trayectoria segura que le permita cubrir todo el entorno. La Figura 6-12 muestra las trayectorias obtenidas en ambos casos.

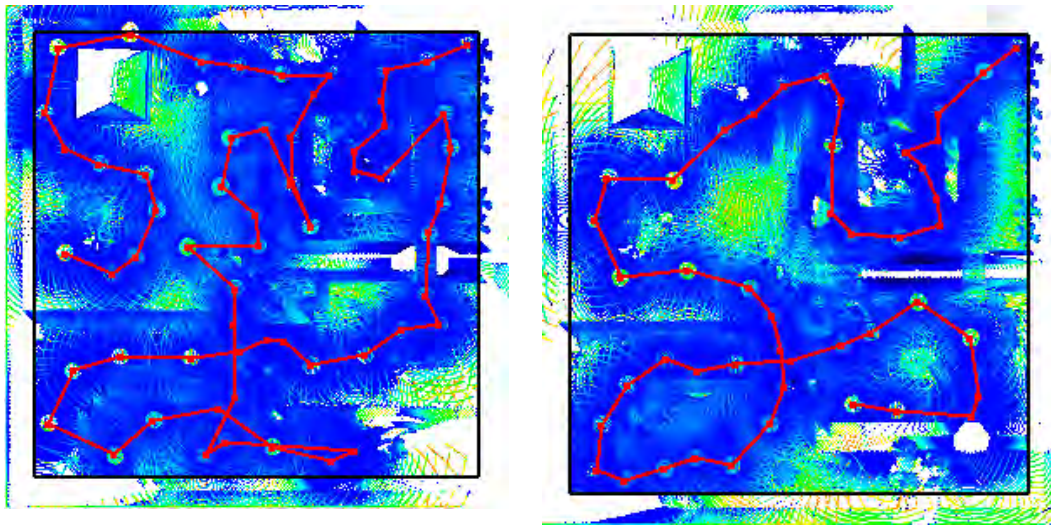


Figura 6-12: Trayectorias Obtenidas en el Experimento 6.2.3.I (izquierda) y Experimento 6.2.3.II (derecha). Los colores más fríos representan mejores calidades por celda.

La Tabla 6-7 muestra los resultados obtenidos para los experimentos y la Figura 6-13 muestra los mapas de navegabilidad.

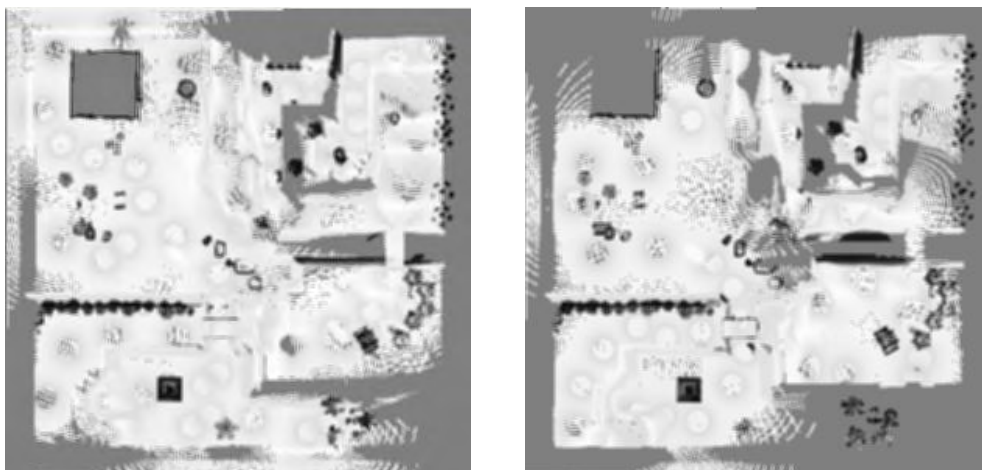


Figura 6-13: Mapas de Navegabilidad; Experimento 6.2.3.I (Izquierda); Experimento 6.2.3.II (Derecha).

En este escenario, los resultados obtenidos en ambos experimentos son notablemente diferentes. Mientras que el Experimento 6.2.3.I realizó una trayectoria mucho más larga y necesitó el 76% más de escaneos, el Experimento 6.2.3.II utilizó mucha menos distancia y

escaneos para cubrir un porcentaje muy similar de terreno. Por otro lado, la puntuación de calidad para el Experimento 6.2.3.I es notablemente más alta.

Tabla 6-7: Resultados Obtenidos en los Experimentos

	Distancia Recorrida	Puntuación de Calidad	Cobertura	Numero de Escaneos
Experimento 6.2.3.I	424 m	0,7833	88,35 %	60
Experimento 6.2.3.II	290 m	0,6013	80,71 %	46

Estas diferencias son explicadas por dos factores: por una parte, el coste de navegación hace que el algoritmo en el primer experimento sea más conservador al escoger vistas; y por otra, la importancia de la mejora en la calidad hace necesario que se realicen más escaneos para cumplir con este requerimiento.

La diferencia en cobertura se puede explicar por el peso del criterio de cobertura de zonas de interés. En el Experimento 6.2.3.II, cuando la casa existente en el entorno fue escaneada, el robot fue apartado de esta por la necesidad de cubrir nuevas zonas no escaneadas. Por el contrario en el Experimento 6.2.3.I, al escanear la casa se presentaron muchas nuevas zonas de interés y su importancia obligó al robot a seguir una trayectoria que le permitiera cubrir toda la casa. El modelo reconstruido puede verse en la Figura 6-14.

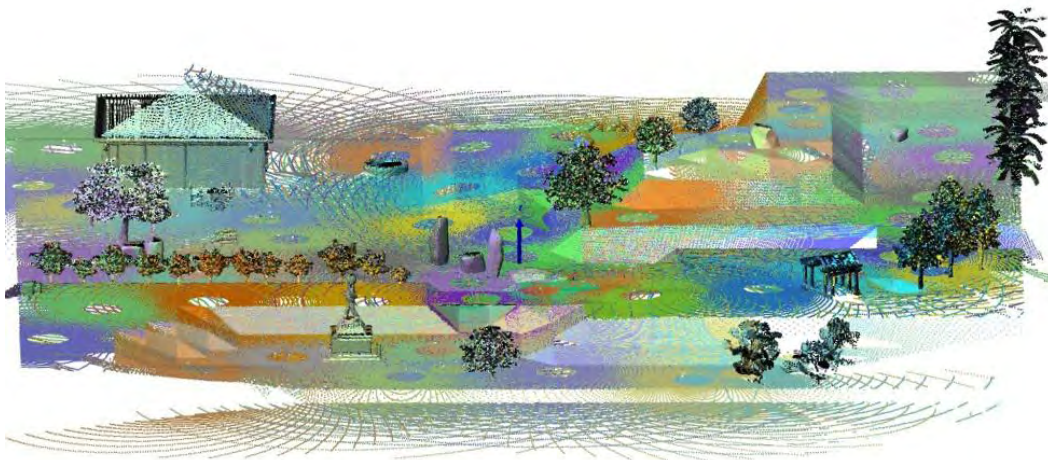


Figura 6-14: Modelo Reconstruido del Entorno Desestructurado

6.3. Entornos Reales

A continuación se presentan resultados de experimentos realizados en dos entornos reales. Tal como se ha indicado, los resultados han sido obtenidos usando el robot y el sistema de escaneado desarrollados en el presente trabajo de tesis, que fueron descritos en el capítulo 5.

El primero de los experimentos se realizó alrededor del edificio de la Fundación CARTIF en Boecillo. Los motivos para seleccionar este lugar fueron varios y de diferente índole. Por un lado, ofrecía excelentes condiciones de accesibilidad para la realización de las pruebas. Por otra parte, se trata de un complejo de edificios geoméricamente bien definidos con una mezcla de caminos de acceso y hierba alrededor de los mismos y con algunos obstáculos y cambios de nivel útiles para probar el algoritmo de extracción de suelos. Finalmente, aunque su forma es geoméricamente simple, también cuenta con diferentes planos de ocultación que son interesantes para verificar el comportamiento del algoritmo.

El segundo entorno es un edificio abandonado llamado “El Pinarón” en Viana de Cega, Valladolid. Este lugar fue seleccionado por ser un lugar ideal para la realización de un simulacro de un escenario de rescate pues cuenta con una gran estructura rodeada de árboles y escombros, con diferentes alturas de terreno y muchos planos de ocultación.

6.3.1. Edificio de la Fundación CARTIF



Figura 6-15: Fundación CARTIF

Se presentan dos experimentos realizados en este entorno. Los valores de los parámetros elegidos en este caso fueron elegidos para alcanzar dos objetivos principales. En el primer experimento, los parámetros se seleccionaron con el objetivo de maximizar la cantidad de información a capturar en cada escaneo y obtener así un modelo rápido del entorno. En el

segundo experimento, la calidad del modelo es el criterio dominante, aunque también se ha aumentado el peso del costo de la navegación para garantizar que los recorridos entre puntos de escaneo sean más cortos, de forma que se pueda obtener un modelo de buena calidad del entorno.

Tabla 6-8: Parámetros Usados en los Experimentos

	w_A	w_Q	w_O	w_C
Experimento 6.3.1.I	0,84	0,10	0,05	0,05
Experimento 6.3.1.II	0,35	0,40	0,15	0,40

Resultados de la extracción de Zonas de Navegación Segura y Obstáculos

Los resultados obtenidos para este proceso se pueden ver en la Figura 6-16 y muestran que el método propuesto en esta investigación es útil para detectar las zonas de navegación seguras y los obstáculos en terrenos reales con diferentes alturas e inclinaciones.

A partir de estos datos se puede ver que el mapa de navegabilidad es útil para crear un plano de suelos del entorno. Si se compara esta representación con los mapas de ocupación usados habitualmente en entornos planos, se puede comprobar, igualmente, que el concepto de navegabilidad es aplicable en estos casos.

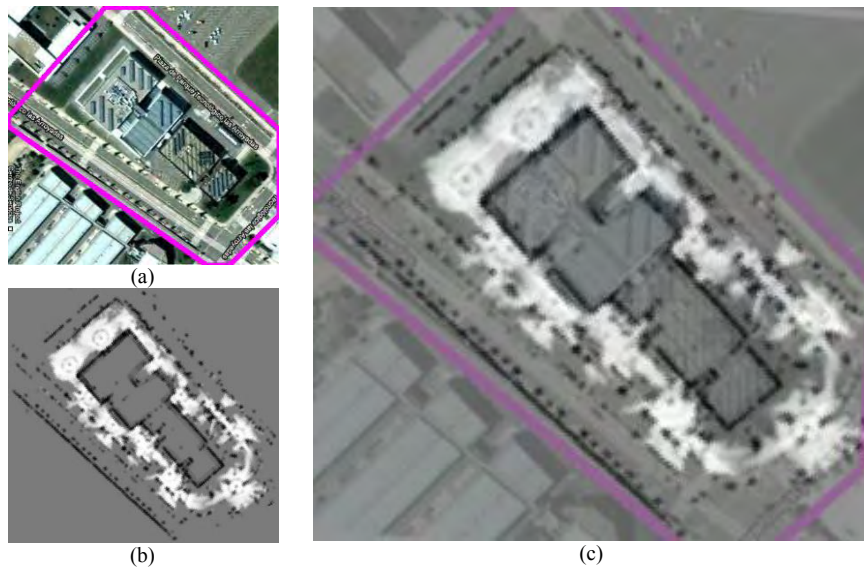


Figura 6-16: (a) Vista aérea de la Fundación CARTIF. (b) Mapa de Navegabilidad Resultante. (c) Imagen Comparativa entre el mapa obtenido y la imagen aérea.

Resultados del Proceso de Exploración

La Figura 6-17 muestra la trayectoria que realizó el robot y el modelo final reconstruido para el Experimento 6.3.1.I. En la Figura 6-17-a, los colores más fríos representan puntuaciones de calidad más alta y en color magenta se muestra la trayectoria del robot. La Figura 6-17-b

muestra el modelo final reconstruido en donde cada escaneo se representa en un color diferente.

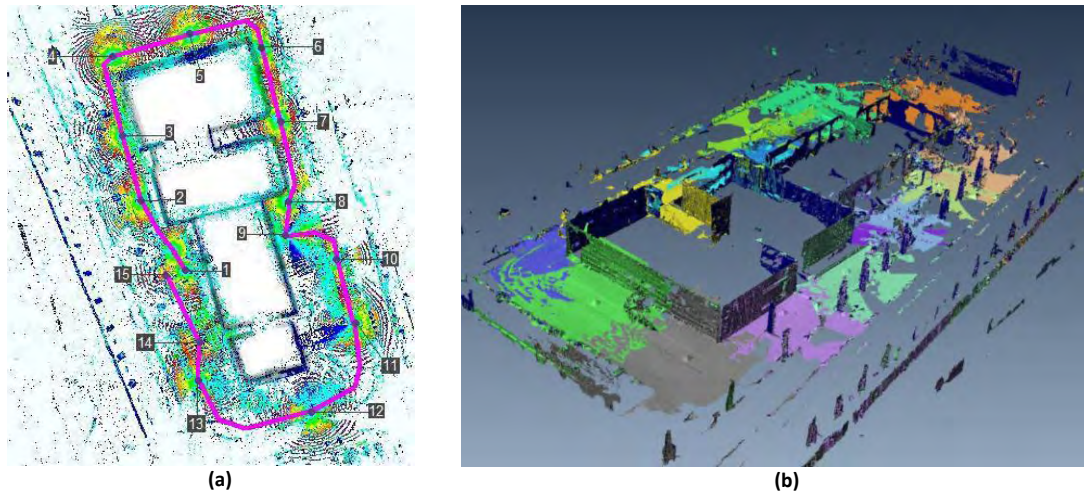


Figura 6-17: Trayectoria realizada por el robot (en magenta) en el Experimento 6.3.1.I, donde la ganancia esperada de la información era el criterio dominante (Izquierda). Los puntos de escaneo están marcados en con numeración en color gris. Modelo resultante para el experimento (Derecha).

La Figura 6-18-a muestra la trayectoria seguida por el robot en el Experimento 6.3.1.II en la imagen superior. En la Figura 6-18-b se puede apreciar una captura de pantalla del modelo reconstruido en este experimento.

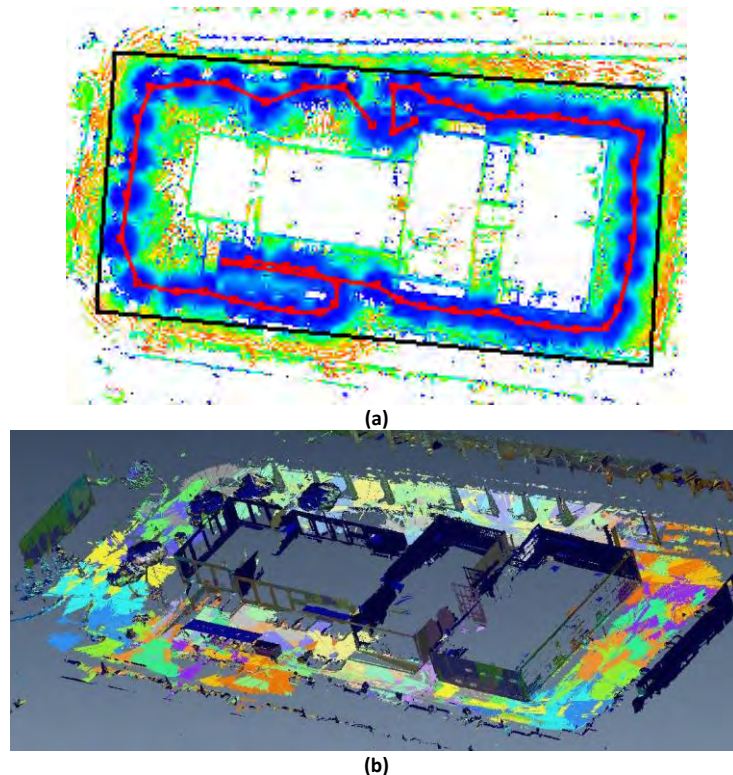


Figura 6-18: Izquierda. Puntuación de calidad del modelo resultante, los colores más fríos representan más calidad. Derecha. Trayectoria realizada por el Robot en magenta.

En este caso, ambos experimentos dieron resultados muy distintos. El primero y más importante de los factores que motivaron esta situación lo constituye el hecho de que en el Experimento 6.3.1.II la calidad era el factor dominante. Esto, sumado al alto costo de

navegación, originó que las posiciones ganadoras estuviesen siempre cerca de la posición del robot.

En el Experimento 6.3.1.I, la calidad no tenía mayor relevancia en la selección de objetivos y el coste de navegación era notablemente más bajo, de forma que las posiciones candidatas seleccionadas fueron aquellas desde las que se podía captar mayor cantidad de información nueva, las cuales estaban en las celdas navegables más cercanas a la frontera entre el espacio conocido y desconocido. Este comportamiento se asemeja mucho a un método de exploración basado en fronteras.

En el Experimento 6.3.1.I, debido a la ventaja que tenían las posiciones de frontera, se perdieron muchas posiciones intermedias que habrían sido claves para obtener un modelo de buena calidad. Sin embargo, el modelo obtenido podría ser útil en algunas aplicaciones y fue capturado con gran rapidez. El Experimento 6.3.1.II por el contrario generó un modelo de una calidad elevada, aunque tardó un tiempo mucho mayor en capturar toda la información 3D del edificio.

Estos resultados demuestran que la configuración de los parámetros del algoritmo permite adaptar su comportamiento a las necesidades de la aplicación específica del modelo.

6.3.2. El "Pinarón"

Este escenario (ver Figura 6-19) es un entorno mucho más complejo que el anterior y por ende, más interesante. Tiene una estructura muy grande rodeada de múltiples objetos como árboles o escombros que dificultan considerablemente tanto la navegación sobre el terreno como la captura de datos tridimensionales. Además su estado de abandono, con accesos limitados y muy deteriorados, lo hace un escenario ideal para comprobar la capacidad del robot de generar modelos 3D útiles, de forma rápida, en escenarios complejos.



Figura 6-19: "El Pinarón"

El terreno es a grandes rasgos plano, pero presenta ciertas elevaciones y las superficies navegables son de tierra y piedras o hierba alta. Esto representa un reto tanto para el sistema de tracción del robot como para la extracción del mapa de navegabilidad.

Con el objetivo de comparar este método contra un sistema estacionario típico, se ha diseñado un simulacro de situación de emergencia en la que es necesario obtener un modelo tridimensional de un edificio que posiblemente esté colapsando, para evaluar los posibles riesgos de una intervención en el mismo. Para esta prueba se han utilizado tanto el robot desarrollado en esta investigación como un sistema estacionario controlado por un experto, con el propósito de hacer una comparación realista entre ambas soluciones.

Primero se analizará el comportamiento del método de exploración autónoma en el proceso de extracción de zonas de navegación seguras y obstáculos, luego se mostrarán los resultados del proceso de exploración; y posteriormente se analizarán los resultados obtenidos con el sistema estacionario. Finalmente se comparan los dos métodos y sus resultados.

Para la exploración robótica, se han configurados los parámetros de forma balanceada, con el propósito de obtener un modelo completo de buena calidad utilizando la menor energía posible. La Tabla 6-9 muestra los valores utilizados.

Tabla 6-9: Parámetros Usados en el Simulacro

	W_A	W_Q	W_O	W_C
Experimento 6.3.2.1	0,55	0,30	0,15	0,16

Extracción de Zonas de Navegación Segura y Obstáculos

Los resultados obtenidos en este proceso pueden ser vistos en la Figura 6-20. En ella se puede apreciar que el método propuesto tiene la capacidad de diferenciar obstáculos y zonas de navegación seguras.

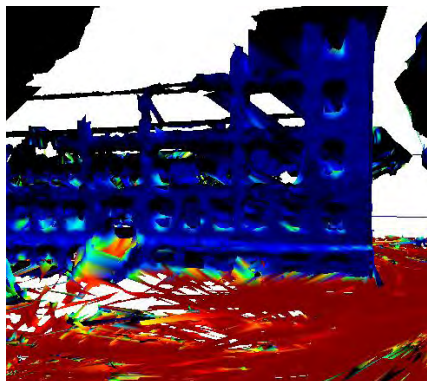


Figura 6-20: Resultado de la detección de zonas seguras de navegación segura en un entorno real. Los colores más cálidos representan las superficies detectadas.

El mapa de navegabilidad obtenido puede ser visto en la Figura 6-21. El resultado obtenido es un mapa de Buena calidad, útil para la planificación de trayectorias y que a la vez es un plano claro del entorno.

Al igual que en el caso del edificio de CARTIF, al comparar este mapa con los mapas de ocupación habituales en los entornos planos, es posible observar las múltiples aplicaciones del concepto de navegabilidad.



Figura 6-21: (a) Vista aérea del sitio de trabajo. (b) Mapa de navegabilidad resultante. (c) Imagen Comparativa entre el mapa de navegabilidad y la vista aérea del entorno.

Resultados del Proceso de Exploración

La Figura 6-22 muestra la trayectoria del robot para la reconstrucción del entorno sobre el mapa de calidad del mismo.

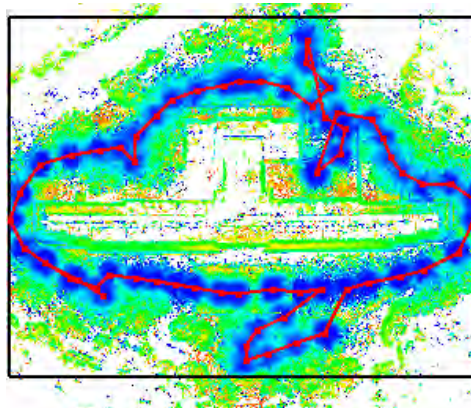


Figura 6-22: Calificación de calidad para el modelo resultante. Los colores más fríos representan mejor calidad y en magenta se encuentra la trayectoria realizada por el Robot.

El levantamiento del modelo con el robot requirió 63 escaneos para la reconstrucción del entorno, los cuales fueron realizados en 1 hora 58 minutos. El recorrido realizado por el robot fue de 690 metros. El modelo resultante tiene una calidad satisfactoria (ver Figura 6-23) que permite la evaluación de vías de acceso. El mapa de navegabilidad obtenido muestra claramente la diferencia entre las zonas más accesibles para el robot y aquellas con elementos que dificultan la navegabilidad (ver Figura 6-21b).

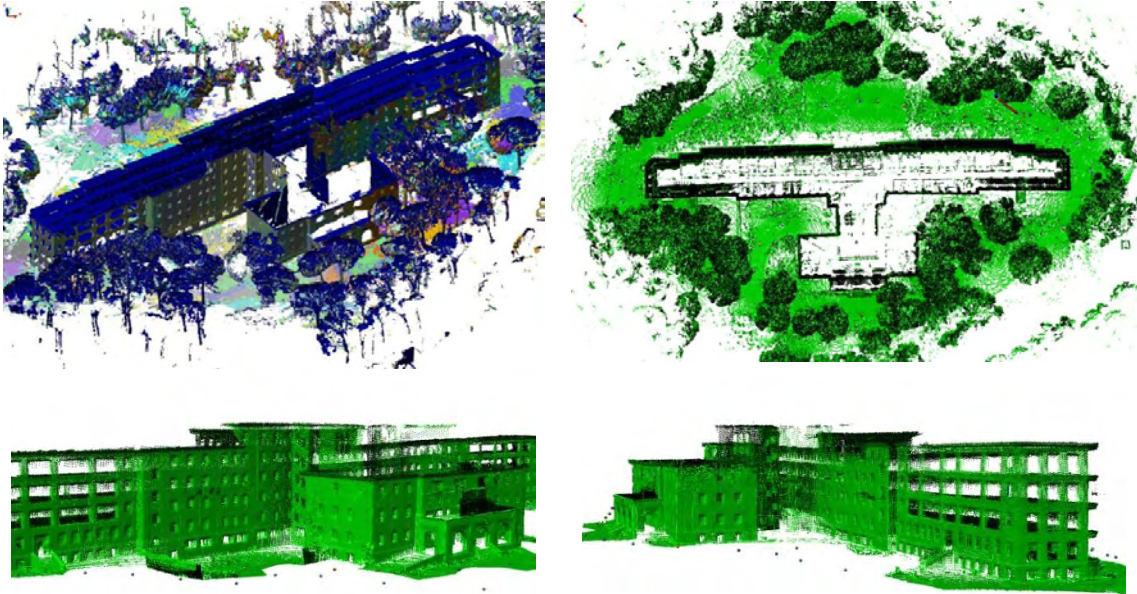


Figura 6-23: Modelo 3D reconstruido por el sistema Robótico

La trayectoria resultante demuestra que este método permite calcular trayectorias seguras en entornos complejos con terreno poco estructurado.

La cobertura del entorno fue del 62%, debido a dos motivos principales. En primer lugar, los criterios de mejora de la calidad del modelo y cobertura de zonas de oclusión hacen que el robot siga una trayectoria alrededor del edificio, de forma que las esquinas del área delimitada fueron omitidas en el primer pase. Una vez culminada esta trayectoria, la necesidad de capturar información nueva, se ve contrarrestada por la dificultad de alcanzar posiciones nuevas. En segundo lugar, algunas zonas fueron detectadas como obstáculos debido a la altura de la hierba y la densidad de los árboles y eso ocasionó que el robot no se adentrara en esos puntos.

Comparación con un Sistema Estacionario

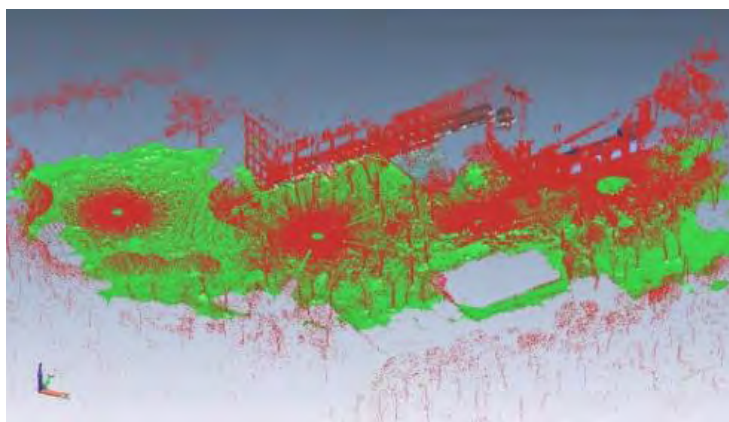


Figura 6-24: Modelo capturado por el sistema estacionario con las tomas alineadas

Para esta comparación un operador experto hizo una reconstrucción parcial del entorno usando un sistema LEICA HDS-3000. Los resultados pueden verse en la Figura 6-24 y la Figura 6-25.

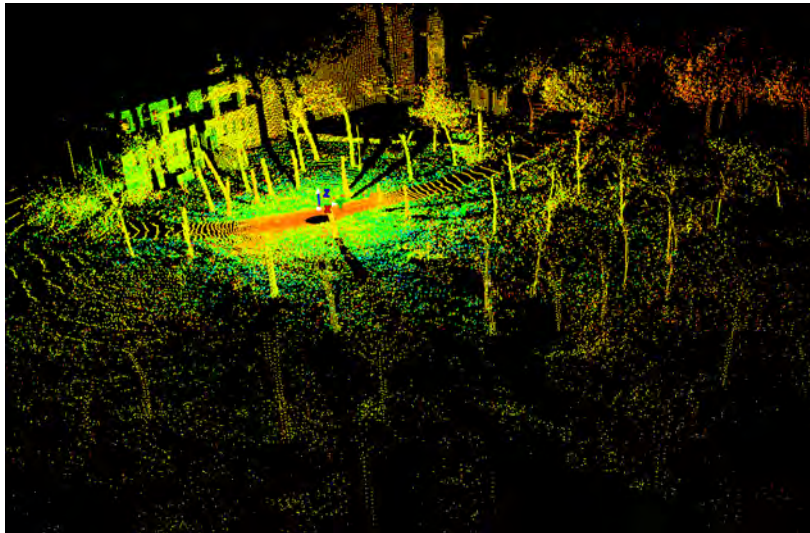


Figura 6-25: Nube de puntos del sistema estacionario

El sistema estacionario tiene un alcance de 100 metros y puede hacer tomas con resoluciones de hasta 1 mm^2 desde esa distancia, de manera que el modelo capturado tiene una calidad que nuestro sistema no puede alcanzar. Por otro lado, el sistema estacionario necesita 5 minutos de calentamiento y preparación, y en este caso cada escaneo requirió 15 minutos para estar listo. Una vez terminado el escaneo, es necesario apagar el sistema por seguridad y transportarlo manualmente hasta la nueva posición. El peso del sistema de escaneo (13 kg) y sus baterías (8 kg), además de la necesidad de transportar también el trípode y el ordenador portátil, hacen de esta una labor ardua para una sola persona.

La reconstrucción realizada en este caso fue solo parcial por cuestiones de tiempo, pero el operador destacó que debido al gran número de oclusiones presentes en el entorno, probablemente serían necesarios hasta 20 escaneos, lo cual se traduce en unas 8 horas de trabajo solo en la captura de datos. Además sería necesaria una fase de alineamiento posterior que podría llegar a requerir un tiempo casi igual al de la toma de datos.

En este sentido, el sistema aquí propuesto resulta mucho más rápido no solo en la fase de captura de datos, sino también en el proceso de alineamiento (el cual se hace de forma simultánea). En concreto, la reconstrucción del mismo entorno se realizó en menos de dos horas (aunque fueran necesarios más escaneos). Obviamente esta reducción de tiempo que se consigue con el sistema propuesto puede ser vital en situaciones de emergencia.

Merece la pena destacar que en algunas situaciones de emergencia donde puedan encontrarse agentes tóxicos o radioactivos, el riesgo para un operador humano puede ser muy elevado. Sin embargo, el sistema robótico puede ser supervisado a una distancia considerable.

En términos de calidad los modelos no son directamente comparables, ya que el sistema estacionario es un equipo de altas prestaciones que puede alcanzar resoluciones muy altas.

Además, la alineación de los escaneos se realiza de forma manual y su precisión depende mucho de la pericia del operador.

6.3.3. Análisis de los Tiempos de Ejecución

Para analizar los tiempos de ejecución del algoritmo, este tiene que ser dividido en diferentes etapas. Primero se presentan los tiempos obtenidos para los procesos previos (triangulación del modelo y SLAM 6D). Luego se presentan los tiempos para el análisis 3D y la evaluación de posiciones candidatas.

Los tiempos de ejecución del algoritmo varían enormemente dependiendo de la cantidad de datos tridimensionales que tiene que ser procesada. Sin embargo, para estos experimentos, la Tabla 6-10 presenta los tiempos máximos obtenidos para los experimentos expuestos en este capítulo usando un ordenador con un procesador Intel Core2 Duo de 3GHz. En la tercera columna de la tabla se puede apreciar la situación específica en la que se produjeron estos tiempos. El tiempo total máximo es la suma del máximo de cada uno de los procesos de forma que representa el peor de los casos.

Tabla 6-10: Tiempos Máximos por Proceso

Proceso	Tiempo Máximo Procesamiento	Observaciones
Triangulación del Modelo	0,12 s	Triangulación de una única nube de puntos con 175.270 puntos.
SLAM 6D	8,5 s	Alineamiento de dos nubes de puntos consecutivas y detección de lazo cerrado con corrección de alineamiento para 15 nubes de puntos.
Análisis Tridimensional	0,45 s	Para una nube de puntos de 147.285 puntos.
Evaluación de Candidatos	15,33 s	Evaluación de 113 posiciones candidatas tras 40 escaneos anteriores.
Tiempo Total Máximo	24,4 s	Este tiempo es la suma de los tiempos anteriores y no el tiempo obtenido en ningún caso.

Estos tiempos de procesamiento demuestran que el algoritmo puede ser ejecutado durante la operación del sistema sin retrasar mucho el proceso de escaneo del entorno.

6.4. Conclusiones del Capítulo

Los resultados obtenidos demuestran que el sistema es capaz de generar modelos útiles de distintos entornos exteriores, de forma rápida y con un esfuerzo considerablemente menor que el requerido por sistemas estacionarios. Demuestran igualmente que esta metodología es capaz de planificar distintos puntos de captura de datos tridimensionales basándose en los criterios seleccionados y en los pesos asignados a estos criterios en cada caso, garantizando un modelo con una calidad mínima seleccionada.

La metodología aquí presentada ha demostrado tener en cuenta las trayectorias hacia los puntos de escaneo evaluados seleccionando trayectorias seguras y manteniendo un balance entre la utilidad de un punto de escaneo y el costo de alcanzarlo.

Los resultados también demuestran que los procedimientos presentados en los capítulos 3 y 4 son válidos y permiten la generación del mapa de información 2D, el cual permite agilizar el cálculo de los criterios utilizados para evaluar las posiciones candidatas a ser el siguiente punto de escaneo. Vale la pena destacar que como se ha podido ver en los resultados, el mapa de navegabilidad, se corresponde con un mapa de celdas de ocupación 2D, pero con validez para entornos complejos.

En este capítulo también se ha podido observar cómo la metodología presentada se adapta a entornos con diferentes grados de estructuración como entornos de tipo urbano o entornos complejos como “El Pinarón”.

Igualmente, se ha realizado una comparativa entre la generación de un modelo tridimensional de un entorno con el sistema aquí propuesto y como el método tradicional de emplazamiento manual de un escáner estacionarios en una serie de posiciones seleccionadas por un experto humano, demostrando que el sistema propuesto permite generar modelos de forma más rápida y a un coste menor.

Capítulo 7.

CONCLUSIONES

La investigación plasmada en esta tesis ha tenido como propósito central el desarrollo de una metodología para obtención automática de modelos tridimensionales de entornos exteriores mediante un robot móvil. Para ello se ha desarrollado un procedimiento de exploración del entorno basado en la información tridimensional adquirida por medio del robot. El procedimiento está orientado a optimizar el proceso de exploración y reconstrucción, permitiendo al robot predecir qué movimientos le aportarán una mayor cantidad de información sobre el terreno de cara a reducir el tiempo y el consumo de recursos al tiempo que se mejora la calidad final del modelo.

Tal como se ha analizado a lo largo de este documento, la obtención de mapas es una misión fundamental que la mayoría de los robots deben de ser capaces de realizar. Para que el proceso de mapeado sea totalmente autónomo es imprescindible considerar la selección autónoma de las acciones a realizar por el robot durante esta etapa de exploración. Existen múltiples algoritmos orientados a la exploración entornos interiores para la consecución de un mapa 2D pero sin embargo, no existen muchas metodologías de este tipo para la generación de modelos 3D de un entorno mediante el uso de robots móviles. Entre las metodologías orientadas específicamente a la exploración del entorno cabe citar (Blaer & Allen, 2009) que sin embargo, basa el proceso de generación del modelo 3D en un mapa 2D conocido con anterioridad.

En este sentido la metodología de exploración propuesta en la presente investigación representa una contribución en el campo de la generación automática de modelos 3D. La exploración se basa únicamente en la información tridimensional capturada del terreno, para perfeccionar el proceso de reconstrucción optimizando el número de capturas necesarias. El robot predice qué movimientos le aportarán una mayor cantidad de información y le permitirán mejorar la calidad del modelo final, reduciendo el consumo energético y el tiempo de operación.

Para lograr ese objetivo se ha desarrollado un procedimiento para la extracción y análisis de la información relevante de las nubes de puntos capturadas de un entorno exterior. Este proceso ha demostrado ser capaz de evaluar de forma exitosa la navegabilidad de un entorno y la calidad de la información tridimensional en los puntos medidos. Igualmente, se ha presentado el mapa de información 2D como herramienta útil para la evaluación de los candidatos a ser puntos de escaneo, sin que esto signifique una simplificación de la información. También vale la pena destacar el mapa de navegabilidad, una representación que ha demostrado una gran utilidad de cara a la estimación de trayectorias en entornos complejos y puede resultar útil en otras aplicaciones.

Manteniendo como eje central el desarrollo de la metodología de exploración, la presente investigación incluye igualmente el desarrollo de un robot que cuenta con las herramientas necesarias para la exploración autónoma de un entorno, así como el desarrollo de un simulador tridimensional. Ambos elementos han sido concebidos como herramientas para el desarrollo y validación del algoritmo de exploración propuesto, constituyendo así una parte integral del proceso de investigación. El desarrollo del robot ha permitido la comprensión de las limitaciones a las que está sometido un robot de exteriores y ha posibilitado soslayar la limitada variedad de opciones disponibles en el mercado para una plataforma comercial. Por su parte, la herramienta de simulación ha demostrado ser útil para el desarrollo y validación de distintos algoritmos para robots móviles en entornos exteriores, permitiendo reducir de forma significativa tiempo de desarrollo de los algoritmos garantizando en todo momento la integridad del robot. Su capacidad de incluir elementos con geometría variable y terrenos topológicamente complejos lo hace una herramienta diferente a las opciones existentes, y sumamente interesante para este tipo de aplicaciones.

En cuanto al sistema de reconstrucción tridimensional desarrollado, ha demostrado ser eficiente y robusto, y que requiere menos esfuerzo que otros métodos convencionales de adquisición tridimensional. Hay que destacar la necesidad de sistemas como el propuesto que embarcados en un robot autónomo, permiten intervenir situaciones de emergencia donde puedan encontrarse agentes tóxicos o peligrosos, y donde el riesgo para el operador humano hace inviable su intervención.

Los resultados experimentales demuestran que el proceso de exploración aquí presentado, es capaz de generar trayectorias eficientes para la reconstrucción de un entorno complejo, garantizando el cumplimiento de requerimientos de calidad en el modelo resultante. Cabe destacar el gran potencial que tiene esta metodología para el despliegue de sistemas de reconstrucción tridimensional de entornos y su capacidad de manejar entornos simulados y reales con diferente nivel de estructuración. Igualmente el sistema aquí propuesto puede resultar de gran utilidad en situaciones de emergencia gracias a su rapidez y a su capacidad para ser supervisado a grandes distancias.

La metodología propuesta cuenta con diversos parámetros de cara a flexibilizar la estrategia de exploración, atendiendo a criterios tales como calidad del modelo, el filtrado de obstáculos, la distancia recorrida, etc. Además, se ha visto que la configuración de dichos parámetros es intuitiva e incluso se puede usar un juego estándar de valores para múltiples misiones, como se ha demostrado con la experimentación.

La experimentación realizada en el presente trabajo de investigación, sobre diferentes tipos de entornos, ha permitido demostrar que la metodología de exploración propuesta representa una contribución significativa en el campo de la generación de modelos 3D. Los resultados obtenidos muestran que con esta metodología, es posible generar modelos útiles de entornos exteriores de distinta naturaleza de forma rápida, sin necesidad de contar con información previa de los mismos y con un esfuerzo considerablemente menor que el requerido por los sistemas estacionarios al uso, operados de forma manual.

7.1. Contribuciones de la Tesis

La contribución principal y más destacable de esta investigación es el desarrollo de una metodología para la exploración y reconstrucción tridimensional de entono a partir de un escáner 3D y un robot móvil todoterreno. Pese a que se existen numerosas propuestas de exploración utilizando mapas 2D, apenas hay desarrollos que permitan la reconstrucción autónoma de modelos 3D. En este sentido la metodología para selección de vistas desarrollada es novedosa con respecto a otras propuestas gracias al uso de información tridimensional como única fuente de información para la selección de las vistas.

Para el campo de la robótica de rescate esta metodología es una aportación importante ya que permite la recogida de información tridimensional real de un entorno de forma rápida y con una necesidad de supervisión y control mucho menor que los sistemas existentes. En este sentido la metodología aquí propuesta es un paso importante para la creación de robots que se desplieguen de forma autónoma y recojan información relevante en una situación de emergencia.

Merece la pena destacar la solución propuesta para evaluar la información 3D y calcular los valores posteriores almacenados en un mapa de información 2D. Generalmente, los métodos existentes utilizan la información tridimensional únicamente para la alineación de la información y la generación de modelos consistentes, o bien para la extracción de primitivas que permitan evaluar las condiciones del entorno para la navegación de un sistema. Sin embargo, existen pocos trabajos que evalúen punto a punto la información tridimensional para extraer información relevante de cara a la planificación de movimientos en entornos con obstáculos y a la obtención de la mejor calidad del modelo. La metodología propuesta en este trabajo ha demostrado que este tipo de evaluación es viable y combinada con el mapa de información, permite una comprensión apropiada del entorno.

El mapa de información 2D es otra aportación destacable, ya que aunque en el sentido estricto es un mapa métrico, en este se almacenan valores adicionales que permiten el desarrollo de múltiples representaciones del entorno. La más interesante de ellas es el mapa de navegabilidad, que permite la estimación de trayectorias en un entorno topológicamente complejo. Esta representación tiene un gran potencial ya que permite extender técnicas de planificación de trayectorias desarrolladas para entornos 2D a entornos 3D complejos.

Los resultados demuestran que la combinación del robot todoterreno y el escáner 3D permite la generación de modelos tridimensionales de un entorno complejo en menos tiempo y con un esfuerzo mucho menor del requerido por un sistema estacionario convencional. Esto es una aportación importante para el campo de la digitalización en general y puede tener múltiples aplicaciones que van más allá de la robótica de emergencias como arqueología, obra civil, conservación del patrimonio y en general todas las aplicaciones que requieran modelos tridimensionales en exteriores.

El robot todoterreno desarrollado también es una aportación interesante, ya que es una herramienta versátil que permite su utilización para múltiples aplicaciones y sus características lo hacen diferente de robots todoterreno existentes en el mercado en términos de versatilidad y asequibilidad. Esto mismo aplica para el escáner tridimensional el cual es una alternativa para otras herramientas existentes que ha demostrado ser económico, funcional y confiable.

El simulador tridimensional es una aportación importante, ya que ha permitido el desarrollo y validación de la metodología presentada, gracias a su capacidad de incorporar terrenos y elementos complejos, lo cual lo hace diferente de los demás simuladores. Dichas simulaciones son muy útiles para el desarrollo de algoritmos en entornos desestructurados. Además el simulador desarrollado se encuentra disponible como un programa de código abierto en <https://sourceforge.net/projects/simbot3d>.

Finalmente el análisis comparativo presentado entre esta metodología y los métodos utilizados de adquisición manual demuestra que esta metodología puede sustituir a los métodos de adquisición manual.

7.2. Trabajo Futuro

Como líneas futuras de investigación a partir de esta tesis, se propone el desarrollo de un modelo matemático que permita que los parámetros utilizados para la selección de puntos de vista sean más lineales, de forma que las proporciones para los valores de cada parámetro representen mejor la importancia de cada factor dentro del proceso. Igualmente en este sentido podría ser interesante crear un sistema de aprendizaje que ajuste estos parámetros a partir de las selecciones de un operador experto. También puede ser interesante profundizar en la metodología propuesta para la inclusión de modelos probabilísticos que consideren la incertidumbre del modelo y de los sensores.

Otra línea de investigación interesante es el etiquetado semántico de objetos relevantes a partir de la información 3D y la inclusión de la relevancia de estos objetos dentro de la evaluación de las vistas de exploración.

Desde el punto de vista de la robótica de rescate sería interesante el desarrollo de una metodología para la inclusión de otro tipo de sensores en el proceso de selección de objetivos

(como sensores para la medición de variables medioambientales como temperatura, humedad, detección y medición de la concentración de gases o radioactividad).

Otro desarrollo interesante sería la integración de este algoritmo dentro de un proceso de recogida de datos continuo que no requiera que el robot se detenga a realizar capturas tridimensionales de un punto, esto es realmente interesante ya que aunque multiplicaría enormemente la complejidad del algoritmo también reduciría notablemente el tiempo requerido para la exploración de un entorno.

Finalmente, la metodología presentada sería fácilmente ampliable a sistemas multirobot, si varios robots terrestres comparten el mismo mapa de información 2D y se incluye una función de coste beneficio que considere la situación de otros robots. En este mismo sentido sería posible utilizar información aérea de un entorno que bien puede provenir de un robot aéreo o de cualquier otra fuente, para incluir una estimación de navegabilidad inicial en el mapa de información 2D o marcar obstáculos y zonas de interés.

PUBLICACIONES

1. **“Plataforma Robótica Para Tareas de Reconstrucción Tridimensional de Entornos Exteriores”**, Jaime Pulido Fentanes, Eduardo Zalama, Jaime Gómez García-Bermejo, Revista Iberoamericana de Automática e Informática industrial, Volumen 09, Número 01, pags 81-92, Enero – Marzo, 2012.
2. **“A New Method for Efficient Three-Dimensional Reconstruction of Outdoor Environments Using Mobile Robots”**, Jaime Pulido Fentanes, Raul Feliz Alonso, Eduardo Zalama, Jaime Gómez García-Bermejo, Journal of Field Robotics, Special Issue: Safety, Security, and Rescue Robotics, Volume 28, Issue 6, paginas 832–853, November/December 2011.
3. **“Algorithm for Efficient 3D Reconstruction of Outdoor Environments Using Mobile Robots”**, Jaime Pulido Fentanes, Eduardo Zalama, Jaime Gómez García-Bermejo, ICRA 2011 (International Conference on Robotics and Automation) proceedings of, paginas 3275 – 3280, Shanghai, Mayo, 2011.
4. **“Reconstrucción Tridimensional de Entornos Exteriores mediante Robots Móviles”**, Jaime Pulido Fentanes, Eduardo Zalama, Jaime Gómez García-Bermejo, actas de III Workshop de Robótica “ROBOT 2011”, paginas 323-330, Noviembre, 2011.
5. **“Diseño De Una Plataforma Robotizada Para El Desarrollo De Tareas En Entornos Exteriores”**, Jaime Pulido Fentanes, Salvador Domínguez Quijada, Eduardo Zalama, Jaime Gómez García-Bermejo, II Workshop de Robótica “ROBOT 2009”, paginas 93-92, Barcelona, Marzo 2009.
6. **“Diseño de un robot de exteriores para labores de vigilancia y reconstrucción de modelos 3D”**, Jaime Pulido Fentanes, Salvador Domínguez Quijada, Eduardo Zalama, Jaime Gómez García-Bermejo, Actas de las XXX Jornadas de Automática, Valladolid 09, paginas 1-8, Valladolid, 2009.
7. **“A Mobile Robot For On-Line Registration Of Indoor 3D Models”**, Jaime Pulido Fentanes, Samuel Marcos Pablos, Salvador Domínguez Quijada, Eduardo Zalama, Jaime Gómez García-Bermejo, Jose R. Perán Gonzalez, ISR 09 (40th INTERNATIONAL SYMPOSIUM ON ROBOTICS), paginas 93-92, Barcelona, Marzo, 2009.

8. **“Construcción de modelos digitales de terreno incrementales”**, Samuel Marcos Pablos, Jaime Pulido Fentanes, Eduardo Zalama Casanova, Jaime Gómez García-Bermejo, Actas de XXVIII Jornadas de Automática. paginas 1-11 Huelva, 2007.
9. **“Robot Learning in a Social Robot”**, Salvador Domínguez, Eduardo Zalama, Jaime Gómez García-Bermejo, Jaime Pulido, Lecture notes in Artificial Intelligence. Vol 4095. pp:691-702. Springer Ed, 2006.

REFERENCIAS

- Adept Technology Inc. (2012, 03 26). *Adept MobileRobots*. Retrieved 03 26, 2012, from Sitio Web de Adept MobileRobots: http://www.mobilerobots.com/Mobile_Robots.aspx
- Albers, S., & Henzinger, M. R. (1997). Exploring unknown environments. *STOC '97 Proceedings of the twenty-ninth annual ACM symposium on Theory of computing* (pp. 416 - 425). El Paso, Texas, United States: ACM.
- Amano, H. (2002). Present status and problems of fire fighting robots. *Proceedings of the 41st Annual Conference of the Society-of-Instrument-and-Control-Engineers (SICE 2002)* (pp. 880-885). Osaka, Japan: Soc Instrument & Control Engineers.
- Basilico, N., & Amigoni, F. (2008). On Evaluating Performance of Exploration Strategies for an Autonomous Mobile Robot. *IROS2008 (IEEE/RSJ International Conference on Intelligent Robots and Systems) Workshop on Performance Evaluation and Benchmarking for Intelligent Robots and Systems*. Nice, France: IEEE/RSJ.
- Basilico, N., & Amigoni, F. (2009). Exploration strategies based on multicriteria decision making for an autonomous mobile robot. *Proceedings of ECMR, 2009*, (pp. 259 - 264).
- Besl, P., & McKay, N. (1992). A method for Registration of 3-D Shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 239-256.
- Birk, A., & Kenn, H. (2002). RoboGuard, a teleoperated mobile security robot. *Control Engineering Practice*, 11(10), 1259 - 1264.
- Blaer, P. S., & Allen, P. K. (2007). Data Acquisition and View Planning for 3-D Modeling Tasks. *IROS*. San Diego, CA: IEEE/RSJ.
- Blaer, P. S., & Allen, P. K. (2009). View Planning and Automated Data Acquisition for Three-Dimensional Modeling of Complex Sites. *Journal of Field Robotics*, 26(11-12), 865-891.
- Blanco, J., Fernández-Madrigal, J., & Gonzalez, J. (2008, January). A Novel Measure of Uncertainty for Mobile Robot SLAM with Rao-Blackwellized Particle Filters. *The International Journal of Robotics Research*, 27(1), 73-89.
- Borenstein, J., Everet, H. R., & Liqiang, F. (1996). *Where am I? sensors and methods for autonomous mobile robot positioning*. Michigan: University of Michigan.
- Bostrom, G., Fiocco, M., Puig, D., Rossini, A., Goncalves, J. G., & Sequeira, V. (2004). Acquisition, Modelling and Rendering of Very Large Urban Environments. *Proceedings of the 3D Data Processing, Visualization, and Transmission, 2nd International Symposium 3DPVT '04* (pp. 191-198). Thessaloniki, Greece: IEEE.
- Bourgault, F., Makarenko, A. A., Williams, S. B., Grocholsky, B., & Durrant-Whyte, H. F. (2002). Information Based Adaptive Robotic Exploration. in *Proceedings IEEE/RSJ International Conference on Intelligent Robots and Systems IROS 2002* (pp. 540--545). Lausanne, Switzerland: IEEE/RSJ.
- Bruemmer, D., Dudenhofer, D., & Marble, J. (2002). Dynamic Autonomy for Urban Search and Rescue. *Proceedings of the 2002 {AAAI} Mobile Robot Workshop* (pág. 3337). AAAI Press.
- Calisi, D., Farinelli, A., Iocchi, L., & Nardi, D. (2007). Multi-objective exploration and search for autonomous rescue robots: Research Articles. *Journal of Field Robotics*, 763--777.
- Castellanos, J. A., Neira, J., & Tardós, J. D. (2001, Diciembre). Multisensor fusion for simultaneous localization and map building. *Robotics and Automation, IEEE Transactions on*, 17(6), 908 - 914.

- Chatila, R., & Laumond, J.-P. (1985). Position referencing and consistent world modeling for mobile robots. *Proc International Conference on Robotics and Automation*. 2, pp. 138-145. St. Louis, MO: IEEE.
- Chen, Y.-L., Sundareswaran, V., Anderson, C., Broggi, A., Grisleri, P., Porta, P. P., et al. (2008, October). TerraMax™: Team Oshkosh urban robot. *Journal of Field Robotics - Special Issue on the 2007 DARPA Urban Challenge, Part III*, 25(10), 841--860.
- Chernyaev, E. V. (1995). *Marching Cubes 33: Construction of Topologically Correct Isosurfaces*. Moscu.
- Choset, H. (2001). Coverage for robotics – A survey of recent results. *Annals of Mathematics and Artificial Intelligence*, 113-126.
- Choset, H., & Nagatani, K. (2001, April). Topological simultaneous localization and mapping (SLAM) : toward exact localization without explicit localization. *IEEE Transactions on Robotics and Automation*, 17, 125-137.
- Danner, T., & Kavraki, L. E. (2000). Randomized Planning for Short Inspection Paths. *Proceedings of IEEE International Conference on Robotics and Automation ICRA*.
- Debenest, P., Fukushima, E. F., Tojo, Y., & Hirose, S. (2005). A New Approach to Humanitarian Demining. *Autonomous robots*, 18(3), 323-336.
- Dellaert, F. (2005). *4D cities spatio-temporal reconstruction from images*. Retrieved 11 10, 2010, from <http://www.cc.gatech.edu/4d-cities/>
- Doucet, A., de Freitas, N., Murphy, K., & Russell, S. (2000). Rao-Blackwellised Particle Filtering for Dynamic Bayesian Networks. *Proceedings of the Sixteenth Conference Annual Conference on Uncertainty in Artificial Intelligence (UAI-00)* (pp. 176-183). San Francisco, CA: Morgan Kaufmann.
- Durrant-Whyte, H., Majumder, S., Thrun, S., de Battista, M., & Scheduling, S. (2003). A Bayesian Algorithm for Simultaneous Localisation and Map Building. In R. Jarvis, & A. Zelinsky, *Springer Tracts in Advanced Robotics: Robotics Research* (Vol. 6, pp. 49-60). Berlin / Heidelberg: Springer.
- Edlinger, T., & von Puttkamer, E. (1994). Exploration of an Indoor-Environment. *IROS '94. Proceedings of the IEEE/RSJ/GI International Conference on Intelligent Robots and Systems '94* (pp. 1278 - 1284 vol.2). Munich , Germany : IEEE/RSJ/GI.
- Freese, M., Matsuzawa, T., Oishi, Y., Debenest, P., Takita, K., Fukushima, E. F., et al. (2007). Robotics-assisted demining with Gryphon. *Advanced Robotics*, 21(15), 1763-1786.
- Früh, C., & Zakhor, A. (2003). Constructing 3D City Models by Merging Ground-Based and Airborne Views. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '03)*, (pág. 562).
- Geeknet, Inc. (2012, 01 19). *UsarSim Project*. Retrieved 03 26, 2012, from Pagina Wiki del proyecto UsarSim: http://sourceforge.net/apps/mediawiki/usarsim/index.php?title=Main_Page
- Gerkey, B. (2010, 11 25). *The Player Project Free Software tools for robot and sensor applications* . Retrieved 03 26, 2012, from Sitio Web del Proyecto Player/Stage: <http://playerstage.sourceforge.net/index.php?src=index>
- Goodrich, M., Olsen Jr., D., Crandall , J., & Palmer, T. (2001). Experiments in Adjustable Autonomy. *Systems, Man, and Cybernetics, 2001 IEEE International Conference on*. 3, págs. 1624-1629. Tucson, AZ , USA: IEEE.
- Guez, A., & Pineau, J. (2010). Multi-tasking SLAM . *IEEE International Conference on Robotics and Automation (ICRA2010)* (pp. 377-384). Anchorage, AK, USA: IEEE.
- Hähnel, D., Burgard, W., & Thrun, S. (2003). Learning Compact 3D Models of Indoor and Outdoor Environments with a Mobile Robot. *Robotics and Autonomous Systems*, 44(2), 15-27.
- Hertzberg, J., Lingemann, K., Lörken, C., Nüchter, A., & Stiene, S. (2008). Does It Help a Robot Navigate to Call Navigability an Affordance? *Lecture Notes in Computer Science Towards Affordance-Based Robot Control*, 16 - 26.
- Jensen, B., Weingarten, J., Kolski, S., & Siegwart, R. (2005). Laser range imaging using mobile robots: From pose estimation to 3D-models. *1st Range Imaging Research Day*, (pp. 129-144). Zurich, Switzerland.

- John, L., How, J., Teller, S., Berger, M., Campbell, S., Fiore, G., et al. (2008, September). A Perception Driven Autonomous Urban Vehicle. *Journal of Field Robotics*, 10(25).
- Klein, K., & Sequeira, V. (2000). View Planning for the 3D Modelling of Real World Scenes. *2000 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2, pp. 943-948. Takamatsu, Japan: IEEE.
- Ko, A., & Lau, H. (2009, November). Intelligent Robot-assisted Humanitarian Search and Rescue System. *International Journal of Advanced Robotic Systems*, 6(2), 121-128.
- Koenig, N., & Hsu, J. (2012, 01 01). *Gazebo*. Retrieved 03 26, 2012, from Sitio Web del simulador Gazebo: <http://gazebosim.org/>
- Koenig, S., Tovey, C. A., & Halliburton, W. (2001). Greedy Mapping of Terrain. *ICRA. 2001 Proceedings of the 2001 IEEE International Conference on Robotics and Automation* (pp. 3594-3599). Seoul, Korea: IEEE.
- Makarenko, A., Williams, S., Bourgault, F., & Durrant-Whyte, H. (2002). An Experiment in Integrated Exploration. *In Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS 2002)* (pp. 534-539). Lausanne, Switzerland: IEEE/RSJ.
- Martín Nevado, M., Gómez García-Bermejo, J., & Zalama Casanova, E. (2004). Obtaining 3D models of indoor environments with a mobile robot by estimating local surface directions. *Robotics and Autonomous Systems*, 131-143.
- May, S., Dröschel, D., Holz, D., Fuchs, S., & Nüchter, A. (2009). Robust 3D-Mapping with Time-of-Flight Cameras. *IEEE/RSJ International Conference on Intelligent Robots and Systems, 2009. IROS 2009*. (pp. 1673 - 1678). St. Louis: IEEE/RSJ.
- Microsoft. (2006, 12 18). *Microsoft Robotics*. Retrieved 03 02, 2012, from Microsoft Robotics Developer Studio: <http://www.microsoft.com/robotics>
- Miller, I., Campbell, M., Huttenlocher, D., Kline, F.-R., Nathan, A., Lupashin, S., et al. (2008, August). Team Cornell's Skynet: Robust perception and planning in an urban environment. *Journal of Field Robotics - Special Issue on the 2007 DARPA Urban Challenge, Part I*, 25(8), 493--527.
- Moravec, H., & Elfes, A. (1985). High Resolution Maps from Wide Angle Sonar. *Proc. 1985 IEEE Int. Conf. on Robotics and Automation*, (pp. 116-121).
- Mountarlier, P., & Chatila, R. (1989). Stochastic multi-sensory data fusion for mobile robot location and environmental modeling. *In Proceeding of 5th ISRR (International Symposium on Robotics Research)*, (pp. 85-94). Tokyo, Japan.
- Murphy, K. (2000). Bayesian Map Learning in Dynamic Environments. *Advances in Neural Information Processing Systems 12, (NIPS)* (pp. 1015 - 1021). Denver, Colorado, USA: MIT Press.
- Murphy, R. (2011, Mayo). An Overview of Robots from USA at Fukushima and the Tsunami Recovery. *International Conference on Robotics and Automation (ICRA2011)*. Shanghai, China.
- Newman, P. (2000). PhD thesis. *On the Structure and Solution of the Simultaneous Localisation and Map Building Problem*. Sydney, Australia: Australian Centre for Field Robotics, University of Sydney.
- Nüchter, A., & Hertzberg, J. (2008, November). Towards semantic maps for mobile robots. *Robotics and Autonomous Systems*, 56(11), 915--926.
- Nüchter, A., Lingemann, K., & Hertzberg, J. (2006). Extracting Drivable Surfaces In Outdoor 6D SLAM. *37nd International Symposium on Robotics (ISR '06)*.
- Nüchter, A., Lingemann, K., Hertzberg, J., & Surmann, H. (2007). 6D SLAM - 3D mapping outdoor environments. *Journal of Field Robotics*, 24(8-9), 699 - 722.
- Nüchter, A., Lingemann, K., Hertzberg, J., Surmann, H., Pervözl, K., Hennig, M., et al. (2005). Mapping of Rescue Environments with Kurt3D. *Proceedings of the International Workshop on Safty, Security and Rescue Robotics (SSRR '05)*, (pp. 158 - 163). Kobe, Japan.
- Nüchter, A., Surmann, H., & Hertzberg, J. (2003). Planning Robot Motion for 3D Digitalization of Indoor Environments. *In Proc. of the 11th International Conference on Advanced Robotics (ICAR)* (pp. 222--227). Coimbra, Portugal: ISR.

- Nüchter, A., Surmann, H., Lingemann, K., Hertzberg, J., & Thrun, S. (2004). 6D SLAM with an Application in Autonomous Mine Mapping. *In Proceedings of the IEEE International Conference on Robotics and Automation, ICRA2004* (pp. 1998 - 2003). New Orleans, Louisiana, USA: IEEE.
- Ohno, K., Tadokoro, S., Nagatani, K., Koyanagi, E., & Yoshida, T. (2009). 3-D mapping of an underground mall using a tracked vehicle with four sub-tracks. *2009 IEEE International Workshop on Safety, Security & Rescue Robotics (SSRR)*, (pp. 1-6). Denver, CO.
- Pathak, K., Birk, A., Vaskevicius, N., Pfingsthorn, M., Schwertfeger, S., & Poppinga, J. (2010). Online three-dimensional SLAM by registration of large planar surface segments and closed-form pose-graph relaxation. *Journal of Field Robotics*, 27(1), 52--84.
- Patz, B. J., Papelis, Y., Pillat, R., Stein, G., & Harper, D. (2008, August). A practical approach to robotic design for the DARPA Urban Challenge. 25(8), 528-566.
- Paz, L. M., Piniés, P., Tardós, J. D., & Neira, J. (2008, Octubre). Large-Scale 6-DOF SLAM With Stereo-in-Hand. *IEEE Transactions on Robotics*, 24(5), 946 - 957.
- Pfaff, P., Triebel, R., Stachniss, C., Lamon, P., Burgard, W., & Siegwart, R. (2007). Towards Mapping of Cities. *IEEE International Conference on Robotics and Automation*, (pp. 4807-4813). Roma.
- Puig, D., Sequeira, V., Ruiz, E., Gonçalves, J., & Mellado, M. (2005). Automated 3D Reconstruction System for Autonomous Mobile Manipulator and Vehicle-Borne. *16th IFAC World Congress*. Prague, Czech Republic.
- Pulido Fentanes, J., Marcos Pablos, S., Dominguez Quijada, S., Zalama Casanova, E., Gómez García-Bermejo, J., & Perán, J. (2009). A Mobile Robot for On-Line Registration of Indoor 3D Models. *Proceedings of the 40th International Symposium on Robotics (ISR 2009)* (pp. 93-100). Barcelona: AER-ATP.
- Pulido Fentanes, J., Zalama, E., & Gómez-García-Bermejo, J. (2011). Algorithm for Efficient 3D Reconstruction of Outdoor. *Proceedings of International Conference on Robotics and Automation (ICRA11)*. Shanghai.
- Rusinkiewicz, S. (2005). Rigid-Body Alignment. 3D Scan Matching and Registration, Part II. *ICCV 2005 Short Course*. Nueva Jersey (EE.UU): Princeton University.
- Rusinkiewicz, S. (n.d.). *Trimesh2*. Retrieved 11 16, 2010, from <http://www.cs.princeton.edu/gfx/proj/trimesh2/>
- Sequeira, V., Ng, K., Wolfart, E., Gonçalves, J., & Hogg, D. (1999). Automated Reconstruction of 3D Models from Real Environments. *ISPRS Journal of Photogrammetry and remote sensing*(54), 1-22.
- Sim, R., & Dudek, G. (2004). Online control policy optimization for minimizing map uncertainty during exploration. *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA)*,. New Orleans, LA, 2004: IEEE.
- Simmons, R. (2009, 11 9). *Inter Process Communication (IPC)*. Retrieved 11 16, 2010, from <http://www.cs.cmu.edu/~IPC/>
- Smith, R. (2007, May 28). *Open Dynamics Engine (ODE)*. Retrieved 11 16, 2010, from <http://www.ode.org/>
- Stachniss, C. (2009). *Robotic Mapping and Exploration*. Berlin Heidelberg: Springer Publishing Company, Incorporated.
- Stachniss, C., & Burgard, W. (2003). Exploring unknown environments with mobile robots using coverage maps. *Proceedings of the 18th international joint conference on Artificial intelligence* (pp. 1127--1132). Acapulco, Mexico: Morgan Kaufmann Publishers Inc.
- Su, P., & Drysdale, R. L. (1996). A comparison of sequential Delaunay triangulation algorithms. *Proceedings of the eleventh annual symposium on Computational geometry (SCG' 96)* (pp. 61--70). Vancouver, British Columbia, Canada: ACM.
- Teller, S. (1997). Automatic acquisition of hierarchical, textured 3D geometric models of urban environments: Project Plan. *Proceedings of the Image Understanding Workshop*. New Orleans.
- Thrun, S. (2002). Robotic mapping: a survey. In G. a. Lakemeyer, *Exploring artificial intelligence in the new millennium* (pp. 1-35). Morgan Kaufmann Publishers Inc.

- Thrun, S., Gutmann, J.-S., Fox, D., Burgard, W., & Kuipers, B. J. (1998). Integrating topological and metric maps for mobile robot navigation: a statistical approach. *AAAI '98/IAAI '98* (pp. 989-995). Madison, Wisconsin, United States: American Association for Artificial Intelligence.
- Ulrich, I., & Borenstein, J. (1998). VFH+: Reliable Obstacle Avoidance for Fast Mobile Robots. *Proceedings of the 1998 IEEE International Conference on Robotics and Automation* (pp. 1572 - 1577). Leuven, Belgium: IEEE.
- Urmson, C., Anhalt, J., Bae, H., Bagnell, J. A., Baker, C. R., Bittner, R. E., et al. (2008, June). Autonomous driving in urban environments: Boss and the Urban Challenge. *Journal of Field Robotics Special Issue on the 2007 DARPA Urban Challenge, Part I*, 25(8), 425-466.
- Velodyne Lidar. (2010). *About Velodyne*. Retrieved 03 23, 2012, from sitio web de Velodyne Lidar: <http://velodynelidar.com/lidar/lidar.aspx>
- Visser, A., & Slamet, B. A. (2008). Including communication success in the estimation of information gain for multi-robot exploration. *WiOPT 2008. 6th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks and Workshops, 2008*. (pp. 680 - 687). Berlin, Germany: IEEE.
- Williams, M. (2010). *Street View: Explore the world at street level*. Retrieved November 10, 2010, from Google Street View Web page: http://maps.google.es/intl/en_us/help/maps/streetview/
- Willow Garage. (2009, 08 17). *ROS*. Retrieved 03 02, 2012, from Robot Operating System: <http://www.ros.org>
- Wulf, O., Nüchter, A., Hertzberg, J., & Wagner, B. (2008). Benchmarking Urban Six-Degree-of-Freedom Simultaneous Localization and Mapping. *Journal of Field Robotics*, 25(3), 148-163.
- Yamauchi, B. (1997). A Frontier-Based Approach for Autonomous Exploration. *Proceedings of the 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation, CIRA '97* (pp. 146-152). Washington, DC, USA: IEEE Computer Society.