



Universidad
Zaragoza

Trabajo Fin de Máster

Sistema de visión omnidireccional y de profundidad para guiado de discapacitados visuales

Autor

Manuel Guerrero Viu

Director

José Jesús Guerrero Campo

Máster en Ingeniería Biomédica

Escuela de Ingeniería y Arquitectura

Enero 2019

DECLARACIÓN DE
AUTORÍA Y ORIGINALIDAD

(Este documento debe acompañar al Trabajo Fin de Grado (TFG)/Trabajo Fin de Máster (TFM) cuando sea depositado para su evaluación).

D./D^a. MANUEL GUERRERO VIU,

con nº de DNI 73160786-V en aplicación de lo dispuesto en el art.

14 (Derechos de autor) del Acuerdo de 11 de septiembre de 2014, del Consejo

de Gobierno, por el que se aprueba el Reglamento de los TFG y TFM de la

Universidad de Zaragoza,

Declaro que el presente Trabajo de Fin de (Grado/Máster)
MÁSTER EN INGENIERÍA BIOMÉDICA, (Título del Trabajo)

SISTEMA DE VISIÓN OMNIDIRECCIONAL Y DE PROFUNDIDAD PARA GUIADO DE
DISCAPACITADOS VISUALES

es de mi autoría y es original, no habiéndose utilizado fuente sin ser citada
debidamente.

Zaragoza, 25 de Enero de 2019

Fdo: Manuel Guerrero Viu

AGRADECIMIENTOS

Quiero agradecer su labor a mis dos directores de TFM, Josechu Guerrero Campo y Alejandro Pérez Yus, por su implicación y tiempo dedicado. Especialmente a Alejandro que debido a problemas administrativos no ha podido figurar oficialmente como director, aunque ha actuado en todo momento como tal.

Sistema de visión omnidireccional y de profundidad para guiado de discapacitados visuales

RESUMEN

El objetivo principal de este trabajo es el desarrollo de un sistema de detección del espacio libre en una escena que ayude en la planificación de trayectorias para el guiado de personas y que pueda servir como complemento a personas con discapacidad visual. En particular, en este proyecto se estudia el problema de detectar e identificar el plano del suelo para poder avanzar por las zonas libres de obstáculos, así como la expansión del mismo para permitir la planificación de movimientos durante la navegación. Para realizar estas tareas se hace uso de información de rango y de color.

El uso de cámaras RGB-D permite adquirir este tipo de información, sin embargo, su limitado campo de vista y su rango de distancias impide obtener información más completa de la escena. Para solventar este problema, se propone una combinación de sensores a través de una cámara de profundidad que proporciona observaciones seguras delante de la persona, y de una cámara de color con gran campo de vista que permite recuperar información adicional significativa de la escena. La configuración propuesta se inspira en la visión humana donde la parte central de la retina (fóvea) proporciona información más rica que la periferia y a la vez permite un campo de vista amplio. Para ello se utilizan técnicas de procesamiento de información 3D, y de imagen de color, a diferencia de otros sistemas de navegación recogidos en el estado del arte. La clave del proyecto reside en cómo aprovechar e integrar la información procedente de ambos sensores para conseguir la mayor expansión posible y la detección de elementos de interés, favoreciendo la planificación de la trayectoria a seguir por el usuario durante la navegación. Este Trabajo de Fin de Máster se enfoca en cuatro actividades principales:

1. Realizar una correcta calibración entre ambas cámaras utilizando un novedoso sistema basado en correspondencias de líneas.
2. Desarrollar un sistema que permita segmentar el suelo de la escena visible combinando la información de la cámara de profundidad y de la cámara fisheye, ampliando de forma notable el campo de vista y el alcance de la cámara de profundidad.
3. Propuesta de ayuda a la navegación para la transmisión al usuario del espacio libre en la escena.
4. Evaluación experimental de las técnicas desarrolladas, en escenarios reales tanto interiores como exteriores y con diferente iluminación, generando una base de datos, inexistente hasta ahora para esta combinación de sensores.

Los resultados experimentales demuestran el buen funcionamiento y la robustez del método propuesto. Se consigue expandir el área del suelo detectado unas 10 veces en entornos interiores y unas 20 veces en entornos de exterior, consiguiendo una elevada precisión.

Omnidirectional and depth vision system for visual impaired guidance

ABSTRACT

The main goal of this work is the development of a free space detection system that helps in the planning of trajectories for people guidance and which can serve as a complement to visual impaired people. In particular, this project studies the floor plane detection and identification to advance through free obstacle zones, as well as the expansion of it, allowing movements planning during navigation. In order to perform these tasks, range and color information is used.

The use of RGB-D cameras allows us to obtain this type of information, however, the field of vision and range of distances prevent to obtain more complete information about the scene. To solve this problem, a combination of sensors is proposed. A depth camera, which provides secure observations in front of the person, and a color camera with a large field of view are used. That system permits to recover significant additional information from the scene. The proposed configuration is inspired by the human vision where the central part of the retina (fovea) provides richer information than the periphery and at the same time offers a broad field of view. Unlike other navigation systems collected in the state of the art, 3D information processing and color image techniques are used. The project key is how to integrate and take advantage of the information from both sensors to achieve the greatest possible expansion and detection of interest elements to enhance the path planning followed by the user during navigation. This Master Thesis focuses on four main activities:

1. Performing a correct calibration between both cameras using a novel system based on lines correspondences.
2. Development of a system which allows to segment the floor of the visible scene by combining information from depth and fisheye camera, expanding significantly the field of view and depth camera range.
3. Proposal of navigation aid for transmission of scene free space to the user.
4. Experimental evaluation of the techniques developed in real scenarios both indoor and outdoor with different lighting, generating a unique database, the first existent one with this combination of sensors.

Experimental results show successful performance and the robustness and reliability of the proposed method. The system makes it possible to expand the area of the detected floor up to around 10 times in indoor environments and around 20 times in outdoor environments with high accuracy.

Índice

1. Introducción	1
1.1. Motivación	1
1.2. Estado del arte	2
1.3. Objetivos y alcance	8
1.4. Estructura de la memoria	10
2. Dispositivos utilizados y calibración	11
2.1. Dispositivos utilizados en este proyecto	12
2.2. Calibración intrínseca del fisheye	13
2.3. Calibración extrínseca entre cámaras	15
2.3.1. Notación utilizada	15
2.3.2. Extracción de líneas de la cámara fisheye	16
2.3.3. Extracción de líneas de la cámara de profundidad	18
2.3.4. Correspondencias de líneas entre cámaras	19
3. Detección de suelo mediante profundidad	23
3.1. Representación de la información 3D	23
3.2. Filtrado de las nubes de puntos	24
3.3. Posición de la cámara	25
3.4. Detección y extracción del plano del suelo	26
4. Expansión de suelo	31

4.1. Conversión de la información 3D a la imagen fisheye	32
4.2. Modelos de color	33
4.3. Segmentación inicial de la escena	35
4.4. Expansión de suelo mediante color	36
4.4.1. Creación de la semilla inicial	37
4.4.2. Expansión de la semilla inicial	39
4.5. Uso de información 3D para la expansión	41
5. Propuesta para la navegación	45
5.1. Algoritmo para navegación	46
5.1.1. Zonas de suelo no transitables	47
5.1.2. Discretización del espacio libre	48
5.2. Detección de puertas	51
6. Evaluación experimental	55
6.1. Etiquetado y ground truth	55
6.2. Entornos interiores	57
6.2.1. Entorno de interior con incidencia de luz solar	57
6.2.2. Entorno de interior con gran incidencia de luz solar	60
6.2.3. Entorno de interior con incidencia de luz artificial	63
6.2.4. Entorno de interior con escasa iluminación	66
6.3. Entornos exteriores	69
6.3.1. Entorno de exterior con color homogéneo	69
6.3.2. Entorno de exterior con terreno irregular	72
6.4. Conclusiones de la evaluación	75
7. Conclusiones y trabajo futuro	77
Anexos	79

A. Adquisición y sincronización de la información	81
B. Ampliación de resultados	85
B.1. Resultados de entorno interior con incidencia de luz solar (I3A)	86
B.2. Resultados de entorno interior con gran incidencia de luz solar (I3A) . .	87
B.3. Resultados de entorno interior con incidencia de luz artificial (I3A) . .	89
B.4. Resultados de entorno interior con escasa iluminación (garaje)	90
B.5. Resultados de entorno exterior con terreno irregular (césped)	90
B.6. Resultados de entorno exterior con color homogéneo (exteriores Torres Quevedo)	91
Bibliografía	93
Lista de Figuras	97
Lista de Tablas	103

Capítulo 1

Introducción

1.1. Motivación

La vista es el sentido que más información aporta acerca de una escena y de los objetos ubicados en la misma. Según [6], y tal y como recoge la Organización Mundial de la Salud, en 2015 había un total de 253 millones de personas con discapacidad visual de las cuales 36 millones presentan ceguera total y 217 millones presentan una moderada o severa discapacidad visual. Debido al crecimiento de la población de avanzada edad en el mundo, el número de personas ciegas podría aumentar hasta 115 millones en 2050. Las personas que presentan estos tipos de discapacidad tienen mayores complicaciones a la hora de desplazarse por un ambiente desconocido para ellos. Actualmente disponen de herramientas (bastón, perros lazarillo, etc) que les permiten salvar los obstáculos que se encuentran en su día a día. Sin embargo, con los actuales avances existentes en tecnología, la visión por computador puede aportar mucho para mejorar las condiciones de vida de dicho colectivo. Hay que tener en cuenta que el 81 % de las personas con este tipo de discapacidades son mayores de 50 años y que además, el problema se agrava en los países sub-desarrollados, por lo que es necesario que los sistemas no sean excesivamente caros ni tecnológicamente complejos.

Se busca encontrar un sistema que pueda ser transportado por una persona y que sirva como complemento para personas con cierta discapacidad visual o personas que necesiten ayuda a la hora de desplazarse por un entorno desconocido. Pese a que el guiado de personas con discapacidad visual ha sido un tema relevante durante muchos años en la visión por computador produciéndose avances en algunos temas específicos, todavía es especialmente complicada la integración de todos ellos. Sin embargo, las continuas mejoras en electrónica que permiten aumentar la capacidad y la portabilidad

de los dispositivos hacen que la creación de dispositivos de asistencia sea actualmente, más relevante que nunca.

Conocer los límites del entorno es la base de cualquier sistema de navegación autónomo. Esta tarea es aún más importante para las personas con algún tipo de deficiencia visual ya que entra en juego la seguridad de la propia persona. Además, aunque la obtención de la distribución estructural de una escena a partir de su imagen es una tarea sencilla para cualquier persona, no es fácil para un sistema de inteligencia artificial. No menos importantes son los aspectos neurológicos y psicológicos de la visión, puesto que son los caminos que nos indican cómo debemos procesar la información del entorno para que sea interpretada intuitivamente por cualquier persona. Utilizar los avances biomédicos para así poder transmitir la información obtenida mediante las técnicas de visión por computador, de la mejor forma posible, es otro de los retos más importantes actualmente y de cara al futuro.

Por otro lado, pese a que ya existen varios trabajos orientados al guiado de personas, la gran mayoría se centra en la evitación de los obstáculos más cercanos al usuario. Sin embargo, una adecuada combinación de sensores puede ser de gran utilidad para conocer zonas más alejadas de la posición del usuario, permitiendo detectar el espacio libre por las que la persona puede desplazarse. De esta manera, se puede ayudar a la planificación de la trayectoria a seguir, siempre manteniendo las condiciones de seguridad necesarias. El conocimiento de grandes zonas de paso en un entorno determinado puede ser de utilidad en otros campos, como la robótica.

Este proyecto se ubica en el área de Ingeniería de Sistemas y Automática de la Universidad de Zaragoza, haciendo uso de la rama de visión por computador. El departamento ha puesto a disposición del proyecto todas las herramientas y dispositivos necesarios, como son cámaras, computadores e incluso la ayuda del personal que ha sido necesaria.

1.2. Estado del arte

El desarrollo de sistemas que ayuden a personas con deficiencia visual a desplazarse por el entorno que les rodea es un problema que ha sido estudiado en distintas ocasiones y continúa atrayendo los esfuerzos de muchos investigadores ya sea tanto para la implementación en personas como para su adaptación a otros campos como la robótica. Para ello se han utilizado multitud de sensores y métodos. A continuación se van a

exponer algunos de ellos.

Muchas de las contribuciones no utilizan sistemas de visión sino que emplean complejos sistemas de hardware que requieren equipar tanto al individuo como al propio entorno para alcanzar su objetivo, algo que limita su uso a entornos relativamente limitados como los interiores. El sistema desarrollado por Oktem y col. [15] es un claro ejemplo que utiliza la tecnología inalámbrica para que el sistema que lleva acoplado la persona se comuniquen con el sistema instalado en un edificio. De esta forma el usuario obtiene información de alrededor porque el sistema del edificio sabe en todo momento dónde está el individuo dependiendo del punto desde el que éste se conecte.

Otro sistema que emplea sensores complejos es el desarrollado por Solon y col. [13] donde acopla a un bastón para ciegos múltiples sensores de ultrasonidos y un sistema GPS que permite conocer la posición del individuo en todo momento. El inconveniente de este sistema es que el sistema GPS tiene problemas de cobertura, por ejemplo en zonas donde exista frondosa vegetación o en el caso de navegar por el interior de un edificio, donde únicamente funcionaría la parte implementada por los sensores de ultrasonidos. Además, este tipo de sensores de ultrasonidos proporcionan una escasa resolución angular así como una información muy ruidosa.

Otro tipo de sensores utilizados para el guiado han sido los sensores láser, que son capaces de ofrecer una información más rica y exacta del entorno que los de ultrasonidos. El inconveniente de este tipo de sensores es su elevado precio, además de que suelen ser pesados. Un ejemplo de este tipo de tecnología es el utilizado por Ueda y col. [21], que dispone de un escáner láser 3D montado en la cabeza utilizado para detectar obstáculos y que dispone de una interfaz de audio para el usuario.

El uso de cámaras como sensor principal para un sistema de asistencia está cada vez más extendido, en primer lugar por su funcionalidad, ya que la visión por computador permite abordar una gran cantidad de tareas: detección de obstáculos, reconocimiento de objetos, de personas... y todo ello con un único sensor. Además, el coste de este tipo de sensores es muy bajo, permitiendo también una gran miniaturización del mismo, lo que lo hace fácilmente transportable.

Un ejemplo en el que se utilizan cámaras para el desarrollo de un sistema de navegación es el sistema que propone Wong y col. [25] que emplea visión estéreo para detectar obstáculos que estén delante del individuo. Uno de los inconvenientes de los sistemas estéreo es la necesidad de que exista cierto paralaje entre las imágenes capturadas por las cámaras ya que sino la distancia a la que se estime que está el

objeto puede ser poco precisa. Además, en los entornos en los que el obstáculo y el suelo sean de colores similares y con poca textura, el sistema no siempre podrá detectar con acierto el obstáculo. Otro sistema de navegación es el que presenta Xu y col. [26] donde propone un sistema implementado en una PDA donde utiliza tanto información de visión como GPS. El inconveniente de este sistema es que necesita que en el suelo exista un camino específico del que se sabe tanto su color como su textura para guiar al individuo por dicha vía. El ejemplo del suelo que necesita se puede observar en la Figura 1.1.



Figura 1.1: Ejemplo del suelo que necesita el sistema propuesto en [26]. Este tipo de suelo se ha ido incorporando en ciudades de Japón. De esta forma, las personas invidentes puedan caminar por dichas vías con su bastón de ciego con la certeza de que no existe ningún tipo de obstáculo en esa zona

Por otra parte, el procesamiento de imagen de color es un campo que ha evolucionado mucho en las últimas dos décadas consiguiendo algoritmos realmente robustos para determinadas tareas. En temas relacionados con la segmentación de imágenes para la identificación del suelo o de un camino o carretera hay numerosas propuestas que utilizan diferentes características de las imágenes para llevar a cabo su cometido. Ulrich y col. [22] propone un sistema que se basa en el modelo de color HSI para la detección de obstáculos. En concreto utiliza el tono y la iluminación para sacar un patrón del suelo de la escena. A partir ahí, realiza comparaciones de los histogramas del suelo referencia y de su imagen original. Todo lo que se aleje de un umbral determinado, será obstáculo para el sistema. El inconveniente de este sistema es su dudosa robustez debido a que los resultados que muestra parten de imágenes donde los obstáculos tienen colores vivos y muy diferentes del suelo.

Birchfield y col. [14] presenta un algoritmo alternativo en el que utiliza la información de rectas además de otras técnicas relacionadas con probabilidades. El sistema que propone se centra en identificar la orientación de las rectas y de clasificarlas en verticales u horizontales. Con esta información consigue segmentar la escena

obteniendo el suelo de la misma. Los resultados que proporciona este algoritmo son realmente buenos pero el sistema necesita tener una zona de la imagen donde siempre exista suelo para poder expandirla a partir de la información extraída por el sistema.

Las técnicas de identificación de la vía transitable se han extrapolado al mundo de la automoción. En concreto se ha utilizado en el desarrollo de sistemas tipo ADAS (Advanced Driver Assistance Systems). El sistema más conocido es el propuesto por Thrun y col. [10] donde se utiliza información láser combinada con información de color. Con el sensor láser consigue extraer la localización de la zona del suelo más cercana al vehículo. A continuación analiza la información del suelo referencia y extrae el resto del suelo de la imagen utilizando modelos de apariencia. Otro sistema relacionado con los sistemas ADAS es el que presenta Álvarez y col. [2] con el que consigue resultados muy interesantes en escenas con iluminación heterogénea.

Sin embargo, el uso de cámaras RGB monoculares tiene también algunas desventajas, ya que pese a que la información que se obtiene es similar a la información que puede ver un ojo humano, es casi imposible conseguir el nivel de abstracción que realiza nuestro cerebro para interpretar correctamente las imágenes. Otros problemas pueden ser la aparición de oclusiones, condiciones de luz cambiantes, desenfoques, “motion blur” debido al movimiento de la escena, etc. Además, el coste computacional de algunos algoritmos de visión impiden que el sistema trabaje a tiempo real. Algunos de estos problemas pueden solucionarse con cámaras más complejas que una simple cámara monocular.



Figura 1.2: Microsoft Kinect (izquierda) y Asus Xtion Pro Live (derecha)

Pese a que, como se ha comentado anteriormente, la profundidad de un objeto puede obtenerse utilizando un sistema de visión estéreo, la ausencia de textura en una escena puede provocar que este sistema falle. Es por ello que surgen las cámaras de profundidad, que son capaces de obtener un mapa de profundidad de la escena (nube de puntos) ya sea mediante emisión de pulsos de luz infrarroja y la medición del retraso tras rebotar la luz en la escena y llegar al receptor, *Time-Of-Flight* (TOF), o mediante

la medida de la distorsión producida tras la proyección de un patrón de luz sobre la escena, *Structured-Light cameras* (SL). La llegada de este tipo de cámaras al mercado de consumo ha reducido su coste en gran medida. Exponentes como la Microsoft Kinect o la Asus Xtion Pro (Figura 1.2) han permitido que este tipo de sensores tengan un gran impacto en este campo. Estas cámaras combinan la información de una cámara RGB convencional con la información de profundidad, de ahí la nomenclatura de cámaras RGB-D.

El procesamiento de la información 3D de sistemas RGB-D, pese a ser un campo relativamente reciente, muestra propuestas interesantes de sistemas para la evitación de obstáculos, como es el caso que propone Peasley y col. [16] donde se utiliza una cámara Kinect para la detección de obstáculos a partir de la extracción del plano del suelo por mínimos cuadrados. En este caso la cámara es transportada por un robot, pero el inconveniente que tiene es que necesita una condición de arranque donde no existan obstáculos para poder extraer el plano del suelo a partir de mínimos cuadrados. Además, en el caso de que exista una posible presencia de puntos espurios, cosa que suele ser habitual en este tipo de sensores dependiendo del entorno, la precisión del plano del suelo extraído puede llegar a ser dudosa. Schäfer y col. [20] presenta un sistema de evitación de obstáculos en terrenos con vegetación basado en tecnología láser. En concreto el sistema se centra en la distinción de obstáculos sólidos que son los que realmente impiden el paso, de los obstáculos debidos a vegetación tales como ramas con hojas, pequeños arbustos... y lo implementan en un robot preparado para desplazarse por terrenos abruptos.

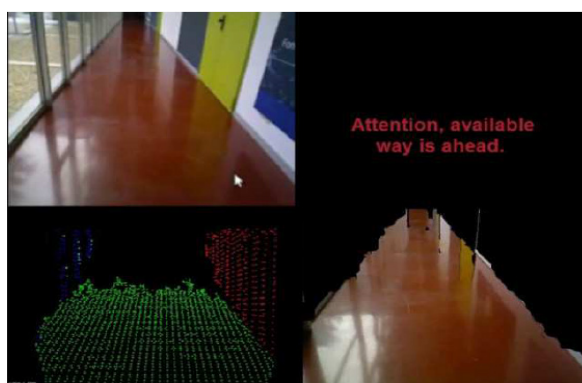


Figura 1.3: Ejemplo del resultado obtenido en el trabajo de Aladren y col. [1]

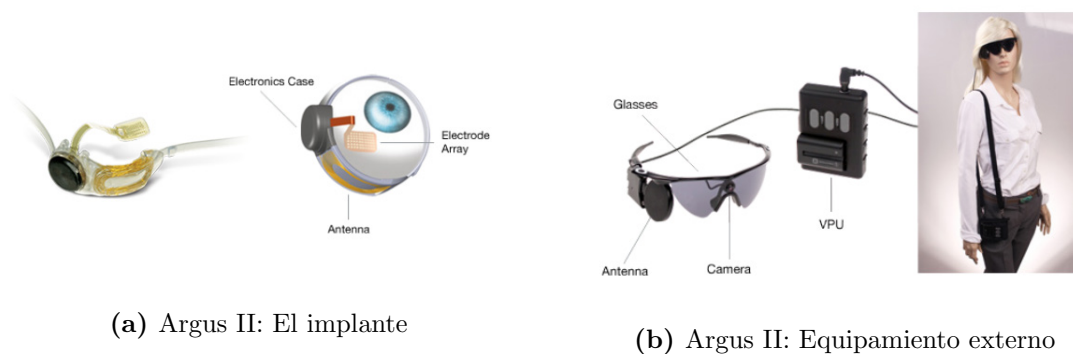
El trabajo de Aladren y col. [1], muestra un ejemplo de lo que se puede hacer combinando la información de profundidad y la de color obtenida por la cámara RGB-D. A través de un algoritmo robusto (RANSAC) se obtiene el plano del suelo a partir de la nube de puntos obtenida por el sensor de profundidad. Posteriormente

se realiza la expansión del suelo utilizando la información de color, así como mediante información obtenida a partir de características geométricas (Figura 1.3). Wang y col. [24] integra técnicas de machine learning para la segmentación de la escena con RGB-D. Sin embargo, el mayor inconveniente de las cámaras RGB-D es el escaso campo de vista del que disponen, lo que limita en gran medida la detección de espacio libre en la escena para la navegación.

A lo largo de esta sección se han citado muchos sistemas cuya interfaz de comunicación con el usuario es mediante sistemas de audio, o mediante sistemas táctiles. Sin embargo, actualmente es posible comunicarse con el usuario por medio de un sistema de prótesis visual. Esto consiste en implantes que aplican estimulación eléctrica en la retina, el nervio óptico o la corteza cerebral por medio de un conjunto de electrodos que permiten a los pacientes percibir puntos de luz llamados fosfenos [7]. Usando una matriz de electrodos es posible generar una red de fosfenos similares a una imagen de puntos de baja resolución [9]. Se requiere un dispositivo que recoja información de la escena, la procese y la transforme en una señal de excitación. El dispositivo de captación de la escena suele ser una cámara convencional y en su procesamiento es donde entran las técnicas de visión por computador para lograr mostrar al paciente por medio de esos estímulos visuales la información más trascendental del mundo que le rodea. La conexión entre la cámara y el ordenador que procesa la información es inalámbrica.

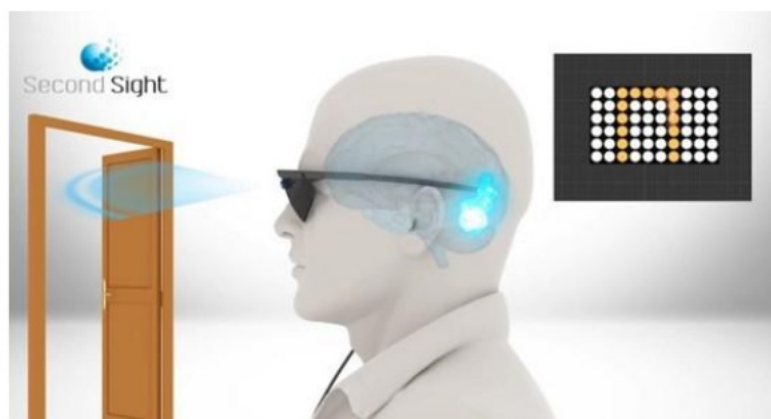
Actualmente hay ya sistemas comerciales disponibles, como el Argus II Retinal Protheses System perteneciente a la empresa estadounidense Second Sight Medical Products, Inc. Se trata de un implante epirretinal de 60 electrodos en el que existe un procesamiento de la imagen previo (Figura 1.4). El segundo sistema aprobado en Europa es el Alpha-IMS de los alemanes Retina Implant AG. Se trata de un implante subretinal de 1500 fotodiodos que capturan la luz proyectada sobre la retina y la transforman en señales eléctricas. En este caso no hay tratamiento de la imagen capturada, pero nos sirve para estimar la resolución que puede alcanzarse con este tipo de prótesis.

Por desgracia, la resolución de la rejilla de fosfenos producidos se ve limitada por la biología, la tecnología y la seguridad del propio paciente [11]. Actualmente se están desarrollando dispositivos que proporcionan unas decenas de fosfenos. Teniendo en cuenta dicha resolución se han realizado importantes esfuerzos en la aplicación de algoritmos de visión para conseguir mejoras [4], [3], aunque la manera de procesar y codificar la información en el dispositivo de manera útil y significativa es aún una



(a) Argus II: El implante

(b) Argus II: Equipamiento externo



(c) Ejemplo de visualización

Figura 1.4: Argus II Retinal Prosthesis. (a) El implante con sus principales componentes. (b) Equipamiento externo: gafas con cámara, antenas para la comunicación inalámbrica y la Unidad de Procesamiento de Video (VPU). (c) Imagen comercial de la visualización de una puerta por medio de fosfenos

cuestión abierta. Todavía queda un largo trabajo en la parte ingenieril para lograr que un paciente ciego con una prótesis visual pueda interpretar su entorno por medio del sentido de la vista. Quizá es este el factor que menos se ha desarrollado en la actualidad, por lo que actualmente es una vía importante de investigación.

1.3. Objetivos y alcance

El objetivo principal del proyecto es desarrollar e implementar un sistema de visión que combina información de profundidad y color para el guiado de personas con discapacidad visual, centrándose principalmente en la percepción y dando un primer paso en la parte de navegación. Proponemos combinar una cámara de profundidad que proporciona observaciones seguras delante de la persona, con una cámara a color con gran campo de vista que permite recuperar información adicional significativa de

la escena. La configuración propuesta se inspira en la visión humana donde la parte central de la retina (fóvea) proporciona información más rica que la periferia y a la vez permite un campo de vista amplio. Para ello se utilizan técnicas de procesamiento y tratamiento de información 3D, así como de procesamiento de imagen donde se demuestre el beneficio de poder combinar la información de profundidad y de color, respecto a otros sistemas de navegación recogidos en el estado del arte. El sistema deberá, previa calibración de los sensores, ser capaz de reconocer el suelo de una escena desconocida y extender la información de interés de la escena hasta zonas donde el sensor de rango no puede llegar, con el fin de detectar las zonas de espacio libre. La clave del proyecto residirá en cómo aprovechar e integrar la información procedente de ambos sensores para conseguir la mayor expansión posible y la detección de elementos de interés, favoreciendo de esta manera la planificación de la trayectoria a seguir por el usuario durante la navegación.

Las tareas a realizar para cumplir los objetivos son las siguientes:

1. Estudio de la documentación pertinente:

- Artículos de investigación sobre el guiado de discapacitados visuales presentes en el estado del arte.
- Estudio y comprensión del entorno de programación bajo el que se va a trabajar.
- Librerías para trabajar con algoritmos de procesamiento y tratamiento de información 3D.
- Librerías para trabajar con algoritmos de procesamiento de imagen y visión por computador.

2. Implementación del sistema. El trabajo se desarrolla en tres fases principales:

- Realizar una correcta calibración entre ambas cámaras utilizando un novedoso sistema basado en correspondencia de líneas.
- Desarrollar un sistema que permita segmentar el suelo de la escena visible combinando la información de la cámara de profundidad y de la cámara fisheye, ampliando de forma notable el campo de vista y el alcance de la cámara de profundidad.
- Propuesta de ayuda a la navegación para la transmisión al usuario del espacio libre en la escena y otros elementos de interés.

3. Evaluación experimental del sistema implementado en diferentes entornos con iluminación variada y valoración de resultados. Generación de una base de datos, inexistente hasta ahora para esta combinación de sensores, que pueda ser utilizada en el futuro.

1.4. Estructura de la memoria

Tras esta introducción, la estructura de este proyecto es la siguiente:

- **Capítulo 2:** Descripción del entorno de trabajo y los dispositivos utilizados, así como la calibración entre ambas cámaras.
- **Capítulo 3:** Detección del plano del suelo de la escena utilizando el sensor de profundidad.
- **Capítulo 4:** Implementación de la extensión del plano del suelo utilizando la información de color procedente de la cámara fisheye, así como información 3D complementaria.
- **Capítulo 5:** Propuesta de un sistema de navegación mediante la discretización del espacio libre detectado en la escena en los capítulos anteriores.
- **Capítulo 6:** Evaluación experimental del algoritmo en diferentes entornos con iluminación diversa, obteniéndose los resultados cuantitativos pertinentes.
- **Capítulo 7:** Recopilación de las conclusiones y opciones de trabajo futuro a realizar.
- Finalmente se presenta un anexo dedicado a la sincronización de las cámaras en ROS, y otro a la ampliación de resultados.

Capítulo 2

Dispositivos utilizados y calibración

El desarrollo de cámaras RGB-D en el mercado de consumo ha sido un soplo de aire fresco en campos como la robótica o la visión por computador, debido a la gran reducción de precios que ha conllevado. Con este tipo de sensores es posible obtener el 3D de una escena de forma simultánea a la imagen RGB, con un único dispositivo y sin ningún coste computacional extra. Sin embargo, las especificaciones de muchas de las cámaras RGB-D del mercado pueden quedarse cortas dependiendo de la aplicación, debido al limitado rango de la cámara de profundidad que no permite ser usada en muy cortas y largas distancias o a su limitado campo de vista. Además, existen otras limitaciones como son la imposibilidad de detectar superficies de determinados materiales como el cristal, o los problemas de funcionamiento en entornos donde la luz del sol incida directamente.

En nuestro caso resultan problemáticos los dos primeros puntos. En la inmensa mayoría de las aplicaciones basadas en la visión y relacionadas con la movilidad tener un gran campo de vista es importante. Es por ello, que surge la idea de usar sistemas más sofisticados. En particular, se propone sustituir la cámara RGB convencional por una cámara tipo fisheye que permita captar una mayor cantidad de información de la escena. De esta manera podemos tener una cámara de profundidad fija que apunta hacia el suelo próximo al usuario para detectar obstáculos cercanos, y a la vez una imagen de color con un gran campo de vista que permita recuperar información adicional significativa de la escena.

2.1. Dispositivos utilizados en este proyecto

Todo el desarrollo del proyecto se ha llevado a cabo bajo el entorno de programación ROS Kinetic¹ (Robot Operating System) y se ha implementado utilizando el lenguaje de programación C++. Para el manejo del sensor RGB-D se ha utilizado el driver OpenNI 2.

Se ha hecho uso de las librerías de uso libre. En concreto se ha utilizado la librería PCL² (Point Cloud Library) para el procesamiento de la información del senso de rango y la librería OpenCV³ para el procesamiento de imagen.

Los sensores utilizados para la realización de este proyecto han sido la cámara RGB-D Asus Xtion Pro Live con una resolución de 640×480 píxeles con un campo de vista de $43^\circ \times 57^\circ$ y la cámara uEye UI-3580CP, con una resolución de 2560×1920 píxeles y con una lente fisheye Lensagon CF5M1414, con un campo de vista de 182° . Sin embargo, para acelerar la adquisición y los tiempos de cómputo, se ha reducido la resolución de la cámara uEye a la mitad, quedando en 1280×960 . En la Figura 2.1 se muestra el sistema de sensores utilizado en este proyecto.



Figura 2.1: Sistema de cámaras utilizado en este proyecto en el que se combinan la cámara fisheye y la cámara RGB-D

Este nuevo sistema necesita de una sincronización de la adquisición de imágenes entre ambos sensores (que se ha conseguido utilizando ROS, y que está explicada en el Anexo A) y de una calibración extrínseca de las cámaras para conocer la posición relativa entre ambas.

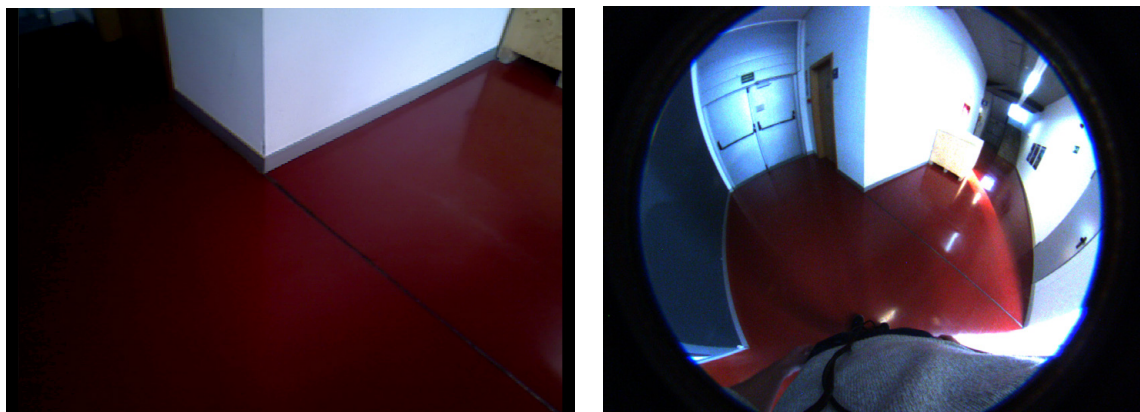
La imagen de la Figura 2.2 ha sido tomada simultáneamente con la cámara RGB-D (Figura 2.2a) y con la fisheye (Figura 2.2b). Se puede observar como el campo de vista

¹<http://www.ros.org/wiki/>

²<http://pointclouds.org/>

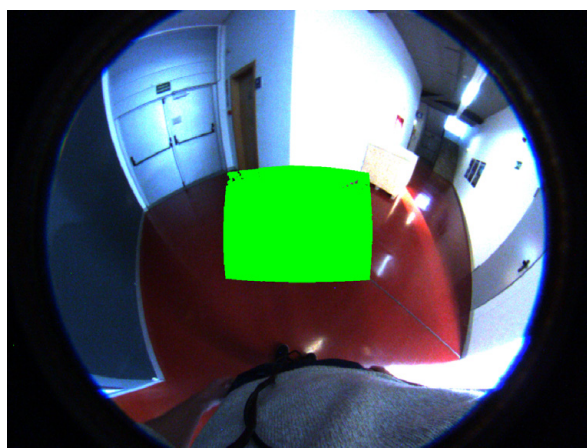
³<https://opencv.org/>

aumenta considerablemente en el segundo caso respecto al primero. En la Figura 2.2c se muestra el mapeo de la nube de puntos obtenida con la cámara de profundidad sobre la imagen de fisheye una vez realizada la calibración. A continuación se va a mostrar como se ha realizado la calibración de ambas cámaras.



(a) Campo de vista de la cámara RGB-D

(b) Campo de vista de la cámara fisheye



(c) Mapeo de los puntos de profundidad sobre la imagen del fisheye

Figura 2.2: (a) Escena vista desde una cámara RGB-D convencional. (b) Misma escena vista desde la cámara fisheye. (c) Proyección de la nube de profundidad (en verde) sobre la imagen del fisheye una vez realizada la calibración

2.2. Calibración intrínseca del fisheye

El primer paso ha sido realizar la calibración de la cámara fisheye, utilizando el modelo paramétrico de Scaramuzza y col. [19], que considera la imagen omnidireccional como una imagen altamente distorsionada. La calibración consiste en obtener los parámetros del polinomio que describe esa distorsión. Con este modelo no es necesario proporcionar un modelo específico del sensor y funciona con todo tipo de cámaras

proyectivas, catadióptricas y dióptricas. Para ello, se han tomado 21 imágenes diferentes de un patrón de ajedrez (Figura 2.3), tratando de que abarcaran todos los grados de libertad de la cámara con el fin de que la calibración sea lo más exacta posible. Para realizar esta calibración se ha utilizado la toolbox de Scaramuzza ⁴ disponible en Matlab.



Figura 2.3: Muestra de algunas de las 21 imágenes tomadas mediante la cámara fisheye del patrón tipo ajedrez para realizar la calibración intrínseca de la misma

Usando el modelo de Scaramuzza, los puntos en el mundo $\mathbf{X}^F = (X, Y, Z)$ tienen el origen de coordenadas en el centro óptico de la cámara O^F , donde la componente z^F del sistema de referencia sigue el eje del sistema dióptrico. El plano del sensor (x_s, y_s) , ortogonal al eje z^F , es un plano teórico en el que las coordenadas están todavía en unidades métricas. Las imágenes están representadas en el plano imagen $\mathbf{u} = (u, w)$, donde la posición de los puntos está expresada en píxeles. Se asume que hay desalineamiento y deformación entre el plano imagen y el plano del sensor, que viene dado por la transformación afín $\mathbf{x}_s = \mathbf{A}\mathbf{u} + \mathbf{t}$, donde $\mathbf{t} = (u_0, w_0)$ es el centro imagen. El vector \mathbf{v} , que apunta a un punto del mundo \mathbf{X} desde O^F , sigue la siguiente ecuación, que relaciona un punto del plano imagen en píxeles y un punto 3D de la escena:

$$\lambda \cdot \mathbf{v} = \lambda \mathbf{g}(\mathbf{A}\mathbf{u} + \mathbf{t}) = \mathbf{P} \cdot \mathbf{X}, \quad \lambda > 0 \quad (2.1)$$

Donde \mathbf{P} es la matriz de proyección perspectiva y la función $\mathbf{g}(x_s, y_s)$, que relaciona un punto del plano del sensor con el vector \mathbf{v} , se define de la siguiente manera:

$$\mathbf{g}(x_s, y_s) = (x_s, y_s, h(x_s, y_s))^T = (x_s, y_s, h(\rho_s))^T \quad (2.2)$$

$$h(x_s, y_s) = a_0 + a_2(\rho_s)^2 + \dots + a_N(\rho_s)^N \quad (2.3)$$

Donde la función h de la proyección de la imagen puede ser descrita por una expansión de la serie de Taylor cuyos coeficientes a_0, a_2, \dots, a_N , (ya que $a_1 = 0$ para

⁴<https://sites.google.com/site/scarabotix/ocamcalib-toolbox>

una cámara fisheye), se estiman resolviendo un problema de minimización lineal por mínimos cuadrados y donde $\rho = \sqrt{(x_s)^2 + (y_s)^2}$. Estos coeficientes junto con la matriz \mathbf{A} y el vector \mathbf{t} son los parámetros necesarios para la calibración intrínseca.

2.3. Calibración extrínseca entre cámaras

Asumiendo que la calibración intrínseca de la cámara de profundidad proporcionada por el fabricante es correcta, el siguiente paso es realizar la calibración extrínseca entre ambas cámaras. Para ello se ha utilizado el método diseñado por Pérez-Yus y col. [17], el cual tiene importantes ventajas con respecto a otros enfoques propuestos en la literatura:

- No se necesita superposición del campo de vista entre las cámaras
- Se puede utilizar para calibrar diferentes combinaciones de sensores 3D de rango y de imagen, siempre que al menos uno de ellos sea una cámara de profundidad
- No es necesario utilizar un patrón de calibración

Este método de calibración se basa en la observación y emparejamiento de líneas en la escena desde diferentes sensores que son usadas para formular restricciones en las posiciones relativas de las cámaras. Para realizar la calibración con este método se necesita lo siguiente:

1. Un sensor debe ser capaz de extraer los parámetros para definir completamente una línea en el espacio 3D (por ejemplo una cámara de profundidad).
2. Al menos un sensor debe ser capaz de extraer las líneas en el plano proyectivo (por ejemplo una cámara estándar u otra cámara de profundidad).

2.3.1. Notación utilizada

Usamos *línea* para referirnos a una línea en el espacio 3D, y *segmento* para referirnos a un conjunto de puntos colineales encontrados en una imagen convencional. Matemáticamente, una *línea* es un conjunto de puntos $\mathbf{p} \in \mathbb{R}^3$ que satisfacen la siguiente ecuación:

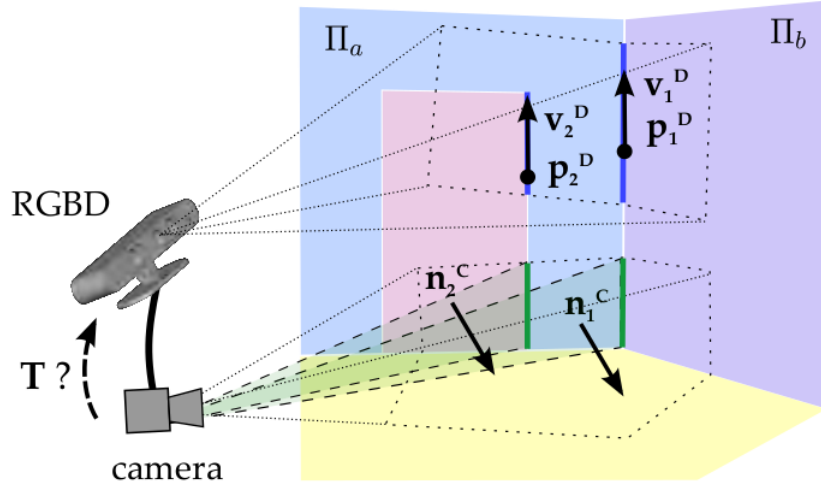


Figura 2.4: Observación de líneas en una escena por una cámara RGB-D y una cámara convencional unidas por una estructura rígida y con campos de vista que no están solapados. Las correspondencias entre líneas se utilizan para formular restricciones geométricas para calcular la posición relativa (T) entre las cámaras. Imagen obtenida de [17]

$$\mathbf{p} = \mathbf{p}_0 + \lambda \mathbf{v} = (p_{0_x}, p_{0_y}, p_{0_z}) + \lambda (v_x, v_y, v_z), \quad \forall \lambda \in \mathbb{R} \quad (2.4)$$

siendo $\mathbf{p}_0 \in \mathbb{R}^3$ un punto en la línea y $\mathbf{v} \in \mathbb{R}^3$ el vector director de la línea (Figura 2.4). También se define el *plano proyectivo* de la línea, Π , como el plano 3D que contiene la línea y el origen del sistema de referencia (es decir, el centro óptico de la cámara). La normal $\mathbf{n} \in \mathbb{R}^3$ de este plano (ver Figura 2.4), también conocido como *vector de momento* de la línea, es el vector perpendicular a \mathbf{p}_0 y \mathbf{v} . Es decir, $\mathbf{n} = \mathbf{p}_0 \times \mathbf{v}$. Dependiendo de la cámara utilizada la extracción de líneas será diferente, y es lo que se va a tratar en las siguientes secciones.

2.3.2. Extracción de líneas de la cámara fisheye

Para la extracción de líneas en la cámara fisheye, se ha utilizado el algoritmo realizado por Bermudez-Cameo y col. [5] en el que se presenta un método de extracción de líneas para cámaras omnidireccionales no calibradas con simetría de revolución. Una de las aportaciones de este trabajo, es que permite resumir la deformación de la proyección de rectas en las diferentes cámaras omnidireccionales a través de un único parámetro, permitiendo así la calibración de un gran número de tipos de cámara. Sin embargo, en nuestro proyecto se va a utilizar la calibración de Scaramuzza obtenida en el apartado anterior, ya que es más precisa, por lo que únicamente nos interesa la

parte de extracción de líneas.

El método [5] utiliza el modelo de cámara esférica para describir la proyección de un punto en sistemas dióptricos (Figura 2.5). Cada punto 3D del mundo \mathbf{X}_i se proyecta primero en un punto \mathbf{x}_i sobre una esfera unitaria alrededor del punto de vista del sistema. En el caso de nuestra lente de fisheye equiangular este punto es proyectado a $\hat{\mathbf{x}}_i$ en el plano imagen utilizando la función de proyección $\hat{r} = f_c \phi$ donde ϕ es el ángulo de elevación en la esfera y f_c es el parámetro principal de calibración. $\hat{\mathbf{x}} = (\hat{r}, \hat{\theta})$ se expresa en coordenadas polares con respecto al centro de la imagen.

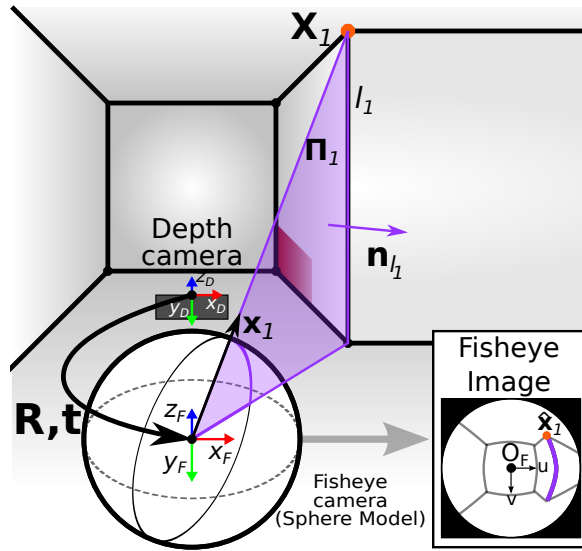


Figura 2.5: Esquema del sistema en una escena del mundo 3D y la correspondiente imagen fisheye

Al contrario que en las cámaras convencionales, las líneas 3D en el espacio no se proyectan como una línea recta en las imágenes omnidireccionales, sino que son proyectadas como líneas curvas. En la Figura 2.5 puede verse resaltada una línea vertical en el modelo esférico y su proyección en la imagen de fisheye. La forma de la proyección de la línea en la imagen cambiará en función del tipo de cámara omnidireccional.

La proyección de una línea l_i en el espacio 3D puede ser respresentado por la normal del plano proyectivo $\mathbf{\Pi}_i$ definido por la propio línea y el punto de vista del sistema, con normal $\mathbf{n}_{l_i} = (n_x, n_y, n_z)^T$. Los puntos \mathbf{X} que se encuentran en una línea 3D l se proyectan a puntos \mathbf{x} satisfaciendo la condición $\mathbf{n}_l^T \mathbf{x} = 0$. La restricción para los puntos de la línea de proyección en las coordenadas imagen para sistemas equiangulares dióptricos con simetría de revolución es:

$$n_x \hat{x} + n_y \hat{y} + n_z \hat{r} \cot(\hat{r}/f_c) = 0 \quad (2.5)$$

donde \hat{x} e \hat{y} son las coordenadas imagen centradas en el punto principal, $\hat{\mathbf{x}} = (\hat{x}, \hat{y})$. La proyecciones de las líneas sobre la imagen son no-polinomiales y no tienen forma cónica. Para extraerlas es necesario resolver un problema de minimización [5].

Así pues, a la hora de implementar esta extracción de líneas se han dado los siguientes pasos:

1. Se aplica un filtro Canny [8] para la extracción de bordes, así como sus gradientes, sobre una imagen en escala de grises.
2. Los puntos de los bordes son agrupados en *fronteras* formadas por puntos consecutivos en la imagen.
3. A la orientación del gradiente se le aplica un filtro paso bajo para reducir el ruido.
4. Utilizando la variación de la orientación del gradiente se realiza una división de las fronteras para poder ajustar las rectas (*splitting*). A cada división se le denomina subfrontera. El método busca minimizar el número de subfronteras. Por lo tanto, una subfrontera puede contener más de un segmento.
5. A cada una de las fronteras se le aplica un RANSAC. Se seleccionan de forma aleatoria puntos de una frontera para generar líneas candidatas que son votadas por los otros puntos de la frontera.
6. Como se ha comentado antes, debido a la naturaleza proyectiva de este tipo de cámaras no se puede calcular el vector director \mathbf{v} , ni ningún punto 3D \mathbf{p} . No obstante, a partir de los puntos que forman los segmentos de la imagen se obtendrán los vectores normales \mathbf{n} de sus planos proyectivos. Esas normales \mathbf{n} se calculan utilizando la expresión 2.5

Este método estaba programado en Matlab, por lo que hubo que adaptarlo a nuestro lenguaje de programación: C++. En la Figura 2.6 se pueden ver ejemplos del resultado.

2.3.3. Extracción de líneas de la cámara de profundidad

La extracción de líneas en 3D mediante la cámara de profundidad se realiza mediante la intersección de planos en 3D. Por ejemplo en la Figura 2.4, la línea $\mathbf{p}_1, \mathbf{v}_1$ es la intersección de los planos $\mathbf{\Pi}_a$ y $\mathbf{\Pi}_b$. Los planos 3D se extraen utilizando un RANSAC (método explicado en la sección 3.4). Un plano $\mathbf{\Pi}_i$ tiene normal \mathbf{n}_i y distancia al origen

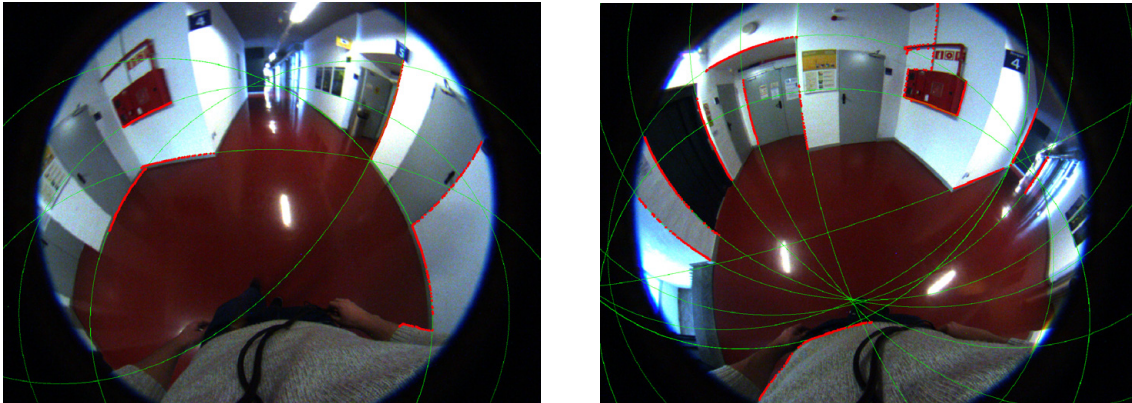


Figura 2.6: Ejemplos de la extracción de líneas de la cámara fisheye: En rojo aparecen las fronteras obtenidas mediante el Canny tras la agrupación, y en verde las líneas obtenidas a partir de esas fronteras

d_i , por tanto, todos los puntos \mathbf{X} pertenecientes al plano satisfacen $\mathbf{n}_i \cdot \mathbf{X} + d_i = 0$. Para calcular la línea 3D entre los dos planos Π_a y Π_b , se obtiene la dirección \mathbf{v} como el producto vectorial entre sus normales, $\mathbf{v} = \mathbf{n}_a \times \mathbf{n}_b$. Un punto 3D de la línea, \mathbf{p}_0 , se obtiene como el punto más cercano al origen que cumple las ecuaciones $\mathbf{n}_a \cdot \mathbf{p}_0 + d_a = 0$ y $\mathbf{n}_b \cdot \mathbf{p}_0 + d_b = 0$.

2.3.4. Correspondencias de líneas entre cámaras

Después de la extracción de líneas se crea un fichero con un conjunto de correspondencias $\Gamma_{i=1\dots N_L}$. Siendo D la información obtenida con la cámara de profundidad y F la obtenida por la cámara fisheye. Cada correspondencia Γ_i consiste en una línea 3D totalmente parametrizada de D y la normal del plano proyectivo de F , es decir $\Gamma_i = \{\mathbf{p}_i^D, \mathbf{v}_i^D, \mathbf{n}_i^F\}$. En este fichero se emparejan todas las líneas con todas, para luego en el proceso de calibración hacer una primera selección (filtrado) de cuáles son posibles emparejamientos válidos y finalmente aplicar un algoritmo robusto entre los posibles candidatos, como veremos a continuación.

El escenario utilizado para obtener las imágenes de las que obtener las líneas puede verse en la Figura 2.7.

La razón por la que se ha elegido una escena como esta es por la simplicidad de la misma, permitiendo obtener un gran número de emparejamientos correctos, en los que no haya interferencias con otro tipo de líneas. Se toma una secuencia de imágenes sincronizadas entre ambos dispositivos, rotando y trasladando las cámaras alrededor de la escena de calibración, mientras de forma automática se van almacenando

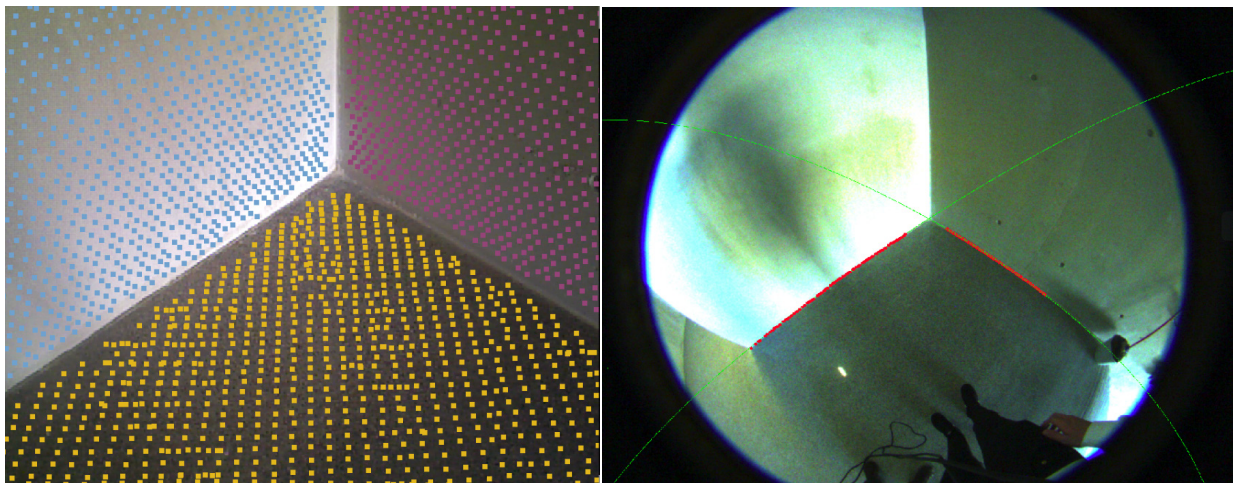


Figura 2.7: Lugar en el que se toma la secuencia necesaria para la calibración

los emparejamientos entre líneas en el fichero, como se ha comentado anteriormente. Durante la obtención de la secuencia hay que tratar de tener en cuenta todos los grados de libertad de las cámaras, por lo que utilizar una escena en la que hay un triedro bien definido es de gran utilidad para ello, ya que permite obtener líneas en las tres direcciones del espacio. Aunque bien es cierto que este método de calibración funciona con cualquier tipo de escena siempre que haya correspondencias suficientes, una escena muy confusa y con muchas líneas daría como resultado un gran número de outliers, por lo que sería necesario una mejor aproximación inicial de la calibración para filtrarlas.

Así pues, en cuanto a la obtención de correspondencias de líneas, una vez se tiene el fichero con los emparejamientos, el siguiente paso es procesar la información del mismo en Matlab. Se parte en nuestro caso de 36916 emparejamientos, por lo que enfrentarlos todos contra todos sería demasiado costoso computacionalmente. Por tanto, gracias al conocimiento a priori que se tiene del posicionamiento de ambas cámaras (Figura 2.1), se eliminan los emparejamientos que disten más 10 cm entre ellos y cuya diferencia de orientación sea mayor a 5° , para que no se tengan en cuenta durante la optimización. De esta forma, se consiguen filtrar un gran número de ellos pasando a tener 3829 emparejamientos.

A partir de estas correspondencias se escoge un conjunto de 3 emparejamientos aleatorios para realizar la estimación tanto de la matriz R de rotación (ecuación 2.6) como el vector t de traslación (ecuación 2.7) entre cámaras de forma desacoplada. A continuación, este proceso se introduce dentro de un método robusto tipo RANSAC para obtener el máximo consenso (inliers), quedando en nuestro caso únicamente 2661 correspondencias.

$$(\mathbf{n}^F)^T \cdot R\mathbf{v}^D = 0 \quad (2.6)$$

$$(\mathbf{n}^F)^T \cdot (R\mathbf{p}^D + \mathbf{t}) = 0 \quad (2.7)$$

Por último, con estas correspondencias inliers se realiza una optimización por mínimos cuadrados del residuo de la ecuación 2.8 para la matriz de rotación R y de la ecuación 2.9 para el vector de traslación t . El objetivo es encontrar la estimación de máxima verosimilitud (MLE) para la transformación entre ambas cámaras.

$$\arg \min_{\mu} \sum_{i=1}^{N_L} (\mathbf{n}_i^T \cdot e^{\mu} R\mathbf{v}_i)^2 \quad (2.8)$$

donde e^{μ} es el mapa exponencial del incremento de rotación μ en R .

$$\arg \min_{\mathbf{t}} \sum_{i=1}^{N_L} \left(\mathbf{n}_i^T \cdot \frac{R\mathbf{p}_i + \mathbf{t}}{\|R\mathbf{p}_i + \mathbf{t}\|} \right)^2 \quad (2.9)$$



(a) Nube de puntos con superposición del color del fisheye



(b) Nube de puntos con superposición del color del fisheye

Figura 2.8: Ejemplos para visualizar el funcionamiento del método de calibración

Así pues, la transformación entre ambas cámaras que se ha obtenido en nuestro caso es la siguiente:

$$R = \begin{pmatrix} 0,9995 & -0,0064 & 0,0312 \\ 0,0086 & 0,9975 & -0,0707 \\ -0,0307 & 0,0709 & 0,9970 \end{pmatrix}$$

$$t = \begin{pmatrix} 0,0204 \\ -0,0347 \\ -0,0821 \end{pmatrix}$$

Finalmente, para ejemplificar como funciona este método de calibración, en la Figura 2.8 se muestran dos ejemplos de la visualización de la nube de puntos obtenida mediante la cámara de profundidad a la que se ha superpuesto el color obtenido por la cámara fisheye en cada uno de los diferentes píxeles de la imagen. Se aprecia como el color encaja correctamente con la nube de puntos, por lo que de una forma cualitativa se comprueba el correcto funcionamiento de la calibración entre ambas cámaras.

Capítulo 3

Detección de suelo mediante profundidad

El primer paso para cualquier sistema de navegación es saber diferenciar en una escena qué es y qué no es obstáculo. Esta tarea es muy complicada, y los resultados pueden ser poco robustos si sólo se utiliza procesamiento de imagen capturada con una cámara convencional sobre una escena de la que a priori no se tiene información del entorno. Por este motivo se va a aprovechar la información que proporciona el sensor de profundidad de la cámara. El primer paso es por tanto segmentar el suelo del resto de la escena para poder avanzar por las zonas libres de obstáculos.

En este capítulo se va a explicar el procesamiento llevado a cabo para la obtención del plano del suelo mediante la información de profundidad. El proceso mostrado en este capítulo es el resultado de numerosas pruebas realizadas para encontrar el método más adecuado teniendo siempre presente el compromiso existente entre coste computacional y precisión de la segmentación.

3.1. Representación de la información 3D

Con la aparición de las cámaras RGB-D, ha sido necesario crear procedimientos para representar la información. En el caso de una imagen 2D, la forma más rápida y común de representarla es a partir de una matriz donde cada elemento de la misma, corresponde con un píxel de la imagen que contiene la información de los 3 canales de color o del nivel de gris dependiendo de si la imagen es en color o en blanco y negro. Para el caso de información de profundidad (en inglés *depth*) se puede trabajar con nubes de puntos o con imágenes de profundidad.

Una nube de puntos representa las coordenadas 3D de cada punto de la escena respecto al sensor infrarrojo. La otra opción es crear un imagen de profundidad a partir de la cual se pueden obtener las coordenadas 3D de cada píxel teniendo en cuenta que los ejes X e Y forman el plano imagen y que todos los píxels tendrán coordenada Z positiva puesto que están por delante de la cámara:

$$X = (i - c_i) \frac{depth[i, j]}{f} \quad (3.1)$$

$$Y = (j - c_j) \frac{depth[i, j]}{f} \quad (3.2)$$

$$Z = depth[i, j] \quad (3.3)$$

Siendo i y j las coordenadas 2D del píxel del que se quiere calcular su posición 3D, c_i y c_j las coordenadas del centro de la imagen de profundidad (el dispositivo que se ha utilizado en este proyecto tiene una resolución de 640×480 píxeles por tanto, para este caso en particular, $c_i = 320$ y $c_j = 240$) y f la distancia focal de la cámara.

Adicionalmente, las cámaras RGB-D pueden obtener una nube de puntos con color, en la que un cuarto número (adicional a X, Y, Z) codifica el color de cada punto mediante un número entero de 32 bits.

3.2. Filtrado de las nubes de puntos

El primer paso es decidir si realmente es necesaria toda la información que es proporcionada por el sensor de profundidad. El sistema que se presenta en este proyecto debe trabajar lo suficientemente rápido para que pueda ser de utilidad para los usuarios por lo que hay que llegar a un compromiso entre la precisión y rapidez del sistema. Cuanto mayor sea el volumen de información a procesar, se podrá conseguir una segmentación de la escena con mucho más detalle pero también aumentará el coste computacional considerablemente.

Un punto cualquiera que esté contenido en una nube de puntos tendrá información muy similar, por no decir idéntica, a la de los puntos que tenga a su alrededor. Por tanto, se va a llevar a cabo un filtrado (*downsampling*) de la nube de puntos con el objetivo de reducir el número de puntos y por tanto el volumen de información a procesar sin perder detalles relevantes de la escena. Concretamente se va a utilizar un filtro Voxelgrid [18].

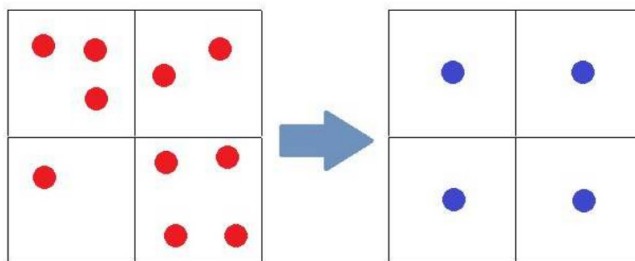


Figura 3.1: Principio de funcionamiento del filtro Voxelgrid. Todos los puntos que estén contenidos en un mismo vóxel, pasan a formar un único punto situado en el centroide de dicho vóxel

La Figura 3.1 muestra el principio de funcionamiento del filtro Voxelgrid. Este filtro divide el espacio en cubos (vóxeles) de dimensiones elegidas por el usuario. Todos los puntos que estén contenidos en cada uno de los vóxeles pasan a formar un único punto que se coloca en el centroide del propio vóxel. Existe un compromiso entre precisión y velocidad del algoritmo, a través del tamaño del voxel escogido (en nuestro caso de 4 cm). Este algoritmo ayuda a reducir el ruido y a suavizar las superficies. Con este tipo de filtro la reducción de puntos que se consigue es muy elevada ya que se pasa de tener 300000 puntos a un número inferior a 10000 sin perder detalle de las características de la escena.

3.3. Posición de la cámara

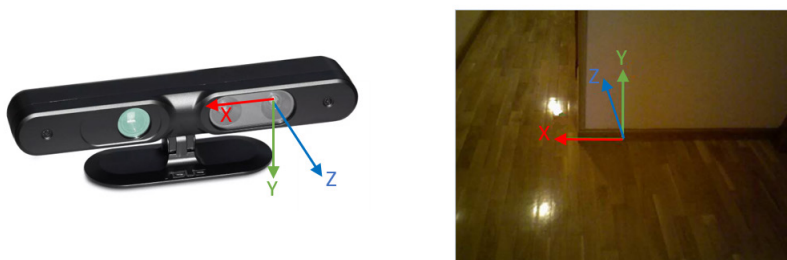


Figura 3.2: Convenio de signos para el sensor de profundidad y para una escena genérica del mundo

La cámara irá colgada del cuello del individuo y apoyada sobre el pecho del mismo tal y como se muestra en la Figura 3.3, con una inclinación de unos -45° respecto al eje X de la cámara de profundidad (cuyo sistema de referencia, así como el de una escena genérica del mundo, se muestra en la Figura 3.2). Esta configuración permite considerar pequeñas las posibles variaciones de orientación respecto a los ejes Z e Y del

sensor de profundidad. Los grados de libertad relacionados con el cambio de posición no son relevantes porque se ha tomado como referencia el cuerpo humano en su conjunto y no una zona determinada. Que la cámara se desplace hacia un lado o hacia el otro no debe afectar ya que todo el cuerpo del individuo debe esquivar el obstáculo y no una parte de él.



Figura 3.3: Posicionamiento del sistema de cámaras sobre el sujeto

3.4. Detección y extracción del plano del suelo

Para llevar a cabo la detección y extracción del plano de suelo de la escena, hay que tener en cuenta qué información se tiene y qué se necesita para formar un plano. Como ya se ha comentado antes, la información de entrada es la nube de puntos filtrada con unos 10000 puntos aproximadamente y para formar un plano se necesitan únicamente 3 puntos. El algoritmo que se ha utilizado para llevar a cabo esta tarea es Random Sample Consensus (RANSAC) [12]. RANSAC es un modelo iterativo que estima, dado un conjunto de datos, un modelo matemático deseado. En concreto, el modelo matemático buscado es un plano cualquiera de la escena, y el conjunto de datos es la nube de puntos filtrada.

El modo de funcionamiento de este algoritmo es sencillo. Para cada conjunto de datos, se le aplica los siguientes pasos:

1. Del total de puntos de la nube de puntos, se seleccionan aleatoriamente el mínimo número de puntos necesario para satisfacer el modelo matemático. En nuestro caso, por ser un plano, se seleccionan 3 puntos con los que formarlo.

2. A continuación se mide la distancia entre el plano generado y el resto de los puntos pertenecientes al grupo. Los puntos cuya distancia sea menor a un umbral votan por este plano. Este proceso se itera un número de veces determinado y al terminar se selecciona el plano con mayor número de votos siendo este la solución buscada.

El número de iteraciones ha sido elegido teniendo en cuenta la probabilidad de que al menos uno de los conjuntos de 3 puntos elegidos aleatoriamente no contenga ningún dato espurio:

$$P = 1 - (1 - (1 - \epsilon)^p)^m \quad (3.4)$$

Despejando:

$$m = \frac{\log(1 - P)}{\log(1 - (1 - \epsilon)^p)} \quad (3.5)$$

Siendo P la posibilidad de no coger un dato espurio, p el número de datos que se cogen aleatoriamente (para un plano $p = 3$), m el número de iteraciones y ϵ el porcentaje de datos espurios sobre el conjunto de datos total. Suponiendo un porcentaje de datos espurios del 60 % (cifra que nunca se alcanza debido a que en la escena siempre hay un plano dominante que ocupa gran parte de la misma, debido a la posición de las cámaras que se ha visto en el apartado anterior (Figura 3.3)), para obtener un 99 % de probabilidades de coger aleatoriamente datos no espurios, sería necesario realizar como mínimo 70 iteraciones. Para este proyecto se ha redondeado esta cifra a 100 iteraciones incrementándose así la probabilidad de escoger tres puntos correctos con el mismo porcentaje de espurios al 99,87 %.

Al aplicar el método RANSAC se obtiene por tanto un plano de la escena, que concretamente coincide con el plano de mayores dimensiones de la misma. Como se ha visto anteriormente, debido al posicionamiento de la cámara, el plano del suelo será el de mayores dimensiones, por lo que, ya se tiene la ecuación del plano del suelo (ver Figura 3.4b), de la forma:

$$AX + BY + CZ + D = 0 \quad (3.6)$$

Siendo los tres primeros parámetros (A, B, C) las coordenadas del vector ortogonal al plano y el parámetro D la distancia del plano al origen de coordenadas (en este caso corresponde con la cámara). Para incrementar la densidad de la nube de puntos de

suelo, y por tanto también su precisión, el siguiente paso será comprobar que puntos de la nube sin filtrar (la que tiene 300000 puntos), cumplen la ecuación del plano obtenida anteriormente. Para ello se fija un umbral de distancia al plano de 2 cm. Los puntos que cuya distancia al plano estén por debajo de ese umbral, pertenecerán al plano, el resto, serán considerados *no suelo*, y por tanto obstáculos.

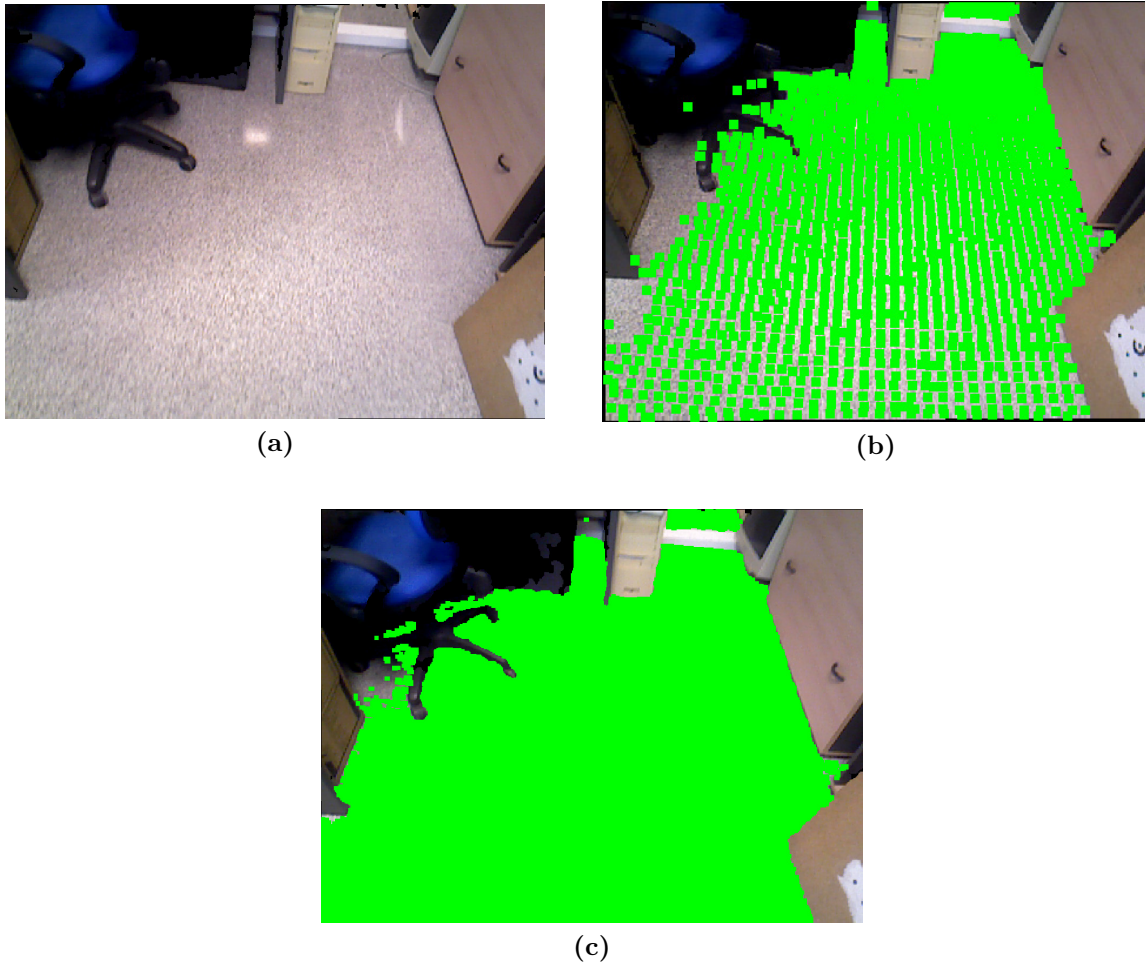


Figura 3.4: Ejemplo de segmentación del plano del suelo mediante el sensor de profundidad: (a) Imagen original. (b) Detección del plano del suelo mediante la nube filtrada. (c) Segmentación del plano del suelo final, utilizando la nube completa

La posición y orientación de la cámara, hace que siempre se visualice suelo en las imágenes, ya que la cámara apunta directamente a él. Se han fijado unos umbrales tanto de orientación sobre el eje X de la cámara, como sobre la componente D del plano detectado que debe estar entre una distancia mínima de 1 metro y máxima de 1,6 metros para ser considerado suelo, además de un número mínimo de puntos que debe tener el plano.

Antes de llevar a cabo la elección de este método para la extracción del plano del suelo, se han implementado otros algoritmos que utilizaban las normales de cada

punto con el fin de tener en cuenta la orientación de estas para extraer el plano, pero que tras la evaluación de sus tiempos de cómputo, se llegó a la conclusión de que no aportaban una clara mejora en cuanto a precisión en contraposición al coste computacional que acarreaban. Sin embargo, con el método propuesto anteriormente, y dado que se tiene información a priori de la posición y orientación de la cámara respecto al suelo, se puede acelerar el proceso de extracción del plano del suelo obteniendo resultados similares a otros métodos más complejos. En la Figura 3.4 se puede apreciar en verde la segmentación del suelo superpuesta a la nube de puntos de color obtenida mediante la cámara RGB-D.

Para la realización de esta segmentación se busca tener gran precisión, ya que este resultado servirá de base para realizar la expansión del suelo que se verá en el Capítulo 4. Como se puede apreciar en la Figura 3.4b la detección del plano del suelo a partir de la nube de puntos filtrada, proporciona menos información debido a su baja densidad. Por tanto, el uso de la nube completa (Figura 3.4c) una vez se obtiene la ecuación del plano del suelo utilizando la nube filtrada, proporciona una gran densidad, y por tanto una mayor precisión a la segmentación.

Capítulo 4

Expansión de suelo

En el Capítulo 3 se ha explicado cómo se lleva a cabo el procesamiento de la información de rango para conseguir una clasificación estructural de la escena. Trabajar únicamente con la información de profundidad sería insuficiente ya que, como se ha podido comprobar en capítulos anteriores, el sensor de rango no proporciona información 3D de la escena completa salvo que todos los puntos de ésta estén situados como máximo a una distancia determinada de la cámara. Esto se debe principalmente a que el dispositivo RGB-D tiene limitaciones de hardware. Según datos proporcionados por el fabricante, el sensor de rango tiene un alcance de 3,5 metros. Esta distancia puede resultar suficiente para la evitación de obstáculos. Sin embargo resulta insuficiente para casos donde se quiera saber información de zonas completas de la escena o se quiera realizar una planificación de movimientos con suficiente antelación para la navegación asistida del usuario, como es nuestro caso.

Para un individuo con deficiencia visual, además de ser capaz de esquivar los obstáculos, resulta muy interesante y práctico, el saber en qué dirección se encuentra el espacio libre de la escena. La utilización de la información de color a través de una cámara tipo fisheye que proporciona un gran campo de vista podría ser útil para conseguir este objetivo. En este capítulo se va a presentar el algoritmo desarrollado para expandir el suelo a zonas donde el sensor de profundidad no alcanza. Para ello se aprovecha la información del plano 3D del suelo extraído con el algoritmo del Capítulo 3, se obtiene la zona de la imagen fisheye que le corresponde a dicho plano y se analizan sus características principales que se utilizan como referencia para poder extraer nuevas zonas similares de la imagen de color que correspondan con el suelo.

4.1. Conversión de la información 3D a la imagen fisheye

El primer paso para poder alcanzar el objetivo de expandir el suelo obtenido con el procesamiento 3D es obtener una referencia del mismo en el plano imagen (reproyección) para poder extraer posteriormente sus características. Para ello se va a aprovechar la información ya obtenida del algoritmo para la extracción del suelo mediante la cámara de profundidad. En concreto se calculará para cada escena la zona de la imagen fisheye a la que pertenece el plano 3D del suelo ya extraído.

Para llevar a cabo esta tarea es necesario realizar tanto la calibración intrínseca de la cámara fisheye, como la calibración extrínseca entre la cámara de profundidad y la cámara fisheye, algo que ya se hizo en el Capítulo 2. A partir de los parámetros de calibración obtenidos, se puede convertir la información del plano 3D del suelo al plano de la imagen.

$$\begin{pmatrix} X^F \\ Y^F \\ Z^F \\ 1 \end{pmatrix} = \begin{pmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} X^D \\ Y^D \\ Z^D \\ 1 \end{pmatrix} \quad (4.1)$$

Donde (X^D, Y^D, Z^D) son las coordenadas 3D de un punto de la nube de puntos, r_{ij} y t_j son respectivamente los elementos de la matriz de rotación R y las componentes del vector de traslación t obtenidas en el Capítulo 2 que relacionan la cámara infrarroja de profundidad con la cámara fisheye. Aplicando la ecuación 4.1 se transforman los puntos 3D de la referencia de la cámara de profundidad a la referencia del fisheye. Posteriormente, aplicando el modelo inverso de Scaramuzza, que corresponde a las inversas de las ecuaciones 2.1, 2.2 y 2.3, se obtiene la posición (coordenadas (x, y)) en el plano imagen de la cámara fisheye de un punto 3D. Si se quiere obtener la posición del plano 3D del suelo en el plano de la imagen, basta con recorrer la nube de puntos y calcular la posición para cada uno de los puntos que la componen.

Sin embargo, el número de puntos pertenecientes al suelo es muy grande, por lo que resulta bastante costoso computacionalmente que para cada frame se deban hacer estas transformaciones para obtener la proyección de cada punto 3D en la imagen. Tras realizar varias pruebas se vió que esto suponía un cuello de botella en el procesamiento, que ralentizaba el algoritmo. Para acelerar este proceso se ha construido una *look-up table*, que básicamente, se trata de una matriz en la que a cada punto 3D se le asocia

las coordenadas correspondientes en píxeles en la imagen. Esta información se guarda en un fichero (por lo que solo hay que realizar los cálculos de reproyección una vez), de forma que únicamente es necesario leerlo para cada imagen, y obtener los píxeles correspondientes a cada punto 3D del suelo. Haciendo la *look-up table* se consigue acelerar el tiempo empleado en la reproyección unas 60 veces, por lo que se elimina este cuello de botella en el algoritmo. En la Figura 4.1 se puede ver un ejemplo del resultado de esta reproyección.

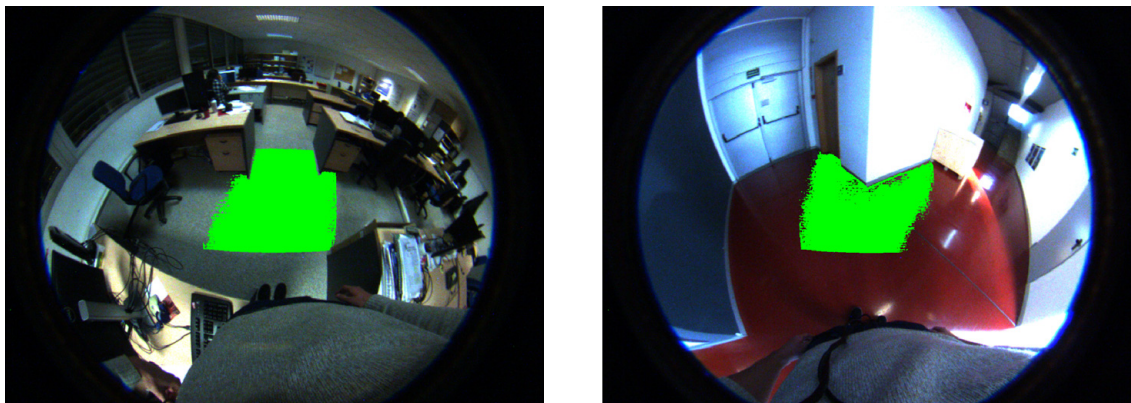


Figura 4.1: Ejemplo de la reproyección de los puntos 3D pertenecientes al plano del suelo, sobre la imagen del fisheye utilizando la *look-up table*

4.2. Modelos de color

En la sección anterior se explica cómo se calcula la localización del plano 3D del suelo en el plano imagen. Esta zona de la imagen se utilizará como referencia para poder extender el suelo a toda la zona visible de la escena. Una vez obtenida la referencia del suelo hay que analizarla para poder extraer las características más significativas de la misma y poder buscar posibles similitudes en el resto de la escena. Una característica que puede resultar a priori la más relevante es el propio color del suelo. Hay que tener en cuenta que el color de una imagen se puede representar de diferentes modos dependiendo del modelo de color que se esté utilizando. En este proyecto en concreto se utilizan los modelos de color RGB y HSV que se explican a continuación.

Existen muchas maneras de componer una imagen de color pero independientemente de cuál se utilice, el resultado final debe ser siempre el mismo ya que sino se estaría alterando la información de la escena. Los modelos de color que existen son muy variados (RGB, CIE Lab, CMYK, HSV...) pero la gran mayoría de ellos componen una imagen de color a partir de 3 canales. Lógicamente, dependiendo de cuál sea el modelo empleado, cada canal representará una información distinta de la escena.

Como ya se ha comentado antes, en este proyecto se utilizarán los modelos de color RGB y HSV. El primero de ellos es el modelo de trabajo de gran mayoría de cámaras convencionales. El segundo de ellos es muy utilizado en la segmentación de imágenes debido a la información tan característica que proporciona cada uno de los canales.

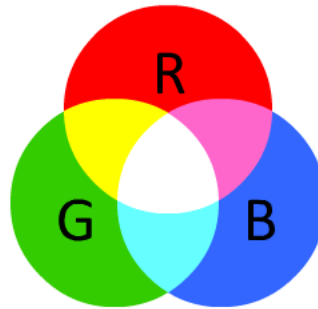


Figura 4.2: Colores aditivos del modelo RGB

Modelo RGB

Todos los colores son combinaciones variables de los tres colores primarios rojo (Red), verde (Green) y azul (Blue). De las tres iniciales de estos colores en inglés surge el nombre de este modelo de color, RGB. En el modelo RGB cada color se representa por un vector de 3 componentes, cada una correspondiente a la intensidad de cada banda fundamental (Red, Green, Blue). En la Figura 4.2 pueden verse las diferentes combinaciones de los tres colores básicos que dan lugar a otros colores.

Modelo HSV

Este modelo está basado en la percepción humana del color. Describe tres características fundamentales: tono (Hue), saturación (Saturation) y valor (Value). El tono es el color propiamente dicho, por ejemplo rojo, naranja, verde, etc. Se mide como una localización en la rueda estándar de colores (Figura 4.3) y se expresa en grados (0 – 360°). La saturación, es la cantidad de gris en el color. Se mide como un porcentaje desde 0 % a 100 % (totalmente saturado). Reducir la saturación implica obtener colores más apagados. En la rueda estándar de colores (Figura 4.3) la saturación se incrementa desde el centro hacia fuera. Por último la el valor o luminosidad, es la luz (brillo) u oscuridad del color. Se mide como un porcentaje desde 0 % (oscuro) a 100 % (brillante).

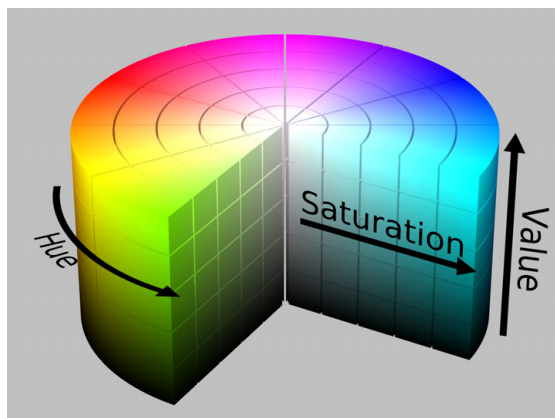


Figura 4.3: Ejemplo de representación del modelo HSV

4.3. Segmentación inicial de la escena

Una vez explicados los espacios de color que se van a utilizar en este proyecto, el siguiente paso es realizar la segmentación adecuada de la imagen de fisheye, ya que la extensión se va a realizar mediante el crecimiento de lo que se considera suelo mediante adición de diferentes regiones de la imagen, previamente segmentada. El crecimiento de suelo mediante la adición de píxeles individuales sería excesivamente costoso computacionalmente, y por tanto, inviable. Así pues, se busca dividir la imagen en regiones más grandes para posteriormente clasificarlas. Se probaron dos métodos de segmentación de color disponibles en la librería OpenCV: watershed¹ y superpíxeles². Tras varias pruebas iniciales de funcionamiento se observó el mejor funcionamiento de este último método que se explica brevemente a continuación.

El método empleado para la segmentación de la imagen de fisheye se realiza a través un algoritmo basado en la extracción de superpíxeles mediante muestreo impulsado por energía (SEEDS) [23] disponible en la librería OpenCV. Este algoritmo se basa en una optimización mediante un algoritmo de escalada (*hill-climbing*). Se parte de una partición inicial en superpíxeles de forma regular y elegida por el usuario. A partir de ahí, los superpíxeles van modificando sus fronteras para afinar cada vez más la partición siguiendo una función de energía basada en imponer la similitud del color entre las fronteras y el histograma de color del superpíxel. Es decir, la función de energía busca que los superpíxeles sean del mismo color. Según la bibliografía, este método se distingue frente a otros en su eficiencia computacional, algo que se ha tenido muy en cuenta a la hora de elegirlo en este proyecto. En la Figura 4.4 se puede apreciar un ejemplo de como trabaja este algoritmo.

¹https://docs.opencv.org/3.3.1/d3/db4/tutorial_py_watershed.html

²<https://docs.opencv.org/3.0-beta/modules/ximgproc/doc/superpixels.html>

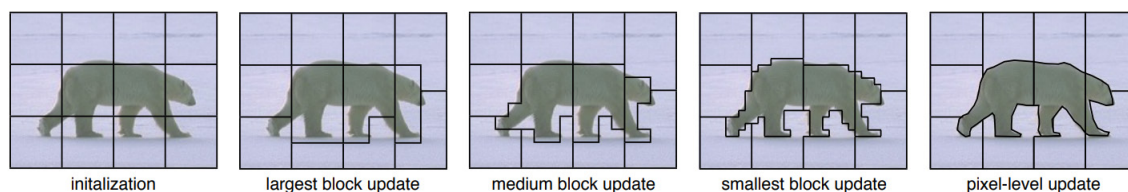


Figura 4.4: Ejemplo de la evolución de las fronteras de los superpíxeles a lo largo de diferentes iteraciones del algoritmo SEEDS partiendo de una inicialización en 12 superpíxeles iguales. Imagen obtenida de [23]

El algoritmo permite seleccionar además del número inicial de superpíxeles, otros aspectos como el número de niveles, es decir, el número de subdivisiones de los superpíxeles que se utilizan para la optimización. A mayor número de niveles, más exactitud tendrá la segmentación, pero mayor tiempo de cálculo se necesitará para llevarla a cabo. También permite seleccionar el grado de suavizado de las fronteras entre superpíxeles, permitiendo una mayor adaptación a los diferentes elementos de la escena, así como el número máximo de iteraciones del algoritmo, entre otros parámetros.

Esta segmentación en superpíxeles se ha realizado en el espacio de color HSV, con el que se obtiene una mejor segmentación de la imagen, y posteriormente se ha devuelto al espacio RGB para su visualización, como se puede ver en la Figura 4.5.

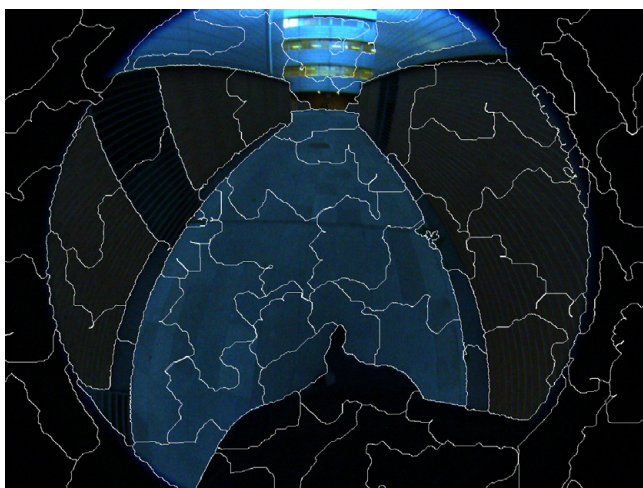


Figura 4.5: Ejemplo de segmentación de una imagen de fisheye mediante SEEDS

4.4. Expansión de suelo mediante color

Dado que se va a segmentar el suelo a partir de una región de referencia, el algoritmo trabajará de forma similar a la de un algoritmo SRG (*Seeded Region Growing*) donde la semilla inicial se creará a partir del plano 3D del suelo obtenido por la cámara de

profundidad una vez re proyectado sobre el plano imagen del fisheye, como ya se ha visto anteriormente. Una vez definidas estas semillas iniciales, se expandirán hacia zonas de la imagen similares y contiguas a las iniciales. Estas nuevas zonas pasarán a ser semillas en la siguiente iteración. Y así sucesivamente hasta que se haya segmentado el suelo de la escena.

4.4.1. Creación de la semilla inicial

El primer paso es seleccionar los superpíxeles que van a ser la semilla inicial a la hora de expandir el suelo. Como se ha comentado anteriormente, se parte de la región de suelo extraída mediante la cámara de profundidad (a la que llamaremos *suelo base*, ver Figura 4.7a). Una vez ha sido re proyectada sobre el plano imagen de la cámara fisheye, se realiza el histograma de esa región de la imagen (al que llamaremos *histograma base*). El espacio de color utilizado vuelve a ser el HSV, y el histograma se realiza sobre los canales H y S, ya que no se pretende distinguir entre el nivel de brillo de un determinado color. Esto supone una ayuda al realizar la segmentación, cuando se tienen reflejos, zonas de sombra y cambios de iluminación, por lo que se evita utilizar el canal V. El siguiente paso es almacenar los superpíxeles candidatos a formar parte de la semilla inicial, que serán aquellos que contengan algún píxel del suelo base. Una vez obtenidos los superpíxeles candidatos, se realizan los histogramas H-S de cada uno de ellos con el fin de poder compararlos uno a uno con el histograma base mediante la métrica de correlación descrita a continuación:

$$d(H_b, H_i) = \frac{\sum_I (H_b(I) - \overline{H_b})(H_i(I) - \overline{H_i})}{\sqrt{\sum_I (H_b(I) - \overline{H_b})^2 \sum_I (H_i(I) - \overline{H_i})^2}} \quad (4.2)$$

donde

$$\overline{H_k} = \frac{1}{N} \sum_J H_k(J)$$

Siendo H_b el histograma H-S del suelo base (el obtenido mediante la cámara de profundidad), H_i el histograma de cada uno de los superpíxeles candidatos a ser semilla inicial del algoritmo, $\overline{H_k}$ la media del histograma y J e I cada uno de los N *bins* de los histogramas. El resultado de esta comparación será un valor numérico donde $d \in [-1, 1]$. Cuando se obtenga $d = -1$, la imagen a comparar será justamente opuesta a la referencia. En el caso de obtener $d = 1$, la imagen a comparar será idéntica a la

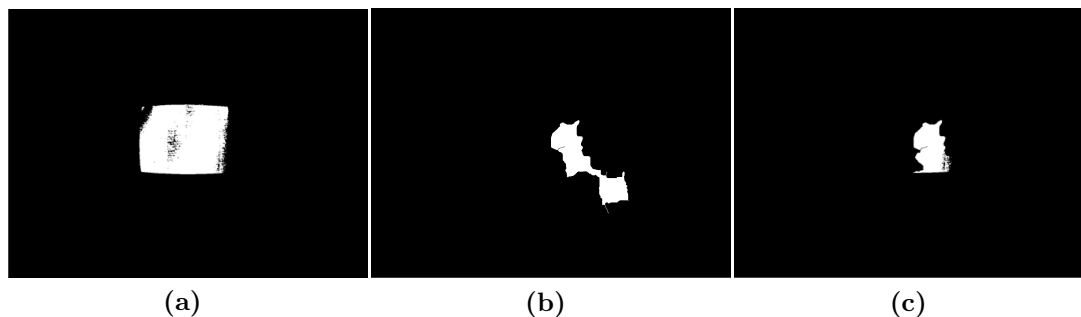


Figura 4.6: Ejemplo de las máscaras binarias utilizadas. (a) Máscara del suelo base. (b) Máscara de uno de los superpíxeles candidatos a ser semilla inicial. (c) Máscara restante tras solapadas las dos anteriores mediante, siendo esta máscara el área del superpíxel ocupada por el suelo base

imagen referencia. Por tanto, siempre que se obtengan valores positivos, la imagen a comparar tendrá cierta similitud con la referencia.

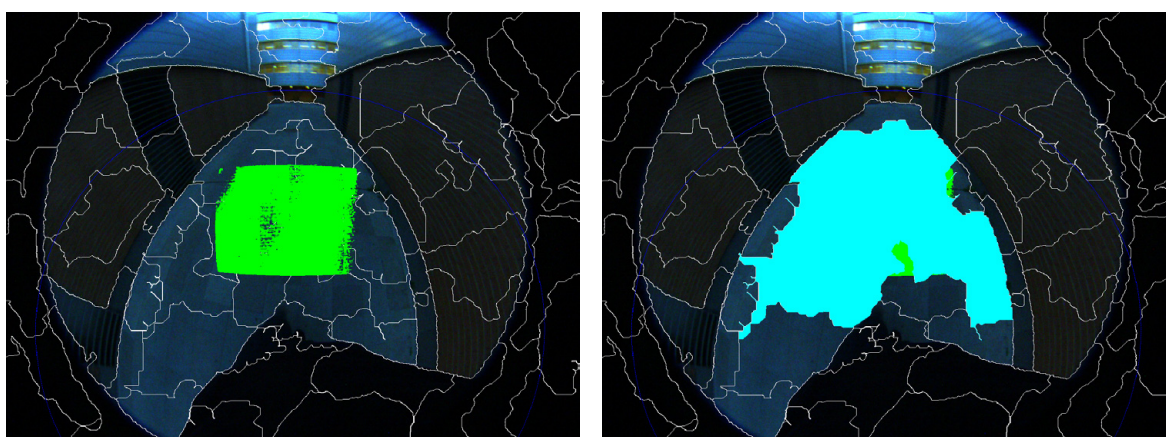
Para la creación de la semilla inicial también será necesario saber el porcentaje de área de un superpíxel ocupada por el suelo obtenido mediante profundidad. Si hay un mínimo valor de área ocupada, ese superpíxel entrará a formar parte de la semilla inicial. Para obtener ese porcentaje de área, se ha recurrido al uso de máscaras binarias, que son de gran utilidad para este tipo de procesamiento. A los píxeles considerados suelo base, se les asignará el valor '1', y al resto '0' en una imagen binaria de la misma dimensión que la imagen de fisheye (1280×960 píxeles), ver Figura 4.6a. Para cada superpíxel se creará una imagen binaria de igual tamaño que la anterior, en la que a los píxeles pertenecientes al superpíxel i se les asignará un '1' y al resto '0' (Figura 4.6b). Las imágenes binarias de los superpíxeles candidatos se solaparán contra la imagen binaria del suelo base, con el fin de conocer el porcentaje de área que ocupa este último en cada uno de los superpíxeles (Figura 4.6c). Si ese porcentaje es mayor a un determinado umbral, el superpíxel pasará a ser semilla inicial del suelo.

Por último, y con el fin de evitar que los posibles reflejos que aparecen en el suelo, puedan perjudicar la segmentación del mismo, distorsionando en exceso los histogramas de color, se va a tener en cuenta también que si en un superpíxel hay un número de puntos de suelo base lo suficientemente grande, pese a que la correlación entre histogramas no sea suficiente, ese superpíxel se considerará igualmente como parte de la semilla inicial.

Así pues, el criterio utilizado para considerar un superpíxel como semilla inicial del algoritmo queda resumido de la siguiente manera:

1. Si hay suficiente similitud entre el histograma base y el del superpíxel y además el porcentaje de área del superpíxel ocupado por el suelo base, es superior a un umbral mínimo, ese superpíxel será considerado como semilla inicial.
2. Si existe en un superpíxel un número de puntos del suelo base que es superior a un mínimo aceptable, ese superpíxel también será considerado como semilla inicial del algoritmo.

Por tanto, si se cumple cualquiera de las dos condiciones anteriores, un superpíxel será considerado como semilla inicial.



(a) Suelo detectado por el sensor de rango y (b) Semilla inicial para la expansión del suelo (en cian)

Figura 4.7: Ejemplo de creación de la semilla inicial del algoritmo (b) a partir del plano del suelo obtenido mediante la cámara de profundidad una vez se ha re proyectado sobre la imagen de fisheye (b)

Como podemos ver en la Figura 4.7, existen 2 superpíxeles candidatos a ser semilla inicial que, pese a tener una similitud bastante grande en cuanto a color con el suelo base, el porcentaje de área ocupado por éste no es lo suficientemente grande, ni tampoco lo es el número de puntos del suelo base presentes en esos superpíxeles, por lo que no se tienen en cuenta para formar parte de la semilla inicial.

4.4.2. Expansión de la semilla inicial

Una vez se tiene la semilla, el siguiente paso será la extensión de la misma para conseguir la segmentación completa del suelo. Para ello se van a utilizar las mismas herramientas que en el apartado anterior, es decir: Histogramas de doble canal H-S y máscaras binarias.

Se busca ahora que la semilla inicial se expanda únicamente a los superpíxeles vecinos, de forma que poco a poco, se vaya propagando por toda la región de suelo de la imagen. Para ello, el primer paso es la creación de la máscara binaria de la semilla inicial (Figura 4.8).

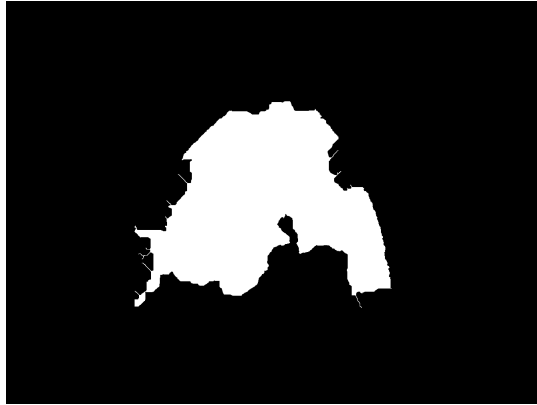


Figura 4.8: Máscara binaria de la semilla inicial de suelo obtenida en el apartado anterior

Esta máscara se va a dilatar mediante una función morfológica que consiste en realizar la convolución de la máscara de la Figura 4.8 con un kernel que tiene una determinada forma (en este caso un círculo) y con un punto de anclaje definido que es el centro del kernel. A medida que el kernel va recorriendo la imagen, se calcula el máximo valor del píxel solapado por el kernel, y se reemplaza el píxel de la imagen en la posición del punto de anclaje con ese valor máximo. Esta operación de maximización hace que las regiones blancas de la imagen “crezcan”.

Al realizar esta dilatación, los superpíxeles contiguos son “inundados” por puntos de la máscara dilatada, por lo que pasan a ser candidatos a pertenecer al suelo. A continuación, se calculan los histogramas H-S de esos superpíxeles candidatos. A partir de estos histogramas, y junto con los que ya se tienen de los superpíxeles de la semilla inicial y del suelo base, el criterio utilizado para decidir si un superpíxel candidato pasa a ser suelo es el siguiente:

1. El histograma de cada superpíxel candidato debe ser lo suficientemente parecido en color a alguno de los superpíxeles que han sido semilla inicial.
2. Además, cada superpíxel candidato debe tener un parecido razonable en color al suelo base obtenido mediante la cámara de profundidad.

Así pues deben cumplirse las dos condiciones anteriores de forma simultánea. Esto otorga robustez al sistema, ya que evita que un superpíxel de la semilla inicial que

contenga, por ejemplo, algún tipo de reflejo (y por tanto sea muy diferente a los demás) arruine la segmentación haciendo que el suelo se propague por zonas inadecuadas. Al mismo tiempo, también permite que otras zonas en las que también haya reflejos, sean admitidos como suelo al parecerse a otro superpíxel similar, ya perteneciente a la semilla inicial.

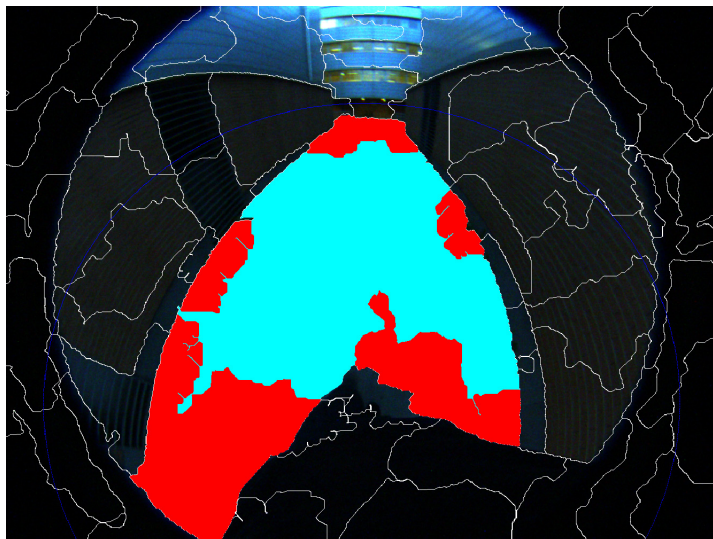


Figura 4.9: Ejemplo de la segmentación del suelo. En cian, la semilla inicial. En rojo, la expansión de la semilla hasta completar la segmentación.

Los píxeles candidatos que hayan cumplido las exigencias definidas anteriormente y se hayan seleccionado como suelo, pasarán a ser semilla en la siguiente iteración y favoreciendo así la propagación de la segmentación hasta que se dejen de cumplir las condiciones establecidas. El resultado final para el ejemplo utilizado, puede verse en la Figura 4.9.

4.5. Uso de información 3D para la expansión

Una vez se ha visto como se ha realizado el algoritmo para extender el suelo a partir del color, en este apartado se va a abordar la utilización de información obtenida a partir de el procesamiento 3D que servirá para hacer más robusto el sistema.

Puede darse el caso en el que una escena sea muy homogénea en cuanto a color, o en la que haya partes de las paredes que sean similares al suelo, o errores en la segmentación por superpíxeles. Esto haría que la segmentación del suelo se propagara por todas las partes de la imagen, haciendo que el resultado final fuera erróneo. Para evitar que esto ocurra, se va a utilizar la línea del horizonte.

La línea del horizonte en un sistema proyectivo (como es el caso), es el lugar geométrico en el cual se encuentran todos los puntos de fuga de las proyecciones de las rectas horizontales en el espacio. Es una recta sobre el plano de proyección, resultado de la intersección del plano de proyección con un plano horizontal que pasa por el origen o punto de vista del sistema proyectivo, es decir el centro óptico de la cámara. Por tanto, manteniendo la cámara con la posición y orientación definidas en el Capítulo 3, ningún punto por encima de esa línea puede ser parte del suelo, ya que la cámara se encuentra situada a la altura del pecho de la persona, y el plano en el que se encuentra la línea del horizonte es un plano paralelo al plano del suelo pero a una altura igual a la altura a la que se encuentra la cámara del suelo. De esta forma, el plano del suelo siempre quedará por debajo de esa línea.

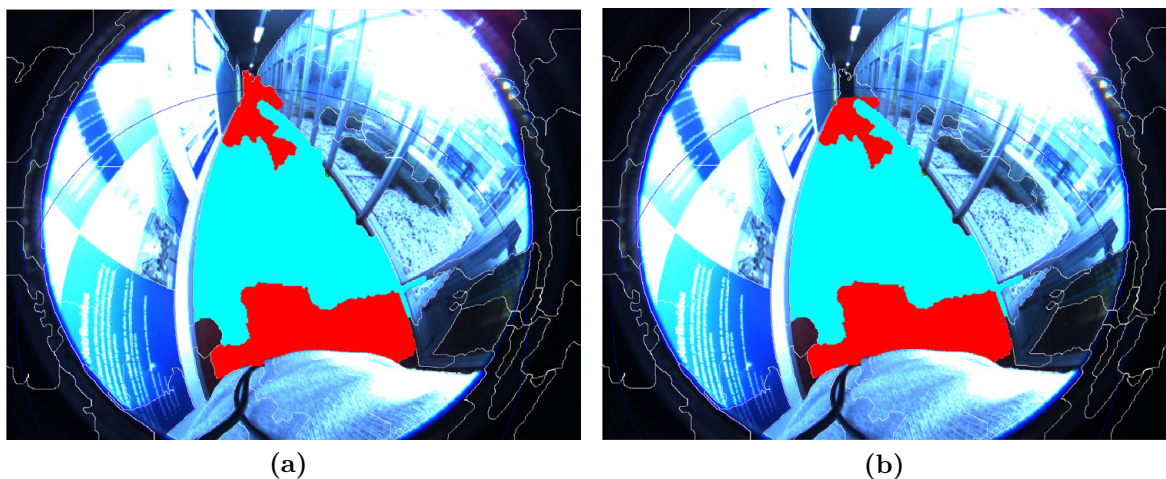


Figura 4.10: Ejemplo de la representación de la línea del horizonte (azul oscuro), y su utilidad para la segmentación: (a) En este caso, no se ha tenido en cuenta la línea del horizonte, por lo que la expansión de suelo se propaga hasta el techo empeorando claramente la segmentación del suelo. (b) En este caso, el uso de la línea del horizonte hace que pese al error en la segmentación, éste no sea muy grave evitando su propagación hacia el techo

Para calcular la línea del horizonte, se va a utilizar la información del plano del suelo obtenida mediante la cámara de profundidad, en concreto, de la ecuación del plano del suelo. Los 3 primeros coeficientes de la ecuación del plano (A, B, C) conforman la normal del plano del suelo en la referencia de la cámara de profundidad. Esta normal hay que llevarla a la referencia de la cámara fisheye utilizando la calibración extrínseca vista en el Capítulo 2. Una vez se ha pasado a la nueva referencia, esta normal es la del plano proyectivo de la cámara fisheye, por tanto, utilizando el modelo de distorsión de la cámara y el método para extracción de líneas de Bermúdez-Cameo y col. [5] explicados en el Capítulo 2 se representa la línea del horizonte en la imagen fisheye (en azul oscuro en la Figura 4.10).

Una vez obtenida, hay que asegurar que todo lo que quede por encima de esta línea

no pueda ser segmentado como suelo. Para ello se va a recurrir de nuevo a las máscaras. Se va a crear una máscara binaria (Figura 4.11) en la que todo lo que queda por encima de la línea del horizonte no se va a tener en cuenta (asignandoles a los píxeles el valor '0') para la segmentación del suelo de la imagen. Superponiendo esta máscara a la imagen al comienzo del procesamiento, solo se tienen en cuenta a la hora de expandir o de realizar la semilla inicial, los píxeles por debajo de esa línea del horizonte.

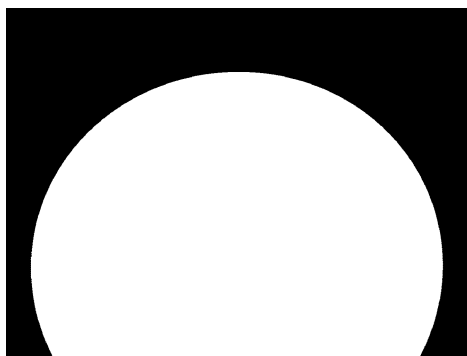


Figura 4.11: Máscara binaria para eliminar los píxeles en negro de la segmentación del suelo de la imagen. Solo se tendrá en cuenta como posible suelo, la zona en blanco

Como se puede ver en la Figura 4.10, al fondo del pasillo se ha producido un error en la segmentación por medio de los superpíxeles que hace que el suelo se propague por el cristal. También se puede apreciar como, pese a que el superpíxel llega hasta el techo, la segmentación errónea del suelo no va más allá de una pequeña porción de cristal en la Figura 4.10b gracias a lo explicado anteriormente. Sin embargo, en la Figura 4.10a en la que no se tiene en cuenta la línea del horizonte, la segmentación es claramente peor ya que el suelo se propaga hasta el techo.

La máscara de la línea del horizonte no permite continuar más allá de ella, ni siquiera llegar a tocar la propia línea, ya que, lo que realmente se ha hecho es dilatar esa línea del horizonte para aumentar la zona a partir de la cual se dejan de tener en cuenta los píxeles para la segmentación del suelo. La dilatación se ha realizado utilizando el mismo método que en la sección anterior. La razón para esta dilatación es que, debido a la naturaleza proyectiva de la cámara, si hubiera puntos del suelo que pertenezcan o que estén muy próximos a la línea del horizonte en la imagen, esos puntos de suelo en 3D, estarían en el infinito. Para evitar esto a la hora de calcular áreas de extensión de suelo y distancias (como se verá en los siguientes capítulos) se ha optado por crear esa banda para que no se desvirtúen las medidas en 3D. Gracias al uso de la línea del horizonte se ha conseguido que pese a haber un error en la segmentación del suelo, ese error no afecte en gran medida a la detección del espacio libre, y a la planificación de posibles trayectorias, que son los objetivos principales de este proyecto.

Capítulo 5

Propuesta para la navegación

Una vez realizada la segmentación del suelo, se va a mostrar nuestra propuesta para la navegación del usuario. Se ha buscado dotar al usuario de una información lo más sencilla e interpretable posible para que éste sea capaz de interpretar con la mayor facilidad posible las instrucciones dadas por el sistema.

Al tratarse de un sistema de detección del espacio libre para permitir planificar trayectorias a corto y medio plazo, no resulta tan crítico transmitir las instrucciones al usuario en tiempo real, ya que el objetivo es aportar información complementaria para la navegación. Este sistema va más allá de un sistema de evitación de obstáculos ya que da información más útil para la navegación debido al gran campo de vista que se consigue. No obstante, la evitación de obstáculos es algo crítico en cualquier sistema de guiado a personas, pero ya ha sido ampliamente tratado en otros trabajos, y en nuestro caso, se puede realizar únicamente con la cámara de profundidad, como se propone en [1].

Las personas con deficiencias en la vista consiguen desarrollar de forma notoria algunos de sus otros sentidos pudiendo así suplir algunas de las funciones que tiene el sentido de la vista. Concretamente los sentidos mejor desarrollados son el del oído y el del tacto. La interfaz del sistema podría aprovechar por tanto estos dos sentidos del usuario para darle instrucciones o informarle del entorno.

El dispositivo más sencillo a utilizar sería un sistema de audio a través del cual se guiaría al usuario a través de instrucciones que le informaran del espacio libre en la escena, tanto de la distancia hasta la que se ha detectado suelo por el que poder desplazarse, como la dirección en la que se encuentra. Adicionalmente, gracias al gran campo de vista disponible, se podrían detectar posibles puertas por las que poder seguir

avanzando.

5.1. Algoritmo para navegación

El primer paso para poder conocer datos de suelo extendido (distancias, áreas extendidas, etc.), es realizar la reproyección de la segmentación en 3D. Para ello, se obtiene el rayo proyectante para cada punto del suelo sobre la imagen, utilizando el modelo proyectivo de la cámara fisheye visto en el Capítulo 2.

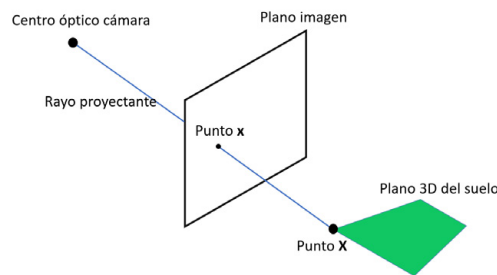


Figura 5.1: Ejemplo de la intersección entre el rayo proyectante y el plano del suelo

En geometría descriptiva, el rayo proyectante es una línea imaginaria que parte desde el centro óptico de la cámara y que tras pasar por un punto 3D (punto \mathbf{X}), proporcionan la proyección de dicho punto 3D en el plano imagen (punto \mathbf{x}). Así pues, el rayo proyectante depende tanto del punto 3D como de la posición de la cámara.

Una vez se obtiene el rayo proyectante para cada punto del suelo de la imagen, y para conocer la posición del punto correspondiente en 3D, hay que tener en cuenta que ya se conoce el plano en 3D sobre el que van a estar todos los puntos del suelo (que se ha obtenido mediante la cámara de profundidad), y se ha transformado a la referencia del fisheye (utilizando la calibración extrínseca entre ambas). Por tanto, únicamente hay que realizar la intersección de cada uno de los rayos proyectantes con el plano del suelo 3D.

Una vez se tienen el plano del suelo en 3D, el siguiente paso es cambiar el sistema de referencia del mismo. Este plano 3D está en la referencia de la cámara fisheye, y para realizar las medidas correspondientes hay que llevar la referencia al suelo, sobre los pies del sujeto que porta la cámara (transformación ${}^F T_S$) con el eje Z hacia delante, y X e Y sobre el plano frontal de la persona, tal y como se muestra en la Figura 5.2

Una vez se ha realizado esta transformación, las medidas que se obtengan en 3D

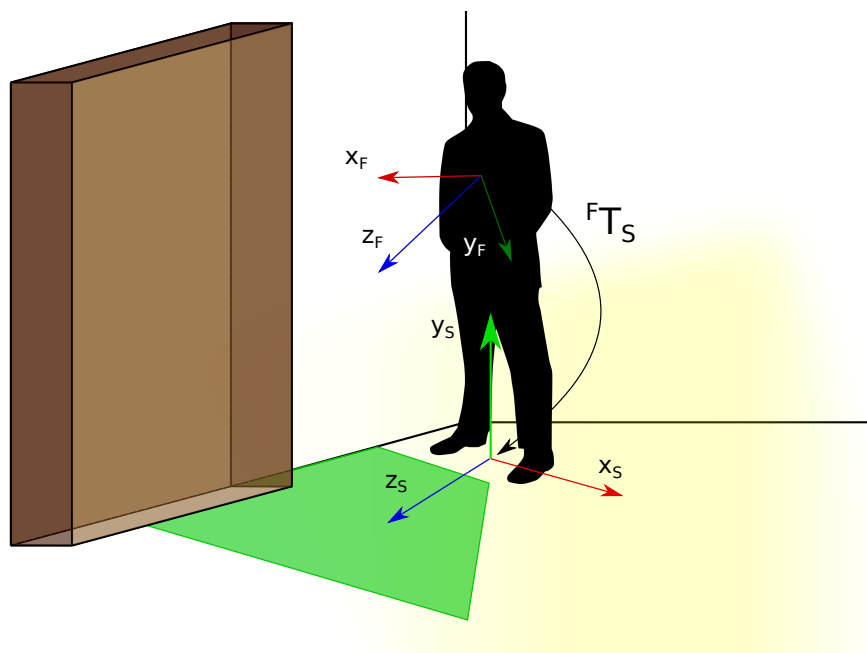


Figura 5.2: Transformación de la referencia cámara a la referencia suelo

tendrán como referencia inicial la posición de la persona por lo que serán más fácilmente interpretables por el usuario.

5.1.1. Zonas de suelo no transitables

En la expansión de suelo llevada a cabo en el capítulo anterior, se han detectado zonas que son suelo pero que no son transitables. Por ejemplo el suelo que hay bajo una mesa, no es transitable por el usuario, pero sigue siendo suelo. Pese a que para validar la expansión del suelo en sí esto no es relevante, a la hora de realizar la navegación por el entorno, al usuario hay que transmitirle únicamente información de las zonas por las que puede transitar. Para eliminar estas zonas que son vistas como suelo pero que no son transitables se va a hacer uso de nuevo de la información 3D.

De la nube de puntos completa, los puntos que cumplen la ecuación del plano del suelo serán considerados como tal (como ya se ha explicado), sin embargo el resto de puntos que no cumplen esa ecuación, y por tanto se encuentran a distinta altura, son normalmente obstáculos. Para eliminar del suelo transitable esos obstáculos, se toman los puntos 3D no segmentados como suelo y se proyectan en 3D sobre el suelo según la vertical. El resultado se proyecta sobre la imagen de fisheye, y con ellos se crea una máscara binaria tal y como se ha hecho anteriormente. Esa máscara binaria se restará a la máscara de suelo que se ha mostrado en el capítulo anterior, para eliminar ese espacio no transitable.



Figura 5.3: (a) Resultado de la segmentación del suelo de la escena. (b) Máscara del suelo segmentado. (c) Máscara del suelo segmentado eliminando los puntos de zonas no transitables

En la Figura 5.3 se puede apreciar un ejemplo de lo explicado anteriormente. La zona de la segmentación bajo la mesa y las sillas (Figura 5.3a), es eliminada, ya que esa zona no es transitable por el usuario pese a ser suelo (Figura 5.3c).

5.1.2. Discretización del espacio libre

De cara a convertir la representación geométrica del espacio libre en un catálogo sencillo de instrucciones de audio para el usuario se propone realizar una discretización del espacio 3D. Se consideran 5 posibles direcciones de izquierda a derecha y cinco rangos de distancias centrados en el usuario. Para ello se divide el espacio a partir de un origen que se encuentra situado en el suelo, a la altura de los pies del usuario. El espacio se discretiza utilizando coordenadas polares, mediante un radio y un ángulo cambiantes. Teniendo en cuenta el campo de vista de la cámara en la posición elegida, en la que se observa más de 260 grados de suelo alrededor, y se ve desde los pies del usuario a puntos muy alejados (del infinito), se escoge un espacio de 260° dividido en 10 sectores iguales de 26° cada uno. El radio de cada sector queda dividido a su vez en otros 5 subsectores, con distancias de 1, 3, 7, 15 y 31 metros. Esta discretización en el espacio 3D, se re proyecta en la imagen fisheye y se realiza una máscara para cada imagen, como la que se muestra en la Figura 5.4

El siguiente paso es decidir cuáles son las zonas por las que hay espacio libre. Para ello, se van recorriendo cada uno de los subsectores de la siguiente forma: Se comienza por el primer sector angular, y dentro de este sector, se van recorriendo los 5 subsectores radiales. Así pues, para cada subsector se va a calcular el porcentaje de área que ocupa la máscara de suelo expandido, mostrada en el Capítulo 4. Si ese área es suficiente se pasará al siguiente subsector radial. En el caso de que no lo sea, se considera que por ese sector angular no hay más espacio libre, por lo que se pasaría al siguiente sector

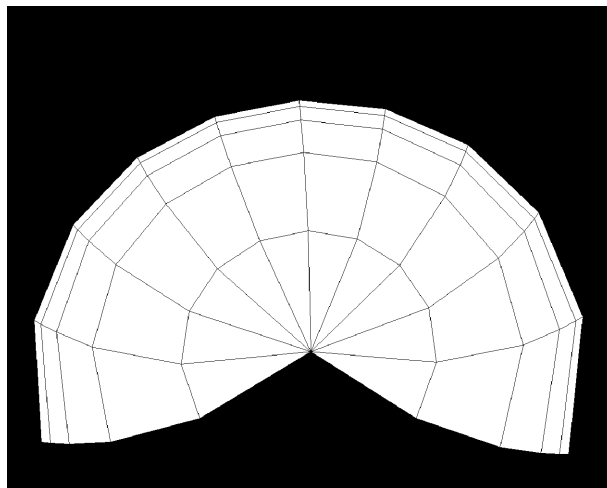


Figura 5.4: Discretización del espacio en sectores: máscara utilizada

angular, y así hasta recorrer todo el espacio. Para el caso de ejemplo utilizado en el capítulo anterior, el resultado se puede observar en la Figura 5.5

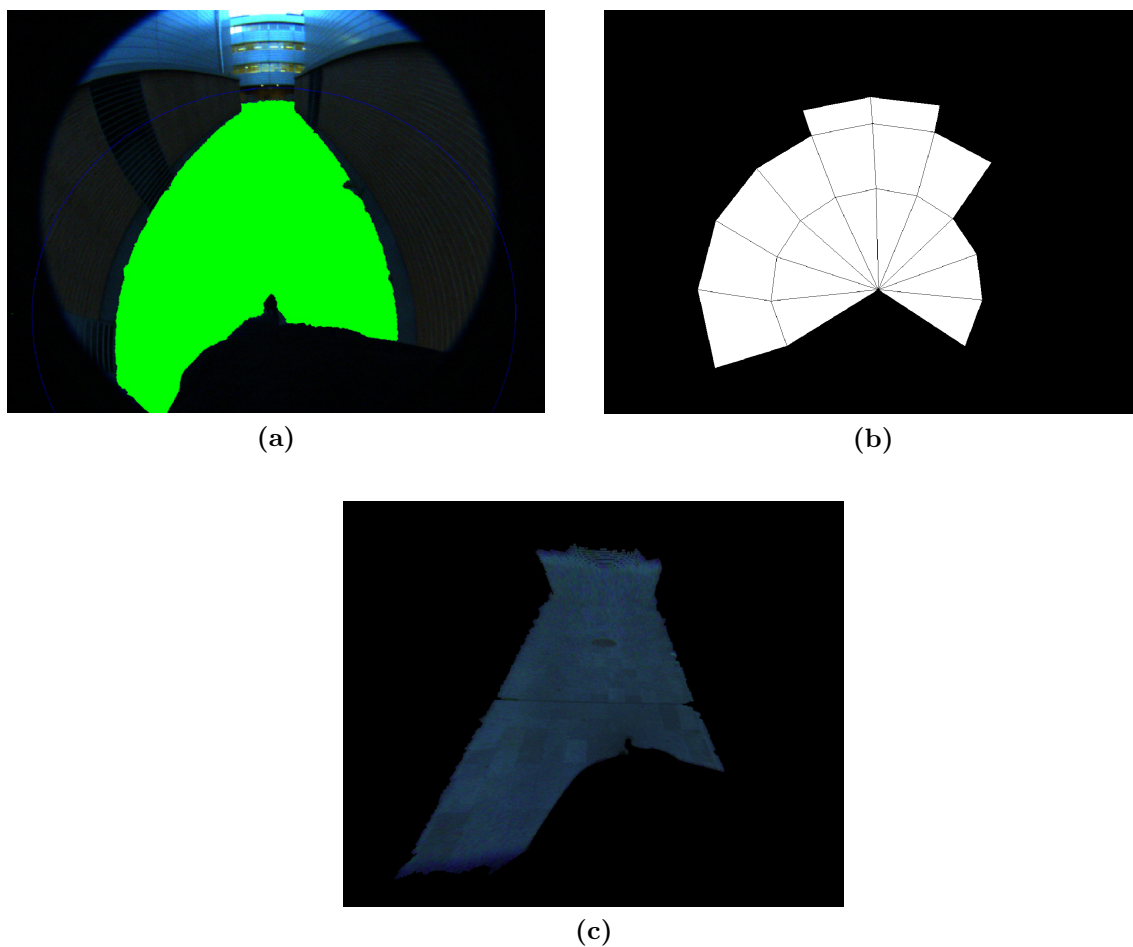


Figura 5.5: (a) Resultado de la segmentación del suelo de la escena. (b) Máscara de espacio libre de la escena en la navegación. (c) Espacio libre de la escena en 3D

Como se puede ver en la Figura 5.5b, la zona con mayor espacio libre es hacia

delante, tal y como se aprecia en la Figura 5.5a. Concretamente, la distancia de avance más grande es de al menos 7 metros (que es lo que se puede asegurar), aunque podría ser mayor. Como vemos, en otras zonas de la imagen, el espacio libre es de unos 3 metros, o de 1 metro en el caso de la zona derecha de la imagen. Esta división en sectores da una idea bastante clara de las zonas libres presentes en la escena, lo que resulta de gran ayuda para la navegación.

En el trabajo de Aladren y col. [1], se creó una interfaz de usuario mediante audio para la evitación de obstáculos. Para la planificación de trayectorias mediante la detección del espacio libre en la escena, se proponen las siguientes instrucciones de audio, que se añadirían al catálogo de [1]. En estas instrucciones se indica tanto la distancia, como la dirección por la que se debe avanzar.

Sectores angulares (de izquierda a derecha)	Instrucción (audio)
1	Espacio libre izquierda
2	Espacio libre izquierda
3	Espacio libre frente-izquierda
4	Espacio libre frente-izquierda
5	Espacio libre de frente
6	Espacio libre de frente
7	Espacio libre frente-derecha
8	Espacio libre frente-derecha
9	Espacio libre derecha
10	Espacio libre derecha

Tabla 5.1: Instrucciones de audio en función del sector angular

Distancia (metros)	Instrucción (audio)
1	Espacio libre 1
3	Espacio libre 3
7	Espacio libre 7
15	Espacio libre 15
31	Espacio libre 31

Tabla 5.2: Instrucciones de audio en función de la distancia. El número indica el mínimo espacio libre disponible en metros

En la Figura 5.6 se muestra un segundo ejemplo de como funciona el algoritmo de navegación. Como puede verse en la Figura 5.6b, el sistema indica la existencia de 7 metros de espacio libre a la izquierda, 7 metros al frente-izquierda, 7 metros al frente, 15 metros al frente-derecha y 15 a la derecha. Observando también la figura 5.6c se aprecia como la zona en la que más espacio libre hay es hacia la derecha, mientras que al frente y a la izquierda también existe espacio libre, aunque algo menos.

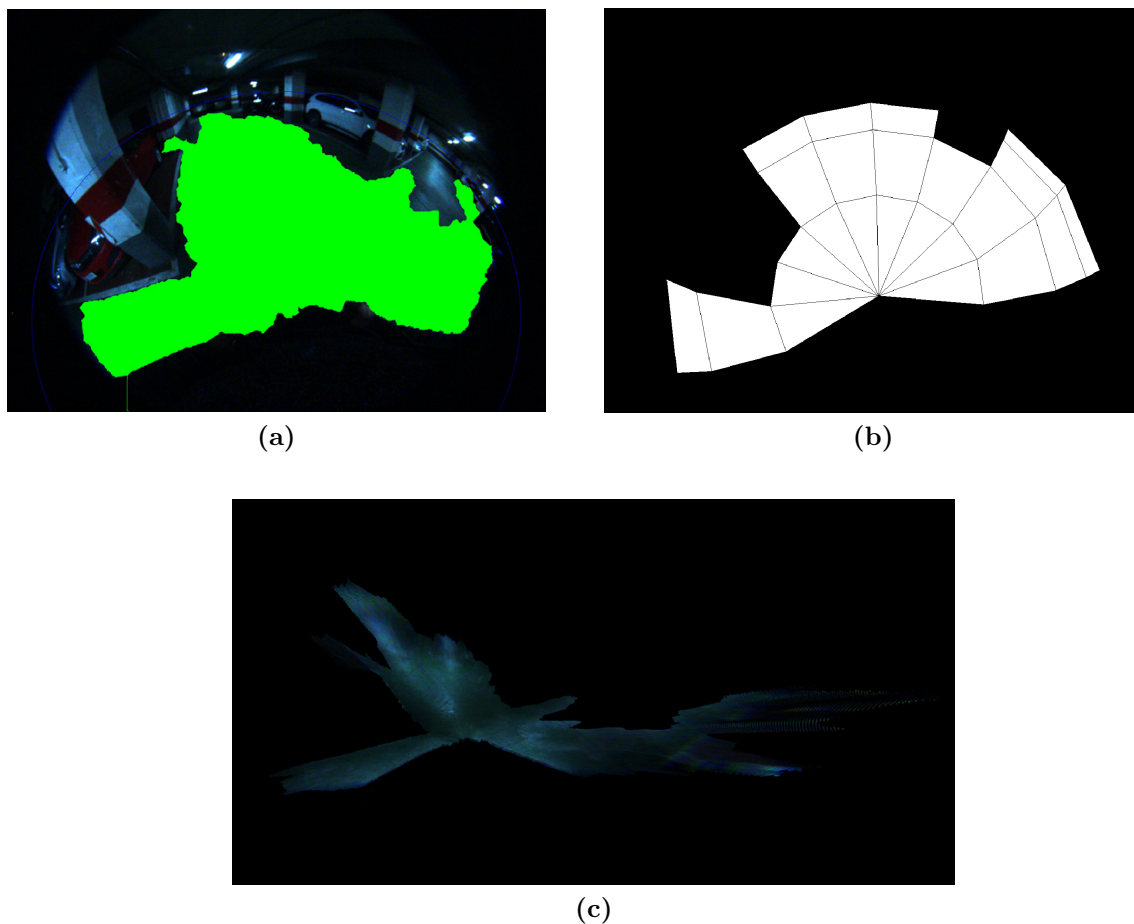


Figura 5.6: (a) Resultado de la segmentación del suelo de la escena. (b) Máscara de espacio libre de la escena en la navegación. (c) Espacio libre de la escena en 3D

5.2. Detección de puertas

Además de la planificación de trayectorias así como la detección del espacio libre, la combinación de este tipo de sensores junto con la información que ya se ha obtenido y procesado a lo largo de este proyecto, va a permitir realizar otro tipo de detecciones como es la detección y localización de puertas.

En esta sección se va a explicar a modo de prueba de concepto cómo, a partir de información obtenida anteriormente de ambos sensores, se podría detectar y localizar puertas.

En la Figura 5.7 se muestra un ejemplo de ello. Como se puede apreciar, hay tres puertas en las imágenes: una en la izquierda, otra en el centro y otra a la derecha. Se parte de la siguiente información:

1. Las líneas principales de la puerta: verticales y horizontales (en rojo).

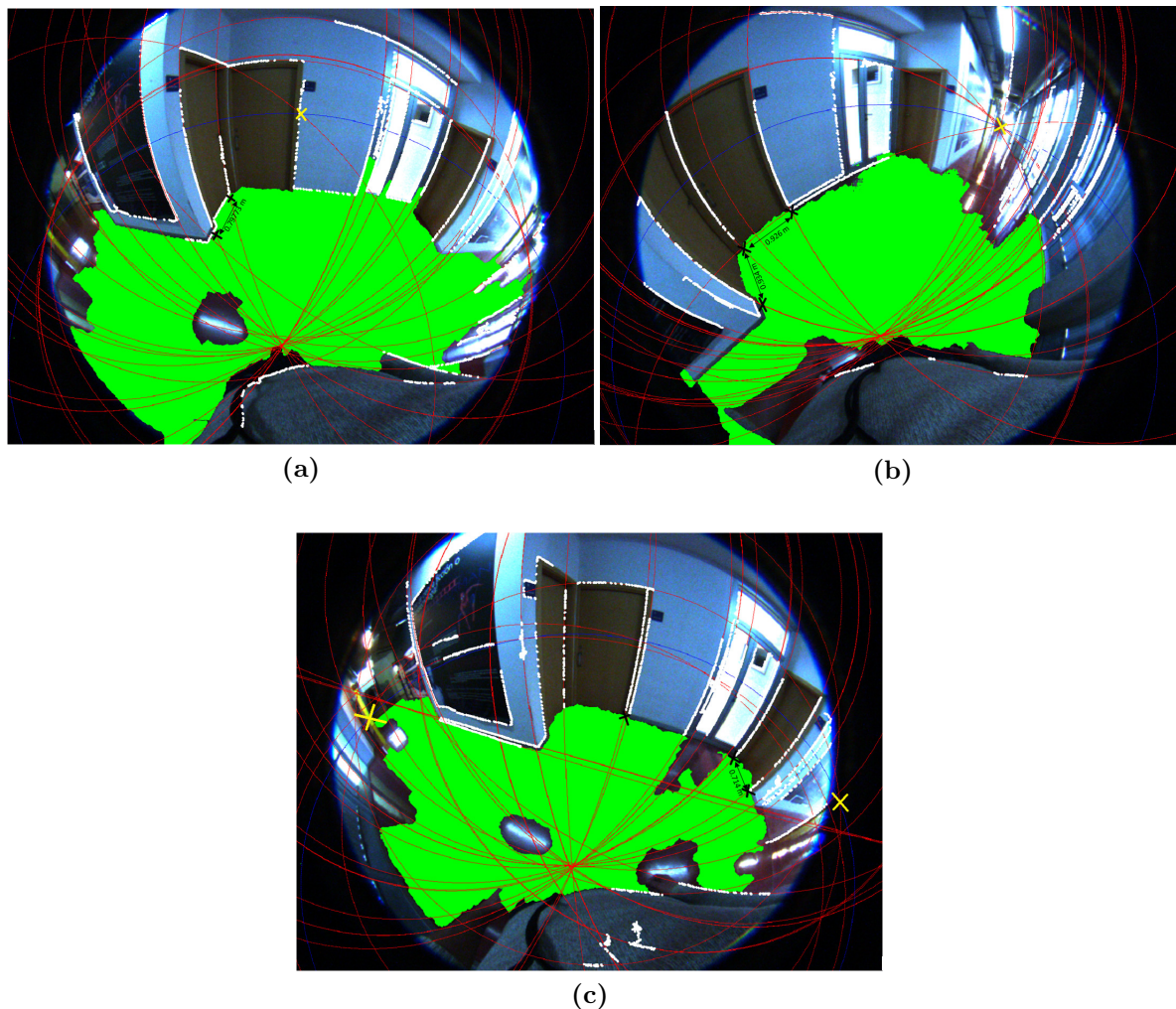


Figura 5.7: Ejemplos de posibles casos en los que establecer hipótesis de puertas

2. El plano del suelo llegando hasta el límite inferior de la puerta (en verde).
3. La línea del horizonte (en azul).

Para realizar una hipótesis de puerta, a partir de esta información, y mediante la intersección de las líneas verticales de la puerta con el plano del suelo en 3D (que también conocemos), representado en la imagen mediante cruces negras, obtendríamos las dimensiones de la puerta en 3D, por lo que se podría establecer unos umbrales mínimo y máximo de anchura como restricción para una hipótesis de puerta. Como puede verse en la Figura 5.7, esta anchura va desde los 70 cm hasta algo más de 90 cm en estos ejemplos. Esta combinación de sensores permitiría también situar la puerta en 3D sabiendo, no solo la anchura de la misma, sino la distancia a la que se encuentra y en que dirección respecto a la persona.

Además, mediante el conocimiento de la línea del horizonte y de una de las líneas

horizontales (superior o inferior) de la puerta, se podrían establecer los puntos de fuga de estas líneas horizontales (representadas mediante cruces amarillas), que estarían sobre la línea del horizonte. De esta forma, para obtener la otra línea horizontal, se partiría de esos puntos de fuga, que junto con las líneas verticales, permitiría inferir la línea horizontal restante que forma el contorno de la puerta, reforzando de esta manera la hipótesis. Como aclaración hay que comentar que en este tipo de imágenes fisheye, existen dos puntos de fuga de las líneas que se encuentran siempre sobre la línea del horizonte. Pueden verse en la Figura 5.7c, ya que en los otros dos ejemplos, uno de los puntos de fuga queda fuera de la imagen.

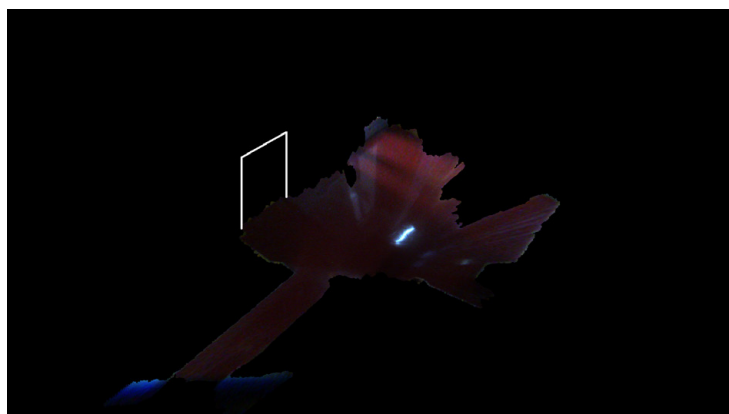


Figura 5.8: Ejemplo de localización de la puerta central de la Figura 5.7b en el espacio 3D

Para diferenciar una puerta de una ventana, o de un cuadro colgado en la pared, se podría también incluir la restricción de que la línea del horizonte debe cortar a las líneas verticales que sean contorno de las puertas aproximadamente por la mitad, ya que la cámara situada sobre el pecho de la persona está a una altura sobre el suelo aproximadamente de la mitad de la altura de la puerta.

En el caso de la puerta derecha de la Figura 5.7c pese a solo tener dos líneas verticales, se podrían inferir también las dos líneas horizontales, ya que los contornos detectados (en blanco) van desde arriba hasta abajo, por lo que con los dos puntos que tocan el suelo, se podría inferir la recta horizontal inferior que tendría sus puntos de fuga sobre la línea del horizonte. A partir de esos puntos de fuga, junto con otro punto presente en el marco superior de la puerta, se podría inferir la otra recta horizontal superior para reforzar la hipótesis de puerta. Esto se podría hacer ya que al tratarse de rectas paralelas (la horizontal superior e inferior), comparten los mismos puntos de fuga. También podría darse el caso de tener las cuatro líneas que forman el contorno de la puerta, como se puede ver en la puerta central de la Figura 5.7b. Otro ejemplo podría ser el de la puerta central de la Figura 5.7c en la que solo se tiene una línea vertical y la horizontal superior. Utilizando el punto de fuga de la línea horizontal

superior (en amarillo) presente sobre la línea del horizonte, y junto con el otro punto de intersección de la línea vertical con el plano del suelo (cruz en negro) se podría inferir la línea horizontal inferior de la puerta para reforzar la hipótesis.

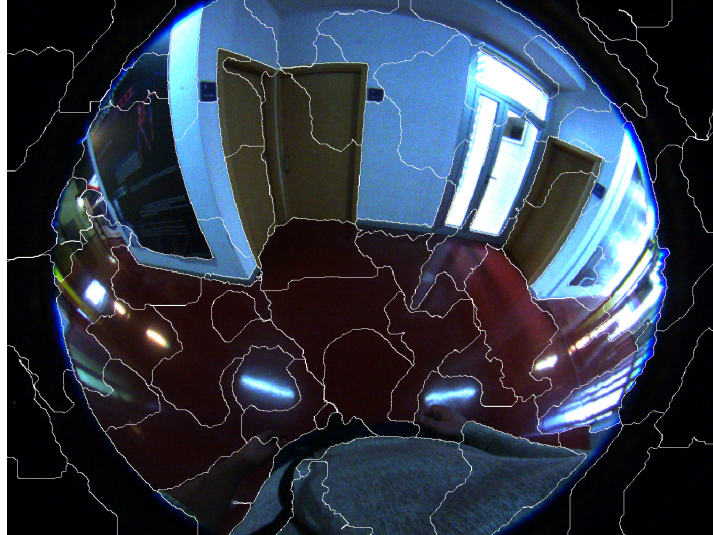


Figura 5.9: Ejemplo de la segmentación de las puertas mediante superpíxeles. Esta información puede ser utilizada para mejorar la hipótesis de puerta

Adicionalmente a todo esto, la utilización de la segmentación mediante superpíxeles utilizando el color, también podría resultar de utilidad, ya que el color de las puertas suele ser homogéneo y distinto del resto de la pared. Como se puede ver en la Figura 5.9, el contorno de las puertas está muy bien diferenciado del resto por medio de la segmentación de superpíxeles, por lo que se podría añadir este tipo de restricciones de color para mejorar la creación de hipótesis.

Capítulo 6

Evaluación experimental

Con los algoritmos de procesamiento de la información 3D y del color de la imagen explicados anteriormente, se va a evaluar experimentalmente el sistema desarrollado en diferentes escenarios. Se ha buscado que estos escenarios fueran diferentes entre sí dentro de lo posible, además de que presentaran condiciones de iluminación variadas así como brillos, sombras y reflejos que dificultaran la segmentación de la escena.

Pese a que el fabricante no recomienda utilizar la cámara de profundidad en entornos exteriores debido a la interferencia que se produce entre la luz infrarroja emitida por la cámara y la emitida por el sol, se han buscado también entornos exteriores en los que el sol no incidiera directamente, zonas de sombra en días soleados, o días nublados, con el fin de evaluar también en estas condiciones el funcionamiento del algoritmo.

6.1. Etiquetado y ground truth

La ausencia de bases de datos en los que se utilice esta combinación de sensores ha dificultado la evaluación experimental, haciendo que todas las imágenes hayan sido obtenidas por nosotros, y haciendo inviable una comparación con otros algoritmos implementados. Además, ha sido necesaria la creación de un sistema para etiquetar el ground truth que permitiera hacer evaluaciones cuantitativas del funcionamiento del algoritmo.

El trabajo de etiquetado de imágenes es costoso y tedioso, por lo que se ha realizado un programa interactivo para facilitar el proceso. En principio se pensó en marcar sobre la imagen un polígono cerrado para definir el suelo, pero al ser imágenes de gran angular los contornos son curvos en la imagen y ese proceso exigía un marcado manual muy

detallado. Por ello, y con el objetivo de que el etiquetado no fuera muy costoso se decidió aprovechar la segmentación automática mediante superpíxeles. Así pues, cada vez que se hace click con el ratón en un punto de cada uno de los superpíxeles, se incluye todo ese superpíxel como parte del ground truth. De esta forma además, se facilita la evaluación de coincidencia o no con el ground truth y el cálculo automático del área en pixels a partir de funciones disponibles en OpenCV tras la generación de máscaras binarias. Evidentemente, a mayor número de superpíxeles, mayor precisión en la segmentación y etiquetado del ground truth, pero también ese tiempo de etiquetado es mayor.

El número de imágenes de cada secuencia es elevado y a la vez la similitud entre imágenes cercanas es alta. Para cada secuencia se han seleccionado una de cada 15 imágenes para ser etiquetadas y evaluar el funcionamiento para cada escena en condiciones realistas.

Para realizar la evaluación cuantitativa se van a obtener los siguientes datos:

- Ratio de área en la imagen que supone el suelo expandido mediante color frente al detectado mediante profundidad en píxeles
- *Precision* y *recall* frente al ground truth en píxeles. Siendo

$$Precision = \frac{\text{Área de suelo expandida correctamente (TP)}}{\text{Área de suelo total expandida (TP + FP)}} \quad (6.1)$$

$$Recall = \frac{\text{Área de suelo expandida correctamente (TP)}}{\text{Área de suelo etiquetada como ground truth (TP + FN)}} \quad (6.2)$$

- Área 3D tanto del suelo detectado mediante profundidad como de la extensión del mismo (ambas en m^2)

A continuación se presentan los resultados obtenidos con el sistema propuesto en varias imágenes de ejemplo tomadas en cada uno de los entornos propuestos, así como los valores medios de los parámetros de evaluación considerados para los diferentes entornos. El resto de imágenes utilizadas junto a los resultados finales de expansión de suelo obtenidos, organizados para las diferentes secuencias utilizadas pueden verse en el Anexo B.

6.2. Entornos interiores

En primer lugar se van a mostrar diferentes entornos de interior en los que se ha evaluado el algoritmo.

6.2.1. Entorno de interior con incidencia de luz solar

En este caso, se trata de un entorno de interior en el que existe una incidencia de luz solar proveniente de la cristalera que se encuentra detrás de la cámara, apareciendo también ciertos reflejos de luz artificial.

Ejemplo interior 1

La Figura 6.1a muestra una escena en la que aparecen grandes reflejos procedentes de la ventana, especialmente en la parte izquierda de la imagen. Debido a la incidencia de la luz solar en ciertas partes de la escena, existe una gran diferencia de iluminación en esta imagen, lo que complica la detección de suelo.

En la Figura 6.1b puede apreciarse como la cámara de profundidad realiza la detección de suelo en una zona de la imagen de fisheye en la que no existen reflejos, por lo que el color en esta zona de la imagen será muy distinta al resto de la misma.

Una vez se tiene el suelo inicial, se procede a la expansión del mismo utilizando el color. Como se puede observar en la Figura 6.1d, la detección del suelo en la parte izquierda de la imagen es correcta a pesar del gran número de reflejos existentes, procedentes del exterior. En la zona de la puerta existe un pequeño error en la detección, debido a un error en la segmentación por superpíxeles, tal y como puede verse en la Figura 6.1c. Bien es cierto que ese error sería subsanable mediante la eliminación de zonas no transitables utilizando la profundidad (visto en la sección 5.1.1). Por otro lado, en la parte derecha de la imagen existe una zona cuya iluminación es muy diferente al resto de la imagen debido a la incidencia directa de la luz solar, lo que hace que esa zona no sea segmentada como suelo.

Una vez se obtiene la segmentación final del suelo, el resultado se reproyecta en 3D (Figura 6.1e), pudiéndose apreciar de una manera más clara, el espacio libre detectado en la escena, así como realizar medidas sobre la misma. En ocasiones, la naturaleza proyectiva de las imágenes, impide ver claramente la extensión de suelo realizada.

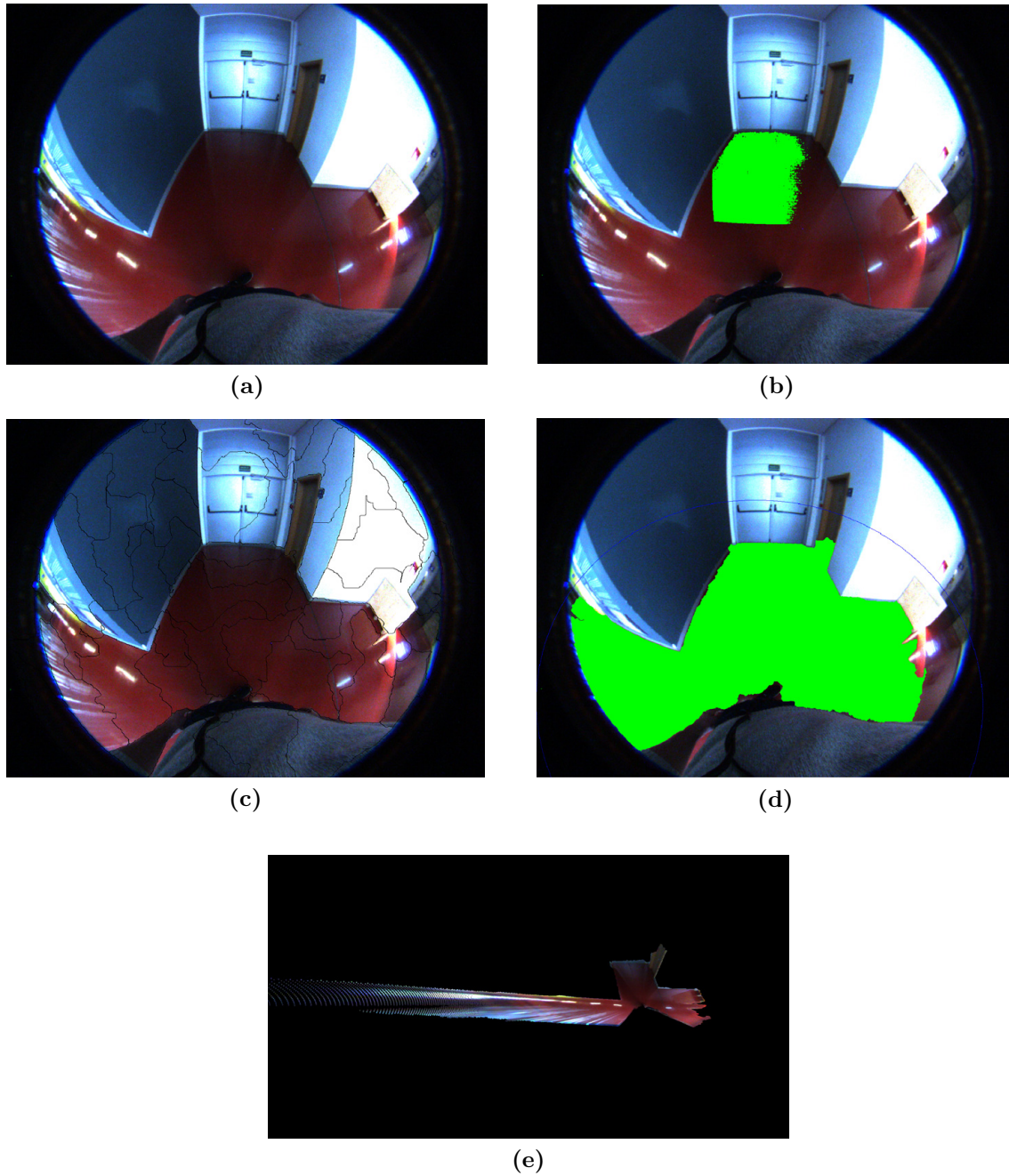


Figura 6.1: (a) Imagen de interior 1. (b) Detección de suelo mediante cámara de profundidad. (c) Segmentación mediante superpíxeles. (d) Resultado final de la detección de suelo. (e) Proyección 3D de la detección de suelo.

Por tanto, observar la proyección sobre el plano del suelo en 3D resulta clave para comprender mejor la capacidad de expansión con este conjunto de sensores. En la Figura 6.1e, se aprecia claramente la gran expansión realizada en la parte izquierda de la escena, que indica la aparición de un espacio libre en esa zona.

Ejemplo interior 2

En la siguiente imagen mostrada en la Figura 6.2a, se puede apreciar como la iluminación en este caso, es algo menos adversa que en el caso anterior, manteniendo la zona del suelo, una iluminación algo más homogénea que en el caso anterior, a excepción de la zona por donde incide la luz del sol. En este caso, la segmentación mediante superpíxeles no tiene errores (Figura 6.2c)

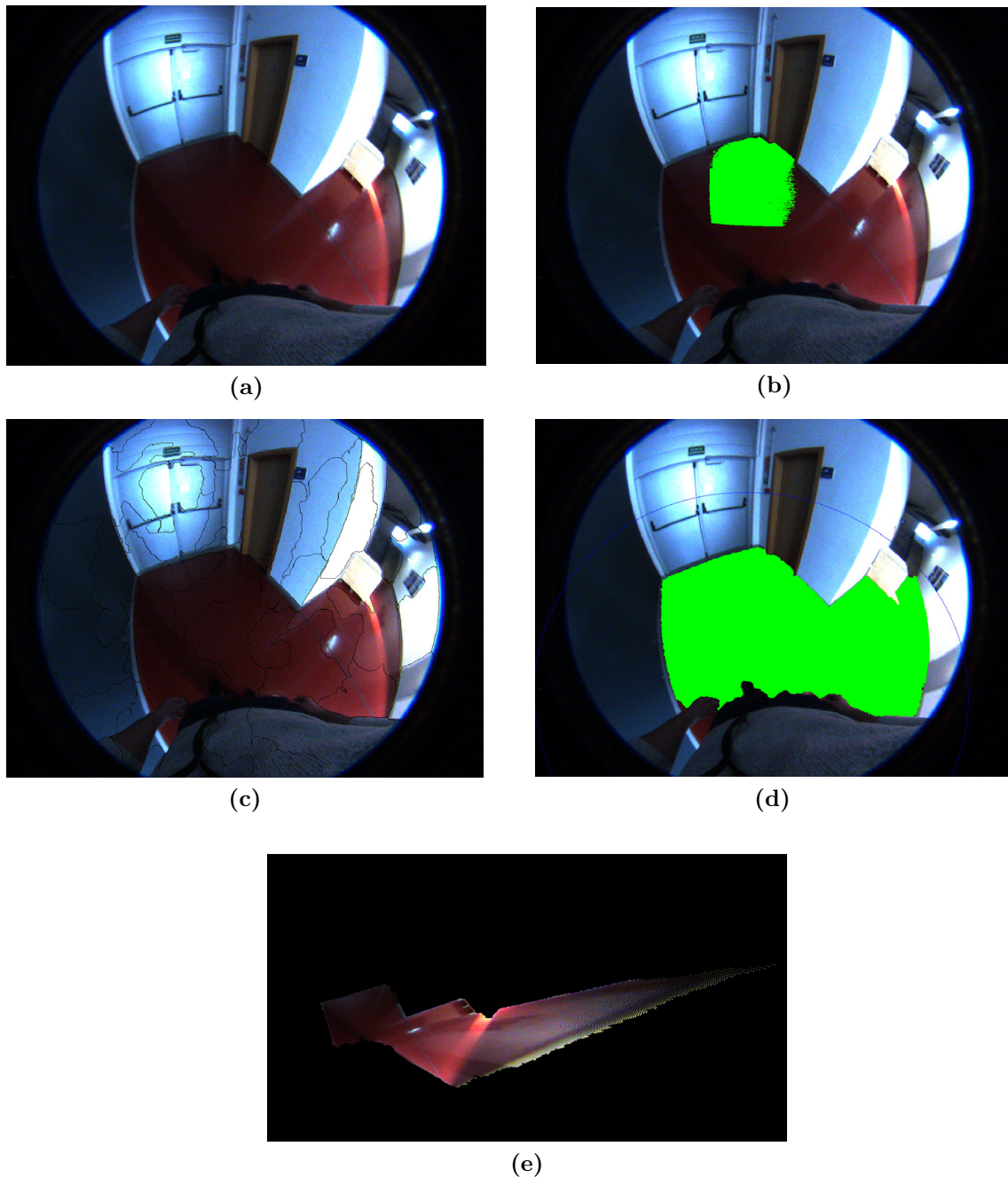


Figura 6.2: (a) Imagen de interior 2. (b) Detección de suelo mediante cámara de profundidad. (c) Segmentación mediante superpíxeles. (d) Resultado final de la detección de suelo. (e) Proyección 3D de la detección de suelo.

En la Figura 6.2d, vemos como el algoritmo es capaz de segmentar perfectamente el suelo de la escena, a pesar de los cambios de iluminación y de la presencia de algún reflejo. En la Figura 6.2e se puede ver la proyección del suelo expandido en 3D.

Evaluación de un entorno de interior con incidencia de luz solar

En la Tabla 6.1 se presentan los resultados obtenidos en la evaluación cuantitativa correspondiente a esta escena. Se muestran los valores medios, para las 6 imágenes etiquetadas de esta secuencia, de cada una de las métricas evaluadas. Vemos como los datos de precisión y recall son muy altos, cercanos a uno, lo que da idea del buen funcionamiento del sistema desarrollado para detectar el suelo. También puede apreciarse como el área expandida es de unas 5 veces en píxeles y de unas 10 veces en superficie real de suelo.

Evaluación entorno de interior 1	
Área expandida/Área profundidad	4.7943
Precision	0.9960
Recall	0.9162
Área 3D profundidad (m^2)	2.9513
Área 3D expandida (m^2)	29.8283

Tabla 6.1: Resultados medios de evaluación de un entorno de interior con incidencia de luz solar

6.2.2. Entorno de interior con gran incidencia de luz solar

Aquí se va a mostrar el entorno de un pasillo al que le llega una gran cantidad de luz solar que provoca un alto número de reflejos sobre el suelo que dificulta la detección del mismo.

Ejemplo interior 3

En la Figura 6.3d, se puede apreciar como, pese a esos reflejos, la segmentación del suelo es muy buena, ya que prácticamente todo el suelo es segmentado como tal. En la Figura 6.3e se observa la proyección 3D del suelo extendido. Pese a lo complicado de la escena, el resultado final de la expansión de suelo es muy satisfactorio.

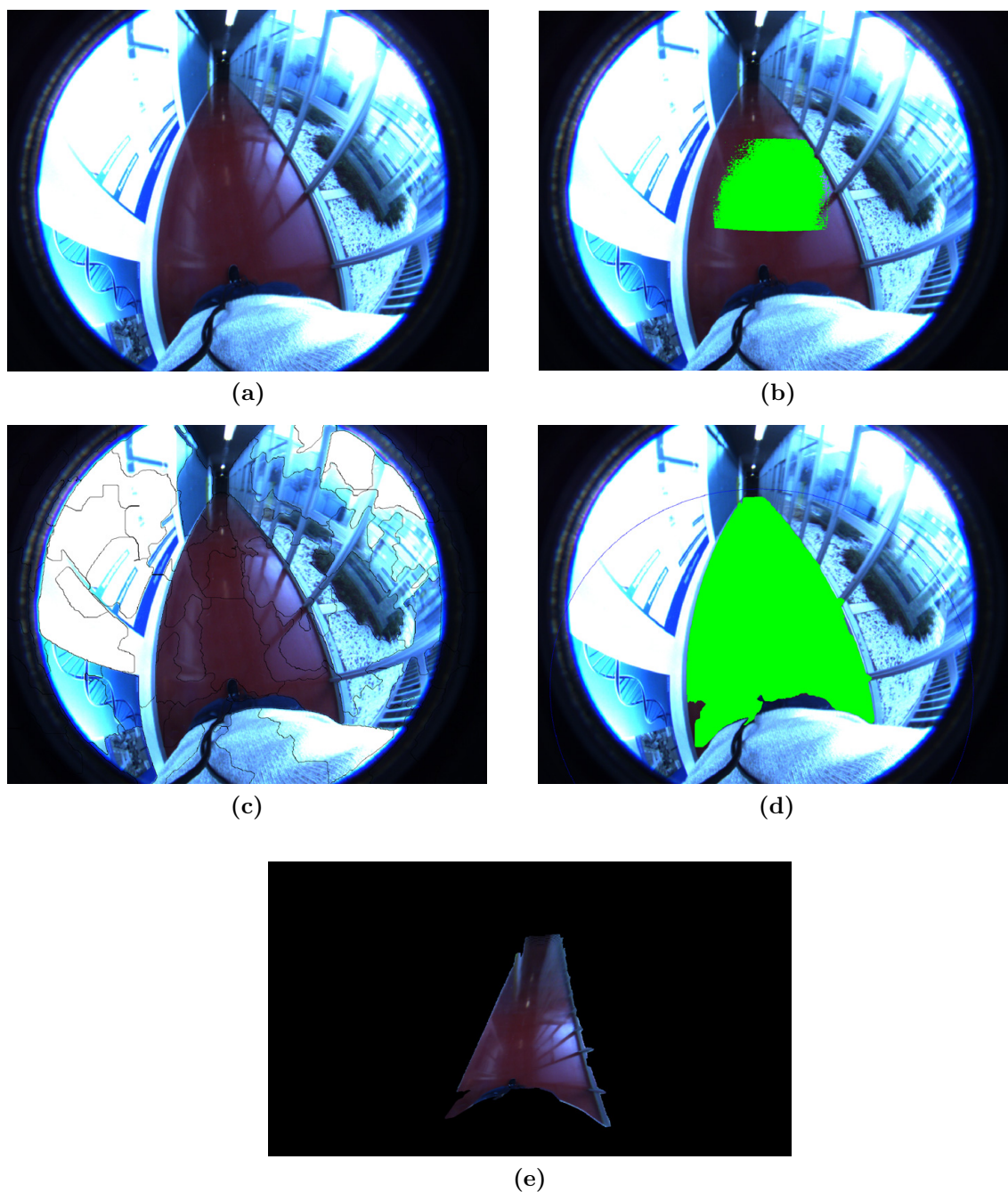


Figura 6.3: (a) Imagen de interior 3. (b) Detección de suelo mediante cámara de profundidad. (c) Segmentación mediante superpíxeles. (d) Resultado final de la detección de suelo. (e) Proyección 3D de la detección de suelo.

Ejemplo interior 4

En este caso se va a mostrar un ejemplo de este mismo entorno en el que hay un claro error en la detección de suelo. Como puede verse en la Figura 6.4a, hay una zona del suelo en la que existe un reflejo tan intenso, que prácticamente tiene el mismo color que la parte acristalada de la derecha de la imagen. Esto hace que el algoritmo etiquete como suelo una zona del cristal (Figura 6.4d). La segmentación por superpíxeles también ha

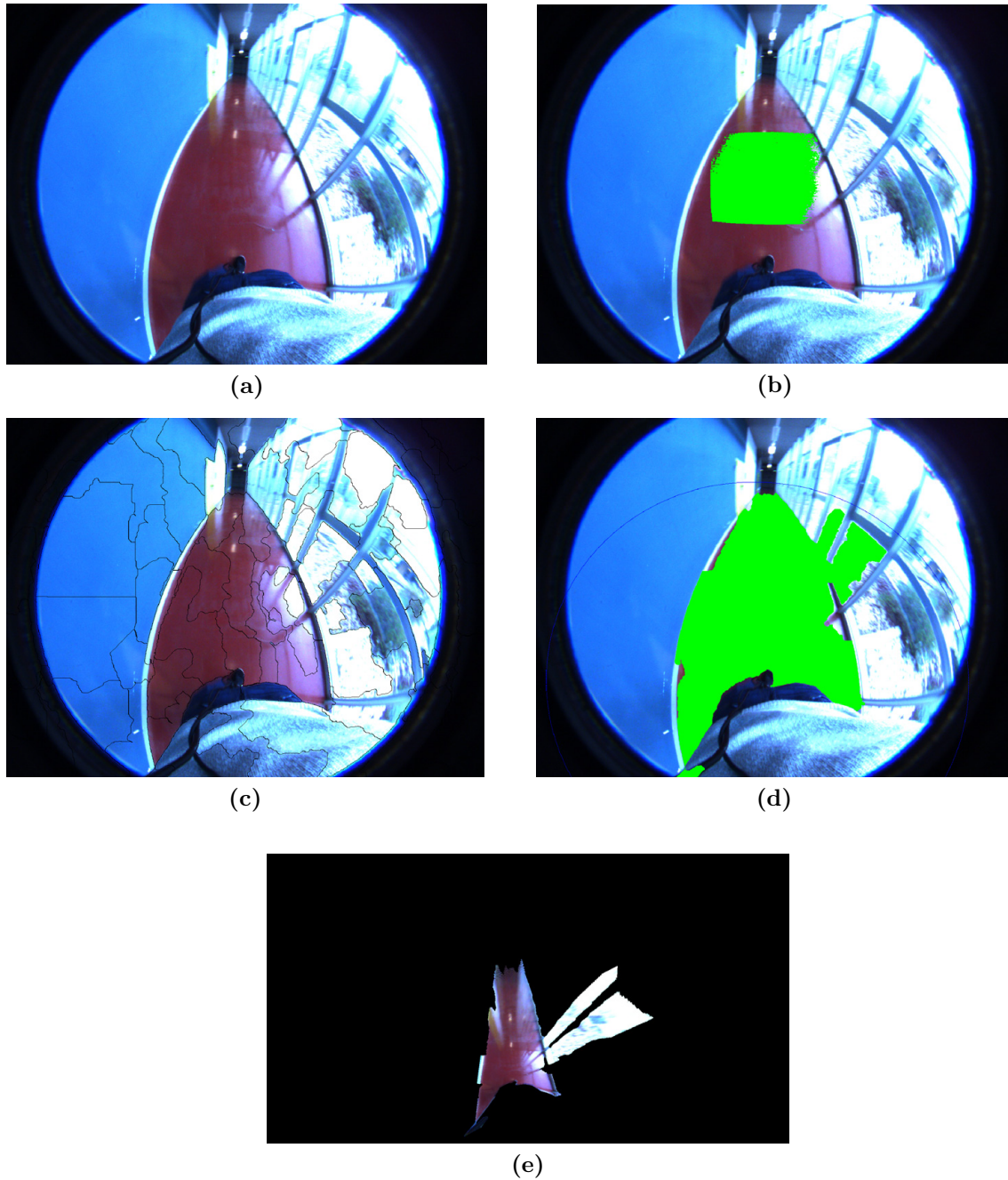


Figura 6.4: (a) Imagen de interior 4. (b) Detección de suelo mediante cámara de profundidad. (c) Segmentación mediante superpíxeles. (d) Resultado final de la detección de suelo. (e) Proyección 3D de la detección de suelo.

fallado segmentando esa parte del suelo en el mismo superpíxel que la parte acristalada. Este error podría ser corregido aumentando el número de superpíxeles para obtener una segmentación más precisa. Sin embargo, el tiempo de cómputo se vería penalizado.

Este ejemplo permite ilustrar bastante bien uno de los mayores inconvenientes en el uso de un sistema con una cámara de gran angular: Al ser capaz de captar luz de tantas zonas de la escena, el ajuste de la exposición de la cámara es muy complicado, ya que

la información viene de lugares muy dispares en cuanto a iluminación. Esto dificulta en gran medida la segmentación de la escena, y controlar estos efectos ha supuesto uno de los principales retos de este proyecto. Aunque en la gran mayoría de los casos ha sido posible hacerlo, existen situaciones en las que no es así, como se ha mostrado en este ejemplo.

Evaluación de un entorno de interior con gran incidencia de luz solar

Evaluación entorno de interior 2	
Área expandida/Área profundidad	2.8746
Precision	0.9788
Recall	0.9015
Área 3D profundidad (m^2)	2.6545
Área 3D expandida (m^2)	36.2142

Tabla 6.2: Resultados medios de evaluación de un entorno de interior con gran incidencia de luz solar

En la Tabla 6.2 se muestran los resultados obtenidos en la evaluación de los valores medios, para las 12 imágenes etiquetadas de esta secuencia, de las métricas evaluadas. Como puede apreciarse, los valores de precision y recall en esta ocasión son algo menores que en el primer caso debido a la mayor complejidad de la iluminación de esta escena frente a la anterior. Al tratarse de un pasillo muy largo, el área media expandida también es mayor que en el caso anterior, aunque la expansión en la imagen en píxeles es menor.

6.2.3. Entorno de interior con incidencia de luz artificial

Los siguientes ejemplos corresponden a un entorno en el que existe una menor influencia de la luz solar, y una mayor influencia de los reflejos de la luz artificial, a diferencia de los dos casos anteriores.

Ejemplo interior 5

En este caso (Figura 6.5), se puede apreciar como los reflejos generados en el suelo por las luces del techo, son tan distintas al resto que le resulta imposible al algoritmo etiquetarlos como suelo. Además, como el suelo obtenido mediante profundidad no toca a los superpíxeles generados en torno a esos reflejos, no pueden ser incluidos

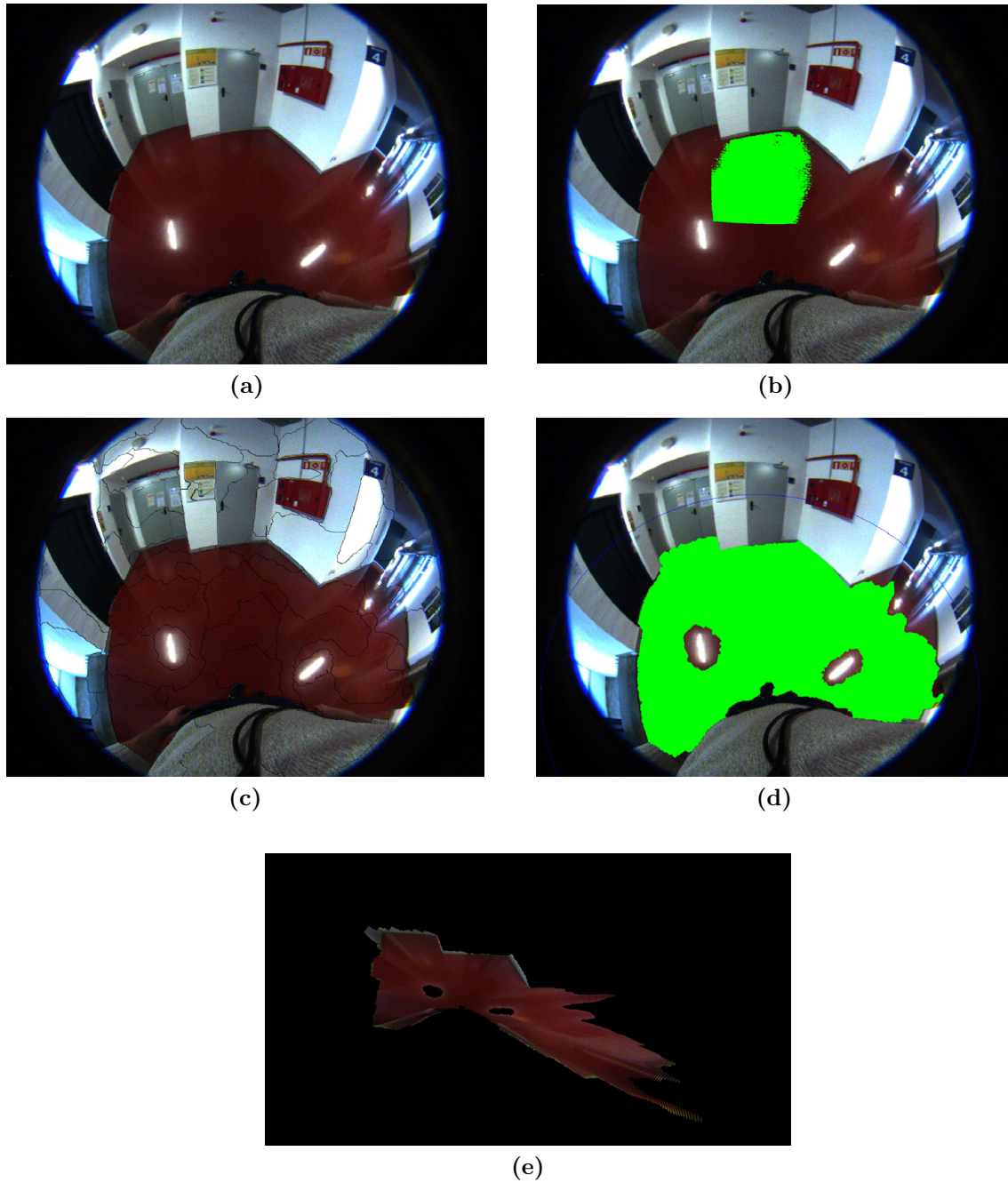


Figura 6.5: (a) Imagen de interior 5. (b) Detección de suelo mediante cámara de profundidad. (c) Segmentación mediante superpíxeles. (d) Resultado final de la detección de suelo. (e) Proyección 3D de la detección de suelo.

como parte del suelo mediante el umbral mínimo de puntos de la profundidad en el superpíxel explicado en el Capítulo 4. Este caso es un ejemplo en el que puede verse como en nuestro algoritmo es más importante obtener un valor de precisión alto, aún a costa de disminuir el recall, ya que es preferible dejar una parte del suelo sin segmentar para evitar que zonas que no son suelo sean etiquetadas como tal.

Ejemplo interior 6

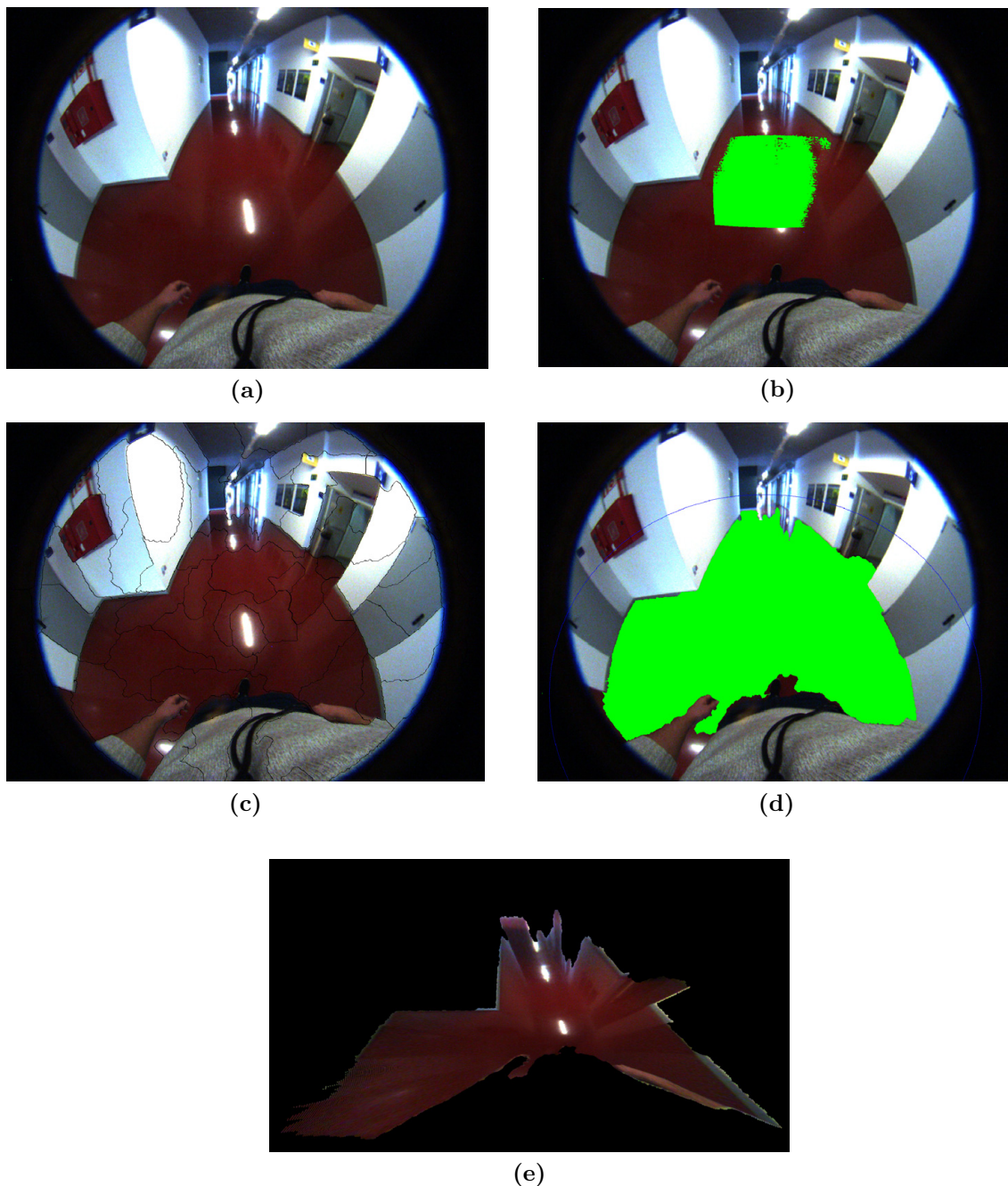


Figura 6.6: (a) Imagen de interior 6. (b) Detección de suelo mediante cámara de profundidad. (c) Segmentación mediante superpíxeles. (d) Resultado final de la detección de suelo. (e) Proyección 3D de la detección de suelo.

En este caso se va a mostrar otro ejemplo del mismo entorno en el que puede verse como el algoritmo es capaz de segmentar correctamente el suelo, a pesar de la existencia de reflejos similares a los anteriores. Como puede verse en la Figura 6.6a, en esta ocasión el suelo detectado por profundidad sí que ocupa una gran parte del superpíxel creado alrededor del reflejo, por lo que, sí que se segmenta como suelo.

Además el resto de reflejos creados por la luz artificial, al ser más pequeños no forman un superpíxel único, sino que forman parte de otro más grande. De esta forma, la similitud de los histogramas es la suficiente como para que sea etiquetado como suelo.

Evaluación de un entorno de interior con incidencia de luz artificial

Los valores medios para las 6 imágenes etiquetadas de esta escena pueden observarse en la Tabla 6.3. De nuevo, los valores de precisión y recall son bastante altos. Los valores de extensión de suelo son inferiores a los anteriores debido en parte a los grandes reflejos que aparecen al fondo del pasillo que dificultan, incluso a simple vista, la segmentación del suelo.

Evaluación entorno de interior 3	
Área expandida/Área profundidad	4.5619
Precision	0.9922
Recall	0.9527
Área 3D profundidad (m^2)	3.1314
Área 3D expandida (m^2)	26.2113

Tabla 6.3: Resultados medios de evaluación de un entorno de interior con incidencia de luz artificial

6.2.4. Entorno de interior con escasa iluminación

El siguiente entorno de interior es el de un garaje, en el cual la iluminación es muy escasa, lo que va a dificultar en gran medida la segmentación del suelo. Al haber esa escasa iluminación, ha sido necesaria la apertura del diafragma de la cámara fisheye al máximo para capturar toda la luz posible. Sin embargo, esta apertura del diafragma repercute negativamente en la nitidez de la imagen, ya que disminuye la profundidad de campo, haciendo que solo una pequeña parte de la imagen esté bien enfocada. Adicionalmente, existe otro hándicap en este entorno, y es la homogeneidad del color en la imagen. La parte inferior de las paredes y de las columnas tiene un color bastante similar al del suelo, donde además la pintura está muy desgastada por el uso, lo que complica la segmentación del suelo de la imagen.

Ejemplo interior 7

En la Figura 6.7a puede observarse lo nombrado anteriormente. Debido a esta homogeneidad en el color, ha sido necesario resintonizar los umbrales de similitud

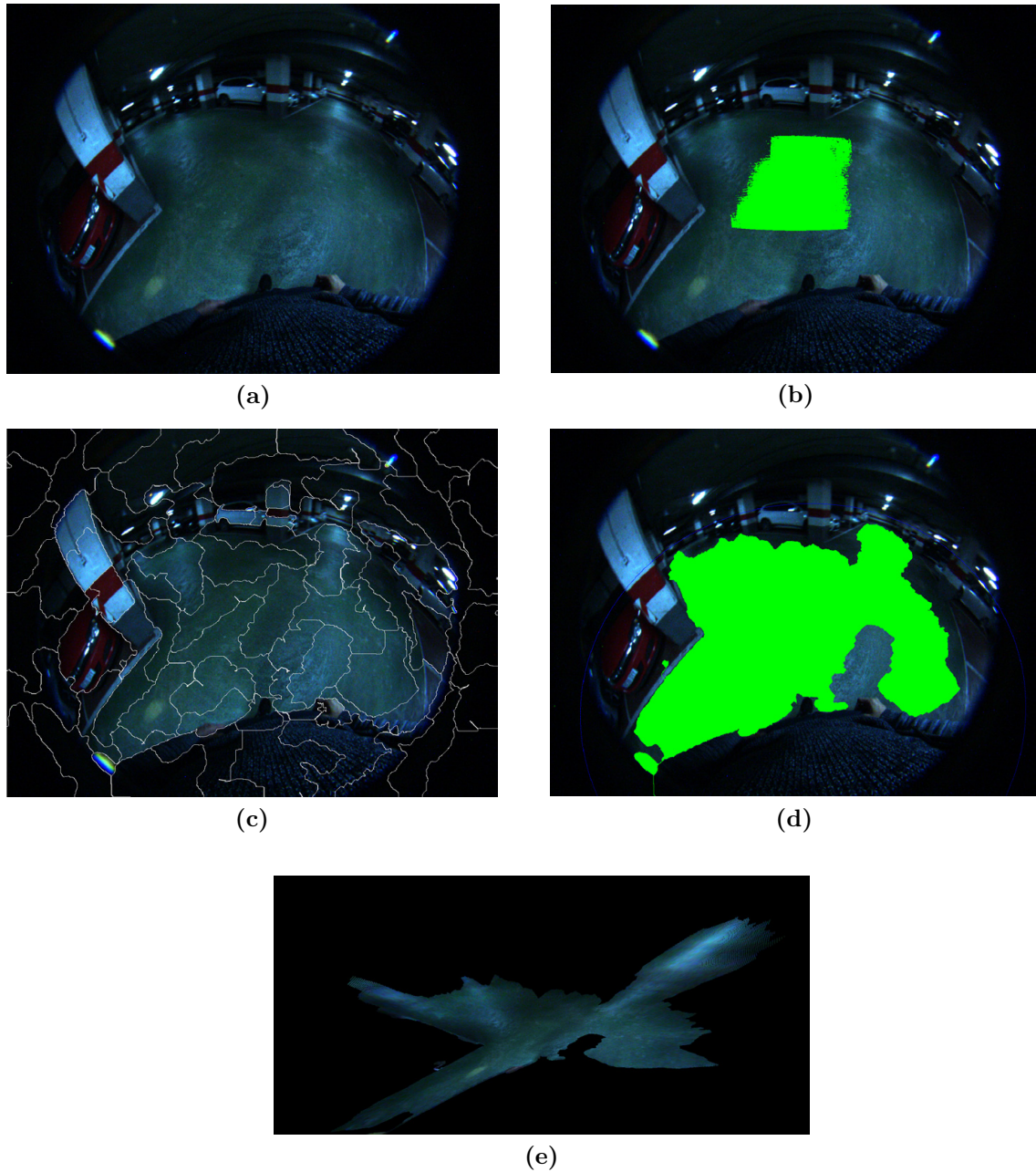


Figura 6.7: (a) Imagen de interior 7. (b) Detección de suelo mediante cámara de profundidad. (c) Segmentación mediante superpíxeles. (d) Resultado final de la detección de suelo. (e) Proyección 3D de la detección de suelo.

de histograma siendo más exigentes. Esto ha provocado que algunas zonas cercanas al usuario hayan sido etiquetadas como suelo. Pese a ello, en la Figura 6.7e pueden apreciarse perfectamente 3 direcciones distintas de espacio libre en esta escena. Esto muestra la capacidad de expansión de este sistema.

Ejemplo interior 8

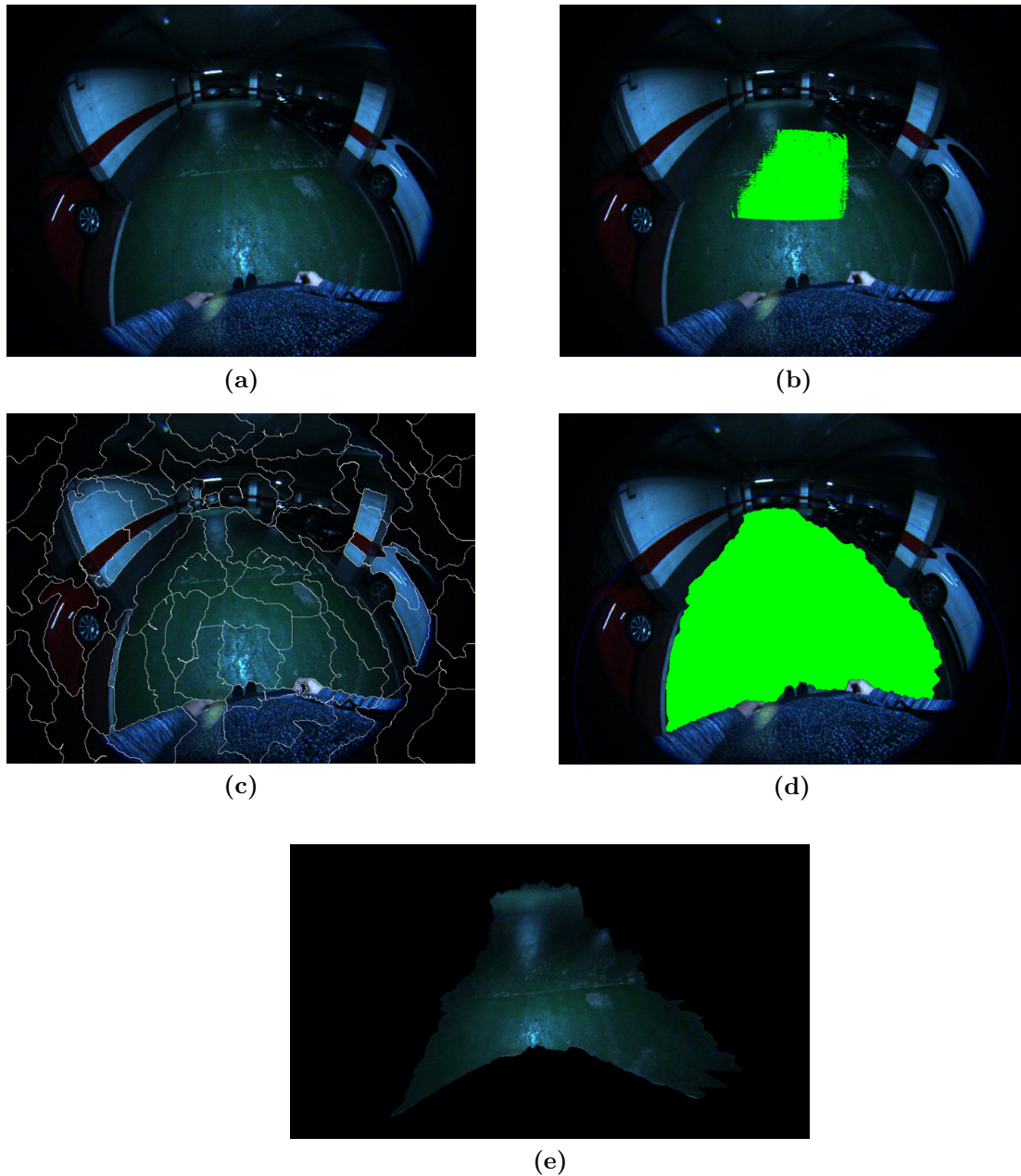


Figura 6.8: (a) Imagen de interior 8. (b) Detección de suelo mediante cámara de profundidad. (c) Segmentación mediante superpíxeles. (d) Resultado final de la detección de suelo. (e) Proyección 3D de la detección de suelo.

La escena de la Figura 6.8a muestra otro ejemplo perteneciente al entorno de garaje. En este caso, la iluminación es algo mejor, y el color del suelo se diferencia algo más del resto. Aunque también aparece algún tipo de reflejo procedente de la luz del techo. Como puede verse en la Figura 6.7e, en este caso, la segmentación del suelo es bastante buena a pesar de la mala iluminación del entorno, en parte porque corresponde con una zona en la que la pintura del suelo está en mejor estado de conservación que en la anterior.

Evaluación de un entorno de interior con escasa iluminación

Los resultados medios de las 4 imágenes etiquetadas de esta secuencia, arrojan unos valores de recall bastante más bajos que en los casos anteriores. Esto se debe principalmente a lo estrictos que se ha sido en cuanto a umbrales para evitar disminuir la precisión del algoritmo, que sigue siendo muy alta. También destaca la gran capacidad de expansión del sistema en este entorno, debido evidentemente al gran espacio libre que presenta la escena.

Evaluación entorno de interior 4	
Área expandida/Área profundidad	4.5594
Precision	0.9927
Recall	0.8254
Área 3D profundidad (m^2)	3.3572
Área 3D expandida (m^2)	65.8452

Tabla 6.4: Resultados medios de evaluación de un entorno de interior con escasa iluminación

6.3. Entornos exteriores

Como se ha comentado al comienzo de este capítulo, no solo se han realizado pruebas en entornos interiores, tal y como recomienda el fabricante de la cámara de profundidad, sino que se ha querido ir más allá y evaluar nuestro algoritmo en entornos exteriores. Se ha comprobado que, en entornos en los que la luz del sol incide directamente, la cámara de profundidad tiene muchas dificultades para obtener la nube de puntos debido a las interferencias con la luz infrarroja del sol. Sin embargo, en zonas de sombra o donde la luz del sol no incida directamente durante los días soleados, y en días nublados, se ha comprobado como no existe ningún problema a la hora de utilizar la cámara de profundidad, consiguiéndose unos resultados más satisfactorios de lo previsto.

6.3.1. Entorno de exterior con color homogéneo

Este escenario es el que se ha utilizado a lo largo de esta memoria para explicar gran parte del funcionamiento de nuestro algoritmo. Así pues se van a mostrar otros ejemplos que complementen al ya mostrado en otros capítulos. Este entorno corresponde a un pasillo exterior en el que existe una dificultad añadida, y es que, tal y como ocurría en el garaje, el inicio de la pared tiene prácticamente el mismo color que el suelo, lo que dificulta la segmentación del mismo

Ejemplo exterior 1

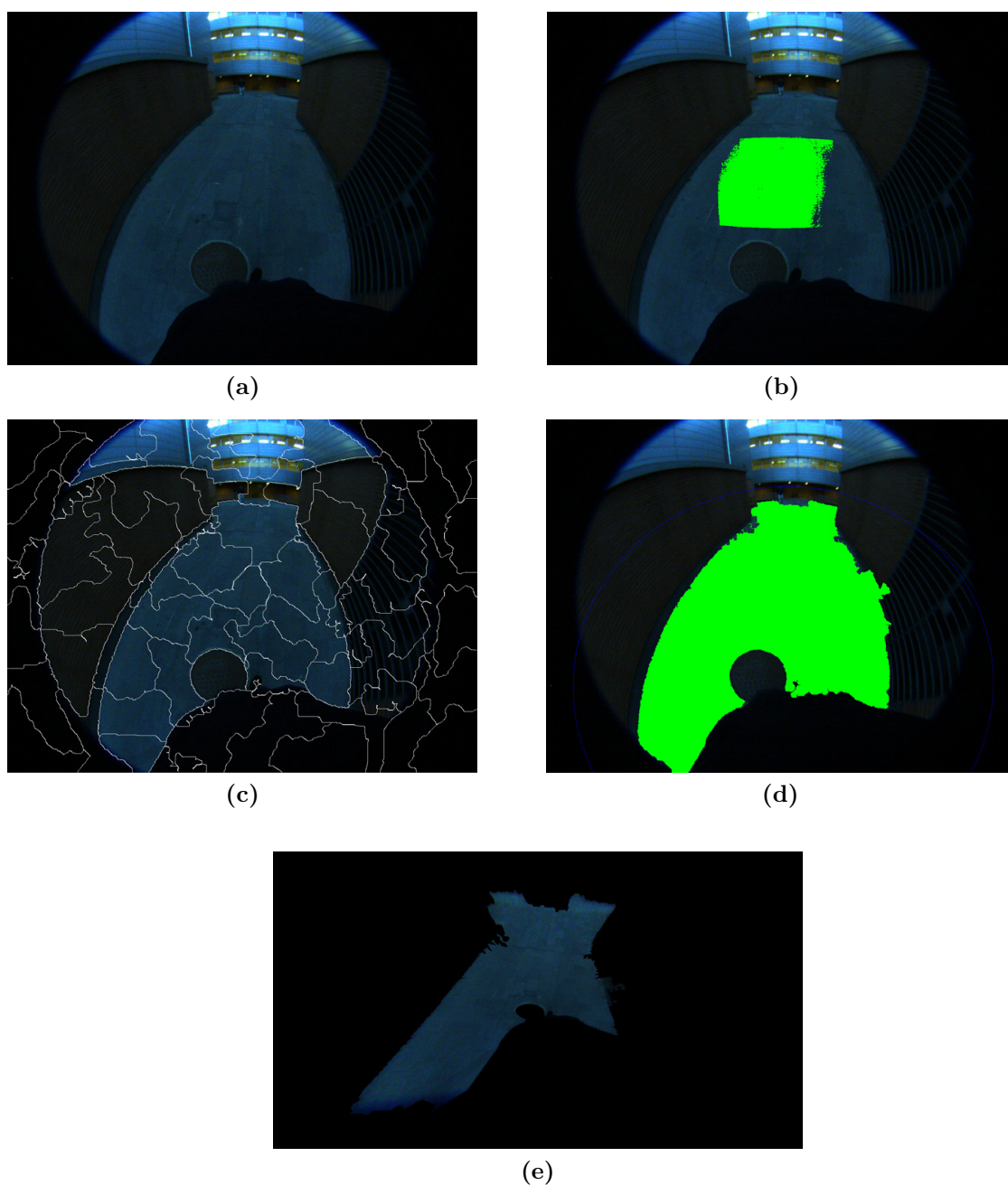


Figura 6.9: (a) Imagen de exterior 1. (b) Detección de suelo mediante cámara de profundidad. (c) Segmentación mediante superpíxeles. (d) Resultado final de la detección de suelo. (e) Proyección 3D de la detección de suelo.

En el ejemplo de la Figura 6.9 se muestra un caso en el que hay un error en la segmentación de la alcantarilla. El color de ésta es distinto al del resto del suelo, y al estar tan cerca del sujeto, se ha creado un superpíxel a su alrededor. Tal y como ocurría con los reflejos en el entorno interior 5, al no haber suficientes puntos del suelo detectado mediante profundidad en ese superpíxel, y ser el color claramente distinto al del resto del suelo, esa parte de la imagen no es segmentada como tal. Este error en la

segmentación se debe también al uso de umbrales muy exigentes en cuanto a similitud de histogramas con el fin de aumentar al máximo la precisión del algoritmo.

Ejemplo exterior 2

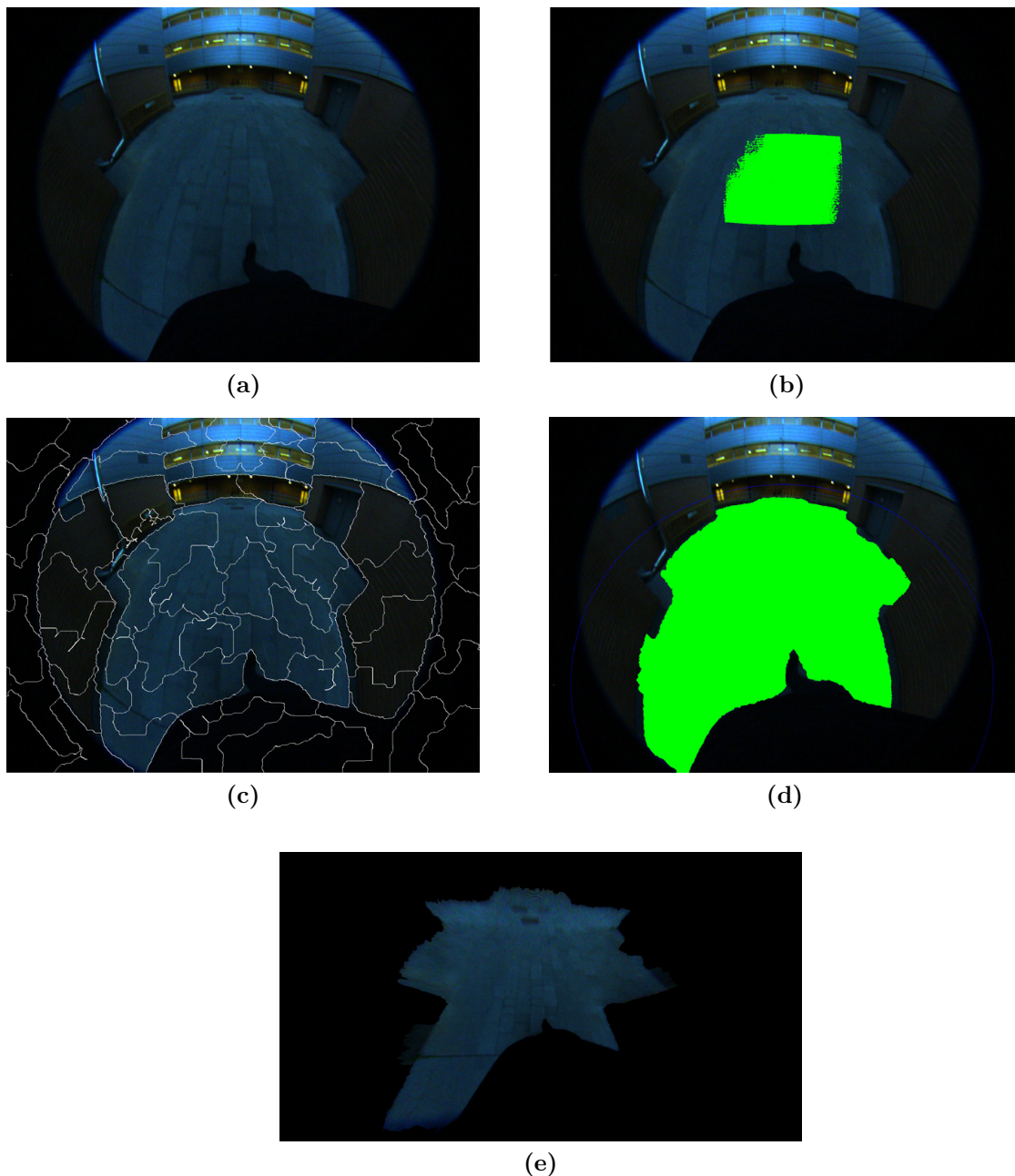


Figura 6.10: (a) Imagen de exterior 2. (b) Detección de suelo mediante cámara de profundidad. (c) Segmentación mediante superpíxeles. (d) Resultado final de la detección de suelo. (e) Proyección 3D de la detección de suelo.

En este caso, la escena de la Figura 6.10a permite más cantidad de expansión de suelo ya que el pasillo ahora es más abierto. Como vemos, la segmentación del plano del suelo es satisfactoria.

Evaluación de un entorno de exterior con color homogéneo

Observando los resultados medios para las 8 imágenes etiquetadas en esta secuencia, presentes en la Tabla 6.5 se aprecia como en este entorno, tanto la precisión como el recall son realmente altos, obteniéndose en ambos casos valores muy cercanos a 1. Esto da una idea del correcto funcionamiento del algoritmo en este entorno de exteriores. Además, la cantidad de área 3D expandida es realmente notoria, llegando a valores unas 20 veces superiores al área detectada por la cámara de profundidad. Este valor de área expandida es grande también gracias a la gran longitud del pasillo y a que al final del mismo aparece una zona mucho más amplia que favorece esa expansión.

Evaluación entorno de exterior 1	
Área expandida/Área profundidad	4.6110
Precision	0.9941
Recall	0.9844
Área 3D profundidad (m^2)	3.1496
Área 3D expandida (m^2)	60.6491

Tabla 6.5: Resultados medios de evaluación de un entorno de exterior con color homogéneo

6.3.2. Entorno de exterior con terreno irregular

Este último entorno corresponde a una zona de exterior en la que la incidencia de la luz solar es mayor que en el ejemplo anterior. Se trata de un entorno en el que hay una zona de césped que es la que se pretende segmentar. Al tratarse de un terreno bastante irregular, el área de suelo detectado por parte de la cámara de profundidad es menor que en los casos anteriores, lo que dificulta la segmentación.

Ejemplo exterior 3

En el ejemplo de la Figura 6.11 se puede observar como la segmentación es bastante satisfactoria. Exceptuando la zona de la esquina que no se ha segmentado como suelo, el resto se consigue etiquetar prácticamente perfecto como tal.

Ejemplo exterior 4

En este último ejemplo, se va a mostrar de nuevo, como la línea del horizonte consigue paliar ciertos errores en la segmentación. En la Figura 6.12c se puede apreciar

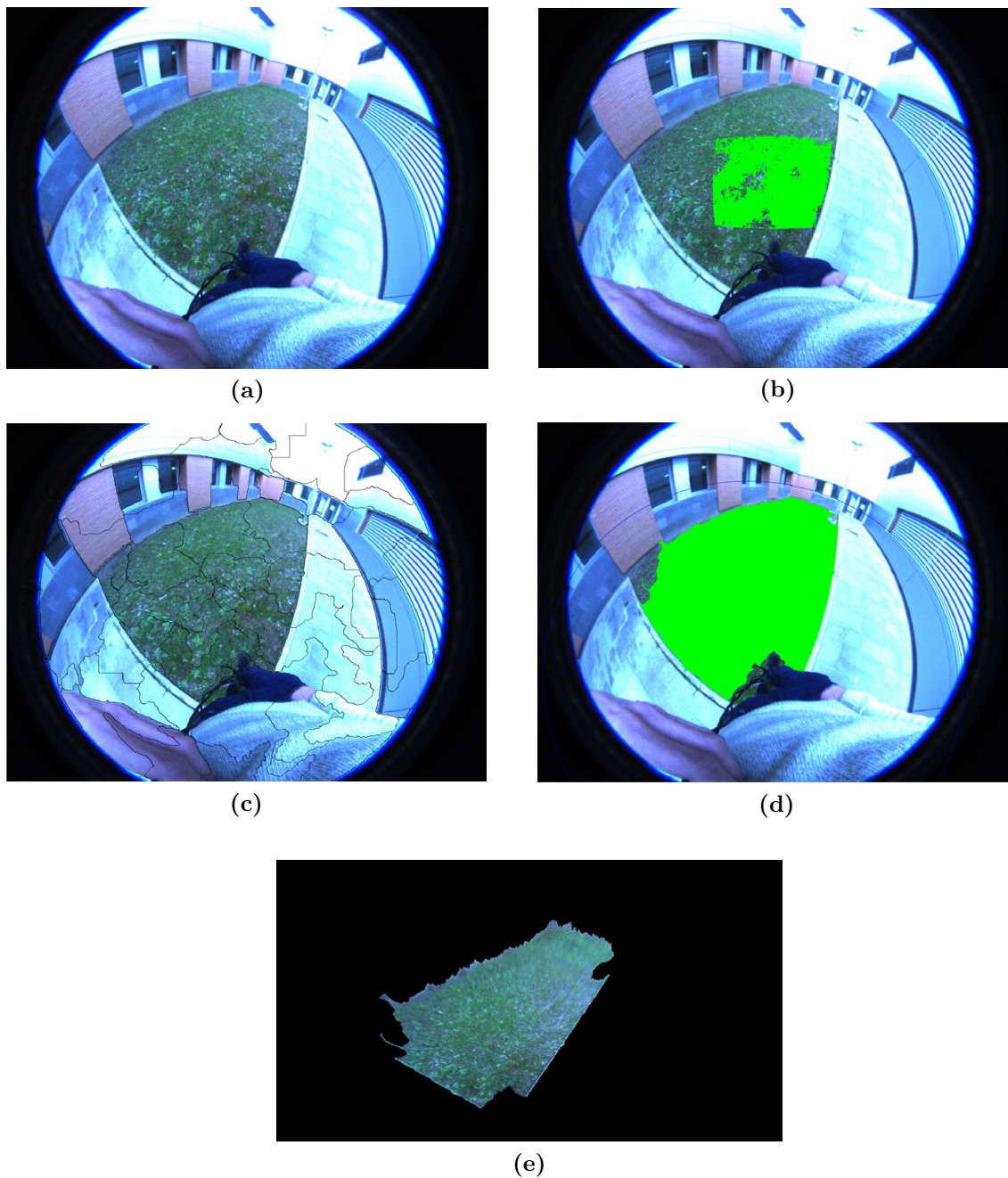


Figura 6.11: (a) Imagen de exterior 3. (b) Detección de suelo mediante cámara de profundidad. (c) Segmentación mediante superpíxeles. (d) Resultado final de la detección de suelo. (e) Proyección 3D de la detección de suelo.

que la segmentación mediante superpíxeles es correcta. Sin embargo en la Figura 6.12d la segmentación de suelo continúa por la pared del fondo. La línea del horizonte evita que esa segmentación errónea continúe por todo el superpíxel, lo cual empeoraría más el resultado final, pudiendo incluso provocar la propagación de esa segmentación errónea por otras partes de la imagen. Gracias al uso de la información 3D, se consiguen contrarrestar posibles errores del algoritmo de color, lo que le otorga robustez al sistema.

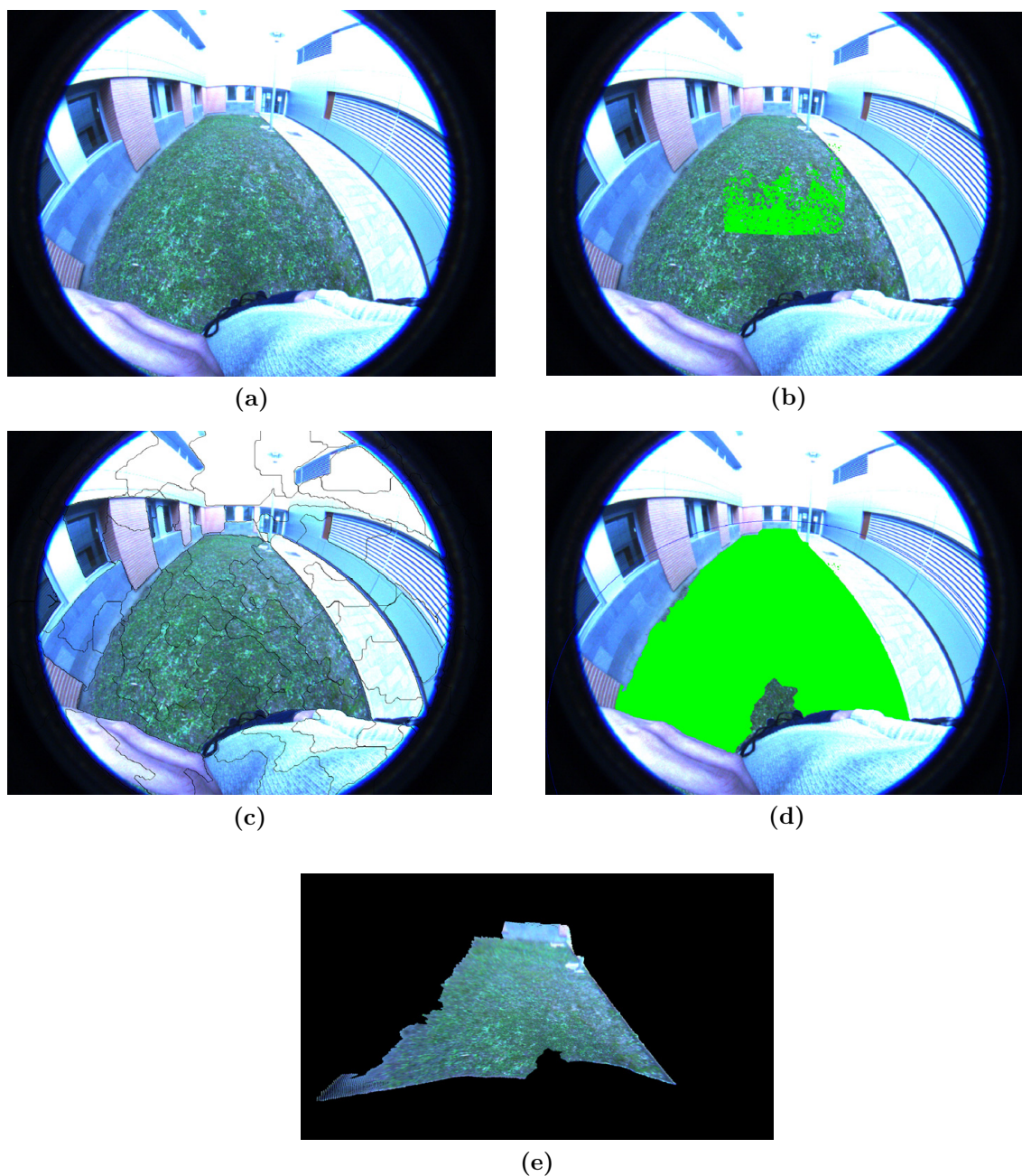


Figura 6.12: (a) Imagen de exterior 4. (b) Detección de suelo mediante cámara de profundidad. (c) Segmentación mediante superpíxeles. (d) Resultado final de la detección de suelo. (e) Proyección 3D de la detección de suelo.

Evaluación de un entorno de exterior con terreno irregular

En cuanto a los resultados recogidos en la Tabla 6.6 que presenta los valores medios para las 4 imágenes de este entorno, destaca en primer lugar el alto valor de la precisión. Este valor está condicionado al propio método de segmentación del ground truth mediante superpíxeles, que hace que se obtenga un valor de precisión optimista, y superior al real. Aún así a la vista de las imágenes, ese valor es muy alto,

aunque evidentemente no es del 100 %. También destaca, tal y como se ha comentado anteriormente, el bajo valor de área 3D obtenida mediante la cámara de profundidad debido a la irregularidad del terreno.

Evaluación entorno de exterior 2	
Área expandida/Área profundidad	3.9314
Precision	1
Recall	0.9613
Área 3D profundidad (m^2)	2.5338
Área 3D expandida (m^2)	56.8581

Tabla 6.6: Resultados medios de evaluación de un entorno con terreno irregular

6.4. Conclusiones de la evaluación

Antes de pasar a las conclusiones de todo lo visto a lo largo de este capítulo, hay que aclarar que en cuanto a los datos de precision y recall, la mejora de uno implica el empeoramiento del otro, por lo que existe un compromiso entre ambos que hay que tener en cuenta. En este algoritmo hay dos aspectos importantes a destacar respecto a la precision y el recall, válidos para todas los entornos vistos

1. En primer lugar, el algoritmo desarrollado se ha ajustado para obtener valores de precision lo más altos posibles, por lo que el recall en ocasiones puede verse perjudicada. Al tratarse de un algoritmo para el guiado de personas en el que está en juego la seguridad del individuo, es preferible dejar de etiquetar algo como suelo aunque realmente lo sea, que indicar que algo es suelo y que realmente no lo sea.
2. El propio método de etiquetado de ground truth, puede condicionar en ocasiones los resultados de precision y recall. Los posibles errores en la segmentación por superpíxeles condicionarán el etiquetado del ground truth generando valores ligeramente diferentes a los que realmente se obtendrían con un ground truth más preciso.

Finalmente y a modo de resumen, se van a mostrar los resultados obtenidos de la evaluación global del algoritmo, diferenciando entre entornos de interior y de exterior.

A partir de los resultados obtenidos en las Tablas 6.7 y 6.8, se pueden sacar las

siguientes conclusiones sobre el funcionamiento del algoritmo en los diferentes entornos en los que se ha testado:

- El dato de recall es claramente mejor en los entornos de exterior. Esto se debe principalmente a que en estos entornos no existen los reflejos que provienen de la luz artificial o de la luz solar, ya que se han tomado en días nublados.
- El valor de precision puede estar en este caso ligeramente condicionado por el procedimiento de etiquetado de ground truth. Aumentando el número de superpíxeles en el etiquetado, probablemente se reduciría algo este número, aunque continuaría siendo muy bueno.
- La expansión de suelo en interiores es menor que en exteriores. Algo que parece lógico, ya que el espacio libre en una escena en interiores suele ser bastante menor que en el exterior.

Evaluación entornos interiores	
Área expandida/Área profundidad	4.0201
Precision	0.9899
Recall	0.8990
Área 3D profundidad (m^2)	3.0236
Área 3D expandida (m^2)	39.5247

Tabla 6.7: Resultados medios de evaluación de entornos interiores

Evaluación entornos exteriores	
Área expandida/Área profundidad	4.2442
Precision	0.9970
Recall	0.9728
Área 3D profundidad (m^2)	2.8417
Área 3D expandida (m^2)	58.7536

Tabla 6.8: Resultados medios de evaluación de entornos exteriores

Capítulo 7

Conclusiones y trabajo futuro

En este proyecto se ha implementado un sistema de detección del espacio libre en una escena que ayude en la planificación de trayectorias para el guiado de personas mediante la extensión de la segmentación 3D de suelo proporcionada por el sensor de rango, utilizando la información de color procedente de la cámara fisheye.

Tras un análisis del estado del arte, se concluyó que a pesar de haberse desarrollado varios sistemas de navegación adaptados a personas con discapacidad visual, no son muchos los que aprovechan la información visual que puede proporcionar una cámara y los que lo hacían, únicamente se centraban en la evitación de obstáculos ya que disponían de un limitado campo de vista. La combinación de sensores utilizada supone, por tanto, una novedad frente a lo que ya existe en la literatura, permitiendo obtener un gran campo de vista de la escena, y de esta manera obtener información más completa de la misma, permitiendo añadir al clásico sistema de evitación de obstáculos, información de espacio libre para ayudar en la planificación de trayectorias.

Fijado el objetivo principal del proyecto se empezó a trabajar en la sincronización y calibración de los dos sensores, tras un análisis del entorno de trabajo (ROS), así como el estudio del método de calibración realizado por [17]. Posteriormente, tras un análisis de las herramientas PCL y después de diferentes pruebas en las que se buscaba un equilibrio entre precisión y coste computacional, se implementó el algoritmo de extracción de suelo 3D mediante la cámara de profundidad. El siguiente paso fue la expansión del mismo mediante la información de color obtenida por medio de la cámara fisheye a través de la librería OpenCV, apoyándose en la calibración y el suelo 3D, y utilizando la información 3D para mejorar la expansión de suelo. También se buscó discretizar el espacio libre obtenido en la segmentación de suelo para poder ofrecer al usuario diferentes instrucciones que le sean útiles en la planificación de

trayectorias y que sean complementarias a las de evitación de obstáculos. Por último en la fase experimental se evaluó el algoritmo en diferentes entornos que tuvieran unas características de iluminación diversas, consiguiendo unos resultados muy satisfactorios y obteniendo una base de datos, inexistente hasta entonces, que pueda ser utilizada en el futuro en otros trabajos.

También es necesario realizar un ejercicio de exploración del trabajo futuro y considerar las posibles mejoras a aplicar. Concretamente, se considera apropiado en primer lugar, implementar la detección de puertas, propuesta en este trabajo como prueba de concepto, pero de la que ya se tienen una gran cantidad de elementos que permitirían implementarla y añadirla al sistema actual. Adicionalmente, también podría resultar útil la mejora del sistema de navegación, pudiendo añadir nuevas instrucciones al catálogo propuesto, así como proponer otro método diferente del realizado en este trabajo, o la optimización del mismo. Otro aspecto importante es el testeo del algoritmo con individuos con deficiencias visuales reales para poder mejorar el sistema en aspectos no contemplados anteriormente y que puedan ser de utilidad para los usuarios.

Para terminar, solo queda añadir que se han cumplido los objetivos planteados al comienzo del proyecto. La sensación al finalizar este trabajo ha sido satisfactoria puesto que no se han encontrado barreras infranqueables, aunque el esfuerzo ha sido considerable. La expectativa ahora mismo, es que este trabajo sirva como inicio para posteriores investigaciones en nuevas técnicas para guiado de personas con deficiencias visuales.

Anexos

Anexo A

Adquisición y sincronización de la información

En este Anexo se va a dar una explicación de los pasos más importantes que se han llevado a cabo para realizar la adquisición y la sincronización de las cámaras. Para ello se ha utilizado el sistema operativo ROS (Robot Operating System).

En cuanto a la adquisición de la información, ROS se organiza mediante nodos, que son procesos que realizan algún tipo de cómputo. En nuestro caso, tendríamos dos nodos que controlarían la cámara de profundidad y la cámara fisheye. Para comunicarse los nodos entre sí, utilizan *topics*. Estos *topics* tienen semánticas anónimas de publicación y suscripción. En general, los nodos no son conscientes de con quién se están comunicando. Los nodos que necesiten adquirir cierta información se suscriben a los *topics* necesarios, mientras que los que generan datos lo publican en un determinado *topic*. Puede haber múltiples editores y suscriptores a un mismo *topic*. En nuestro caso, para obtener la información tanto de profundidad como de color, es necesario suscribirse a los siguientes *topics*:

- `camera/depth_registered/points`
- `ueye/image_raw`

Para decirle a ROS que se desea recibir mensajes sobre un determinado *topic* se debe hacer mediante una función llamada *callback()*. Cada vez que llegue un nuevo mensaje se llamará a esa función para realizar la adquisición de los datos.

ROS también permite grabar secuencias, en las que se guardan los datos adquiridos de los *topics* seleccionados. Posteriormente se pueden reproducir, simulando una

adquisición online de los sensores, pudiendo cambiar la velocidad de reproducción o pausarla, lo cual es muy útil para testear un algoritmo siempre en las mismas condiciones durante el proceso de desarrollo del mismo, pudiendo corregir y pulir ciertos errores de forma más sencilla. Estas secuencias se denominan *rosbags*.

En cuanto a la sincronización de los dispositivos, una vez se ha realizado la adquisición mediante la función *callback()*, se utiliza un filtro sincronizador, el cual realiza la sincronización mediante mensajes que estén lo más cercanos posibles en tiempo, tratando de optimizarlo entre todos los *topics* que aparecen. En la web de ROS se explica este método con mayor claridad¹. Dentro del *main()* se encuentra el *mainLoop()* que contiene tanto la función *callback()* como la función *startMainLoop()*, dentro de la cual se encuentran tanto la suscripción a los topics, como la sincronización entre ellos. Una vez llega un nuevo dato y se comprueba que la adquisición y la sincronización son correctas, comienza a ejecutarse el algoritmo de detección y expansión de suelo (llamado en nuestro caso *execute()*). Las funciones *callback()* y *startMainLoop()* se detallan a continuación:

```
void callback (const sensor_msgs::ImageConstPtr &msg_ueye, const sensor_msgs::PointCloud2::ConstPtr &msg_cloud)
{
    {
        boost::mutex::scoped_try_lock lock(data_ready_mutex_);
        if (!lock)
            return;

        cv_bridge::CvImagePtr cv_ptr_ueye;
        try
        {
            cv_ptr_ueye = cv_bridge::toCvCopy(msg_ueye);
            img_ueye = cv_ptr_ueye->image;

            color_cloud.reset(new pcl::PointCloud<pcl::PointXYZRGB>);
            cloud.reset(new pcl::PointCloud<pcl::PointXYZ>);
            pcl::fromROSMsg(*msg_cloud,*color_cloud);
            pcl::copyPointCloud(*color_cloud,*cloud);
            // pcl::fromROSMsg(*msg_cloud,*color_cloud);

        }
        catch (cv_bridge::Exception& e)
        {
            ROS_ERROR("cv_bridge exception: %s", e.what());
            return;
        }

        //image_pub.publish(ros_image);
    }
    data_ready_cond_.notify_one();
}
```

¹http://wiki.ros.org/message_filters/ApproximateTime

```
void startMainLoop (int argc, char* argv[])
{
    //ROS subscribing
    ros::init(argc, argv, "kinect_navigator_node");
    ros::NodeHandle nh;

    // Create a ROS subscriber for the input point cloud
    message_filters::Subscriber<sensor_msgs::Image> sub_ueye (nh, "ueye/image_raw", 2);
    message_filters::Subscriber<sensor_msgs::PointCloud2> sub_cloud (nh, "camera/depth_registered/points", 2);

    while((sub_ueye.getSubscriber().getNumPublishers()==0) or (sub_cloud.getSubscriber().getNumPublishers()==0))
    {
        ROS_INFO("Waiting for subscribers");
        sleep(1);
    }

    ROS_INFO("Got publishers");

    typedef message_filters::sync_policies::ApproximateTime<sensor_msgs::Image, sensor_msgs::PointCloud2> MySyncPolicy;
    message_filters::Synchronizer<MySyncPolicy> sync(MySyncPolicy(10), sub_ueye, sub_cloud);
    sync.registerCallback(boost::bind(&mainLoop::callback, this, _1, _2));

    capture_.reset(new ros::AsyncSpinner(0));

    {
        boost::unique_lock<boost::mutex> lock(data_ready_mutex_);
        capture_->start();

        get_ocam_model(&o, "./calib_results_mod.txt");
        std::cout << "calib_results.txt loaded" << std::endl;

        while (nh.ok())
        {
            bool has_data = data_ready_cond_.timed_wait (lock, boost::posix_time::millisec(10000));
            if (has_data)
                this->execute ();
        }
    }
}
```

Anexo B

Ampliación de resultados

En este Anexo se va a mostrar una ampliación de los resultados obtenidos mediante el sistema de expansión de suelo. Se van a mostrar el resto de imágenes utilizadas para evaluar los resultados obtenidos en el Capítulo 6. Los entornos utilizados han sido el edificio del Instituto de Investigación e Ingeniería de Aragón (I3A), el garaje de un domicilio particular, y los exteriores del edificio Torres Quevedo, con la idea de que esos entornos fueran lo más complementarios posibles y de esta forma enfrentar al algoritmo a situaciones variadas.

Para cada entorno se van a mostrar por este orden de izquierda a derecha: imagen utilizada, detección de suelo por parte de la cámara de profundidad y expansión final del suelo utilizando la cámara fisheye.

B.1. Resultados de entorno interior con incidencia de luz solar (I3A)

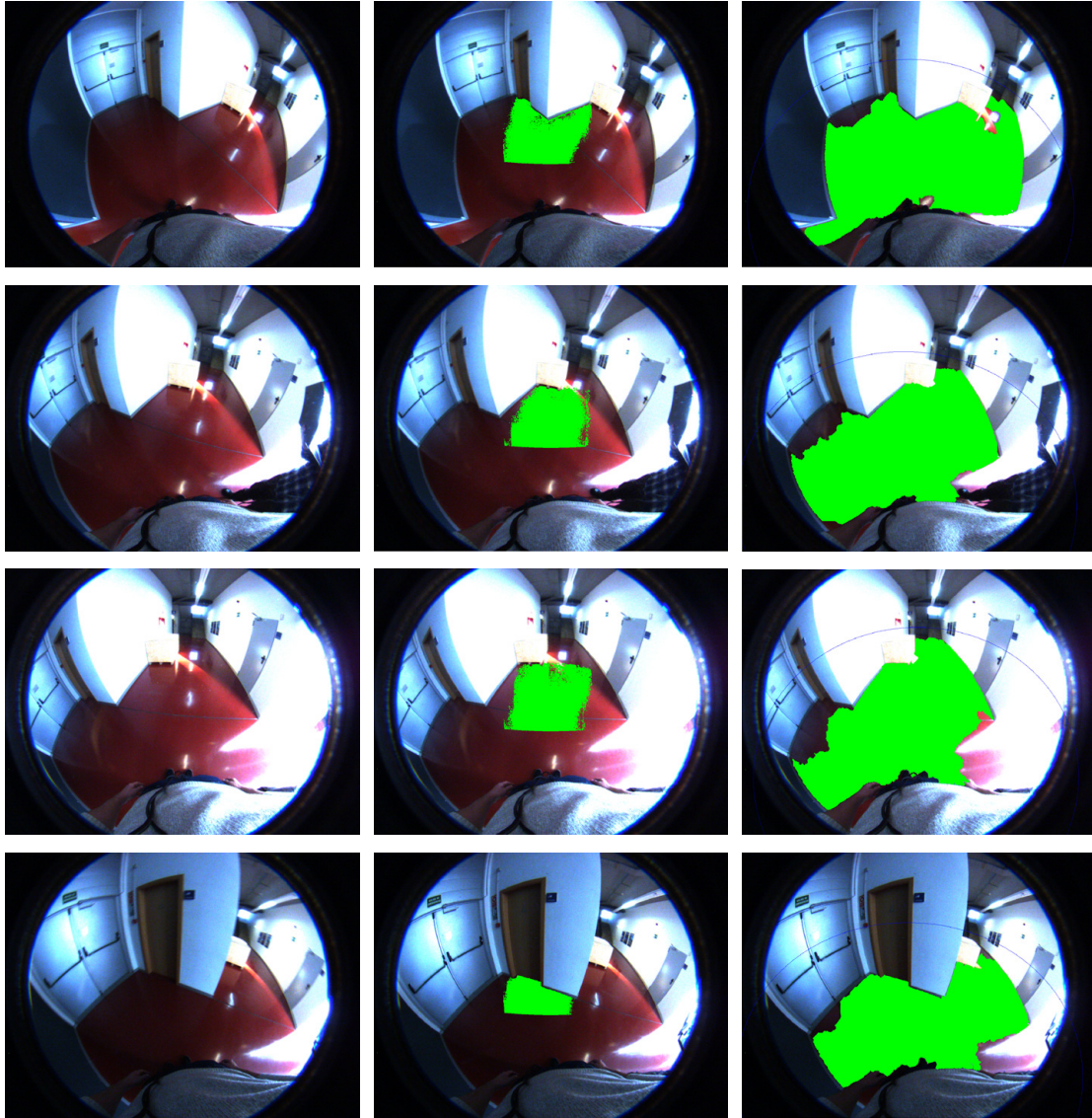
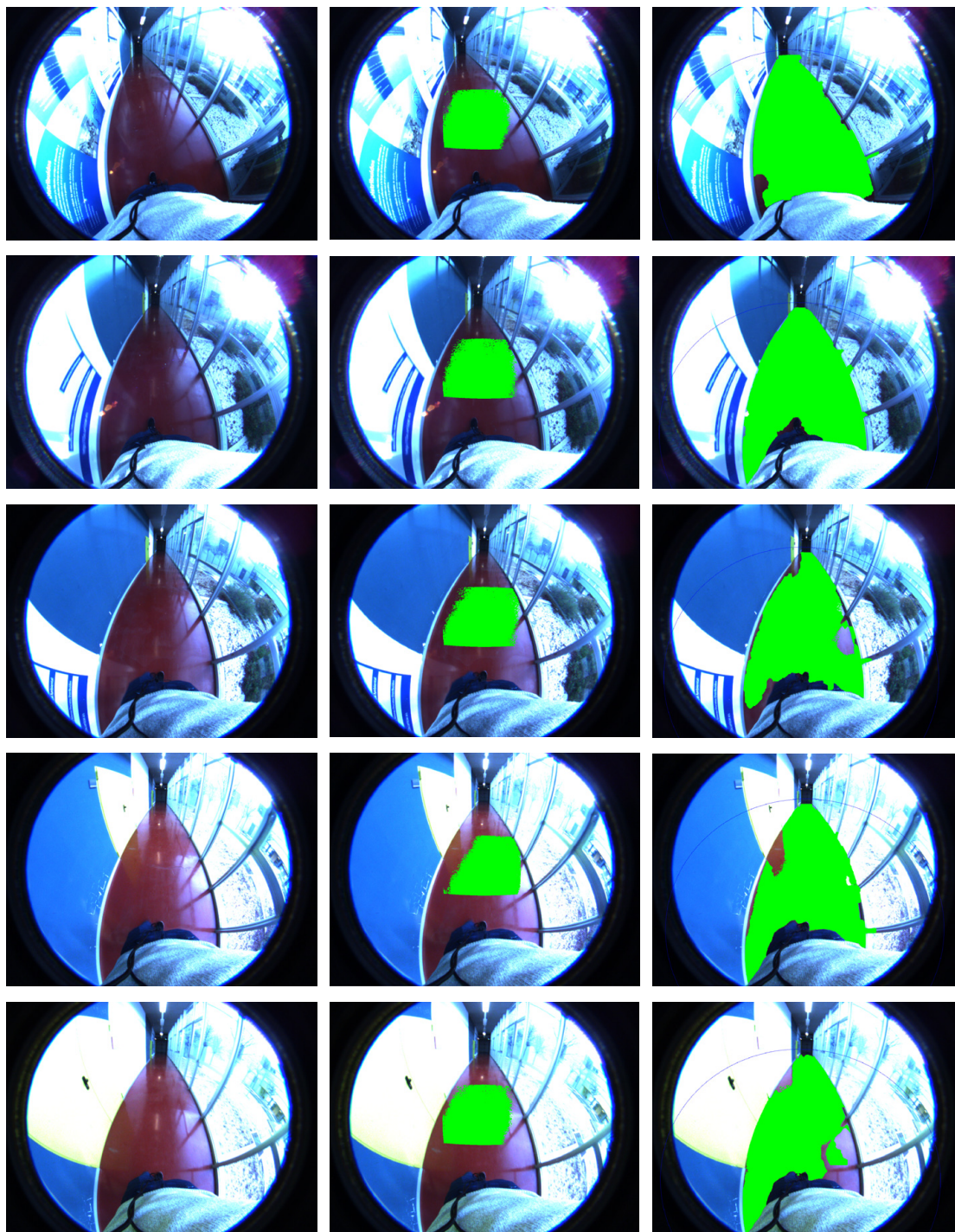


Figura B.1: Ejemplos experimentales obtenidos para el entorno interior con incidencia de luz solar

B.2. Resultados de entorno interior con gran incidencia de luz solar (I3A)



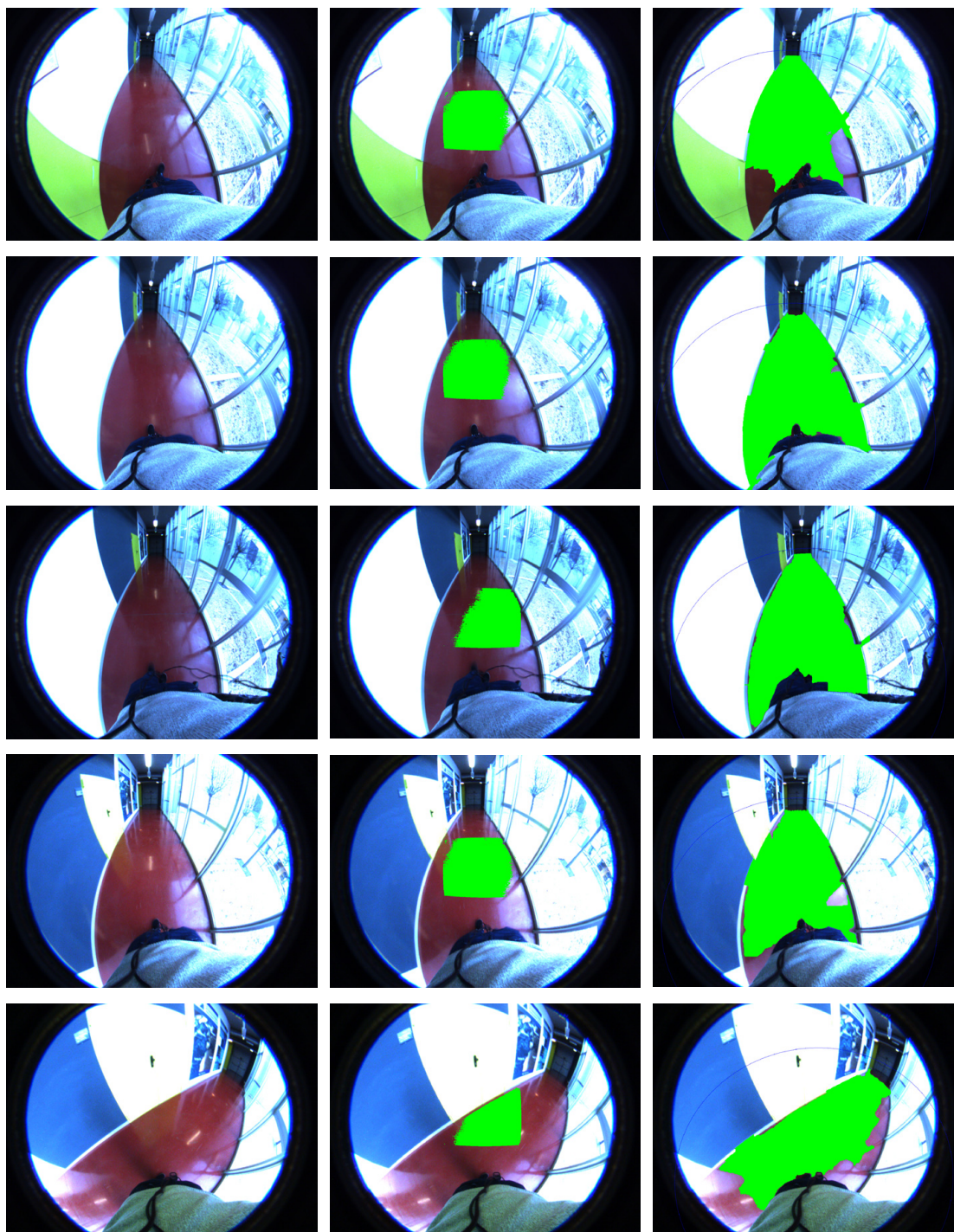


Figura B.2: Ejemplos experimentales obtenidos para el entorno interior con gran incidencia de luz solar

B.3. Resultados de entorno interior con incidencia de luz artificial (I3A)

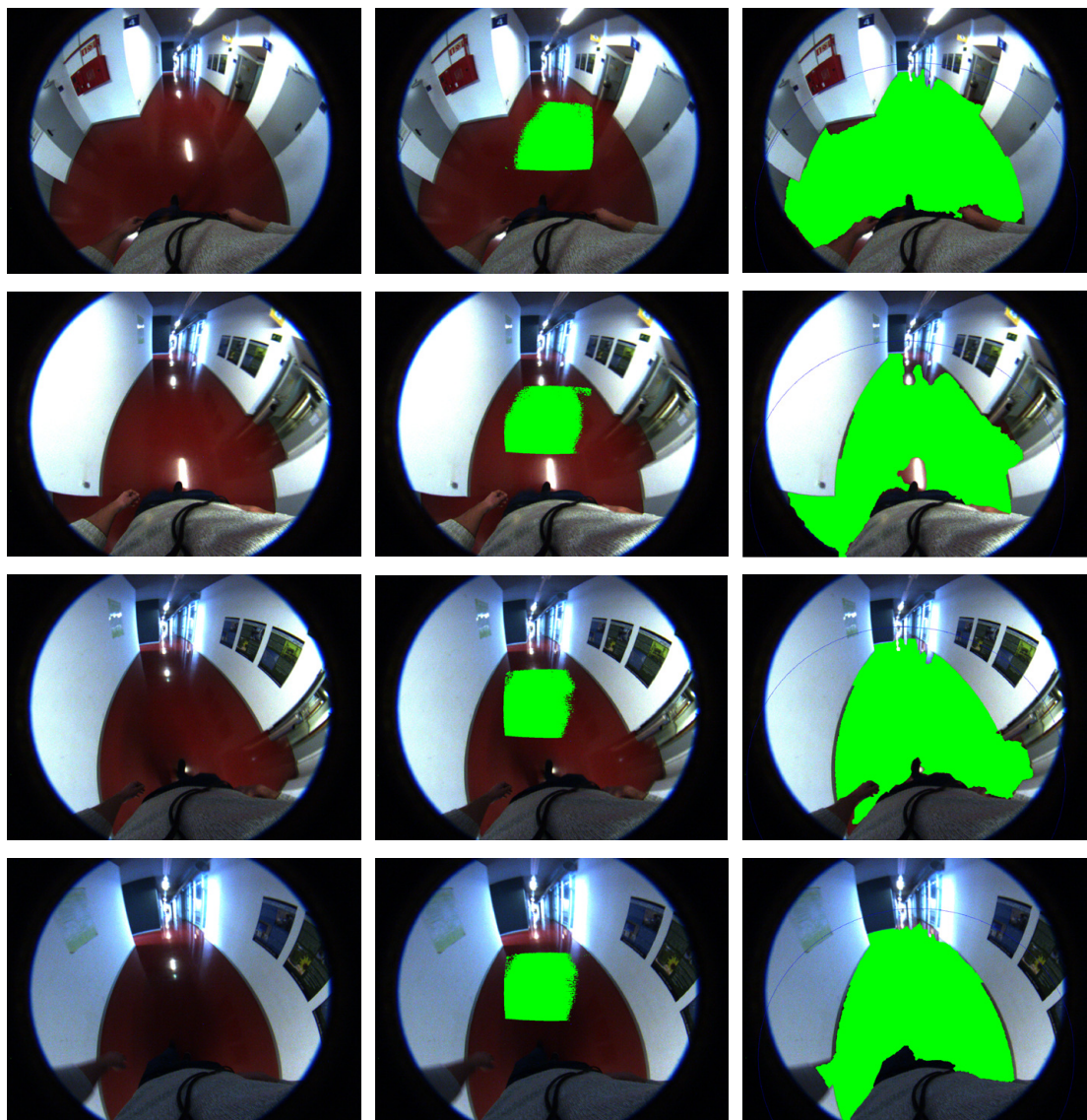


Figura B.3: Ejemplos experimentales obtenidos para el entorno interior con incidencia de luz artificial

B.4. Resultados de entorno interior con escasa iluminación (garaje)

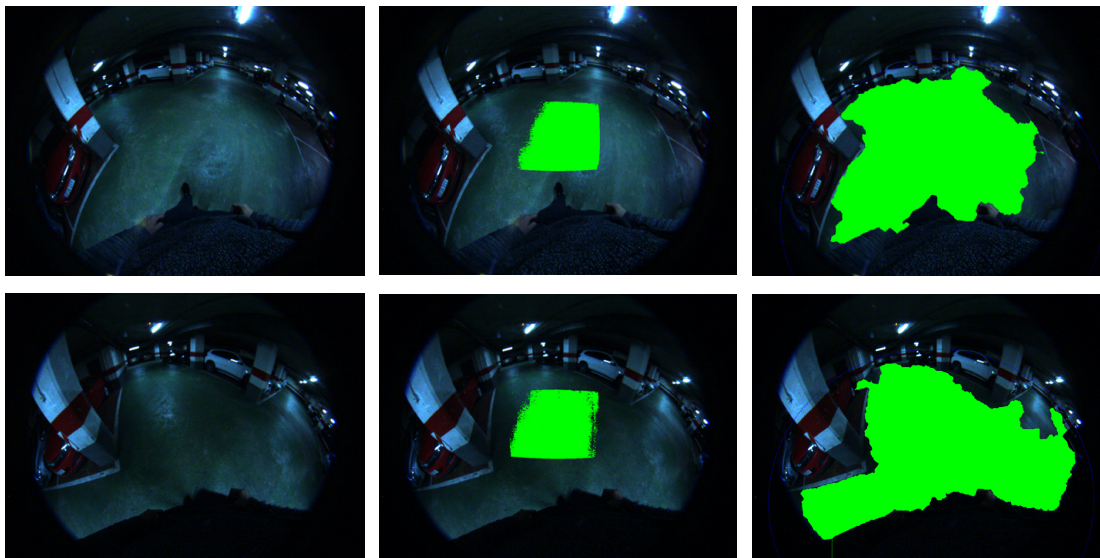


Figura B.4: Ejemplos experimentales obtenidos para el entorno interior escasa iluminación

B.5. Resultados de entorno exterior con terreno irregular (césped)

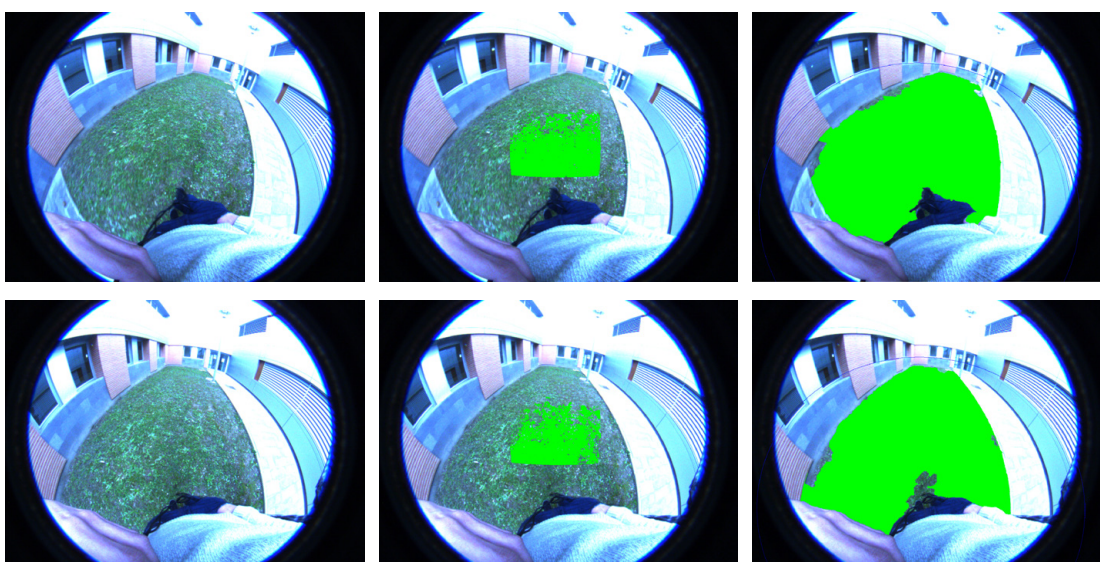


Figura B.5: Ejemplos experimentales obtenidos para el entorno exterior con terreno irregular

B.6. Resultados de entorno exterior con color homogéneo (exteriores Torres Quevedo)

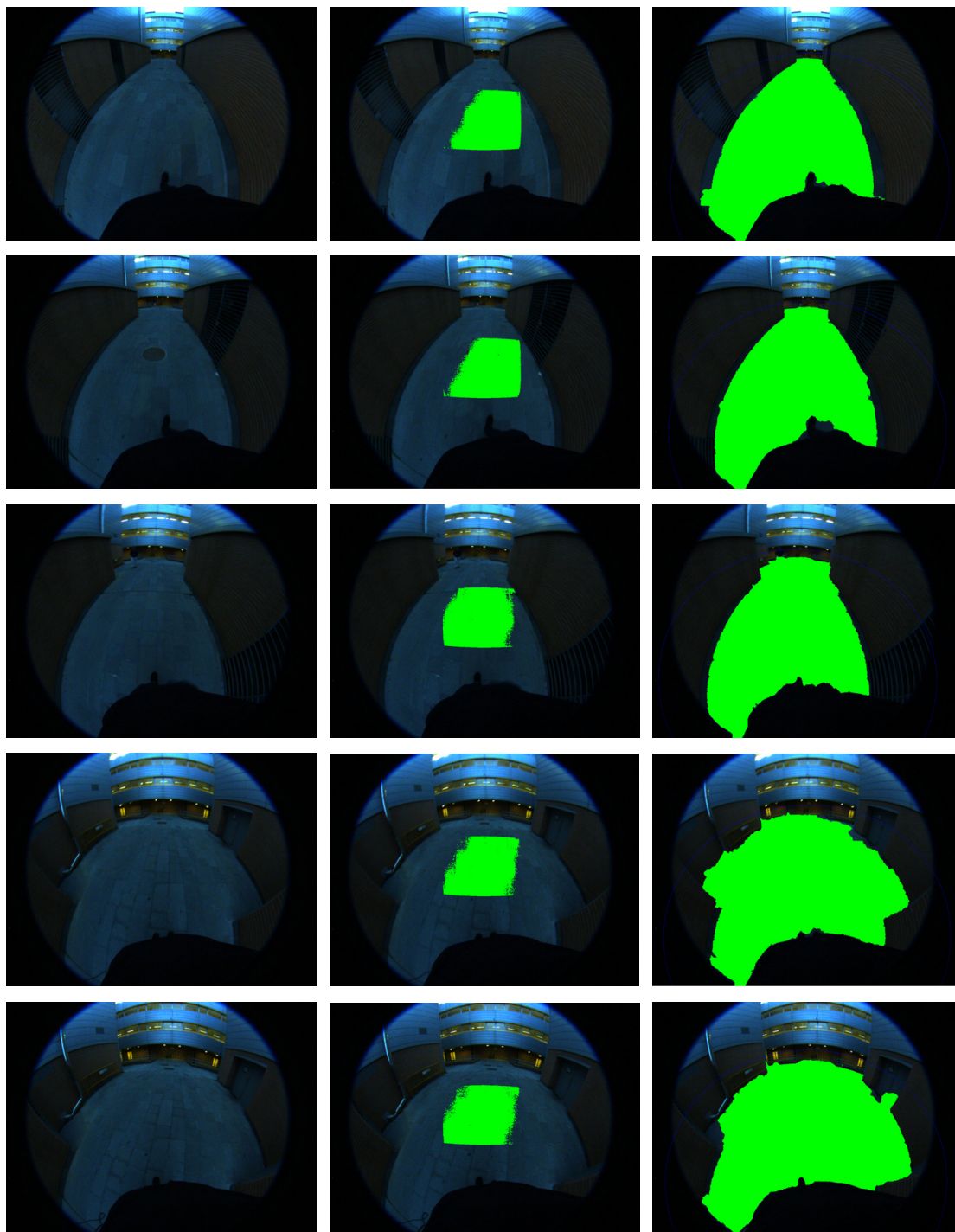


Figura B.6: Ejemplos experimentales obtenidos para el entorno exterior con color homogéneo

B.6. RESULTADOS DE ENTORNO EXTERIOR CON COLOR HOMOGÉNEO
(EXTERIORES TORRES QUEVEDO)

Bibliografía

- [1] ALADREN, A., LÓPEZ-NICOLÁS, G., PUIG, L., AND GUERRERO, J. J. Navigation assistance for the visually impaired using rgb-d sensor with range expansion. *IEEE Systems Journal* 10, 3 (2016), 922–932.
- [2] ALVAREZ, J. M., LOPEZ, A., AND BALDRICH, R. Illuminant-invariant model-based road segmentation. *Intelligent Vehicles Symposium, 2008 IEEE* (2008), 1175–1180.
- [3] BARNES, N. An overview of vision processing in implantable prosthetic vision. *Image Processing (ICIP), 2013 20th IEEE International Conference on* (2013), 1532–1535.
- [4] BERMUDEZ-CAMEO, J., BADIAS-HERBERA, A., GUERRERO-VIU, M., LOPEZ-NICOLAS, G., AND GUERRERO, J. Rgb-d computer vision techniques for simulated prosthetic vision. *Iberian Conference on Pattern Recognition and Image Analysis* (2017), 427–436.
- [5] BERMUDEZ-CAMEO, J., LOPEZ-NICOLAS, G., AND GUERRERO, J. Automatic line extraction in uncalibrated omnidirectional cameras with revolution symmetry. *International Journal of Computer Vision* 114, 1 (2015), 16–37.
- [6] BOURNE, R., FLAXMAN, S., BRAITHWAITE, T., CICINELLI, M., DAS, A., JONAS, J., KEEFFE, J., KEMPEN, J., LEASHER, J., AND LIMBURG, H. Magnitude, temporal trends, and projections of the global prevalence of blindness and distance and near vision impairment: a systematic review and meta-analysis. *The Lancet Global Health* 5, 9 (2017), e888–e897.
- [7] BRINDLEY, G., AND LEWIN, W. The sensations produced by electrical stimulation of the visual cortex. *The Journal of physiology* 196, 2 (1968), 479–493.
- [8] CANNY, J. A computational approach to edge detection. *IEEE Transactions on pattern analysis and machine intelligence* (1986), 679–698.

- [9] DAGNELIE, G. Visual prosthetics: physiology, bioengineering, rehabilitation. *Springer Science Business Media* (2011).
- [10] DAHLKAMP, H., KAEHLER, A., STAVENS, D., THRUN, S., AND BRADSKI, G. Self-supervised monocular road detection in desert terrain. *Proceedings of Robotics: Science and Systems* (August 2006).
- [11] DENNIS LUI, W., BROWNE, D., KLEEMAN, L., DRUMMOND, T., AND LI, W. Transformative reality: Improving bionic vision with robotic sensing. *Engineering in Medicine and Biology Society (EMBC), 2012 Annual International Conference of the IEEE* (2012), 304–307.
- [12] FISCHLER, M., AND BOLLES, R. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. In *Commun*, vol. 24. ACM, 1981, p. 381–395.
- [13] GUIMARAES, C. S. S., BAYAN HENRIQUES, R. V., AND PEREIRA, C. E. Analysis and design of an embedded system to aid the navigation of the visually impaired. *Biosignals and Biorobotics Conference (BRC), 2013 ISSNIP* (2013), 1–6.
- [14] LI, Y., AND BIRCHFIELD, S. T. Image-based segmentation of indoor corridor floors for a mobile robot. *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on* (2010), 837–843.
- [15] ÖKTEM, R., AYDIN, E., AND CAGILTAY, N. E. An indoor navigation aid designed for visually impaired people. *IECON 2008* (2008), 2982–2987.
- [16] PEASLEY, B., AND BIRCHFIELD, S. Real-time obstacle detection and avoidance in the presence of specular surfaces using an active 3d sensor. In *Robot Vision (WORV), 2013 IEEE Workshop on* (2013), pp. 197–202.
- [17] PEREZ-YUS, A., FERNANDEZ-MORAL, E., LOPEZ-NICOLAS, G., GUERRERO, J., AND RIVES, P. Extrinsic calibration of multiple rgb-d cameras from line observations. *IEEE Robotics and Automation Letters* 3, 1 (2018), 273–280.
- [18] RUSU, R., AND COUSINS, S. 3d is here: Point cloud library (pcl). *IEEE International Conference on Robotics and Automation (ICRA)* (2011), 1–4.
- [19] SCARAMUZZA, D., MARTINELLI, A., AND SIEGWART, R. A toolbox for easily calibrating omnidirectional cameras. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (2006), 5695–5701.

- [20] SCHAFER, H., HACH, A., PROETZSCH, M., AND BERNIS, K. 3d obstacle detection and avoidance in vegetated off-road terrain. In *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on* (2008), pp. 923–928.
- [21] UEDA, T., KAWATA, H., TOMIZAWA, T., OHYA, A., AND YUTA, S. Visual information assist system using 3d sokuiki sensor for blind people, system concept and object detecting experiments. *IEEE Conference on Industrial Electronics (IECON)* (2006), 3058–3063.
- [22] ULRICH, I., AND NOURBAKHSI, I. Appearance-based obstacle detection with monocular color vision. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence* (2000), pp. 866–871.
- [23] VAN DEN BERGH, M., BOIX, X., ROIG, G., AND VAN GOOL, L. Seeds: Superpixels extracted via energy-driven sampling. *International Journal of Computer Vision (IJCV)* (2014).
- [24] WANG, Z., LIU, H., WANG, X., AND QIAN, Y. Segment and label indoor scene based on rgb-d for the visually impaired. In *MultiMedia Modeling*. Springer, 2014, p. 449–460.
- [25] WONG, F., NAGARAJAN, R., AND YAACOB, S. Application of stereovision in a navigation aid for blind people. *Information, Communications and Signal Processing, 2003 and Fourth Pacific Rim Conference on Multimedia. Proceedings of the 2003 Joint Conference of the Fourth International Conference on 2* (2003), 734–737.
- [26] XU, J., FANG, Z. G., DONG, D. H., AND ZHOU, F. An outdoor navigation aid system for the visually impaired. In *Industrial Engineering and Engineering Management (IEEM), 2010 IEEE International Conference on* (2010), pp. 2435–2439.

Lista de Figuras

1.1. Ejemplo del suelo que necesita el sistema propuesto en [26]. Este tipo de suelo se ha ido incorporando en ciudades de Japón. De esta forma, las personas invidentes puedan caminar por dichas vías con su bastón de ciego con la certeza de que no existe ningún tipo de obstáculo en esa zona	4
1.2. Microsoft Kinect (izquierda) y Asus Xtion Pro Live (derecha)	5
1.3. Ejemplo del resultado obtenido en el trabajo de Aladren y col. [1]	6
1.4. Argus II Retinal Prosthesis. (a) El implante con sus principales componentes. (b) Equipamiento externo: gafas con cámara, antenas para la comunicación inalámbrica y la Unidad de Procesamiento de Video (VPU). (c) Imagen comercial de la visualización de una puerta por medio de fosfenos	8
2.1. Sistema de cámaras utilizado en este proyecto en el que se combinan la cámara fisheye y la cámara RGB-D	12
2.2. (a) Escena vista desde una cámara RGB-D convencional. (b) Misma escena vista desde la cámara fisheye. (c) Proyección de la nube de profundidad (en verde) sobre la imagen del fisheye una vez realizada la calibración	13
2.3. Muestra de algunas de las 21 imágenes tomadas mediante la cámara fisheye del patrón tipo ajedrez para realizar la calibración intrínseca de la misma	14

2.4.	Observación de líneas en una escena por una cámara RGB-D y una cámara convencional unidas por una estructura rígida y con campos de vista que no están solapados. Las correspondencias entre líneas se utilizan para formular restricciones geométricas para calcular la posición relativa (T) entre las cámaras. Imagen obtenida de [17]	16
2.5.	Esquema del sistema en una escena del mundo 3D y la correspondiente imagen fisheye	17
2.6.	Ejemplos de la extracción de líneas de la cámara fisheye: En rojo aparecen las fronteras obtenidas mediante el Canny tras la agrupación, y en verde las líneas obtenidas a partir de esas fronteras	19
2.7.	Lugar en el que se toma la secuencia necesaria para la calibración . . .	20
2.8.	Ejemplos para visualizar el funcionamiento del método de calibración .	21
3.1.	Principio de funcionamiento del filtro Voxelgrid. Todos los puntos que estén contenidos en un mismo vóxel, pasan a formar un único punto situado en el centroide de dicho vóxel	25
3.2.	Convenio de signos para el sensor de profundidad y para una escena genérica del mundo	25
3.3.	Posicionamiento del sistema de cámaras sobre el sujeto	26
3.4.	Ejemplo de segmentación del plano del suelo mediante el sensor de profundidad: (a) Imagen original. (b) Detección del plano del suelo mediante la nube filtrada. (c) Segmentación del plano del suelo final, utilizando la nube completa	28
4.1.	Ejemplo de la reproyección de los puntos 3D pertenecientes al plano del suelo, sobre la imagen del fisheye utilizando la look-up table	33
4.2.	Colores aditivos del modelo RGB	34
4.3.	Ejemplo de representación del modelo HSV	35

4.4.	Ejemplo de la evolución de las fronteras de los superpíxeles a lo largo de diferentes iteraciones del algoritmo SEEDS partiendo de una inicialización en 12 superpíxeles iguales. Imagen obtenida de [23]	36
4.5.	Ejemplo de segmentación de una imagen de fisheye mediante SEEDS .	36
4.6.	Ejemplo de las máscaras binarias utilizadas. (a) Máscara del suelo base. (b) Máscara de uno de los superpíxeles candidatos a ser semilla inicial. (c) Máscara restante tras ser solapadas las dos anteriores mediante, siendo esta máscara el área del superpíxel ocupada por el suelo base . .	38
4.7.	Ejemplo de creación de la semilla inicial del algoritmo (b) a partir del plano del suelo obtenido mediante la cámara de profundidad una vez se ha re proyectado sobre la imagen de fisheye (b)	39
4.8.	Máscara binaria de la semilla inicial de suelo obtenida en el apartado anterior	40
4.9.	Ejemplo de la segmentación del suelo. En cian, la semilla inicial. En rojo, la expansión de la semilla hasta completar la segmentación.	41
4.10.	Ejemplo de la representación de la línea del horizonte (azul oscuro), y su utilidad para la segmentación: (a) En este caso, no se ha tenido en cuenta la línea del horizonte, por lo que la expansión de suelo se propaga hasta el techo empeorando claramente la segmentación del suelo. (b) En este caso, el uso de la línea del horizonte hace que pese al error en la segmentación, éste no sea muy grave evitando su propagación hacia el techo	42
4.11.	Máscara binaria para eliminar los píxeles en negro de la segmentación del suelo de la imagen. Solo se tendrá en cuenta como posible suelo, la zona en blanco	43
5.1.	Ejemplo de la intersección entre el rayo proyectante y el plano del suelo	46
5.2.	Transformación de la referencia cámara a la referencia suelo	47
5.3.	(a) Resultado de la segmentación del suelo de la escena. (b) Máscara del suelo segmentado. (c) Máscara del suelo segmentado eliminando los puntos de zonas no transitables	48

5.4.	Discretización del espacio en sectores: máscara utilizada	49
5.5.	(a) Resultado de la segmentación del suelo de la escena. (b) Máscara de espacio libre de la escena en la navegación. (c) Espacio libre de la escena en 3D	49
5.6.	(a) Resultado de la segmentación del suelo de la escena. (b) Máscara de espacio libre de la escena en la navegación. (c) Espacio libre de la escena en 3D	51
5.7.	Ejemplos de posibles casos en los que establecer hipótesis de puertas . .	52
5.8.	Ejemplo de localización de la puerta central de la Figura 5.7b en el espacio 3D	53
5.9.	Ejemplo de la segmentación de las puertas mediante superpíxeles. Esta información puede ser utilizada para mejorar la hipótesis de puerta . .	54
6.1.	(a) Imagen de interior 1. (b) Detección de suelo mediante cámara de profundidad. (c) Segmentación mediante superpíxeles. (d) Resultado final de la detección de suelo. (e) Proyección 3D de la detección de suelo.	58
6.2.	(a) Imagen de interior 2. (b) Detección de suelo mediante cámara de profundidad. (c) Segmentación mediante superpíxeles. (d) Resultado final de la detección de suelo. (e) Proyección 3D de la detección de suelo.	59
6.3.	(a) Imagen de interior 3. (b) Detección de suelo mediante cámara de profundidad. (c) Segmentación mediante superpíxeles. (d) Resultado final de la detección de suelo. (e) Proyección 3D de la detección de suelo.	61
6.4.	(a) Imagen de interior 4. (b) Detección de suelo mediante cámara de profundidad. (c) Segmentación mediante superpíxeles. (d) Resultado final de la detección de suelo. (e) Proyección 3D de la detección de suelo.	62
6.5.	(a) Imagen de interior 5. (b) Detección de suelo mediante cámara de profundidad. (c) Segmentación mediante superpíxeles. (d) Resultado final de la detección de suelo. (e) Proyección 3D de la detección de suelo.	64

6.6. (a) Imagen de interior 6. (b) Detección de suelo mediante cámara de profundidad. (c) Segmentación mediante superpíxeles. (d) Resultado final de la detección de suelo. (e) Proyección 3D de la detección de suelo.	65
6.7. (a) Imagen de interior 7. (b) Detección de suelo mediante cámara de profundidad. (c) Segmentación mediante superpíxeles. (d) Resultado final de la detección de suelo. (e) Proyección 3D de la detección de suelo.	67
6.8. (a) Imagen de interior 8. (b) Detección de suelo mediante cámara de profundidad. (c) Segmentación mediante superpíxeles. (d) Resultado final de la detección de suelo. (e) Proyección 3D de la detección de suelo.	68
6.9. (a) Imagen de exterior 1. (b) Detección de suelo mediante cámara de profundidad. (c) Segmentación mediante superpíxeles. (d) Resultado final de la detección de suelo. (e) Proyección 3D de la detección de suelo.	70
6.10. (a) Imagen de exterior 2. (b) Detección de suelo mediante cámara de profundidad. (c) Segmentación mediante superpíxeles. (d) Resultado final de la detección de suelo. (e) Proyección 3D de la detección de suelo.	71
6.11. (a) Imagen de exterior 3. (b) Detección de suelo mediante cámara de profundidad. (c) Segmentación mediante superpíxeles. (d) Resultado final de la detección de suelo. (e) Proyección 3D de la detección de suelo.	73
6.12. (a) Imagen de exterior 4. (b) Detección de suelo mediante cámara de profundidad. (c) Segmentación mediante superpíxeles. (d) Resultado final de la detección de suelo. (e) Proyección 3D de la detección de suelo.	74
B.1. Ejemplos experimentales obtenidos para el entorno interior con incidencia de luz solar	86
B.2. Ejemplos experimentales obtenidos para el entorno interior con gran incidencia de luz solar	88
B.3. Ejemplos experimentales obtenidos para el entorno interior con incidencia de luz artificial	89
B.4. Ejemplos experimentales obtenidos para el entorno interior escasa iluminación	90

B.5. Ejemplos experimentales obtenidos para el entorno exterior con terreno irregular	90
B.6. Ejemplos experimentales obtenidos para el entorno exterior con color homogéneo	91

Lista de Tablas

5.1. Instrucciones de audio en función del sector angular	50
5.2. Instrucciones de audio en función de la distancia. El número indica el mínimo espacio libre disponible en metros	50
6.1. Resultados medios de evaluación de un entorno de interior con incidencia de luz solar	60
6.2. Resultados medios de evaluación de un entorno de interior con gran incidencia de luz solar	63
6.3. Resultados medios de evaluación de un entorno de interior con incidencia de luz artificial	66
6.4. Resultados medios de evaluación de un entorno de interior con escasa iluminación	69
6.5. Resultados medios de evaluación de un entorno de exterior con color homogéneo	72
6.6. Resultados medios de evaluación de un entorno con terreno irregular . .	75
6.7. Resultados medios de evaluación de entornos interiores	76
6.8. Resultados medios de evaluación de entornos exteriores	76