



Universidad
Zaragoza

Trabajo Fin de Grado

Detección de puertas mediante un sistema de visión
omnidireccional y de profundidad

Autor

Alejandro de Nova Guerrero

Director

José Jesús Guerrero Campo

Grado en Ingeniería de Tecnologías Industriales
Escuela de Ingeniería y Arquitectura

Junio 2019

RESUMEN

El objetivo de este proyecto es el desarrollo de un algoritmo que permita la detección de puertas, de forma que sirva para la planificación de trayectorias de personas con deficiencias visuales para ayudarlas a moverse por entornos desconocidos.

Para realizar esta tarea se utilizará la información obtenida a partir de dos sensores, una cámara RGB-D y una cámara fisheye, a través de las cuales se obtiene tanto información 3D de una pequeña región de la escena como información de color de una amplia región que sirva como complemento. Esta combinación de sensores se encuentra a la altura del pecho de los usuarios, ya que es una posición discreta y estable para las cámaras, a la vez que permite captar imágenes desde un ángulo que facilita la navegación.

La información recogida se procesará mediante el uso de algoritmos de procesamiento de información 3D y de color, a través de las librerías PCL y OpenCV. Además, se utilizarán los resultados obtenidos de un proyecto previo, de forma que es la base principal sobre la que se apoya este trabajo.

Finalmente se obtiene un algoritmo con una tasa de falsos positivos prácticamente nula y una capacidad de detección de puertas de casi el 50%.

Tabla de contenidos

1. Introducción	1
1.1. Motivación	1
1.2. Estado del arte	2
1.3. Objetivos y alcance	3
2. Obtención de información inicial de la escena	5
2.1. Extracción de las rectas verticales a partir de la escena	5
2.2. Búsqueda de la intersección de las rectas verticales con la línea del horizonte	7
2.3. Búsqueda de la intersección de las rectas verticales con el plano del suelo	8
3. Planteamiento de hipótesis de puerta a partir de dos rectas verticales	11
3.1. Selección de las parejas de rectas que forman cada hipótesis	11
3.2. Segmentación de las posibles puertas	12
3.2.1. Creación de la semilla inicial	13
3.2.2. Expansión de la semilla inicial	13
3.3. Eliminación de falsos positivos	15
3.3.1. Cross ratio	15
3.3.2. Uso de fronteras para evitar cuadros	18
4. Planteamiento de hipótesis de puerta a partir de una recta vertical	20
4.1. Segmentación de las posibles puertas	20
4.2. Filtrado de falsos positivos	23
4.2.1. Bounding box	24
5. Evaluación de resultados	26
5.1. Caso 1	27
5.2. Caso 2	29
5.3. Caso 3	31
5.4. Caso 4	33

5.5. Caso 5	35
5.6. Conclusiones de la evaluación.....	37
6. Conclusiones y trabajo futuro	40
Bibliografía.....	41
Anexos	42
A. Resumen del trabajo previo	43
A.1. Sensores utilizados.....	43
A.2. Calibración de las cámaras	44
A.3. Detección del suelo mediante la cámara de profundidad	45
A.4. Expansión del suelo	46
A.5. Uso de la información 3D en la expansión del suelo	48
B. Software, hardware y recursos externos utilizados y cronograma del proyecto	50
Lista de figuras	52

Capítulo 1

Introducción

1.1. Motivación

A lo largo de los años la deficiencia visual en los humanos ha sido uno de los principales factores que condicionan la vida diaria de las personas, pues la vista es el sentido que aporta mayor información del entorno. Según la OMS en 2018 el número de personas con algún tipo de deficiencia visual grave era 217 millones, mientras que el número de personas que padecían ceguera aumentaba hasta 36 millones. Además, puesto que este tipo de enfermedades son más comunes en personas de avanzada edad, y debido al envejecimiento de la población mundial, se puede suponer que el número de personas afectadas irá en aumento con el paso del tiempo.

Si bien existen diferentes herramientas para la asistencia de estas personas como puede ser el uso de bastones o los perros guía, el avance tecnológico junto con la visión por computador puede suponer una gran ayuda en la vida diaria de aquellos que sufren algún tipo de deficiencia visual.

Se busca un sistema que pueda ser transportado por una persona, de forma que la combinación de dos sensores y el uso de algoritmos permitan el reconocimiento de puertas en las imágenes tomadas, continuando directamente un Trabajo de Fin Máster previo [6] en el que utilizando el mismo sistema de cámaras se lograba detectar el espacio libre de obstáculos de la escena.

Se busca que este algoritmo de detección de puertas pueda implementarse en un sistema de navegación para así poder ayudar a estas personas a desplazarse por entornos desconocidos. Además, se podría llegar a implantar este sistema en una prótesis visual, siempre que los avances tecnológicos permitan la integración de los sensores necesarios, así como la transmisión de la información, pues la detección de objetos mediante visión por computador es un problema estudiado desde hace años, pero que por desgracia presenta una clara dependencia de los sensores utilizados en la toma de imágenes, así como de la naturaleza de las imágenes, siendo la iluminación uno de los factores más influyentes a la hora de procesar la información.

Por otro lado, no sólo se debe asociar la información acerca de las puertas del entorno a la asistencia en el guiado de personas sino que también es posible integrar esta información en el sistema de navegación autónomo de los robots por distintos entornos, siendo este último el ámbito en el que más se han centrado los estudios de los últimos años.

El proyecto se sitúa en el área de Ingeniería de Sistemas y Automática de la Universidad de Zaragoza, haciendo uso de la visión por ordenador. El departamento ha puesto en disposición todo el material requerido, como cámaras, ordenadores y funciones ya desarrolladas para el tratamiento de las imágenes.

1.2. Estado del arte

Si bien hay numerosos estudios acerca de la detección de puertas mediante el uso de la información obtenida a partir de distintos sensores, la mayoría de estos proyectos están orientados a permitir una planificación de ruta de robots o al ámbito de la robótica en general mediante el uso de cámaras RGB monoculares o sensores RGB-D. A continuación se detallarán algunos de estos estudios.

En el sistema desarrollado por Zhou y col. [4] se hace uso de un sensor Kinect, que se corresponde con un sensor RGB-D que capta información 3D y de color de una pequeña región, a través del cual se captan imágenes de una escena de interiores, de forma que mediante el procesamiento de la información de profundidad de la escena se realiza la detección de puertas a partir de sus contornos verticales.

De manera análoga al caso anterior, el método de Han-Yuan y col. [5] emplea un sensor Kinect para obtener las imágenes en interiores, aunque a diferencia del modelo descrito en [4] el algoritmo de detección de puertas utiliza tanto la información de profundidad como la información de color que suministra el sensor utilizado.

Existen otros dispositivos utilizados para la detección de puertas, como el uso de sensores láser. Es el caso del trabajo desarrollado por Fernández-Caramés y col. [7] donde se hace uso de un sensor láser y una cámara RGB-D para la obtención de información 3D. A partir de esta información se obtienen los bordes de las posibles puertas, mientras que gracias al sensor láser se obtiene la anchura entre parejas de bordes, que servirá para definir lo que finalmente es puerta.

En el trabajo desarrollado por Murillo y col. [1], a diferencia de los casos anteriores sólo se utiliza la información de la propia imagen. Además, el proceso de detección de puertas se realiza teniendo en cuenta tanto la forma como el color de las puertas, de forma que tras un aprendizaje por parte del algoritmo a partir de un conjunto de imágenes de prueba es capaz de plantear hipótesis de puerta y decidir si realmente lo son o no.

Si bien los sistemas anteriores trataban la detección de puertas en interiores para el guiado de robots, el sistema desarrollado por Talebi y col. [3] se basa en la detección de puertas de edificios mediante el uso de información 3D y de color. Cabe destacar la importancia de este sistema, pues

en exteriores se producen mayores y más continuos cambios en la iluminación, dificultando el procesado de la información.

En general, los sistemas desarrollados utilizan sensores RGB o RGB-D, que obtienen información de color e información 3D de la escena. Sin embargo, presentan un problema, y es su reducido campo de visión de pocos metros, por lo que la información obtenida no es suficiente como para realizar una planificación de la trayectoria a seguir por el usuario durante la navegación, por lo que en este proyecto también se utiliza una cámara de gran campo de visión como complemento que permita un mejor tratamiento de las imágenes.

1.3. Objetivos y alcance

El objetivo principal es desarrollar un sistema de visión basado en dos sensores, una cámara de profundidad que aporta información con un alto grado de fiabilidad de una pequeña región de la escena y una cámara de color con un gran campo de visión que aporte información adicional del resto de la escena, de forma que sea posible la detección de puertas en escenarios de interiores a una distancia suficiente como plantear trayectorias a seguir por los usuarios. Además, se quiere demostrar que la combinación de sensores elegidos puede aportar ventajas a la detección de puertas respecto a los sistemas desarrollados donde sólo se emplean sensores de profundidad.

Por otro lado, se busca desarrollar un algoritmo robusto que permita un elevado grado de efectividad en la detección de puertas con un tiempo de cómputo razonable, así como una tasa de falsos positivos lo más baja posible, ya que se plantea que este algoritmo sea implementado en un sistema de guiado de personas con discapacidad visual en tiempo real.

Para cumplir los objetivos planteados se ha dividido el proyecto en tres fases:

1. Estudio de las herramientas necesarias:
 - Librerías para el tratamiento de la información 3D, como son las librerías PCL.
 - Librerías para el tratamiento de las imágenes, en este caso OpenCV.
 - Estudios, artículos y proyectos anteriores para obtener una base de conocimiento inicial sobre la visión por computador.
 - Comprensión y estudio del lenguaje de programación a utilizar, en este caso C++.
2. Implementación del algoritmo de detección de puertas:
 - En primer lugar se obtendrá la información más relevante de la escena, como pueden ser las líneas verticales o la intersección de estas con el suelo.
 - Se desarrollará el primer método de detección de puertas, consistente en hipótesis de puerta a partir de parejas de rectas verticales.

- Por último se desarrollará el segundo método de detección de puertas, esta vez partiendo de una única recta vertical.
3. Evaluación de resultados y conclusiones.

Capítulo 2

Obtención de información inicial de la escena

Previo a la explicación de cómo se ha desarrollado el proyecto es necesario comprender la base en la que se fundamenta, por lo que en el Anexo A se puede encontrar un resumen del trabajo desarrollado por Guerrero-Viu [6], cuyos resultados son pilar fundamental del proyecto actual.

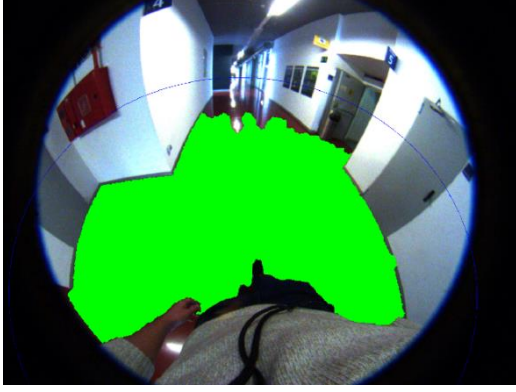
En este anexo se detalla la elección de los sensores utilizados, así como la posición de las cámaras sobre el usuario y el procedimiento realizado para segmentar el espacio libre de obstáculos de la escena.

Tras esto, se pasará a detallar el método de detección de puertas, que se basará principalmente en utilizar rectas extraídas de la imagen.

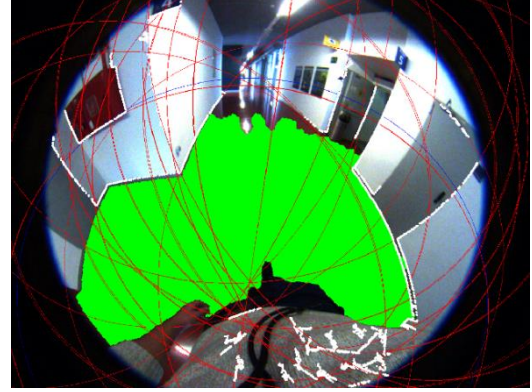
Para llevar a cabo esto es necesario, en primer lugar, obtener cierta información de la imagen. Esta información se corresponde con las rectas verticales obtenidas mediante el algoritmo descrito en [2], así como los puntos de intersección de cada una de estas rectas con la línea del horizonte y con el plano del suelo.

2.1. Extracción de las rectas verticales a partir de la escena

Tras aplicar este algoritmo se obtienen tanto el vector que describe cada una de las rectas detectadas como los puntos que conforman las fronteras a partir de los que se obtienen las rectas. En la Figura 2.1a se puede observar el resultado tras aplicar este algoritmo.



(a)



(b)

Figura 0.1. (a) Imagen inicial. (b) Resultado tras la detección de rectas (en rojo) y puntos de frontera (en blanco)

Para el caso de detección de puertas sólo se utilizarán las rectas verticales, ya que en general con el algoritmo de extracción de líneas se obtienen muchas más rectas verticales que horizontales. Por este motivo se ha realizado un filtrado de las rectas detectadas mediante la ayuda de la línea del horizonte.

La línea del horizonte es perpendicular a las rectas verticales, de modo que mediante un producto escalar de los vectores que definen tanto la línea del horizonte como el resto de rectas se puede conocer la verticalidad de cada recta, según la siguiente expresión:

$$\vec{u} \cdot \vec{v} = u_1 \cdot v_1 + u_2 \cdot v_2 + u_3 \cdot v_3 \quad (2.1)$$

Dónde (u_1, u_2, u_3) es el vector director de la línea del horizonte y (v_1, v_2, v_3) el vector director de cada una de las rectas detectadas.

En el caso del producto escalar el resultado debe ser igual a cero para que ambas rectas sean perpendiculares, sin embargo, como las rectas se han obtenido de forma numérica mediante un algoritmo que emplea umbrales, las rectas no van a ser perfectamente verticales. Así pues, se ha utilizado un pequeño umbral a partir del cual se clasificará cada recta como vertical o no, de forma que en la Figura 2.2 se puede observar el resultado tras el filtrado.

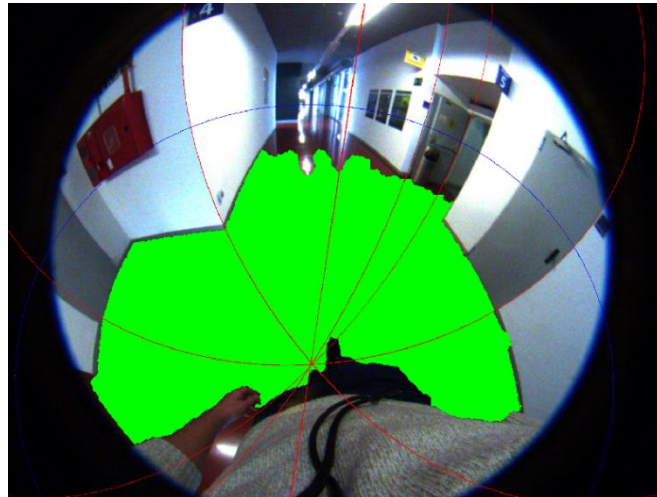


Figura 0.2. Rectas verticales tras el filtrado de la Figura 2.1.

2.2. Búsqueda de la intersección de las rectas verticales con la línea del horizonte

Una vez se tienen las rectas verticales que se utilizarán se pasará a buscar el punto de intersección de cada una de las rectas verticales con la línea del horizonte, lo cual se reduce a un producto vectorial en la geometría proyectiva, pues los puntos en el plano imagen se corresponden con la intersección de un rayo proyectante desde el centro óptico de la cámara con el propio plano. De esta forma, al conocer los vectores que definen las rectas verticales y la línea del horizonte, el producto vectorial de ambas dará como resultado el rayo proyectante al que pertenece el punto buscado. Por último, haciendo uso de la calibración intrínseca del fisheye vista en el Anexo A se obtiene cada una de las intersecciones en el plano imagen.

Pese a que esta operación resulta sencilla hay que tener en cuenta que debido al elevado campo de visión del fisheye (182°) determinadas rectas verticales pueden presentar intersección con la línea del horizonte en dos puntos distintos dentro de la imagen, correspondientes a la zona ‘delantera’ y ‘trasera’ de la cámara, respectivamente. Además, el producto vectorial de dos rectas tiene dos soluciones, ambas de mismo módulo y dirección pero de sentido contrario, dando lugar a los dos puntos de intersección para cada recta.

Lo habitual es que en la imagen la mayoría de rectas solo tengan una intersección con la línea del horizonte, ya que la otra aparecería fuera de los límites de la imagen, sin embargo, en los casos en los que aparezcan las dos intersecciones en la imagen será necesario seleccionar el punto adecuado. Para esto se ha tenido en cuenta la información de los puntos que forman fronteras, la cual se ha obtenido del algoritmo de extracción de rectas utilizado.

Como criterio de selección de la intersección correcta en el caso de que existan dos intersecciones en la imagen se han realizado los dos posibles productos vectoriales, obteniendo para cada resultado el punto en la imagen y se ha tenido en cuenta la cercanía de este punto a los puntos que forman la frontera a partir de la cual se extrajo la recta de la cual se quiere conocer la intersección. De esta forma, aquella intersección que esté a una distancia menor que un umbral determinado de algún punto de frontera de la recta a la que pertenece dicho punto será elegida como el punto de intersección buscado. En la Figura 2.3 se pueden observar ejemplos del resultado de obtención de estos puntos de intersección.

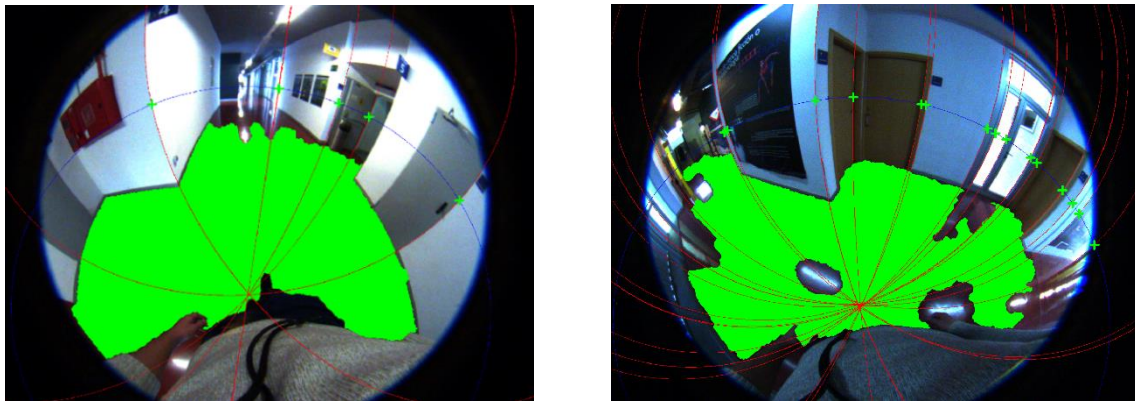


Figura 0.3. Ejemplos de imágenes donde se ha representado la intersección de cada una de las rectas verticales con la línea del horizonte

2.3. Búsqueda de la intersección de las rectas verticales con el plano del suelo

Una vez obtenidos los puntos de intersección de las rectas verticales con la línea del horizonte el siguiente punto de interés a obtener es la intersección de cada una de las rectas con el plano de suelo, pues estos puntos serán los que permitan plantear las primeras hipótesis de puerta más adelante.

Para encontrar estas intersecciones se va a utilizar la información que se tiene sobre el plano imagen, es decir, se trabajará con píxeles. Para ello se buscará para cada recta vertical la posición en la imagen en la que corte con la región del suelo expandido, que también se conoce.

Este método, sin embargo, presente un problema principalmente, pues cada recta corta numerosas veces a la región del suelo, por lo que se procesará la imagen para obtener únicamente el contorno del suelo expandido.

En primer lugar, se realiza una máscara binaria de la región de suelo (Figura 2.4a), la cual presenta niveles de gris de valor 1 (blanco) en los píxeles correspondientes al suelo, mientras que para el resto de píxeles el nivel de gris es 0 (negro).

La máscara binaria se expande mediante una función morfológica llamada dilatación que consiste en la convolución de la imagen con un elemento estructural (kernel) cuya forma es un círculo en este caso. Este kernel presenta un punto de anclaje justo en su centro, de forma que conforme el kernel recorre la imagen se calcula el valor máximo de los píxeles solapados por él y se sustituye el nivel de gris del píxel de la imagen en la posición del punto de anclaje por el valor máximo que se ha calculado. De esta forma se consigue ‘expandir’ la región del suelo.

Una vez se ha dilatado la máscara binaria se realiza una resta entre la máscara inicial y la máscara expandida, obteniendo una imagen en la que sólo se representa el contorno del suelo (Figura 2.4b).

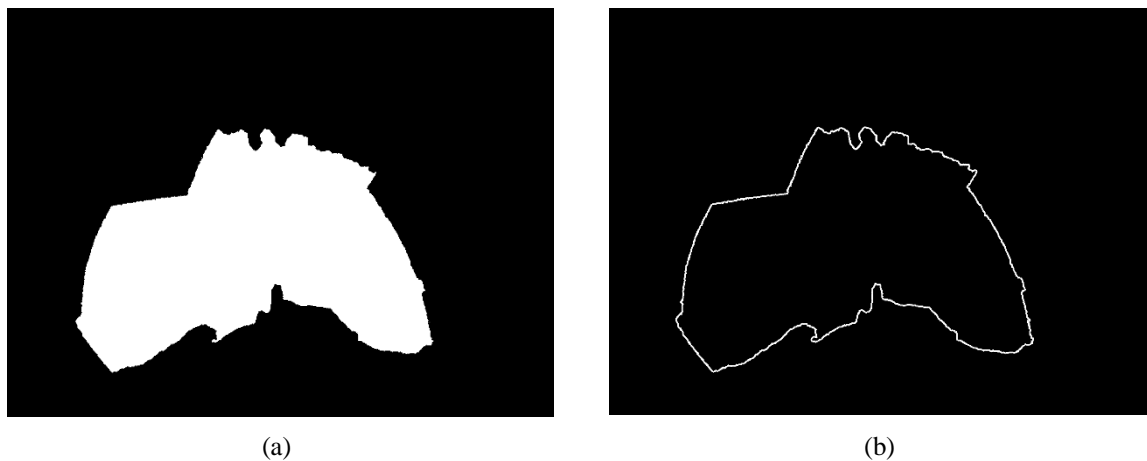
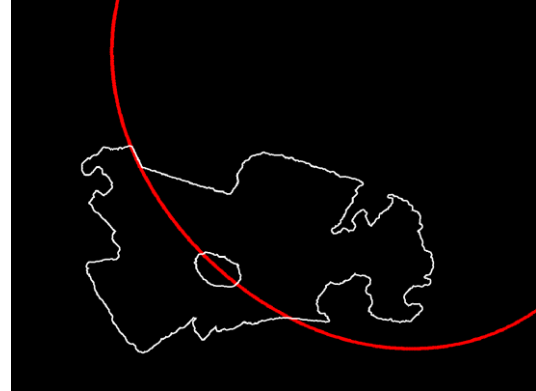


Figura 0.4. (a) Máscara binaria del suelo. (b) Máscara binaria del contorno del suelo

Pese a haber reducido los puntos de contacto entre las rectas y el suelo, cada una de las rectas tendrá generalmente dos puntos de intersección con el suelo debido a la naturaleza del fisheye, como muestra la Figura 2.5a. También se puede dar el caso en el que alguna línea corte más de dos veces con el suelo debido a irregularidades del mismo, pues la segmentación del suelo no es perfecta, como se aprecia en la Figura 2.5b.



(a)



(b)

Figura 0.5. (a) Ejemplo de recta vertical (en rojo) que corta en dos puntos a la máscara del contorno del suelo. (b) Ejemplo de recta vertical (en rojo) que corta en más de dos puntos a la máscara del contorno del suelo

Para decidir el punto de corte adecuado se utilizará la información proporcionada por la intersección de cada recta con la línea del horizonte. De esta forma se evaluarán todos los puntos de corte de cada recta vertical con la máscara binaria del contorno del suelo y se elegirá como la intersección adecuada aquel punto que se encuentre más cercano a la intersección de la línea vertical correspondiente con la línea del horizonte. En la Figura 2.6 se puede observar el resultado de aplicar este algoritmo en algunas imágenes.

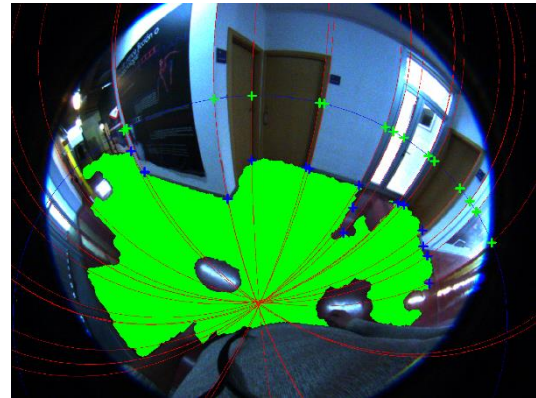
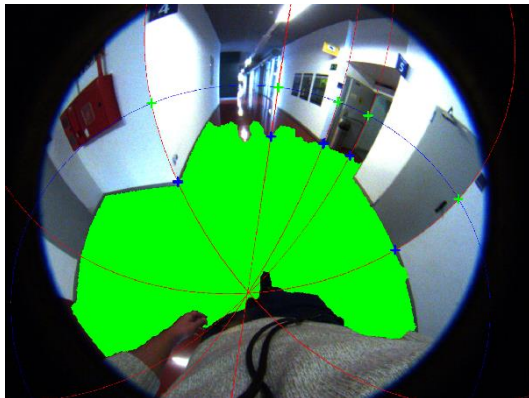


Figura 0.6. Ejemplos de imágenes donde se ha hallado la intersección (en azul) de cada línea vertical con el suelo

Capítulo 3

Planteamiento de hipótesis de puerta a partir de dos rectas verticales

Las puertas se caracterizan por presentar una forma rectangular definida por un marco horizontal y dos marcos verticales, de forma que se aprovecharán las rectas verticales de la imagen para plantear hipótesis de puerta.

En la actualidad las puertas que se comercializan presentan unas medidas normalizadas, siendo su anchura de 62,5 centímetros, 72,5 centímetros u 82,5 centímetros, mientras que su altura es de 2,03 metros. Aprovechando esta normalización se pueden suponer hipótesis de puerta sobre la imagen midiendo la distancia entre rectas verticales y comprobando si cumplen estas medidas.

3.1. Selección de las parejas de rectas que forman cada hipótesis

Para obtener la distancia entre las distintas rectas se utilizarán los puntos de intersección con el plano del suelo, sin embargo, estos puntos se encuentran sobre el plano imagen por lo que será necesario reproyectarlos al plano 3D del suelo para así poder realizar las medidas en metros y compararlas con las cotas normalizadas de las puertas reales.

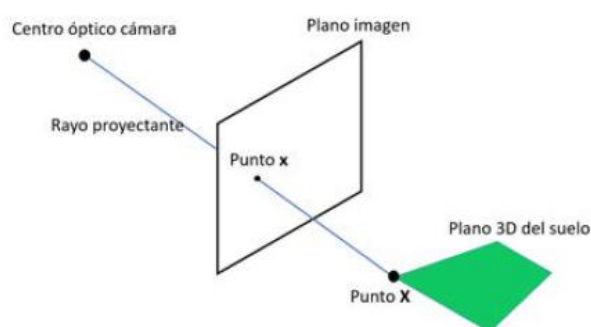


Figura 0.1. Intersección de un rayo proyectante con el plano del suelo. Imagen obtenida de [6]

En geometría descriptiva estos puntos 3D (punto **X** de la Figura 3.1) se corresponden con la intersección de un rayo proyectante con origen en el centro óptico de la cámara con el plano 3D del suelo, formando la proyección de ese mismo punto en el plano imagen (punto **x**). De esta forma, a partir de las intersecciones con el suelo en el plano imagen se obtiene cada uno de los rayos proyectantes a través de la calibración intrínseca de la cámara fisheye. Por último se realiza

la intersección de cada rayo con el plano 3D del suelo, que se corresponde con el plano del suelo obtenido por la cámara de profundidad.

Si tomamos cada rayo proyectante con la siguiente forma:

$$(x - x_0, y - y_0, z - z_0) = \lambda(a, b, c) \quad (3.1)$$

Dónde (a, b, c) es el vector director del rayo, (x_0, y_0, z_0) un punto cualquiera perteneciente al rayo y (x, y, z) el punto de intersección con el suelo 3D, si se sustituye en (A.1) se obtiene la siguiente expresión:

$$\lambda = -\frac{Ax_0 + By_0 + Cz_0 + D}{Aa + Bb + Cc} \quad (3.2)$$

De esta ecuación se conocen todos los parámetros excepto (x_0, y_0, z_0) , sin embargo, al ser un punto perteneciente al rayo proyectante, el cuál parte del centro óptico de la cámara (origen de coordenadas), se puede sustituir por el valor del origen de coordenadas, es decir $O(0, 0, 0)$. De esta forma se puede despejar el parámetro λ y con él obtener cada punto de intersección en el plano 3D del suelo.

A la hora de plantear las hipótesis de puerta se podría medir la distancia de cada recta con todas las demás, sin embargo, esto conllevaría un aumento del tiempo de cómputo y además carecería de sentido desde el punto de vista de la realidad, ya que cada puerta está definida por dos rectas consecutivas. Por esta razón se ha decidido medir sólo la distancia entre los puntos de intersección con el suelo 3D consecutivos.

Por otro lado, aunque las puertas presentan cotas normalizadas hay que tener en cuenta que las rectas extraídas pueden pertenecer al exterior de los marcos por lo que el umbral a utilizar debe ser algo mayor que la máxima anchura posible de la puerta, siendo en este caso el umbral máximo 1,05 metros y el umbral mínimo 0,6 metros.

3.2. Segmentación de las posibles puertas

Una vez se obtienen las parejas de rectas que podrían formar una puerta es necesario segmentar la región que representa la puerta en la imagen. Para ello, se va a utilizar una semilla inicial que se expandirá a zonas contiguas y similares de la imagen, de forma que mediante sucesivas iteraciones se segmente finalmente cada posible puerta.

Este método de expansión se implementará aprovechando la segmentación inicial de la imagen utilizando el método de los superpíxeles visto en el Anexo A.

3.2.1. Creación de la semilla inicial

Para definir la semilla inicial de cada posible puerta es necesario elegir un superpíxel que pertenezca a la posible puerta, por lo que se elegirá para cada pareja de rectas el superpíxel que contenga el punto correspondiente al punto medio asociado a las intersecciones con la línea del horizonte de la pareja de rectas, tal y como se ha representado en la Figura 3.2b. En la Figura 3.2a se han representado los puntos medios correspondientes a las parejas de rectas que suponen una hipótesis.

Durante la expansión de la semilla inicial será necesario comparar su color con el de los superpíxeles contiguos mediante la comparación de histogramas, por lo que se almacena el histograma de la región inicial, el cual se realizará en el espacio de color HSV, concretamente sobre los canales H (tono) y S (saturación), pues no se quiere tener en cuenta el nivel de brillo de un mismo color. De esta forma se conseguirá una mejor segmentación de las regiones cuando aparezcan cambios de iluminación.

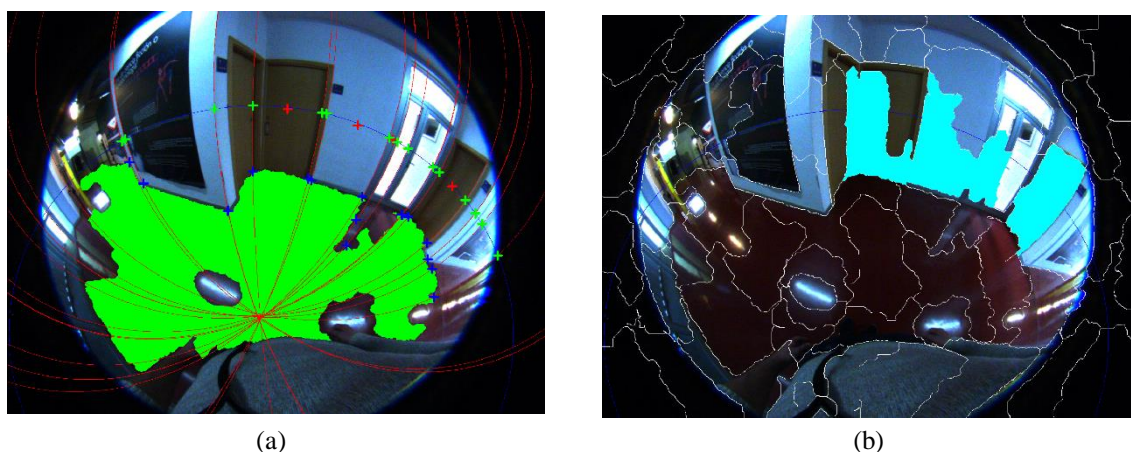


Figura 0.2. (a) Puntos en los que se escogerá la semilla inicial (en rojo) para cada hipótesis. (b) Semilla inicial para cada una de las hipótesis (en azul)

3.2.2. Expansión de la semilla inicial

Tras obtener la semilla inicial se pasa a realizar la expansión de la misma hasta obtener la segmentación completa de la posible puerta. Para ello, se almacenan en primer lugar los superpíxeles vecinos a la región inicial, los cuales se detectan mediante el uso de una máscara binaria.

La máscara binaria se expande mediante la función morfológica de dilatación ya explicada en el Capítulo 2 para conseguir finalmente una ‘expansión’ de la región inicial.

Una vez dilatada la máscara binaria se conocen los superpíxeles contiguos a la semilla inicial, por lo que se calcula el histograma de cada uno de ellos para a continuación compararlos uno a uno con el histograma de la región inicial. Esta comparación se realiza mediante una correlación obteniendo como resultado un valor $d \in [-1,1]$, de manera que cuanto mayor es el valor obtenido mayor es el parecido entre los superpíxeles.

Para que un superpíxel sea candidato a formar parte de la posible puerta debe cumplir la siguiente condición:

- El histograma debe ser suficientemente similar al del superpíxel que forma la semilla inicial.

Pese a ser una única condición es suficiente para conseguir una correcta expansión, ya que generalmente las puertas presentan un color bien diferenciado de la pared, de forma que planteando un umbral para la similitud de histogramas lo suficientemente alto se conseguirá expandir los superpíxeles pertenecientes a una puerta sin que continúe la expansión hacia otras zonas como puede ser la pared.

Cabe destacar, sin embargo, que existe cierto riesgo de fallo al utilizar únicamente esta condición, pues si el superpíxel inicial se corresponde con un reflejo en una puerta no se conseguirá realizar una expansión correcta y por tanto, no se podrá detectar esa puerta. O por el contrario, si en una zona el color de la puerta es similar al de la pared debido a una mala iluminación, la expansión de la semilla inicial llegará hasta las paredes, haciendo difícil que se pueda detectar la puerta.

Una vez se han decidido los superpíxeles que pasarán a formar parte de la posible puerta se actualiza la máscara binaria que se utilizará en la siguiente iteración para obtener los superpíxeles vecinos. De esta forma se realizan sucesivas iteraciones hasta que se consigue una segmentación final de las regiones que podrían ser puerta, tal y como se muestra en la Figura 3.3.

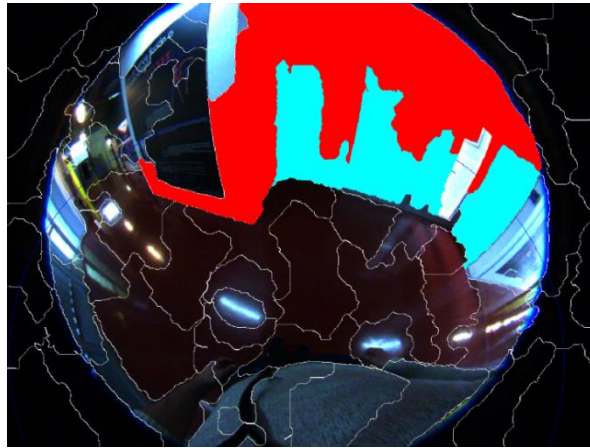


Figura 0.3. Resultado tras la expansión de cada una de las semillas iniciales de la Figura 3.2b

3.3. Eliminación de falsos positivos

Llegado el punto en el que se tiene la región expandida para cada semilla inicial es necesario comprobar si realmente la región se corresponde con una puerta o no, pues aunque las hipótesis iniciales se han realizado teniendo en cuenta que la distancia entre parejas de rectas sea similar a la anchura de una puerta puede suceder que este requisito se cumpla entre rectas pertenecientes a cuadros, secciones de pared, etc.

Para poder reducir al máximo el número de falsos positivos tras la ejecución del algoritmo se han utilizado varios métodos aprovechando tanto información 3D de la escena (línea del horizonte) como información extraída de la propia imagen (puntos de intersección de las rectas, fronteras que forman las rectas, etc.).

3.3.1. Cross ratio

Tal y como se ha comentado en el capítulo 2, las puertas presentan cotas normalizadas, tanto de anchura como de altura, siendo esta última la que se va a tener en cuenta para aumentar la probabilidad de detectar en última instancia puertas verdaderas.

A diferencia del procedimiento realizado para medir la anchura entre parejas de rectas al inicio de este capítulo, no es posible aplicar el mismo método para medir la altura de las regiones expandidas y determinar si pueden ser puerta o no.

Esto se debe principalmente al hecho de que para poder realizar las mediciones en el mundo 3D sería necesario conocer un plano vertical sobre el que proyectar los rayos proyectantes correspondientes. Sin embargo, sólo se conoce el plano del suelo (horizontal) determinado por la información 3D de la cámara de profundidad, utilizado para obtener la anchura entre parejas de rectas.

Por este motivo es necesario buscar otro método para considerar la altura de las regiones expandidas sin necesidad de conocer su altura real en metros, de forma que se ha hecho uso del cross ratio.

El cross ratio o razón armónica se trata de una relación geométrica que tiene como principal característica ser invariante en la geometría proyectiva. Esto quiere decir que se mantiene constante al aplicar cualquier tipo de transformación. Esta relación consiste en un cociente que involucra las distancias entre 4 puntos alineados pertenecientes a cuatro rectas concurrentes (en el caso de la geometría proyectiva se corresponden con los rayos proyectantes con origen en el centro óptico), tal y como se muestra en la Figura 3.4a.

$$(ABCD) = \frac{CA \cdot DB}{CB \cdot DA} \quad (3.3)$$

Siendo, en este caso, el punto de intersección de la vertical con el suelo A , el punto de intersección de la vertical con la línea del horizonte B , el punto de intersección de la vertical con el marco superior de la puerta C , y el punto de fuga de las rectas verticales D . Por otro lado, el punto de fuga de las rectas verticales se encuentra en el infinito, por lo que se puede reducir la relación geométrica a la siguiente expresión:

$$(ABCD) = \frac{CA}{CB} \quad (3.4)$$

Como la relación involucra a las distancias asociadas al punto superior de la puerta (punto C), las cuales son las que se desean comprobar que se cumplen sobre la imagen, se transformará la expresión para involucrar otras distancias, de la siguiente forma:

$$(ABCD)' = \frac{1}{(ABCB)} = \frac{CB}{CA} = \frac{CA - BA}{CA} = 1 - \frac{BA}{CA} \quad (3.5)$$

$$(ABCD)'' = \frac{CA}{BA} \quad (3.6)$$

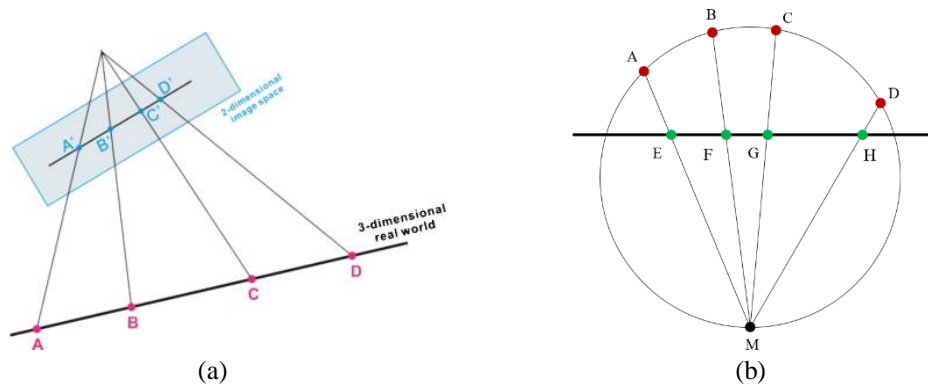


Figura 0.4. (a) Ejemplo de rayos proyectantes y puntos correspondientes para el cálculo del cross ratio en la referencia 3D. Imagen obtenida de [11]. (b) Ejemplo de rayos proyectantes y puntos correspondientes para el cálculo del cross ratio en la referencia fisheye.

La naturaleza invariante de esta relación permite poder establecer un valor del ratio según las medidas reales de una puerta y comprobar que se cumpla con las distancias que se obtienen a partir de la imagen. De esta forma, la distancia CA sería el valor de la altura de una puerta, es decir, 2,03 metros, mientras que la distancia BA será la distancia entre el suelo y la línea del horizonte. Como las cámaras se encuentran colocadas sobre el pecho del usuario, la línea del horizonte suele encontrarse a una altura del orden de 1,20 metros sobre el suelo, por lo que se usará ese valor para calcular el cross ratio, obteniendo un valor nominal para las puertas de 1,7.

Una vez calculada la relación sólo hay que comprobar para qué regiones expandidas en la imagen se cumple. Para ello, se ha obtenido para cada región expandida el punto sobre la línea vertical correspondiente más alejado de la intersección con el suelo, de forma que en caso de ser una región perteneciente a una puerta se correspondería con el punto superior C de la misma.

Una vez se tienen los tres puntos necesarios sobre la imagen se transformarían mediante la calibración intrínseca del fisheye para obtener los rayos proyectantes correspondientes. Sin embargo, tal como se muestra en la Figura 3.4b, debido a la naturaleza de la cámara fisheye y su representación de los puntos sobre una esfera, no se pueden relacionar las distancias directamente sino que es necesario proyectar los rayos sobre un plano vertical para finalmente plantear si se cumple el cross ratio. En la Figura 3.5 se puede observar el efecto que tiene sobre la segmentación realizada el uso del cross ratio.



Figura 0.5. Segmentación de las posibles puertas (en rojo) tras aplicar el criterio del cross ratio a la Figura 3.3

Sin embargo, el hecho de transformar estos puntos del plano imagen a rayo proyectante para después reprojectarlos sobre un plano vertical conlleva un elevado tiempo de cómputo, por lo que se ha decidido comprobar que se cumple la relación sobre las medidas en píxeles de la imagen. De esta forma se ha observado que para un valor de cross ratio de 1,8 se consigue un buen funcionamiento del algoritmo frente al valor de 1,7 calculado teóricamente.

3.3.2. Uso de fronteras para evitar cuadros

Se ha podido observar que mediante el uso del cross ratio es posible eliminar un gran número de falsos positivos, sin embargo, hay algunos casos que se deben tener en cuenta, como por ejemplo el hecho de que la región expandida pertenezca a un cuadro o a un cartel colgado de la pared de dimensiones similares a una puerta, o simplemente una expansión de una zona de pared en la que hay un reflejo dando lugar a una forma similar a una puerta, como es el caso mostrado en la Figura 3.6.

Para evitar que esta clase de casos sean tomados como una puerta verdadera por parte del algoritmo se utilizará la distancia entre los píxeles de la región expandida y el suelo. Puesto que para cada región el número de píxeles es elevado y que sólo es necesario conocer la distancia de los píxeles inferiores se aplicará una dilatación a la región expandida para posteriormente restarle la región antes de modificar. De esta forma se obtienen los píxeles que forman el contorno de la posible puerta, reduciendo considerablemente el número de operaciones a realizar.

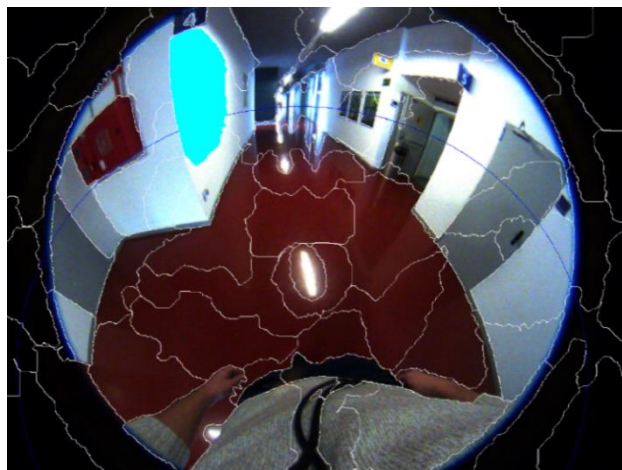


Figura 0.6. Ejemplo de segmentación de una de las posibles puertas con dimensiones similares a una puerta, siendo realmente una sección de pared

Una vez se conocen los píxeles a tratar se calcularía la distancia de cada uno de ellos al suelo, sin embargo, realizar esta operación es costosa, pues no se puede realizar directamente sobre la imagen sino que es necesario transformar cada uno de los píxeles a la referencia del suelo 3D (primero mediante la calibración intrínseca del fisheye y después utilizando la calibración extrínseca entre cámaras) para después realizar la distancia al plano del suelo 3D. Debido al elevado coste computacional de este proceso se ha optado por una solución mucho más sencilla sobre la propia imagen.

Dado que para una puerta real aparecen píxeles cercanos al suelo, y más concretamente a la intersección de cada recta con el suelo, se ha medido la distancia en píxeles de cada punto del contorno con cada una de las intersecciones con el suelo de la pareja de rectas utilizadas para plantear la hipótesis, de forma que si alguna de estas distancias es menor que un umbral determinado se considerará esa región como puerta.

Tras aplicar los dos criterios descritos se ha observado que el resultado obtenido es suficientemente válido como para dar por finalizado el algoritmo, cuyos resultados se evaluarán más adelante. En la Figura 3.7 se pueden observar varios ejemplos de los resultados obtenidos tras aplicar este algoritmo.



Figura 0.7. Ejemplos de imágenes tras aplicar el primer algoritmo de detección de puertas (en rojo)

Capítulo 4

Planteamiento de hipótesis de puerta a partir de una recta vertical

Como se ha observado, los resultados obtenidos a través de esta primera hipótesis son razonablemente válidos, sin embargo, en muchas de las imágenes utilizadas quedan puertas sin detectar por el algoritmo. Esto se debe principalmente al hecho de que el método de extracción de rectas no suele obtener las dos rectas que forman los marcos de una puerta sino que en muchos casos sólo se obtiene una o ninguna recta.

Por este motivo se ha desarrollado un segundo método de detección de puertas, similar al ya explicado, con la característica de que sólo emplea una recta para establecer la hipótesis de puerta, por lo que se trata de un algoritmo más general.

4.1. Segmentación de las posibles puertas

En este segundo método se utilizará en gran parte el mismo procedimiento que el utilizado para la hipótesis a partir de dos rectas. De esta forma, se hará uso de los puntos relevantes de la imagen, es decir, las intersecciones con la línea del horizonte y con el suelo de cada una de las rectas, así como las rectas verticales extraídas.

La principal diferencia con el primer caso es que en primera instancia no se puede realizar un filtrado inicial de rectas que podrían pertenecer a una puerta, pues no se puede tener en cuenta la anchura entre rectas. Este problema conlleva que no se pueden establecer semillas iniciales concretas sino que para cada una de las rectas verticales extraídas habrá que plantear una semilla inicial que luego se expandirá para finalmente comprobar mediante diferentes criterios la posibilidad de ser una puerta.

Al trabajar con una sola recta vertical para el planteamiento de hipótesis de posibles puertas hay que tener en cuenta la región que se elige como semilla inicial para la posterior expansión. En este caso, se puede plantear un sencillo razonamiento para la elección de dicha región para cada una de las rectas, pues si tomamos una recta vertical de la imagen que forme parte del marco

de una puerta se puede afirmar que la puerta se encontrará a la izquierda o a la derecha de la recta seleccionada.

De esta forma, para cada una de las rectas verticales obtenidas se plantearán dos hipótesis de puerta, siendo las semillas iniciales los superpíxeles que se encuentren a la izquierda y a la derecha de la recta, respectivamente. Además, de forma análoga al primer método desarrollado se elegirán los superpíxeles que se encuentran a la altura de la intersección de la recta con la línea del horizonte para que pertenezcan con seguridad a la posible puerta, tal y como se muestra en la Figura 4.1a.

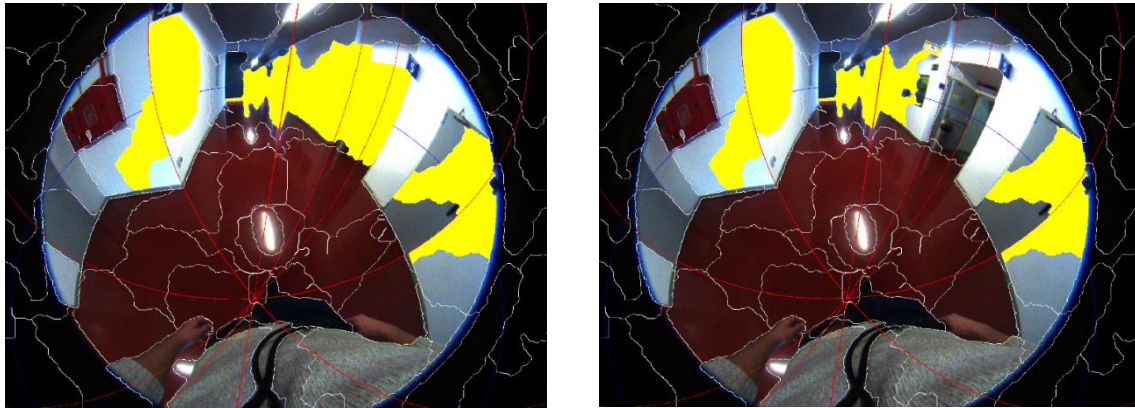


Figura 0.1. (a) Ejemplo de semillas iniciales para cada una de las rectas verticales. (b) Ejemplo de semillas iniciales después de tener en cuenta los resultados obtenidos en el primer método de detección de puertas para la Figura 4.1a

Por otro lado, hay que tener en cuenta el hecho de que este método conlleva un elevado coste computacional, ya que para cada una de las rectas hay que realizar dos hipótesis. Por ello, se va a utilizar información obtenida a partir del primer método de detección de puertas para disminuir el número de rectas a evaluar.

Durante el primer método de detección de puertas se ha utilizado información referente a la anchura y altura de las puertas para reducir el número de hipótesis erróneas, por lo que cabe esperar un buen funcionamiento del método. Por este motivo, en este segundo método no se van a tener en cuenta las rectas verticales que pertenezcan a lo que el primer método ha reconocido como puerta, de forma que se reducirán el número de hipótesis a realizar y por tanto, el número de semillas a expandir. En la Figura 4.1b se puede observar un ejemplo de la disminución del número de semillas iniciales al tener en cuenta los resultados del primer método desarrollado.

Una vez se han elegido las semillas iniciales para cada una de las rectas verticales se realizará el mismo proceso de expansión que el descrito en el Capítulo 3. Si bien es cierto que a cada recta hay asociadas dos hipótesis (semillas a expandir), se realizará la expansión de las regiones iniciales por separado, ya que en última instancia se pretende descartar la hipótesis falsa en caso de que una recta pertenezca a una puerta real.

Por este motivo, se debe tener en cuenta que durante la expansión de cada semilla inicial se puede producir que para una región inicial a la izquierda de la recta correspondiente haya una expansión hacia la derecha de la recta o viceversa. Sin embargo, tal y como se ha explicado, se han elegido las hipótesis razonando que una puerta real estará a un lado u otro de la recta vertical que corresponda, por lo que para cada una de las regiones expandidas sólo se tendrá en cuenta la región que se encuentre al mismo lado de la recta vertical que la semilla inicial correspondiente.

Para conseguir esto se ha hecho uso de una máscara binaria, en este caso, se ha implementado según la posición de la semilla inicial respecto a su recta correspondiente una máscara que tome el valor 1 en todos los píxeles que se encuentran al mismo lado de la recta que la semilla inicial y el valor 0 en los píxeles que se encuentran al otro lado, tal como se observa en las Figuras 4.2a y 4.2b.

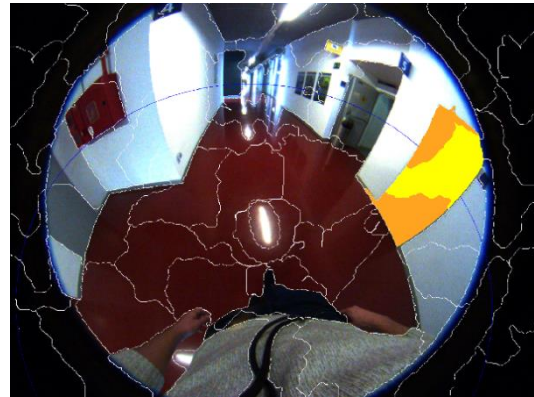


Figura 0.2. (a) Máscara binaria a la izquierda de una recta vertical. (b) Máscara binaria a la derecha de una recta vertical

Una vez se tienen estas máscaras binarias se aplican a la región expandida, es decir, se superponen los píxeles de ambas imágenes de forma que el resultado es el producto de los valores de cada pareja de píxeles en la misma posición, por lo que la región de valor 0 en la máscara binaria permitirá eliminar las regiones irrelevantes. En la Figura 4.3b se puede observar el resultado de aplicar la máscara binaria a una región expandida.



(a)



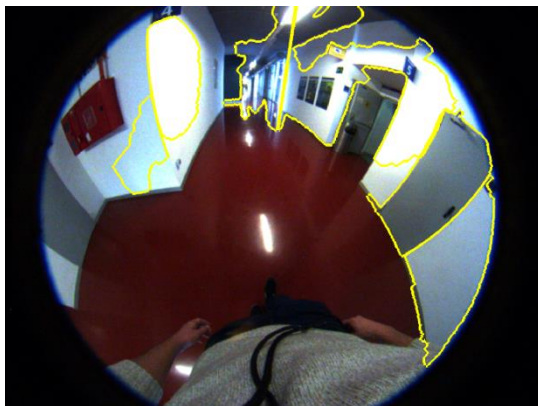
(b)

Figura 0.3. (a) Ejemplo de región expandida para la Figura 4.1b. (b) Región expandida tras aplicar la máscara binaria de la Figura 4.2a

4.2. Filtrado de falsos positivos

Una vez se ha expandido cada una de las semillas iniciales de las hipótesis planteadas es necesario comprobar las hipótesis que pueden ser consideradas verdaderas puertas, por lo que es necesario eliminar en la medida de lo posible los falsos positivos, de igual forma que se ha hecho en el primer método implementado.

De esta forma se utilizarán las herramientas vistas en el capítulo 3 para filtrar las regiones expandidas (Figura 4.4a), pudiéndose observar en la Figura 4.4b el resultado tras aplicar estos métodos.



(a)



(b)

Figura 0.4. (a) Resultado tras la expansión de las semillas iniciales de la Figura 4.2. (b) Resultado tras aplicar los criterios de cross ratio y filtrado de posibles cuadros

Se puede observar que pese al filtrado realizado por medio de estas dos herramientas siguen apareciendo falsos positivos. Esto se debe principalmente a que sólo se están teniendo en cuenta relaciones referentes a la altura de cada una de las regiones para discernir si es o no una puerta.

Sin embargo, a diferencia del primer método, no se ha utilizado todavía información conocida acerca de la anchura de una puerta real, por lo que aparecen regiones que cumplen las condiciones de altura impuestas pero que corresponden a secciones de pared u otros elementos cuya anchura es considerable.

Para poder disminuir todavía más la tasa de falsos positivos en el algoritmo se volverá a utilizar información geométrica de la imagen, en este caso, se utilizará el denominado bounding box.

4.2.1. Bounding box

El bounding box en el caso de una imagen en dos dimensiones es un elemento geométrico que se caracteriza por ser el polígono de menor área que contiene un conjunto de puntos. En este caso se utilizará un cuadrilátero que delimite cada una de las regiones expandidas, de forma que se pueda trabajar con las proporciones de esta nueva geometría.

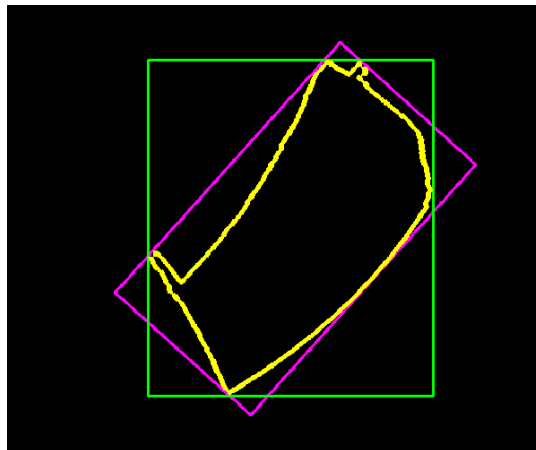


Figura 0.5. Ejemplo de un rectángulo delimitador recto (en verde) y un rectángulo delimitador rotado (en lila) para una región determinada (en amarillo)

En la Figura 4.5 se puede observar el cuadrilátero que delimita un área concreta, habiendo dos posibilidades, un rectángulo recto y un rectángulo rotado. En este caso, debido a que en la imagen pueden aparecer puertas con cierto grado de inclinación se usará el rectángulo rotado para delimitar cada una de las regiones expandidas, de forma que se adapte mejor a su forma.

Pese a la distorsión que aparece en el fisheye las proporciones geométricas características de una puerta se mantienen con un elevado grado de exactitud, tal como se ha visto en el caso del cross ratio.

De esta forma se van a tener en cuenta dos factores para discernir entre que regiones son puerta y cuales no lo son:

- La relación entre la altura y la anchura del cuadrilátero obtenido para cada región debe encontrarse entre dos valores límite.
- Para dar robustez al algoritmo se impondrá que la relación entre el área de la región expandida y el área bajo el cuadrilátero delimitador sea superior a cierto umbral

Para decidir los umbrales a utilizar se han realizado diferentes pruebas con imágenes en las que el algoritmo detecta correctamente al menos una puerta, de forma que se han comprobado tanto la relación de áreas como la relación de altura y anchura para las regiones que se corresponden con una puerta. Una vez realizadas estas mediciones se ha tomado un umbral inferior y un umbral superior para la relación altura-anchura, así como un umbral inferior para la relación entre áreas. En la Figura 4.6 se puede observar el resultado final de la segmentación de puertas tras aplicar el criterio del bounding box.



Figura 0.6. Resultado del segundo método de detección de puertas para la Figura 4.1b

Capítulo 5

Evaluación de resultados

Una vez finalizado el algoritmo de detección de puertas en sus dos variantes según las rectas verticales que se consiguen extraer de la imagen es necesario evaluar experimentalmente el funcionamiento del algoritmo.

Debido a que la combinación de sensores utilizada en el proyecto es una propuesta novedosa no existen bases de datos con las que probar el algoritmo, por lo que se han tomado imágenes propias sobre las que se evaluarán los resultados. Se ha intentado tomar imágenes variadas en cuanto a iluminación se refiere para tratar de conseguir los resultados más fiables posibles.

En cuanto a las imágenes tomadas, cada secuencia tomada presenta un gran número de imágenes, pues el tiempo de muestreo a la hora de tomarlas era pequeño. Si bien las imágenes consecutivas pueden resultar similares en cuanto a características, el algoritmo de extracción de rectas sobre el que se basa todo el algoritmo de detección de puertas es muy sensible, por lo que de una imagen a otra el número de rectas extraídas puede variar en gran medida. Por este motivo, se ha decidido evaluar imágenes similares de una misma secuencia para poder apreciar los cambios significativos que tiene sobre el algoritmo.

Por otro lado, puesto que se han desarrollado dos alternativas durante el proyecto, se evaluarán por separado, tanto en porcentaje de acierto en la detección de puertas como en tiempo de cómputo, ya que el primer algoritmo se podría decir que es un método más específico al depender de la extracción de las dos rectas que forman los marcos verticales de las puertas, mientras que el segundo método es más general, al depender sólo de una recta. Además, el fin del proyecto es que se pueda implementar en un futuro en un sistema de guiado de personas con deficiencia visual, por lo que se podría plantear el uso de una u otra variante en función de los resultados que se quieran conseguir y el tiempo de cómputo máximo.

5.1. Caso 1

En este segundo ejemplo se observa en la Figura 5.1a una puerta de madera bien diferenciada de la pared, así como una puerta de cristal en la que hay grandes reflejos.

Una vez segmentada la imagen (Figura 5.1b) y extraído las rectas verticales se observa en la Figura 5.1c que para la puerta de madera aparecen numerosas rectas, por lo que se podrá plantear una hipótesis a partir de dos rectas. Sin embargo, para la puerta de cristal sólo se planteará hipótesis a partir de una recta, pues aunque aparecen varias rectas la distancia es demasiado grande como para que el primer algoritmo trate esta región.

En la Figura 5.1d se pueden ver las regiones expandidas para las hipótesis de dos rectas, una de ellas se corresponde con la puerta de madera, mientras que la otra es un falso positivo de la pared. Tras filtrar las regiones (Figura 5.1e) se aprecia que este falso positivo se ha eliminado y sólo ha quedado la puerta de madera.

En cuanto a las hipótesis a partir de una recta, se ve en la Figura 5.1f que aparecen numerosas regiones, entre ellas la puerta de cristal y la puerta de madera, la cual no debería aparecer ya que el primer algoritmo ya ha detectado esa región como puerta. Sin embargo, debido a que se han extraído cuatro rectas pertenecientes a la puerta, aunque con el primer algoritmo se han eliminado dos de ellas aún quedan dos rectas que son tratadas como hipótesis de una recta.

Tras realizar el filtrado para este segundo algoritmo se aprecia en la Figura 5.1g que se han eliminado todos los falsos positivos y sólo aparece la puerta de madera. Sin embargo, la puerta de cristal ha sido eliminada debido a que los reflejos han impedido una buena expansión, por tanto la región queda a una altura considerable del suelo y el algoritmo la elimina por ser un posible cuadro.

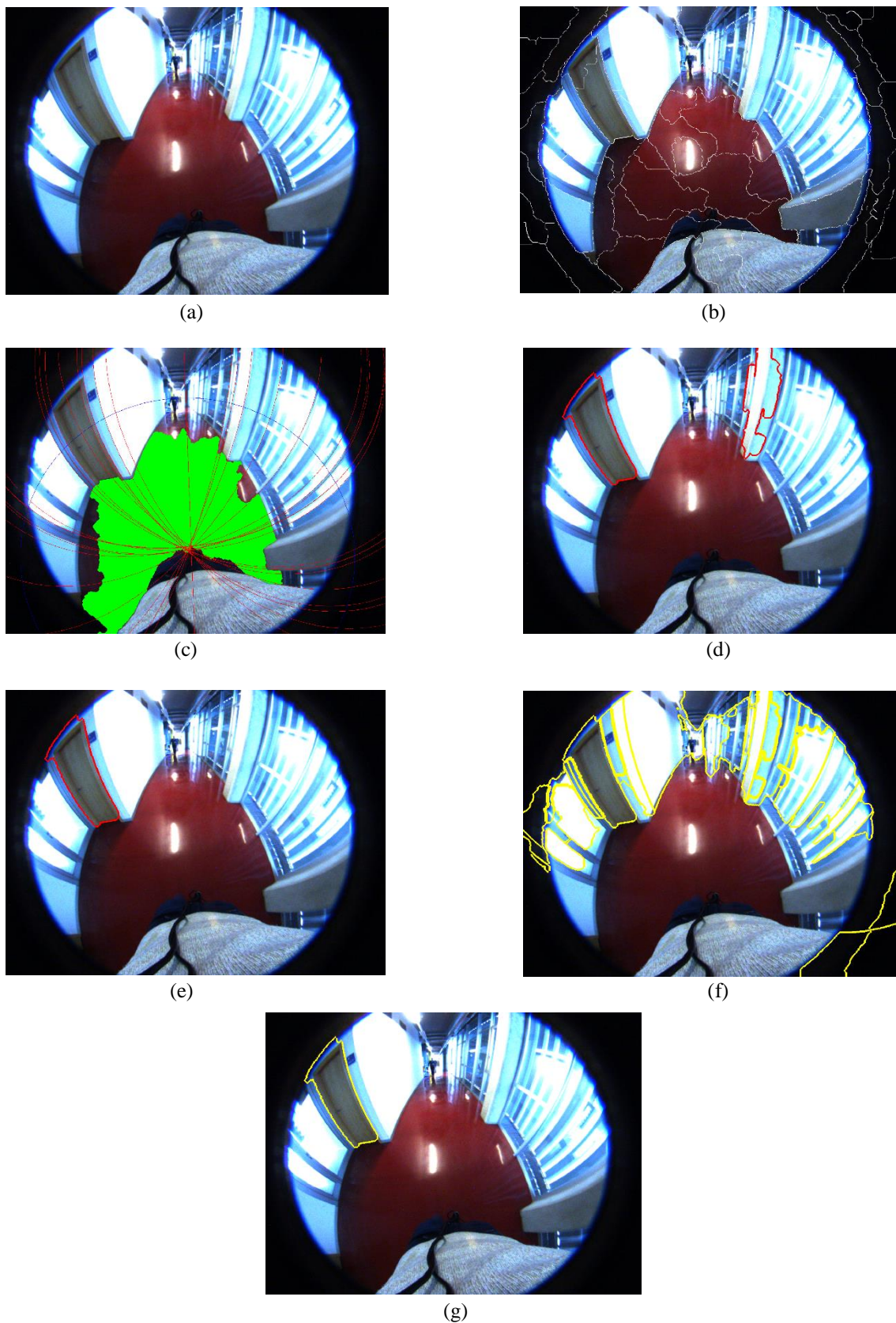


Figura 0.1. (a) Imagen del Caso 1. (b) Segmentación de la escena. (c) Rectas verticales y suelo expandido. (d) Expansión de las semillas iniciales para las hipótesis a partir de dos rectas. (e) Resultado tras filtrar las regiones. (f) Expansión de las semillas iniciales para las hipótesis a partir de dos rectas. (g) Resultado tras filtrar las regiones

5.2. Caso 2

En la Figura 5.2a se observa una imagen similar a la del Caso 1, aunque desde una zona más alejada, de forma que además de la puerta de cristal junto a la puerta de madera aparecen dos puertas de madera a la izquierda de la imagen.

Tras realizar la segmentación de la escena (Figura 5.2b) y extraer la rectas verticales (Figura 5.2c) se aprecia que para la puerta de cristal no aparece ninguna recta, mientras que para las dos puertas de la izquierda se tienen rectas suficientes como para plantear hipótesis a partir de dos rectas. Por último, para la última puerta de madera solo se tiene una recta, por lo que actuará sobre esta el segundo algoritmo.

Tras expandir las semillas iniciales de las hipótesis a partir de dos rectas se observa en la Figura 5.2d que se tienen las dos puertas de la izquierda perfectamente delimitadas, aunque aparecen las dos en una misma región, ya que al estar pegadas y ser del mismo color ambas semillas iniciales se han expandido a lo largo de las dos puertas. Por otro lado, tras aplicar el filtrado (Figura 5.2e) se obtiene la misma región.

En cuanto a las hipótesis a partir de una recta, se observa en la Figura 5.2f que aparecen un gran número de falsos positivos, así como las dos puertas de madera de la izquierda, ya que pese a que han sido detectadas aún hay una recta en esa zona que sirve como hipótesis. Por último también aparece expandida la última puerta de madera.

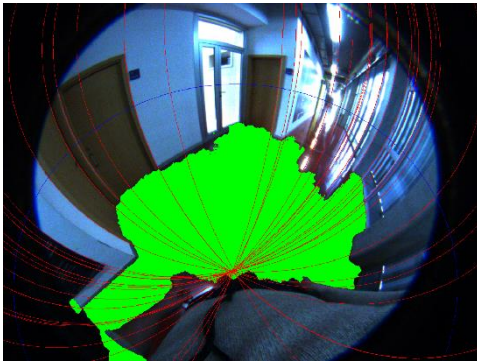
Tras aplicar el filtrado se obtiene en la Figura 5.2g tanto una de las puertas de madera de la izquierda como la puerta contigua a la puerta de cristal, sin embargo, la puerta situada más a la izquierda ha sido eliminada. Esto se debe a que la expansión ha comenzado en un superpíxel que ha permitido que se expanda hacia parte de la puerta derecha, sin embargo, no la llega a ocupar completamente por lo que al aplicar el bounding box no se cumple la relación de altura y anchura. Por otro lado, cabe destacar que todos los falsos positivos han sido eliminados.



(a)



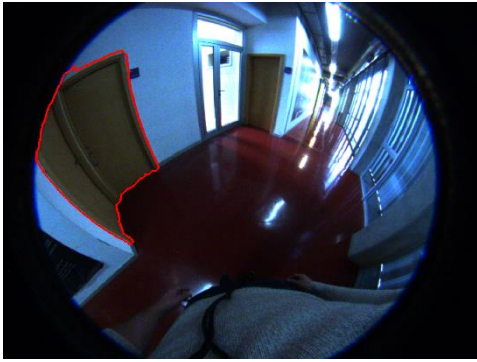
(b)



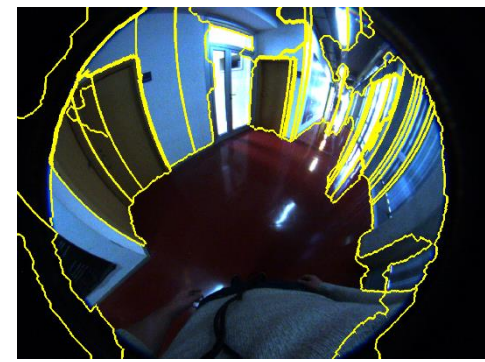
(c)



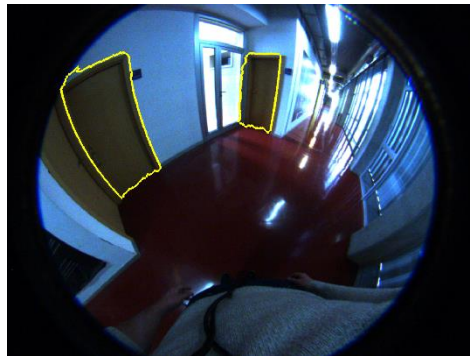
(d)



(e)



(f)



(g)

Figura 0.2. (a) Imagen del Caso 2. (b) Segmentación de la escena. (c) Rectas verticales y suelo expandido. (d) Expansión de las semillas iniciales para las hipótesis a partir de dos rectas. (e) Resultado tras filtrar las regiones. (f) Expansión de las semillas iniciales para las hipótesis a partir de dos rectas. (g) Resultado tras filtrar las regiones

5.3. Caso 3

En la Figura 5.3a se aprecia una puerta a la izquierda con un color similar a la pared que la rodea, una segunda puerta algo escondida en una zona de color muy uniforme y una última puerta a la derecha mejor diferenciada de la pared.

Tras realizar la segmentación por superpíxeles (Figura 5.3b) y extraer las rectas verticales de la escena (Figura 5.3c) se observa que sólo para la puerta de la izquierda se tienen las dos rectas de sus marcos, mientras que para las otras dos puertas sólo se tiene una recta para cada una.

Tras realizar la expansión de las semillas iniciales (Figura 5.3d) tenemos una región que contiene la puerta correspondiente, aunque se expande por ciertas zonas de la pared, pues como se ha comentado, el color es muy similar al de la puerta. Tras aplicar las herramientas de filtrado (Figura 5.3e) se aprecia que sigue manteniendo la región como puerta, lo cual es cierto, aunque debido a la mala expansión no debería ser detectada como puerta real. Cabe destacar que si se hubiera hecho uso del bounding box en este primer algoritmo esta región no hubiera sido detectada como puerta finalmente.

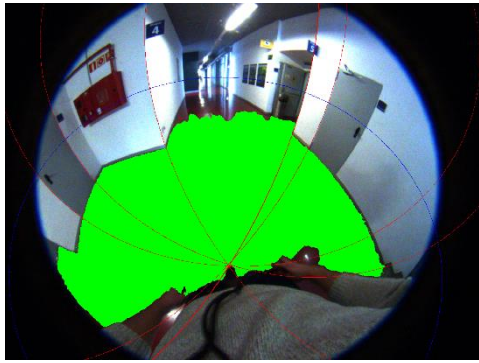
En cuanto a las hipótesis a partir de una recta se ve en la Figura 5.3f la gran cantidad de regiones, dos de las cuales contienen a las dos puertas que faltan por detectar. Sin embargo, tras realizar el filtrado (Figura 5.3g) se observa que el algoritmo no detecta ninguna puerta. Es decir, se han eliminado todos los falsos positivos pero también las dos puertas reales. Esto se debe, en el caso de las dos puertas a que la expansión de la región no se adapta perfectamente a la puerta, por lo que las relaciones de altura y anchura utilizadas en el bounding box no se cumplen. Se podría solucionar restringiendo más aún el umbral de similitud de histogramas durante la expansión de las semillas iniciales para intentar que no se expandan las regiones por las paredes.



(a)



(b)



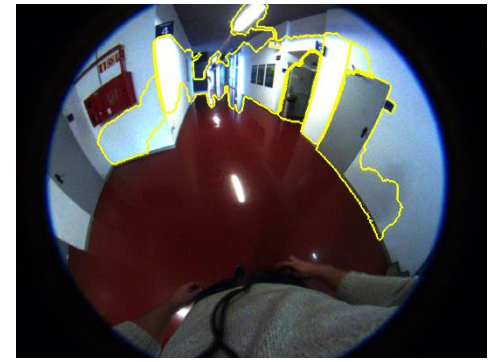
(c)



(d)



(e)



(f)



(g)

Figura 0.3. (a) Imagen del Caso 3. (b) Segmentación de la escena. (c) Rectas verticales y suelo expandido. (d) Expansión de las semillas iniciales para las hipótesis a partir de dos rectas. (e) Resultado tras filtrar las regiones. (f) Expansión de las semillas iniciales para las hipótesis a partir de dos rectas. (g) Resultado tras filtrar las regiones

5.4. Caso 4

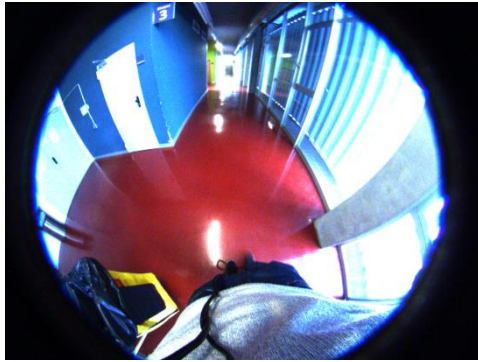
En la Figura 5.4a se observa una escena con una gran iluminación, de forma que a la izquierda aparece una puerta con un color idéntico al de la pared que la rodea, mientras que en el centro de la imagen aparece otra puerta bien diferenciada de la pared.

Tras segmentar la escena (Figura 5.4b) y extraer las rectas verticales (Figura 5.4c) se aprecia que para ambas puertas se tiene sólo una recta, por lo que en un principio sólo actuará el algoritmo de hipótesis a partir de una recta.

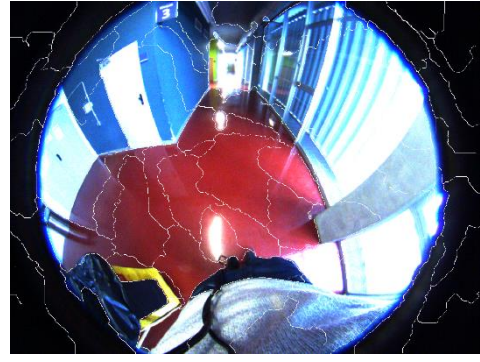
Sin embargo, tal y como se aprecia en la Figura 5.4d, tras aplicar el algoritmo de hipótesis a partir de dos rectas se tiene que una de las puertas ha sido detectada y expandida correctamente. Esto se debe a la mala segmentación del suelo (Figura 5.4c) de forma que aparece una intersección con el suelo que no sería la real haciendo que se cumpla el umbral impuesto de distancia entre rectas para el primer algoritmo. Además, aparece otra región que se corresponde con un falso positivo.

Tras filtrar las regiones se observa en la Figura 5.4e que finalmente sólo aparece la puerta, de forma que el falso positivo es eliminado.

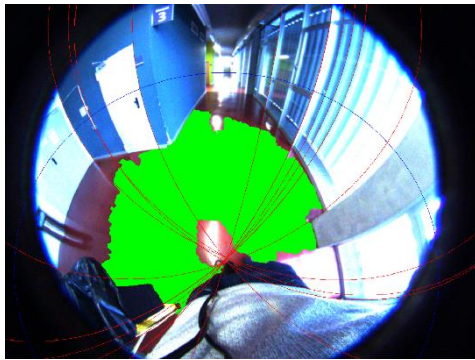
Por otro lado, el algoritmo de hipótesis a partir de una recta ya no actuará sobre la recta perteneciente a esta puerta detectada, por lo que sólo queda la puerta de la izquierda por detectar. En la Figura 5.4f se ve que se ha expandido una región que se adapta a la puerta restante, aunque también contiene parte de la pared debido a la similitud de color. También hay que destacar el elevado número de falsos positivos detectados. Tras aplicar el filtrado (Figura 5.4g) se obtiene finalmente que ninguna región expandida se corresponde a una puerta. Si bien ha eliminado todos los falsos positivos, la puerta de la izquierda también ha sido eliminada. Esto se debe a la mala expansión de la región de forma que no se cumple el criterio del bounding box. Este es un error prácticamente imposible de solventar en esta imagen, ya que la iluminación en esa zona hace que la pared y la puerta no sean diferenciables.



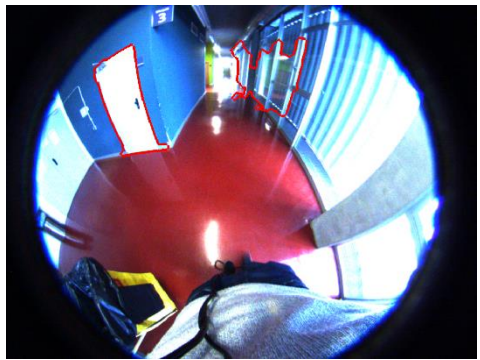
(a)



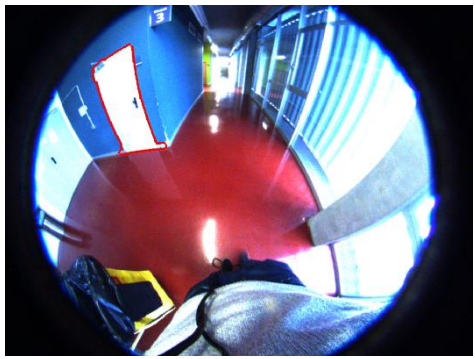
(b)



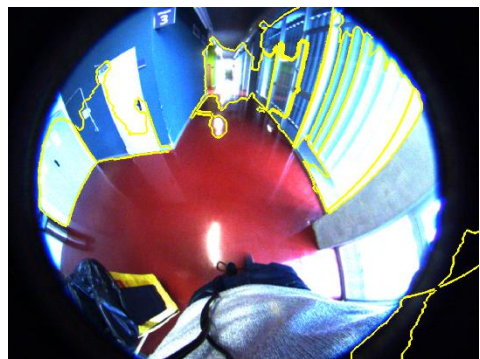
(c)



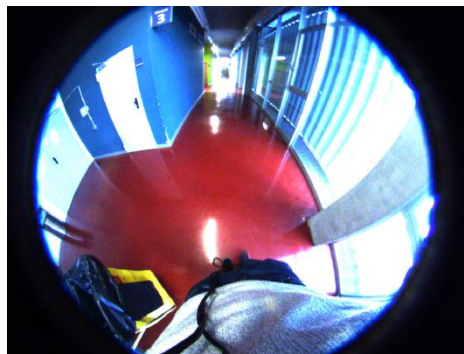
(d)



(e)



(f)



(g)

Figura 0.4. (a) Imagen del Caso 4. (b) Segmentación de la escena. (c) Rectas verticales y suelo expandido. (d) Expansión de las semillas iniciales para las hipótesis a partir de dos rectas. (e) Resultado tras filtrar las regiones. (f) Expansión de las semillas iniciales para las hipótesis a partir de dos rectas. (g) Resultado tras filtrar las regiones

5.5. Caso 5

En la Figura 5.5a se observa una doble puerta a la izquierda con un gran reflejo, así como una puerta de madera y una última puerta a la derecha con un pequeño reflejo.

Tras segmentar la escena (Figura 5.5b) y extraer las rectas verticales (Figura 5.5c) se observa que para la puerta doble y para la puerta de madera aparecen rectas suficientes para plantear hipótesis a partir de dos rectas, mientras que la tercera puerta sólo presenta una recta.

Tras aplicar la expansión del primer algoritmo (Figura 5.5d) se aprecia la expansión de la región a la que pertenece la puerta de madera, una región correspondiente a un falso positivo y una pequeña región perteneciente a la puerta doble, concretamente la región se adapta al reflejo que aparece.

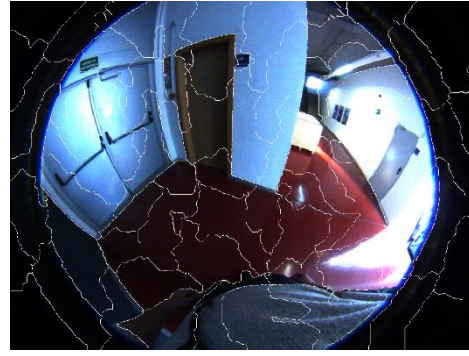
Tras realizar el filtrado (Figura 5.5e) se aprecia que sólo queda la región de la puerta de madera, mientras que el falso positivo y la puerta doble han sido eliminadas. Esta última región ha sido eliminada debido a la mala expansión, ya que al tener una gran región con un color bastante uniforme debido a la poca iluminación de esa zona la semilla inicial se expande en exceso, llegando hasta la pared.

En la Figura 5.5f se observa la expansión de regiones tras aplicar el segundo algoritmo, de forma que se tienen numerosos falsos positivos, pero también la puerta de la derecha de la imagen, así como la puerta doble.

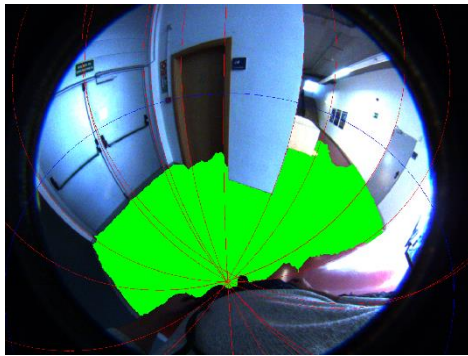
Tras el filtrado (Figura 5.5g) se aprecia que ninguna región ha sido elegida finalmente como puerta. En este caso la doble puerta se ha eliminado debido a las grandes dimensiones de la región, igual que en el caso anterior al no cumplirse el cross ratio. Por otro lado, la puerta de la derecha ha sido descartada debido a una mala expansión también, en este caso fruto de una mala segmentación de los superpíxeles, de forma que la relación entre altura y anchura de la región no se cumple. Esto se podría solventar modificando ciertos parámetros de la segmentación por superpíxeles para que se adapten mejor a la puerta.



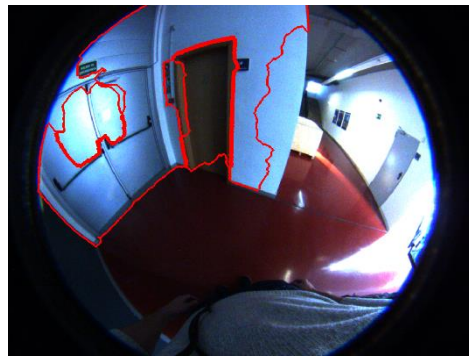
(a)



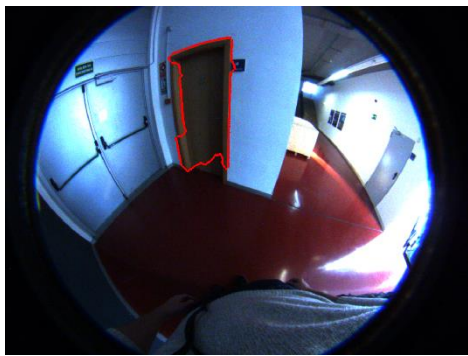
(b)



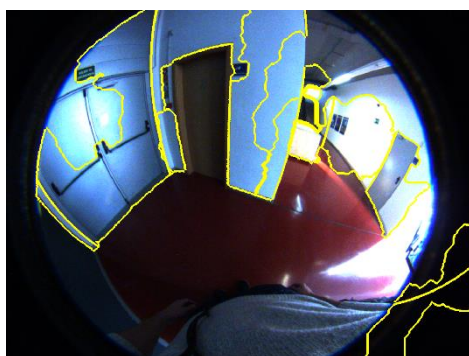
(c)



(d)



(e)



(f)



(g)

Figura 0.5. (a) Imagen del Caso 5. (b) Segmentación de la escena. (c) Rectas verticales y suelo expandido. (d) Expansión de las semillas iniciales para las hipótesis a partir de dos rectas. (e) Resultado tras filtrar las regiones. (f) Expansión de las semillas iniciales para las hipótesis a partir de dos rectas. (g) Resultado tras filtrar las regiones

5.6. Conclusiones de la evaluación

Una vez concluida la evaluación experimental se pasará a analizar el resultado de estas pruebas. Para ello se ha confeccionado una tabla en la que para ejemplo y algoritmo se ha detallado el tiempo de cómputo, el porcentaje de puertas sobre el que ha actuado el algoritmo, pues no siempre se consiguen extraer rectas en todas las puertas de la escena, y el porcentaje de puertas detectadas respecto a las puertas sobre las que se aplica el algoritmo correspondiente.

Sin embargo, puesto que las hipótesis a partir de una recta se basan en el resultado de las hipótesis a partir de dos rectas, durante la confección de la tabla se tendrá en cuenta para los datos del algoritmo de detección a partir de una recta las puertas detectadas por el primer algoritmo, ya que este segundo método es más general y podría detectar las puertas detectadas mediante hipótesis a partir de dos rectas. En la Tabla 5.1 se han detallado los resultados para el primer algoritmo mientras que en la Tabla 5.2 Se ha evaluado el segundo algoritmo.

Hipótesis a partir de dos rectas				
	Tiempo de cómputo (s)	% de puertas sobre las que se aplica el algoritmo	% de puertas detectadas	% de falsos positivos
Caso 1	2,23	50	100	0
Caso 2	3,02	75	100	0
Caso 3	0,86	33,33	100	0
Caso 4	1,6	50	100	0
Caso 5	2,65	75	33,33	0
MEDIA	2,07	56,67	86,67	0
		% PUERTAS DETECTADAS TOTAL	49,11	

Tabla 0.1. Resultados de la evaluación experimental para el algoritmo de hipótesis a partir de dos rectas

Hipótesis a partir de una recta				
	Tiempo de cómputo (s)	% de puertas sobre las que se aplica el algoritmo	% de puertas detectadas	% de falsos positivos
Caso 1	2,25	100	33,33	0
Caso 2	5,9	100	50	0
Caso 3	16,31	75	66,67	0
Caso 4	4,24	100	50	0
Caso 5	5,82	100	25	0
MEDIA	6,90	95	45	0
		% PUERTAS DETECTADAS TOTAL	42,75	

Tabla 0.2. Resultados de la evaluación experimental para el algoritmo de hipótesis a partir de dos rectas

Se puede observar a partir de estos resultados que el primer algoritmo sólo se aplica a la mitad de las puertas aproximadamente, ya que no es usual que aparezcan las dos rectas de los marcos tras extraer las rectas verticales, sin embargo, estas puertas se detectan con un elevado grado de acierto (86,67%) con un tiempo de cómputo bastante reducido, siendo el porcentaje final de puertas detectado de casi el 50%.

Por otro lado, el segundo algoritmo trabaja sobre un mayor número de puertas (95%), ya que generalmente siempre se extrae al menos una recta perteneciente a cada puerta, por lo que el algoritmo presenta un elevado porcentaje de aplicación. Sin embargo, el porcentaje de puertas detectadas es bajo, pues no llega al 50%, siendo el porcentaje final de detección algo menor que en el primer algoritmo. Además, el tiempo de cómputo es mayor y se pueden apreciar imágenes con un tiempo de cómputo mayor de 10 segundos, lo cual no sería favorable para implementar en un sistema en tiempo real.

Pese a que el segundo algoritmo parece peor, hay que destacar que el algoritmo actúa sobre prácticamente el doble de puertas, consiguiendo un resultado de detección similar. Sin embargo, como se ha visto en la evaluación de resultados, en muchos casos estos errores son evitables mediante una mejor segmentación de la escena mediante superpíxeles, lo cual está ligado directamente al *hardware* utilizado. En el caso de este proyecto el ordenador portátil utilizado no es excesivamente potente, por lo que mejorando ligeramente la capacidad de cómputo del equipo se mejoraría considerablemente el porcentaje de acierto del segundo algoritmo, convirtiéndolo en un buen mejor método de detección de puertas.

Por otro lado, pese a que algunos errores son evitables, también aparecen otros que no se pueden evitar, pues el campo de la visión por ordenador es muy dependiente de la iluminación como se ha visto, de forma que aquellas puertas con grandes reflejos o que no se diferencien bien de la pared debido a las sombras por una mala iluminación son realmente difíciles de detectar.

Sin embargo, hay que tener en cuenta como uno de los puntos más importantes del algoritmo el hecho de que la tasa de falsos positivos es nula en los ejemplos estudiados, y por tanto, se espera que con un volumen mayor de imágenes esta tasa también sea mínima, lo que ayuda en gran medida a la implementación de este algoritmo en un sistema de navegación en tiempo real.

Al no existir falsos positivos las puertas detectadas se podrían almacenar para en un instante posterior realizar un ‘tracking’ de las mismas en función de hacia donde se mueve el usuario de forma que en todo momento se tengan localizadas pese a que en esos instantes el algoritmo deje de detectar esas puertas.

Por este motivo es tan importante el hecho de que el algoritmo de detección de puertas a partir de hipótesis de una recta funcione sobre un elevado porcentaje de puertas de la escena, ya que

permitiría, mediante alguna pequeña mejora, poder detectar y tener localizadas un gran número de puertas del entorno. También hay que tener en cuenta que para imágenes similares puede cambiar en gran medida el funcionamiento del algoritmo, de forma que en un instante determinado no se detecte una puerta y al instante siguiente sí que se consiga detectar, por lo que este planteamiento para la navegación sería un sistema a tener en cuenta para aprovechar al máximo la capacidad del algoritmo de detección de puertas.

Por último cabe destacar que si bien el segundo algoritmo sería el más indicado para el sistema de navegación, los tiempos de cómputo pueden llegar a ser muy elevados, mientras que para el primer algoritmo son tiempos muy razonables para un sistema en tiempo real, por lo que habría que llegar a un compromiso acerca del sistema a utilizar en función de si se quiere detectar un menor número de puertas pero que exista una gran capacidad a la hora de plantear trayectorias, o por el contrario se prefiere detectar un mayor número de puertas a costa de un mayor tiempo de cómputo y por tanto, una menor capacidad para plantear trayectorias.

Capítulo 6

Conclusiones y trabajo futuro

A lo largo del proyecto se ha conseguido desarrollar un algoritmo con dos variantes que permita la detección de puertas de una escena, a partir de la información captada por un sistema de dos cámaras, un sensor RGB-D para captar información 3D y una cámara fisheye para captar información de color de una gran región de la escena.

Haciendo uso de esta información, y mediante herramientas para el tratamiento de imágenes, se ha llegado a desarrollar un algoritmo robusto que en sus dos variantes, cada una con sus ventajas e inconvenientes, permite detectar aproximadamente un 50% de las puertas de la escena. Aunque se ha comprobado que para instantes consecutivos se pueden detectar puertas nuevas o dejar de detectar puertas que en un instante anterior eran detectadas.

Por este motivo cobra especial importancia la baja tasa de falsos positivos, que permitirá en un futuro implementar estos algoritmos en un sistema de navegación para personas con deficiencias visuales y que sea capaz de actuar en tiempo real.

Con los resultados de este proyecto, y junto con el proyecto de detección de espacio libre en una escena [6] el siguiente paso para avanzar en este proyecto y acercarse al objetivo de ayudar a las personas sería comenzar el desarrollo de un sistema de navegación, de forma que se plantee la base de su funcionamiento y se implementen los algoritmos de detección de puertas y del espacio libre de obstáculos, para tener finalmente un sistema que permita la planificación de trayectorias en personas con deficiencia visual.

Además, sería adecuado poder realizar pruebas con personas que sufran este tipo de enfermedades visuales para poder conocer sus impresiones y detectar posibles cambios o mejoras a los algoritmos ya diseñados que permitan una mejor experiencia para este colectivo.

Por último, cabe destacar que todos los objetivos planteados al inicio del proyecto se han cumplido, pese a la dificultad que conlleva el trabajar en el campo de la visión por computador, tan dependiente de un gran número de factores que en muchos casos son inevitables.

Bibliografía

- [1] A. C. Murillo, J. Košecká, J. J. Guerrero, and C. Sagüés, “Visual door detection integrating appearance and shape cues,” *Rob. Auton. Syst.*, vol. 56, no. 6, pp. 512–521, 2008.
- [2] J. Bermudez-Cameo, G. Lopez-Nicolas, and J. J. Guerrero, “Automatic Line Extraction in Uncalibrated Omnidirectional Cameras with Revolution Symmetry,” *Int. J. Comput. Vis.*, vol. 114, no. 1, pp. 16–37, 2015.
- [3] M. Talebi, A. Vafaei, and A. Monadjemi, “Vision-based entrance detection in outdoor scenes,” *Multimed. Tools Appl.*, vol. 77, no. 20, pp. 26219–26238, 2018.
- [4] Y. Zhou, G. Jiang, G. Xu, X. Wu, and L. Krundel, “Kinect depth image based door detection for autonomous indoor navigation,” *Proc. - IEEE Int. Work. Robot Hum. Interact. Commun.*, vol. 2014-Octob, no. October, pp. 147–152, 2014.
- [5] T. H. Yuan, F. H. Hashim, W. M. D. W. Zaki, and A. B. Huddin, “An automated 3D scanning algorithm using depth cameras for door detection,” *Proc. - 2015 Int. Electron. Symp. Emerg. Technol. Electron. Information, IES 2015*, pp. 58–61, 2016.
- [6] M. Guerrero Viu, “Sistema de visión omnidireccional y de profundidad para guiado de discapacitados visuales”, T.F. Máster Ing. Biomédica, EINA, Universidad de Zaragoza, 2019.
- [7] C. Fernández-Caramés, V. Moreno, B. Curto, J. F. Rodríguez-Aragón, and F. J. Serrano, “A Real-time Door Detection System for Domestic Robotic Navigation,” *J. Intell. Robot. Syst. Theory Appl.*, vol. 76, no. 1, pp. 119–136, 2013.
- [8] M. Van den Bergh, X. Boix, G. Roig, and L. Van Gool, “SEEDS: Superpixels Extracted Via Energy-Driven Sampling,” *Int. J. Comput. Vis.*, 2015.
- [9] D. Scaramuzza and R. Siegwart, “A Practical Toolbox for Calibrating Omnidirectional Cameras,” in *Vision Systems: Applications*, 2012.
- [10] M. A. Fischler and R. C. Bolles, “Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography,” *Commun. ACM*, 1981.
- [11] T. W. Wong, C. H. Tao, Y. K. Cheng, K. H. Wong, and C. N. Tam, “Application of cross-ratio in traffic accident reconstruction,” *Forensic Sci. Int.*, 2014.

Anexos

Anexo A

Resumen del trabajo previo

En este anexo se propone un pequeño resumen del Trabajo Fin de Máster de Guerrero-Viu [6] que ha servido como base para la realización de este proyecto.

A.1. Sensores utilizados

El uso de cámaras RGB-D permite obtener tanto información 3D de una escena como información de color (RGB) con un único dispositivo, sin embargo, presenta una serie de limitaciones debido principalmente al escaso rango de visión y a la presencia de ruido que presenta la cámara.

Por este motivo se ha planteado el uso de un dispositivo adicional y complementario, en este caso una cámara tipo fisheye, cuyo campo de visión es mucho mayor, pudiendo captar una mayor información de la escena. De esta forma se obtiene la información 3D de la escena mediante la cámara de profundidad y la información RGB mediante la cámara fisheye.

Los sensores utilizados para el desarrollo del proyecto han sido la cámara RGB-D Asus Xtion Pro Live con una resolución de 640 x 480 píxeles y la cámara uEye UI-3580CP, cuya resolución se ha limitado a 1280 x 960 píxeles para disminuir el tiempo de cómputo, en combinación con una lente fisheye Lensagon CF5M1414 logrando un campo de visión de 182°. En la Figura A.1 se muestra el conjunto de sensores utilizado.



Figura 0.1. Sistema de cámaras utilizado en el proyecto, compuesto por una cámara fisheye y una cámara RGB-D

Por otro lado, esta combinación de cámaras irá colgada del cuello del usuario, de forma que se apoye sobre el pecho formando un ángulo de 45° aproximadamente, tal y como se muestra en la Figura A.2.



Figura 0.2. Posición del conjunto de cámaras sobre el sujeto

A.2. Calibración de las cámaras

Debido a la distorsión que presenta la cámara fisheye es necesario realizar una calibración intrínseca de la misma, esto es, hallar los parámetros que definen esta distorsión. Para ello se ha utilizado el modelo de Scaramuzza y col. [9] que se puede aplicar a todo tipo de cámaras.

Por último, suponiendo que la calibración de la cámara de profundidad dada por el fabricante es adecuada es necesario realizar la calibración extrínseca entre ambos dispositivos. Para esto se ha utilizado el método desarrollado por Pérez-Yus y col. [2], que se basa en la extracción de líneas de la escena a través de ambos dispositivos (Figura A.3) para después buscar correspondencias entre estas, de forma que se formulan restricciones geométricas para calcular la posición relativa entre las cámaras.

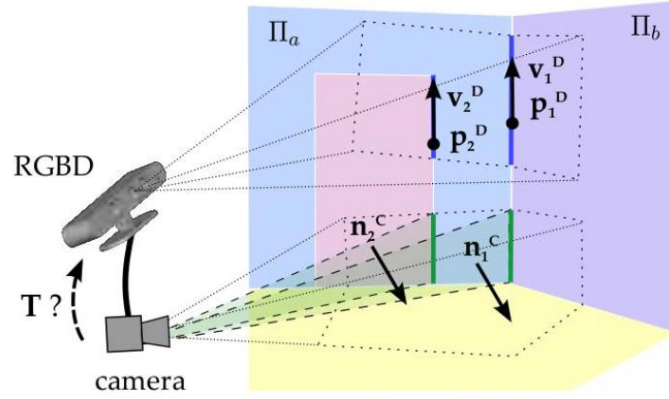


Figura 0.3. Observación de líneas en una escena por una cámara RGB-D y una cámara convencional con campos de visión no solapados. Imagen obtenida de [11]

A.3. Detección del suelo mediante la cámara de profundidad

En cualquier sistema de navegación es necesario conocer primero el espacio libre de obstáculos, por lo que en primer lugar hay que realizar la segmentación del suelo utilizando la información 3D proporcionada por la cámara de profundidad. Esta información se corresponde con una nube de puntos que representa las coordenadas 3D de cada uno de los píxeles de la escena.

Para detectar y extraer el suelo de la escena se ha utilizado el algoritmo Random Sample Consensus (RANSAC) [10], que consiste en un modelo iterativo que estima un modelo matemático deseado a partir de la información de entrada. En este caso el modelo buscado es un plano y la información de entrada es la nube de puntos.

El algoritmo elige al azar tres puntos que formarán un plano y el resto de puntos votan o no por este plano en función de su distancia al mismo (mediante un umbral). Este proceso se itera un número determinado de veces y una vez finalizado se elige el plano con más votos como la solución buscada, siendo su ecuación de la siguiente forma:

$$AX + BY + CZ + D = 0 \quad (A.1)$$

Siendo los parámetros A , B y C las coordenadas del vector ortogonal al plano y el parámetro D la distancia del plano al origen de coordenadas, en este caso la cámara.

Cabe destacar que al aplicar el RANSAC el plano que se obtiene coincide con el de mayor tamaño en la escena, por lo que debido al posicionamiento de las cámaras siempre será el plano del suelo, tal y como se observa en la Figura A.4b.

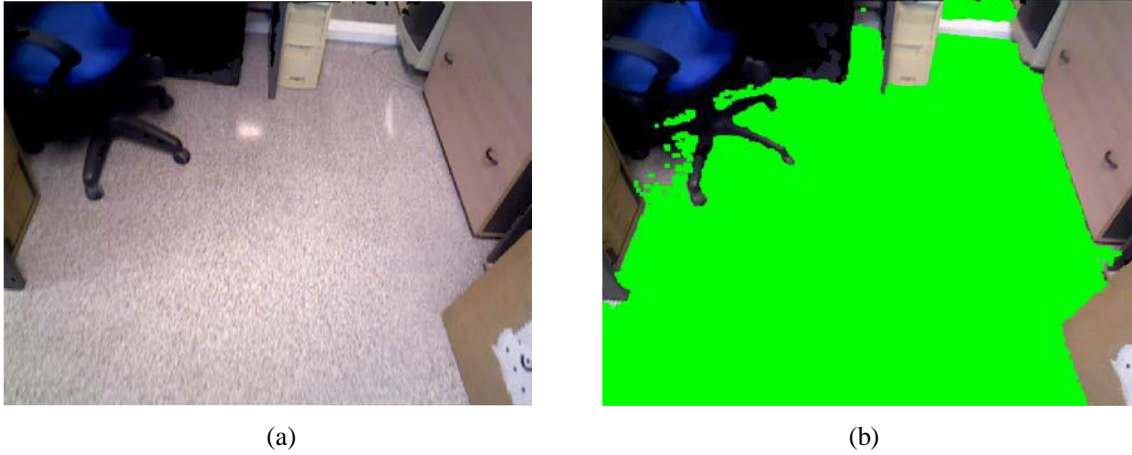


Figura 0.4. (a) Imagen original. (b) Segmentación del plano del suelo mediante la cámara de profundidad

A.4. Expansión del suelo

Con la información obtenida a través de la cámara de profundidad sería posible la evitación de obstáculos, sin embargo, sería conveniente obtener una información más precisa del resto de la escena para la navegación del usuario. Por ello es necesario convertir la información del plano 3D del suelo a la cámara fisheye para posteriormente expandir el espacio libre de obstáculos con la información obtenida por el fisheye, obteniendo finalmente una segmentación completa del suelo de la escena.

El paso de la información de una referencia a otra se realiza mediante la matriz homogénea obtenida durante la calibración extrínseca de las cámaras, siendo la ecuación que relaciona ambas referencias de la siguiente forma:

$$\begin{pmatrix} X^F \\ Y^F \\ Z^F \\ 1 \end{pmatrix} = \begin{pmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} X^D \\ Y^D \\ Z^D \\ 1 \end{pmatrix} \quad (A.2)$$

Dónde (X^D, Y^D, Z^D) son las coordenadas 3D de un punto obtenido por medio de la cámara de profundidad, r_{ij} y t_j son respectivamente, los elementos de la matriz de rotación y la matriz de traslación que relacionan la cámara de profundidad con la cámara fisheye obtenidos a partir de la calibración extrínseca.

Una vez se tiene la información 3D en la referencia fisheye se utilizan los parámetros obtenidos durante la calibración intrínseca con el modelo de Scaramuzza para obtener la posición en el plano imagen de la cámara fisheye de cada punto 3D, como se muestra en la Figura A.5.

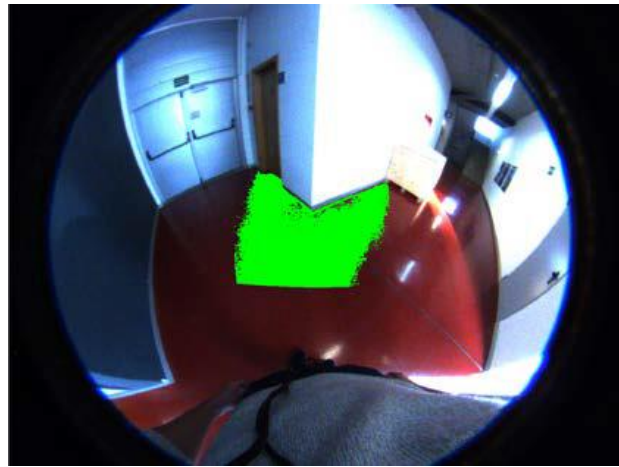


Figura 0.5. Reproyección de los puntos 3D del suelo a la imagen de fisheye

Una vez se tienen los píxeles pertenecientes al suelo en la imagen del fisheye se procede a realizar la expansión del mismo. Esto se consigue mediante la adición de regiones de la imagen cuyas características son similares al suelo inicial. Para esto es necesario segmentar la imagen en regiones para poder analizarlas posteriormente.

En este caso se ha utilizado un método de extracción de superpíxeles, que consiste en un algoritmo de muestreo impulsado por energía (SEEDS) [8] implementado en la librería OpenCV. Mediante iteraciones sucesivas los superpíxeles van modificando sus fronteras para formar regiones imponiendo la similitud de color entre fronteras, tal y como se aprecia en la Figura A.6.

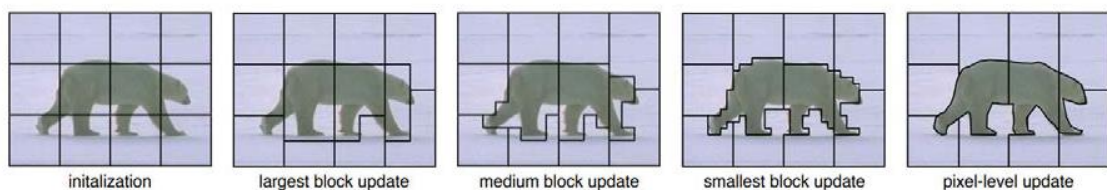


Figura 0.6. Evolución de las fronteras de los superpíxeles a lo largo de sucesivas iteraciones utilizando el algoritmo SEEDS. Imagen obtenida de [8]

Una vez segmentada la imagen se realiza la expansión del suelo mediante la comparación de histogramas de los superpíxeles vecinos a los que pertenecen al suelo. Se inicia el proceso iterativo con una semilla inicial formada por los superpíxeles que contienen puntos pertenecientes al suelo calculados con la información de la cámara de profundidad.

Una vez se tiene la semilla inicial se calcula el histograma completo de esa región y se comparan los histogramas de los superpíxeles vecinos con dicho histograma y, además, con el histograma del suelo base obtenido mediante la cámara de profundidad.

Los superpíxeles que cumplen estas dos condiciones pasan a formar parte del suelo en la siguiente iteración, de forma que al final del proceso se consigue una segmentación completa del suelo, como se observa en la Figura A.7.

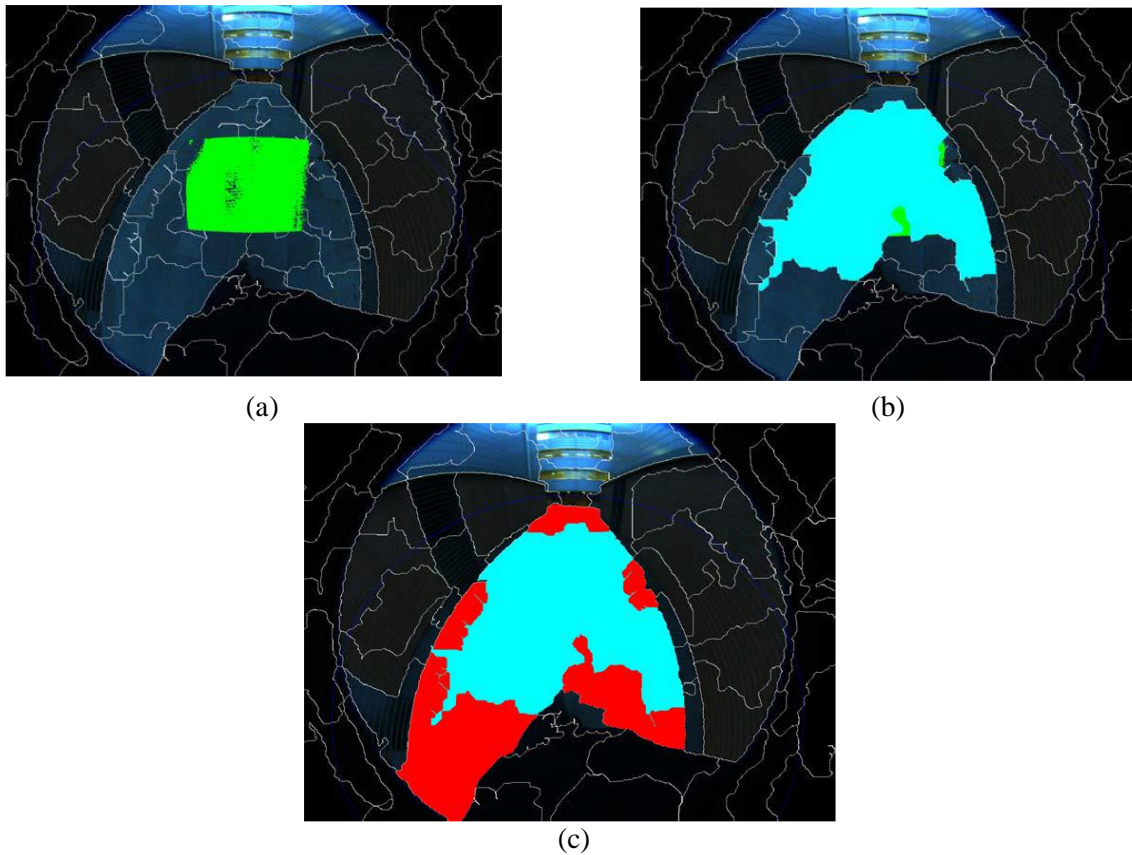


Figura 0.7. (a) Suelo detectado por la cámara de profundidad re proyectado a la imagen de fisheye. (b) Semilla inicial para la expansión del suelo (en azul). (c) Expansión de la semilla inicial (en rojo)

A.5. Uso de la información 3D en la expansión del suelo

Una vez segmentado el suelo, y para lograr mayor robustez y evitar errores en la segmentación del suelo, se propone utilizar la línea del horizonte.

En el espacio proyectivo la línea del horizonte es el lugar geométrico donde se encuentran todos los puntos de fuga de las proyecciones de las rectas horizontales en el espacio. Se corresponde con la intersección entre el plano de proyección y un plano horizontal que pasa por el origen, en este caso el centro óptico de la cámara de profundidad. Por este motivo, y teniendo en cuenta la posición de la cámara (a la altura del pecho) ningún superpíxel por encima de la misma puede pertenecer al suelo.

Para obtener la línea del horizonte se utiliza la ecuación del plano del suelo obtenida mediante la cámara de profundidad. Siendo los parámetros A, B y C las coordenadas del vector normal al plano del suelo, al pasarlo a la referencia fisheye se obtiene la normal del plano de proyección de la cámara fisheye, es decir, la línea del horizonte.

Una vez se tiene la línea del horizonte parametrizada se representa en la imagen fisheye (en azul oscuro en la Figura A.8a) utilizando el método de extracción de líneas de Bermúdez-Cameo y col. [2], explicado en capítulos anteriores. Posteriormente mediante el tratamiento de la imagen se impone que los superpíxeles por encima de la misma y los cercanos a ella pero por debajo no puedan formar parte del suelo (Figura A.8b).

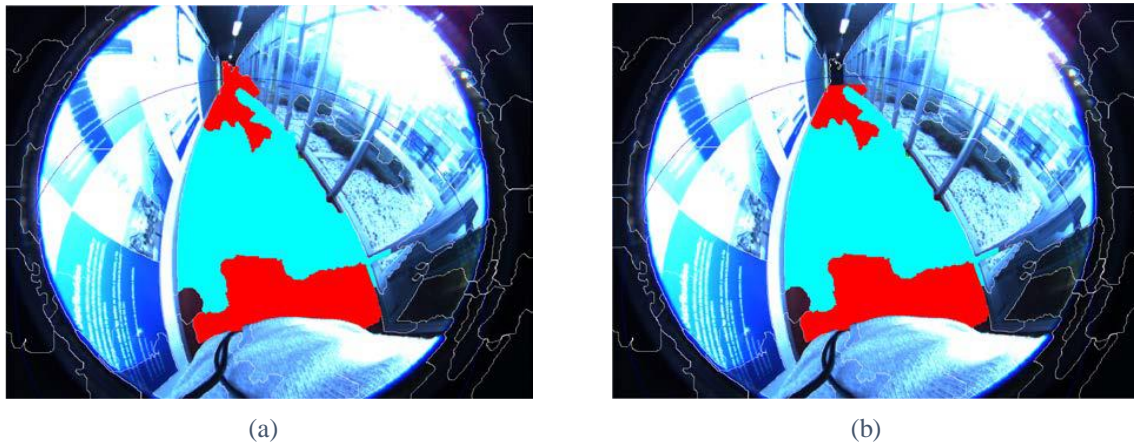


Figura 0.8. (a) Segmentación del suelo sin tener en cuenta la línea del horizonte. (b) Segmentación del suelo teniendo en cuenta la línea del horizonte

Anexo B

Software, hardware y recursos externos utilizados y cronograma del proyecto

En este anexo se detallará el software utilizado para la realización de este proyecto, indicando la versión de cada una de las librerías y del sistema operativo empleados. Además se indicará el curso realizado antes de comenzar el proyecto. Por último se detallará un cronograma con el tiempo estimado utilizado para cada una de las fases del proyecto.

Comenzando por el software utilizado:

- Para el tratamiento de la información 3D se han utilizado las librerías PCL versión 1.2.
- Para el tratamiento de las imágenes se han utilizado las librerías OpenCV versión 2.4.2.
- Por último, en cuanto al sistema operativo en el que se ha realizado todo el proceso de programación se ha utilizado Ubuntu 16.04.

En cuanto al hardware, se ha utilizado un ordenador portátil Y50-70 con las siguientes especificaciones:

- Procesador Intel Core i7-4700HQ (2.4 GHz)
- Memoria RAM 12GB DDR3
- Tarjeta gráfica Nvidia GeForce GTX 860M 2GB

Por otro lado, en cuanto al curso previo realizado para obtener un conocimiento inicial de los algoritmos más utilizados en la visión por computador, se ha hecho uso del siguiente curso:

- Introducción a la visión por computador: desarrollo de aplicaciones con OpenCV.

Este curso online ha sido elaborado por profesores de la Universidad Carlos III de Madrid y ha sido visualizado en la plataforma edX.

Por último, se va a detallar un cronograma con las fases realizadas durante el proyecto y una estimación de las horas que ha llevado realizar cada una de estas actividades.

- Formación inicial mediante el curso anterior: 20 horas.
- Estudio de trabajos previos y en especial el Trabajo Fin de Máster desarrollado en [6]: 15 horas.
- Desarrollo del primer algoritmo de detección de puertas: 120 horas
- Desarrollo del segundo algoritmo de detección de puertas: 100 horas
- Documentación de la memoria: 70 horas

Lista de figuras

Figura 2.1. (a) Imagen inicial. (b) Resultado tras la detección de rectas (en rojo) y puntos de frontera (en blanco).....	6
Figura 2.2. Rectas verticales tras el filtrado de la Figura 2.1.	7
Figura 2.3. Ejemplos de imágenes donde se ha representado la intersección de cada una de las rectas verticales con la línea del horizonte	8
Figura 2.4. (a) Máscara binaria del suelo. (b) Máscara binaria del contorno del suelo	9
Figura 2.5. (a) Ejemplo de recta vertical (en rojo) que corta en dos puntos a la máscara del contorno del suelo. (b) Ejemplo de recta vertical (en rojo) que corta en más de dos puntos a la máscara del contorno del suelo.....	10
Figura 2.6. Ejemplos de imágenes donde se ha hallado la intersección (en azul) de cada línea vertical con el suelo	10
Figura 3.1. Intersección de un rayo proyectante con el plano del suelo. Imagen obtenida de [6]	11
Figura 3.2. (a) Puntos en los que se escogerá la semilla inicial (en rojo) para cada hipótesis. (b) Semilla inicial para cada una de las hipótesis (en azul)	13
Figura 3.3. Resultado tras la expansión de cada una de las semillas iniciales de la Figura 3.2b.	15
Figura 3.4. (a) Ejemplo de rayos proyectantes y puntos correspondientes para el cálculo del cross ratio en la referencia 3D. Imagen obtenida de [11]. (b) Ejemplo de rayos proyectantes y puntos correspondientes para el cálculo del cross ratio en la referencia fisheye.	17
Figura 3.5. Segmentación de las posibles puertas (en rojo) tras aplicar el criterio del cross ratio a la Figura 3.3.....	18
Figura 3.6. Ejemplo de segmentación de una de las posibles puertas con dimensiones similares a una puerta, siendo realmente una sección de pared	19
Figura 3.7. Ejemplos de imágenes tras aplicar el primer algoritmo de detección de puertas (en rojo).....	19
Figura 4.1. (a) Ejemplo de semillas iniciales para cada una de las rectas verticales. (b) Ejemplo de semillas iniciales después de tener en cuenta los resultados obtenidos en el primer método de detección de puertas para la Figura 4.1a	21
Figura 4.2. (a) Máscara binaria a la izquierda de una recta vertical. (b) Máscara binaria a la derecha de una recta vertical	22
Figura 4.3. (a) Ejemplo de región expandida para la Figura 4.1b. (b) Región expandida tras aplicar la máscara binaria de la Figura 4.2a	23
Figura 4.4. (a) Resultado tras la expansión de las semillas iniciales de la Figura 4.2. (b) Resultado tras aplicar los criterios de cross ratio y filtrado de posibles cuadros	23
Figura 4.5. Ejemplo de un rectángulo delimitador recto (en verde) y un rectángulo delimitador rotado (en lila) para una región determinada (en amarillo)	24

Figura 4.6. Resultado del segundo método de detección de puertas para la Figura 4.1b	25
Figura 5.1. (a) Imagen del Caso 1. (b) Segmentación de la escena. (c) Rectas verticales y suelo expandido. (d) Expansión de las semillas iniciales para las hipótesis a partir de dos rectas. (e) Resultado tras filtrar las regiones. (f) Expansión de las semillas iniciales para las hipótesis a partir de dos rectas. (g) Resultado tras filtrar las regiones	28
Figura 5.2. (a) Imagen del Caso 2. (b) Segmentación de la escena. (c) Rectas verticales y suelo expandido. (d) Expansión de las semillas iniciales para las hipótesis a partir de dos rectas. (e) Resultado tras filtrar las regiones. (f) Expansión de las semillas iniciales para las hipótesis a partir de dos rectas. (g) Resultado tras filtrar las regiones	30
Figura 5.3. (a) Imagen del Caso 3. (b) Segmentación de la escena. (c) Rectas verticales y suelo expandido. (d) Expansión de las semillas iniciales para las hipótesis a partir de dos rectas. (e) Resultado tras filtrar las regiones. (f) Expansión de las semillas iniciales para las hipótesis a partir de dos rectas. (g) Resultado tras filtrar las regiones	32
Figura 5.4. (a) Imagen del Caso 4. (b) Segmentación de la escena. (c) Rectas verticales y suelo expandido. (d) Expansión de las semillas iniciales para las hipótesis a partir de dos rectas. (e) Resultado tras filtrar las regiones. (f) Expansión de las semillas iniciales para las hipótesis a partir de dos rectas. (g) Resultado tras filtrar las regiones	34
Figura 5.5. (a) Imagen del Caso 5. (b) Segmentación de la escena. (c) Rectas verticales y suelo expandido. (d) Expansión de las semillas iniciales para las hipótesis a partir de dos rectas. (e) Resultado tras filtrar las regiones. (f) Expansión de las semillas iniciales para las hipótesis a partir de dos rectas. (g) Resultado tras filtrar las regiones	36
Figura A.1. Sistema de cámaras utilizado en el proyecto, compuesto por una cámara fisheye y una cámara RGB-D.....	43
Figura A.2. Posición del conjunto de cámaras sobre el sujeto	44
Figura A.3. Observación de líneas en una escena por una cámara RGB-D y una cámara convencional con campos de visión no solapados. Imagen obtenida de [11].....	45
Figura A.4. (a) Imagen original. (b) Segmentación del plano del suelo mediante la cámara de profundidad	46
Figura A.5. Reproyección de los puntos 3D del suelo a la imagen de fisheye	47
Figura A.6. Evolución de las fronteras de los superpíxeles a lo largo de sucesivas iteraciones utilizando el algoritmo SEEDS. Imagen obtenida de [8]	47
Figura A.7. (a) Suelo detectado por la cámara de profundidad re proyectado a la imagen de fisheye. (b) Semilla inicial para la expansión del suelo (en azul). (c) Expansión de la semilla inicial (en rojo)	48
Figura A.8. (a) Segmentación del suelo sin tener en cuenta la línea del horizonte. (b) Segmentación del suelo teniendo en cuenta la línea del horizonte	49