Universidad Zaragoza
1542

Trabajo Fin de Máster

Reconocimiento y localización de objetos en imágenes de 360 grados

Autor

Luis Miguel Perez Morente

Directores

José Jesús Guerrero Campo

Jesús Bermúdez Cameo

ESCUELA DE INGENIERIA Y ARQUITECTURA

2021

TRABAJOS DE FIN DE GRADO / FIN DE MÁSTER

Escuela de Ingeniería y Arquitectura Universidad Zaragoza

DECLARACIÓN DE AUTORÍA Y ORIGINALIDAD

(Este documento debe entregarse en la Secretaría de la EINA, dentro del plazo de depósito del TFG/TFM para su evaluación).

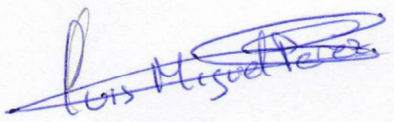
D./Dª. Luis Miguel Perez Morente ,en aplicación de lo dispuesto en el art. 14 (Derechos de autor) del Acuerdo de 11 de septiembre de 2014, del Consejo de Gobierno, por el que se aprueba el Reglamento de los TFG y TFM de la Universidad de Zaragoza,

Declaro que el presente Trabajo de Fin de (Grado/Máster) Master (Título del Trabajo)

Reconocimiento y localización de objetos en imágenes de 360 grados.

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

Zaragoza, 16 de enero de 2021



Fdo: Luis Miguel Perez Morente

Agradecimientos

Me gustaría agradecer su paciencia, ayuda y sobre todo su apoyo a mi familia y a mi pareja.

A mis amigos del instituto, por su continua presencia y apoyo permanente, y a los que han ido apareciendo su creencia constante en mí.

Por último, su apoyo a Catedra Mobility University of Zaragoza.

Resumen

En este trabajo, se ha desarrollado un software capaz de reconstruir el layout de una habitación, colocando los objetos presentes en la misma utilizando una única imagen panorámica de 360 grados de campo de vista horizontal. Dicho programa es la continuación de dos trabajos desarrollados anteriormente, encargados de devolver la forma del layout de la habitación y de detectar los objetos presentes en la escena. Combinando razonamientos geométricos con restricciones impuestas por el entorno, siendo la principal restricción, la suposición de Mundo Manhattan, de forma que se consigue el posicionamiento de los objetos en un entorno reconstruido, sin ningún otro dato, a parte de los extraídos de las imágenes.

Abstract

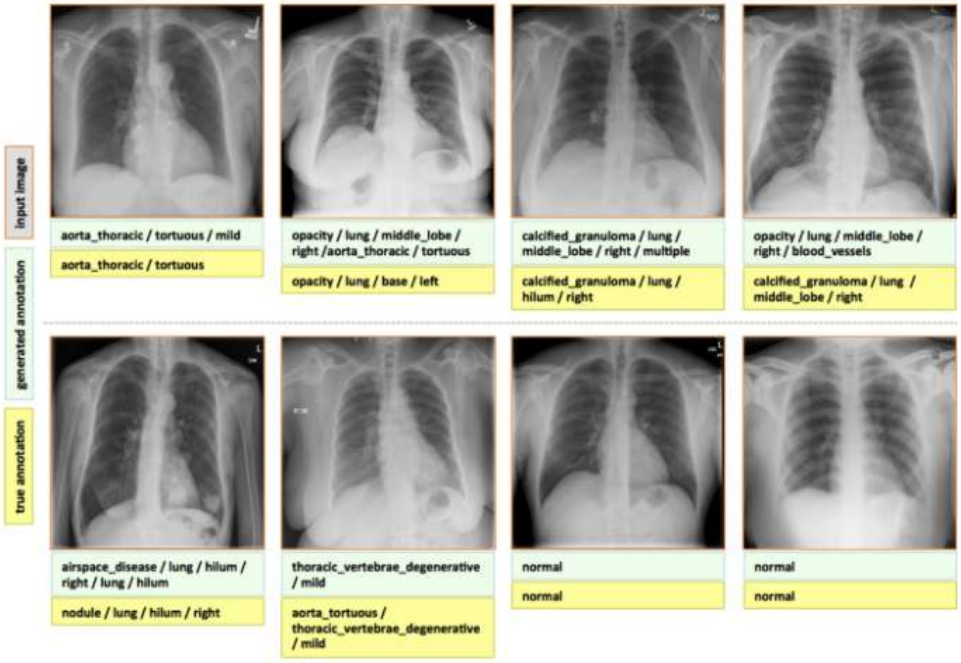
In this work, we have developed a software that is able to get the layout recovery of indoor room and also to place the recognized objects from a single 360 degrees panoramic image. This software is a continuation of two previous projects,which get the layout and detects the objects in the scene. Combining geometric reasoning and restrictions from the enviroment, and assuming the Manhattan World hypotheses, we get the positioning of the objets in a scene recovered without further information except the one obtained from the image.

Índice general

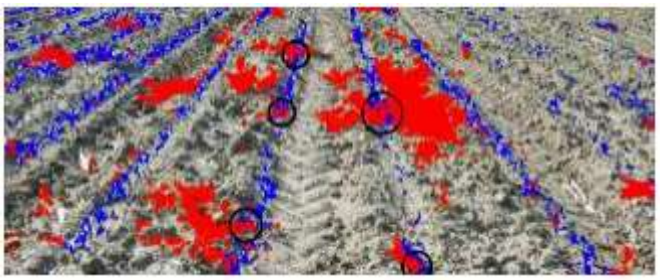
Capítulo 1

Introducción

La visión, es el sentido que más información proporciona del entorno que le rodea al ser humano. La visión por computador, por lo tanto, es una parte esencial para la extracción de información a través del estudio de imágenes obtenidas con una cámara. Hay multitud de ejemplos de aplicaciones de la visión por computador, como puede ser el uso de la misma para el cuidado de la salud, a través del diagnóstico de pacientes a partir de radiografías y otras pruebas consistentes en imágenes, consiguiendo resultados hasta 150 veces más rápidos que una persona, habiéndose utilizado imágenes de rayos X del tórax de pacientes para lograr un diagnóstico de la Covid-19 [1]. También tiene usos en agricultura para mejorar la producción de la misma y reducir el uso de herbicidas y otras sustancias similares, gracias a la detección automática de los mejores puntos de rociado [2].



a Diagnóstico de enfermedades



b Detección de malas hierbas

Figura 1.1: Ejemplos de aplicaciones de la visión por computador

Uno de los principales problemas, en este campo, es la reconstrucción de entornos tridimensionales a partir de una única imagen, habiendo sido abordado en multitud de ocasiones, utilizando distintos enfoques y distintos tipos de imágenes de entrada.

En los últimos años, ha crecido el interés en la reconstrucción de interiores, debido a todas sus aplicaciones, como son; la navegación en interiores, robótica, realidad aumentada y virtual, o la ayuda a personas con visión reducida.

En [3] se utiliza la visión por computador para la detección de los límites de puertas o ascensores en imágenes interiores para facilitar la movilidad de personas ciegas.

Se utiliza también en entornos exteriores para la reconstrucción del 3D, facilitando así la navegación de coches autónomos o en coches novedosos para el aparcamiento asistido, en Tesla, por ejemplo, se usa de forma muy exitosa en la seguridad de sus vehículos.

También es conocida la aplicación de Google Tango, que aborda la navegación por interiores.

1.1 Estado del arte

En este apartado se va a proceder a presentar algunos trabajos que enfrentan el mismo problema que en este, o trabajos que afrontan problemas similares. Estos trabajos se diferencian entre sí en distintos aspectos, como son, el tipo de imágenes que utilizan, o en el método usado para resolver el problema.

En [11, 12], Hedau utiliza imágenes convencionales y simplifica el problema de reconstrucción de un layout. Supone la habitación como una caja de cuatro paredes y utiliza un mapa de características geométricas para la detección de objetos. Para los objetos también asume una forma de caja orientada según las direcciones principales de la escena.

PanoContext trabaja en [13] con panoramas para estimar el layout de la habitación y las cajas asociadas a cada objeto, también supone las habitaciones como una caja tridimensional de cuatro paredes.

En [14], se trabaja con panoramas, de los cuales se extrae la geometría de una escena interior. Para ello busca los elementos estructurales básicos para calcular el layout. Además, se utilizan partes del panorama de las que se extraen señales tales como líneas, puntos de fuga u orientaciones.

En [15], Furlan emplea la asunción de mundo Manhattan junto al conocimiento del recorrido hecho por una cámara, a partir de estos datos logra la reconstrucción del layout visto casi a frecuencia de video.

1.2 Objetivos

En este trabajo, se propone un método que combina el razonamiento geométrico con restricciones debidas al tipo de escena para reconstruir una escena interior, a partir de una única imagen panorámica.

El motivo por el que se usan las imágenes panorámicas es su mayor campo de visión, que permite obtener información de toda la escena a diferencia de lo que ocurre en una imagen convencional, aunque, por otro lado, hay que tener en cuenta su mayor complejidad.

Al procesar estas imágenes, obtenidas de la base de datos pública SUN360 [4] con los trabajos [5] y [6] se consiguen las imágenes de entrada que se utilizarán para reconstruir la escena representada. Las imágenes obtenidas representan las esquinas de la habitación y los objetos detectables de la misma. A partir de estas imágenes se quiere conseguir una reconstrucción total de la escena, utilizando un algoritmo totalmente desarrollado durante este trabajo, aprovechando la asunción de mundo Manhattan [10]. Esta asunción, presupone la existencia de tres direcciones principales ortogonales entre sí, que rigen el posicionamiento de los objetos en la escena.

Capítulo 2

Trabajo previo

Este trabajo, parte de dos anteriores presentados en [5] y en [6], se va a recoger aquí un resumen de ambos trabajos y sus metodologías, de forma que esta memoria sea auto contenida y no sea necesario nada más para entenderla totalmente.

Figura 2.1: Panorama original escena de interior

En el primero de ellos [5], se utilizan técnicas de aprendizaje profundo para estimar el layout de una habitación. Una red con arquitectura encoder decoder basada en U-Net [9] predice dos imágenes: un mapa de probabilidad de contornos de la imagen pertenecientes a elementos estructurales de la escena, (paredes, techo y suelo) y un mapa de probabilidad de las esquinas que definen el layout de la habitación. Estas imágenes son usadas para inferir una geometría 3D básica y sin escala de la escena suponiendo que existen tres direcciones dominantes perpendiculares.

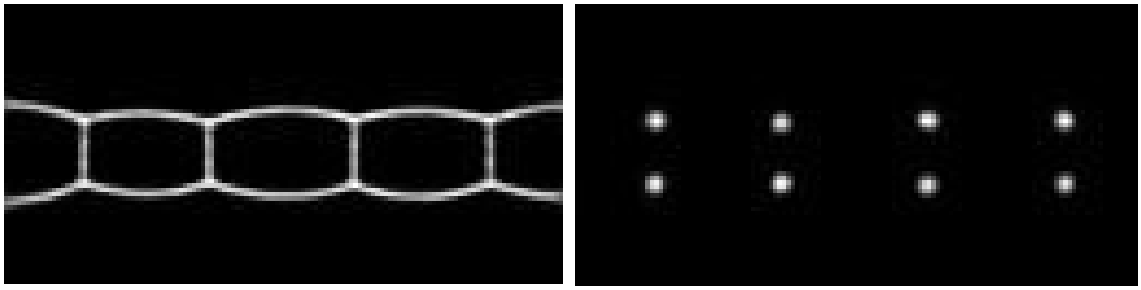


Figura 2.2: Salida de [5]

En el segundo, se desarrolla un modelo basado en el aprendizaje profundo llamado Panoramic BlitzNet, utilizando como base la red BlitzNet, y adaptándola para trabajar con imágenes equirectangulares, aborda las tareas de detección de objetos y segmentación semántica en entornos de interior. Está compuesto por una red convolucional compartida prácticamente en su totalidad por las dos tareas, que cuenta con conexiones skip y que realiza reconocimiento multi-escala, analizando además en profundidad el impacto de las convoluciones equirectangulares. De este trabajo se recibe una imagen panorámica que presenta una máscara, de un color previamente definido, sobre los pixeles de cada objeto.

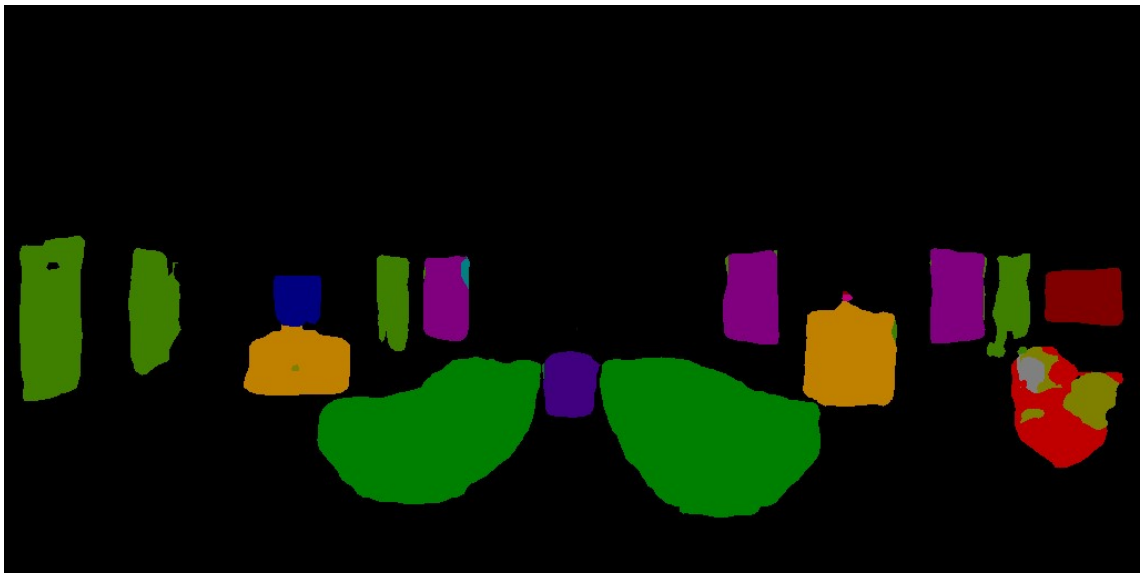


Figura 2.3: Salida de [6]

2.1 Imágenes panorámicas

En todo el proceso seguido para la reconstrucción del layout con el posicionamiento de los objetos presentes en la escena se utilizan imágenes panorámicas, en lugar de imágenes convencionales, esto es debido al mayor campo de visión de las imágenes panorámicas, las cuales presentan 360º de campo de vista horizontal por 180º de campo de vista vertical.

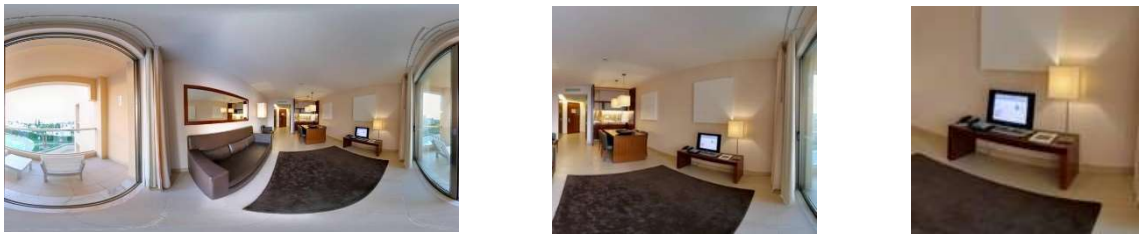


Figura 2.4: Diferencias entre los campos de vista de una imagen panorámica, lo que ve una persona, y una imagen convencional

La particularidad que presenta este tipo de imágenes, radica en que debido al tipo de proyección son imágenes en las cuales es más difícil comprender proporciones o la distribución de las escenas mostradas.

Por un lado, las líneas verticales de la escena se mantienen verticales mientras que por otro las líneas horizontales se convierten en líneas curvas, la curvatura de las mismas es mayor conforme se alejan de la posición central de la imagen.

Debido a las diferencias entre lo que un ser humano es capaz de ver, y la información captada en una imagen panorámica, sucede, que las imágenes pueden ser confusas para las personas al verlas, ya que, por ejemplo, una persona no es capaz de observar lo que hay a su espalda.

2.2 Proyección esférica

Debido a la importancia del tipo de proyección utilizado, en las imágenes a la hora de desarrollar el algoritmo, ya que ha complicado algunos aspectos del desarrollo, se considera necesaria la explicación de dicho modelo de proyección. Este tipo de proyección es el visto normalmente en los mapas del mundo, es una proyección en el que las zonas superior e inferior de la escena se encuentran deformadas.

Figura 2.5: Tipos más comunes de proyecciones, en orden: Esférica, cilíndrica, rectilínea y ojo de pez

En este apartado por tanto se va a detallar el paso desde coordenadas esféricas a coordenadas en la imagen. Si se define una imagen panorámica de *W* x *H* pixeles, siendo W la anchura y H la altura de la imagen panorámica, donde W representa 360 grados de campo de vista horizontal y H 180 grados de campo de vista vertical, de aquí se puede extraer además que $W = 2H$, considerando el origen de coordenadas en el centro de la imagen se tiene que el origen será $\left(\frac{W}{2}, \frac{H}{2}\right)$.

Un sistema de coordenadas esférico determina la posición de un punto mediante una distancia y dos ángulos, por lo tanto, un punto P (*X, Y, Z*) queda completamente representado a partir de una distancia y dos ángulos, colatitud (θ) y azimut (φ), el colatitud tiene un recorrido desde -90 grados a +90 grados, y el azimut cubre desde -180 grados hasta +180 grados.

Los pasos a seguir para la transformación de las coordenadas de un punto P a coordenadas esféricas se detallan a continuación:

El primer paso es la proyección del punto P (*X, Y, Z*) sobre la esfera de radio unidad p (*x, y, z*)

$$(x, y, z) = \frac{1}{\sqrt{X^2 + Y^2 + Z^2}}(X, Y, Z)$$

Calculo de las coordenadas esféricas (la distancia es igual a uno)

$$(\sin\varphi * \cos\theta\,, \cos\theta * \cos\varphi\,, \sin\theta) = (x, y, z)$$

Obtención de las coordenadas en la imagen (*u, v*)

$$(u, v) = \left(\varphi\frac{W}{2\pi} + \frac{W}{2}, \theta\frac{H}{\pi} + \frac{H}{2}\right)$$

Finalmente, los ángulos θ y φ quedaran

$$\varphi = \frac{\left(u - \frac{W}{2}\right)2\pi}{W}$$

$$\theta = \frac{\left(v - \frac{H}{2}\right)\pi}{H}$$

Este proceso, va a emplearse mucho a lo largo del trabajo, ya que es muy necesario para el posicionamiento de información de la imagen en una reconstrucción tridimensional de la misma.

Además de este método, para conseguir la colocación de cualquier punto reconocible de la imagen en el 3D hace falta conocer algún otro valor para obtener a que distancia del origen está colocado cada punto, para cada situación se explicará más adelante el proceso de actuación utilizado.

2.3 Clasificación de coordenadas y objetos

Para simplificar el entendimiento de los distintos tipos de datos que se van a utilizar, ya que puede llegar a ser un tema complejo, se va a hacer un pequeño desglose.

Se tienen tres tipos de coordenadas, que son, tridimensionales, proyecciones en la esfera unidad, y bidimensionales o en imagen. Las primeras son las coordenadas cartesianas, que definen cualquier punto del espacio *(X, Y, Z)*. Las segundas son las proyecciones de estas coordenadas sobre una esfera de radio unidad centrada en la cámara *(x, y, z)*. Las últimas son las coordenadas que corresponden a un punto del espacio en la imagen panorámica (*u, v*), estas coordenadas al igual que las coordenadas proyectadas representan el rayo que une un punto del espacio con la cámara.

En cuanto a los objetos con los que se trabaja, se separan en dos categorías principales, planos y volumétricos.

Los objetos planos son todos aquellos que pueden aproximarse a una lámina, como ventanas, puertas o espejos, son objetos que se encuentran colocados sobre las paredes. Los objetos volumétricos son aquellos que como su nombre indica ocupan un volumen, por ejemplo camas o mesas.

Capítulo 3

Reconstrucción de la escena

En este capítulo, se va a explicar el punto de partida de este trabajo, junto a las distintas dificultades encontradas a la hora de conseguir reconstruir el entorno tridimensional visto en una imagen panorámica.

Los pasos a seguir, para lograr obtener una reconstrucción tridimensional de la escena son:

Primero recuperar la forma y dimensiones de la habitación, además de la orientación de la misma. Segundo, hay que aislar cada objeto por separado para obtener su forma y su posición en la habitación, además del tipo de objeto con el que se está trabajando en cada momento. Tercero, cada objeto debe ser colocado en la reconstrucción de la habitación, la forma de hacerlo dependerá del objeto que se esté colocando en cada momento.

En el planteamiento inicial, se va a explicar la primera aproximación para lograr la reconstrucción del layout, junto con sus problemas y las soluciones de los mismos; para más tarde, ver con mayor detalle cómo se ha acabado solucionando el problema.

3.1 Planteamiento inicial

Se inicia el proceso de reconstrucción de la habitación, a partir de los resultados recibidos de [5] (Figura 2.2), a partir de esta imagen, se busca el pixel correspondiente a cada esquina de la habitación, para esto se busca el centroide de cada grupo de pixeles, cuyo valor es mayor de un determinado umbral, para evitar posibles problemas que se pueden dar en el caso de que una de las esquinas sea reconocida en uno de los extremos laterales de la imagen, esta se concatena tres veces, de esta forma en la parte central de la nueva imagen se tendrán las esquinas de la habitación perfectamente colocadas en cualquier situación, tras esto se guarda cada centroide correspondiente a una esquina obteniendo así un pixel para cada esquina de la habitación.

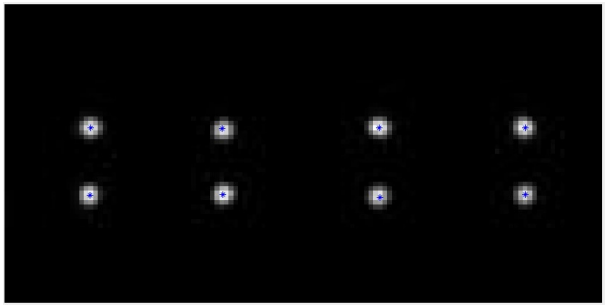
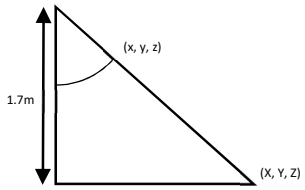


Figura 3.1: Corner map con las esquinas de la habitación

Una vez conocidos estos pixeles, utilizando el método visto en la sección 2.2, se pueden obtener las proyecciones en la esfera de radio unidad de cada una de las coordenadas de las esquinas de la habitación, tras lo cual solo será necesario un dato de estos puntos para poder situarlos en su posición real, en este caso lo que se va a hacer es suponer que la cámara utilizada para hacer la fotografía se encuentra a la altura media de una persona, a 1.7 metros del suelo, de esta forma la coordenada *Z* de los puntos pertenecientes al suelo será -1.7 metros. Conocido este dato se procede de la siguiente manera



$$(X,Y,Z)_{suelo} = t_{suelo} * (x,y,z)_{suelo}$$

$$t_{suelo} = -1.7/(z)_{suelo}$$

Tras esto, considerando que los puntos del techo están siempre en vertical sobre los del suelo y que por tanto las coordenadas *X* e *Y* de los puntos serán las mismas

$$t_{techo} = (X)_{suelo}/(x)_{techo}$$

$$(X,Y,Z)_{tech} = t_{tec} * (x,y,z)_{techo}$$

Por último, se asegura que todos los puntos pertenecientes al techo se encuentran a la misma altura, utilizando la altura media de los puntos obtenidos.

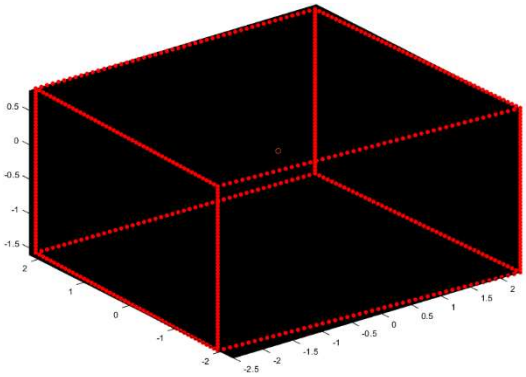


Figura 3.2: Layout obtenido

Una vez conocida la forma de la habitación, se inicia el proceso para posicionar los objetos, este proceso se inicia con la obtención del número de objetos reconocidos en la escena de entre los catorce tipos buscados en [6], estos tipos son: pinturas, camas, mesas, espejos, ventanas, cortinas, sillas, lámparas, sofás, puertas, armarios, mesillas de noche, televisiones, y estanterías.

Dada la máscara de cada uno, haciendo uso de la misma, además de los puntos de fuga se buscan las líneas que siguen los puntos de fuga de la imagen,

intentando de esta manera obtener la forma regular que mejor comprende la máscara de dicho objeto. De esta forma se obtienen cuatro líneas, las cuales siguen las direcciones principales de la escena, y que encuadran cada objeto en la misma.

La red detecta dos tipos de objetos principalmente, objetos planos, como puertas o ventanas, y objetos volumétricos, como camas o armarios. En el caso de los objetos planos, las líneas coinciden normalmente con el marco de dicho objeto, quedando así completamente definido. En el caso de los objetos volumétricos, y al escoger las cuatro líneas de mayor longitud que se adaptan a la máscara del objeto, suelen ser definidos a medias, pudiendo observarse solo una parte de los bordes de dicho objeto señalados por estas líneas.

Para colocar los objetos en el modelado tridimensional de la escena, de cuyo layout se tienen las coordenadas, se trabaja de forma diferente dependiendo de si los objetos son planos o tridimensionales.

En cuanto a los objetos planos, se supone que todos ellos están colocados en los muros de la habitación, por ello el primer paso es obtener, en cuál de estos muros se encuentra, a partir de la dirección normal a dicho objeto y a cada muro de la habitación. Una vez se conoce el muro en el que está colocado dicho objeto, el procedimiento es similar al usado para calcular la posición de los vértices de la habitación, para cada punto perteneciente al contorno del objeto se sigue el mismo procedimiento, el primer paso consiste en convertir las coordenadas de cada punto en la imagen a coordenadas tridimensionales proyectadas en la esfera unidad. El siguiente paso es obtener las coordenadas tridimensionales de cada punto, para ello, conocida la distancia del origen al muro en el que se encuentra el objeto en dirección perpendicular al mismo, se tiene que las coordenadas cartesianas de cada punto serán:

$$(X, Y, Z)_{punto} = abs(t_{punto}) * (x, y, z)_{punto}$$

$$t_{punto} = dist/(n)_{punto}$$

Siendo n la coordenada en dirección normal al objeto.

En cuanto a los objetos volumétricos, el procedimiento para colocarlos en la escena es más complejo, pero bastante similar, para este tipo de objetos se buscan las líneas de definen el objeto y con ayuda del suelo y las paredes se coloca dicho objeto contra la pared en el lugar donde la máscara entra en contacto con la misma. Utilizando de igual manera las ecuaciones antes descritas, para relacionar los puntos con las superficies presentes; más tarde y con ayuda de los puntos calculados, se calcula la bounding box que mejor se adapta a los puntos obtenidos colocando luego dicha bounding box en el entorno tridimensional, de forma que se obtiene la superficie en planta al relacionar el objeto con el suelo y la altura al relacionar el objeto con la pared detrás del mismo.

3.2 Problemas presentes

Este sistema presentaba ciertos problemas, en parte debidos a que las líneas obtenidas para definir cada objeto a menudo no seguían de forma precisa las direcciones esperadas y no aislaban el objeto tan bien como sería deseable, motivo por el cual al presentar los objetos en el modelo tridimensional muchos acababan deformados. Debido al funcionamiento del sistema con los objetos volumétricos tal y como estaba planteado, aunque no acabado, cualquier objeto que no estuviera situado junto a una pared era deformado y se colocaba de forma errónea.

Por estos problemas fundamentales, junto a algunos otros de menor escala, se procedió al desarrollo de una solución alternativa, aunque similar en algunos aspectos para lograr obtener una representación tridimensional a partir de un único panorama de la habitación.

3.3 Planteamiento final

Inicialmente, la reconstrucción del layout se plantea de la misma forma con una corrección final la cual se presentará en la subsección 3.4.

En cuanto al reconocimiento y posicionamiento de los objetos, el primer paso consiste en la manera de aislar la máscara de cada objeto, debido a que se supone que todos los objetos de la escena están colocados según las direcciones principales de la escena, como ya se ha dicho antes, el primer paso era encontrar líneas paralelas a estas direcciones y que enmarcan la máscara del objeto. Debido, a que no se pueden obtener las ecuaciones que representen una recta en el espacio tridimensional en las coordenadas de la imagen de forma sencilla, ya que la relación entre las coordenadas es no lineal, el proceso seguido para obtener estos límites, consiste en el cálculo de líneas que siguen las direcciones principales en 3D, la proyección de las mismas sobre la imagen, y la obtención de las líneas en contacto con la máscara, obteniendo los siguientes resultados:

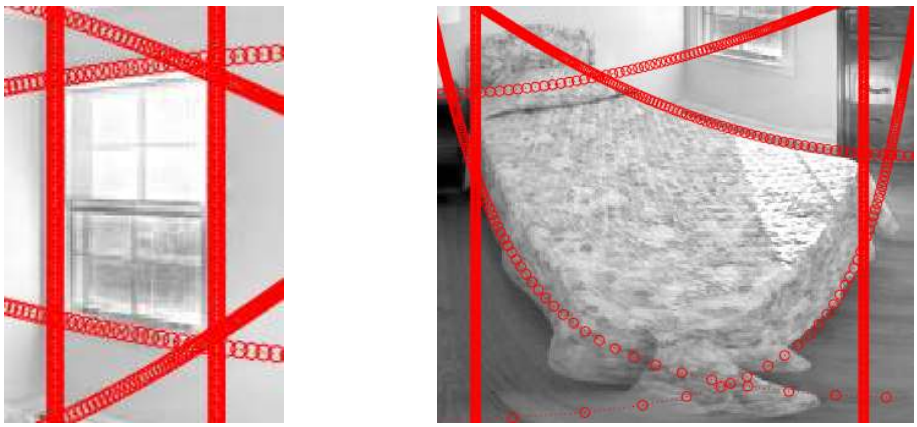


Figura 3.3: Limites obtenidos para cada tipo de objeto, en orden objeto plano y objeto volumétrico.

El proceso seguido es un proceso iterativo, consistente en el cálculo de líneas que siguen las direcciones principales de la escena, y que más tarde son proyectadas sobre la imagen para comprobar si entran en contacto con la máscara que se trata de aislar.

Se calcula un número determinado de líneas por cada pared de la habitación, hasta que se obtiene la línea, que, al proyectarla, entra en contacto con la máscara, una vez se tiene esta línea este proceso se inicia de nuevo con la misma dirección principal, pero en sentido contrario para obtener el limite contrario de la máscara.

Tras esto, se sigue el mismo proceso para cada dirección principal, obteniendo finalmente las seis líneas que delimitan cada objeto.

En el caso de los objetos planos, se tienen dos líneas más de las que se necesitan. Se tienen dos líneas horizontales que no son necesarias, éstas se descartan, sabiendo sobre que pared se encuentran dichos objetos, las líneas perpendiculares a la pared sobre la que se encuentran son descartadas.

Una vez conocidas las líneas que encuadran cada máscara, el siguiente paso es la obtención de los vértices de dicho objeto, para ello la forma más sencilla es la de calcular las intersecciones entre cada par de líneas, ya que cada caso tiene dos intersecciones, a la hora de elegir cuales de entre todas las intersecciones son las que se buscan, vuelve a depender del tipo de objeto con el que se está trabajando, ya sea plano o volumétrico.

En el caso de ser un objeto plano, se buscan cuatro vértices al igual que se buscaban solo cuatro líneas, éstas serán siempre dos líneas verticales, y dos líneas horizontales, y los cuatro vértices escogidos serán intersecciones de estas cuatro líneas, para cada par de líneas se obtienen dos intersecciones debido al tipo de proyección, de entre las intersecciones entre estas dos líneas verticales y dos horizontales, se escogen aquellas que más cerca se encuentren de la máscara. Para ello se calcula el centroide de la máscara y las distancias entre cada vértice obtenido y dicho centroide.

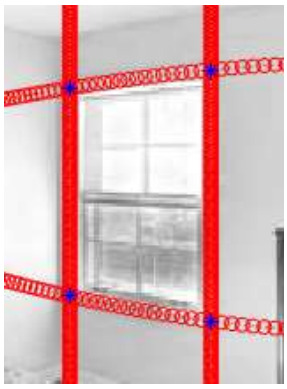


Figura 3.4: Vértices objeto plano

En el caso de los objetos volumétricos se buscan seis líneas, dos para cada dirección principal, y seis vértices, intersecciones entre estas seis líneas, el

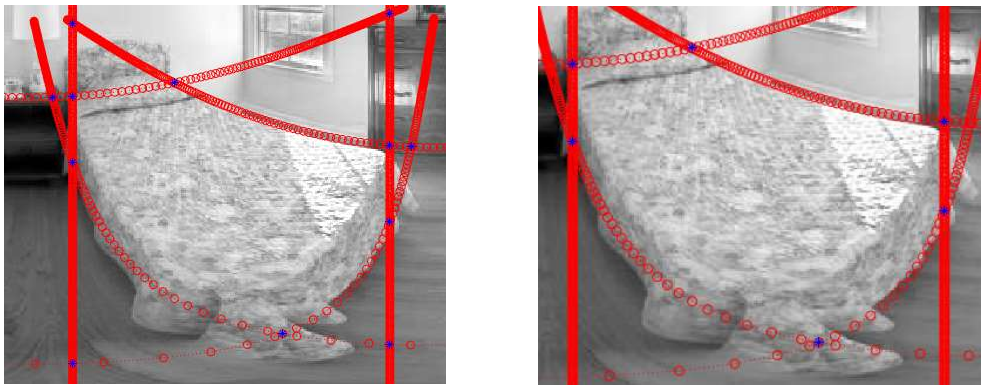


Figura 3.5: Vértices obtenidos de la intersección de las líneas y vértices buscados

problema, es que como antes al calcular estas intersecciones se obtienen más de seis vértices, en este caso no solo debido a las dos intersecciones entre cada par de líneas, si no también debido a la intersección entre proyecciones de rectas que no corresponden con vértices del objeto. Para seleccionar de entre todas las intersecciones aquellas que corresponden a los vértices visibles del bounding box del objeto, se parte de la intersección de cada línea con la máscara y se busca el punto de intersección más cercano a lo largo de la línea en ambas direcciones. De esta forma, se obtiene una lista redundante de vértices del bounding box del objeto. Para evitar posibles errores se establecen algunos parámetros, como son que los vértices pertenecientes a las líneas verticales saldrán siempre de estas líneas, ya que tal y como están definidas se evitan errores. Además, cualquier punto que no se encuentre entre estas dos líneas verticales es descartado, quedando de esta forma los seis puntos buscados.

Ahora que ya se tienen los vértices, que definen cada objeto, el siguiente paso es colocar estos objetos en el modelo 3D de la habitación. De la misma manera que en los pasos anteriores, el proceso dependerá del tipo de objeto. Primero, para los objetos planos, al estar siempre situados en las paredes, el primer paso es obtener sobre que pared está situado, para lo cual se compara la máscara del objeto con la superficie perteneciente a cada pared, aquella pared que tenga una mayor superficie en común será la buscada. Una vez la pared es conocida y utilizando la distancia perpendicular, desde la pared hasta el origen de coordenadas, se obtiene la proporción entre la coordenada de los vértices de dicho objeto en la esfera unidad, en la dirección marcada por la perpendicular a la pared. Con esta proporción se calculan las coordenadas tridimensionales del objeto en el modelo 3D de la habitación.

$$(X, Y, Z)_{contorno} = t_{punto} * (x, y, z)_{contorno}$$

$$t_{punto} = {}^{D}/_{d}$$

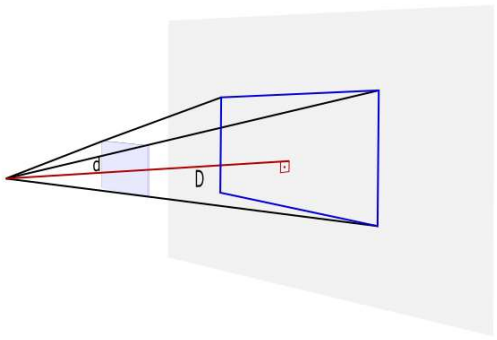


Figura 3.6: Posicionamiento de un objeto plano.

Para el posicionamiento de objetos volumétricos, se supone que están colocados sobre el suelo. Debido a que se tienen seis vértices de los ocho que definirían un objeto completo, y no se sabe cuáles de ellos pertenecen al plano inferior del objeto y cuales, al plano superior, se deben seguir ciertos pasos para lograr su posicionamiento. Primero se seleccionan los cuatro vértices que se encuentran en las dos líneas verticales, que limitan el objeto en la imagen. Los dos inferiores serán puntos del objeto que están sobre el suelo, es decir, su coordenada Z será la del suelo, de esta manera se pueden obtener sus coordenadas. Los otros dos irán sobre estos puntos con mismas coordenadas X e Y, y la coordenada Z que corresponda [Figura 3.7 (a)].

$$(X,Y,Z)_{punto\ inferior} = t_{inferior} * (x,y,z)_{punto\ inferior}$$

$$t_{inferior} = -1.7/(z)_{punto\ inferior}$$

$$(X,Y,Z)_{punto\ superior} = t_{superior} * (x,y,z)_{punto\ superior}$$

$$t_{superior} = (X)_{punto\ superior}/(x)_{punto\ superior}$$

Faltan ahora cuatro puntos por posicionar, de los cuales solo se conocen dos, no sabiendo si son puntos pertenecientes a la base del objeto o a la parte superior. Por ello en el segundo paso se presupone que ambos pertenecen al suelo y se comparan las coordenadas obtenidas X e Y, con las pertenecientes a los puntos ya colocados. Si la posición obtenida para los puntos se diferencia con la de los puntos ya colocados menos de un cierto umbral, se da por correcta la suposición de que los puntos pertenecían al suelo, en caso contrario se pone a la altura de los puntos superiores [Figura 3.7 (b)].

El tercer paso coloca los dos puntos restantes, en caso de que en el paso anterior un punto se coloque en el suelo se coloca uno con las mismas coordenadas X e Y a la altura de la parte superior del objeto, en caso contrario el punto se coloca a la altura del suelo.

Por último, se aplican ciertas restricciones a las coordenadas calculadas de cada objeto, de esta forma si las coordenadas de un punto sobrepasan los límites de la habitación dicho punto será proyectado sobre la pared, y se iguala la altura de todos los puntos de la parte superior del objeto.

Siguiendo este proceso para cada uno de los objetos detectados, se consiguen resultados como los mostrados en la [Figura 3.8].

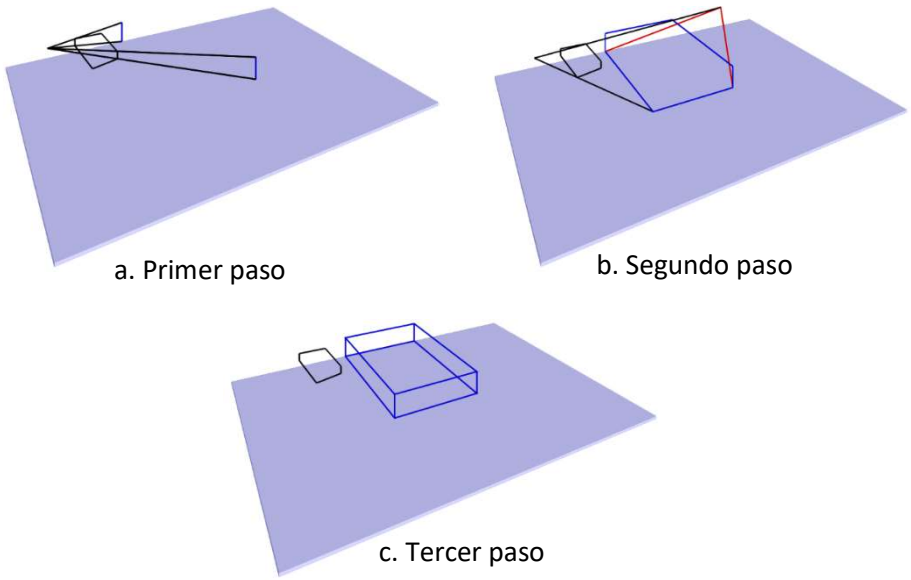


Figura 3.7: Procedimiento de colocación de los objetos volumétricos.

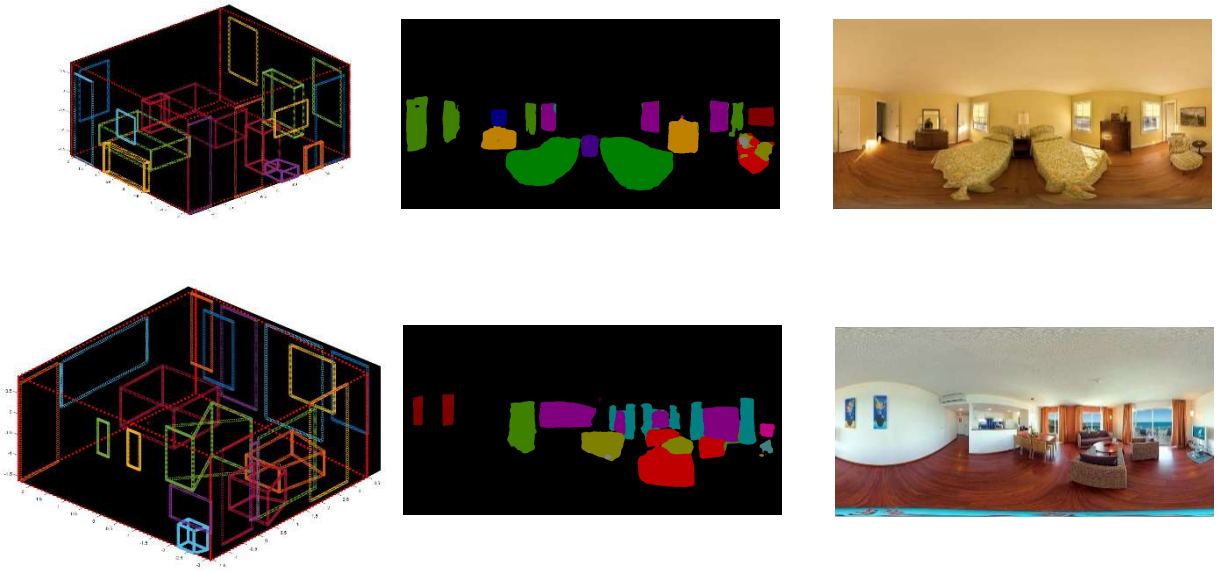


Figura 3.8: Ejemplos de funcionamiento, izquierda reconstrucción, centro segmentación, derecha original.

3.4 Orientación de la escena

Para que el programa funcione correctamente, debe cumplirse al menos que las paredes del entorno sean perpendiculares a los ejes cartesianos.

A pesar de que las imágenes con las que se trabaja, respetan siempre la dirección vertical, a veces presentan giros respecto del eje Z. Para eliminar estos giros, de forma que las paredes queden en dirección perpendicular a cada eje, se utilizan los puntos de fuga como matriz de rotación, donde:

$$R_Z = [Vp_1 \quad Vp_2 \quad Vp_3]$$

Siendo solo un giro respecto del eje Z se tiene finalmente:

$$R_Z = \begin{bmatrix} Vp_{1,1} & Vp_{2,1} & 0 \\ Vp_{1,2} & Vp_{2,2} & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Utilizando esta matriz de rotación, se giran las coordenadas obtenidas del corner map [Figura 3.1] de forma que al reconstruir el layout, y obtener las paredes de la habitación, éstas quedan perpendiculares a los ejes. [Figura 3.2]

Como la bounding box de cada objeto se obtiene con líneas paralelas, que siguen las direcciones marcadas por los ejes, es importante girar también la imagen de segmentación [Figura 2.3] para evitar errores debidos a la orientación

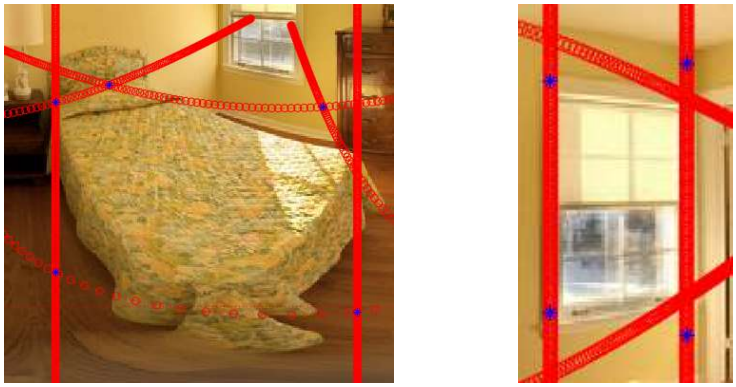


Figura 3.9: Bounding box de una imagen mal orientada.

como los que se ven en [Figura 3.9], el ángulo que debe girarse se obtiene de la matriz de rotación, ya que la misma está definida como se ve a continuación:

$$R_Z = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Y siendo las imágenes de segmentación, imágenes con una resolución de 1024x512 pixeles, donde los 1024 pixeles corresponden a 360º, se puede obtener cuantos pixeles hay que girar la imagen para orientarla de forma correcta.

Para ello, y una vez obtenido el ángulo θ se obtienen los pixeles que se deben mover en la imagen de la siguiente manera:

$$Pixeles = \frac{1024 * \theta}{2 * \pi}$$

Este número de pixeles, será el que se debe coger del extremo izquierdo de la imagen para colocar en el derecho, obteniendo así la imagen girada el ángulo requerido.

Figura 3.10: Giro necesario para colocar las paredes de forma perpendicular a los ejes.

Capítulo 4

Experimentación

Debido a la importancia de los resultados obtenidos en [5] y [6], se va a dedicar un pequeño apartado a comentarlos brevemente de forma que se puedan poner en contexto los valores presentados por este trabajo.

En [5] se puede ver en el capítulo 6, en el cual se explican de forma muy clara los experimentos realizados y los valores obtenidos, que se alcanza una precisión media para la predicción y detección del layout de una habitación a partir de una única imagen panorámica de un 90.5%.

En [6], el capítulo 5 se dedica por completo a la evaluación y presenta una explicación muy completa de los experimentos realizados para valorar el rendimiento de dicho trabajo. Se debe tener en cuenta la precisión y número de objetos recuperados, en función del umbral escogido para determinar si un objeto es detectado o no (Figura 5.4 [6]). Se puede ver que, para umbrales más bajos, la precisión obtenida será menor, pero se detectarán casi todos los objetos presentes en la imagen, mientras que para un umbral más elevado los objetos se detectan con una precisión muy alta, aunque solo algunos de ellos. Se entiende como umbral el número de pixeles que deben ser vistos para considerar que un objeto es detectado. Se puede obtener de dicha imagen que para un umbral de detección de un 0.5 se obtiene una precisión de un 80% detectando aproximadamente la mitad de los objetos presentes en la escena.

Por último, se debe comentar que de la base de datos utilizada en [5] y en [6], se deben seleccionar imágenes con las que trabajar. De todas las imágenes procesadas en ambos trabajos se ha escogido una pequeña selección de veinte panoramas, en parte debido a dificultades causadas por la situación excepcional vivida este año. Pero, sobre todo, debido a ciertos requerimientos del programa para trabajar adecuadamente, para conseguir resultados positivos, se necesitan imágenes que salgan de [6] habiéndose detectado cada objeto con una sola máscara y no en varias divididas, como se ve en [Figura 4.1]. En caso de que un objeto sea cubierto por más de una máscara, es imposible saber que las diferentes máscaras se refieren al mismo objeto, intentándose colocar distintos objetos del mismo tipo, en lugar de uno solo.

Figura 4.1: Resultado desechado

Una vez seleccionadas las imágenes que cumplan este requisito, se procesan para obtener el layout de la habitación de [5] y esos son los resultados a partir de los que se trabaja.

La forma más adecuada de valorar los resultados, sería la comparación de las coordenadas tridimensionales de los objetos colocados en la reconstrucción, con las coordenadas exactas de los objetos vistos en la escena, por desgracia, no se dispone de estas coordenadas. Otra forma indirecta de valorar el funcionamiento del programa, y que además permite valorar la mejora que se logra obtener a partir de las imágenes de segmentación obtenidas de [6], consiste en la obtención de la máscara que crearía el objeto colocado en la reconstrucción en la panorámica inicial, comparando luego esta máscara obtenida con la presente en el ground truth utilizado en [6] para el entrenamiento de la red. La forma de valoración será el porcentaje de solapamiento o intersection over unión (IoU). Se compara el número de pixeles de intersección entre el Ground truth y la máscara reconstruida, con la unión de ambas mascaras.

$$IoU = \frac{Pixeles\ coincidentes}{Pixeles\ coincidentes + Pixeles\ no\ coincidentes}$$

Los resultados obtenidos se van a analizar utilizando distintas clasificaciones en función del rendimiento del programa para cada objeto. Inicialmente se va a estudiar los resultados para objetos planos y volumétricos al ser esta clasificación la más notable para los objetos.

	Planos	*Volumétricos*
IoU	78.47	84.23

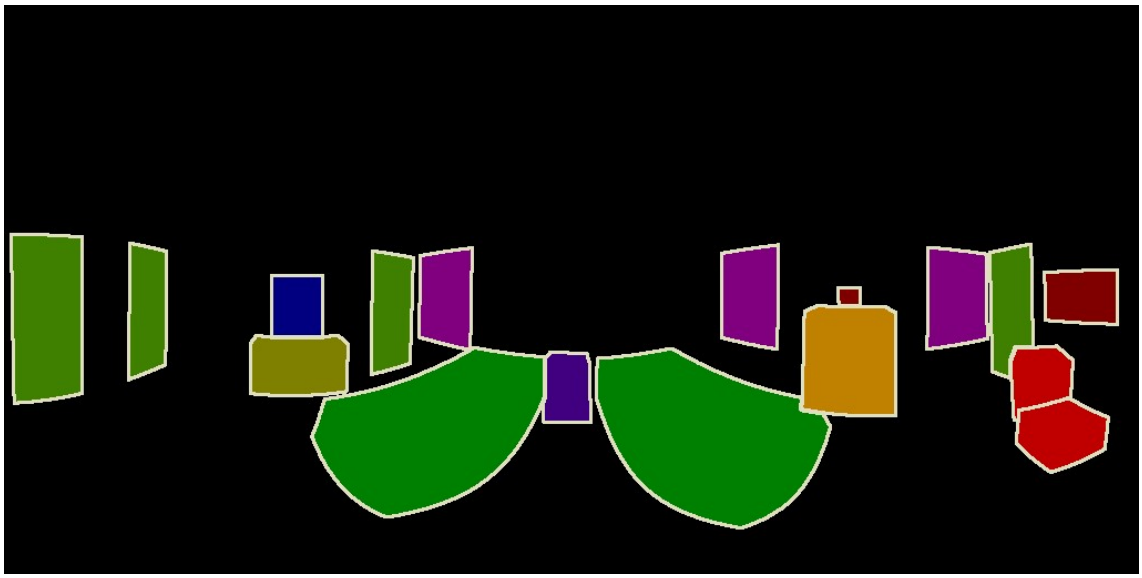


Figura 4.2: Ground truth utilizado para las correcciones

Antes de comenzar a hablar de los resultados, es importante decir que se ven afectados además de por el rendimiento del programa, por cómo están definidas las imágenes del ground truth, estas imágenes se ven muy afectadas siempre que un objeto se encuentra entre otro objeto y la cámara; esto se puede ver en la imagen [Figura 4.2], donde se pueden ver interferencias entre una cómoda y la cama de la derecha de la escena o la puerta de la derecha y el sofá delante de ella. Detalles como estos, afectan negativamente a los valores.

Los objetos planos, alcanzan un valor de media algo superior a un 75%, si se desglosa este valor entre los obtenidos, para algunos de los objetos planos se pueden observar más detalles.

	Cuadro	*Espejo*	*Ventana*	*Cortina*	*Puerta*
IoU	83.86	81.25	87.39	45.47	78.42

Se puede ver que todos los objetos alcanzan unos valores muy elevados, a excepción de las cortinas, las cuales se quedan muy atrás, el motivo de esto es que tal y como funciona el programa, y al combinar esto con algunos resultados obtenidos de [6], provoca que las cortinas a menudo se coloquen encima de las ventanas cuando deberían quedarse a los lados de estas, provocando que el porcentaje buscado baje mucho. Una posible solución, que se explicara en el capítulo 5, consistiría en añadir restricciones, en este caso se limitaría el ancho de las cortinas, siempre que se encuentren junto a una ventana, lo cual es algo que siempre sucede, al borde de la misma, de esta forma las cortinas alcanzarían también valores más altos.

En cuanto a los objetos volumétricos todos ellos tienen un rendimiento muy similar en el programa, alcanzando todos ellos valores muy similares

	Cama	*Mesa*	*Mesilla*	*Sofá*	*Armario*
IoU	92.02	77.97	69.70	73.91	83.72

En este tipo de objetos el aspecto que más influye además del reconocimiento de objetos en [6], que obviamente está muy presente en todo el trabajo, es el layout ya que el posicionamiento de estos objetos se ve muchas veces afectado por la posición de las paredes (se ha explicado en [Subsección 3.3] que las coordenadas obtenidas para cada objeto se ven limitadas por las paredes), también afecta el tamaño de los mismos.

Capítulo 5

Conclusión y trabajo futuro

Este trabajo, supone un paso adelante en la reconstrucción de escenas de interior, a partir de una única imagen panorámica consiguiendo combinar los resultados obtenidos en [5] y [6]. Los resultados obtenidos permiten concluir que se han cumplido los objetivos planteados al principio de este trabajo.

Para conseguir unos resultados positivos, en un mayor número de imágenes, se debe trabajar para conseguir una salida más conveniente de [6], sobre todo a la hora de conseguir que las máscaras de cada objeto reconocido sean únicas, en lugar de separarse en muchas más pequeñas, ya que esto provoca que el algoritmo no funcione correctamente.

Otros aspectos a mejorar a menor escala, consisten en considerar la altura de cortinas y ventanas, de la misma forma que ya se hace en este trabajo con las puertas, además de limitar las cortinas a los lados de las ventanas para que no se solapen.

También serían mejorables otros aspectos, como puede ser el distinto tratamiento para cada tipo de objeto, llegándose a descartar máscaras de distinto número de pixeles dependiendo el tipo de objeto con el que se trabaja, ya que ahora son todos descartados a partir del mismo umbral. Podría estudiarse además alguna forma de detectar si más de una máscara pertenece al mismo objeto, (máscaras del mismo tipo próximas entre sí) haciendo uso del layout conocido y de restricciones aplicables para cada tipo de objeto, de esta forma el programa sería aplicable a mas escenas obtenidas de los trabajos previos.

Bibliografía

[1] Sethi, Rachna; Mehrotra, Monica; Sethi, Dhaarna. Deep learning based diagnosis recommendation for COVID-19 using chest x-rays images. En 2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA). IEEE, 2020. p. 1-4.

[2] Santillán, Iván Danilo García. Métodos de visión por computador para detección automática de líneas de cultivo curvas/rectas y malas hierbas en campos de maíz. Tesis Doctoral. Universidad Complutense de Madrid. 2017.

[3] Tian, YingLi, et al. Toward a computer vision-based wayfinding aid for blind persons to access unfamiliar indoor environments. Machine vision and applications, 2013, vol. 24, no 3, p. 521-535.

[4] J. Xiao, K. Ehinger, A. Oliva, and A. Torralba. Recognizing scene viewpoint using panoramic place representation. In IEEE Conference on Computer Vision and Pattern Recognition, pages 2695–2702, 2012.

[5] Fernandez-Labrador, C., Perez-Yus, A., Guerrero, J. J. Estimación del layout 3D en interiores a partir de imágenes panorámicas, Trabajo de fin de master, Escuela de Ingeniería y Arquitectura. Universidad de Zaragoza, 2017.

[6] Guerrero-Viu, J., Fernandez-Labrador, C., Guerrero, J. J. Object Recognition on Panoramic Images, Trabajo de fin de grado, Escuela de Ingeniería y Arquitectura. Universidad de Zaragoza, 2019.

[7] Clara Fernandez-Labrador, Jose M Facil, Alejandro Perez-Yus, Cédric Demonceaux, Javier Civera, Jose J Guerrero (2020/1/17). Corners for layout: End-to-end layout recovery from 360 images. ICRA 2020 & IEEE Robotics and Automation Letters

[8] Julia Guerrero-Viu, Clara Fernandez-Labrador, Cédric Demonceaux, Jose J Guerrero (2020). What's in my Room? Object Recognition on Indoor Panoramic Images Conference. IEEE International Conference on Robotics and Automation, ICRA 2020

[9] Ronneberger, O., Fischer, P., & Brox, T. (2015, October). U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention* (pp. 234-241). Springer, Cham

[10] J. M. Coughlan and A. L. Yuille. Manhattan world: Compass direction from a single image by bayesian inference. In IEEE International Conference on Computer Vision, volume 2, pages 941–947, 1999.

[11] Hedau, Varsha; Hoiem, Derek; Forsyth, David. Thinking inside the box: Using appearance models and context based on room geometry. En European Conference on Computer Vision. Springer, Berlin, Heidelberg, 2010. p. 224-237.

[12] Hedau, Varsha; Hoiem, Derek; Forsyth, David. Recovering free space of indoor scenes from a single image. En 2012 IEEE Conference on Computer Vision and Pattern Recognition. IEEE, 2012. p. 2807-2814.

[13] Y. Zhang, S. Song, P. Tan, and J. Xiao. PanoContext: A whole-room 3D context model for panoramic scene understanding. In European Conference on Computer Vision, pages 668–686. Springer, 2014.

[14] Yang, Yang, et al. Automatic 3d indoor scene modeling from single panorama. En *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018. p. 3926-3934.

[15] Furlan, Axel, et al. Free your Camera: 3D Indoor Scene Understanding from Arbitrary Camera Motion. En *BMVC*. 2013.