Universidad Zaragoza

Trabajo Fin de Máster

Estimación del *layout* 3D en interiores a partir de imágenes panorámicas

Autor

Clara Fernández Labrador

Directores

Jose Jesús Guerrero Campo

Alejandro Pérez Yus

ESCUELA DE INGENIERIA Y ARQUITECTURA

2017

TRABAJOS DE FIN DE GRADO / FIN DE MÁSTER

Escuela de Ingeniería y Arquitectura Universidad Zaragoza

DECLARACIÓN DE AUTORÍA Y ORIGINALIDAD

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

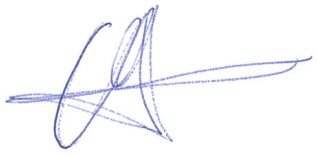
D./Dª. Clara María Fernández Labrador,

con nº de DNI 73019096F en aplicación de lo dispuesto en el art. 14 (Derechos de autor) del Acuerdo de 11 de septiembre de 2014, del Consejo de Gobierno, por el que se aprueba el Reglamento de los TFG y TFM de la Universidad de Zaragoza,

Declaro que el presente Trabajo de Fin de (Grado/Máster) Máster, (Título del Trabajo) Estimación del layout 3D en interiores a partir de imágenes panorámicas,

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

Zaragoza, 21/09/2017



Fdo: Clara María Fernández Labrador

Resumen

En este trabajo se ha desarrollado un método de reconstrucción 3D de habitaciones a partir de una única imagen panorámica de 360 grados de campo de vista horizontal. Este método tiene la principal novedad de combinar razonamientos geométricos de visión por computador y técnicas de aprendizaje profundo (*Deep Learning*) adaptadas a la geometría del tipo de imágenes que proponemos utilizar. Nuestro método utiliza la extracción de esquinas estructurales como punto de partida para elaborar hipótesis sin información previa acerca de la forma de la habitación y con la única restricción de Mundo Manhattan. En particular, dichas esquinas se extraen como intersecciones entre líneas que son ortogonales en el espacio 3D. Este proceso se ha mejorado con el uso de una Red Neuronal Convolucional que detecta bordes estructurales y permite filtrar líneas pertenecientes a otros objetos no relevantes. A partir de estas posibles esquinas dibujamos hipótesis de diseño y escogemos aquella solución que encaja mejor con el mapa de normales obtenido con otro método de aprendizaje profundo. En este trabajo se muestran resultados de reconstrucciones 3D con imágenes de la base de datos pública SUN360 usada por otros trabajos del estado del arte. Con ellos demostramos la efectividad del método con respecto a trabajos existentes y las ventajas de introducir redes neuronales profundas en el desarrollo del proceso.

Abstract

In this work we have developed a method for 3D layout recovery of indoor scenes from a single 360 degrees panoramic image. This method has the main novelty of combining geometric reasoning on computer vision and deep learning techniques adapted to the proposed image geometry. Our method uses the extraction of structural corners as a starting point to construct layout hypotheses assuming Manhattan World and without any prior information about the room shape. In particular, corners are extracted as intersections of lines that are orthogonal in 3D space. This process has been enhanced with a Convolutional Neural Network that detects structural edges and allows filtering lines belonging to other non-relevant objects. From these possible corners we draw layout hypotheses and choose the best fitting solution to the normals' map extracted with another CNN. We show results of 3D layouts recovered from images of the SUN360 public dataset. We demonstrate the effectiveness of our method with respect to existing works and the advantages of the introduction of deep neural networks in the pipeline of the process.

Índice general

1 Introducción **1**
- 1.1 Motivación . . . 2
- 1.2 Estado del arte . . . 2
  - 1.2.1 Reconstrucción del *Layout* . . . 3
  - 1.2.2 Reconstrucción del *Layout* en Imágenes omnidireccionales . . . 4
  - 1.2.3 Redes Neuronales Convolucionales . . . 5
- 1.3 Objetivos . . . 6

2 Imágenes panorámicas **8**
- 2.1 Modelo de proyección esférica . . . 9

3 Razonamiento Geométrico basado en Visión Artificial **11**
- 3.1 Extracción de Líneas en Imágenes Panorámicas . . . 11
- 3.2 Extracción de los Puntos de Fuga en Imágenes Panorámicas . . . 13
- 3.3 Clasificación de Líneas . . . 15

4 Técnicas de *Deep Learning* **16**
- 4.1 Detección de Líneas Informativas . . . 16
- 4.2 Detección de Normales . . . 18

5 Estimación del *Layout* de la Habitación **20**
- 5.1 Eliminación de Líneas No Significativas . . . 20
- 5.2 Hipótesis de Esquinas Relevantes . . . 21
- 5.3 Generación de Hipótesis de Diseño / *Layout* . . . 23
- 5.4 Evaluación de Hipótesis de Diseño / *layout* . . . 25

6 Experimentos **27**
- 6.1 Herramientas y tecnología utilizada . . . 27
- 6.2 Generación de *Ground Truth* . . . 28
- 6.3 Resultados numéricos . . . 28
  - 6.3.1 Comparación de nuestro método con el estado del arte . . . 29
  - 6.3.2 Eliminación de líneas no significativas con el mapa de bordes . 29
  - 6.3.3 Comparación del mapa de normales con otros mapas de características del estado del arte . . . 30
- 6.4 Resultados visuales . . . 32

7 Conclusión **36**

A Artículo enviado al IEEE ICRA **40**

Índice de figuras

1.1 Aplicaciones de la visión artificial 1
1.2 Aplicaciones de la reconstrucción 3D de escenas 2
1.3 Limitaciones de las imágenes convencionales 3
1.4 Ejemplo trabajo [27] 4
1.5 Ejemplo trabajo [18] 6
1.6 [Ejemplo trabajo [14] 6
1.7 Descripción general del algoritmo 7

2.1 Importancia del campo de vista 8
2.2 Tipos de proyección 9
2.3 Proyección esférica 10

3.1 Líneas en imágenes esféricas 11
3.2 Plano proyectivo 3D 12
3.3 Puntos de fuga en imágenes esféricas 14
3.4 Extracción y clasificación de líneas 15

4.1 Partición imagen panorámica para la red [18] 17
4.2 Mapas de bordes estructurales de la red neuronal [18] 17
4.3 Partición imagen panorámica para la red [7] 18
4.4 Mapa de normales de la red [7] 19

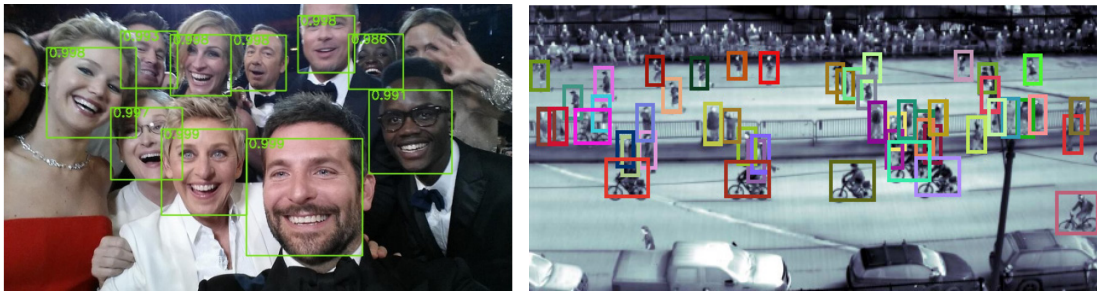
5.1 Obtención líneas significativas 21
5.2 Esquinas candidatas 22
5.3 Ejemplo estimación 3D del diseño de la habitación 23
5.4 Ejemplos generación de hipótesis 25
5.5 Evaluación de hipótesis 26

6.1 Generación del *Ground Truth* 28
6.2 Evaluación de la precisión de píxeles 29
6.3 Comparación de nuestro método con el estado del arte 30
6.4 Ventajas de las redes neuronales 30
6.5 Métodos de evaluación del estado del arte: OM, GC y MM. 31
6.6 Comparación de distintos métodos de evaluación 31
6.7 Habitaciones de cuatro paredes 33
6.8 Habitaciones de cuatro paredes 34
6.9 Habitaciones de seis paredes 35

Capítulo 1

Introducción

La visión es el sentido más importante que tiene el ser humano ya que es el que nos permite obtener más información de nuestro entorno de una manera rápida y efectiva. La visión artificial o visión por computador es un campo esencial de la inteligencia artificial cuyo objetivo es conseguir que los ordenadores vean extrayendo información del mundo a través de las imágenes obtenidas por una cámara. Esta disciplina cubre un amplio abanico de problemas y técnicas tales como el reconocimiento de patrones (*e.g.* reconocimiento facial Fig. 1.1a), la reconstrucción de modelos del entorno, la identificación y seguimiento de objetos o la localización (*e.g.* Seguimiento de personas Fig. 1.1b).



(a) Detección Facial (b) Seguimiento de personas

Figura 1.1: Ejemplos de aplicaciones y problemas de la visión por computador

Uno de los problemas fundamentales de investigación en este campo es la reconstrucción 3D de una escena a partir de una única imagen. El problema ha sido abordado por muchos investigadores que, a lo largo de los años, han experimentado con distintos tipos de imágenes llegando a enfrentarse a imágenes omnidireccionales y dejando atrás las imágenes convencionales con campos de vista más limitados. Además, se han probado diversas aproximaciones incluyendo en la etapa más reciente técnicas de aprendizaje profundo (*Deep Learning*) con el objetivo de mejorar el estado del arte.

Gracias a las nuevas tecnologías, contamos a día de hoy con infinitas posibilidades que ayudan a lograr resultados impresionantes y es importante aprovecharlas y exprimir sus numerosas ventajas.

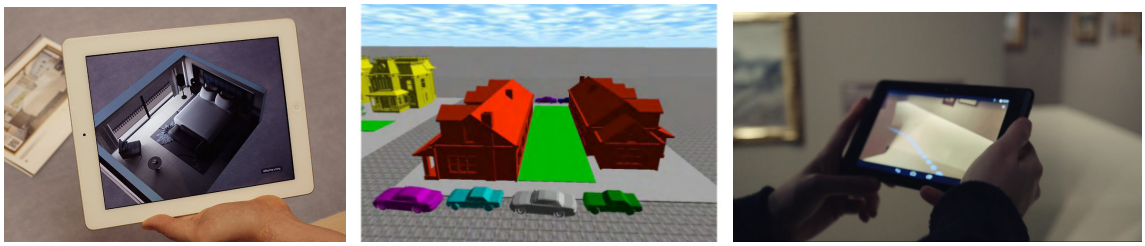
1.1 Motivación

En los últimos años ha crecido el interés por la comprensión y la reconstrucción 3D de escenas debido al paso esencial que ha supuesto en el campo de la visión artificial. Recientemente está siendo de gran utilidad para muchas tareas tales como la navegación en interiores, SLAM [17], coches autónomos, realidad virtual y aumentada y robótica en general.

En la Fig. 1.2a aparece la imagen de una aplicación desarrollada por *Moon Flower technologies*. Se trata de una representación virtual con realidad aumentada de la habitación de un hotel. Los sistemas visuales artificiales y en particular aquellos basados en visión estereoscópica o 3D pueden ayudarnos notablemente a reconstruir la escena y detectar los objetos que se encuentran ella, con el objetivo de etiquetar la realidad, añadir información de su estructura, disposición, color o identificación, e introducir objetos virtuales.

La Fig. 1.2b muestra un ejemplo donde interesa la reconstrucción 3D de un escenario, en este caso exterior, con el objetivo de que un coche autónomo sea capaz de navegar por dicho escenario o realizar acciones más complejas como por ejemplo la de aparcar por si mismo.

Por otro lado, es conocida por todos la aplicación *Google Maps* de Google que ofrece un excelente servicio de navegación en exteriores. En la Fig. 1.2c se muestra un nuevo proyecto de Google Tango que aborda esta vez la navegación en interiores y que fue utilizada en el tour de un museo en el Mobile World Congress 2016.



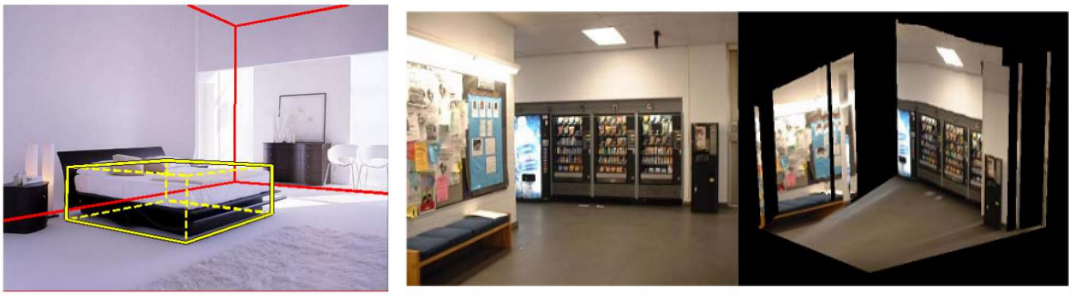
(a) Realidad aumentada (b) Coches autónomos (c) Navegación en interiores

Figura 1.2: Aplicaciones de la comprensión y reconstrucción 3D de espacios

Una imagen es una proyección 2D del mundo 3D, lo cual hace que se pierda una dimensión. A través de razonamientos puramente geométricos es imposible inferir el 3D de una escena a partir de una sola imagen salvo que se realicen hipótesis adicionales (*e.g.* asunción de un mundo Manhattan [4]). Un reto importante de este trabajo es el de integrar ese tipo de hipótesis y tratar de inferir el *layout* 3D de escenas de interior aprovechando técnicas de aprendizaje profundo.

1.2 Estado del arte

Dado que este trabajo engloba distintas contribuciones en aspectos muy interesantes cada uno de ellos por separado, procedemos en esta sección a describir algunos de los trabajos más relevantes de cada uno de ellos organizados en subsecciones separadas. En la subsección 1.2.1 nos centramos en los trabajos de reconstrucción 3D



(a) Simplificación caja 3D (b) Reconstrucción con imágenes convencionales

Figura 1.3: Limitaciones de la reconstrucción de *layouts* con imágenes convencionales.

de *layouts* o diseño de habitaciones en general. A lo largo del trabajo nos referiremos con ambas expresiones indistintamente a este término. En la subsección 1.2.2 hablamos de los distintos tipos de imágenes que se utilizan en la literatura con este propósito centrándonos sobretodo en la tipología de imagen por la que hemos apostado nosotros, las imágenes omnidireccionales. Por último, la subsección 1.2.3 recoge trabajos que apuestan por las novedosas redes neuronales enfocadas a fines similares o relacionados con el tema que pueden resultar igualmente muy interesantes.

1.2.1 Reconstrucción del *Layout*

Probablemente el primer intento de abordar el desafío de realizar reconstrucciones 3D de espacios interiores a partir de una única imagen fueron Delage *et al.* en [6], cuyo algoritmo encuentra límites entre el suelo y las paredes de habitaciones usando un modelo de red Bayesiano. Un ejemplo de este trabajo se muestra en la Fig. 1.3b. Por otro lado, Lee *et al.* [15] utiliza segmentos de líneas para generar hipótesis de diseño evaluando su validez con un Orientation Map (OM), *i.e.* mapa con la orientación de cada superficie de la imagen, pudiendo así evitar confiar en propiedades específicas de la escena como colores o gradientes de la imagen.

Desafortunadamente, las habitaciones suelen estar llenas de objetos que ocluyen los bordes que realmente pertenecen a la estructura real de la habitación (*e.g.* bordes entre paredes o entre paredes y suelo) así como las esquinas, lo cual hace que aparezcan segmentos de líneas engañosos que alteran la estimación de la reconstrucción del diseño de la escena. Para combatir este problema se hacen algunas suposiciones y, consecuentemente, se proponen un conjunto de reglas basadas en cierta coherencia física. Por lo general, los principales supuestos son que todas las estructuras en ambientes interiores están compuestas por superficies planas y que estas superficies están orientadas de acuerdo con tres direcciones ortogonales principales (conocida como suposición de Manhattan World [4]). Esta suposición se da para la mayoría de los ambientes interiores, y es ampliamente utilizada en la literatura [12, 18, 21, 22, 27, 28, 31].

Otros trabajos [10, 11, 23], tratan de simplificar el problema asumiendo que la habitación es una caja 3D de cuatro paredes, y utilizan el mapa de características llamado Geometric Context (GC), *i.e.* etiquetas geométricas como contexto para la detección de objetos, en lugar del Orientation Map (OM), lo cual ayuda a detectar

Figura 1.4: Estimación de *layout* y objetos de [27]

el desorden de la escena (*e.g.* muebles, plantas...). En algunos de estos trabajos, los objetos son detectados y tratados también como cajas que delimitan los bordes de estos. Probablemente los primeros en modelar la interacción 3D entre objetos y *layout* fueron Gupta *et al.* en [8]. Un ejemplo de las simplificaciones a cajas se muestra en la Fig. 1.3a.

1.2.2 Reconstrucción del *Layout* en Imágenes omnidireccionales

La mayoría de los trabajos enfocados a este fin utilizan imágenes convencionales con campo de vista limitado, lo cual impide una reconstrucción total de la escena o implica "inventar" partes de ella que no vemos y que pueden no ajustarse a la realidad (*e.g.* en Fig. 1.3).

Recientemente, sin embargo, se han propuesto algunas alternativas para ampliar el campo de vista. Lopez-Nicolas *et al.* en [16] llevan a cabo la recuperación del *layout* utilizando un sistema catadióptrico. Perez-Yus *et al.* en [21] realizan las hipótesis para la reconstrucción del *layout* combinando imágenes tomadas por una cámara de ojo de pez (180 grados de campo de visión horizontal) y con información de profundidad dada por un sensor RGB-D para proporcionar una escala.

Actualmente, y con el fin de acabar con los problemas recientemente mencionados, se están utilizando incluso imágenes panorámicas de 360 grados de campo de vista horizontal. Este tipo de imágenes son muy fáciles de obtener a día de hoy con matrices de cámaras, lentes especiales o algoritmos de combinación automática de imágenes [9]. En [12], su método muestra las ventajas de contar con un campo de vista completo sobre vistas parciales de la misma escena en comparación con métodos previos. *PanoContext* [31] hace uso de panoramas para estimar tanto el *layout* de la habitación como las cajas asociadas a cada objeto que se encuentra en su interior. También ellos asumen como simplificación la habitación como una caja 3D de cuatro paredes. De manera similar, Jiu Xu *et al.* en [27] proporciona resultados que no se limitan a simples cajas 3D a la hora de estimar el diseño de la habitación y confía, al igual que en [31], en mapas de características como GC o OM. Un ejemplo de su *output* se muestra en la Fig. 1.4. Probablemente, estos dos últimos trabajos mencionados [31, 27] son los que más nos han servido de inspiración a la hora de afrontar nosotros este desafío. En [28] por otro lado, haciendo uso de este mismo tipo de imágenes, tratan el problema como un gráfico con líneas y

superpíxeles como nodos y lo resuelven con complejas restricciones geométricas en lugar de evaluar con los mapas de características mencionados anteriormente.

Las imágenes omnidireccionales no solo están teniendo un gran impacto en el problema de la reconstrucción 3D de distintos escenarios. Debido a su reciente popularidad, otros investigadores se han fijado en ellas. Jianxiong *et al.* por ejemplo, en su trabajo [26], se centra en resolver el problema de clasificar el tipo de lugar que aparece en el panorama así como en reconocer el punto de vista del observador dentro de esa categoría de lugar.

1.2.3 Redes Neuronales Convolucionales

Las Redes Neuronales Convolucionales (también llamadas CNNs, del inglés *Convolutional Neural Network*) son un tipo de red neuronal artificial con la característica peculiaridad de hacer la suposición explícita de que las entradas a la red son imágenes. Dicha red es *feed-forward*, *i.e.* la información se mueve a lo largo de la red únicamente hacia adelante, desde los nodos de entrada, a través de los nodos ocultos (si los hay) a los nodos de salida. No hay ciclos ni bucles en la red. Se trata de un modelo computacional diseñado para emular el comportamiento de la corteza visual. Las células neuronales de la corteza visual son sensibles a regiones específicas del campo visual y todas ellas producen percepción visual. Es la base detrás de las CNNs, donde los filtros buscan características específicas en la imagen de entrada y aprenden de ellas, abordando temas clave de la visión artificial con precisión y rapidez.

En los últimos años, los investigadores han abordado también el problema de la reconstrucción de *layouts* con estas Redes Neuronales Convolucionales obteniendo muy buenos resultados. Por ejemplo, [29] utiliza una CNN para segmentar el plano del suelo superando a los métodos tradicionales. DeLay [5] proporciona mapas de escenas con cinco etiquetas asociadas a suelo, techo y a las tres paredes visibles en imágenes convencionales. Algunos trabajos utilizan CNNs para extraer información de los bordes estructurales de escenas de interior, siendo capaces de encontrarlos con bastante precisión incluso en presencia de oclusiones [18, 30], un ejemplo de ello se aprecia en la Fig. 1.5. Por otro lado, en [14] predicen la localización de las esquinas de la habitación usando RoomNet, una red encoder-decoder en la cual se entrenan todos los parámetros a la vez. También en este trabajo se simplifican las habitaciones a cajas de cuatro paredes. Un ejemplo del output de esta red se puede ver en la Fig. 1.6.

Otros trabajos de *deep learning* no están en principio relacionados con la reconstrucción de *layouts* pero generan resultados que pueden resultar de mucho interés para este tipo de tareas. Por ejemplo, Eigen *et al.* [7] utiliza tres CNN apiladas para procesar las imágenes en tres diferentes escalas, extrayendo una estimaciones de profundidad, estimaciones de las direcciones normales de las superficies y etiquetas semánticas a partir de simples imágenes RGB. Redes como esta pueden proporcional información derivada de la profundidad similar a la que se consigue típicamente con cámaras RGB-D, aunque de manera menos fiable.

Este tipo de CNNs han demostrado obtener muy buenos resultados pero siempre centrándose en imágenes tradicionales con campo de vista limitado, dificultando así su uso con imágenes omnidireccionales.

Muy recientemente han comenzado a combinarse este tipo de técnicas de *deep*

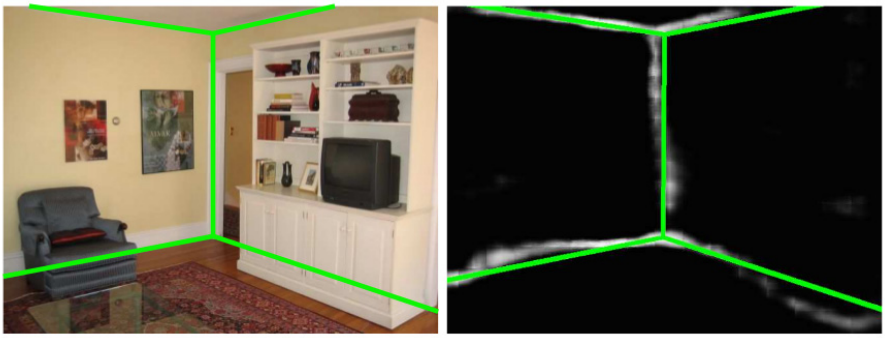


Figura 1.5: *Ground Truth* vs. mapa estructural obtenido por la red de Mallya et al. [18]

Figura 1.6:]
Detección de esquinas con la red neuronal de [14]

learning con las de visión artificial tradicionales. Yuzhuo *et al.* [22] proponen un método de dos fases: una primera en la que estiman aproximadamente el *layout* con una red neuronal, y una segunda fase de optimización.

1.3 Objetivos

En este trabajo, proponemos un método que combina técnicas de razonamiento geométrico y técnicas de aprendizaje profundo (*deep learning*) para estimar el diseño completo 3D de una escena de interior a partir de una única imagen RGB panorámica.

A pesar de su complejidad adicional, elegimos imágenes panorámicas ya que, gracias a su amplio campo de vista, es posible acceder a toda la información de la escena de una vez, incluyendo la parte del techo que suele no ser visible en imágenes convencionales y que, al ser la parte normalmente con menos oclusiones, puede resultar muy útil a la hora de buscar información relevante para la reconstrucción, permitiendo obtener soluciones de habitación cerradas basadas en la mejor distribución total de la escena.

Además, a diferencia de trabajos anteriores [15, 31, 21], nuestro método aprovecha el aprendizaje profundo a lo largo del proceso. Investigaciones recientes en el campo, muestran que estos enfoques basados en grandes cantidades de datos superan a los métodos tradicionales, que necesitan elaborar razonamientos cada vez más complejos para tener éxito en nuevos problemas o para cumplir los requisitos de precisión actuales.

La principal novedad de nuestro trabajo es la explotación de trabajos de aprendizaje profundo para imágenes panorámicas aplicadas al problema de la estimación de *layouts*, para lo cual proponemos un nuevo método flexible que integra técnicas antiguas y nuevas, sin restricciones de forma de caja 3D de cuatro paredes, calibración de cámaras o campo de vista.

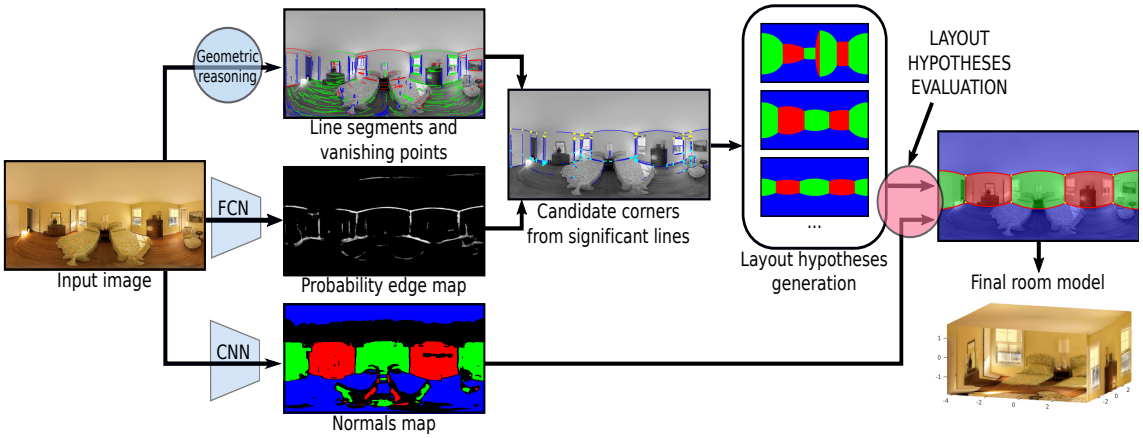


Figura 1.7: Descripción general del algoritmo.

Una visión general de nuestro método se muestra en la Fig. 1.7: En primer lugar, extraemos segmentos de línea y los puntos de fuga del panorama. Paralelamente, el panorama se ejecuta a través de la red de [18] que permite filtrar líneas no informativas procedentes del desorden de la escena. A continuación, las líneas significativas se utilizan para extraer esquinas como intersecciones de líneas ortogonales, que luego se utilizan para dibujar hipótesis de diseño de diversas formas. Las hipótesis se comparan con un mapa de normales de referencia obtenido con otra CNN [7]. La solución que más se ajusta a este mapa de normales es el modelo de habitación final.

La evaluación experimental con imágenes panorámicas de la base de datos pública SUN360 [26] de escenas interiores muestra una mejora con respecto a otros trabajos del estado del arte y revela las ventajas de utilizar redes neuronales profundas en el proceso.

Capítulo 2

Imágenes panorámicas

Figura 2.1: Importancia visual de las diferencias en el campo de vista entre una imagen panorámica, una imagen convencional y lo que vemos las personas.

En este trabajo hemos apostado por utilizar imágenes panorámicas como entrada a nuestro algoritmo con el objetivo de llevar a cabo la reconstrucción de *layouts*. La Fig. 2.1 muestra un ejemplo de la diferencia de cómo vemos una escena en función del campo de vista.

Cada vez más nos estamos acostumbrando a ver imágenes de este tipo, sin embargo, la complejidad del tipo de proyección que emplean hace que no sea tan intuitivo o directo comprender las proporciones o la distribución de las escenas que estas imágenes muestran. Esto es debido a que, tratándose de una proyección esférica, lo que veríamos como una línea recta en la realidad o en una imagen convencional, aparece como una línea curva en la imagen panorámica (Ver Fig. 3.1). Además, el ser humano no es capaz de ver lo que hay detrás de sí mismo y sin embargo sí es posible con estas imágenes, lo cual muchas veces resulta extraño para nuestro cerebro.

2.1 Modelo de proyección esférica

Dada la importancia y la poca convencionalidad de este tipo de imágenes hemos considerado importante dedicar una sección al tipo de proyección característico de las imágenes panorámicas y a cómo se ha trabajado con ellas, dado que ha supuesto un reto importante en el inicio de este trabajo. Con esto hacemos referencia a la proyección esférica o proyección equirectangular, que es la propia de los mapas del mundo que estamos acostumbrados a ver, en la que los polos se encuentran deformados. Ver Fig. 2.2.

Figura 2.2: Proyecciones más comunes según su campo de vista (en orden): Esférica, Cilíndrica, Rectilinear y Ojo de pez.

En esta sección por tanto, se procede a detallar el paso de coordenadas esféricas a coordenadas en la imagen.

Definimos la resolución de la imagen panorámica como $W \times H$ píxeles, siendo W la anchura de la imagen y H, la altura de ésta. Este tipo de imágenes no requiere reproyección, como es el caso de la proyección cilíndrica. La textura es simplemente reciclada y guardada en un sistema de coordenadas de latitud / longitud que cubre 360 grados de campo de vista horizontal y 180 grados de campo de vista vertical, por lo tanto podemos saber que $W = 2H$. Situamos el centro de coordenadas en el centro de la imagen, *i.e.* $(\frac{W}{2}, \frac{H}{2})$.

En este tipo de representaciones, tanto de interiores como de exteriores, se establece normalmente una altura para la línea de horizonte de aproximadamente la media de una persona, asumimos por tanto que el centro de la cámara esta situado a una altura de *e.g.* 1.7 m.

El sistema de coordenadas esféricas se utiliza para determinar la posición espacial de un punto mediante una distancia y dos ángulos. En consecuencia, un punto $P(X, Y, Z)$ queda representado por un conjunto de tres magnitudes: la distancia en píxeles sobre la imagen (u o v), el colatitud θ y el azimut ϕ. El colatitud θ es el ángulo complementario de la latitud, *i.e.* cubre desde -90 grados a +90 grados, y el azimut ϕ se refiere al ángulo de la orientación sobre la superficie de una esfera real o virtual y cubre desde -180 grados a +180 grados.

La transformación de las coordenadas esféricas a las coordenadas en la imagen se pueden apreciar visualmente en la Fig. 2.3 y a continuación se muestran los pasos a seguir en detalle:

Paso de coordenadas 3D en el mundo P(X,Y,Z) a su proyección sobre la esfera unidad p(x,y,z)

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

Paso a coordenadas esféricas

$$(sin\phi \cdot cos\theta, cos\theta \cdot cos\phi, sin\theta) = (x, y, z)$$

Convertir a coordenadas de imagen (u,v)

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

Los ángulos ϕ y θ quedarían por tanto:

$$\phi = \frac{(u - \frac{W}{2}) \cdot 2\pi}{W}$$

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

Este cambio de coordenadas esta muy presente a lo largo de todo el trabajo ya que precisamente el objetivo de éste es trasladar la información de la imagen a una reconstrucción 3D de la misma.

Además, como consecuencia del tipo de proyección, no se pueden aplicar directamente la mayoría de los métodos del estado del arte a estas imágenes, *i.e.* extractores de líneas y puntos de fuga, redes neuronales, etc. Lo cual nos llevará a lo largo del trabajo a proponer nuevos algoritmos adaptados a su geometría y a su adaptación para el uso de cualquier red neuronal de la literatura, empleando en este trabajo dos de ellas, [18, 7].

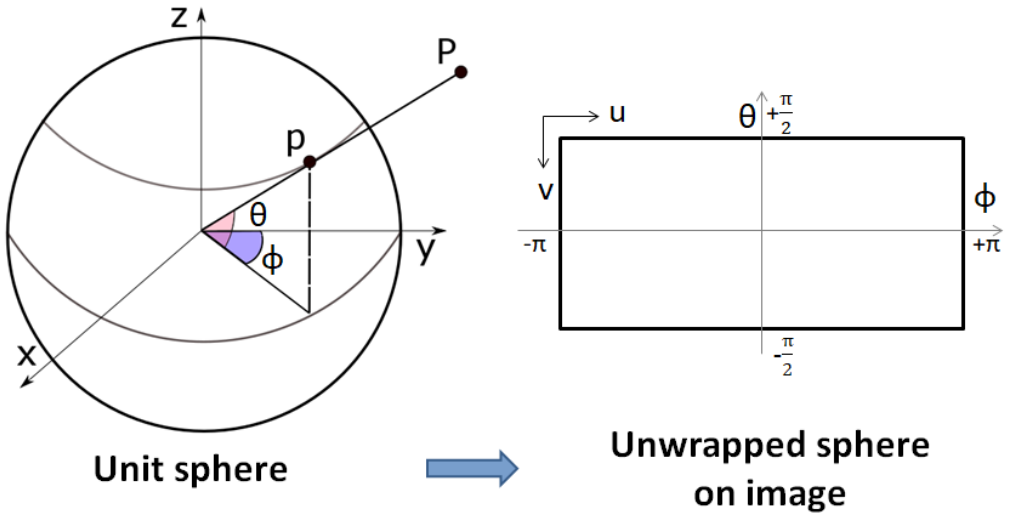


Figura 2.3: Proyección esférica

Capítulo 3

Razonamiento Geométrico basado en Visión Artificial

En este capítulo presentamos las principales tareas llevadas a cabo para la obtención y clasificación de las piezas de información básicas de nuestro método: las líneas. Estas tareas se basan en técnicas de visión artificial y han sido abordadas por múltiples trabajos a lo largo de los años [15, 12, 21, 31].

3.1 Extracción de Líneas en Imágenes Panorámicas

Nuestra propuesta comienza con la extracción de líneas de la imagen panorámica. Existen muchos enfoques para cámaras omnidireccionales, como [2] que es capaz de extraer las líneas para una amplia variedad de sistemas dióptricos y catadióptricos sin necesidad de calibración previa. Otra alternativa es [16], que utiliza la *toolbox* de Matlab de Bazin adaptando las ecuaciones para un sistema hipercatadióptrico. *PanoContext* [31] trabaja con panoramas y los divide en un conjunto de imágenes en perspectiva y ejecuta el algoritmo LSD (Line Segment Detection) [25] en cada una de ellas por separado y luego proyecta las líneas de nuevo al panorama. [20] extiende el método LSD para hacer frente a panoramas utilizando un detector de arcos en grandes círculos.

Nosotros hemos desarrollado un método basado en el procedimiento RANSAC (RANdom SAmple Consensus) [32] que utiliza directamente el panorama sin ne-

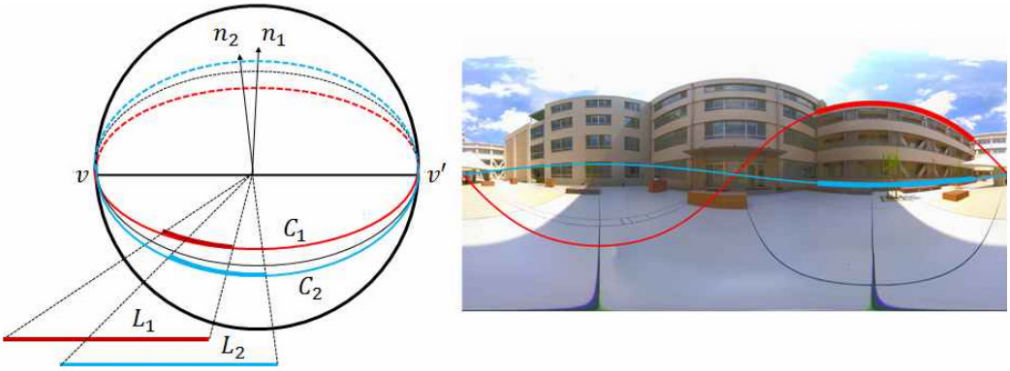


Figura 3.1: Líneas en la realidad proyectadas en la imagen esférica como arcos de un gran círculo. (Imagen de Seon Ho oh *et al.* [19])

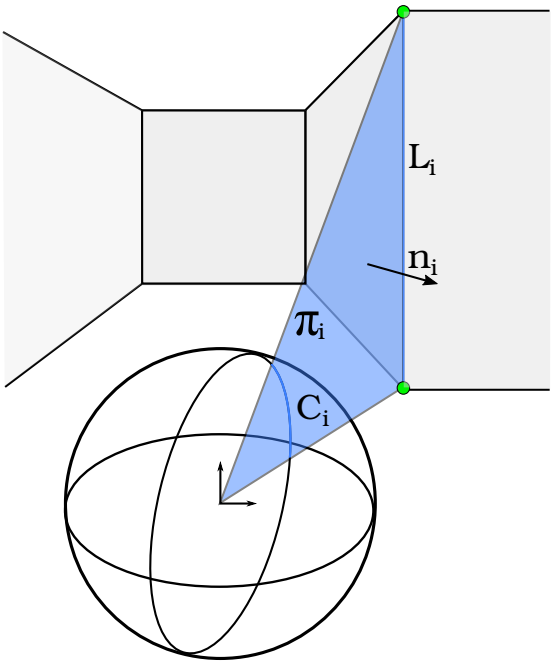


Figura 3.2: Representación del plano proyectivo 3D que incluye la línea y el centro de la cámara con su vector normal.

cesidad de dividirlo o hacer rectificaciones. De esta manera logramos evitar líneas duplicadas procedentes de diferentes divisiones, así como obtener segmentos de línea completos mejorando la posterior generación de hipótesis y la eficiencia global del método.

Dado que estamos trabajando con imágenes panorámicas, se debe tener en cuenta que una línea recta en el mundo, L_i, es proyectada como un segmento de arco en un gran círculo sobre la esfera, C_i, y por tanto, aparece como un segmento de línea curva en la imagen. Por este motivo, cada segmento de arco es representado por el vector normal, n_i, del plano proyectivo, π_i, que incluye la propia línea y el centro de la cámara (ver Fig. 3.2). Además, debido a las características de este tipo de proyección, las líneas que son paralelas en el mundo son proyectadas en la esfera intersectándose en dos puntos antipodales, $v - v'$. En la Fig. 3.1 aparece representado este fenómeno.

Nuestro método se basa en extracción de contornos y el concepto de normales de las líneas. En primer lugar aplicamos un filtro Canny [3], que permite detectar bordes en imágenes. Individualizamos los N bordes, $l_{1..N}$, y eliminamos aquellos que están repetidos o cuya longitud es menor a un determinado *threshold* que decidimos experimentalmente, ya que asumimos que no pertenecen a bordes estructurales de la escena si no a pequeños objetos que podrían dar lugar a confusiones.

A continuación, partiendo de los bordes recién obtenidos en la imagen aplicamos nuestro algoritmo tipo RANSAC que desarrollamos como se explica a continuación. Este procedimiento elige inicialmente de manera aleatoria dos puntos en la imagen, (p_i, p_j), que se corresponden en el espacio 3D con dos rayos, (r_i, r_j), de uno de los bordes l_k para generar líneas candidatas de la imagen que son votadas por el resto de puntos del mismo borde. Para ello, se computa su producto vectorial obteniendo así una posible dirección normal para este grupo de puntos, $n_k = (r_i^k \times r_j^k)$. La normal obtenida se compara con el resto de rayos del grupo considerándose *inliers*

Algorithm 1 Line extraction from panorama - RANSAC algorithm

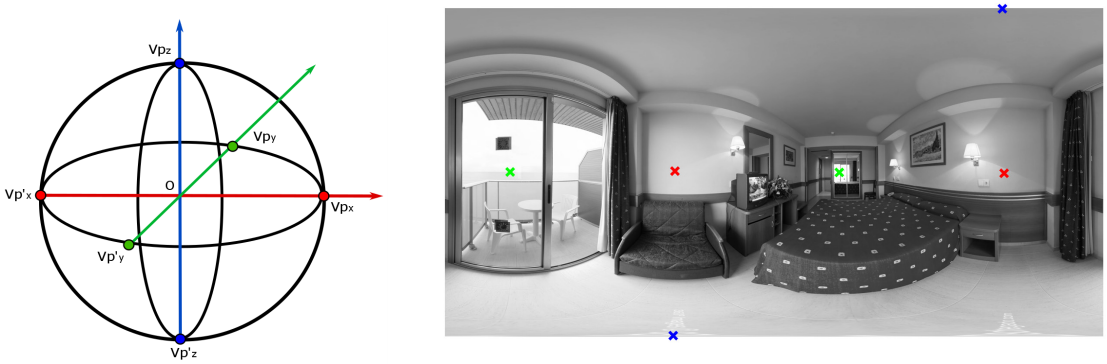
function LINE EXTRACTION(bestInliers, bestOutliers)
  $l \leftarrow \{r_1, r_2, ..., r_i, ..., r_m\}$
  $r_i \leftarrow \{x_i, y_i, z_i\}$
  for each edge, l **do**
    for k iterations, **do**
      (r_i,r_j) $\leftarrow$ rays random selection
      $n \leftarrow r_i \times r_j$, normalized vector product
      $\alpha_{1..m} \leftarrow$ angle between n and $r_{1..m}$
      $inliers \leftarrow r_{1...m}(\alpha_{1..m} \leq 0.5°)$ ▷ $\simeq 90°$
      $outliers \leftarrow r_{1...m} \not\subset inliers$
      if $n°inliers > n°bestInliers$ **then**,
        $bestInliers \leftarrow inliers$
        $bestOutliers \leftarrow outliers$
      end if ▷ Best solution is found
    end for
  end for
end function

del modelo aquellos que cumplen la condición de perpendicularidad con la normal n_k bajo un determinado *threshold* angular (*e.g.* 0,5°) determinado experimentalmente, y *outliers* aquellos que no la cumplen. Este procedimiento se repite un número fijo de veces dado que se trata de un algoritmo iterativo. Finalmente, la iteración que haya dado lugar a un mayor número de *inliers* se considera el mejor modelo, dando la dirección normal que más se ajusta a la línea. A continuación, si el número de *inliers* de la línea es superior a la longitud de segmento mínima establecida, conservamos dicha línea para el siguiente paso del algoritmo. En caso contrario, la línea es eliminada. Este mismo procedimiento se aplica para cada uno de los bordes que se habían obtenido anteriormente, permitiéndonos así disponer finalmente de un conjunto de líneas candidatas de longitud considerable y con su dirección normal como información, *i.e.* su orientación. Ver Algoritmo 1.

3.2 Extracción de los Puntos de Fuga en Imágenes Panorámicas

La estimación de los puntos de fuga en imágenes es muy común en temas de visión artificial y es un problema que ha sido abordado desde hace más de una década dado que la identificación de dichos puntos nos permite comprender estructuras 3D a partir de características 2D. Los puntos de fuga son aquellos puntos en el plano imagen donde convergen las proyecciones de las líneas paralelas del mundo. Son características invariantes a escala y rotación, por lo que pueden ser utilizadas para múltiples tareas como correspondencia entre imágenes, calibración de la cámara o reconocimiento de objetos.

Para llevar a cabo esta tarea adoptamos la suposición del mundo de Manhattan [4] por la cual existen tres puntos de fuga ortogonales dominantes en la esfera alineados con tres direcciones dominantes en el mundo, uno por cada orientación posible



(a) Puntos de fuga en la esfera

(b) Puntos de fuga en la imagen panorámica

Figura 3.3: Representación gráfica, sobre la esfera y sobre la imagen, de los tres puntos de fuga principales que definen una escena y sus antipodales

de las aristas. Es importante mencionar que las líneas paralelas en el mundo intersectan en un único punto de fuga mientras que en imágenes esféricas las proyecciones lineales dan lugar a curvas de modo que las líneas paralelas se intersectan en dos puntos de fuga antipodales. Para la estimación de dichos puntos de fuga elegimos el trabajo de Seon Ho Oh *et al.* [19].

A partir de las líneas extraídas siguiendo el método de la Sec. 3.1, obtenemos los puntos de fuga aplicando de nuevo un algoritmo tipo RANSAC. De la extracción de líneas anterior, tenemos como información las coordenadas de proyección de cada línea sobre la esfera unidad y la dirección normal del círculo que forma la línea en dicha esfera, n_i.

El algoritmo tipo RANSAC se inicia con una selección aleatoria de tres líneas de entre todas las extraídas. Con las dos primeras se computa el primer punto de fuga (*vanishing point* en inglés, *vp*) de manera que $vp_1 = n_1 \times n_2$. El segundo punto de fuga se computa mediante la intersección del primero y la normal de la tercera línea elegida por el algoritmo tal que $vp_2 = vp_1 \times n_3$. Por último, el punto de fuga en la tercera dirección se calcula con los dos anteriores como $vp_3 = vp_1 \times vp_2$. Se repite el proceso para una serie de iteraciones hasta quedarnos con el resultado con el mayor número de *inliers*. Se define una línea como *inlier* cuando su distancia geométrica es menor de un determinado *threshold* angular a una de las direcciones principales ($|n_i \times vp_j| \leq th$, con j=1,2 ó 3). Esto implica que la línea está alineada con el mundo de Manhattan, es decir, la dirección de sus aristas apuntan a los puntos de fuga. En caso contrario, la línea es descartada.

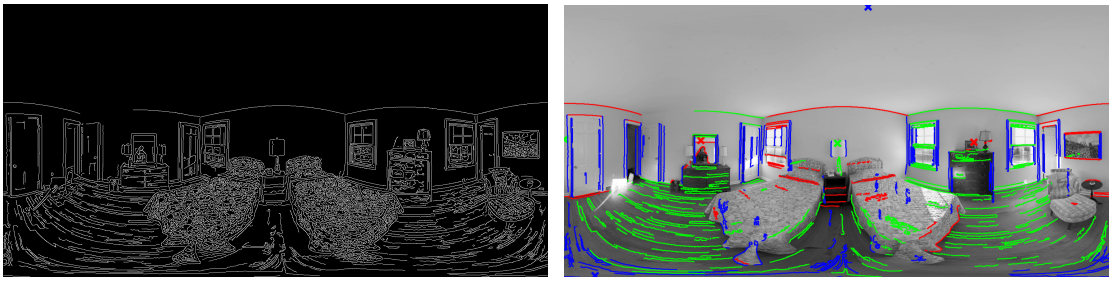
En la Fig. 3.3a se puede observar la esfera unidad con los tres puntos de fuga principales y sus antipodales representados en ella. A su lado, Fig. 3.3b un ejemplo de imagen panorámica con sus puntos de fuga coloreados según la dirección que representan. En la imagen, el punto de fuga vertical (azul) y su antipodal no aparecen uno sobre el otro, esto se debe a que realmente cualquier pixel de la primera y de la última fila están en el mismo ángulo y punto debido a la proyección esférica de la imagen.

3.3 Clasificación de Líneas

Los segmentos de línea se clasifican de acuerdo a las direcciones del mundo de Manhattan, por tanto se asocia cada línea a uno de los tres puntos de fuga existentes obtenidos como ha sido explicado en el apartado anterior, *i.e.* a una de las tres direcciones principales.

Para llevar a cabo tal clasificación sabemos que el vector normal de cada línea n_i debe ser ortogonal a la dirección de Manhattan con la que esta está orientada. Verificamos el ángulo entre las normales de las líneas y las direcciones de Manhattan. Si la perpendicularidad de la línea satisface un determinado *threshold* con una dirección, etiquetamos la línea como orientada en dicha dirección. Aquellas líneas cuyas normales no son perpendiculares a ninguna de las tres direcciones del mundo de Manhattan son descartadas.

Las bordes iniciales obtenidos con el filtro Canny se muestran en la Fig. 3.4a. Por otro lado, las líneas finales extraídas y clasificadas se pueden observar en la Fig. 3.4b. Las líneas asociadas a la misma dirección de Manhattan se muestran del mismo color.



(a) Lineas iniciales en el panorama (b) Líneas finales clasificadas según dirección

Figura 3.4: Proceso inicial de obtención y clasificación de líneas y puntos de fuga

En la Fig. 3.4b se ve como el número de líneas detectadas es muy elevado, perteneciendo la mayoría de ellas a partes de la escena que no conforman la estructura principal de la habitación, *i.e.* aparecen muchas líneas marcando dibujos del suelo, de las camas y de los muebles que se encuentran repartidos por la habitación en general. Se trata de un problema muy común y en este trabajo lo abordaremos de una manera distinta y mas novedosa con respecto a trabajos anteriores como veremos más adelante.

Capítulo 4

Técnicas de *Deep Learning*

Las Redes Neuronales Convolucionales (CNNs) han sido aplicadas con éxito a una gran variedad de tareas como reconocimiento de objetos, clasificación de escenas, segmentación semántica, etc. Pero en los últimos años, debido a los rápidos avances en este área, los investigadores han explorado la posibilidad de usar este tipo de redes para estimación de *layouts*.

En este trabajo no entrenamos directamente una red neuronal de extremo a extremo (*end-to-end*) con imágenes omnidireccionales dado que, hasta donde sabemos, no existe ningún *dataset* con la cantidad suficiente de datos etiquetados o con la amplia variedad de distribución de datos que es necesaria para entrenar una red neuronal profunda (DNN). En cambio, existen *datasets* de escenas de interior etiquetadas suficientemente grandes que se han utilizado con éxito para entrenar DNNs (*e.g.* el conjunto de datos RGB-D de NYUDv2 para segmentación semántica [24], grabado con cámaras convencionales con sensores de profundidad).

Aquí elegimos utilizar dos CNNs de la literatura entrenadas con imágenes convencionales para las cuales adaptamos tanto la entrada como la salida a la geometría de imagen propuesta. Por lo tanto, nuestro enfoque demuestra que es posible aprovechar estas nuevas técnicas sin necesidad de etiquetar grandes conjuntos de datos ni de entrenar redes complejas.

Para ello, las imágenes panorámicas son separadas en varias imágenes en perspectiva con cierto solape con un campo de vista similar al de las imágenes convencionales con las que han sido entrenadas. Ejecutamos el algoritmo en cada una de ellas separadamente y finalmente las juntamos de nuevo en su posición original mediante warping (pandeo) [26] resolviendo las zonas de solape en cada caso de distinta manera. Para llevar a cabo la separación del panorama escogemos, por un lado, los distintos puntos que serán el centro de cada imagen en perspectiva repartidos a lo largo de los 360 grados en horizontal y 180 grados en vertical de la imagen esférica y, por otro lado, el campo de vista que queremos aplicar a cada una de las imágenes. La elección del campo de vista se basó en un estudio de funcionamiento que realizamos para distintas fotos con varios campos de vista, comprobando que para 70 grados de campo de vista horizontal obteníamos los mejores resultados.

4.1 Detección de Líneas Informativas

Mallya *et al.* [18] proponen una Fully Convolutional Network (FCN) que ha sido entrenada para estimar mapas de probabilidades que representan los bordes de la

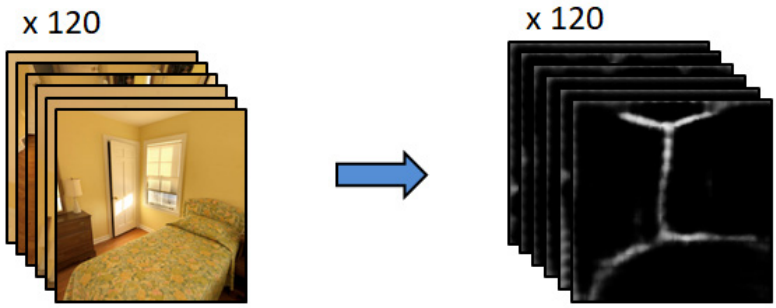


Figura 4.1: Entrada y salida de la red neuronal de Mallya *et al.* [18]

caja 3D proyectada que encaja mejor con la habitación, incluso en presencia de desorden y oclusiones. El hecho de que sea Fully Convolutional implica que esta compuesta únicamente por capas convolucionales de inicio a fin, es decir, sin ninguna capa completamente conectada (fully-conected) típicas en la parte final de las redes convolucionales originales.

Para mejorar el mapa obtenido de bordes significativos eliminamos el ruido que devuelve la red en las partes superior e izquierda de la imagen, así como los valores de los píxels asociados a una menor probabilidad por debajo de un determinado *threshold* obtenido experimentalmente (0.2 sobre 1). Este *threshold* se aplica debido a que las probabilidades más bajas están asociadas a líneas con menor probabilidad de ser realmente estructurales y comprobamos que era un valor adecuado para obtener un resultado más fino.

En la Fig. 4.1 aparece a la izquierda un conjunto de imágenes que representa las imágenes en perspectiva que introducimos a la red neuronal, y a su derecha se puede observar el mismo conjunto de imágenes procesadas ya por la red con los mapas de bordes donde aparecen estos marcados en un blanco graduado en función de la probabilidad que tienen asociada y el fondo negro.

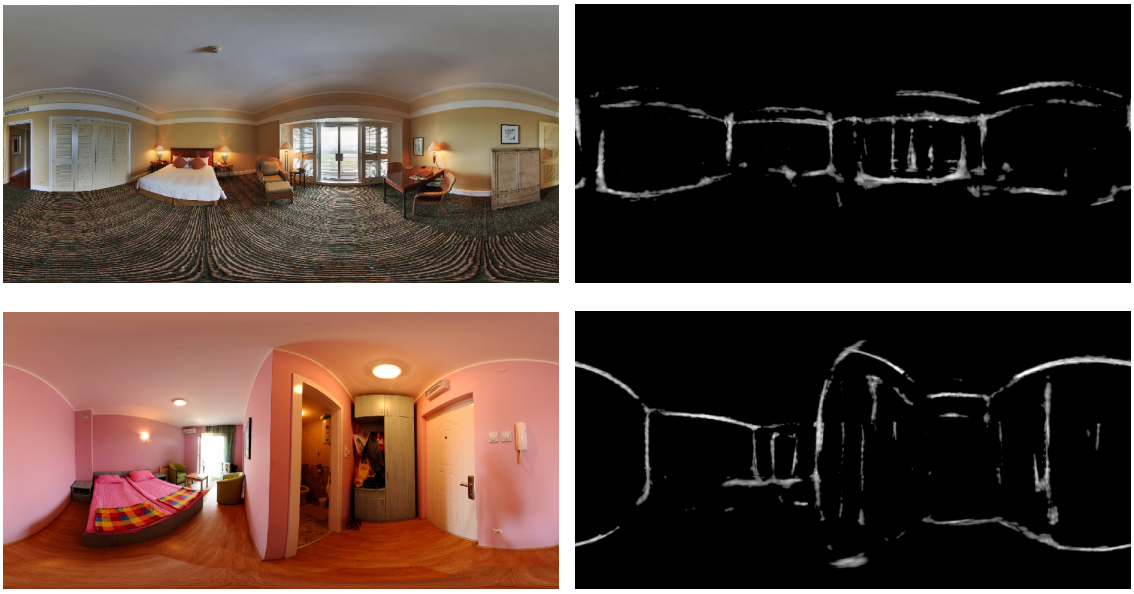


Figura 4.2: Entrada y salida de la red neuronal. Mapa de bordes estructurales.

Las zonas de solape han sido abordadas de manera que se elige en cada caso el valor máximo de probabilidad asociado a cada pixel para no perder información de manera que, si en una imagen en perspectiva no se ha detectado una línea pero en la adyacente si, podamos tenerla en cuenta en nuestro algoritmo. En la Fig. 4.2 se aprecian dos ejemplos donde a la izquierda se ve la imagen panorámica de entrada y a la derecha el mapa de bordes estructurales que se obtiene a la salida de la red neuronal.

4.2 Detección de Normales

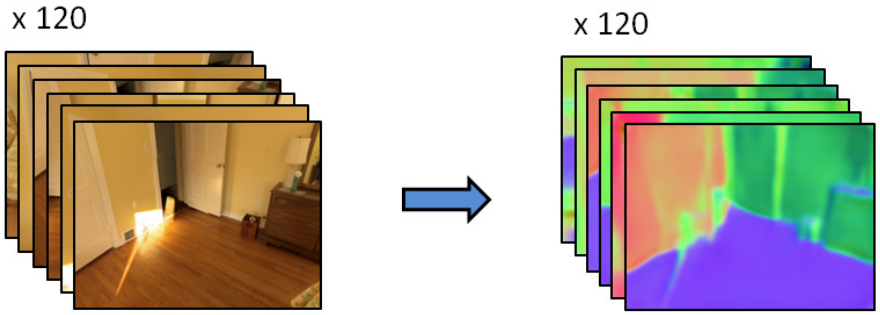


Figura 4.3: Entrada y salida de la red neuronal de Eigen *et al.* [7]

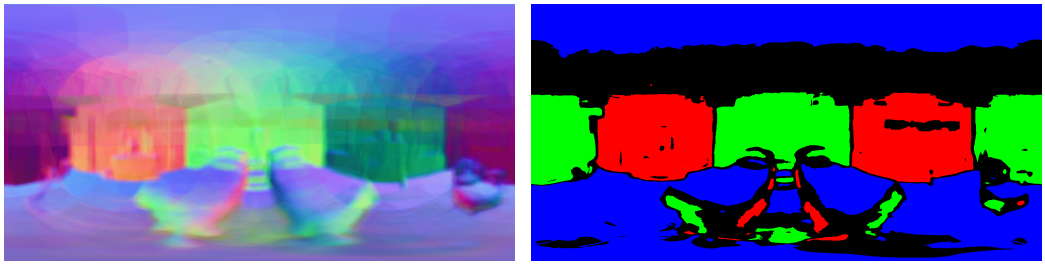
Eigen *et al.* [7] abordan en su trabajo tres tareas: predicción de profundidad, estimación de las direcciones normales de las distintas superficies y el etiquetado semántico utilizando una única red convolucional multiescala. Para nuestro trabajo hemos hecho uso de la parte de la red que extrae las normales de las superficies dado que nos permite obtener información por cada pixel de la orientación de cada superficie de la habitación. Esto lo utilizamos para evaluar las hipótesis de disposición de las paredes en lugar de los típicos mapas de características como los Orientation Maps (OM) o Geometric Context (GC). Eigen *et al.* llevan a cabo esta estimación mediante la predicción de las componentes x, y y z de la dirección normal por cada pixel.

En la Fig. 4.3 aparece en primer lugar un conjunto de imágenes que representa las imágenes en perspectiva que introducimos a la red neuronal, mientras que a su derecha se puede observar el mismo conjunto de imágenes procesadas ya por la red con los colores correspondientes en cada caso a la dirección de la normal de cada superficie.

En el caso de esta red neuronal, para devolver las imágenes de perspectiva obtenidas por la red a la imagen panorámica de nuevo, necesitamos rotar las normales para establecerlas en un marco de referencia común. Se realizan dos rotaciones, una primera rotación asociada a aquella llevada a cabo inicialmente para generar dicha imagen en perspectiva a partir de la imagen panorámica original (con las coordenadas de los puntos centrales de cada una de ellas mencionados al inicio de la sección) y una segunda rotación asociada a los puntos de fuga de la imagen panorámica completa.

Las áreas de solape para este caso se resuelven haciendo la media de los valores en cada pixel para lograr una mejor continuidad de la imagen global.

Un ejemplo de la imagen panorámica tras realizar las rotaciones a las normales obtenidas se puede ver en la Fig. 4.4a. Se ha observado que el techo es la parte peor estimada por la red ya que las imágenes con las que ha sido entrenada, como se ha comentado anteriormente, son de campo de vista limitado y en estas no suele aparecer el techo. A la hora de clasificar las paredes según su dirección de acuerdo a los puntos de fuga, a estas zonas conflictivas, clasificadas como tal en función de un *threshold* angular que determina si pertenecen o no a una dirección principal, les damos valor 0 para evitar confusiones. Aparecen en negro en la imagen de la Fig. 4.4b.



(a) Mapa de normales sin realizar rotaciones

(b) Mapa de normales rotadas

Figura 4.4: Transformación del mapas de normales tras salir de la red neuronal

Capítulo 5

Estimación del *Layout* de la Habitación

Como se ha mencionado varias veces a lo largo del trabajo, nuestro objetivo es extraer la estructura principal de una habitación, es decir, los bordes entre paredes, entre paredes y techo y entre paredes y suelo, obviando los objetos que se encuentra en su interior. Para ello hemos desarrollado un método que genera hipótesis de diseño de habitaciones a partir de posibles esquinas obtenidas combinando el razonamiento geométrico aplicado a la imagen panorámica y la información proporcionada por los procedimientos de *deep learning*. Nuestro algoritmo se divide en cuatro fases.

5.1 Eliminación de Líneas No Significativas

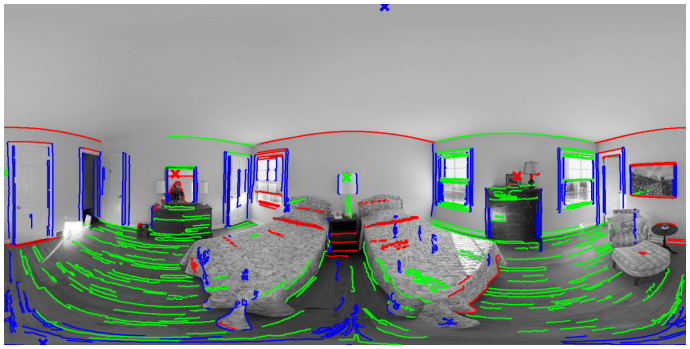
Es en esta sección donde la Fully Convolutional Network (FCN) propuesta por Mallya y Lazebnik [18] juega un papel importante.

La principal pieza de información utilizada para crear hipótesis de diseño son las líneas. Sin embargo, es imposible saber a priori de donde proceden éstas, dado que pueden provenir tanto de las deseadas intersecciones entre paredes, como de otros elementos de la escena (*e.g.* objetos). Con el fin de abordar este problema, proponemos evaluar las líneas extraídas en la imagen panorámica (Cap. 3) con el mapa de bordes informativos (Sec. 4.1).

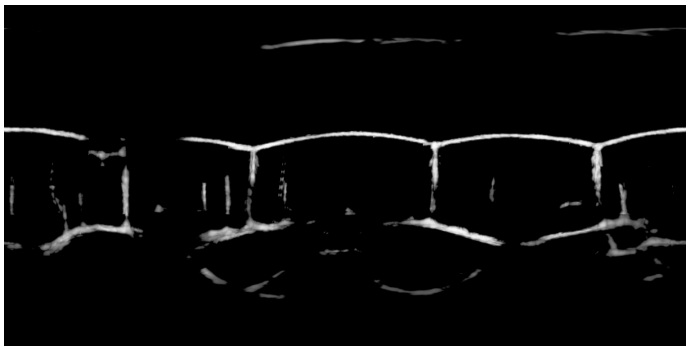
Cada línea viene asociada a una puntuación calculada como la suma de los valores de probabilidad de los píxeles que ocupa en el mapa de bordes. De esta manera, aquellas líneas cuya probabilidad se encuentre por debajo de un determinado *threshold* o directamente tengan probabilidad cero, serán eliminadas, mientras que las otras serán calificadas como líneas significativas. Esto nos permite trabajar directamente con las líneas que realmente nos dan información más precisa sobre las estructura principal de la habitación ignorando aquellas que pertenecen al desorden de la escena, *e.g.* muebles o plantas. Tras llevar a cabo esta fusión de información, el número de líneas se ve reducido a una tercera parte o incluso a una cuarta parte en muchos de los casos.

Un ejemplo de la evolución de este proceso se puede observar en las imágenes que se muestran a continuación. En la Fig. 5.1a, las líneas extraídas y orientadas en el Cap. 3 según los puntos de fuga. En la Fig. 5.1b, el mapa de probabilidades de bordes obtenido de la red neuronal 4.1. En la Fig. 5.1c, las líneas significativas tras combinar ambas herramientas, orientadas de acuerdo a los puntos de fuga. En estas imágenes

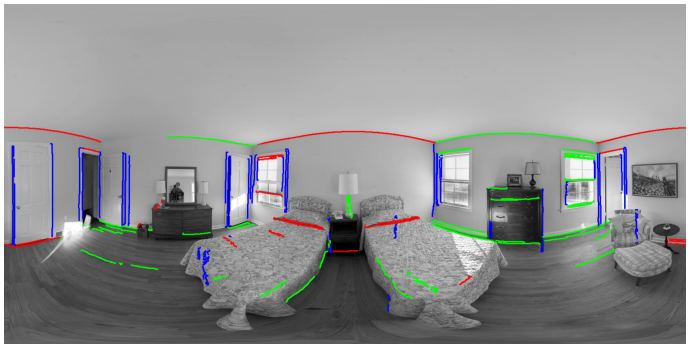
es posible apreciar claramente la ventaja que ofrece fusionar ambas herramientas. Se puede observar que prácticamente todas las líneas pertenecientes al parqué, a las mesillas e incluso muchas de ventanas, cuadros y puertas que suelen dar más problemas, han sido eliminadas y prácticamente sólo quedan líneas estructurales.



(a) Líneas obtenidas iniciales



(b) Mapa estructural



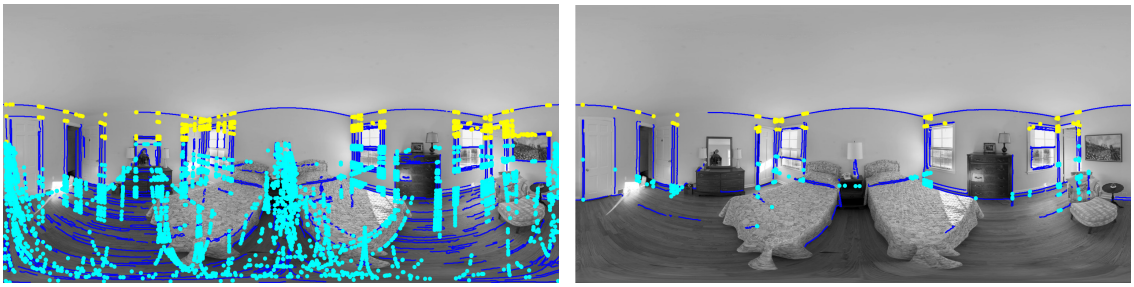
(c) Líneas significativas

Figura 5.1: Proceso de eliminación de líneas no significativas combinando la extracción de líneas inicial y la red neuronal de Mallya *et al.* [18]

5.2 Hipótesis de Esquinas Relevantes

Nuestro proceso de generación de *layouts* se basa en esquinas, es decir, intersecciones estructurales entre dos paredes, o entre paredes y techo o suelo.

En el mundo de Manhattan, dos líneas son suficientes para definir una esquina, por tanto, para obtener todas las esquinas posibles, intersectamos todas las líneas



(a) Esquinas candidatas sin usar la red neuronal (b) Esquinas candidatas usando la red neuronal

Figura 5.2: Hipótesis de esquinas de techo y suelo en color amarillo y cían respectivamente.

significativas (obtenidas como ha sido indicado en la Sec. 5.1) de diferentes direcciones (x, y, z) dos a dos, siempre y cuando éstas no se crucen (*i.e.* la extensión de la línea en la imagen no atraviesa el segmento de la otra). Otros trabajos como [12], tienden a dar más énfasis a las líneas verticales y a la extensión de estos segmentos a la hora de definir las esquinas, lo cual puede ser problemático si hay oclusiones o condiciones de mala luminosidad.

Para obtener la dirección del vector de la esquina basta con realizar el producto vectorial de las normales de las líneas que intersectan en dicha esquina, $Corner_{xyz} = n_i \times n_j$. Cada esquina tiene directamente una puntuación asociada dada por la suma de las puntuaciones de las líneas que la han generado. El proceso anterior de eliminación de líneas no significativas hace que estas esquinas sean ya buenas candidatas para las hipótesis de diseño, no siendo necesario así emplear algoritmos más complejos de puntuación basados en la longitud de las líneas o distancias entre estas [21]. La puntuación de las esquinas nos permite eliminar aquellas con puntuación menor a un determinado *threshold* determinado experimentalmente facilitando la posterior generación de hipótesis.

Las imágenes panorámicas tienen la ventaja de proveer una vista completa de la habitación, permitiéndonos siempre observar techo, paredes y suelo. Esto hace posible combinar una estimación separada de esquinas en la parte superior y en la parte inferior de la imagen o, lo que es lo mismo, por encima y por debajo de la línea de horizonte (lugar geométrico en el cual se encuentran todos los puntos de fuga de las proyecciones de las rectas horizontales en el espacio). Gracias a esta doble detección de posibles esquinas, para cada borde estructural tendremos detectada la esquina de un extremo u otro, haciendo posible obtener la complementaria por simetría.

Dado que sólo contamos con la dirección del vector de cada esquina y no con sus coordenadas 3D, asumimos que todos los vectores de las esquinas de cada hemisferio intersectan en un único plano de techo o suelo respectivamente. Los vectores normales de ambos planos son la dirección vertical del mundo de Manhattan, vp_z ó vp_3.

La Fig. 5.2a muestra todas las esquinas candidatas que selecciona el algoritmo sin utilizar como filtro la red neuronal de Mallya *et al.* [18]. En contraposición, en la Fig. 5.2b, aparecen las esquinas candidatas seleccionadas por el algoritmo tras hacer el filtrado con la red neuronal. En ambas imágenes se pueden observar en

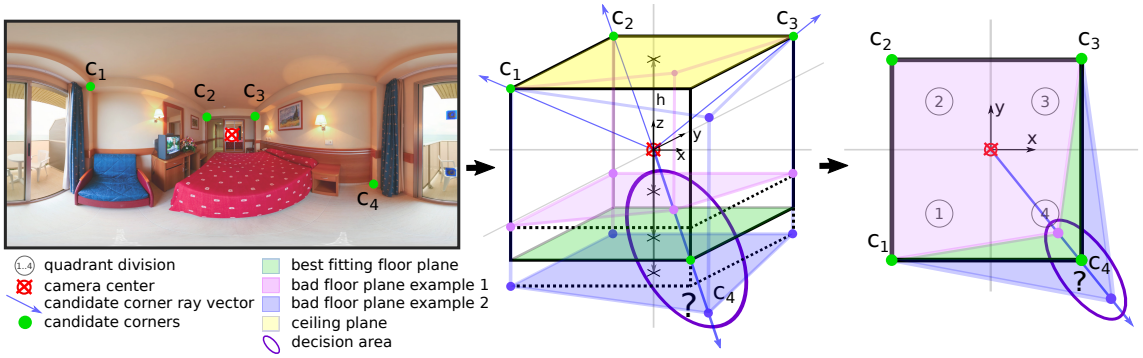


Figura 5.3: **Decisión de la altura de la habitación**: La solución que mejor encaja, cumpliendo con la asunción del mundo de Manhattan, nos proporciona una altura estimada de la habitación para nuestra hipótesis de diseño. Si la hipótesis obtenida se corresponde con una altura de habitación desproporcionadamente grande o pequeña, dicha hipótesis es eliminada.

amarillo las posibles esquinas detectadas sobre la línea de horizonte de la imagen que irán proyectadas sobre un plano techo y, en azul celeste, las detectadas bajo la línea de horizonte que se proyectaran en un plano suelo. A simple vista puede apreciarse la mejora que supone introducir el filtrado de la red, dado que en la Fig. 5.2b se ve claramente que muchas de las esquinas detectadas por nuestro algoritmo son ya buenas candidatas y que el número total de éstas para la posterior realización de hipótesis de diseño se reduce notablemente, logrando así reducir el número de iteraciones para lograr un diseño fiable y por tanto, reducir tiempos.

Más adelante, en la sección de experimentos (Sec. 6), se demuestra, además de visualmente como acabamos de hacer, numéricamente la mejora de precisión del algoritmo utilizando la red neuronal y sin utilizarla.

5.3 Generación de Hipótesis de Diseño / *Layout*

Dado que la generación de hipótesis es siempre un método iterativo y con el fin de reducir el número de iteraciones necesarias, procedemos inicialmente a realizar una distribución clara de la escena, dividiendo ésta en cuatro cuadrantes alrededor del centro de la cámara teniendo en cuenta los puntos de fuga como se muestra en las Figs. 5.3 y 5.4. Esta distribución resulta muy útil a la hora de inicializar el algoritmo dado que sabemos que, por ejemplo, en habitaciones de cuatro paredes siempre hay una esquina en cada cuadrante o, en habitaciones de seis paredes, siempre hay una esquina en cada cuadrante excepto en uno de ellos donde habrá tres esquinas.

Nuestro algoritmo para la generación de hipótesis de diseño comienza con una selección pseudo-aleatoria de entre las posibles esquinas. Decimos pseudo-aleatoria porque la selección esta sujeta a una serie de condiciones que simplifican el proceso:

- Se generan grupos iniciales de tres, cuatro o cinco esquinas de manera aleatoria en cada iteración
- Al menos debe existir una esquina en tres de los cuatro cuadrantes
- En el grupo seleccionado debe haber al menos una esquina de cada hemisferio de la imagen, *i.e.* al menos una esquina de techo y una de suelo. Esta condición

es importante por dos motivos. Por un lado, nos permite estimar la altura de la habitación y, por otro lado, nos permite encontrar esquinas que, no siendo visibles en uno de los hemisferios, si lo son en el otro (*e.g.* una intersección entre paredes puede estar ocluida en su parte inferior por un sillón y estar despejada en su parte superior, o puede estar ocluida en su parte superior por una cortina y estar despejada en su parte inferior).

Podemos definir por tanto un modelo de *layout* basado en los puntos de fuga ($vp_j, j = 1, 2, 3$) de la imagen y en un número de esquinas (*corners* en inglés, c_i) equivalente al número de paredes que tenga cada habitación (*e.g.* cuatro o seis).

$$Layout = (c_1, c_2, ..., c_n, vp_j))$$

Como se muestra en la Fig. 5.3, las esquinas candidatas se proyectan en el plano $x-y$ del modelo de la esfera y se ordenan de acuerdo a la dirección de las agujas del reloj. Las esquinas sobre la línea de horizonte se proyectan como un punto (c_1, c_2 y c_3) en un plano de referencia techo, *ceiling plane*, mientras que la esquina bajo la línea de horizonte se proyecta como un rayo (c_4) a lo largo del cual se buscará la solución que mejor se ajuste al modelo de diseño que estamos buscando. A continuación las hipótesis de diseño se generan uniendo las esquinas en orden, cuando sea posible, con paredes orientadas en el mundo de Manhattan [4].

Muchos trabajos simplifican el problema estimando habitaciones como cajas 3D de cuatro paredes, ya sea por falta de información por el uso de imágenes convencionales con menor campo de vista [10, 11, 23] o por restar complejidad al problema [31]. Aquí hemos querido dar un paso más y para ello nos enfrentamos a diseños más complejos introduciendo también la posibilidad de estimar esquinas intermedias entre las seleccionadas inicialmente por el algoritmo. En aquellos casos en los que el conjunto de esquinas seleccionado no puede generar una hipótesis con distintas paredes orientadas que satisfagan la asunción del mundo de Manhattan, se selecciona un nuevo conjunto de esquinas pasando a la siguiente iteración.

En la Fig. 5.4 se muestran dos ejemplos de generación de hipótesis de diseño para una habitación de seis paredes. En el ejemplo superior un grupo inicial de esquinas candidatas aleatorias es seleccionado (c_1, c_3, c_5). A continuación se inicia un proceso de unión de las esquinas comenzando por c_1 y encontrando en primer lugar un rayo espacial asociado a la esquina de suelo seleccionada. Para encontrar la posición optima de esta esquina a lo largo de su rayo 3D, el algoritmo busca posibilidades con las esquinas más cercanas y dibuja una solución intermedia, c_2. En el tercer cuadrante, teniendo en cuenta la dirección de las uniones que se han llevado a cabo previamente (x ó y), nuestro algoritmo decide cuál es la mejor solución para c_4. En el cuadrante vacío se lleva a cabo una intersección entre las esquinas más cercanas a éste obteniendo c_6. Para cada unión se comprueba la condición del mundo de Manhattan bajo un determinado *threshold* angular decidido mediante la realización de varias pruebas, ($90° \pm 5°$). En la parte inferior se muestra un ejemplo de hipótesis de diseño errónea que sería descartada por nuestro algoritmo por dos motivos. En primer lugar sería eliminada por no cumplir la asunción del mundo de Manhattan y en segundo lugar, en caso de que las paredes fueran más ortogonales y la hipótesis se llegase a generar, al compararla con el mapa de normales de [7] que utilizamos para evaluar la coincidencia de píxeles sería baja dado que se habría detectado una habitación de cuatro pareces y sin embargo es de seis, por tanto una hipótesis así nunca sería seleccionada por nuestro método como hipótesis final.

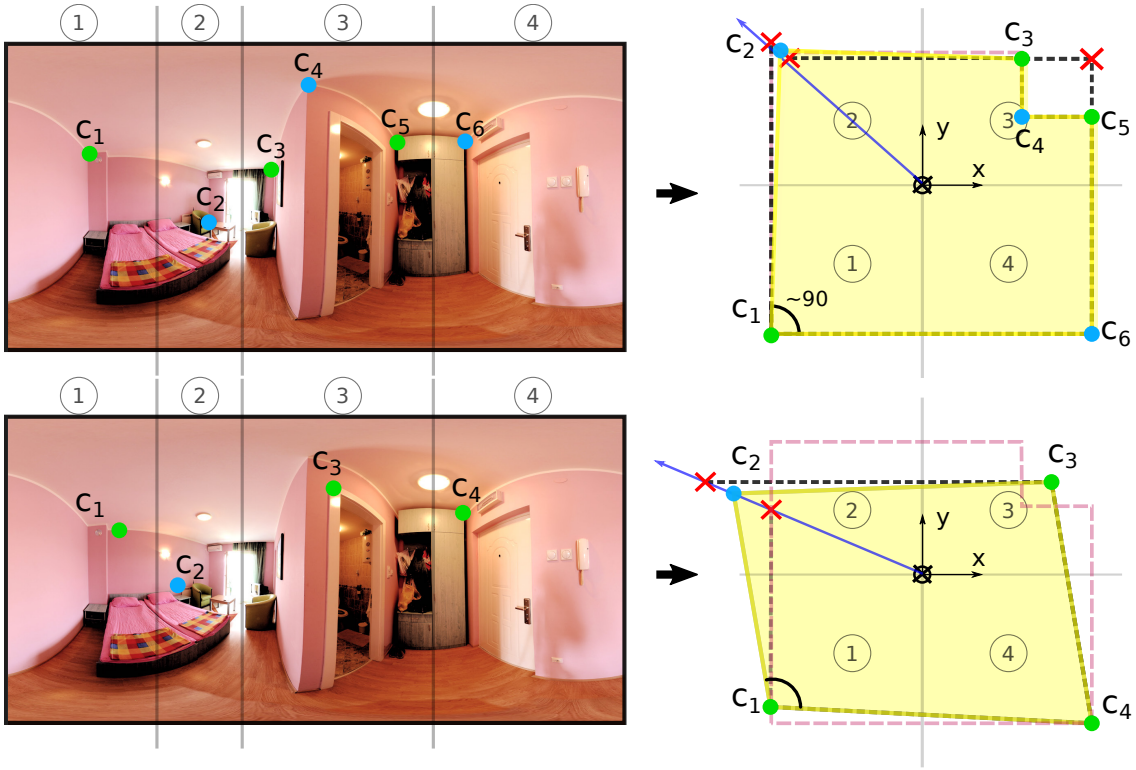


Figura 5.4: Ejemplos de generación de hipótesis de diseño para una habitación de seis paredes.

Sin pérdida de generalidad, como en trabajos previos [31, 27], asumimos que el centro de la cámara se encuentra a una determinada altura (*e.g.* 1.7 metros), lo que nos permite calcular la posición de ambos planos, techo y suelo, en 3D. Nuestro método encuentra la altura de techo que hace que la posición 3D de las esquinas produzca el mejor diseño Manhattan. Gracias a la simetría entre suelo y techo, ya sea un punto del límite entre las paredes y el techo o un punto entre las paredes y el suelo, es suficiente para especificar ambos.

5.4 Evaluación de Hipótesis de Diseño / *layout*

Para todas aquellas hipótesis de diseño 3D de la habitación que no sean descartadas y que, por tanto, cumplan con la asunción del mundo de Manhattan, se genera un mapa de normales a partir del resultado obtenido en dichas hipótesis.

Estos mapas son evaluados pixel a pixel con el mapa de normales obtenido a través de la red neuronal de Eigen *et al.* [7] explicada en la Sec. 4.2. Aquel que tenga el mayor número de píxeles coincidentes será la solución final y, por tanto, la mejor hipótesis de diseño.

Un ejemplo de como se llega a las hipótesis resultantes para habitaciones de cuatro o seis paredes respectivamente es mostrado a continuación en la Fig. 5.5.

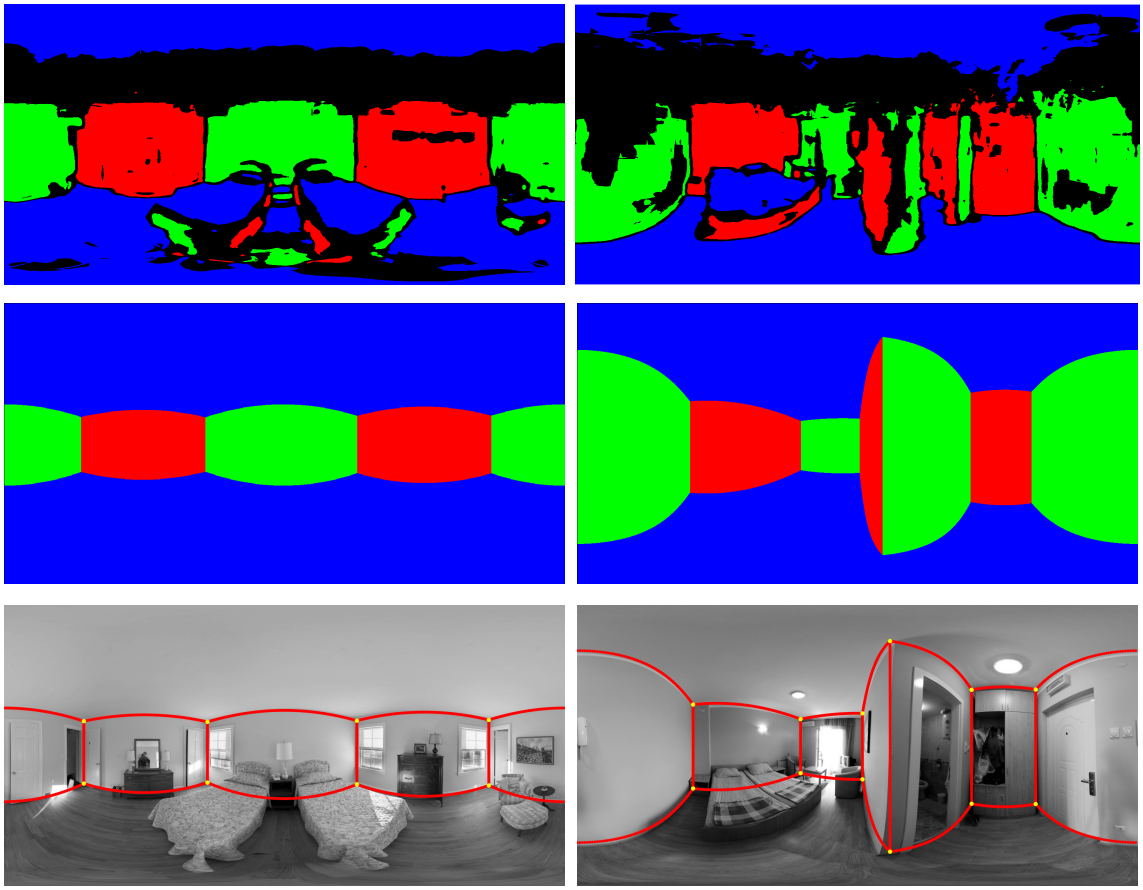


Figura 5.5: En la primera fila, los mapas de normales obtenidos de la red [7]. En la fila central, los mapas de normal asociados a las hipótesis de diseño generadas. En la ultima fila, la imagen con los bordes estructurales y las esquinas tras realizar la evaluación de hipótesis y obtener aquella para la que los mapas coincidían en un mayor número de píxeles.

Capítulo 6

Experimentos

Para la evaluación experimental hemos recogido un subconjunto de 46 imágenes panorámicas de la base de datos pública SUN360 (SceneUNderstanding360) [26]. La misma base de datos ha sido utilizada en trabajos previos [31, 27, 28].

Estas imágenes cubren 360° de campo de vista longitudinal y 180° de campo de vista latitudinal, utilizando proyección esférica o, lo que es lo mismo, proyección equirectangular. Todas ellas tienen una resolución de 9104×4552 píxeles que reducimos alrededor de seis veces para ahorrar tiempo de computación. Nuestra selección de imágenes incluye dormitorios con distinto número de paredes.

6.1 Herramientas y tecnología utilizada

A continuación se citan las tecnologías más relevantes que hemos utilizado para cada una de las partes del proyecto.

Redes Neuronales

- Mapa de bordes de probabilidades: Para la obtención de los mapas de probabilidades que proporcionan la estructura de la estancia se ha usado la red propuesta por Mallya *et al.* en [18]. Para ejecutar este modelo de red se ha utilizado Caffe [13], un *framework* de redes neuronales.
- Mapa de normales: Para la extracción de las direcciones normales de las distintas superficies de la imagen se ha utilizado la red que propusieron Eigen *et al.* en [7] y también los parámetros de la red. La red esta entrenada utilizando Theano [1] y hemos utilizado el mismo Framework con interfaz en Python.

Ambas redes han sido ejecutadas en GPU utilizando CUDA.

Desarrollo y experimentos Tanto el algoritmo de tratamiento de imagen, como la adaptación de las imágenes a las redes neuronales, la generación de hipótesis de diseño y los experimentos han sido implementados en Matlab. El entorno utilizado ha sido Ubuntu.

6.2 Generación de *Ground Truth*

Para obtener resultados de evaluación hemos creado nuestro propio *Ground Truth*. Para ello hemos generado un algoritmo que, a partir de las coordenadas de los puntos marcados a mano (esquinas) como se muestra en la Fig. 6.1, devuelve un mapa de normales en el que cada píxel de la imagen está etiquetado según la dirección de la superficie a la que pertenece.

No es necesario marcar todas las esquinas dado que las oclusiones lo hacen una tarea difícil. Hemos seguido las mismas condiciones que en la sección 5.3.

Figura 6.1: Ventana de etiquetado de *Ground Truth*

Zhang *et al.*, para su trabajo *PanoContext* [31], crearon una herramienta WebGL de anotación en la web para que cualquier usuario pudiese etiquetar las imágenes de este mismo dataset. El motivo principal por el cual no hemos aprovechado dicha anotación es que en *PanoContext* simplifican el diseño de las habitaciones a cajas de cuatro paredes y eso nos impide realizar una comparación con nuestros resultados. Además, el hecho de que haya sido etiquetado por usuarios de la red hace que no sea un *Ground Truth* muy preciso como hemos comprobado al observar algunos casos.

6.3 Resultados numéricos

En esta sección evaluamos nuestros resultados calculando el *Pixel Accuracy*, que mide el nivel de precisión teniendo en cuenta la coincidencia de píxeles entre el *Ground Truth* y la mejor hipótesis de diseño obtenida, dividido por el número total de píxeles en la imagen. Una explicación más detallada de cómo obtener la coincidencia de píxeles aparece en la Fig. 6.2. En esta imagen se muestran los mapas de normales correspondientes a lo que podría ser una hipótesis y el *Ground Truth* de una misma imagen, divididos en sus tres canales R, G, B respectivamente. Para obtener la coincidencia entre ambos mapas se compara cada canal con su correspondiente con el operador lógico &, que devuelve el valor booleano *true* si ambos operandos son *true*, *i.e.* devuelve el valor "1" en todos aquellos píxeles que estén en blanco en las dos imágenes que se comparan. De esta manera se suman todos los valores *true* de la comparación de los tres canales obteniendo el número de píxeles que coinciden

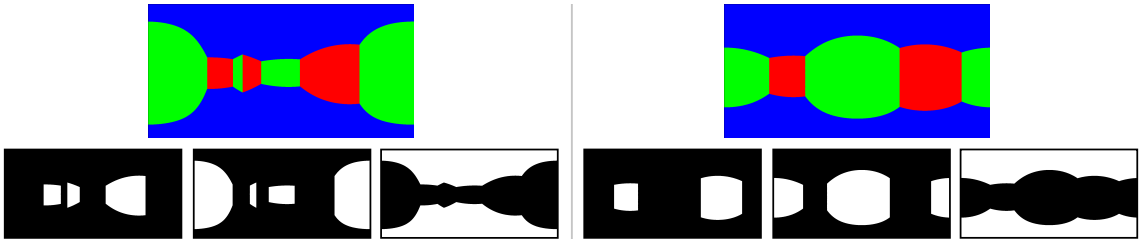


Figura 6.2: Mapas de normales divididos en sus tres canales R, G, B respectivamente.

entre ambos mapas de normales. En el caso de este ejemplo el número de píxeles coincidentes será bajo.

$$Pixel\ Accuracy = \frac{\text{n° de pixeles coincidentes}}{\text{n° de pixeles en la imagen}}$$

Cada resultado mostrado a lo largo de esta sección es un promedio de las 10 veces que se ha realizado cada experimento. El número de hipótesis generadas se especifica en cada experimento.

6.3.1 Comparación de nuestro método con el estado del arte

En este apartado llevamos a cabo una comparación con el trabajo *PanoContext* [31] dado que, hasta donde sabemos, es el único trabajo del estado del arte cuyo código esta disponible.

La comparación se lleva a cabo con la primera etapa de su algoritmo en la que obtienen, al igual que nosotros, estimaciones de diseño de habitaciones. En una segunda fase, este trabajo realiza detección de objetos y por tanto no tiene sentido compararnos en dicho punto.

En la Figura 6.3 mostramos un resultado gráfico en el que aparecen reflejados los resultados de cada trabajo variando el número de hipótesis de diseño evaluadas.

Podemos ver cómo claramente nuestro método supera al de *PanoContext*. La diferencia es mayor cuando se extraen pocas hipótesis, y disminuye a medida que aumenta su número obteniendo la mínima para 100 hipótesis. En particular, nuestro método con sólo 20 hipótesis proporciona mejores resultados que el de *PanoContext* con 100. Este resultado demuestra el buen desempeño de nuestra eliminación de líneas no significativas que permite a nuestro algoritmo proporcionar mejores hipótesis. Además, todas aquellas habitaciones que cuentan con un diseño más complejo, son resueltas por nuestro método y no por el suyo, ya que en *PanoContext* asumen siempre la simplificación de habitaciones con forma de caja de cuatro paredes.

6.3.2 Eliminación de líneas no significativas con el mapa de bordes

Para este experimento aplicamos nuestro método con y sin eliminar las líneas no significativas con el mapa de bordes obtenido usando la FCN de [18]. Seleccionamos arbitrariamente realizar 100 hipótesis de diseño antes de la evaluación de hipótesis. En la Fig. 6.4 mostramos un diagrama de barras con el resultado de precisión de píxeles para cada imagen del conjunto de datos elegido. Podemos observar cómo,

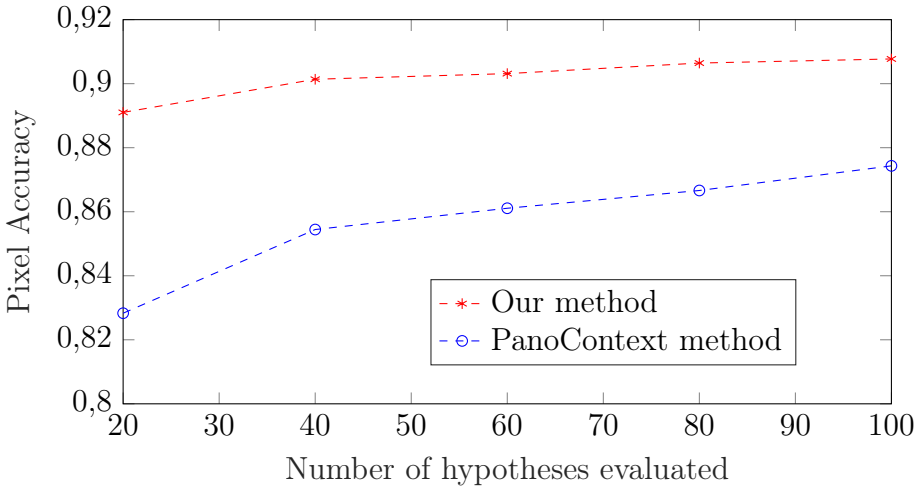


Figura 6.3: Comparación de nuestro método con el de [31]. Mostramos el *Pixel Accuracy* en función del número de hipótesis evaluadas. Nuestro método mejora al de [31] y es capaz de dar mejores resultados incluso con un número bajo de hipótesis.

en la gran mayoría de los casos, se experimenta una importante mejora en el resultado que no debe ser pasada por alto, obteniendo un $90,5\,\%$ de precisión media introduciendo la red y un $83,8\,\%$ sin introducirla. Esta mejora es debida al buen desempeño que ofrece la eliminación de líneas no significativas, ya que logra mantener prácticamente sólo las líneas estructurales de la habitación y rara vez elimina líneas verdaderamente útiles. Con esto queremos destacar los beneficios de nuestro enfoque que aprovecha la ventaja de fusionar métodos clásicos de visión por computador con nuevas técnicas de aprendizaje profundo.

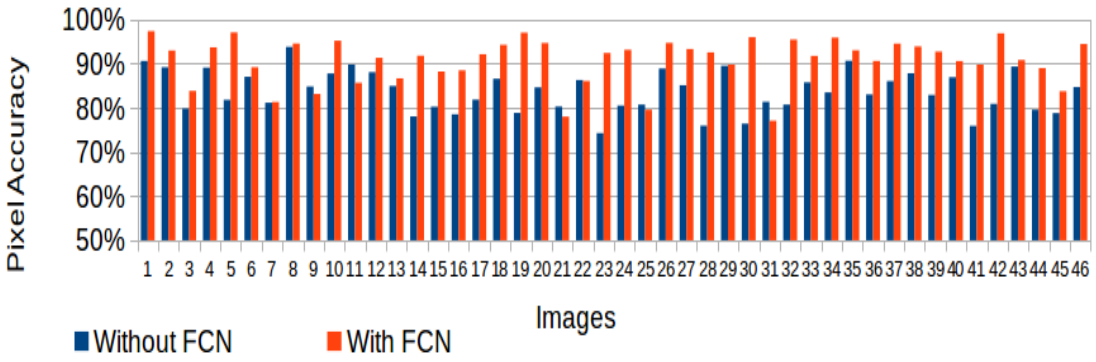


Figura 6.4: *Pixel Accuracy* empleando la FCN de [18] para eliminar líneas no significativas y sin emplearla. Obtenemos un $90,5\,\%$ de precisión media introduciendo la red y un $83,8\,\%$ sin introducirla.

6.3.3 Comparación del mapa de normales con otros mapas de características del estado del arte

En este experimento ejecutamos nuestro algoritmo como ha sido explicado a lo largo del trabajo, pero comparamos nuestros resultados sustituyendo el método de

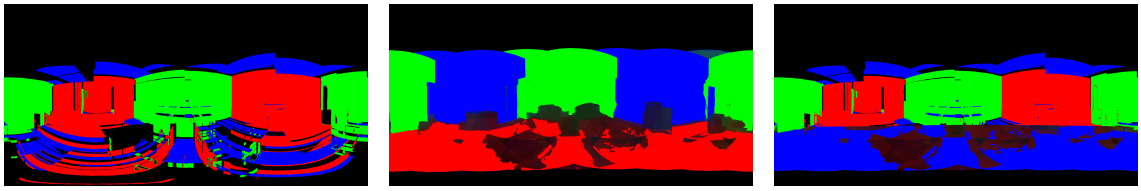


Figura 6.5: Métodos de evaluación del estado del arte: OM, GC y MM.

evaluación de hipótesis con otros tres del estado del arte. También usamos 100 iteraciones arbitrariamente para este experimento. Optamos por comparar nuestro método con el *Orientation Map* (OM) [15], el *Geometric Context* (GC) [10] y una combinación de ambos propuesta por los autores de *PanoContext* [31] a la que llamamos *Merge Map* (MM). El MM consiste en utilizar la parte superior del OM y la parte inferior del GC, lo que mejora los resultados ya que el GC elimina el desorden que aparece más a menudo en la parte inferior de la imagen (*e.g.* los objetos). Un ejemplo de cada uno de ellos se muestra en la Fig. 6.5, observándose en orden el OM, el GC y el MM, demostrando este último un buen desempeño pero con el coste de tener que generar los dos anteriores previamente que por sí solos ofrecen un peor resultado.

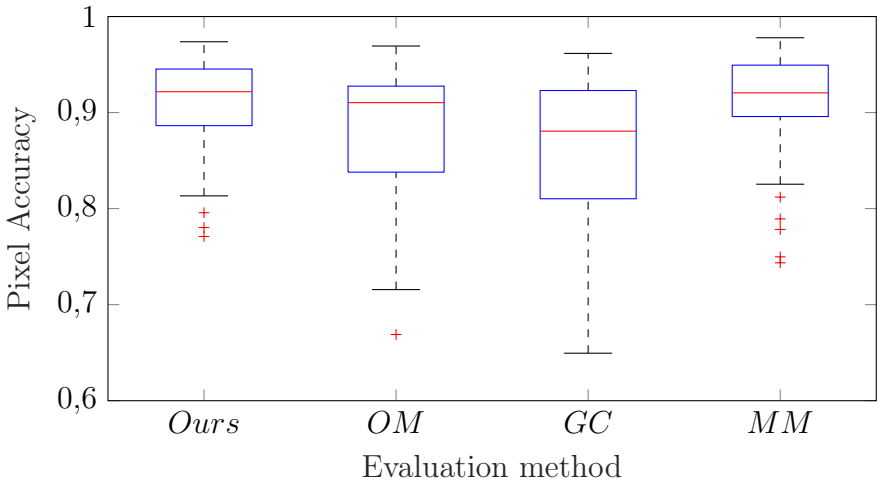


Figura 6.6: Boxplot del nivel de precisión de nuestro método de evaluación de hipótesis y otras tres alternativas del estado del arte. el Boxplot esta limitado por los cuartiles 25th y 75th con la mediana en su interior. Los bigotes alcanzan los valores más extremos y los *outliers* aparecen marcados con cruces.

En la Fig. 6.6 mostramos un Box-plot con los resultados de nuestro método y los otros mencionados donde se pueden ver valores mínimos y máximos de cada método, cuartil superior e inferior, valores de la mediana y valores atípicos o extremos.

Atendiendo a los resultados, podemos ver que nuestro método (media de precisión de los píxeles de 90.5 %) funciona mejor que el OM (89,7 %) y el GC (86.64 %) por separado, especialmente si se tiene en cuenta la varianza de los resultados. El MM funciona ligeramente mejor en media (91.8 %) pero genera más *outliers*. Es de esperar dado que es un método *ad hoc* que combina los puntos fuertes del OM y del GC. Creemos que en nuestro trabajo, la introducción de segmentación de objetos y su

eliminación del mapa de Normales mejoraría considerablemente los resultados.

6.4 Resultados visuales

En esta sección se muestra una colección de ejemplos de hipótesis finales de la estimación del diseño de habitaciones con distinto número de paredes. En la columna de la izquierda se muestran los bordes estructurales de cada habitación en rojo y las esquinas en amarillo. En la columna de la derecha aparecen las mismas imágenes con el mapa de direcciones normales de cada superficie superpuesto. Ver Figs. 6.7, 6.8 y 6.9.

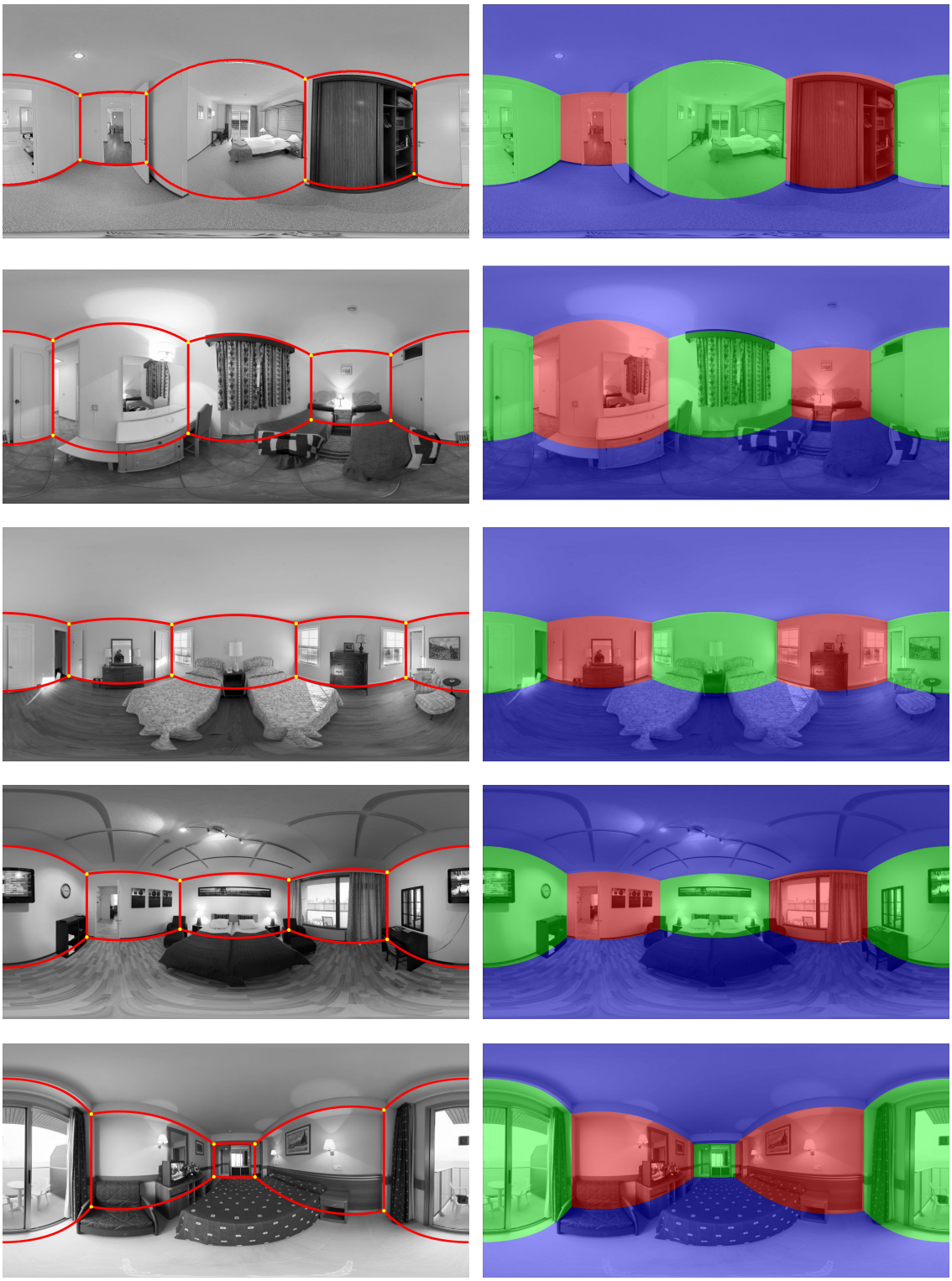


Figura 6.7: Habitaciones de cuatro paredes

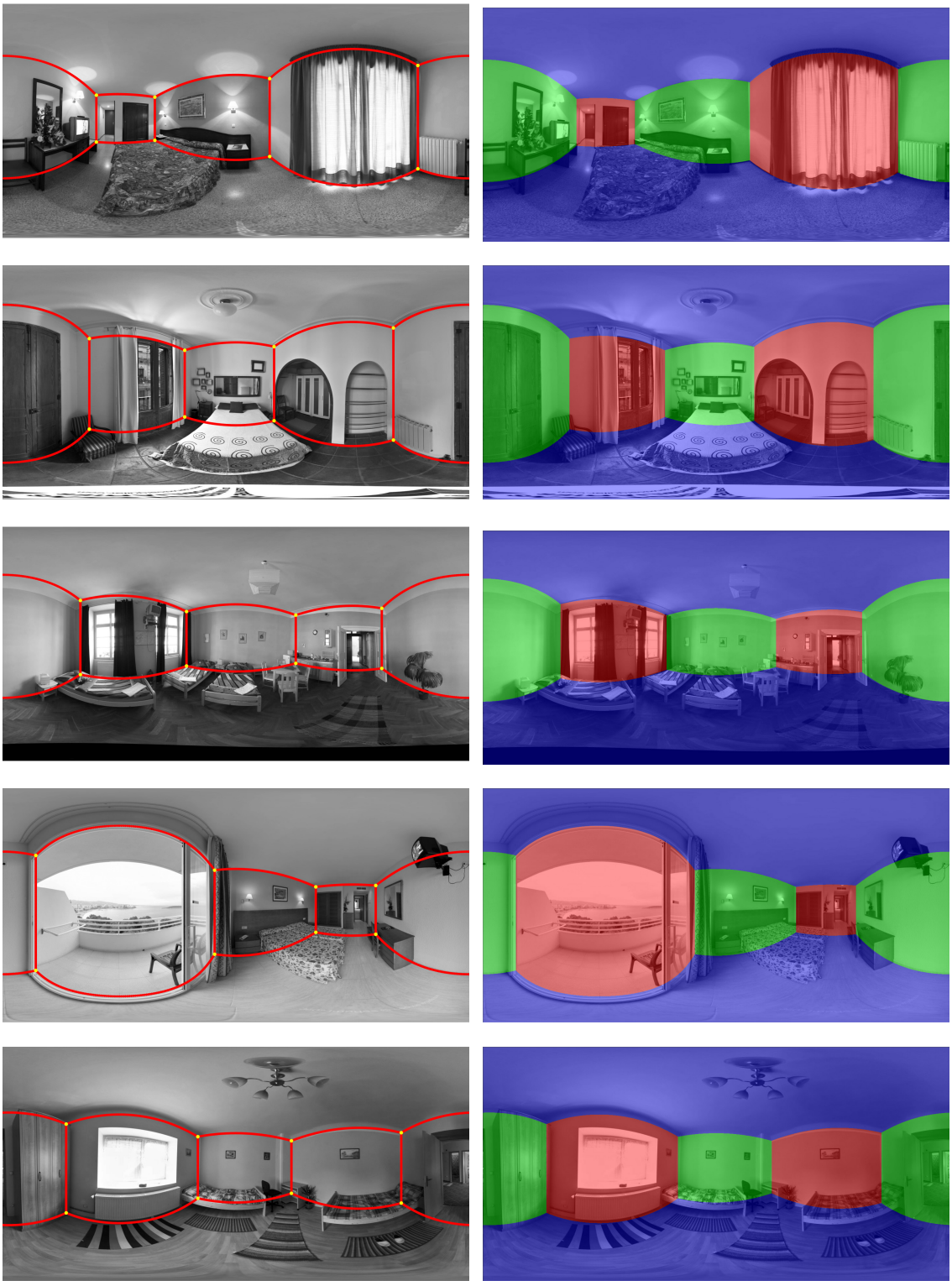


Figura 6.8: Habitaciones de cuatro paredes

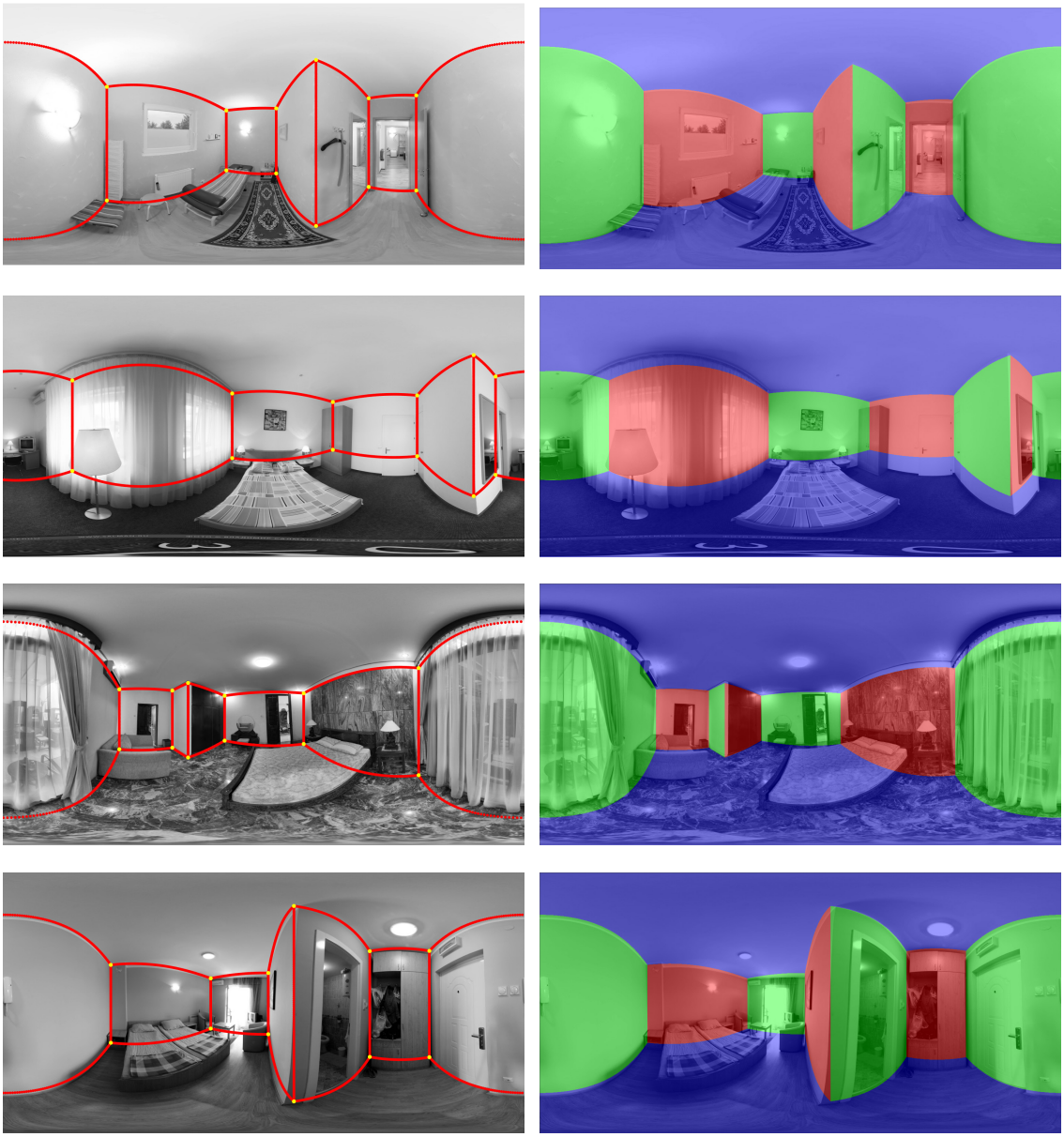


Figura 6.9: Habitaciones de seis paredes

Capítulo 7

Conclusión

Una de las principales contribuciones de este trabajo ha sido la explotación de técnicas de aprendizaje profundo, (*deep learning*), aplicadas al problema de la reconstrucción 3D de una escena a partir de una única imagen. En concreto, se han aprovechado las impresionantes ventajas que ofrecen dos redes neuronales de la literatura [18, 7] y han sido aplicadas en distintas fases de nuestro algoritmo. Otra de las contribuciones de nuestro método esta relacionada con la tipología de imagen que hemos seleccionado. En este trabajo se ha apostado por imágenes panorámicas debido a la gran ventaja que ofrecen por poseer un campo de vista completo de las escenas que muestran. Sin embargo, estas imágenes ofrecen una complejidad extra como consecuencia del tipo de proyección que utilizan y que impide aplicar directamente la mayoría de los métodos del estado del arte directamente en ellas (*e.g.* extractores de líneas y puntos de fuga, redes neuronales, etc.). Esto nos ha llevado a proponer algoritmos nuevos para su tratamiento y tareas de adaptación para su uso en redes neuronales, lo cual, hasta donde sabemos, no se había realizado hasta la fecha. Como tercera contribución, es importante mencionar que nuestro trabajo ofrece una gran flexibilidad sin restricciones en cuanto al número de paredes en el diseño de las habitaciones (a diferencia de muchos otros trabajos), campo de vista o calibración de cámaras, y ha sido realizado bajo la única asunción del mundo de Manhattan.

Nuestros resultados experimentales demuestran que el algoritmo propuesto tiene un buen desempeño en la interpretación de escenas con imágenes de visión completa y supera a otros trabajos del estado del arte. Además, la ventaja de combinar técnicas tradicionales de la visión artificial con nuevos algoritmos de aprendizaje profundo, también es apoyada por los resultados. Debido a originalidad del algoritmo y a la mejora que presenta con respecto a otros métodos del estado del arte, el contenido de este trabajo esta actualmente bajo revisión para ser presentado en la conferencia internacional del IEEE International Conference on Robotics and Automation (ICRA). El titulo del articulo es: *Geometry and Deep Learning in Layout Estimation from Panoramic Images*, y esta incluido en el Apéndice A.

Bibliografía

[1] R. Al-Rfou, G. Alain, A. Almahairi, C. Angermueller, D. Bahdanau, N. Ballas, F. Bastien, J. Bayer, A. Belikov, A. Belopolsky, et al. Theano: A python framework for fast computation of mathematical expressions. *arXiv preprint*, 2016.

[2] J. Bermudez-Cameo, G. Lopez-Nicolas, and J. J. Guerrero. Automatic line extraction in uncalibrated omnidirectional cameras with revolution symmetry. *International Journal of Computer Vision*, 114(1):16–37, 2015.

[3] J. Canny. A computational approach to edge detection. *IEEE Transactions on pattern analysis and machine intelligence*, (6):679–698, 1986.

[4] 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.

[5] S. Dasgupta, K. Fang, K. Chen, and S. Savarese. Delay: Robust spatial layout estimation for cluttered indoor scenes. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 616–624, 2016.

[6] E. Delage, H. Lee, and A. Y. Ng. A dynamic bayesian network model for autonomous 3D reconstruction from a single indoor image. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 2418–2428, 2006.

[7] D. Eigen and R. Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *IEEE International Conference on Computer Vision*, pages 2650–2658, 2015.

[8] A. Gupta, M. Hebert, T. Kanade, and D. M. Blei. Estimating spatial layout of rooms using volumetric reasoning about objects and surfaces. In *Advances in neural information processing systems*, pages 1288–1296, 2010.

[9] K. He, H. Chang, and J. Sun. Rectangling panoramic images via warping. *Transactions on Graphics*, 32(4):79, 2013.

[10] V. Hedau, D. Hoiem, and D. Forsyth. Recovering the spatial layout of cluttered rooms. In *IEEE International Conference on Computer Vision*, pages 1849–1856, 2009.

[11] V. Hedau, D. Hoiem, and D. Forsyth. Thinking inside the box: Using appearance models and context based on room geometry. *European Conference on Computer Vision*, pages 224–237, 2010.

[12] H. Jia and S. Li. Estimating structure of indoor scene from a single full-view image. In *IEEE International Conference on Robotics and Automation*, pages 4851–4858, 2015.

[13] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 675–678. ACM, 2014.

[14] C. Lee, V. Badrinarayanan, T. Malisiewicz, and A. Rabinovich. RoomNet: End-to-end room layout estimation. In *IEEE International Conference on Computer Vision*, 2017.

[15] D. C. Lee, M. Hebert, and T. Kanade. Geometric reasoning for single image structure recovery. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2136–2143, 2009.

[16] G. Lopez-Nicolas, J. Omedes, and J.J. Guerrero. Spatial layout recovery from a single omnidirectional image and its matching-free sequential propagation. *Robotics and Autonomous Systems*, 62(9):1271–1281, 2014.

[17] R. Lukierski, S. Leutenegger, and A. J. Davison. Room layout estimation from rapid omnidirectional exploration. In *IEEE International Conference on Robotics and Automation*, pages 6315–6322, 2017.

[18] A. Mallya and S. Lazebnik. Learning informative edge maps for indoor scene layout prediction. In *IEEE International Conference on Computer Vision*, pages 936–944, 2015.

[19] S. H. Oh and S. Jung. Ransac-based orthogonal vanishing point estimation in the equirectangular images. *Journal of Korea Multimedia Society*, 15(12):1430–1441, 2012.

[20] S. H. Oh and S. K. Jung. A great circle arc detector in equirectangular images. In *International Conference on Computer Vision Theory and Applications*, pages 346–351, 2012.

[21] A. Perez-Yus, G. Lopez-Nicolas, and J.J. Guerrero. Peripheral expansion of depth information via layout estimation with fisheye camera. In *European Conference on Computer Vision*, pages 396–412, 2016.

[22] Y. Ren, S. Li, C. Chen, and C.-C. J. Kuo. A coarse-to-fine indoor layout estimation (cfile) method. In *Asian Conference on Computer Vision*, pages 36–51. Springer, 2016.

[23] A. G. Schwing, S. Fidler, M. Pollefeys, and R. Urtasun. Box in the box: Joint 3D layout and object reasoning from single images. In *IEEE International Conference on Computer Vision*, pages 353–360, 2013.

[24] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus. Indoor segmentation and support inference from RGBD images. pages 746–760. Springer, 2012.

[25] R. G. Von Gioi, J. Jakubowicz, J.-M. Morel, and G. Randall. Lsd: A fast line segment detector with a false detection control. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(4):722–732, 2010.

[26] 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.

[27] J. Xu, B. Stenger, T. Kerola, and T. Tung. Pano2CAD: Room layout from a single panorama image. In *IEEE Winter Conference on Applications of Computer Vision*, pages 354–362, 2017.

[28] H. Yang and H. Zhang. Efficient 3D room shape recovery from a single panorama. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 5422–5430, 2016.

[29] S. Yang, D. Maturana, and S. Scherer. Real-time 3d scene layout from a single image using convolutional neural networks. In *IEEE International Conference on Robotics and Automation*, pages 2183–2189, 2016.

[30] W. Zhang, W. Zhang, K. Liu, and J. Gu. Learning to predict high-quality edge maps for room layout estimation. *Transactions on Multimedia*, 19(5):935–943, 2017.

[31] 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.

[32] M. Zuliani. Ransac toolbox for matlab. *web page] http://www. mathworks. com/matlabcentral/fileexchange/18555, Nov*, 2008.

Apéndice A

Artículo enviado al IEEE ICRA

Geometry and Deep Learning in Layout Estimation from Panoramic Images

Clara Fernandez-Labrador, Alejandro Perez-Yus, Gonzalo Lopez-Nicolas, Jose J. Guerrero

***Abstract*— In this work we have developed a method for 3D layout recovery of indoor scenes from a single 360 degrees panoramic image. This method has the main novelty of combining geometric reasoning on computer vision and deep learning techniques adapted to the proposed image geometry. Our method uses the extraction of structural corners as a starting point to construct layout hypotheses assuming Manhattan World and without any prior information about the room shape. In particular, corners are extracted as intersections of lines that are orthogonal in 3D space. This process has been enhanced with a Convolutional Neural Network that detects structural edges and allows filtering lines belonging to other non-relevant objects. From these possible corners we draw layout hypotheses and choose the best fitting solution to the normals' map extracted with another CNN. We show results of 3D layouts recovered from images of the SUN360 public dataset. We demonstrate the effectiveness of our method with respect to existing works and the advantages of the introduction of deep neural networks in the pipeline of the process.**

I. INTRODUCTION

Recent years have seen a growing interest in indoor scene understanding from a single image. It is an essential step for a wide variety of computer vision tasks and has recently received great attention from several applications like augmented reality, scene reconstruction or indoor navigation and SLAM [12]. Probably the first attempt to address this challenge was [3] which finds floor-wall boundaries by using a Bayesian network model. In contrast, Lee *et al.* [10] use line segments to generate layout hypotheses evaluating with an Orientation Map (OM).Other works [6], [7] try to simplify the problem by assuming that the room is a 3D box with only four walls, and using Geometric Context (GC) instead of OM, which additionally helps detecting clutter.

Most of these works use conventional images with limited field of view (FOV). Recently, some alternatives to extend the FOV have been proposed. Lopez-Nicolas *et al.* in [11] perform the layout recovery using a catadioptric system. In [16], layout hypotheses are made combining fisheye images with depth information that provides scale. Even 360 degrees panorama images have been used. These can be easily obtained nowadays with camera arrays, special lenses or automatic image stitching algorithms [5]. In [8], their method shows the advantages of having a complete scene view over partial views of the same scene with previous methods [10]. *PanoContext* [25] uses panoramas to recover both the layout (which is also assumed as a 3D box) and bounding boxes of the main objects inside the room. Similarly, [21] provides results not limited to simple box shaped rooms, relying on feature maps such as GC and OM like [25]. In [22] they treat the problem as a graph with lines and superpixels as nodes, solving it with complex geometric constraints instead.

Instituto de Investigación en Ingeniería de Aragón (I3A), Universidad de Zaragoza, Spain. {cfernandez, alperez, gonlopez, josechu.guerrero} @unizar.es

Fig. 1: **From input to output**. Top: Full-view panorama input. Below: Best layout hypothesis and 3D reconstruction from the single-view.

In the last years, researches have also tackled layout recovery problems with Deep Learning, and especifically Convolutional Neuronal Networks (CNNs), with impressive results. For example, [23] uses a CNN to segment the ground plane outperforming traditional methods. DeLay [2] provides separate belief maps of the walls, ceiling and floor of the scene. Alternatively, some works use CNNs to extract the informative structural edges of indoor scenes ignoring those edges from clutter [13], [24]. Instead, in [9] they predict the location of the room layout corners. These CNNs have good performance but they are always focused on traditional images with limited FOV. Other deep learning works are not related to layout retrieval but produce an interesting outcome for the task. For instance, Eigen *et al.* [4] extract an estimation of the depth and surface normals from simple RGB images.

In this work, we propose a method that combines geometric reasoning and deep learning techniques to estimate the full 3D layout from a single panoramic view. See Fig. 1 for a brief example. Despite their additional complexity, we choose panoramic images since, thanks to their wide FOV, the whole scene information is acquired at once, including the usually less cluttered ceiling part, and thus

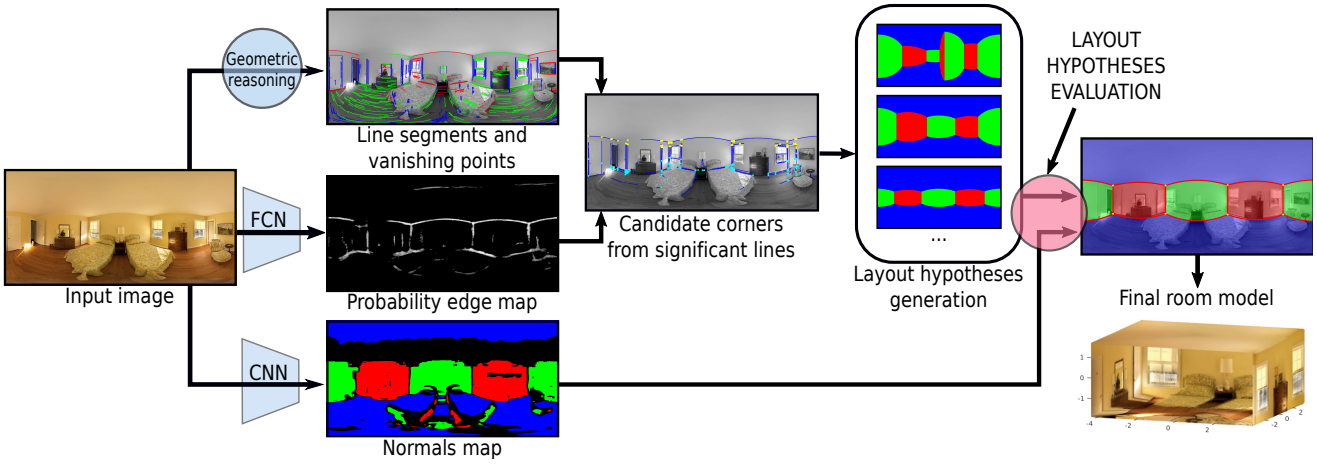


Fig. 2: **Algorithm overview**: We present a new approach for indoor layout estimation. Starting with a single panoramic image, the proposed method combines geometric reasoning on computer vision (lines and vanishing points estimation) and deep learning techniques adapted to the proposed geometry (two deep neural networks from the literature). Candidate corners are generated from significant lines after pruning the first line extraction with the edge map from [13] and layout hypotheses are generated from them. Hypotheses satisfying Manhattan world are evaluated, remaining as the final model the one which fits better with the normal map from [4].

allowing closed room solutions based on the best consensus distributed around the scene. However, unlike previous approaches [10], [25], [16], our method takes advantage of Deep Learning throughout the process. Recent research in the field shows that these data-driven approaches outperform traditional methods, that need to elaborate more and more complex reasoning to succeed in newer problems or accuracy requirements. The main novelty of our work is the exploitation of Deep Learning approaches for panoramic images applied to the problem of layout estimation, for which we propose a new flexible method that integrates old and new techniques.

The pipeline of our method is shown in Fig. 2: First, we extract line segments and vanishing points from the panorama. In parallel, the panorama is run through the network from [13] that allows to filter uninformative lines coming from clutter. Then the informative lines are used to extract corners as orthogonal line intersections, which are then used to draw layout hypotheses of diverse shapes. The hypotheses are compared to a reference normals map obtained with another CNN [4]. The best fitting solution is the final room model. Experimental evaluation with panoramic images from the public SUN360 database [20] of indoor environments shows an improvement with respect to the state of the art and reveals the advantages of using deep neural networks in the process.

II. Lines and Vanishing Points in Panoramas

Our proposal begins with extraction of lines from the image. There are many approaches for omnidirectional cameras, such as [1] that is able to extract the lines for a wide variety of dioptric and catadioptric systems without requiring previous calibration. Other alternative is [11], which uses Bazin's Matlab toolbox adapting the equations to hypercatadioptric system. *PanoContext* [25] works with panoramas and they split them to a set of perspective images and run the LSD algorithm [19] in each one separately and then project the lines back to the panorama. In [15] they extend the LSD method to deal with panoramas using the great circle arc detector.

We have developed a RANSAC method that works with panoramas directly without needing to split and rectify. Hence, our method is fast and shows entire and unique line segments, avoiding duplicate lines coming from different splits and thus improving the overall efficiency of the method. Since we work with panoramas we have to take into account that a straight line in the world is projected as an arc segment on a great circle onto the sphere and thus it appears as a curved line segment in the image. For this reason, each line is represented by the normal vector n_i of the 3D projective plane that includes the line itself and the camera center.

Our RANSAC based approach is as follows: First we run a Canny edge detector on the panorama. Then, we cluster the contiguous edge points in edge groups. Two points (spatial rays) of each group are randomly selected (r_i, r_j) to generate candidate line-images which are voted by the other points of the same group. To do that its vector product is computed obtaining a possible normal direction for this edge group, $n_{1...n} = (r_i \times r_j)_{1...n}$. The normal obtained is compared with the other rays of the group, considering *inliers* those that fulfill the condition of perpendicularity with the normal, n_i, under a certain angular threshold (*e.g.* 0.5°). This process is repeated a certain number of iterations and outputs the model leading to the highest number of *inliers* giving the normal direction that best fits the line. Edge groups with few points are discarded.

From these lines we extract the vanishing points (VPs)

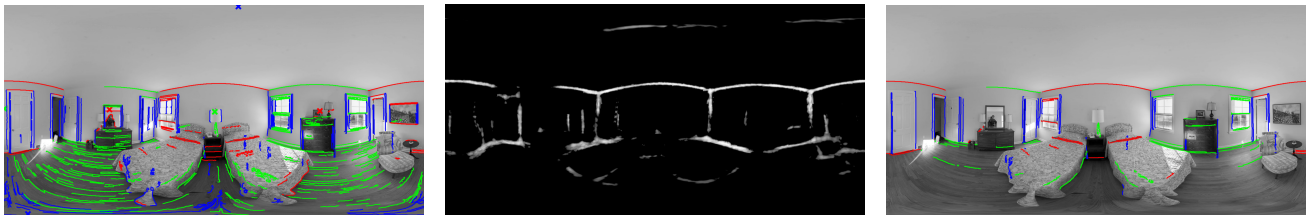


Fig. 3: Left: Oriented lines and vanishing points estimation with geometric reasoning on computer vision. Center: Mallya's *et. al.* [13] FCN output after stitching all the perspective images back to the panorama showing the resultant edge probability map. Right: Resulting lines after combining the extracted lines of the panoramic image and the informative edge map showing a large reduction of the number of lines, remaining those more significant.

applying another RANSAC method based on [14]. We adopt the Manhattan World assumption: there exist three dominant orthogonal VPs in the sphere aligned with three dominant directions in the world. It is worth mentioning that parallel lines in the 3D world intersect in one single VP while in spherical images, line projections result in curved line-images so that parallel lines intersect in two antipodal VPs. The line segments are classified according to the Manhattan directions, so we determine which line is associated to which VP. Lines whose normals are not perpendicular to neither of the directions are discarded. The lines which are associated with the same Manhattan direction are shown as identical color in Fig. 3 (left and right).

III. Deep learning techniques

Convolutional neural networks have been successfully applied to a wide variety of tasks such as object detection, scene classification or semantic segmentation. But in the last years, researchers have explored the possibility of using such CNNs for room layout estimation. In this work, we do not directly train an end-to-end neural network with omnidirectional images because, to the best of our knowledge, it does not exist any dataset with the enough amount of labeled data or wide variety of data distribution required to train a deep neural network (DNN). Instead, there are large enough indoor scene labeled datasets that have been successfully used for training DNNs (e.g. the NYUDv2 RGBD dataset [18]). Here, we choose to adapt two DNNs from the literature trained with conventional images [13], [4] to the image geometry proposed. Thus, our approach shows that it is possible to take advantage of these novel techniques without needing to label huge datasets and to train complex networks.

For this task, we proceed splitting the panoramas into a set of overlapping perspective images with a FOV similar to conventional images and planar projection. We run the algorithms in each of them separately and finally stitch them all back to the panorama by means of warping [20], solving the overlapping zones in each case in different ways. To define the set of virtual perspective images compounding the panorama, we choose, on the one hand, the different points that will be the center of each perspective image distributed along the 360 degrees in horizontal and 180 degrees in vertical of the spherical image and, on the other hand, the FOV that we want to apply to each of the images. The choice of the FOV was based on experimental results, where with 70 degrees we obtained good results. Now we describe the procedure for each network:

A. *Informative edges detection*

Mallya *et al.* [13] propose a Fully Convolutional Network (FCN) which has been trained to estimate probability maps representing the room edges of the projected 3D box that fits better with the room, even in the presence of clutter and occlusions. To improve the edge map avoiding noise, we remove low probability pixel values below a certain experimental threshold (0.2 out of 1). In the overlapping regions of the virtual perspective images compounding the panorama, we choose the maximum value of probability to not lose information. Fig.3 (Center) shows an example of informative edges detection on a panoramic image.

B. *Normals detection*

Eigen *et al.* [4] address three tasks: depth prediction, surface normal estimation and semantic labeling using a single multiscale convolutional network. For our work we use the surface normal estimation since it provides pixelwise information of the walls orientation, which we use to evaluate layout hypotheses rather than the typical feature maps such as Orientation Maps (OM) or Geometric Context (GC). The network provides a prediction of the x, y and z components of the normal direction at each pixel.

In this case, in order to stitch perspective images back to the panorama we need to rotate the normals to set them in a common reference frame. Two rotations are carried out: a first rotation associated with that initially performed to generate the perspective image from the original panorama (with the coordinates of the center points of each image) and one second rotation associated with the scene VPs. Overlapping areas are tackled in this case by doing the per-pixel average to achieve a better continuity of the overall image. Then we apply an angular threshold to determine whether or not the normals from each pixel belong to a main direction and label them accordingly. Resulting normal map is shown in Fig.4 (Left). It can be noticed that the ceiling is the worst part estimated by the network, since black pixels means uncertain areas (i.e. not belonging to any

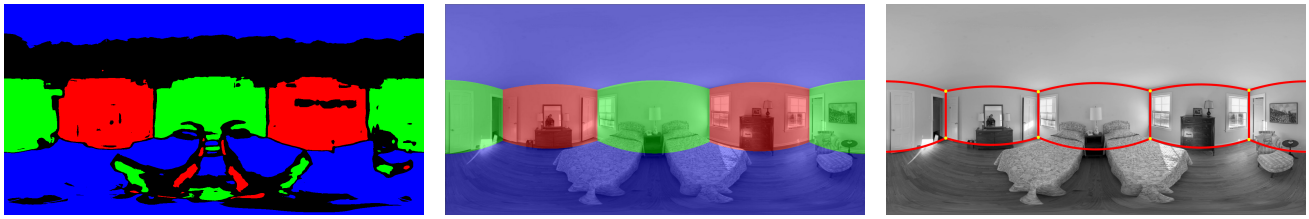


Fig. 4: Left: Map of normals given by the neural network of Eigen *et. al.* [4] after stitching all the perspective images back to the panorama. Center: Map of normals generated from the best hypothesis at the evaluation stage. Right: Structural edges on image with lines and corners of the best estimated layout hypothesis.

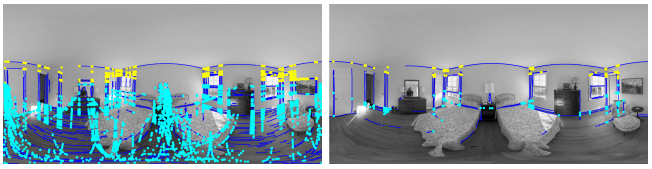


Fig. 5: Candidate corners from both ceiling and floor in yellow and cyan color respectively before introducing the neuronal network from [13] on the left and after that on the right. This significant reduction makes these corners already good candidates for the hypotheses generation stage.

main direction). This happens because in the training data the ceiling does not usually appear and thus, the network cannot predict the normals as expected in that areas.

IV. Room Layout Estimation

Our goal is to extract the main structure of an indoor environment *i.e.* the distribution of floor, ceiling and walls, abstracting all objects within rooms. For this purpose we have developed a method to generate layout hypotheses from relevant corners by combining line-based geometric reasoning on the panoramic image and the information provided by deep learning procedures. Our algorithm is divided in four stages:

A. *Non-significant lines removal*

It is in this section where the Fully Convolutional Network (FCN) proposed by Mallya and Lazebnik [13] plays an important role. The main piece of information we use to create layout hypotheses are lines. However, we do not know a priori whether they come from actual wall intersections, or from other elements of the scene. In order to tackle this problem we propose to evaluate the extracted lines on the panoramic image (Sec. II) with the informative edge map (Sec. III(a)). Each extracted line is associated to a score calculated as the sum of the corresponding probability values to the pixels it occupies in the edge map. In this way, those lines whose score is below a certain threshold, or directly have zero probability, are removed, while the others are classified as significant lines. This allows us to work directly with the lines that give us more significant information about the main structure of the room without taking into account those that belong to clutter. After carrying out this merger of information, the number of lines is reduced to one-third or even a quarter in several cases.

An example of this process can be observed in Fig. 3. On the left, the image lines drawn and oriented according to VPs, at the center, the probability edge map obtained in Sec. III(a) and on the right the significant lines after combining both tools. It is possible to see clearly the advantage of merging both approaches. It can be observed that practically all the lines belonging to the parquet, the tables and even many windows, pictures and doors that usually give more problems, have been removed and practically only structural lines remain.

B. *Relevant corner hypotheses*

Our layout generation process is based on corners, *i.e.* structural intersections between two walls and ceiling or floor. In a Manhattan World, two line segments are enough to define a corner so we intersect all the significant lines in different directions (x, y, z) among themselves in pairs as long as they do not cross each other. Other works such [8] tend to give more emphasis to vertical lines and the extension of these segments when defining corners, which can be problematic if there are occlusions or poor lighting conditions.

The direction vector of the corner point is computed with a cross product of the normal of the lines intersecting in that corner, $Corner_{xyz} = n_i \times n_j$. Each corner has an associated score given by the scores sum of the lines that have generated this corner. The previous elimination process of non-significant lines makes these extracted corners already good candidates, instead of needing complex scoring algorithms based on line length and distances, like in other works [16]. The corners scores allow us to eliminate those with a score lower than a certain threshold facilitating the subsequent generation of hypotheses.

Panoramic images have the advantage of providing an entire view of the room, allowing us to always observe ceiling, walls and floor, making it possible to combine corner estimation separately from both above and below the horizon line in the image. Thanks to this double detection of possible corners we ensure that, for each structural edge, we will have

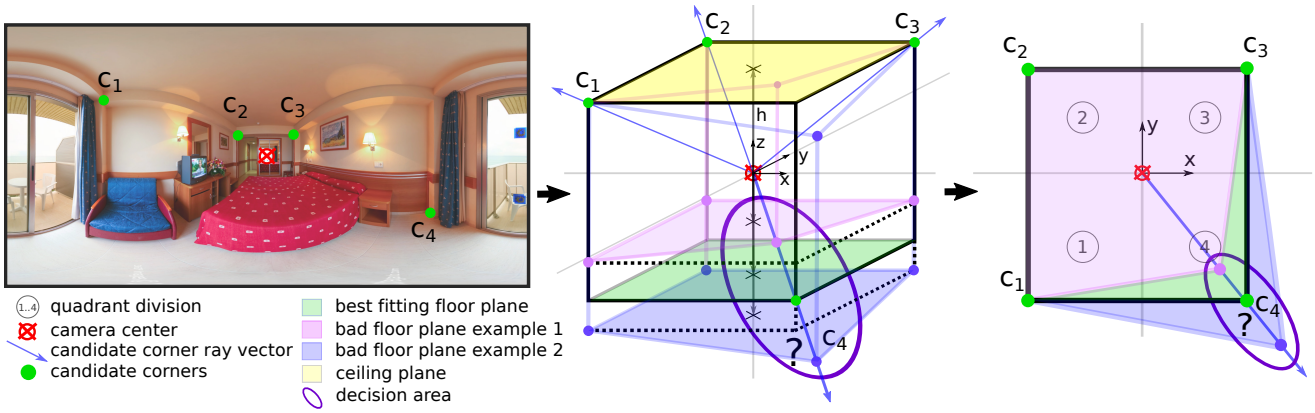


Fig. 6: **Room height decision**: The best fitting solution fulfilling the Manhattan assumption provide us an estimated room height for our layout hypothesis.

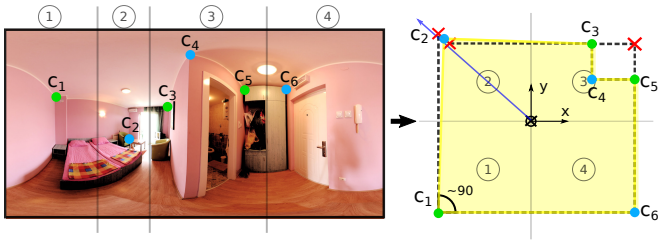


Fig. 7: **Layout hypothesis generation**: In this example, an initial random group of candidate corners is selected (c_1, c_3, c_5). Then, a joining corner process starts from c_1 finding at first a floor spatial ray. In order to find the optimal corner position along this ray, the algorithm finds possibilities with its nearest corners and draws an intermediate solution, c_2. In the third quadrant, taking into account the direction $(x - y)$ from previous unions, our algorithm would decide which solution is better for c_4. In the empty quadrant a simple intersection between nearest corners is done obtaining c_6. For each union the Manhattan assumption is checked with a certain threshold $(90° \pm 5°)$.

detected the corner of one end or another, making it possible to obtain the complementary one by symmetry.

While we do not have the 3D coordinates of the corners but just their direction vector, we assume all corner vectors from each hemisphere intersect in a single ceiling and floor plane respectively, whose normal vectors are the vertical Manhattan direction in both cases.

Fig. 5 shows, on the left, all candidate corners selected by the algorithm without filtering with the Mallya's *et al.* [13] neuronal network. In contrast, on the right, the candidate corners selected by the algorithm appear after filtering with the neural network. In both images appear in yellow the possible corners detected above the horizon line of the image that will be projected on a ceiling plane and, in indigo blue, those detected under the horizon line that will be projected in a floor plane. Getting a similar result by filtering this just with geometry constraints and reasoning will be much costly and prone to errors, whereas this approach filters the data efficiently thus reducing both the number of iterations needed to achieve a reliable design and computing times.

C. *Layout hypotheses generation*

We generate layout hypotheses by means of an iterative method. Therefore, in order to reduce the number of iterations needed, we proceed initially to make a clear distribution of the scene, dividing it into four quadrants around the center of the camera and taking into account the VPs. This distribution is very useful when initializing the algorithm since we know that, *e.g.*, in rooms with four walls there is always a corner in each quadrant or, in more complex rooms, *e.g.* rooms with six walls, there is always a corner in each quadrant except in one of them where there will be three corners.

Our algorithm for layout hypotheses generation starts with a pseudo-random selection of possible corners subject to the next conditions:

- Initial groups of three, four or five corners are randomly generated at each iteration
- There must be corners in at least three of the four quadrants. Thus, the corner in the fourth quadrant can be estimated assuming closed Manhattan layouts.
- In the selected group there must be at least one corner of each hemisphere of the image, *i.e.* at least one ceiling and one floor corner. This condition is important for two reasons. On the one hand, it allows us to estimate the height of the room and, on the other hand, it allows us to find corners that, not being visible in one hemisphere, they are in the other.

We can therefore define a layout model based on the image VPs v_{xyz} and on a number of corners $c_{1...n}$ equivalent to the number of walls that each room has.

$$Layout = (c_1, c_2, ..., c_n, v_{xyz})$$

As the Fig. 6 shows, the candidate corners are projected in the $x - y$ plane of the sphere model and are ordered clockwise. The corners above the horizon line are projected as a point (c_1, c_2 and c_3) on the ceiling reference plane,

while the corner below the horizon line is projected as a ray (c_4) along which we find the best fitting floor plane for the design model we are looking for. Then layouts are generated by joining corners in order with Manhattan-oriented walls whenever possible.

Many works simplify the layout generation problem by assuming that the room is a 3D box of four walls, sometimes because of lack of information due to the use of conventional images with less FOV [6], [7], [17], or just to subtract complexity to the problem, [25]. Here, we go one step further and face more complex designs introducing the possibility of estimating in-between hidden corners when required. Whenever the set of corners withdrawn cannot generate layout hypotheses with alternatively oriented walls satisfying Manhattan assumption, a new set of corners is selected. In Fig. 7 an example of layout hypothesis generation is shown and explained.

Without loss of generality, like in previous works [25], [21], we assume that the camera center is placed at a typical height (*e.g.* 1.7 meters), which allows us to compute the floor plane and therefore the position of the corners in 3D. Our method finds the ceiling height so that the 3D position of the corners would produce the best Manhattan layout. Due to the ceiling-floor symmetry, either a point in the ceiling-wall boundary or in the floor-wall boundary is sufficient to specify both. Additionally, layout hypotheses with abnormal values of height of the ceiling can be discarded.

D. Layout hypotheses evaluation

For all those layout hypotheses that fulfill the Manhattan world assumption, a normal map is generated from the result obtained in such hypotheses. These maps are evaluated pixel-to-pixel with the normal map obtained through the Eigen *et al.* neural network [4]. The one with the largest number of equally-oriented pixels will be the final solution and, therefore, the best design hypothesis.

V. Experiments

For the experimental evaluation we have collected a subset of 46 full-view equirectangular panoramas of indoor scenarios, from the public SUN360 database [20]. The same database was used in prior work [25], [21], [22]. All of them have a resolution of 9104×4552 pixels, which we downsize around six times to save computation time. Our selection of images include bedrooms with different number of walls not restricted to box-shaped rooms.

To obtain results we have use the Ground Truth (GT) manually labeled by ourselves, in which each pixel in the image is labeled according to the direction of the surface it belongs to. *PanoContext* had made also a GT with the same dataset but it was no useful since all cases were considered as four-wall rooms. We evaluate our results computing the *Pixel Accuracy*, which measures the accuracy level by taking into account the pixel coincidence between the labels of the ground truth and the labels of the best hypotheses divided by the total number of pixels in the image. Each result shown is an average of 10 times performing the experiment. The hypotheses number drawn are specified in each experiment.

a) ***Comparison of our method with the state of the art****:* We perform a comparison with PanoContext [25], since it is to our knowledge the only method with available code. We proceed to compare it with the first stage of their algorithm that reaches the same point as our work does, since after layout extraction they introduce object detection in the method. In Fig. 9 there is a graphical result that shows the results from each method varying the number of hypotheses evaluated. We can see how our method clearly outperforms PanoContext. The difference is larger when only a few hypotheses are drawn, and decreases as the amount of hypotheses rises. In particular, our method with only 20 hypotheses provides better results than the PanoContext with 100. This result shows the good performance of our non-significant lines removal which allows our method to provide better hypotheses. Besides, some scenes have more than four walls, which our method is able to solve unlike theirs.

b) ***Non-significant lines removal with the edge map****:* For this experiment we apply our method with and without removing non-significant lines with the edge map obtained using the FCN from [13]. We arbitrarily choose to draw 100 layout hypotheses prior the hypotheses evaluation. In Fig. 8 we show a bar diagram with the pixel accuracy result for each image of the dataset, and we can observe how in the great majority of cases there is an important improvement that must not be overlooked. With this we want to highlight the benefits of our approach with the advantage of fusing classical computer vision methods with new deep learning techniques.

c) ***Comparison of the Normals Map with state of the art methods****:* In this experiment we run our algorithm, but we compare our results substituting the hypotheses evaluation method with three others from the state of the art. We use 100 iterations arbitrarily for this experiment as well. We choose to compare our method with the Orientation Map (OM) [10], the Geometric Context (GC) [6] and a combination of the two that was proposed by the *PanoContext* authors [25] and that we call *Merge Map* (MM). The MM consists of using the upper part from the OM and the lower part of the GC, which improves results since the GC removes clutter which appears more often in the lower part of the image. In Fig. 10 we show a boxplot with the results of our method and the others mentioned. Looking at the results, we can see that our method (mean pixel accuracy = 90.5%) performs better than OM (89,7%) and GC (86.64%) alone, especially considering the variance of the data. The MM performs slightly better in average (91.8%) but producing more outliers. That is to be expected since this is *ad hoc* method which combines strengths of OM and GC. We believe that introducing an object segmentation and removal from the Normals map would improve the results considerably.

VI. Conclusion

In this work we have introduced the main novelty of the exploitation of Deep Learning approaches for single

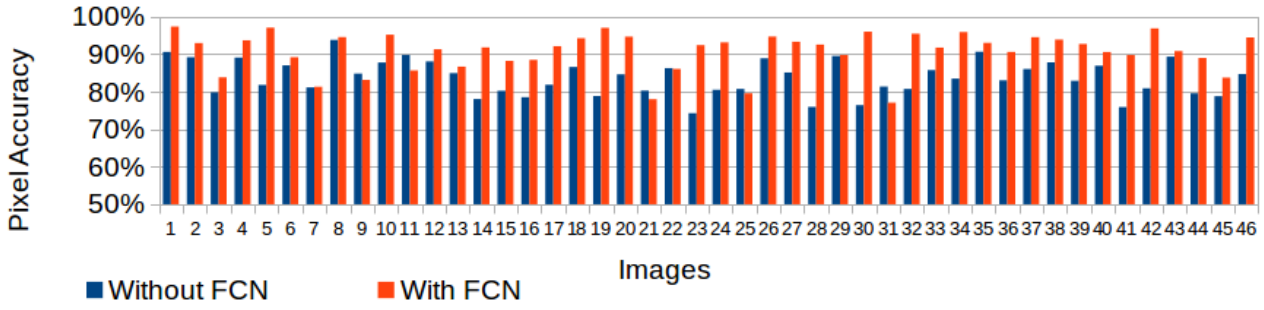


Fig. 8: Pixel Accuracy using the FCN from [13] to remove non-significant lines and without using it. We obtain a $90,5\%$ of mean accuracy value introducing it and $83,8\%$ without it.

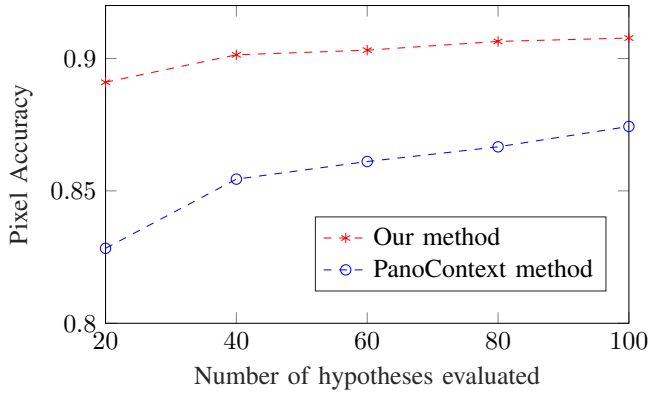


Fig. 9: Comparison of our method with the PanoContext [25]. We show the Pixel Accuracy against the number of hypotheses. Our method outperforms PanoContext and is able to provide much better results with fewer hypotheses.

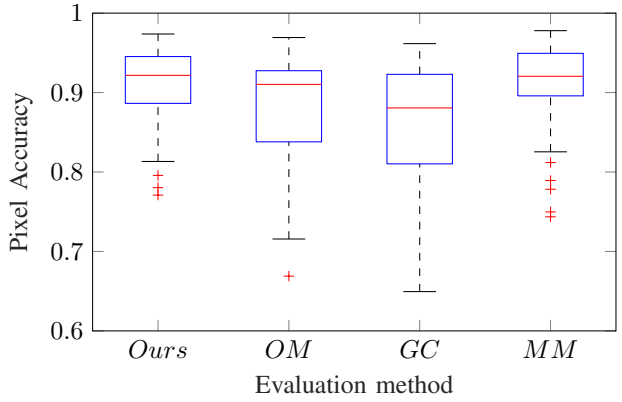


Fig. 10: Boxplot of the accuracy level for our hypotheses evaluation method and three alternatives from the state of the art. The box is limited by the 25th and 75th quartile with the median inside. The whiskers reach the most extreme points and the outliers are marked with a cross.

panoramic images applied to the problem of 3D room layout estimation with Manhattan World assumption (without four-walls simplifications), for which we propose a new flexible method that integrates old and new techniques fusing geometric reasoning in computer vision with two different deep neural networks [13], [4] adapted to the proposed image geometry.

Our experimental results imply that the proposed algorithm has a good performance in scene interpretation of full-view images and overcomes the state of the art.

ACKNOWLEDGMENT

This work was supported by Projects DPI2014-61792-EXP and DPI2015-65962-R (MINECO/FEDER, UE) and grant BES-2013-065834 (MINECO).

REFERENCES

[1] J. Bermudez-Cameo, G. Lopez-Nicolas, and J. J. Guerrero. Automatic line extraction in uncalibrated omnidirectional cameras with revolution symmetry. *International Journal of Computer Vision*, 114(1):16–37, 2015.

[2] S. Dasgupta, K. Fang, K. Chen, and S. Savarese. Delay: Robust spatial layout estimation for cluttered indoor scenes. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 616–624, 2016.

[3] E. Delage, H. Lee, and A. Y. Ng. A dynamic bayesian network model for autonomous 3D reconstruction from a single indoor image. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 2418–2428, 2006.

[4] D. Eigen and R. Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *IEEE International Conference on Computer Vision*, pages 2650–2658, 2015.

[5] K. He, H. Chang, and J. Sun. Rectangling panoramic images via warping. *Transactions on Graphics*, 32(4):79, 2013.

[6] V. Hedau, D. Hoiem, and D. Forsyth. Recovering the spatial layout of cluttered rooms. In *IEEE International Conference on Computer Vision*, pages 1849–1856, 2009.

[7] V. Hedau, D. Hoiem, and D. Forsyth. Thinking inside the box: Using appearance models and context based on room geometry. *European Conference on Computer Vision*, pages 224–237, 2010.

[8] H. Jia and S. Li. Estimating structure of indoor scene from a single full-view image. In *IEEE International Conference on Robotics and Automation*, pages 4851–4858, 2015.

[9] C. Lee, V. Badrinarayanan, T. Malisiewicz, and A. Rabinovich. RoomNet: End-to-end room layout estimation. In *IEEE International Conference on Computer Vision*, 2017.

[10] D. C. Lee, M. Hebert, and T. Kanade. Geometric reasoning for single image structure recovery. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2136–2143, 2009.

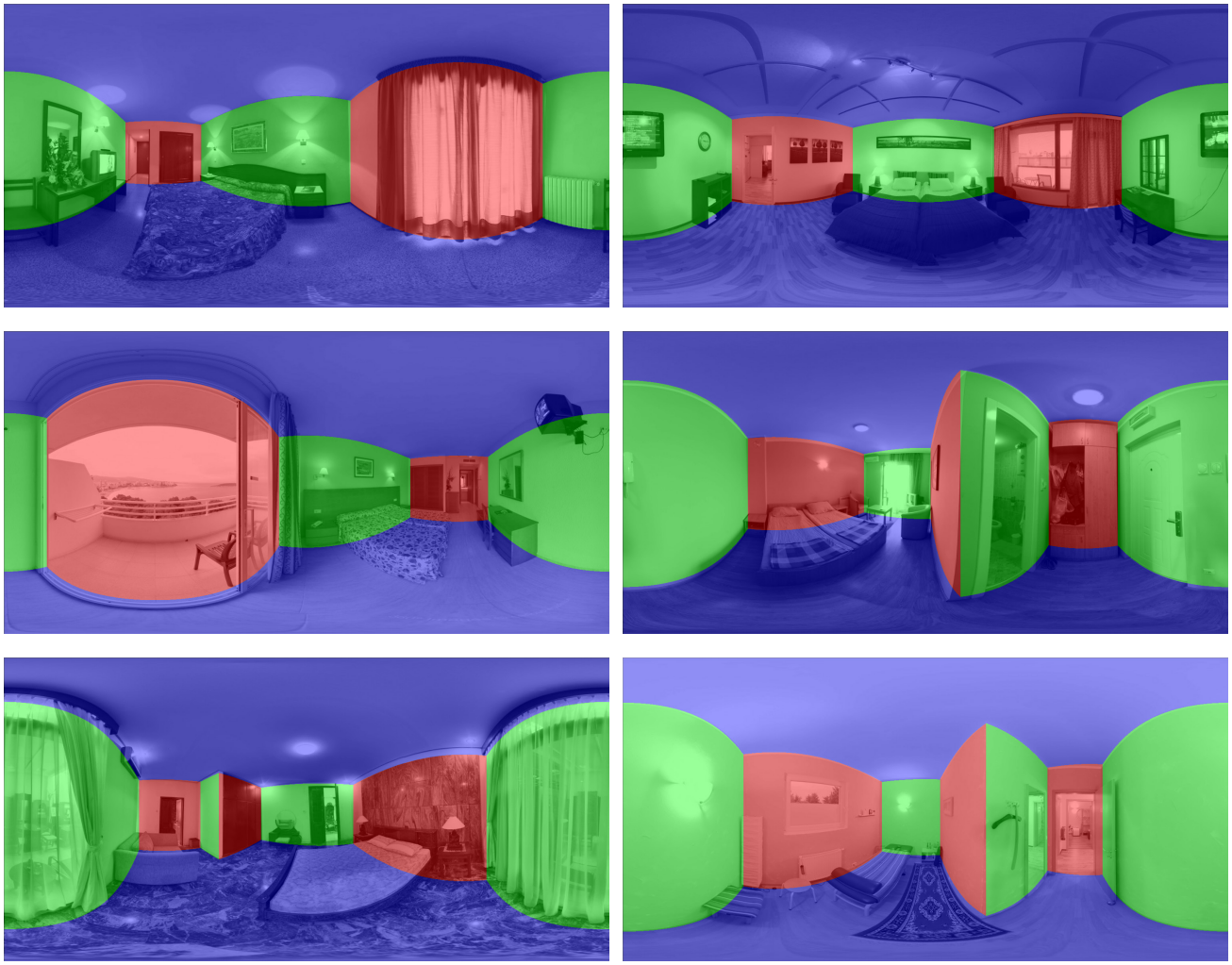


Fig. 11: Final room layout estimation examples with different complex wall distributions shown on the left with the structural edges on the input image with lines and corners and on the right with its normal map overlapped. (Best seen in color).

[11] G. Lopez-Nicolas, J. Omedes, and J.J. Guerrero. Spatial layout recovery from a single omnidirectional image and its matching-free sequential propagation. *Robotics and Autonomous Systems*, 62(9):1271–1281, 2014.

[12] R. Lukierski, S. Leutenegger, and A. J. Davison. Room layout estimation from rapid omnidirectional exploration. In *IEEE International Conference on Robotics and Automation*, pages 6315–6322, 2017.

[13] A. Mallya and S. Lazebnik. Learning informative edge maps for indoor scene layout prediction. In *IEEE International Conference on Computer Vision*, pages 936–944, 2015.

[14] S. H. Oh and S. Jung. Ransac-based orthogonal vanishing point estimation in the equirectangular images. *Journal of Korea Multimedia Society*, 15(12):1430–1441, 2012.

[15] S. H. Oh and S. K. Jung. A great circle arc detector in equirectangular images. In *International Conference on Computer Vision Theory and Applications*, pages 346–351, 2012.

[16] A. Perez-Yus, G. Lopez-Nicolas, and J.J. Guerrero. Peripheral expansion of depth information via layout estimation with fisheye camera. In *European Conference on Computer Vision*, pages 396–412, 2016.

[17] A. G. Schwing, S. Fidler, M. Pollefeys, and R. Urtasun. Box in the box: Joint 3D layout and object reasoning from single images. In *IEEE International Conference on Computer Vision*, pages 353–360, 2013.

[18] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus. Indoor segmentation and support inference from RGBD images. pages 746–760. Springer, 2012.

[19] R. G. Von Gioi, J. Jakubowicz, J.-M. Morel, and G. Randall. Lsd: A fast line segment detector with a false detection control. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(4):722–732, 2010.

[20] 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.

[21] J. Xu, B. Stenger, T. Kerola, and T. Tung. Pano2CAD: Room layout from a single panorama image. In *IEEE Winter Conference on Applications of Computer Vision*, pages 354–362, 2017.

[22] H. Yang and H. Zhang. Efficient 3D room shape recovery from a single panorama. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 5422–5430, 2016.

[23] S. Yang, D. Maturana, and S. Scherer. Real-time 3d scene layout from a single image using convolutional neural networks. In *IEEE International Conference on Robotics and Automation*, pages 2183–2189, 2016.

[24] W. Zhang, W. Zhang, K. Liu, and J. Gu. Learning to predict high-quality edge maps for room layout estimation. *Transactions on Multimedia*, 19(5):935–943, 2017.

[25] 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.