Universidad Zaragoza

Trabajo Fin de Grado

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

Autor

Juan Carlos Medina Franca

Directores

Jose Jesús Guerrero Campo

Clara Fernandez Labrador

ESCUELA DE INGENIERIA Y ARQUITECTURA

2018

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. Juan Carlos Medina Franca,

con nº de DNI 73159432 c 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) ~~Estimación del Layout 3D en interiores a partir de imágenes.~~, (Título del Trabajo) Estimación del Layout 3D en interiores a partir de imágenes.,

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

Zaragoza, 22-11-2018

Fdo:

Resumen

En este trabajo se ha desarrollado un método de identificación de los bordes estructurales de habitaciones a partir de una única imagen. Han sido muchos los trabajos que han tratado de resolver este problema a lo largo de la última década. Estos métodos generalmente se basan en la generación de diferentes hipótesis de modelos de *layouts* a partir de razonamientos puramente geométricos, o bien mas recientemente, utilizando técnicas de aprendizaje profundo (para, o bien apoyar las hipótesis basadas en la geometría o bien hacer hipótesis basadas solamente en el aprendizaje profundo). La principal limitación de realizar hipótesis que impliquen razonamientos geométricos es que en las imágenes con muchas oclusiones las direcciones principales pueden ser muy difíciles de detectar, mientras que confiar las hipótesis solamente a las técnicas de aprendizaje profundo (*Deep Learning*) no es del todo eficaz, ya que el uso de estas para este fin todavía esta en fase de desarrollo y no tienen la eficacia deseada. Este trabajo tiene la principal novedad de combinar dos tipos de hipótesis, una basada en razonamientos geométricos de visión por computador y otra a partir únicamente de técnicas '*Deep Learning*' siendo capaces de detectar la mejor solución en cada caso. En este trabajo se muestran resultados de reconstrucción de *layouts* con imágenes de la base de datos pública LSUN (Large-scale Scene Understanding Challenge) usada por otros trabajos del estado del arte. Con ellos demostramos la efectividad del método con respecto a trabajos existentes, situándonos en nuestros primeros experimentos a la cabeza del estado del arte.

Abstract

In this work we have developed a method of identifying the structural edges of rooms from a single image. There have been many works that have tried to solve this problem over the last decade. These methods are generally based on the generation of different hypotheses of layouts models from purely geometric reasoning, or more recently, using deep learning techniques (to either support hypotheses based on geometry or make hypotheses based only in deep learning). The main limitation of carrying out hypotheses involving geometric reasoning is that in the images with many occlusions the main directions can be very difficult to detect, while trusting the hypotheses only to the depth learning techniques is quite effective, since the use of these for this purpose are still under development and do not have the desired effectiveness. This work has the main novelty of combining hypothesis types, some based on geometric reasoning of computer vision and others only from Deep Learning techniques being able to detect the best solution in each case. In this paper we show results of reconstruction of layouts with images from the public database LSUN (Large-scale Scene Understanding Challenge) used by other works of the state of the art. With them we demonstrate the effectiveness of the method with respect to existing works, situating us in our first experiments at the head of the state of the art.

Índice general

Capítulo 1

Introducción

La visión artificial trata de producir el mismo efecto que nuestros ojos y cerebro hacen para que podamos comprender el mundo que nos rodea, es decir,trata de que los ordenadores puedan percibir y comprender una imagen o secuencia de imágenes. Esta disciplina cubre un amplio abanico de problemas y técnicas tales como el reconocimiento de patrones (*e.g.* reconocimiento facial Fig. 1.1).

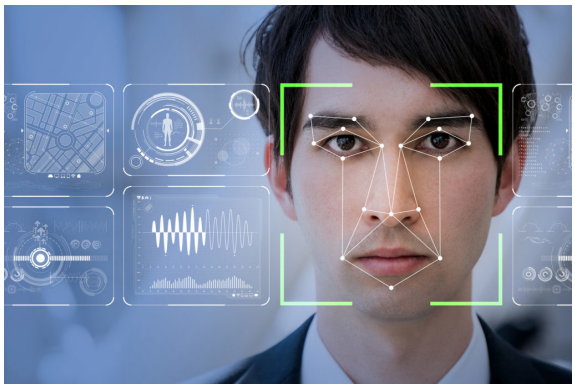


Figura 1.1: Detección Facial

Uno de los campos de investigación actuales fundamentales en el campo de la visión artificial es el de la reconstrucción 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. En concreto, nosotros nos vamos a aprovechar de las numerosas tecnologías con las que contamos hoy en día para la detección de los bordes estructurales de habitaciones de interior a partir de una única imagen convencional.

Para ilustrar el interés del problema que vamos a abordar, en la Fig. 1.2 se muestran dos aplicaciones que se están usando en la actualidad que están directamente relacionadas con nuestro trabajo.

En la Fig. 1.2a se muestra la aplicación *IKEA Place augmented reality*, una aplicación de IKEA que le permite al consumidor ver como quedaran los muebles en su propia casa u oficina antes de comprarlos. La Fig. 1.2b nos muestra el robot roomba que esta diseñado para aspirar todo el espacio disponible, en los rincones y a lo largo de las paredes. 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) IKEA Place augmentedreality app

(b) Robot Roomba

(c) Google Tango

Figura 1.2: Aplicaciones de la comprensión y reconstrucción de habitaciones de interior.

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 [1]). Un reto importante de este trabajo es el de integrar ese tipo de hipótesis y tratar de inferir el *layout* de escenas de interior aprovechando técnicas de aprendizaje profundo.

1.1 Estado del arte

En esta sección se van a describir algunos de los trabajos más relevantes en el campo, haciendo especial hinca pie en trabajos directamente relacionados con nuestro problema.

Probablemente los primeros intentos de abordar el desafío de realizar reconstrucciones de espacios interiores a partir de una única imagen fueron Delage *et al.* en [4] y Lee *et al.* [11]. El algoritmo del primero encuentra límites entre el suelo y las paredes de habitaciones usando un modelo de red Bayesiano. Por otro lado, el segundo 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.

Entrando mas en nuestro tipo de imagen y metodología, Hedau *et al.* [8] propone un *framework* clásico para la estimación de *layouts*. Determina los puntos de fuga y lanza 10 rayos desde el horizontal mas lejano y el vertical en ángulos uniformes. Las esquinas así formadas las une al punto de fuga restante. Así, hace hipótesis de *layout* y selecciona aquella que ha obtenido una mayor puntuación. La Fig. 1.3 muestra un ejemplo de como obtienen Hedau *et al.* sus *layouts*. Schwing *et al.* [15] utiliza una densidad de rayos mayor(50). Otros trabajos realizaron muestreo de rayos adaptativos teniendo en cuenta las características de bordes y esquinas [3] o las uniones de bordes [13]. Finalmente, Schwing et al. [16] evitó el muestreo del rayo frontal explícito a través de su procedimiento de inferencia exacto. Mallya *et*

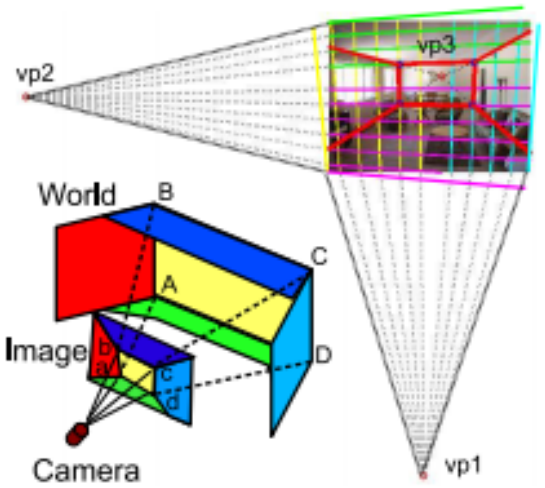


Figura 1.3: Generación de Layouts de [8]

al. [12] y Zhang *et al.* [18] buscan el sector con mas probabilidad de contener el borde estructural correcto en lugar de simplemente lanzar rayos de forma uniforme en toda la imagen, y una vez encontrado este, lanzan los rayos desde el punto de fuga horizontal mas lejano y punto de fuga vertical dentro de los sectores seleccionados. En 1.4 se muestra un ejemplo de la búsqueda de sectores de Mallya *et al.* Otras de las aportaciones mas recientes son la de Lee *et al.* [10] quien adopta una estimación mas directa del problema con una solución basada en encontrar todas las esquinas de la habitación con la ayuda de una red *end-to-end* entrenada para ello y la de Zhao *et al.* [20] que basa su método en la transferencia semántica (*semantic transfer*) y lo que el llama *physics inspired optimization* , esto es, ideas para optimizar el problema basadas en la física.

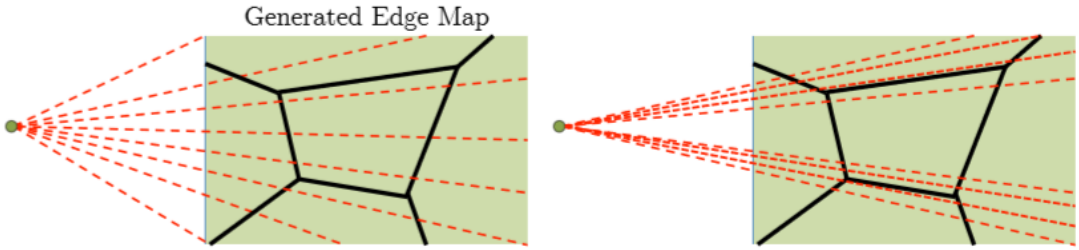


Figura 1.4: Ejemplo de búsqueda iterativa de sectores realizada por [12]

1.2 Objetivos

El principal objetivo de este trabajo es la realización de un algoritmo que a partir de una única imagen RGB cualquiera de interior, independientemente de las oclusiones que esta tenga, obtenga los bordes estructurales de la habitación.

Hay que destacar que inferir la estructura de una habitación si solo poseemos una

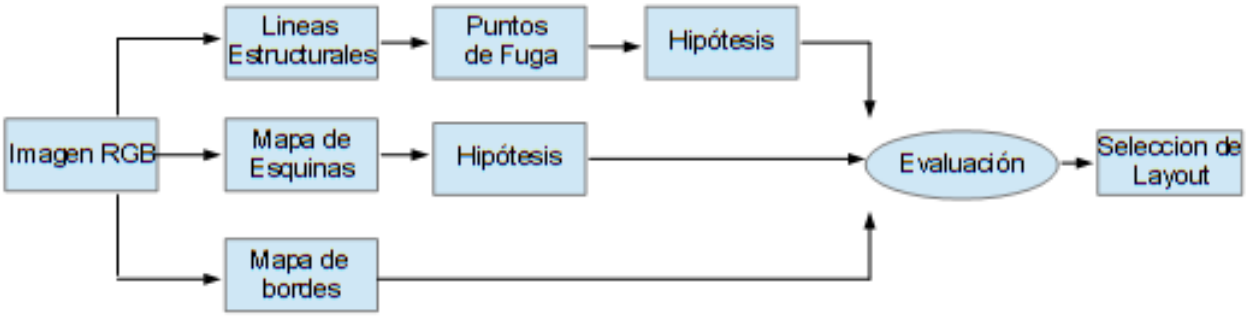


Figura 1.5: Descripción general de nuestro Algoritmo

vista parcial de ella es un gran reto puesto que tenemos una gran falta de contexto en comparación a otro tipo de imágenes (por ejemplo las imágenes panorámicas) y sólo tenemos como información una imagen RGB (a diferencia de otros métodos que hacen uso de información de profundidad o de múltiples vistas).

Por lo general, en la literatura, hay algunos supuestos que son ampliamente utilizados para resolver este tipo de tareas y en los que nosotros también nos hemos basado. Estas consisten en suponer que la habitación es una caja 3D de cuatro paredes y que sus superficies están orientadas de acuerdo con tres direcciones ortogonales principales (conocida como suposición de *Manhattan World* [1]).

Para tratar de resolver el problema, hemos propuesto un algoritmo que genera dos tipos diferentes de hipótesis, una basada en razonamientos geométricos (apoyándonos en técnicas *deep learning*) y otra basada únicamente en el aprendizaje profundo, lo que nos permite obtener un resultado mucho mejor que haciendo solo un tipo de hipótesis, como se había hecho hasta las fecha. Las dos hipótesis que obtenemos las evaluamos con mapas procedentes de la Red Neuronal de [5].

Una visión general de nuestro método se muestra en la Fig. 1.5: Como se puede observar en la imagen, una de las hipótesis la hacemos mediante rayos desde los puntos de fuga, y otra mediante rayos desde las esquinas que ha detectado la red neuronal. Para hacer ambos tipos de hipótesis nos apoyamos en los mapas de bordes, procedentes también de la misma red neuronal [5]. Al final, volvemos a usar los mapas de bordes, para determinar cual de las dos hipótesis se adapta mejor a la realidad.

Capítulo 2

Razonamiento Geométrico basado en Visión Artificial

En este capítulo presentamos las principales tareas llevadas a cabo para la obtención y clasificación de las líneas, una de las piezas básicas de información en nuestro método. Estas tareas se basan en técnicas de visión artificial y han sido abordadas por múltiples trabajos a lo largo de los años.*e.g* [15, 3] .

2.1 Extracción de Líneas.

Al igual que muchos de los trabajos del estado del arte, vamos a utilizar el algoritmo LSD (*line segment detector*) [17] para llevar a cabo esta tarea, que es fundamental hacer de la mejor manera posible para que nuestras hipótesis basadas en la geometría sean lo mejores posibles.

El LSD está dirigido a detectar contornos rectos en las imágenes, lo que llamamos segmentos de línea. Los contornos son zonas de la imagen donde el nivel de gris está cambiando lo suficientemente rápido de oscuro a claro o lo contrario. Por lo tanto, el degradado y las líneas de nivel de la imagen son conceptos clave y se ilustran en la Fig. 2.1.

En la Fig. 2.2 podemos ver la estructura de este algoritmo en pseudocódigo. Para entrar en profundidad a entenderlo se recomienda leer [17].

Una vez ejecutado, el algoritmo LSD nos devuelve un conjunto de líneas que están etiquetadas según una dirección determinada. Cada una de las líneas viene definida por los puntos inicial y final de cada línea. Lo primero que hacemos con estas

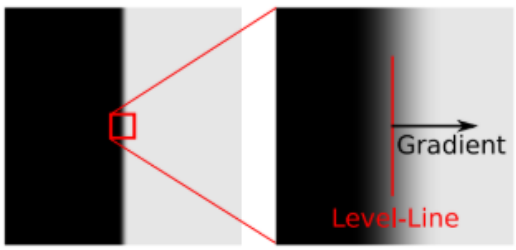


Figura 2.1: Detección de líneas con el algoritmo LSD basado en cambios de gradientes.

Algorithm 1: LSD: Line Segment Detector

input: An image I.
output: A list *out* of rectangles.

$I_S \leftarrow \text{ScaleImage}(I, S, \sigma = \frac{\Sigma}{S})$
$(\text{LLA}, |\nabla I_S|, \text{OrderedListPixels}) \leftarrow \text{Gradient}(I_S)$
$\text{Status} \leftarrow \begin{cases} \text{USED}, & \text{pixels with } |\nabla I_S| \leq \rho \\ \text{NOT USED}, & \text{otherwise} \end{cases}$
foreach *pixel* $P \in$ OrderedListPixels **do**
  if $\text{Status}(P) = \text{NOT USED}$ **then**
    $region \leftarrow \text{RegionGrow}(P, \tau)$
    $rect \leftarrow \text{Rectangle}(region)$
    while $\text{AlignedPixelDensity}(rect, \tau) < D$ **do**
      $region \leftarrow \text{CutRegion}(region)$
      $rect \leftarrow \text{Rectangle}(region)$
    end
    $rect \leftarrow \text{ImproveRectangle}(rect)$
    $nfa \leftarrow \text{NFA}(rect)$
    if $nfa \leq \varepsilon$ **then**
      Add $rect \rightarrow out$
    end
  end
end

Figura 2.2: Estructura del algoritmo LSD.

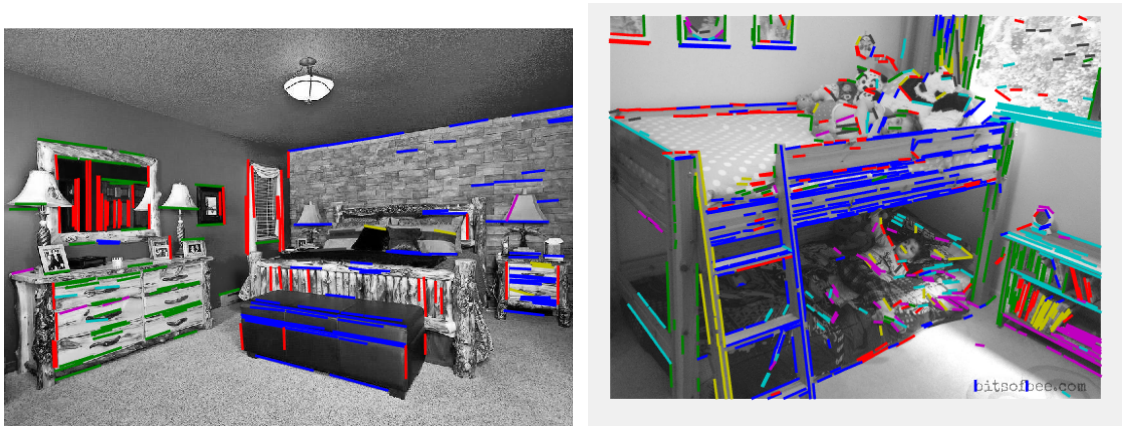
líneas es eliminar las que son mas pequeñas que un determinado treshold, obtenido experimentalmente y dependiente del tamaño de la imagen, y las líneas con una etiqueta mayor que un determinado valor (ya que estos grupos de líneas casi siempre proceden del desorden de la imagen y no de una dirección principal). Después, agrupamos las líneas en conjuntos según la etiqueta que les ha sido asignada.

Ahora bien, nosotros trabajamos bajo la hipótesis de Mundo Manhattan [1], por tanto suponemos que en nuestra imagen existen 3 direcciones principales. Lo ideal seria que los 3 conjuntos con mayor numero de lineas (lineas clasificadas según las 3 primeras etiquetas) se correspondieran siempre con estos conjuntos,como ocurre en la Fig.2.3a pero por desgracia, esto no siempre es así como se puede ver en la Fig. 2.3b,lo cual indica que vamos a necesitar hacer algo que nos haga quedarnos solo con las pertenecientes a las 3 direcciones principales.

2.2 Puntos de Fuga

El punto de fuga es aquel donde convergen todas las rectas paralelas proyectadas en una dirección determinada (Fig.2.4) Teniendo en cuenta la definición de punto de fuga, si tenemos agrupadas las lineas según su etiqueta tenemos grupos de líneas paralelas pertenecientes a una misma dirección, por tanto la intersección de estas, nos dará el punto de fuga asociado a las lineas con esa etiqueta.

Ahora, nos quedaría por tanto, saber quedarnos con los puntos de fuga que se han formado por la intersección de las rectas pertenecientes a cada una de las direcciones principales y saber desechar el resto. Para ello hemos aplicado los criterios que se explican en el siguiente apartado.



(a) (b)

Figura 2.3: Ejecución del algoritmo LSD. (a) Caso Ideal: Las 3 primeras etiquetas coinciden con las 3 direcciones principales (lineas azules,verdes y rojas) (b) Principal problema del LSD:Las 3 primeras etiquetas NO coinciden con las 3 direcciones principales.

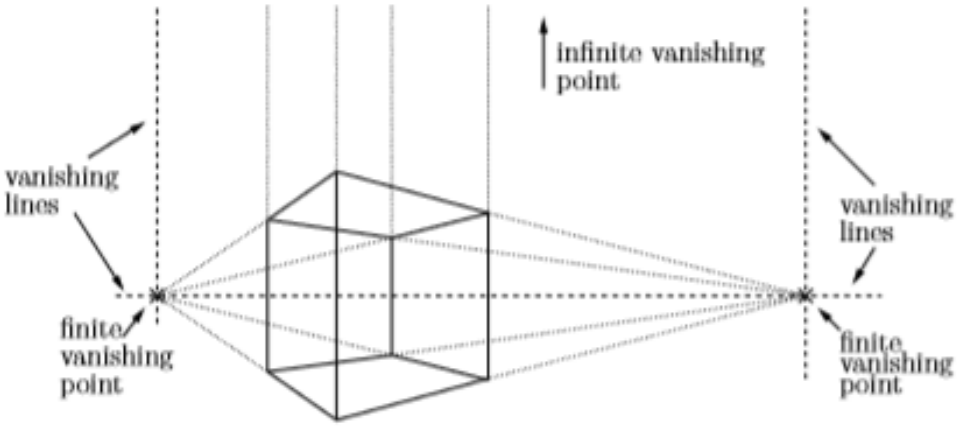


Figura 2.4: Puntos de Fuga [14].

2.2.1 Criterios de Ortogonalidad

Para conseguir nuestro objetivo, que no es otro que el de detectar los 3 puntos de fuga pertenecientes a las direcciones principales hemos aplicado los criterios de ortogonalidad propuestos por [14] para detectar los conjuntos de puntos de fuga válidos, dichos criterios se mencionan a continuación:

1. Tres puntos de fuga finitos (p1,p2 y p3).

   El criterio de ortogonalidad en este caso se define por que cada uno de los lados del triángulo formado por p1,p2 y p3 sea menos de 90º.

2. Dos puntos de fuga finitos (p1 y p2) y uno en el infinito (p3)

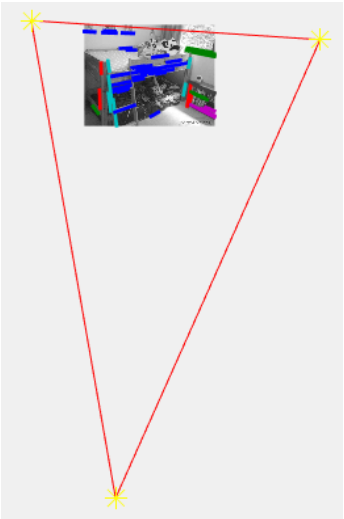
   En este caso el criterio de ortogonalidad se cumple cuando la dirección de p3 forma 90º con la línea que une p1 y p2.

3. Un punto de fuga finito (p1) y dos en el infinito (p2 y p3)

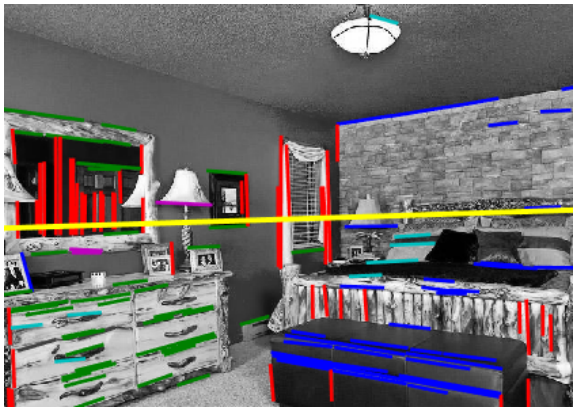
   El criterio de ortogonalidad en este caso se cumple si las direcciones de p2 y p3 son ortogonales.

Así pues, nosotros obtendremos una lista de los conjuntos de puntos de fuga que cumplen los criterios de ortogonalidad, y suponiendo que los conjuntos de esta lista sean válidos, sacaremos hipótesis de *layout* obtenidas a partir de estos conjuntos.

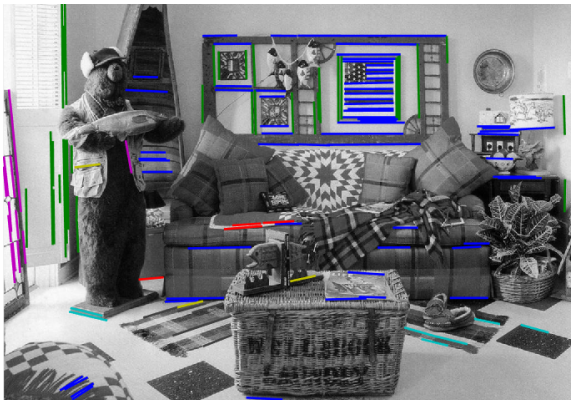
En la figura 2.5 se muestran ejemplos de conjuntos de puntos de fuga detectados por nosotros y se ilustra como se cumplen los criterios de ortogonalidad mencionados en cada uno de los casos. Se puede observar que los ángulos del triángulo que forman los 3 puntos de fuga de la Fig. 2.5a son menores de 90 grados, también se observa como la recta que une los 2 puntos de fuga finitos en la Fig. 2.5b (linea pintada en amarillo) es perpendicular a la dirección del infinito (lineas rojas) y por último en la Fig.2.5c como, en el caso de que 2 puntos de fuga estén en el infinito, la dirección de estos infinitos es perpendicular (lineas verdes y azules).



(a)



(b)



(c)

Figura 2.5: Ejemplo de los criterios de ortogonalidad de [14]. Fig (a): Tres puntos de fuga finitos. El triángulo formado por los 3 puntos de fuga tiene todos sus ángulos menores de 90º. Fig(b): Dos puntos de fuga finitos y uno infinito. La línea que une los 2 puntos de fuga finitos(pintada en amarillo) es perpendicular a la dirección del infinito (líneas rojas). Fig(c): Dos puntos de fuga infinitos. Las direcciones de los infinitos son perpendiculares (lineas verdes y azules).

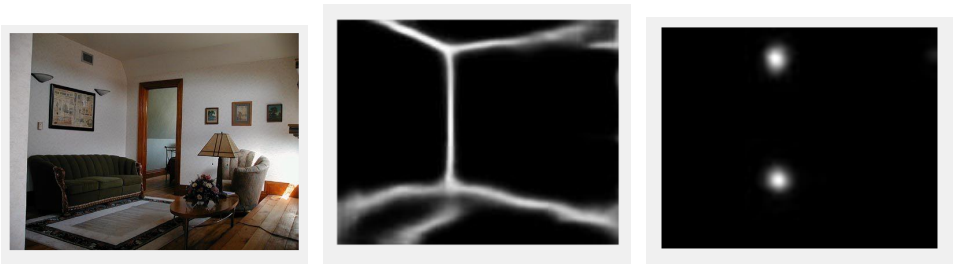
Capítulo 3

Técnicas de *Deep Learning*

Las Redes Neuronales son un campo muy importante dentro de la Inteligencia Artificial. Inspirándose en el comportamiento conocido del cerebro humano (principalmente el referido a las neuronas y sus conexiones), trata de crear modelos artificiales que solucionen problemas difíciles de resolver mediante técnicas algorítmicas convencionales.

En particular, 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*.

Nosotros hemos escogido la red neuronal de [5] puesto que nos devuelve mapas de esquinas y mapas de bordes que se ajustan bastante bien a la realidad. Un ejemplo de los mapas de bordes y mapas de esquinas que obtenemos con esta red se ilustran en la Fig. 3.1b y en la Fig. 3.1c respectivamente.



(a) Imagen (b) *Edgemap* (c) *Cornermap*

Figura 3.1: Ejemplo de funcionamiento de la Red neuronal de [5]

Hasta ahora, algunos trabajos proponen combinar mapas de bordes como apoyo a las hipótesis realizas por métodos geométricos [12, 20, 6]. Paralelamente, en [9] proponen una red *end-to-end* que predice las esquinas de las habitaciones y directamente extrae el *layout*. Este trabajo predice un mapa de esquina por cada una de las posibles entre unos tipos predefinidos, haciendo un total de 48 predicciones, incluyendo como esquinas las intersecciones con el borde de la imagen. La red que utilizamos nosotros [5] sin embargo, predice un único mapa que incluye solo las esquinas reales y no tenemos en cuenta si la imagen es de un tipo o de otro, reduciendo así el número de asunciones.

Capítulo 4

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 dos tipos distintos de hipótesis de diseño:

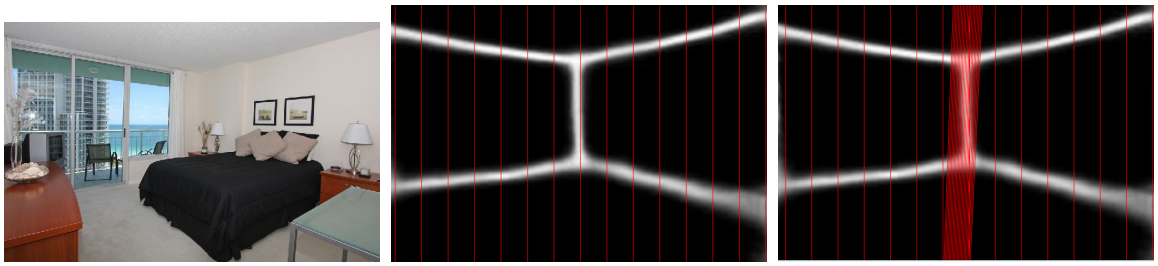
1. Una basada en razonamientos geométricos de visión por computador, en la que lanzaremos rayos desde los puntos de fuga vertical y horizontal mas lejano para, con la ayuda de técnicas de aprendizaje profundo determinar primero las secciones que tienen mas probabilidad de contener bordes estructurales y una vez detectadas estas secciones, volver a lanzar rayos desde los puntos de fuga para, nuevamente con la ayuda de la red neuronal, elegir los rayos que mas probabilidad tienen de ser los bordes estructurales de la imagen.

2. Otra a partir únicamente de técnicas '*Deep Learning*', donde con una red neuronal que detecta zonas donde hay probabilidad de que se encuentren las verdaderas esquinas estructurales, primero seleccionamos cuales son las esquinas con mayor probabilidad, y desde estas lanzamos una serie de rayos, entre los cuales, somos capaces de seleccionar los que mas probabilidad tienen de ser los verdaderos bordes, al igual que anteriormente.

Finalmente evaluamos con los mapas de bordes cual de las hipótesis es mejor, si la generada mediante geometría o la hipótesis generada a partir de las esquinas detectadas con el mapa de esquinas. A continuación se procede a explicar por separado cada uno de los dos métodos de diseño seguidos y la selección de la mejor hipótesis de *layout*.

4.1 Generación de hipótesis basadas en razonamientos geométricos

Nuestra hipótesis basada en geometría comienza con el primer conjunto de 3 puntos de fuga valido que detectamos (cabe destacar que podríamos hacer mas de una hipótesis de este tipo ya que generalmente tenemos mas de un conjunto de puntos



(a) Imagen RGB original (b) Búsqueda por sectores(vertical) (c) Sectores verticales mas probables

Figura 4.1: División de la imagen en sectores y búsqueda del mas probable.

de fuga valido, pero desechamos esta posibilidad debido a que apenas mejora los resultados y nos genera un tiempo de computo mucho mayor, ya que hacemos muchas mas hipótesis), primeramente detectamos cual es el vertical y el horizontal mas lejano. Una vez tenemos estos 2 puntos, comenzamos la búsqueda de las secciones con mas probabilidad de contener bordes estructurales.

4.1.1 Búsqueda de los sectores mas probables

El primer paso es dividir la imagen en 15 secciones desde el punto de fuga vertical como se muestra en la Fig. 4.1b para así ver cual o cuales son las secciones con mas probabilidad de encontrar bordes estructurales verticales. Una vez hemos detectado cual es la sección vertical mas probable (Fig.4.1c) pasamos a la horizontal, con la que procedemos exactamente de la misma manera, aunque hay una razón por la que buscamos primero la sección vertical, la cual se entenderá a continuación. A partir de ahora se va a explicar el proceso mas detalladamente desde el punto de fuga horizontal mas lejano, pero desde el vertical se ha procedido exactamente de la misma manera. En la Fig. 4.4a se puede ver la imagen dividida en 15 sectores desde el punto de fuga horizontal mas lejano. Ahora, el problema a resolver es ver en cual o cuales de los sectores tenemos que buscar las lineas que son parte de la estructura del *layout* (4.4b). Una primera idea podría ser, simplemente, el sumar el nivel de gris de los píxeles del mapa de bordes dentro de cada sector y quedarnos con los máximos de la parte superior e inferior, siempre que este valor sea de mayor que un determinado *threshold* horizontal, ya que, como se ha explicado anteriormente, los mapas de bordes que usamos no son perfectos y en algunas imágenes pueden dar lugar a confusión y si no ponemos un *theshold* dar lugar a la detección de falsos bordes.

Esto es lo que propone [12], aunque su método puede dar lugar a error como se indica en la Fig. 4.2.

Tras analizar de donde procede este tipo de error, hemos propuesto una solución a este diferente de la que proponen Zhang et al.: eliminar del mapa del bordes los niveles de gris de la parte que nos 'molesta', es decir, si el punto de fuga horizontal mas lejano es el derecho, eliminamos el nivel de gris desde la coordenada x de la sección vertical seleccionada (esta es la razón por la que hacemos la sección vertical primero) hasta la anchura total de la imagen, así evitamos los píxeles que inducen a error en el método de Mallya *et al.*. Esta idea se basa en que la linea vertical

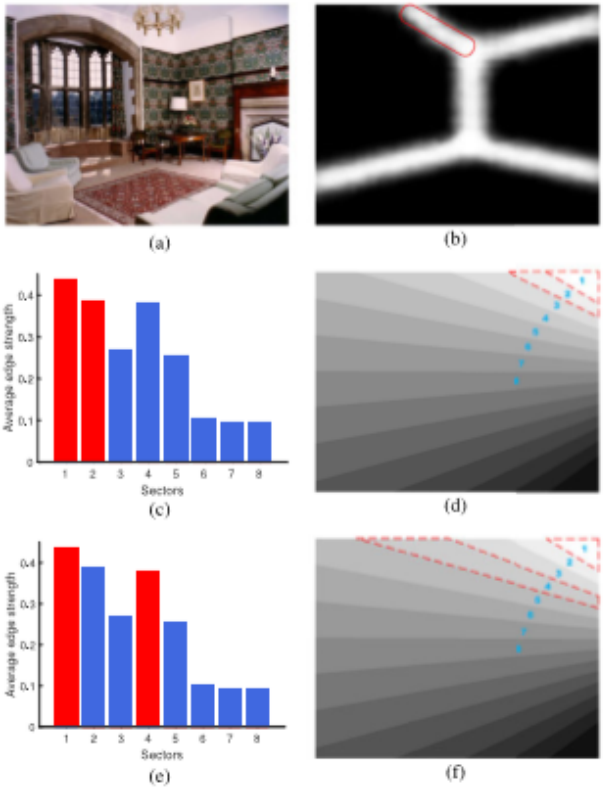
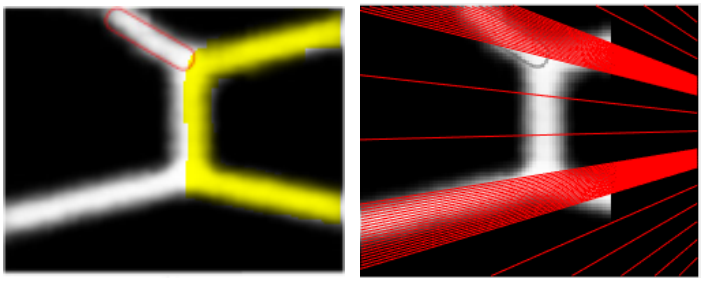


Figura 4.2: (a) es la imagen original mientras que (b) es la es el mapa de bordes o *edge map*. Se han numerado los sectores de la parte superior del punto de fuga de 1 a 8 en (c) donde se han puesto en rojo los sectores seleccionados de acuerdo a su nivel de gris en (b).Como se ve , no se incluye el sector 4 (que es el bueno) Las figuras (e) y (f) muestran los resultados del método de Zhang , donde si que selecciona el sector numero 4. Imagen tomada del trabajo de [18].



(a) Píxeles que inducen a error

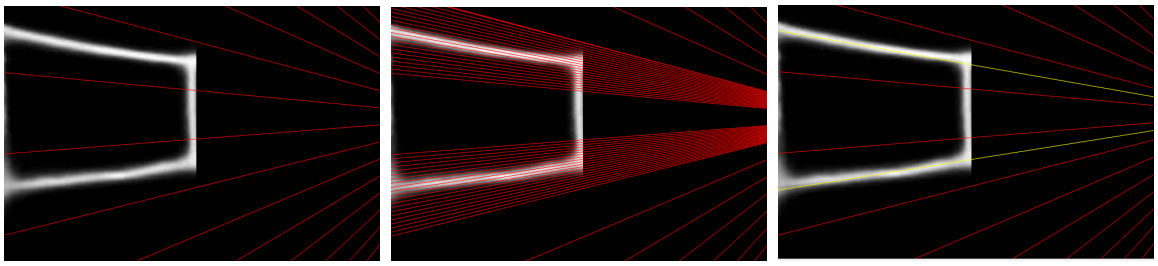
(b) Corrección del error

Figura 4.3: Se muestra nuestra corrección del ejemplo que muestra Zhang *et al.* en su trabajo. Podemos ver que nuestro método seleccionando solo una sección de la zona superior ya nos selecciona la sección buena mientras que si se observa con detenimiento la figura, se observa que el método de corrección del error propuesto por Zhang *et al.* necesita seleccionar dos secciones dentro de la parte superior ya que si seleccionara solo la que mas peso tiene se quedaría con la primera.

detectada es aquella que define la intersección entre dos paredes y, por tanto, el cambio de dirección de esta.

En la Fig. 4.3a se ilustra esto, resaltando en amarillo la zona que induce a error. En la Fig. 4.3b se muestra como, nuestro algoritmo corrige el error de Mallya et al. de una forma mas efectiva que el trabajo de Zhang *et al.* pues nos quedamos con una única sección de la parte superior y una de la parte inferior en vez de dos (ya que no tenemos esa necesidad para encontrar cual es la buena). Zhang et al. necesita escoger dos ya que, si se quedara con una, no conseguiría seleccionar la correcta como se muestra en la Figs.4.2 (e). Esto supone un ahorro en el numero de hipótesis erróneas que hacemos muy grande (en el caso de que la imagen tenga un borde estructural vertical y dos horizontales, Zhang et al. necesitaría hacer 4 hipótesis con cada conjunto bueno y en una imagen con dos bordes verticales y dos horizontales necesitaría hacer 8).

4.1.2 Selección de los Bordes estructurales de la habitación



(a) Búsqueda por sectores(horizontal) (b) Sectores horizontales mas probables (c) Selección del mejor borde estructural

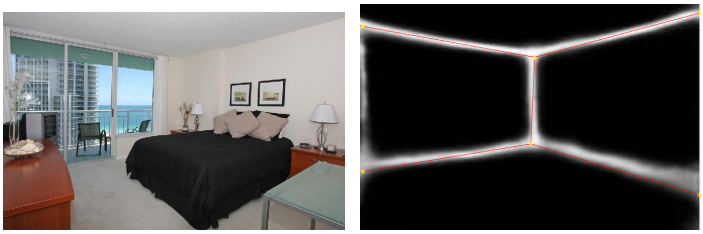
Figura 4.4: División de la imagen en sectores,búsqueda del mas probable y dentro de este el mejor borde estructural

Una vez hemos determinado cuales son las secciones mas probables tanto verticales como horizontales estamos a un paso de poder generar nuestra hipótesis basada en geometría. Primeramente, dentro de cada una de las sección mas probables lanzamos 20 rayos y evaluamos cual de ellos obtiene una mayor puntuación en cada una de la secciones (Fig. 4.4c).

Una vez tenemos las lineas horizontales y verticales que mejor se adaptan a la estructura de la habitación según el mapa de bordes, estamos en disposición de mostrar nuestras primeras hipótesis realizadas en imágenes, basadas en razonamientos geométricos apoyados en mapas de bordes como se ha ido explicando a lo largo de la sección. En la Fig.4.5 se muestran algunos resultados, obtenidos con este tipo de hipótesis.

4.2 Generación de hipótesis basadas únicamente en técnicas de aprendizaje profundo

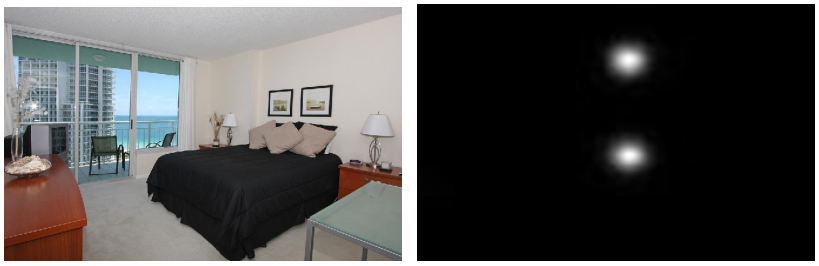
Para la realización de este tipo de hipótesis nos vamos a basar únicamente en la información proporcionada por los mapas de esquinas y los mapas de bordes (que



(a) ImagenRGB (b) Hipótesis

Figura 4.5: Obtención de *layouts*

los usamos con la misma finalidad que para el otro tipo de hipótesis) generados por la red neuronal de Fernández *et al.* [5]. La figura 4.6 ilustra un ejemplo de mapa de esquinas que devuelve la red. En los mapas de esquinas (*corner maps*) las



(a) ImagenRGB (b) Mapa de esquinas.

Figura 4.6: Se muestra un ejemplo de imagen RGB con su mapa de esquinas correspondiente.

esquinas están representadas por gausianas, cuyo centro (valor máximo) es el que seleccionamos como coordenada de nuestra esquina estructural. Además nos interesa establecer un cierto *treshold* , ya que si no, al igual que sucedía con los mapas de bordes, en algunas imágenes podríamos detectar esquinas que realmente no existen como se muestra en la Fig.4.7

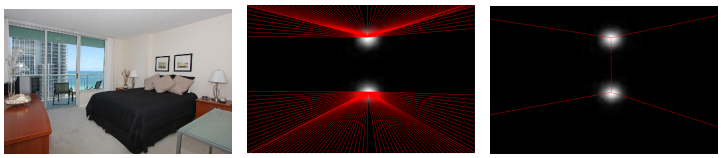
(a) ImagenRGB (b) Necesidad de un *treshold*

Figura 4.7: Se muestra un ejemplo de imagen que solo tiene una esquina estructural, y que, de no poner un *treshold* detectaríamos 2.

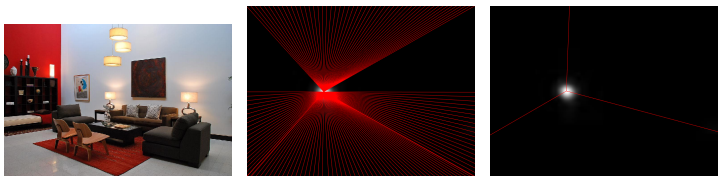
Una vez hemos detectado cuales son las esquinas estructurales de la imagen, nos queda tratar de predecir cuales van a ser los bordes estructurales de nuestra

habitación. Para ello, nuestro algoritmo detecta el tipo de imagen que es en función de las esquinas estructurales detectadas y lanza rayos desde las esquinas a las zonas donde deberían estar los bordes de acuerdo al tipo de imagen (Fig.4.9) que hemos detectado. Estos, nuevamente, los evaluamos con el mapa de bordes. En las Fig.4.8b y 4.8e se muestran dos ejemplo de como lanzamos los rayos, para una imagen que predecimos que es de tipo 5, y en el otro en una en la que predecimos que es de tipo 4 respectivamente.

Por último, ya nos quedar determinar en cada zona, cual es rayo que mas probabilidad tiene de ser un borde estructural. Como hemos dicho antes, la tarea de evaluar que rayos son mejores, la hacemos con la ayuda del mapa de bordes. En las Figs. 4.8c y 4.8f se muestran ejemplos de como seleccionamos los rayos que mejor se adaptan a los bordes estructurales del *layout*.



(a) Imagen RGB (b) Imagen de tipo 5 (c) Hipótesis1



(d) Imagen RGB (e) Imagen de tipo 3 (f) Hipótesis2

Figura 4.8: Muestras del lanzamiento de rayos en función del tipo de imagen detectada y selección de estos.

4.3 Selección de la mejor hipótesis

Como se ha mencionado a lo largo del trabajo, nuestro algoritmo genera 2 tipos de hipótesis: una basada en la geometría y otra basada en el aprendizaje profundo. Para decidirnos por una u otra calculamos la probabilidad media en el mapa de bordes de las esquinas detectadas por uno y otro método y nos quedamos con el que nos proporciona un mayor valor. Véase que esto lo hacemos así debido a que, en algunos casos, el tipo de imagen detectado con un método y con otro, es diferente (Fig.4.10).

El poder elegir entre hipótesis generadas por distintos métodos es la principal novedad de nuestro trabajo, y la principal razón de que nuestro trabajo se situé a la cabeza del estado del arte, ya que como se ilustra en la (Fig.4.10) hay imágenes que mediante razonamientos geométricos predicaríamos erróneamente pero que con las técnicas *deep learning* predecimos bien y viceversa.

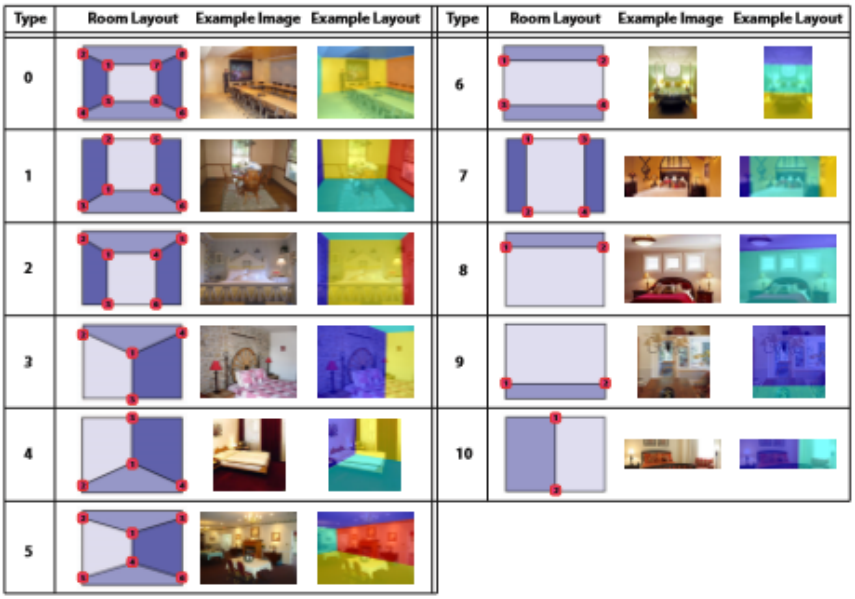


Figura 4.9: Tabla con los posibles tipos de imagen [19].

(a) Hipótesis realizada con técnicas *deep learning* (Tipo 4).

(b) Hipótesis realizada con razonamientos geométricos (Tipo 5).

Figura 4.10: En esta imagen se observa, por un lado como podemos predecir distintos tipos de *layout* según hayamos realizado la hipótesis con un método u otro. Además, se puede ver como en este caso, gracias a la hipótesis que hacemos basada en *deep learning*, hacemos una predicción muy buena, cosa que no es así (para esta imagen) en la hipótesis hecha usando geometría.

Capítulo 5

Experimentos

En esta sección se van a mostrar los experimentos realizados de nuestro método. Por un lado, vamos a hablar del *dataset* utilizado. A continuación se mostrara una evaluación de nuestros resultados tanto de forma cuantitativa comparándonos con otros trabajos del estado del arte como de forma cualitativa mostrando algunos ejemplos de diseño obtenidos con nuestro algoritmo.

5.1 *Dataset* utilizado

Para la evaluación experimental hemos recogido un subconjunto aleatorio de 41 imágenes procedentes de la base de datos pública LSUN (*Large-scale Scene Understanding Challenge*). La misma base de datos ha sido utilizada por la gran mayoría de trabajos con nuestro mismo objetivo [12, 18, 9, 8, 20, 7, 21, 2].

5.2 Evaluación de layouts

Para obtener resultados de evaluación hemos usado la *toolkik* de la pagina de LSUN [19]. Hemos evaluado los nuestros resultados de las dos formas que se propone en el *Large-scale Scene Understanding Challenge*, calculando por un lado el *corner error* (distancia promedio entre pares de esquinas de la habitación correspondientes, normalizada por la diagonal del borde de la imagen) y por otro el *pixelwise error* (porcentaje de píxeles que están etiquetados de la misma manera entre la predicción y el *ground truth*). Además, también hemos realizado una evaluación del tiempo de ejecución de nuestro algoritmo.

5.2.1 Resultados cuantitativos:

Esta subsección tiene como principal objetivo el compararnos con otros trabajos del estado del arte con el mismo objetivo y la misma base de datos. Además seremos los primeros en mostrar una evaluación de tiempos de ejecución de nuestro algoritmo, algo que, aunque muchas veces se pasa por alto, resulta imprescindible en muchas aplicaciones.

Tabla 5.1: Performance benchmarking for the LSUN dataset

Method	**Corner Error (%)**	**Pixel Error(%)**
Hedau *et al.* (2009) [8]	15.48	24.23
Mallya *et al.* (2015) [12]	11.02	16.71
Zhang *et al.* (2017) [18]	8.70	12.49
Dasgupta *et al.* (2016) [2]	8.20	10.63
LayoutNet (2018) [21]	7.63	11.96
Ren *et al.* (2016) [7]	7.95	9.31
RoomNet (2017) [9]	6.30	9.86
Zhao *et al.* (2017) [20]	3.84	**5.29**
Ours	**2.64**	5.39

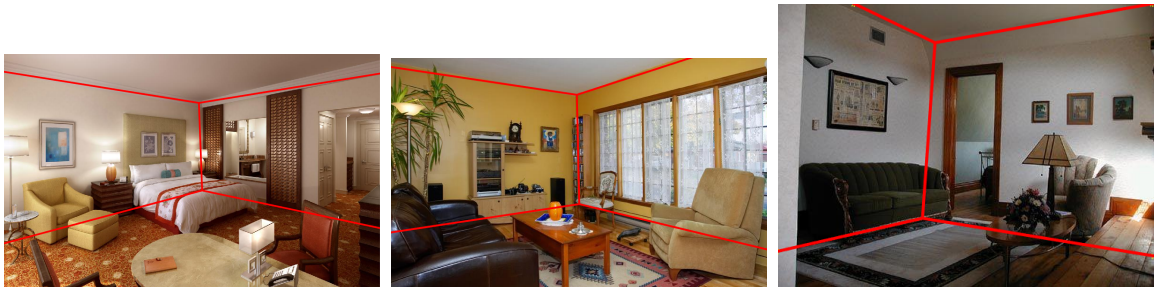
Como se indica en la Tabla 5.1 nuestro trabajo se sitúa a la cabeza del estado del arte a falta de evaluarlo con todas las imágenes del *dataset*. La evaluación la hemos realizado con la mediana de los valores de *pixelerror* y *cornererror* obtenidos de nuestras 41 imágenes aleatorias. Como ya hemos adelantado anteriormente, la eficacia de nuestro método por encima del resto radica en que no confiamos nuestras hipótesis solamente a los razonamientos geométricos como hacen [18, 18, 8, 15, 13, 16] o solamente a las técnicas de aprendizaje profundo como recientemente propone [9] con su red *end to end*, si no que nuestro algoritmo es capaz de, en función de las esquinas obtenidas por un método u otro, elegir la hipótesis que mejor se adapta al *layout* de la habitación.

Por otro lado, hemos realizado una evaluación de tiempos de nuestro algoritmo, obteniendo un valor de la mediana de todos ellos de 40.46 segundos con una desviación típica de 44,7 segundos, lo que indica una gran variabilidad en los tiempo de ejecución. Esta variabilidad se debe a que el numero de lineas que detecta el LSD es muy variable entre unas imágenes y otras, y por ello tarda mas o menos tiempo en ejecutarse.

5.2.2 Resultados cualitativos

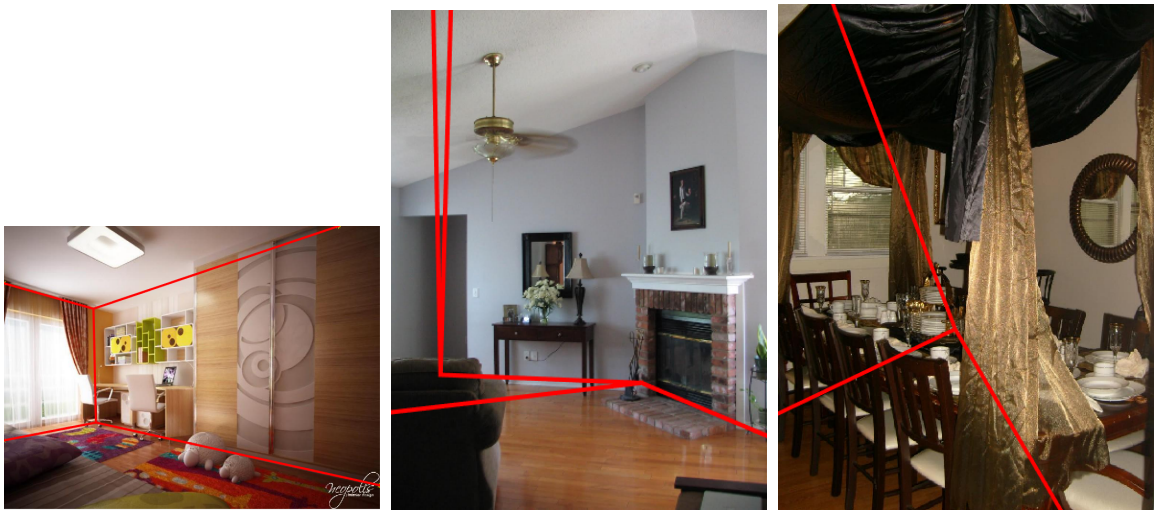
En esta subsección se muestra una colección de ejemplos de hipótesis finales de la estimación del diseño de habitaciones.

En la Fig.5.1 se muestran resultados para imágenes de dificultad promedio. Como se ve, nuestro algoritmo detecta los bordes estructurales de las habitaciones con una gran precisión en general. No obstante, en la base de datos usada hay algunas imágenes que no cumplen las hipótesis que hemos supuesto como se ilustra en las Fig.5.1g (la imagen mostrada no cumple la hipótesis de caja 3D de 4 paredes, no obstante, la resolvemos bastante bien), Fig.5.1h (la imagen no cumple la hipótesis de mundo manhattan y nuestro algoritmo no es capaz de dar una solución buena a esa habitación.) Por otro lado, en la base de datos se tienen imágenes especialmente complejas en las que apenas vemos bordes de la imagen. La Fig.5.1i muestra la solución que nuestro algoritmo proporcionaría en una imagen en la que las oclusiones no nos permiten ver ninguno de los bordes estructurales.



(a) Im1 (b) Im2 (c) Im3

(d) Im4 (e) Im5 (f) Im6



(g) IM Dificil1 (h) IM Dificil2 (i) IM Dificil3

Figura 5.1: Representación general de *layouts* obtenidos

Capítulo 6

Conclusiones

Una de las principales aportaciones de este trabajo a sido la de generar dos tipos diferentes de hipótesis de *layout* y ser capaces de quedarnos en cada caso con la que mejor se adapta a la estructura de la habitación.

Hasta ahora todos los trabajos del estado del arte con nuestro mismo objetivo (ser capaces de detectar los bordes estructurales de una habitación a partir de una única imagen RGB) proponían una generación de hipótesis que implicaba razonamientos geométricos o bien, mas recientemente basar las hipótesis únicamente en técnicas de aprendizaje profundo. En este trabajo se propone por primera vez el, generar por un lado una hipótesis basada en razonamientos geométricos y por otro una basada en técnicas de aprendizaje profundo (*Deep Learning*) y hacer un algoritmo que sea capaz de seleccionar que hipótesis es mejor en cada caso.

Nuestros resultados experimentales demuestran que el algoritmo propuesto tiene un buen desempeño en la reconstrucción de los bordes estructurales en habitaciones de interior con una sola imagen RGB, situándose a la cabeza del estado del arte.

Bibliografía

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

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

[3] L. Del Pero, J. Bowdish, D. Fried, B. Kermgard, E. Hartley, and K. Barnard. Bayesian geometric modeling of indoor scenes. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2719–2726. IEEE, 2012.

[4] 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.

[5] C. Fernandez-Labrador, J. M. Facil, A. Perez-Yus, C. Demonceaux, and J. J. Guerrero. Panoroom: From the sphere to the 3d layout. *arXiv preprint arXiv:1808.09879*, 2018.

[6] C. Fernandez-Labrador, A. Perez-Yus, G. Lopez-Nicolas, and J. J. Guerrero. Layouts from panoramic images with geometry and deep learning. *arXiv preprint arXiv:1806.08294*, 2018.

[7] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[8] 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.

[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] C.-Y. Lee, V. Badrinarayanan, T. Malisiewicz, and A. Rabinovich. Roomnet: End-to-end room layout estimation. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, pages 4875–4884. IEEE, 2017.

[11] 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. IEEE, 2009.

[12] A. Mallya and S. Lazebnik. Learning informative edge maps for indoor scene layout prediction. In *Proceedings of the IEEE international conference on computer vision*, pages 936–944, 2015.

[13] S. Ramalingam, J. K. Pillai, A. Jain, and Y. Taguchi. Manhattan junction catalogue for spatial reasoning of indoor scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3065–3072, 2013.

[14] C. Rother. A new approach to vanishing point detection in architectural environments. *Image and Vision Computing*, 20(9-10):647–655, 2002.

[15] A. G. Schwing, T. Hazan, M. Pollefeys, and R. Urtasun. Efficient structured prediction for 3d indoor scene understanding. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2815–2822. IEEE, 2012.

[16] A. G. Schwing and R. Urtasun. Efficient exact inference for 3d indoor scene understanding. In *European Conference on Computer Vision*, pages 299–313. Springer, 2012.

[17] R. G. Von Gioi, J. Jakubowicz, J.-M. Morel, and G. Randall. Lsd: a line segment detector. *Image Processing On Line*, 2:35–55, 2012.

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

[19] Y. Zhang, F. Yu, S. Song, P. Xu, A. Seff, and J. Xiao. Large-scale scene understanding challenge: Room layout estimation. *accessed on Sep*, 15, 2015.

[20] H. Zhao, M. Lu, A. Yao, Y. Guo, Y. Chen, and L. Zhang. Physics inspired optimization on semantic transfer features: An alternative method for room layout estimation. *arXiv preprint arXiv:1707.00383*, 2017.

[21] C. Zou, A. Colburn, Q. Shan, and D. Hoiem. Layoutnet: Reconstructing the 3d room layout from a single rgb image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2051–2059, 2018.