Universidad Zaragoza
1542

Trabajo Fin de Grado

Estimación de profundidad a partir de una única imagen 360º mediante aprendizaje profundo

Single-image depth estimation of 360º panoramas with deep learning

Ingeniería Informática

Autor

Javier Giménez Garcés

Directores

Belén Masiá Corcoy

Daniel Martín Serrano

ESCUELA DE INGENIERÍA Y ARQUITECTURA

2020

AGRADECIMIENTOS

Este trabajo se ha realizado en el grupo *Graphics and Imaging Lab* de la Universidad de Zaragoza. Ante todo, me gustaría dar las gracias a mis directores, Belén y Daniel, por confiar en mi para realizar este proyecto, por el esfuerzo y la guía constante durante todo este tiempo, ya que he aprendido y mejorado mucho gracias a ellos. Gracias a los dos. No podía tener mejores mentores.

Gracias a los miembros del grupo, pese a que no he podido conocerlos tanto debido a las circunstancias extraordinarias de los últimos meses.

Muchas gracias a mis amigos, que han estado siempre ahí para lo que fuese y me han regalado muchos buenos momentos.

Y gracias a mis padres, María y José, y a mi hermana, Noelia, quienes sin importar la distancia, me han acompañado en todo el camino y han conseguido que me convierta en lo que soy hoy en día.

Por haber estado siempre conmigo, gracias.

Estimación de profundidad en imágenes 360º

RESUMEN

La Realidad Virtual (RV) es un paradigma de interacción persona-ordenador que ha ganado relevancia en los últimos años. El contenido empleado en RV puede tener diversos formatos. Entre ellos se encuentran los panoramas equirrectangulares, o panoramas 360º. Este tipo de contenido suele ser capturado mediante una cámara omnidireccional situada en un punto de vista estático. Debido a esto, la información de profundidad de la escena suele estar limitada o, incluso, no estar disponible. Sin embargo, esta información puede ser útil para distintas aplicaciones como la conducción autónoma, robótica (odometría), edición digital, etc. Los panoramas 360º, además, son un tipo de proyección fácilmente reproyectable en una esfera, lo cual motiva más su uso. El objetivo de este proyecto es crear y evaluar un sistema basado en aprendizaje profundo capaz de estimar la profundidad de una imagen 360º.

Primero, se ha realizado un estudio del estado del arte sobre la estimación de profundidad con redes neuronales profundas para conocer cómo se está abordando este problema. Dadas las limitaciones que se encontraron en estos sistemas, en su mayoría entrenados con imágenes tradicionales, se ha propuesto el estudio, diseño, modelado y evaluación de un sistema capaz de hacer frente al problema de estimación de profundidad en 360º.

Para alcanzar el objetivo de este trabajo, se han planteado una serie de modelos basados en aprendizaje profundo, y en concreto en redes neuronales convolucionales, utilizando convoluciones tradicionales y esféricas, y funciones de pérdida típicas, como el error cuadrático medio, también en ambas versiones (convencional y esférica). Las versiones esféricas tienen la propiedad de tratar los datos de manera diferente, enfocándose en los rasgos de las imágenes equirrectangulares.

Seguidamente, se han evaluado los diferentes sistemas propuestos. También se han comparado esos sistemas con otros del estado del arte para evaluar su mejoría. Además, se ha comparado el mejor modelo implementado con una estimación real de una cámara 360º.

En conclusión, tras observar los resultados obtenidos de diferentes modelos del estado del arte y propios del proyecto, se ha obtenido una mejora con respecto a los mismos, y se han identificado vías de mejora a futuro.

Índice

Capítulo 1

Introducción y objetivos

En los últimos años, la Realidad Virtual ha ganado relevancia en campos como la investigación, la simulación, videojuegos o entretenimiento. Este impulso se ha dado gracias a precios más asequibles, una tecnología más desarrollada o el aumento de contenido, lo cual ha permitido que la RV esté cada vez más presente en muchos hogares.

En RV existen dos tipos de contenido: el contenido sintético y el contenido capturado. El primero es generado por ordenador, con información perfecta de la escena. El segundo, por el contrario, es tomado por una - o múltiples - cámara, situada en un punto estático. Uno de los formatos más comunes para capturar una escena para RV es el panorama equirrectangular, que es un tipo de proyección que abarca una escena completa en 360º, aunque presenta distorsiones y deformaciones, sobre todo en la parte superior e inferior de la imagen. Este tipo de proyección es especialmente útil en realidad virtual, ya que su reproyección a una geometría esférica es trivial, facilitando su visualización y permitiendo un alto grado de realismo.

Para la obtención de panoramas existen una serie de dispositivos especializados en capturar este tipo de imágenes. Normalmente estos dispositivos disponen de una plataforma (o "esfera") de 2 o más cámaras cubriendo los 360º de la plataforma uniformemente, obteniendo información de profundidad buscando las mismas características en las diferentes imágenes de cada cámara (profundidad a partir de estéreo). Este método está limitado por la distancia máxima entre cada cámara, resultando en un paralaje entre imágenes insuficiente para obtener una buena información de profundidad. Otras opciones de captura serían las cámaras omnidireccionales basadas en espejos que no permiten obtener información de profundidad debido a que suelen ser un tipo de accesorio para cámaras convencionales.

La información de profundidad es especialmente útil, e incluso necesaria, en muchas

(a) Plataforma **(b)** Esfera **(c)** Espejos

Figura 1.1: Diferentes tipos de cámaras omnidireccionales. La **(a)** consiste en una plataforma con 2 o más cámaras repartidas uniformemente en 360º, mientras que en el dispositivo **(b)** las cámaras son repartidas uniformemente alrededor de una esfera. La **(c)** es una cámara convencional con una "lente"de espejo omnidireccional.

aplicaciones como la conducción autónoma, robótica (odometría), edición digital, etc. Algunos de estos sistemas son capaces de obtener esta información mediante sensores estéreo u otras tecnologías (láser, sonar, etc.), pero otros muchos carecen de estas herramientas, por lo que necesitan estimarla. No obstante, la mayoría de estos sistemas están diseñados para estimar la profundidad en base a imágenes 2D convencionales, por lo que fallan a la hora de estimar la profundidad en imágenes equirrectangulares, especialmente debido a sus distorsiones.

Por tanto, el objetivo final de este trabajo es construir un sistema que consiga estimar la profundidad de imágenes con proyección equirrectangular con mayor precisión que los sistemas actuales en el estado del arte. Para ello, se probarán diferentes modelos, herramientas y técnicas de aprendizaje profundo que permitan dicha mejora. Se compararán los resultados obtenidos de los diferentes modelos. Además, se evaluará la mejora de los modelos propuestos frente a los del estado del arte actual, así como frente al algoritmo propio de una cámara 360º.

(a) Imagen original **(b)** Mapa de profundidad

Figura 1.2: Ejemplo de un panorama equirrectangular RGB (a) y su mapa de profundidad (b). Los colores más oscuros (morados) indican cercanía mientras que los más claros (amarillo/naranja) indican lejanía.

1.1. Objetivos y alcance del proyecto

El objetivo general de este trabajo de fin de grado es mejorar la estimación de mapas de profundidad en imágenes equirrectangulares para su uso posterior en diferentes aplicaciones.

Para ello, las tareas realizadas en este trabajo de fin de grado se resumen en:

- Estudio y exploración de la literatura ya existente en cuanto a sistemas de estimación de profundidad y otros sistemas de RV que explotan este tipo de información (Sección 2).
- Introducción al aprendizaje profundo y las redes convolucionales explicando en qué consisten las redes neuronales, las convoluciones y sus parámetros, y las diferentes arquitecturas de redes existentes en el estado del arte (Sección 3).
- Implementación de un sistema real de estimación de profundidad basado en el estado del arte utilizando técnicas del aprendizaje profundo. Evaluación y comparación de los modelos implementados y su posterior propuesta e implementación de mejoras (función de pérdida). Evaluación y comparación de los modelos finales frente a sistemas del estado del arte (Sección 4).
- Propuesta de mejoras y posibles estudios futuros para mejorar el rendimiento del sistema implementado. Algunos ejemplos son la mejora de la función de pérdida y el estudio de activación de capas de la red para el análisis de las convoluciones esféricas (Sección 5).
- Conclusiones y resumen de los resultados obtenidos del sistema implementado. Se ha logrado alcanzar unos resultados considerables, además de destacar los conocimientos adquiridos durante el proceso (Sección 6).

1.2. Planificación y herramientas

Este trabajo de fin de grado se ha dividido en una serie de tareas relacionadas con la estructura inherente al proyecto. A cada una de esas tareas se le ha dedicado un total de horas que se puede consultar en la Figura 1.1.

La implementación de la red encargada de entrenar los modelos se ha llevado a cabo usando el conjunto de herramientas de aprendizaje automático Pytorch [Paszke et al., 2017] mediante el lenguaje de programación Python. También se ha utilizado

la librería TensorFlow [Abadi et al., 2015] para la reproducción de otros sistemas del estado del arte, [Laina et al., 2016]. Además, se han utilizado librerías como OpenCV [Bradski, 2000] y MathPlotLib [Hunter, 2007] para el tratamiento y visionado de las imágenes. Los procesos de entrenamiento y evaluación los modelos se han llevado a cabo en su mayoría en una máquina con un procesador Intel i7 de octava generación, 16 GB de RAM y una tarjeta gráfica Nvidia RTX 2060, haciendo uso de la tecnología CUDA, acelerando este proceso de entrenamiento.

El desarrollo del código utilizado para la implementación del sistema se ha llevado a cabo utilizando la herramienta de control de versiones Github en el repositorio público: `https://github.com/JaviBite/Depth-estimation-on-360-images`

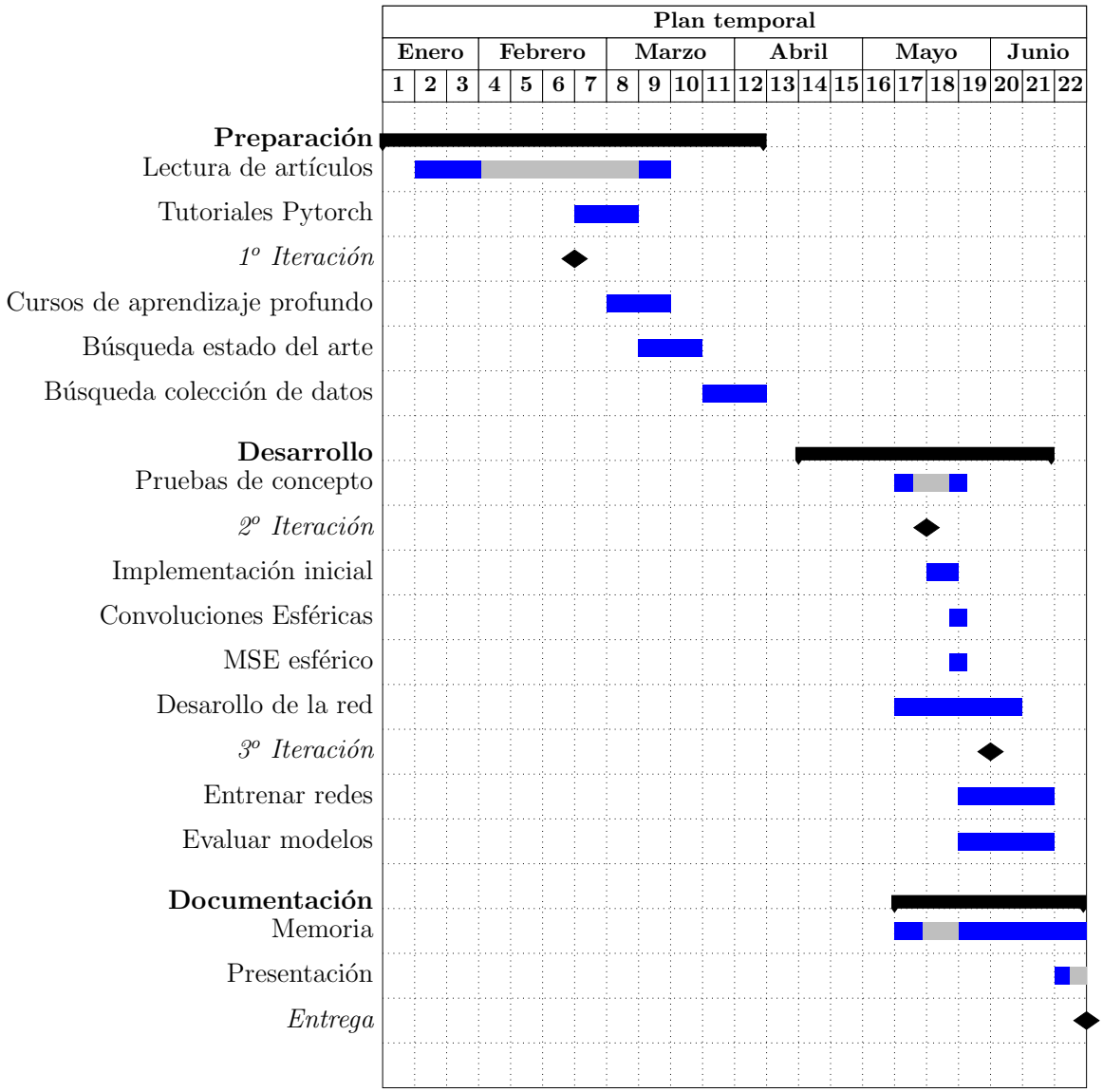


Figura 1.3: Planificación temporal de las diferentes tareas del proyecto desglosadas en un diagrama de Gantt.

Tarea	**Horas**
Preparación	**70**
- Lectura de artículos	25
- Tutoriales PyTorch	10
- Cursos de aprendizaje profundo	15
- Búsqueda estado del arte	10
- Búsqueda de colección de datos	10
Desarrollo	**107**
- Prueba de concepto	12
- Implementación Inicial	15
- Convoluciones y perdida esférica	10
- Reuniones	10
- Desarrollo de la red	35
- Evaluaciones y comparaciones	25
Documentación	**65**
- Memoria	40
- Contenido gráfico	10
- Presentación	15
TOTAL	**242**

Tabla 1.1: Número de horas dedicadas al proyecto desglosadas en las diferentes tareas relacionadas.

Capítulo 2

Trabajo relacionado

Como primer paso en el proyecto, se ha llevado a cabo un proceso de estudio del estado del arte actual en diferentes ámbitos de relevancia para el mismo.

2.1. Estimación de profundidad mediante aprendizaje profundo

La estimación de profundidad consiste en generar un mapa de profundidad de una imagen a partir de esta, donde cada píxel contiene un valor numérico que representa la distancia a la que se encuentra ese punto respecto a la cámara. Hoy en día, hay numerosos dispositivos que realizan esta tarea de diferentes maneras (láser, cámaras dobles, etc.), mientras que otros no pueden obtener esta información, o, de hacerlo, es poco precisa. Por eso, la estimación de profundidad mediante aprendizaje profundo es una buena alternativa, que ha sido ampliamente estudiada, ofreciendo buenos resultados [Godard et al., 2017].

Entre las técnicas de aprendizaje automático, se encuentran las redes convolucionales (CNN), que han demostrado su efectividad para muchas tareas de visión por computador que requieren trabajar con grandes volúmenes de imágenes, como puede ser la detección de objetos, reconocimiento facial, segmentación semántica, o estimación de profundidad. Este último será el caso de estudio de este trabajo.

Una de las técnicas más comunes que se ha utilizado para llevar a cabo este problema consiste en alimentar una red residual (ResNet) con una arquitectura codificador-decodificador utilizando imágenes con información de profundidad (RGBD), entrenándola para que a partir de los canales de color (RGB) sea capaz de predecir la profundidad (D).

Otra de las técnicas más recientes, y que ha obtenido resultados con gran precisión,

es la conocida como Monodepth2 [Godard et al., 2017]. Esta técnica se basa en calcular las disparidades de dos imágenes tomadas desde dos puntos diferentes, pero relativamente cercanos (estéreo), permitiendo que la red aprenda a predecir un punto de vista desde el otro, y según la disparidad entre ambos, estima su profundidad.

Sin embargo, en la mayoría de los trabajos dedicados a este problema han empleado imágenes tradicionales para entrenar y probar dichos modelos. Debido a esto, cuando se usa cualquiera de estos modelos sobre imágenes panorámicas equirrectangulares la estimación de profundidad no es muy precisa, e incluso puede contener incoherencias.

2.2. Adquisición y visualización de contenido 360º para realidad virtual

La realidad virtual es una tecnología en auge que ha evolucionado mucho en los últimos años y que ha abierto un amplio campo de investigación. Algunos trabajos como "*Instant 3D photography*", [Hedman et al., 2017], o "*Casual 3D photography*", [Hedman and Kopf, 2018], permiten generar escenas tridimensionales a través de una serie de imágenes RGBD bidimensionales. Otros trabajos, como "*Motion parallax for 360º RGBD video*", [Serrano et al., 2019], permiten añadir paralaje a un vídeo 360º capturado desde un punto de vista único: esto es, permite al usuario moverse libremente en la escena, infiriendo información sobre la escena que no fue capturada originalmente, pero ofreciendo una mejor experiencia al consumir el contenido. Aunque son trabajos en diferentes líneas, todos ellos tienen una necesidad común: información de profundidad.

Existen variedad de dispositivos que permiten adquirir imágenes panorámicas en 360º (equirrectangulares) fácilmente, incluso los móviles actuales pueden capturar este tipo de contenido. Sin embargo, su precisión no es muy elevada y, en general, no son capaces de capturar información de profundidad, o esta tampoco es especialmente precisa, dificultando el desarrollo de trabajos como los citados previamente.

Este trabajo persigue obtener mapas de profundidad de imágenes equirrectangulares sin necesidad de ningún dispositivo adicional, estimando dichos mapas gracias a técnicas de aprendizaje profundo.

2.3. Aprendizaje profundo en panoramas

En la actualidad en el estado del arte existen trabajos que tratan con imágenes panorámicas o de 360º para resolver problemas utilizando el aprendizaje profundo, algunos de estos ejemplos son los mapas de saliencia y detección o clasificación de imágenes. Los mapas de saliencia indican las zonas de una imagen que tienden a ser más atractivas al ojo humano, es decir, hacia dónde se suele mirar en dicha imagen. Un ejemplo de esto orientado a imágenes panorámica sería el trabajo de Zhang et al. [2018], que aplica técnicas novedosas como convoluciones y funciones de pérdida esféricas. La detección y clasificación de imágenes consiste en detectar "objetos" en imágenes o clasificarlas por clases, un ejemplo de esto aplicado en imágenes equirrectangulares sería SphereNet [Coors et al., 2018].

Las técnicas que se usan en estos trabajos son, sobre todo, el uso de unas convoluciones especiales esféricas y una función de pérdida (esférica) adaptada también a las imágenes equirrectangulares. Las convoluciones esféricas tratan de adaptar sus filtros para hacer frente a las distorsiones de las imágenes equirrectangulares. La función de pérdida esférica amplifica el error generado en función de las zonas menos distorsionadas de las imágenes equirrectangulares, por lo que será mayor en el ecuador de la imagen.

En este proyecto se incluirán y probarán las técnicas utilizadas en estos dos trabajos, evaluando así su aplicabilidad en el problema de estimación de profundidad.

Capítulo 3

Marco teórico

Este proyecto se basa en el uso del aprendizaje profundo, y más concretamente, en el uso de redes neuronales convolucionales. Para su desarrollo, ha sido necesario construir una base de conocimiento que abarca algunos conceptos que quedan fuera del marco teórico del grado de Ingeniería Informática, y que se introducen a continuación.

3.1. Redes neuronales

Las redes neuronales son una técnica de aprendizaje automático cuyo objetivo principal es encontrar un modelo capaz de resolver un problema concreto, de forma que, dada una entrada, sea capaz de generar una salida lo más precisa posible. Las redes neuronales intentan imitar el funcionamiento de las conexiones neuronales de organismos vivos, mediante un conjunto de neuronas conectadas entre sí que se activan o desactivan según la entrada (impulso) que reciben.

Estas neuronas se reparten a lo largo de la red agrupadas en capas, generalmente de manera secuencial. Cada neurona tiene asignado cierto *peso* que decide si, dada una entrada, se activa dicha neurona o no. Para llevar a cabo ese ajuste de pesos, las redes neuronales son entrenadas con un conjunto de datos del cuál, generalmente, se sabe la salida que se desea obtener. De esta forma, la red predice una salida para cada entrada, y se ajusta iterativamente en función de cómo de buena ha sido su predicción con respecto a esa salida esperada. Esta función es la conocida como función de pérdida (o *loss* en inglés)

El proceso de entrenamiento suele llevarse a cabo hasta que la función de pérdida converge y la red no sigue aprendiendo. Para ello, el proceso de entrenamiento se divide en épocas, donde cada época consiste en una iteración completa sobre todo el conjunto de datos. La evolución de la función de pérdida permite reconocer posibles problemas

de sobre ajuste (se ha ajustado demasiado a los datos de entrenamiento, y no es capaz de generalizar a otros datos) o sub ajuste (no se ha ajustado lo suficiente).

3.2. Redes convolucionales y aprendizaje profundo

Como se ha comentado en la Sección 2, dentro de las diferentes técnicas de aprendizaje automático, hay un tipo de redes conocidas como redes convolucionales (CNN). Este tipo de redes han resultado ser especialmente útiles en problemas que trabajan con imágenes o vídeos.

Las redes convolucionales son un tipo de red neuronal donde se aplican una serie de convoluciones (en inglés, *kernel*) sobre la entrada, de forma que, en cada paso, se extraen distintas características intrínsecas de los datos (en inglés, *features*). En este caso, la entrada a la red suele ser una imagen (matriz o tensor) a la que se le aplican una serie de convoluciones (ver Figura 3.1) para extraer las características.

La entrada a cada capa de la red se puede definir por un tensor con tamaño (B, C, H, W) siendo B el número de muestras con que se alimentará a la red en cada entrada (en inglés, *batch size*). La C representa el número de canales de la capa. Por ejemplo, para una imagen RGB tradicional, existen tres canales, uno por cada color (rojo, verde y azul), mientras que una imagen en escala de grises solo tendría un canal. En capas ocultas (intermedias), el número de canales aumenta, representando las distintas características que se extraen de los datos. Por último, la H y W indican el tamaño de la entrada en dos dimensiones (altura y anchura). Cada operación de convolución tiene una serie de parámetros que definen cómo será la operación, qué entrada admite, y qué salida generará. Entre ellos, se encuentran el tamaño del filtro (en inglés, *kernel size*), número de canales de entrada y salida, el paso (en inglés, *stride*) y margen (en inglés, *padding*).

- **Canales:** El número de canales esperados para la entrada y salida. Cada canal se centra en aprender un cierto rasgo (*feature*) de la entrada, por lo que cada uno tiene unos pesos diferentes.
- **Filtro** (*Kernel*): Es una matriz NxN que aplica la operación de convolución en función de los pesos que se han ajustado durante el entrenamiento.
- **Margen** (*Padding*): Son las filas y columnas adicionales que se añaden para permitir que la operación de convolución se ejecute tantas veces como se requiera para obtener el tamaño de salida adecuado.

– **Paso** (*Stride*): El paso o desplazamiento del filtro entre muestreos de la convolución. Nótese que un paso diferente de 1 reduciría la resolución de la entrada.

Una convolución consiste en aplicar a cada dato de un tensor o matriz (píxel si son imágenes) una operación en la que participan también sus N vecinos. Esta operación se define por la suma de las multiplicaciones de cada vecino y el dato en sí por su peso correspondiente del filtro (*Kernel*) correspondiente.

$$g(x,y) = w * f(x,y) = \sum_{dx=-a}^{a} \sum_{dy=-b}^{b} w(dx,dy) f(x+dx, y+dy) \tag{3.1}$$

Donde $g(x,y)$ es la salida, $f(x,y)$ la entrada, la w representa los pesos del filtro (*Kernel*) donde $0 \leq dx \leq a$ y $0 \leq dy \leq b$, siendo a la altura y b la anchura del filtro.

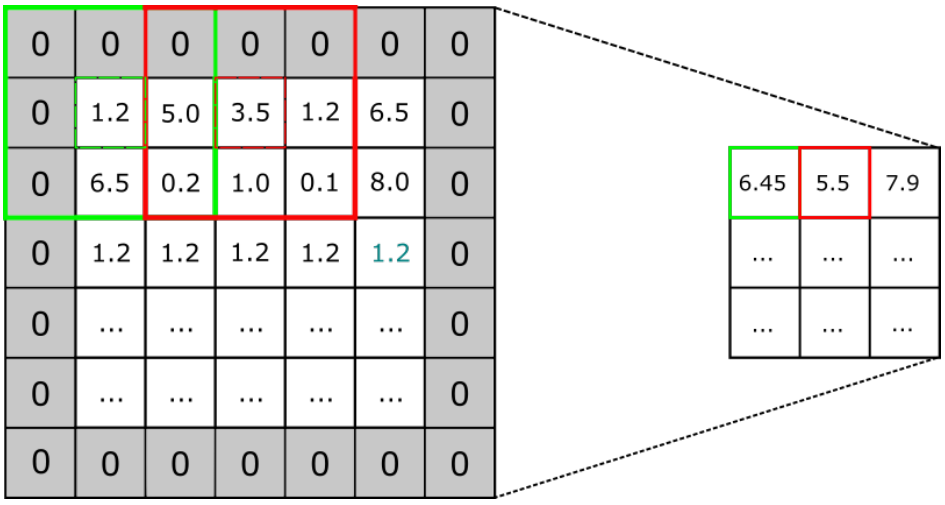


Figura 3.1: Ejemplo de una imagen 5x5 a la que se le está aplicando una convolución con un tamaño de filtro de 3 (cuadrados verde y rojo) con todos sus pesos a 0.5, margen (*padding*) de ceros de tamaño 1 (filas y columnas de 0 grises) y un paso (*stride*) de 2, por lo que el filtro avanza de dos en dos posiciones resultando en una imagen más pequeña que la original.

En la Figura 3.1 se puede ver como la convolución se está aplicando a cada píxel de la entrada, por ejemplo, el primer píxel de salida (cuadrado verde) es calculado por la siguiente fórmula: $w * (1{,}2 + 5{,}0 + 6{,}5 + 0{,}2) = 0{,}5 * (1{,}2 + 5{,}0 + 6{,}5 + 0{,}2) = 6{,}45$

Además de las convoluciones también se pueden realizar operaciones como concatenación de tensores, juntar varias salidas en una sola, normalización de la entrada (*Batch norm*), desescalado (*down-sampling*), importante cuando se aumenta el número de canales, o escalado (*up-sampling*), en caso contrario.

3.3. Arquitecturas de CNN: ResNet

Conforme se ha ido investigando en técnicas de aprendizaje profundo se han explorado diferentes tipos de arquitecturas de redes convolucionales para solventar diversos problemas [Standford, 2020]. Las más relevantes son las siguientes:

- **AlexNet o ImageNet:** Una de las primeras arquitecturas que mostraron que era capaz de reconocer objetos en imágenes con solo ocho capas, [Krizhevsky et al., 2012]. Más tarde fue superada por ZFNet [Zeiler and Fergus, 2013].
- **VGGNet:** Una red más profunda que sus predecesoras que además utilizaba convoluciones 3x3 seguidas actuando como una sola de 7x7. Prueban que redes más grandes funcionan mejor. [Simonyan and Zisserman, 2014].
- **GoogLeNet:** Una de las arquitecturas que tratan de buscar eficiencia con convoluciones 1x1 de cuello de botella (reducen el número de canales de features) y evita el uso de capas conectadas completamente usando una media de agrupación global, [Szegedy et al., 2014].
- **ResNet:** Redes extremadamente profundas, desde 18 capas a 152. Consiguió una precisión similar a la de los seres humanos, y es una arquitectura que ha sido ampliamente utilizada para múltiples problemas. Este tipo de redes da prioridad a la eficiencia en lugar de a un aumento no muy significativo de la precisión. [He et al., 2013].

En este proyecto, se ha seguido la arquitectura ResNet para solventar el problema, ya que, tras estudiar el estado del arte, se ha observado que esta arquitectura ha sido enormemente explotada, además de ser una de las que más precisión consigue. En la arquitectura global de los sistemas diseñados en este proyecto, la ResNet forma parte de lo que se denomina codificador, que extrae las características (*features*) de las imágenes. El decodificador utiliza esas características (*features*) para, a través de varios escalados y convoluciones (con menos canales de salida que entrada), conseguir la salida deseada, que para el caso de este proyecto, será un mapa de profundidad.

Capítulo 4

Un modelo de estimación de profundidad

Se va a implementar un sistema capaz de estimar la profundidad de una imagen equirrectangular con técnicas de aprendizaje profundo con el objetivo de conseguir una mejor estimación que los sistemas actuales en el estado del arte. Se probarán y evaluarán diferentes modelos para analizar el rendimiento de cada uno y determinar cuál es el más preciso. También se evaluará el mejor modelo frente a otros sistemas del estado del arte.

4.1. Arquitectura de la red

Como se ha comentado previamente, la arquitectura de la red se basa en el esquema codificador-decodificador, en concreto, en la arquitectura de ResNet18. La red está inspirada en "Monodepth2"[Godard et al., 2017], salvo que en este caso no se genera un nuevo punto de vista, y se realiza el proceso a partir de una única imagen. A lo largo del proyecto, las convoluciones y la función de pérdida se han alternado entre sus implementaciones tradicionales y otras implementaciones adecuadas a casos panorámicos [Zhang et al., 2018] [Coors et al., 2018].

El codificador (ver especificaciones en Tabla 4.1) se encarga de extraer las características (*features*) de la imagen de entrada (RGB normal), mientras que el decodificador (ver especificaciones en Tabla 4.2) estima un mapa de profundidad, haciendo uso de conexiones "*skip*"[Shelhamer et al., 2016] de los bloques de activación del codificador, facilitando así el proceso de obtención de la salida.

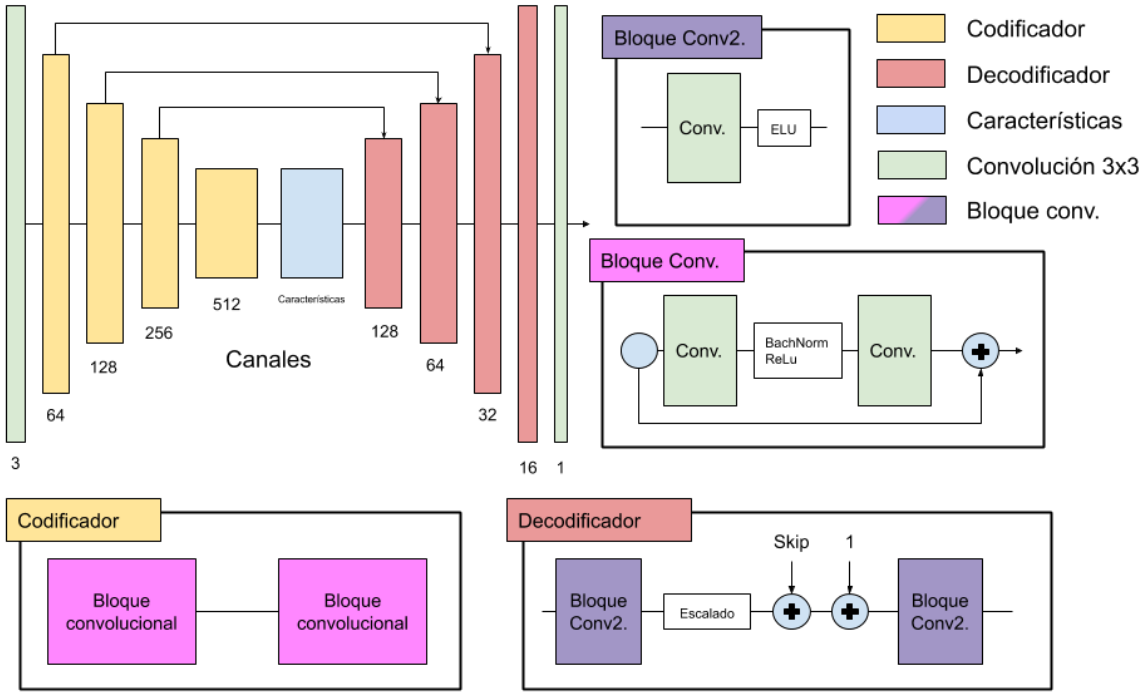


Figura 4.1: La estructura de la red neuronal se basa en la naturaleza de codificador-decodificador común en diversos modelos de este ámbito. Las capas de codificación reducen la dimensionalidad, agregando más canales, mientras que las decodificaciones realizan la tarea inversa, hasta alcanzar la salida esperada. Se puede apreciar el uso de conexiones "*skip*"[Shelhamer et al., 2016] para tratar problemas de diferentes resoluciones.

Codificador				
capa	**Tamaño de filtro (*kernel*)**	**Paso (*stride*)**	**canales (entrada/salida)**	**entrada**
econv1	3	2	3/64	img
maxpoll	3	2	64/64	econv1
convs2	3	1	64/64	maxpool
econv2	3	1	64/64	convs2
convs3	3	2	64/128	econv2
econv3	3	1	128/128	convs3
convs4	3	2	128/256	econv3
econv4	3	1	256/256	convs4
convs5	3	2	256/512	econv4
econv5	3	1	512/512	convs5

Tabla 4.1: Arquitectura del codificador basado en ResNet con 18 capas. Se trata de una serie secuencial de convoluciones de mayor a menor resolución y de menor a mayor número de canales, todas ellas precedidas por una convolución inicial. En cada bloque de convoluciones, la salida es normalizada (*BatchNorm*). Algunas capas de la arquitectura se han ocultado y agrupado con otras ya que repiten convoluciones 3x3 sin cambiar el número de canales o resolución de la entrada (convs2 a convs5). Se encarga de extraer las características de la entrada.

Decodificador de profundidad					
capa	**Tamaño de filtro (*kernel*)**	**Paso (*stride*)**	**canales**	**entrada**	**activación**
upconv5	3	1	256	econv5	ELU
iconv5	3	1	256	upconv5, econv4	ELU
upconv4	3	1	128	iconv5	ELU
iconv4	3	1	128	upconv4, econv3	ELU
disp4	3	1	1	iconv4	Sigmoid
upconv3	3	1	64	iconv4	ELU
iconv3	3	1	64	upconv3, econv2	ELU
disp3	3	1	1	iconv3	Sigmoid
upconv2	3	1	32	iconv3	ELU
iconv2	3	1	32	upconv2, econv1	ELU
disp2	3	1	1	iconv2	Sigmoid
upconv1	3	1	16	iconv2	ELU
iconv1	3	1	16	upconv1	ELU
disp1	3	1	1	iconv1	Sigmoid

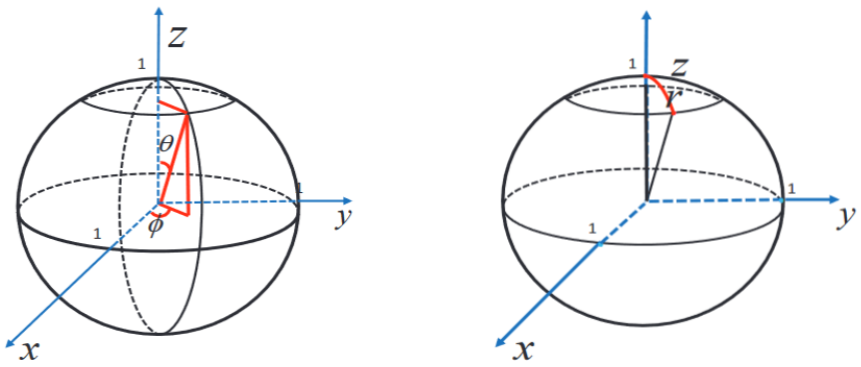
Tabla 4.2: Arquitectura del decodificador. Conjunto de convoluciones y escalados de menor a mayor resolución y de mayor a menor número de canales hasta un solo canal y a la resolución de entrada. Se encarga de construir el mapa de profundidad partiendo de las características extraídas por el codificador.

4.2. Convoluciones

En un principio, la arquitectura de la red contaba con diferentes tamaños de convoluciones en sus capas, pero para poder comparar los diferentes modelos propuestos (con convoluciones esféricas y normales) se han tenido que modificar algunos parámetros de estas convoluciones como el tamaño de filtro a 3 y en algunos casos, el margen (para ser consistente con el cambio del tamaño del filtro). Este cambio se debe a que la implementación elegida de las convoluciones esféricas está preparada para usar un tamaño de kernel de 3x3.

4.2.1. Convoluciones esféricas (SphereNet)

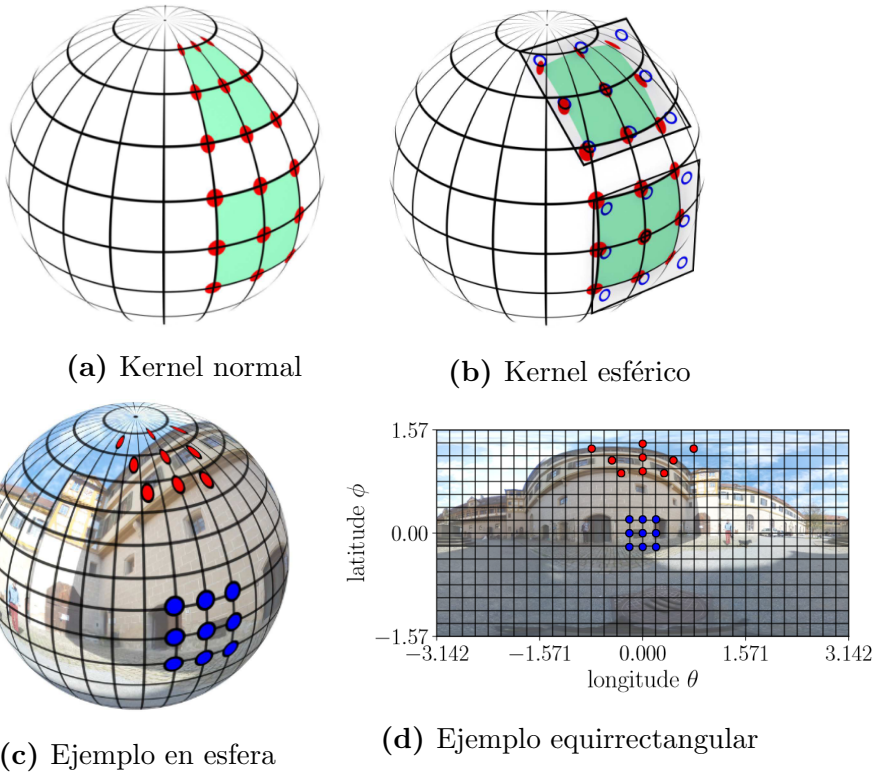
Los filtros que son aplicados en las convoluciones tradicionales pueden no funcionar correctamente con imágenes equirrectangulares por sus distorsiones. Esto se debe a que en una imagen con proyección perspectiva normal, cada punto de la imagen corresponde con su posición real en el espacio, mientras que en proyección equirrectangular esto no ocurre (ver Figura 4.3). Existen unos tipos especiales de convoluciones esféricas 3x3 [Coors et al., 2018] que sí tienen en cuenta este problema.



(a) Ángulo Sólido **(b)** Relación

Figura 4.2: (a) es una representación tridimensional de un ángulo sólido definido por θ y ϕ en una esfera. En la Figura **(b)** r indica el grado de distorsión según el ángulo sólido en el que se encuentra.

La idea principal de estas convoluciones consiste en trasladar las operaciones locales que se hacen en una convolución convencional al dominio de la superficie de una esfera. Esto se consigue representando el kernel como un pequeño plano tangente (ver figura 4.3) a la superficie de dicha esfera (imagen equirrectangular) representando una porción de la imagen sin distorsiones [Coors et al., 2018].



(a) Kernel normal **(b)** Kernel esférico

(c) Ejemplo en esfera **(d)** Ejemplo equirrectangular

Figura 4.3: En la imagen **(a)** se puede ver como se aplicaría el filtro (*kernel*) de una convolución tradicional a una imagen equirrectangular. La imagen **(b)** muestra cómo se aplicaría la misma convolución que en **(a)** pero utilizando un filtro (*kernel*) esférico. En la imagen **(c)** podemos ver un ejemplo aplicado en una escena con su equivalente en proyección equirrectangular **(d)** donde se observa fácilmente como actúa el filtro (*kernel*) esférico para lidiar con las distorsiones. [Coors et al., 2018]

.

4.3. Funciones de pérdida

Las funciones de pérdida (*loss*) convencionales suelen ser el error absoluto medio y el error cuadrático medio (MSE). En este proyecto se usará esta última junto con una función de pérdida especial para las imágenes equirrectangulares, el error cuadrático medio esférico [Zhang et al., 2018], que utiliza una serie de pesos que tienen en cuenta las distorsiones de las imágenes según el ángulo solido en el que se encuentra cada píxel.

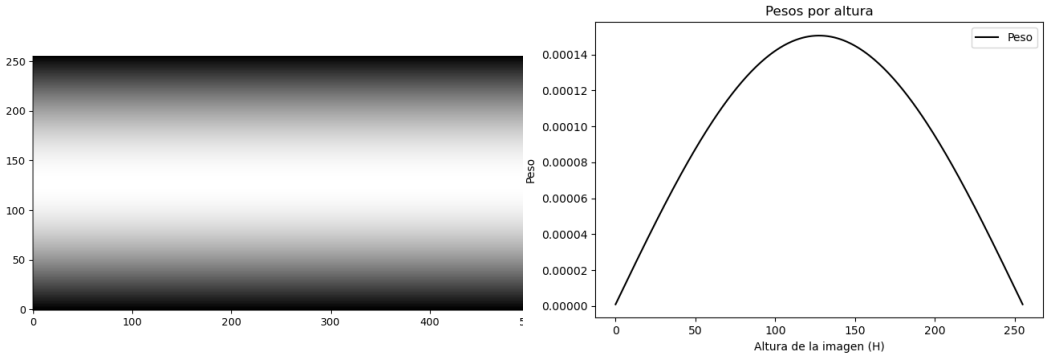
4.3.1. Error cuadrático medio esférico

El error cuadrático medio es una función muy utilizada en problemas similares, solo que está basada en imágenes convencionales en las que la discretización se realiza de forma homogénea en el espacio de la imagen. No obstante, en las imágenes equirrectangulares, esa homogeneidad no se cumple, ya que hay zonas distorsionadas (p.e., los polos) que ocupan más píxeles que las zonas céntricas.

Para tener en cuenta esto, se utiliza una modificación del error cuadrático medio que introduce el uso de una matriz de pesos en la que cada componente indexada por el ángulo sólido ($\Omega(\theta, \phi)$) de la esfera (ver Figura 4.2) tiene asignado un peso proporcional al ángulo sólido en el que se encuentra, penalizando así más los errores en el ecuador, y menos los errores en puntos cercanos a los polos. La función de pérdida (*loss*) utilizada sería la siguiente:

$$L = \frac{1}{n} \sum_{K=1}^{n} \sum_{\theta=0,\phi=0}^{\Theta,\Phi} w_{\theta,\phi} (S_{\theta,\phi}^{(k)} - \hat{S}_{\theta,\phi}^{(k)})^2 \tag{4.1}$$

donde w representa la matriz de pesos (ver Figura 4.4), θ y ϕ representan la latitud y la longitud de cada píxel en sus coordenadas esféricas. S (Mapa de profundidad real) y $\hat{S}$ (estimación del mapa de profundidad). K representa la imagen k-ésima del total de n imágenes.



(a) Matriz de pesos **(b)** Peso según la altura

Figura 4.4: La imagen **(a)** representa la matriz de pesos asociada a el error cuadrático medio esférico. Los colores blancos indican un peso mas elevado mientras que los negros son pesos más bajos. En este caso se ha inicializado de la siguiente manera: $w_{\theta,\phi} = \Omega(\theta, \phi)/4\pi$ (4π es el ángulo sólido unidad). La matriz tiene tamaño 256x516 como las imágenes de la colección de datos. La **(b)** muestra la variación de pesos en función de la coordenada angular vertical.

4.4. Modelos implementados

A lo largo del trabajo, se han implementado modelos que combinaban los dos tipos de convoluciones y de funciones de pérdida. De esta serie de modelos, se va a evaluar cuál de todos obtiene resultados más precisos. Para ello se han especificado una serie de métricas cuantitativas.

4.4.1. Métricas

Las métricas utilizadas para evaluar los modelos son ampliamente conocidas y utilizadas en diferentes sistemas similares para evaluar resultados. Han sido utilizados en otros proyectos [Godard et al., 2017], [Zhan et al., 2018]. También se ha introducido el uso del error cuadrático medio esférico (Sección 4.3.1).

- **Error absoluto relativo** (Abs rel): Media de la división de la diferencia de los datos correctos y los estimados entre los datos correctos.

$$\frac{1}{N}\sum_{i}^{N}\frac{\mid y_i - \hat{y}_i \mid}{y_i}$$

- **Error cuadrático relativo** (Sq rel): Media de la división de la diferencia al cuadrado de los datos correctos y los estimados entre los datos correctos.

$$\frac{1}{N}\sum_{i}^{N}\frac{(y_i - \hat{y}_i)^2}{y_i}$$

- **Raíz del error cuadrático medio** (RMSE): Raíz de la media de la diferencia al cuadrado de los datos correctos y los estimados.

$$\sqrt{\frac{1}{N}\sum_{i}^{N}(y_i - \hat{y}_i)^2}$$

- **Error cuadrático medio esférico** (SMSE): Media de la diferencia al cuadrado de los datos correctos y los estimados multiplicados por un peso correspondiente a su ángulo sólido en coordenadas de la imagen (ver Sección 4.3.1).

- **Raíz del error cuadrático medio logarítmico** (RMSE log): Raíz de la media de la diferencia al cuadrado del logaritmo de los datos correctos y el logaritmo de los estimados.

$$\sqrt{\frac{1}{N}\sum_{i}^{N}(\log y_i - \log \hat{y}_i)^2}$$

- **Tasa de acierto** (a1-3): Media de la tasa de acierto por dato. Un dato (píxel) se considera acertado si tras obtener el máximo entre la división del dato real y el predicho y su inversa, éste es inferior al umbral de 1,25 (a1), $1{,}25^2$ (a2) o $1{,}25^3$ (a3)

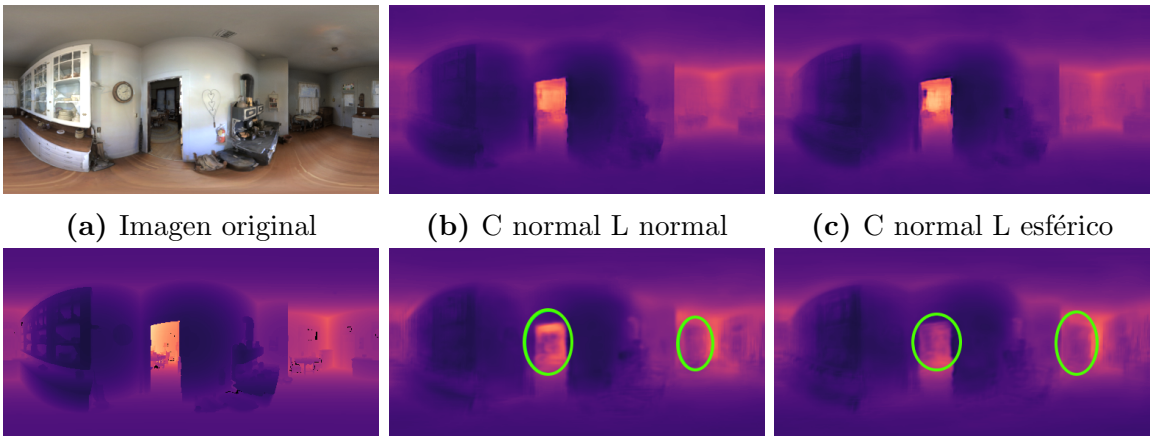
4.4.2. Aproximación inicial

Como primera aproximación, se han implementado y entrenado cuatro modelos, combinando los diferentes parámetros que se han comentado previamente. Los resultados pueden observarse en la Tabla 4.3.

Convoluciones	Pérdida	Épocas	Abs rel	Sq rel	RMSE	SMSE	RMSE log	a1	a2	a3
Normal	Normal	40	3.746	1.872	**0.062**	0.086	0.687	**0.894**	**0.947**	**0.958**
	Esférica		**3.706**	**1.851**	**0.062**	**0.085**	**0.684**	0.888	0.946	**0.958**
Esférica	Normal		5.427	2.192	0.077	0.125	0.736	0.842	0.931	0.953
	Esférica		5.107	2.118	0.075	0.119	0.716	0.846	0.933	0.954
Esférica	Esférica	100	4.796	2.158	0.073	0.113	0.705	0.859	0.937	0.955

Tabla 4.3: Tabla de resultados tras evaluar los diferentes modelos combinando convoluciones y funciones de pérdida esféricas con normales. El mejor resultado es la combinación de convolución tradicional con función de pérdida esférica. Las métricas son la expuestas en la Sección 4.4.1. Las métricas en azul claro significan que más bajo mejor, en morado cuanto más altas, mejor. Los mejores resultados quedan indicados en negrita, mientras que los peores quedan indicados con un subrayado.

Como se puede observar en la Tabla 4.3, el modelo con convoluciones normales y función de pérdida esférica (SMSE) era el que mejores resultados ofrecía, muy cerca del modelo convencional (convoluciones y loss ambos normales).

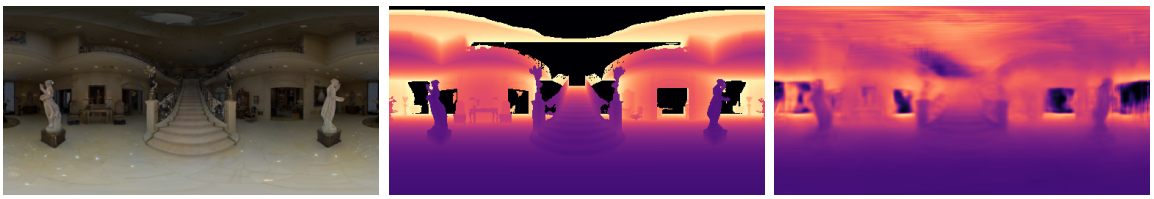


(a) Imagen original **(b)** C normal L normal **(c)** C normal L esférico

(d) Profundidad real **(e)** C esféricas L normal **(f)** C esféricas L esférico

Figura 4.5: Resultados cualitativos del primer modelo. Los círculos verdes indican zonas con artefactos y errores notorios respecto a la imagen original. C = convoluciones, L = pérdida.

Explorando los resultados cualitativamente (ver Figura 4.5 y 4.6), se observó que una parte del error podría deberse a la interpretación de los datos. En la práctica, las imágenes de la colección de datos empleada para entrenar la red poseen varias zonas de profundidad no definida codificada como 0. Estas zonas suelen estar en lugares de



(a) Imagen original **(b)** Mapa de profundidad **(c)** Estimación

Figura 4.6: Resultado del modelo completamente esférico. De izquierda a derecha: original, mapa de profundidad real y mapa de profundidad estimado.

la imagen donde la distancia de profundidad debería ser muy alta (por lo que no está definida), pero al codificarse como 0 se dificulta el aprendizaje de la red. Por tanto, en una segunda iteración, se modificará la función de pérdida.

También se probó un modelo con más épocas (ver última fila de la Tabla 4.3) de las normales para comprobar si eran necesarias más épocas para que el sistema convergiera, pero no se notó diferencia y las gráficas mostraban una convergencia alrededor de la época 35. Por tanto, se estableció 40 como el número de épocas a emplear en las siguientes iteraciones.

4.4.3. Mejora de la función de pérdida

Como se ha introducido en la sección anterior, la tasa de error era muy alta en imágenes con demasiada información no definida (el mapa de profundidad no estaba definido en varios puntos) y esto podía dificultar el aprendizaje de la red.

Para solventar este problema se modificó la función de pérdida y las métricas para no tener en cuenta el error causado por la predicción en zonas de la imagen donde no estaba definida la profundidad. Esto mejoró los resultados prediciendo valores acordes con el mapa de profundidad original, y dejando en las zonas sin definir valores relativamente acertados (sin contradicciones). Los resultados cuantitativos obtenidos con estos modelos se pueden ver en la Tabla 4.4.

Convoluciones	Pérdida	Épocas	Abs rel	Sq rel	RMSE	SRMSE	RMSE log	a1	a2	a3
Normal	Normal	40	0.092	0.007	0.042	0.034	0.132	0.903	0.978	0.993
	Esférica		**0.066**	**0.004**	**0.032**	**0.020**	**0.101**	**0.944**	**0.989**	**0.997**
Esférica	Normal		0.124	0.011	0.054	0.056	0.168	0.855	0.961	0.987
	Esférica		0.088	0.006	0.041	0.033	0.129	0.910	0.980	0.994

Tabla 4.4: Tabla de resultados del segundo modelo cambiando la función de pérdida explicado en la sección 4.4.3. Utiliza las métricas de Tabla 4.3 modificadas al igual que la función de pérdida, no penalizando las zonas de profundidad no definidas.

En este modelo se puede apreciar como las tasas de errores han bajado

considerablemente en todas sus versiones. Sin embargo, la tendencia es la misma, ya que el mejor modelo sigue siendo el de convoluciones tradicionales y función de pérdida esférica.

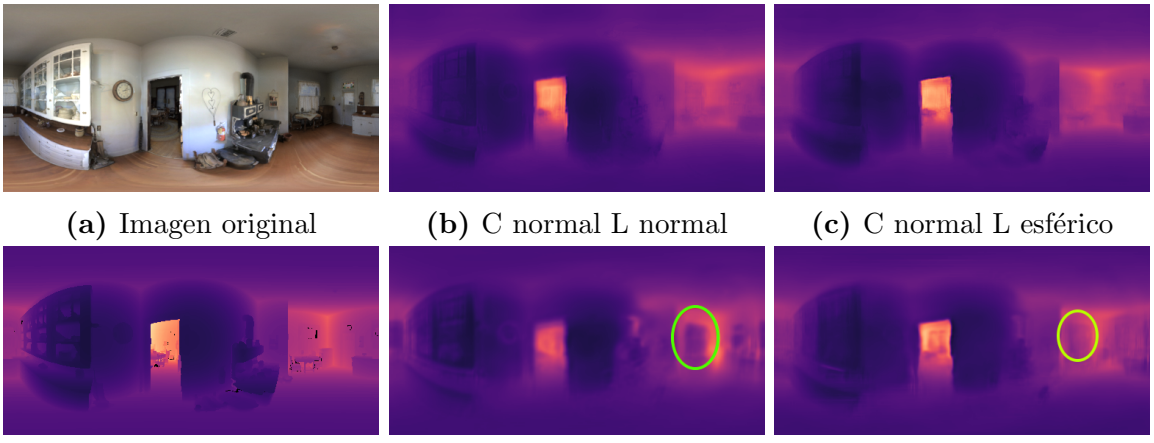
Los modelos con convoluciones esféricas siguen si mostrar unos resultados tan precisos como los modelos con convoluciones tradicionales. Una posible explicación de este fenómeno sería que, en el caso de las convoluciones normales la red ha podido aprender la estructura inherente a los panoramas, y la misma red podría estar corrigiendo esas distorsiones. Todo esto se podría comprobar observando las activaciones de la red, dejando este paso como trabajo futuro.

Observando cualitativamente los resultados de este segundo modelo (Figura 4.7) se puede apreciar cierta mejora en todos los modelos en general, destacando que algunos de los artefactos que aparecían en el modelo anterior (ver Figura 4.5) parecen haberse atenuado.

4.5. Comparativa

Se han comparado los resultados de diferentes sistemas del estado del arte con el diseñado en este trabajo para evaluar el rendimiento del mismo. Se ha utilizado el mismo conjunto de datos de test que en las evaluaciones anteriores al igual que las ultimas métricas utilizadas.

Se puede observar la mejora respecto a otros sistemas del estado del arte cuantitativa



(a) Imagen original **(b)** C normal L normal **(c)** C normal L esférico

(d) Mapa de profundidad real **(e)** C esféricas L normal **(f)** C esféricas L esférico

Figura 4.7: Resultados cualitativos del segundo modelo. Se ha utilizado la misma imagen que en los resultados del modelo 1 (ver Figura 4.5). Los círculos verdes indican zonas con artefactos y errores notorios respecto a la imagen original. C = convoluciones, L = pérdida.

y cualitativamente en la Tabla 4.5 y Figura 4.8. Los sistemas implementados estiman un mapa de profundidad bastante acertado mientras que los sistemas del estado de arte actual generan unos resultados menos precisos. Esta falta de precisión en los sistemas del estado del arte puede deberse, por un lado, a que no están diseñados para funcionar sobre panoramas equirrectangulares. Además, son modelos entrenados con escenas con rangos de profundidad ligeramente distintos. Dado esto, un estudio más intensivo sería interesante.

Comparando solo los modelos implementados en este proyecto, se puede observar, como se ha indicado antes, que el modelo con mejor resultado es el de convoluciones normales y función de pérdida esférica ya que se puede apreciar que es bastante similar al mapa de profundidad original y no se aprecian los artefactos vistos en las Figuras 4.5 y 4.7.

Una de las mejoras que ofrece el sistema desarrollado frente al resto es que no sobrestima la profundidad, es decir, no se estima que todo esté más lejos de lo que en realidad está, como ocurre con los otros sistemas, además de que no aparecen artefactos tan notorios como los que podemos encontrar, por ejemplo, en el sistema de Godard et al. [2019], adaptándose mejor a las características de las imágenes equirrectangulares.

Puede parecer que el sistema implementado sistema falla al predecir zonas muy alejadas debido a que en el mapa de profundidad original aparecen zonas negras mientras que el modelo implementado estima lejanía, esto indica que la profundidad no está definida en esa zona por lo que, en términos prácticos, no se consideraría un error, además de que en su mayoría estas zonas se encuentran en píxeles correspondientes a zonas muy alejadas. Como se explica en la Sección 4.4.3 esto ayudó al sistema a disminuir su error.

Observando el sistema desarrollado por Laina et al. [2016], cualitativamente ofrece

Modelo	**Abs rel**	**Sq rel**	**RMSE**	**SRMSE**	**RMSE log**	**a1**	**a2**	**a3**
Normal Normal	0.092	0.007	0.042	0.034	0.132	0.903	0.978	0.993
Normal Esférico	**0.066**	**0.004**	**0.032**	**0.020**	**0.101**	**0.944**	**0.989**	**0.997**
Esférico Normal	0.124	0.011	0.054	0.056	0.168	0.855	0.961	0.987
Esférico Esférico	0.088	0.006	0.041	0.033	0.129	0.910	0.980	0.994
[Godard et al., 2019] mono	0.858	0.258	0.234	0.706	1.472	0.217	0.390	0.557
[Godard et al., 2019] mono stereo	0.971	0.324	0.260	0.875	1.475	0.197	0.357	0.522
[Laina et al., 2016]	1.309	0.536	0.339	1.582	0.837	0.131	0.254	0.426

Tabla 4.5: Tabla de resultados comparando los sistemas implementados frente a otros del estado del arte. Los modelos implementados están nombrados indicando primero el tipo de convolución y la función de pérdida en ese orden.

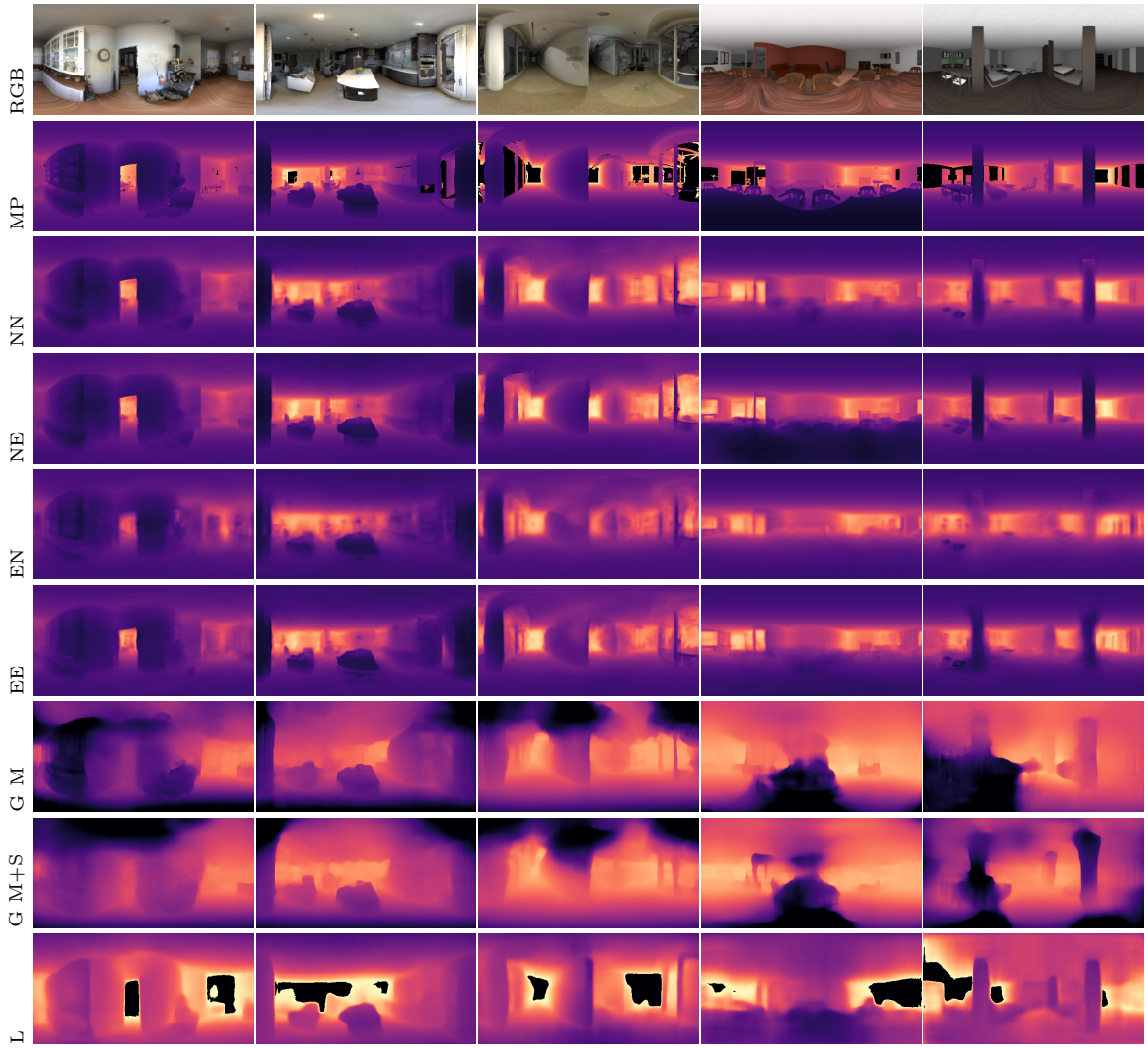
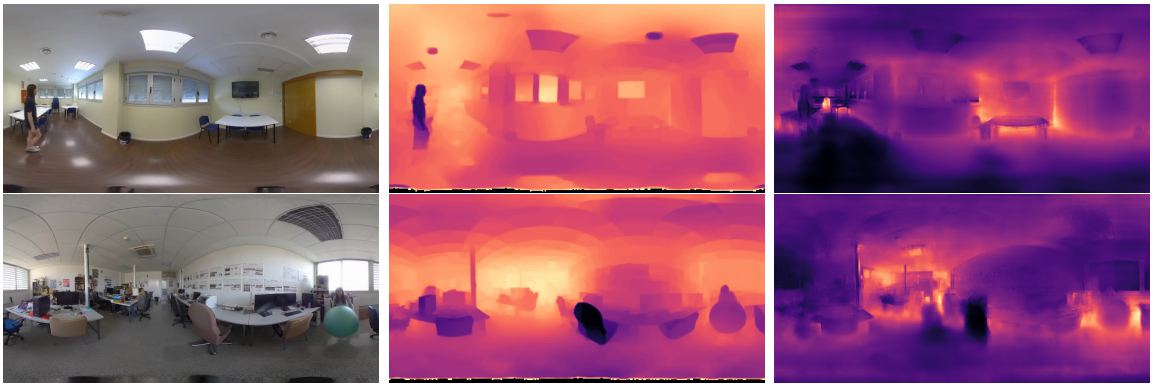


Figura 4.8: Matriz de resultados comparando todos los últimos modelos con varios sistemas del estado del arte. De izquierda a derecha: imagen original (RGB), mapa de profundidad real (MP), modelo normal (NN), convoluciones normales y pérdida esférica (NE), convolución esférica y pérdida normal (EN), esférico (EE), [Godard et al., 2019] monocular (GM) y monocular con estéreo (GM+S) y [Laina et al., 2016] (L). Para más resolución ver Figura A.1 y A.2.

unos resultados aceptables a pesar de estimar la escena más alejada de lo que está, pero falla en los polos al igual que pasa en los modelos implementados excepto en el de convoluciones normales y función de pérdida esférica. Esto se aprecia en la columna cuatro de la Figura 4.8.

4.5.1. Comparativa con cámara 360º

Se han comparado el modelo con convoluciones normales y función de pérdida esférica frente a los resultados del algoritmo propio de una cámara 360º. Este algoritmo consiste en buscar pares de características iguales entre sus diferentes cámaras (estéreo) y en base a la distancia entre las cámaras obtener la profundidad mediante paralaje.



(a) Imagen original **(b)** Profundidad cámara **(c)** Estimación NE

Figura 4.9: Comparación del mapa de profundidad generado por el algoritmo de una cámara 360º y la estimación resultante del modelo implementado con convoluciones normales y función de pérdida esférica. La estimación (c) se han normalizado para que la comparación visual sea más clara.

Como se puede observar en la Figura 4.9 la estimación del modelo implementado estima en general más cercanía que el mapa de profundidad de la cámara, pero los cambios de profundidad no son tan drásticos como en el generado por la cámara. En la primera imagen podemos ver como las esquinas y la mesa en la parte izquierda de la imagen ofrecen un mejor resultado en la estimación del sistema desarrollado que en la generada por la cámara ya que en este caso a penas se percibe diferencia.

En las ventanas o zonas muy iluminadas (parte izquierda de la primera imagen o ventana derecha e izquierda de la segunda, Figura 4.9) no se consigue tanta precisión con nuestro modelo, estimando estas zonas más cercanas de lo que en realidad están, posiblemente por cómo ha aprendido la red a identificar estas zonas. Algo similar ocurre con la profundidad que calcula la cámara, que falla en ventanas, luces y otras superficies especulares. Esto probablemente se deba al método que utilizan, la profundidad a partir

de estéreo, buscando las mismas características entre las imágenes capturadas por cada cámara, las características que se obtiene de estas superficies especulares varían en función del punto de vista por lo que dan lugar a inconsistencias en la información de profundidad.

En general, en ciertas zonas se ofrece un resultado más preciso, pero en otras pueden aparecer inconsistencias debido a la iluminación de la escena.

Capítulo 5

Trabajo futuro

Este proyecto deja abiertas algunas potenciales mejoras y pruebas que se pueden realizar para expandir el trabajo realizado.

La arquitectura de la red actualmente es la más básica que puede ofrecer una arquitectura ResNet, pero como se ha visto en la Sección 3.3, esta arquitectura ofrece varias opciones en cuanto a número de capas que pueden ayudar a mejorar el rendimiento considerablemente. Otra opción sería cambiar el tamaño del filtro a uno mayor en las primeras, o incluso concatenar más convoluciones para comprobar si mejora la precisión.

Otro posible trabajo futuro consiste en estudiar cómo se activan las neuronas, qué características las disparan, analizando así, los resultados de las convoluciones esféricas. También puede ser interesante el estudio del uso de funciones de pérdida más sofisticadas, para mejorar el aprendizaje de la red, por ejemplo ajustando la matriz de pesos de la función de pérdida esférica según cada panorama. Estas funciones se pueden mejorar añadiendo diferentes factores como el suavizado para mejorar el resultado final, además de hacer una prueba de ablación para comprobar que estas mejoras son eficaces.

Actualmente, para entrenar y ajustar los modelos generados se ha utilizado una colección de imágenes equirrectangulares de interiores, por lo que los resultados para proyecciones equirrectangulares exteriores son algo menos precisas. Por tanto, el entrenamiento con exteriores queda pendiente como línea futura. Si no encontrasen conjuntos, o para reforzarlos, se podría añadir escenas sintéticas desarrolladas en motores como Unity. Esta inclusión de datos de entornos de exteriores se podría abordar tanto en un modelo único como en un segundo modelo. Un estudio comparativo entre estas posibilidades sería otra línea interesante.

La información de profundidad es muy útil en numerosas aplicaciones (ver Sección 2), por lo que probar los resultados de este sistema en alguna de ellas podría demostrar la mejora que ofrece tener una información de profundidad más precisa.

Capítulo 6

Conclusiones

Los cuatro modelos finales superan a dos sistemas del estado el arte, por lo que se puede concluir que el uso de una colección de datos equirrectangulares y el uso de una función de pérdida esférica, ayuda a aumentar el rendimiento de estos sistemas a la hora de estimar la profundidad de imágenes equirrectangulares.

Se han investigado y explorado diferentes artículos del estado del arte que trataran problemas similares y relacionados con la estimación de profundidad y el uso de imágenes equirrectangulares, llegando a la conclusión de que una arquitectura de red basada ResNet, un codificador y decodificador, similar a la utilizada en Godard et al. [2019] sería una buena base debido al amplio uso de este tipo de arquitectura. Además, se hizo uso de diferentes técnicas especiales para tratar con imágenes equirrectangulares.

Se han probado diferentes modelos combinando las convoluciones y funciones de pérdida tradicionales con sus versiones esféricas obteniendo el mejor resultado con la combinación de convoluciones normales y función de pérdida esférica. Posteriormente, introduciendo una mejora en la función de pérdida que no aumentaba el error en zonas no definidas del mapa de profundidad original, se han mejorado los resultados frente a los modelos anteriores. El uso de las convoluciones esféricas no obtuvo mejores resultados en diferencia a otros trabajos, dando lugar a una posible opción de trabajo futuro investigando la causa de este resultado. Además, también está la posibilidad de probar otras mejoras a las funciones de pérdida o incluso otras arquitecturas de la red o aumentar el número de capas de la actual arquitectura ResNet.

En conclusión, la elaboración de este proyecto ha supuesto un trabajo de investigación del estado del arte y desarrollo considerable y se han obtenido unos resultados razonables, además de haber adquirido nuevos conocimientos.

6.1. Conocimientos adquiridos

Durante el desarrollo de este proyecto se han adquirido varios conocimientos relacionados con el ámbito del aprendizaje profundo y se ha profundizado más en las redes convolucionales y las diferentes arquitecturas que existen. También se han adquirido conocimientos sobre artículos relacionados en el estado del arte sobre la estimación de profundidad. Además, se han desarrollado capacidades de análisis y evaluación de sistemas de aprendizaje profundo más extensas.

Se han aplicado conocimientos adquiridos durante la carrera de ingeniaría informática como las redes neuronales (Inteligencia artificial), tratamiento de imágenes por ordenador (Visión por computador), incluyendo también otras más básicas como programación y matemáticas. También se han adquirido competencias sobre cómo enfrentarse a un proyecto de investigación analizando el problema y buscando soluciones relacionadas con problemas similares en el estado del arte.

Bibliografía

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. URL `https://www.tensorflow.org/`. Software available from tensorflow.org.

Iro Laina, Christian Rupprecht, Vasileios Belagiannis, Federico Tombari, and Nassir Navab. Deeper depth prediction with fully convolutional residual networks. pages 239–248, 2016.

G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.

J. D. Hunter. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3):90–95, 2007. doi: 10.1109/MCSE.2007.55.

C. Godard, Oisin Mac AodhaGabriel, and J. Brostow. Unsupervised monocular depth estimation with left-right consistency. *University College London*, 2017.

Peter Hedman, Suhib Alsisan, Richard Szeliski, and Johannes Kopf. Casual 3D Photography. 36(6):234:1–234:15, 2017.

Peter Hedman and Johannes Kopf. Instant 3D Photography. 37(4):101:1–101:12, 2018.

A. Serrano, I. Kim, z. Chen, S. DiVerdi, D. Gutierrez, A. Hertzmann, and B. Masia. Motion parallax for 360º rgbd video. *Universidad de Zaragoza*, 2019.

Ziheng Zhang, Yanyu Xu, Jingyi Yu, and Shenghua Gao. Saliency detection in 360º videos. *ShanghaiTech University, Shanghai, China*, 2018.

Benjamin Coors, Alexandru Paul Condurache, and Andreas Geiger. Spherenet: Learning spherical representations fordetection and classification in omnidirectional images. *Autonomous Vision Group, MPI for Intelligent Systems and University of Tübingen*, 2018.

Standford. CS231n convolutional neural networks for visual recognition, 2020. URL `http://cs231n.stanford.edu/index.html`.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutionalneural networks. *University of Toronto*, 2012.

Matthew D. Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. *New York University*, 2013.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale visual recognition. *University of Oxford*, 2014.

Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott E. Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. *CoRR*, abs/1409.4842, 2014. URL `http://arxiv.org/abs/1409.4842`.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *Microsoft Research*, 2013.

E. Shelhamer, J. Long, and T. Darrell. Fully convolutionalnetworks for semantic segmentation. *PAMI*, 2016.

Huangying Zhan, Ravi Garg, Chamara Saroj Weerasekera, Kejie Li, Harsh Agarwal, and Ian Reid. Unsupervised learning of monocular depth estimation and visual odometry with deep feature reconstruction. June 2018.

C. Godard, Oisin Mac AodhaGabriel, and J. Brostow. Digging into self-supervised monocular depth estimation. *University College London*, 2019.

Lista de figuras

Lista de tablas

Anexos

Anexo A

Comparativa a más resolución

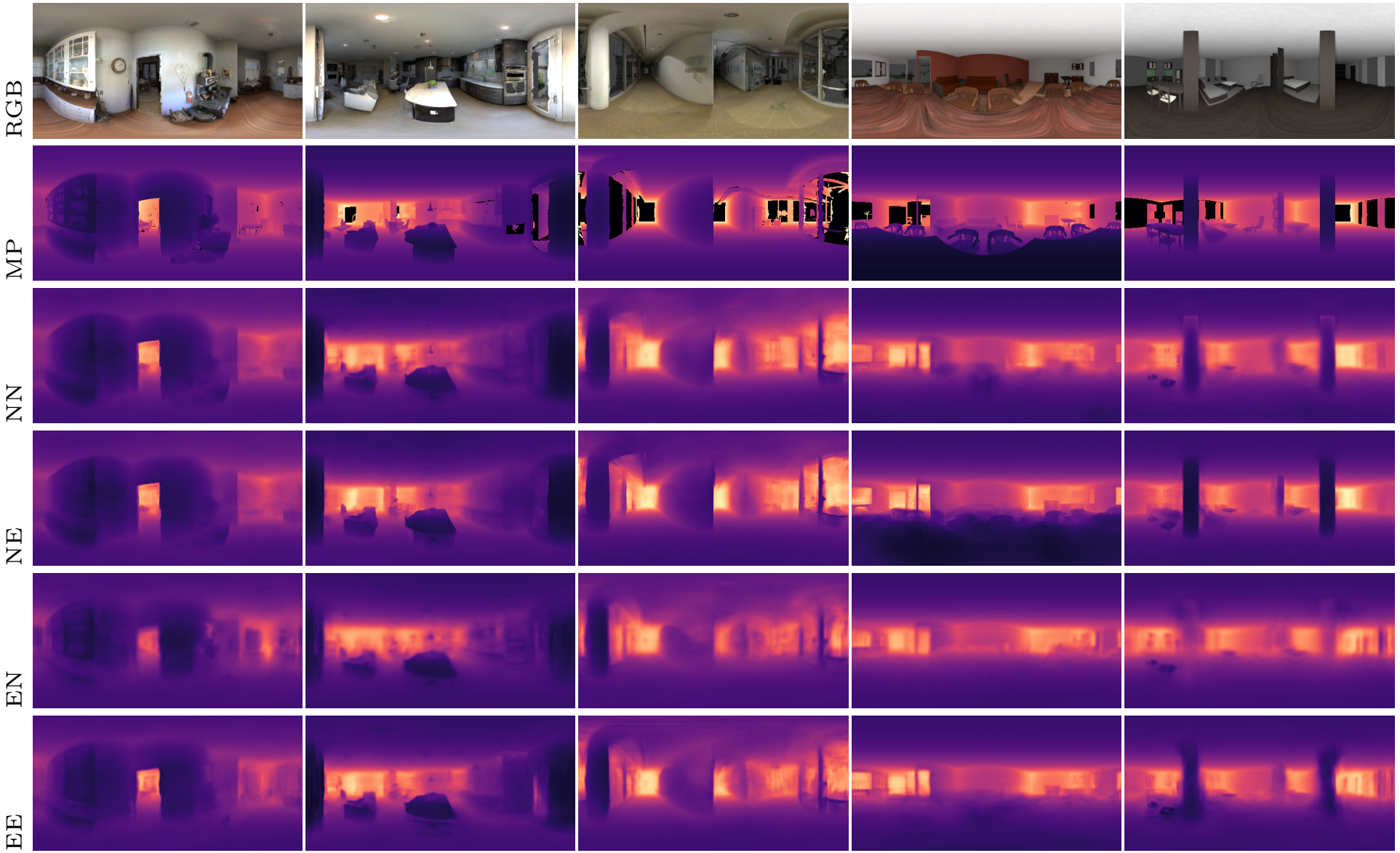


Figura A.1: Figura original en Página 26

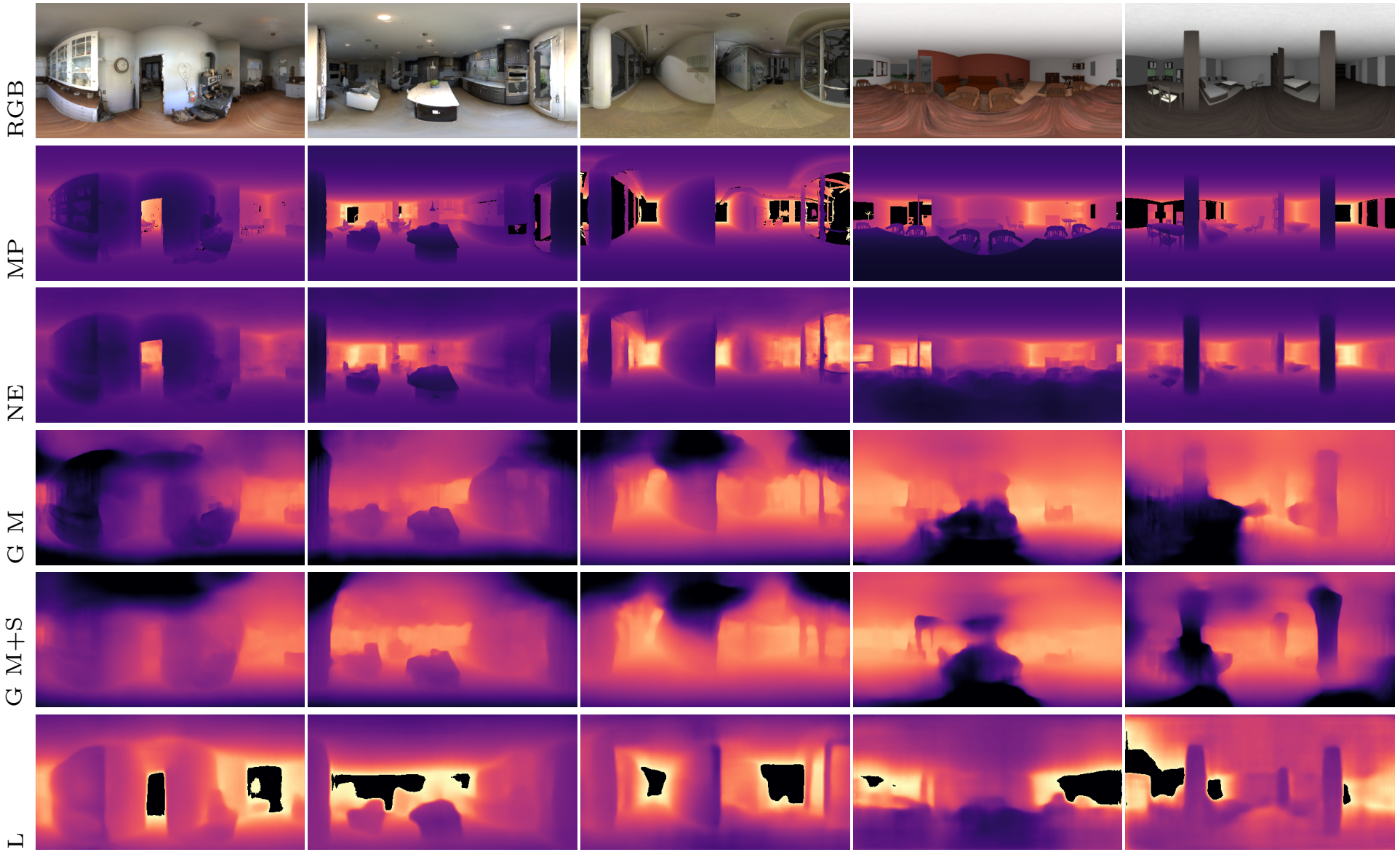


Figura A.2: Figura original en Página 26