UNIVERSIDAD POLITÉCNICA DE MADRID

ESCUELA TÉCNICA SUPERIOR DE
INGENIEROS DE TELECOMUNICACIÓN

**GRADO EN INGENIERÍA DE TECNOLOGÍAS
Y SERVICIOS DE TELECOMUNICACIÓN**

TRABAJO DE FIN DE GRADO

DISEÑO E IMPLEMENTACIÓN DE UNA SOLUCIÓN DE ESTIMACIÓN DE MAPAS DE PROFUNDIDAD A PARTIR DE VÍDEO MONOCULAR BASADA EN APRENDIZAJE PROFUNDO

JAVIER SÁNCHEZ REDONDO

2024

GRADO EN INGENIERÍA DE TECNOLOGÍAS Y SERVICIOS DE TELECOMUNICACIÓN

TRABAJO DE FIN DE GRADO

Título: Diseño e implementación de una solución de estimación de mapas de profundidad a partir de vídeo monocular basada en aprendizaje profundo

Autor: Javier Sánchez Redondo

Tutor: Javier Usón Peirón

Ponente: Julián Cabrera Quesada

Departamento: Señales, Sistemas y Radiocomunicaciones

Grupo: Grupo de Tratamiento de Imágenes

Miembros del Tribunal

Presidente:

Vocal:

Secretario:

Suplente:

Los miembros del tribunal arriba nombrados acuerdan otorgar la calificación de:

Madrid, a de , 2024

UNIVERSIDAD POLITÉCNICA DE MADRID

ESCUELA TÉCNICA SUPERIOR DE
INGENIEROS DE TELECOMUNICACIÓN

**GRADO EN INGENIERÍA DE TECNOLOGÍAS
Y SERVICIOS DE TELECOMUNICACIÓN**

TRABAJO DE FIN DE GRADO

DISEÑO E IMPLEMENTACIÓN DE UNA SOLUCIÓN DE ESTIMACIÓN DE MAPAS DE PROFUNDIDAD A PARTIR DE VÍDEO MONOCULAR BASADA EN APRENDIZAJE PROFUNDO

JAVIER SÁNCHEZ REDONDO

2024

Gracias a mis padres, por ayudarme siempre que lo he necesitado y hacer tantos sacrificios por mi.

Gracias a mi hermana, por animarme cuando he estado agobiado.

Gracias a mis abuelos, por cuidarme cada verano.

Gracias a mis amigos, por preocuparos y por todos los buenos momentos que hemos vivido juntos.

Gracias a Usón, por siempre estar dispuesto a guiarme a lo largo de todo este trabajo y ayudarme cuando las cosas no salían.

Gracias a las personas voluntarias que me han ayudado durante el desarrollo de este trabajo.

Gracias a ti Irene, por quererme, cuidarme y ser mi hogar.

Resumen

Este Trabajo de Fin de Grado se enmarca dentro de la línea de investigación en vídeo inmersivo multivista del Grupo de Tratamiento de Imágenes (GTI), en concreto del sistema *FVV Live*. *FVV Live* es un sistema de punto de vista libre que permite experiencias inmersivas realistas ya que el usuario puede navegar libremente alrededor de una escena. Este sistema depende directamente de la calidad de los datos geométricos generados sobre la escena, los cuales se representan por medio de mapas de profundidad.

El objetivo principal de este proyecto es utilizar las innovaciones de la estimación de profundidad monocular para solucionar los desafíos que presenta el *FVV Live*, especialmente en la generación precisa de datos geométricos. Este enfoque pretende superar los límites de la captura de profundidad aprovechando los datos volumétricos del *FVV Live* y la eficiencia de los modelos de estimación de profundidad a partir de una sola imagen. Al combinar estos datos con los de estimación de profundidad monocular, se pretende mejorar significativamente la calidad y precisión de las reconstrucciones 3D, resolviendo así los problemas que podían encontrarse previamente.

Además del objetivo anterior, en este trabajo se abordarán otros objetivos como el aprendizaje y posterior uso de distintas metodologías para trabajar con redes neuronales y aprendizaje profundo. También se usarán herramientas para la captura y el procesado de imágenes que posteriormente serán utilizadas en el *FVV Live*.

Abstract

This Final Degree Project is part of the immersive multiview video research of Grupo de Tratamiento de Imágenes (GTI), more specifically, the *FVV Live* system. *FVV Live* is a free viewpoint system that allows for immersive virtual experiences, as the user can freely navigate around a scene. This system directly depends on the quality of the geometric data generated on the scene, which are represented through depth maps.

The main aim of this project is to use the innovations in monocular depth estimation to solve the challenges presented by the *FVV Live*, especially in the accurate generation of the geometric data. This approach aims to overcome the limitations of depth capture taking advantage of the volumetric data of the *FVV Live* and the efficiency of depth estimation models from a single image. By combining this data with monocular depth estimation data, the aim is to significantly improve the quality and accuracy of 3D reconstructions, thus solving the problems that could be encountered previously.

In addition to the previous objective, this project will also address other goals such as learning and subsequent use of various methodologies to work with neural networks and deep learning. Tools will also be used to capture and process images that will later be used in the *FVV Live*.

Palabras clave

Profundidad, Aprendizaje Profundo, Monocular, Punto de Vista Libre

Key words

Depth, Deep Learning, Monocular, Free Viewpoint

Índice

Índice de figuras

Índice de tablas

Glosario

AMCW	Amplitude Modulated Continuous Wave
CNN	Convolutional Neural Network
FVV	Free Viwepoint Video
FoV	Field of View
GPU	Graphics Processing Unit
IMU	Inertial Measurement Unit
IR	Infra Rojo
LED	Light Emitting Diode
MDE	Monocular Depth Estimation
NLP	Natural Language Processing
PNG	Portable Network Graphics
RGB	Red, Green, Blue
RNN	Recurrent Neural Network
RAM	Random Access Memory
ToF	Time of Flight
USB	Universal Serial Bus
YAML	YAML Ain’t Markup Language

1. Introducción

1.1. Motivación

Este trabajo y todos los desarrollos que se han realizado en el mismo, se enmarcan dentro de la línea de investigación destinada al sistema de punto de vista libre *FVV Live* del Grupo de Tratamiento de Imágenes (GTI). El sistema *FVV Live*, cuyo ejemplo de montaje se puede ver en la Figura 1, es un sistema que permite al espectador navegar libremente alrededor de una escena desde distintos puntos de vista, creando experiencias interactivas y dinámicas, y destacando frente a otros sistemas de punto de vista libre por su capacidad de trabajar en tiempo real y el uso reducido de hardware de consumo.

Figura 1: Ejemplo de montaje del sistema *FVV Live*

Para poder lograr ver la escena desde distintos puntos de vista, se ha de capturar la misma con varias cámaras apuntando desde distintos ángulos. Actualmente, el sistema *FVV Live* trabaja con cámaras estéreo que generan información de profundidad de la escena y, en función de esa profundidad, se realiza la síntesis de la vista virtual. La calidad de las imágenes de profundidad determinará la calidad de la escena virtual creada.

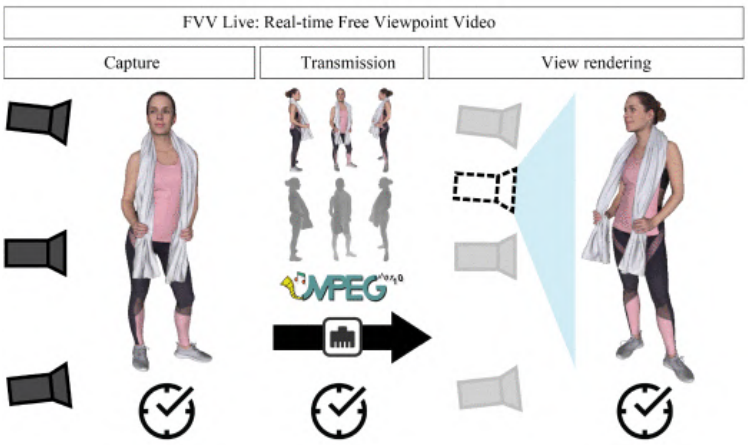


Figura 2: Esquema del sistema *FVV Live* [1]

La búsqueda de esta calidad en las imágenes de profundidad obliga a que este sistema trabaje con una codificación sin pérdidas, más concretamente, mono-canal de 12 bits por píxel, generando un gran volumen de datos a transmitir, provocando así limitaciones en el mismo. Además, se puede dividir el sistema en varios módulos que pueden verse en la Figura 2, como son: el módulo de captura, que se encarga de procesar y transmitir las imágenes introducidas en el sistema mediante un conjunto de cámaras, y un módulo de síntesis, que toma las imágenes obtenidas en el módulo de captura y genera un vídeo de salida.

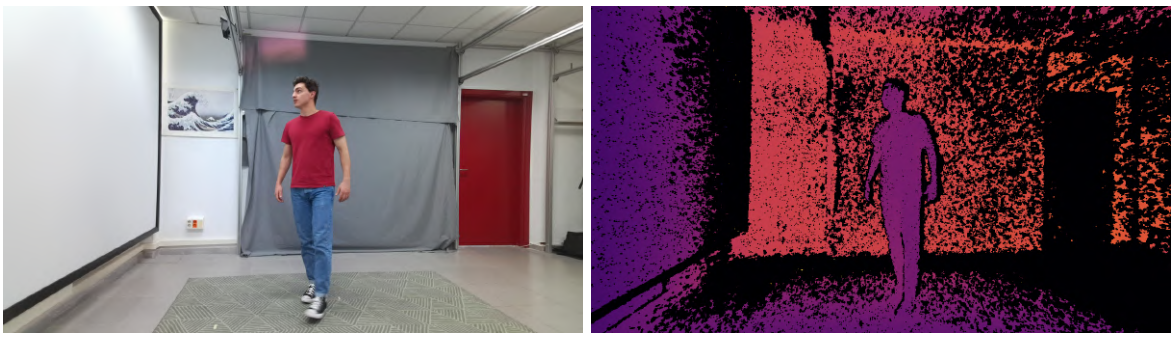
La motivación de este proyecto es utilizar las innovaciones que está aportando el aprendizaje profundo para mejorar la calidad de las imágenes de profundidad realizadas en el módulo de captura del sistema *FVV Live*, y con ello mejorar la calidad de las vistas virtuales sintetizadas. Para ello, se utilizarán modelos de estimación de profundidad monocular, ya que se ha observado que estos modelos están dando muy buenos resultados a la hora de mejorar la calidad y precisión de las imágenes de profundidad, además de permitir el uso de menor número de cámaras.

1.2. Objetivos

El objetivo principal de este trabajo es el desarrollo de un sistema que mejore, mediante modelos de estimación de profundidad monocular y técnicas de aprendizaje profundo, la calidad de las imágenes de profundidad generadas por las cámaras utilizadas para la realización de este proyecto, consiguiendo así que al introducir estas imágenes en el sistema *FVV Live*, se optimice la calidad del espacio virtual con el que se trabaja.

Para alcanzar este objetivo, anteriormente se tendrán que haber cumplido dos sub-objetivos. En primer lugar, la captura de distintas secuencias que registren tanto la información de color como la de profundidad asociada, además de realizar una calibración previa para corregir la geometría de dichas imágenes y obtener la posición relativa entre ellas. En segundo lugar, será necesaria la integración de estas secuencias capturadas con un modelo de estimación de profundidad monocular del estado del arte.

Para lograr el primero de los sub-objetivos mencionados, se utilizarán unas cámaras que capturan tanto las imágenes RGB como las imágenes de profundidad (Figura 3) de la escena basándose en una técnica de "tiempo de vuelo", que medirá las profundidades y distancias de la escena con gran precisión, obteniendo mejores resultados que los que se pueden conseguir con dispositivos estéreo convencionales.



(a) Imagen de color (b) Imagen de profundidad

Figura 3: Imagen de color y su profundidad asociada

El segundo de los sub-objetivos comenzará por estudiar y seleccionar un modelo de estimación de profundidad monocular del estado del arte y procesar con dicho modelo las distintas secuencias grabadas. Para ello, se tomarán las imágenes de profundidad capturadas por las cámaras y mediante una técnica de regresión lineal se tratará de mejorar la calidad de dichas imágenes de profundidad.

Finalmente, se integrarán los resultados obtenidos con el sistema *FVV Live*, realizando pruebas para comprobar las mejorías obtenidas en la visualización de las escenas finales y comparándolas con las obtenidas con las cámaras. Además, se dará una gran importancia a que todos los desarrollos realizados durante este proyecto estén los más automatizados posibles, facilitando así la incorporación de nuevas secuencias, más cámaras y nuevos modelos de estimación de profundidad monocular en trabajos futuros.

2. Estado del arte

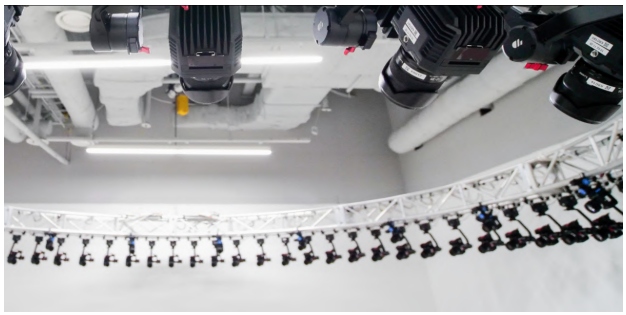
2.1. Sistemas de vídeo de punto de vista libre

El vídeo de punto de vista libre, FVV por sus siglas en inglés, es una técnica en la cual se permite al espectador navegar libremente alrededor de una escena desde cualquier punto de vista arbitrario.

Los sistemas de vídeo de punto de vista libre han destacado por su interactividad, ya que el punto de vista es elegido por el usuario y no por el creador de contenido, como se había hecho hasta ahora [2]. Estos sistemas suelen utilizar un número elevado de cámaras para conseguir capturar la escena desde varios ángulos, asegurando que hay una cobertura correcta de la escena.

Esta tecnología se ha desarrollado en varios campos, pero actualmente, se puede destacar su uso en el mundo del deporte. Al ser entornos profesionales, se requiere de un número elevado de cámaras de una calidad muy alta y operaciones muy complejas. Ejemplos de aplicación de estos sistemas son 4DReplay [3], que se ha utilizado en 1.440 eventos en todo el mundo y que actualmente se usa en la NLCS y se retransmite a través de la aplicación móvil de Fox Sports. Otro ejemplo es el de Canon Global Free Viewpoint Video System [4], que destaca por su uso en la Copa del Mundo de Rugby de 2019, y finalmente, cabe mencionar el Intel True View [5], que ha transformado totalmente las transmisiones de la NFL y la NBA, obteniendo perspectivas nunca antes vistas.

El sistema 4DReplay (Figura 4) destaca por su capacidad de crear vídeos en un rango de tiempo de 2,5 a 5 segundos gracias a su tecnología de sincronización temporal. Además, cuenta con la tecnología 4DLive para transmitir eventos en vivo, capturando con un número muy elevado de cámaras todo el contenido en calidad 4K.



(a) Ejemplo de montaje del sistema

(b) Cuadro de vídeo generado por el sistema

Figura 4: Sistema 4DReplay

El sistema Free Viewpoint Video de Canon Global (Figura 5) ha llegado a utilizar hasta 125 cámaras 4K, que operan a 60 cuadros por segundo y que están modificadas para capturar vídeo volumétrico en eventos de distintos tipos. No obstante, dependiendo del deporte que se vaya a retransmitir, el número de cámaras podría variar, incluso ser

mucho menor, facilitando la velocidad de procesado.

(a) Ejemplo de montaje del sistema

(b) Cuadro de vídeo generado por el sistema

Figura 5: Sistema Free Viewpoint de Canon Global [6]

El sistema Intel True View (Figura 6) utiliza más de 30 cámaras de ultra alta resolución para capturar datos volumétricos de la escena. Estas cámaras se colocan estratégicamente alrededor del lugar donde esté ocurriendo el evento, de tal manera que se tenga una cobertura completa del mismo y se puedan generar repeticiones inmersivas.

(a) Ejemplo de montaje del sistema

(b) Cuadro de vídeo generado por el sistema

Figura 6: Sistema Intel True View

2.1.1. FVV Live

Además de los sistemas vistos previamente, se ha desarrollado el sistema *FVV Live* [7], el cual presenta una solución más económica y sencilla, que utiliza hardware de consumo y que destaca principalmente por la capacidad de trabajar en tiempo real, con un retardo extremo a extremo de 250 milisegundos. Las cámaras utilizadas son las Stereolabs ZED, que además de capturar vídeo, son capaces de estimar la profundidad de la escena. Además, el número de cámaras a usar es escalable, pudiendo aumentar su número en función del requerimiento de la escena [1].

Por otro lado, gracias a las condiciones del sistema *FVV Live*, que transmite grandes cantidades de datos en tiempo real y sin retrasos significativos, permiten al sistema

aprovecharse de tecnologías innovadoras como pueden ser el 5G o la computación en la nube [8].

(a) Ejemplo de montaje del sistema

(b) Cuadro de vídeo generado por el sistema

Figura 7: Sistema *FVV Live*

2.2. Dispositivos de captura de mapas de profundidad

Cuando se habla de las nuevas innovaciones introducidas en el mundo de la visión por ordenador o el vídeo inmersivo, resulta muy útil poder trabajar con entornos 3D, que ayuden a los ordenadores a ver y entender el mundo. Para hacer esto posible, se usan las imágenes de profundidad. Con esta información, se consigue convertir una foto normal plana en una estructura 3D completa.

Los fabricantes de dispositivos de captura de profundidad han lanzado al mercado varias opciones que proporcionan datos de profundidad en tiempo real sincronizados con imágenes en color. Durante este proyecto, se va a utilizar uno de estos dispositivos, que se introducirá a continuación.

2.2.1. Cámaras Azure Kinect de Microsoft

Las cámaras Azure Kinect de Microsoft [9] nacen como una actualización de las cámaras para videojuegos Xbox, ya que sus fabricantes observaron que no solo se usaban para jugar, si no para aplicaciones en otros campos, como la visión por ordenador.

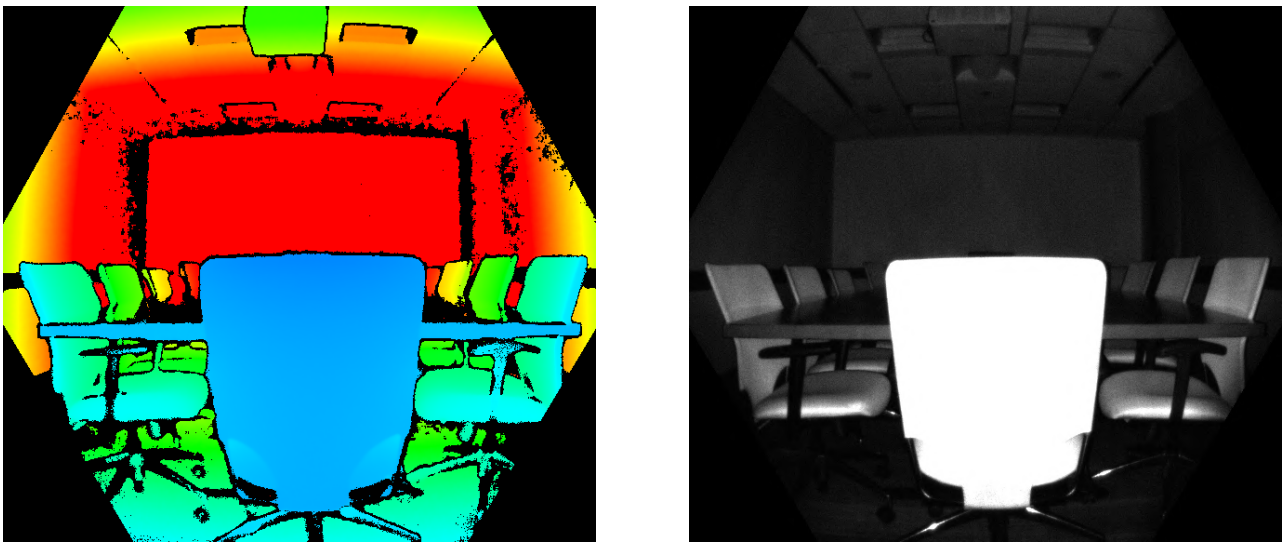
Estas cámaras están diseñadas para ser versátiles, ya que tienen numerosos modos y opciones de montaje, facilitando su uso en múltiples entornos. Además, cuenta con los sensores más avanzados del sector en el campo de la inteligencia artificial, la visión y la voz, permitiendo su uso a empresas de distintos sectores como pueden ser la salud o los medios de comunicación. Se destaca finalmente su software controlador de código abierto y su integración con los servicios en la nube de Azure.

En cuanto a sus especificaciones, cuenta con una cámara de profundidad de 1MP, además de una cámara de 12MP RGB para detalles de alta definición y 7 micrófonos para una captura de audio precisa. Además de lo anterior mencionado, también cuenta

con un sensor de orientación IMU con acelerómetro y giroscopio para orientar la cámara y sincronizar varios dispositivos.

Figura 8: Cámara Azure Kinect y sus sensores [9]

Las cámaras de profundidad de Azure Kinect [10] se rigen por el principio de Tiempo de Vuelo (ToF), donde la cámara ilumina la escena utilizando ondas continuas moduladas en amplitud (AMCW). Posteriormente, la cámara, mediante un transmisor IR y un receptor, registra el tiempo que tarda la luz en viajar a la escena y volver. Una vez obtenidos estos datos, son utilizados para generar un mapa de profundidad. Además, junto al mapa de profundidad, también se obtendrá una lectura de IR de la escena.



(a) Imagen de profundidad (b) Lectura IR asociada

Figura 9: Imágenes obtenidas por una cámara Azure Kinect [10]

2.3. Redes neuronales y aprendizaje profundo

Las redes neuronales se inspiran principalmente en actuar tal y como lo haría el cerebro humano, que difiere significativamente de cómo actúan las computadoras tradicionales. A diferencia de estas últimas, que operan de manera secuencial, el cerebro humano está preparado para realizar tareas paralelamente, lo que permite realizar varias a la vez. Por tanto, una red neuronal artificial es aquella que tiene capacidad para emular el procesamiento de la información de la misma manera que lo haría el sistema nervioso biológico. [11].

Las redes neuronales artificiales se utilizan ampliamente en el aprendizaje profundo en distintas actividades como la visión artificial, el reconocimiento de voz, la clasificación de imágenes e incluso previsiones de demanda de energía, entre muchas otras.

Como se ha mencionado, dadas su arquitectura y fundamentos, las redes neuronales artificiales comparten cualidades similares con las del cerebro humano, incluyendo la capacidad de aprender de experiencias anteriores, extrapolar a situaciones nuevas conocimientos de situaciones ya conocidas o distinguir la información importante de la que no lo es. Es por esto, que podemos destacar principalmente varias ventajas, como son: el aprendizaje adaptativo, la tolerancia a errores, la organización automática, los cómputos en tiempo real y la sencilla integración en las tecnologías actuales [12].

En cuanto a los elementos que componen una red, se pueden destacar tres niveles o capas principales, como se ve en la Figura 10:

- Capa de entrada, aquella por donde se reciben los datos de entrada del problema [13] y se encarga de distribuir los datos de entrada a través de las capas siguientes.
- Capas ocultas, situadas entre la capa de entrada y la capa de salida. Pueden ser una o varias, y su número dependerá de la complejidad del problema a resolver [14]. Estas capas se encargan de analizar la salida de la capa anterior, donde mediante funciones, aplican transformaciones que la procesan aún más, y se pasa a la siguiente capa [15].
- Capa de salida, genera la salida del modelo. La salida puede ser diferente en función del tipo de ejercicio. Se puede tener un único nodo, utilizado en tareas de regresión o múltiples nodos, usado por ejemplo en clasificación.

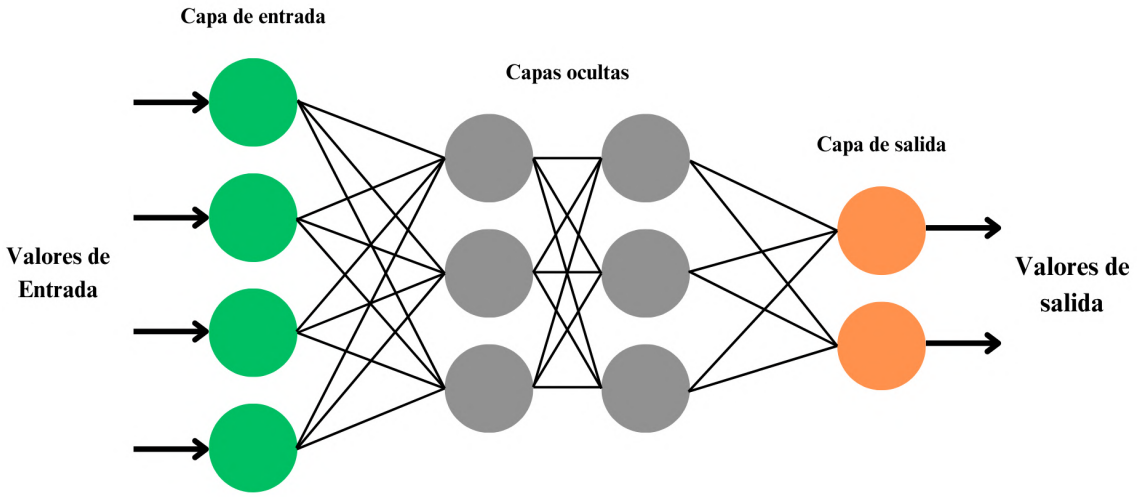


Figura 10: Ejemplo de una red neuronal totalmente conectada

Las redes neuronales se clasifican, no solo por la arquitectura que define su estructura interna, sino también por el tipo de aprendizaje profundo que adoptan para procesar y aprender de los datos. A continuación, se verán en detalle las clasificaciones que dan forma a sistemas que son capaces desde reconocer patrones complejos hasta generar respuestas a problemas de distintas características.

2.3.1. Tipos de redes neuronales

- Red neuronal monocapa o perceptrón simple: es el tipo de red neuronal artificial más simple que existe [16]. Como se puede ver en la Figura 11, está compuesta por una única capa de neuronas, sin capas ocultas entre la entrada y la salida.

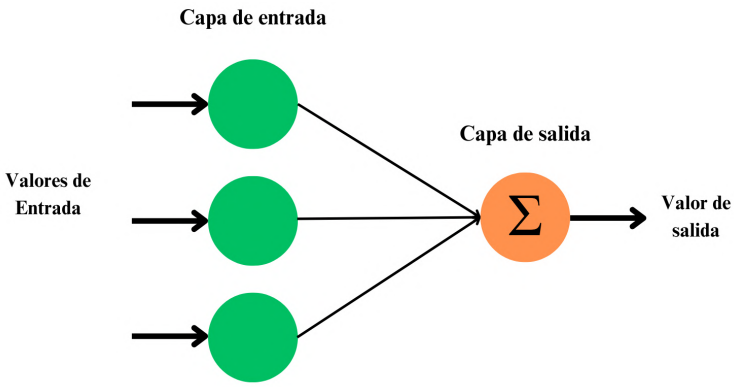


Figura 11: Ejemplo de una red neuronal monocapa o perceptrón simple

- Red neuronal multicapa o perceptrón multicapa: está organizada en capas y consta de una capa de entrada, una capa de salida y una capa o grupo de capas ocultas entre la de entrada y la de salida, como se puede apreciar en la Figura 10. Estas capas pueden estar completamente conectadas entre sí o parcialmente conectadas.

- Redes neuronales recurrentes (RNN): son un tipo avanzado de redes neuronales capaces de procesar secuencias de datos en cualquier dirección y hacer bucles entre distintas capas, memorizando información temporalmente para usarse en otro momento [17]. Este tipo de red, tal y como se ve en la Figura 12, tiene distintas aplicaciones como pueden ser el procesado de series temporales, y la resolución de problemas con datos de texto o audio [18].

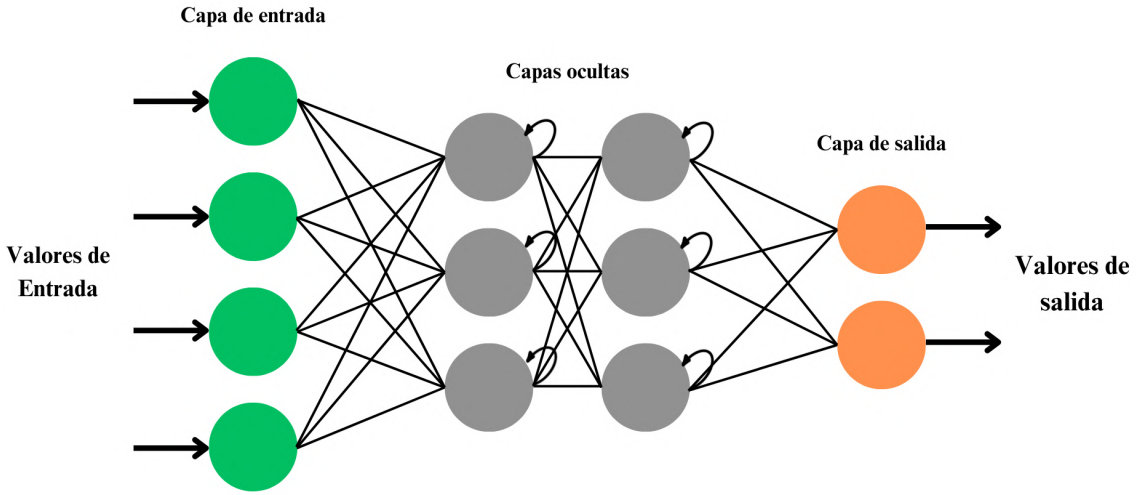


Figura 12: Ejemplo de una red neuronal recurrente

- Redes neuronales convolucionales (CNN): son un tipo de red neuronal que utiliza capas especiales (Figura 13) para identificar y clasificar patrones como formas y texturas. Estas redes emplean filtros en sus capas convolucionales para extraer características relevantes de los datos visuales [18]. Son muy eficaces en el análisis de imágenes y tareas de visión por computadora, como la segmentación de imágenes e identificación de objetos y es la principalmente utilizada en la estimación de profundidad monocular [19].

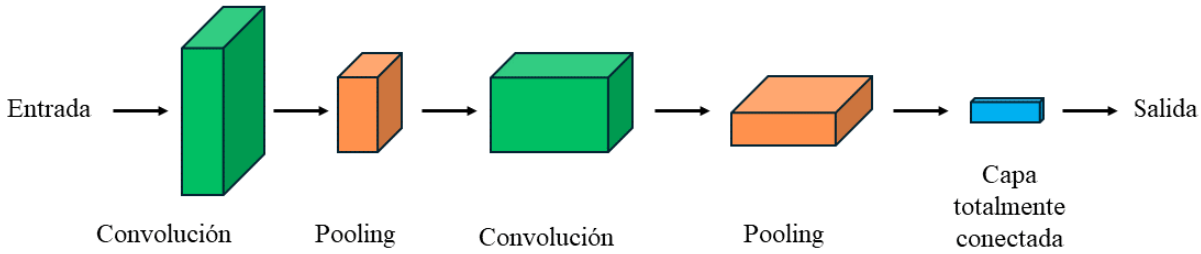


Figura 13: Ejemplo de una red neuronal convolucional

- *Transformers*: este tipo de red neural, ha destacado frente a las redes neuronales recurrentes (RNN) y las redes neuronales convolucionales (CNN). Esto se ha logrado debido a que los *transformers* agregan un mecanismo de atención múltiple [20].

A diferencia de otras redes neuronales, que procesan sus entradas de una en una, los *transformers* pueden operar sobre la distintas partes de la secuencia de manera simultánea gracias a estos mecanismos de autoatención, haciendo que sea una red más eficiente y rápida de entrenar [21]. Aunque en un principio, se utilizaban en tareas de procesado de lenguaje natural (NLP), últimamente ha ganado importancia en el campo de la visión por computadora [22].

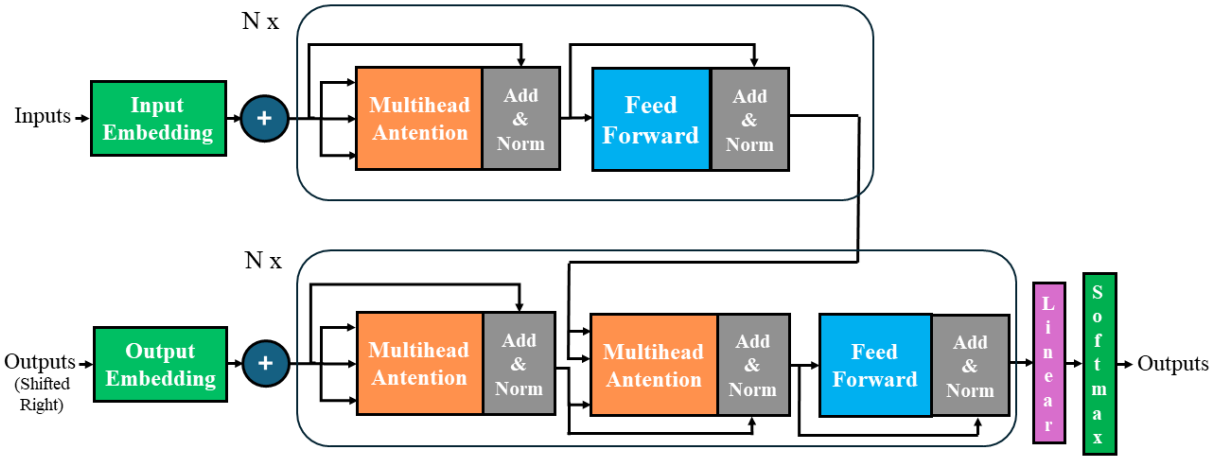


Figura 14: Ejemplo de un *transformer*

2.3.2. Tipos de aprendizaje profundo

- Aprendizaje supervisado: en este tipo de aprendizaje, se proporcionan las entradas y salidas correspondientes al algoritmo de aprendizaje. Esto se hace con el objetivo de ajustar los parámetros del modelo para que sus predicciones se alineen lo máximo posible con las salidas objetivo. Este método implica dos tipos principales de tareas: la clasificación, que permite al modelo determinar la categoría correcta de una entrada de entre un conjunto predefinido, y la regresión, donde el modelo predice valores continuos a partir de las entradas dadas. A través de un entrenamiento exhaustivo con pares de datos etiquetados, el modelo desarrolla la capacidad de generar predicciones precisas para entradas nuevas. Las aplicaciones de este tipo de entrenamiento son varias, de entre los que se destaca, por ejemplo, la detección de objetos [22].
- Aprendizaje no supervisado: se refiere al proceso de examinar y analizar conjuntos de datos sin etiquetar, con el objetivo de descubrir patrones ocultos, distribuciones peculiares o consistencias [22]. Esta metodología se enfoca en explorar los datos disponibles para identificar estructuras naturales o agrupaciones intrínsecas sin una referencia especificada.
- Aprendizaje por refuerzo: en este tipo de aprendizaje, el agente selecciona distintas acciones basadas en el estado actual del entorno optimizando así los resultados

futuros [23]. Se pueden destacar tres elementos principales en este aprendizaje, como son: el agente, que se encarga de tomar decisiones, el intérprete, que informa tras una decisión del agente del estado actual del entorno, y el entorno, que son las circunstancias con las que el agente interactúa. [24]

2.4. Estimación de profundidad monocular

La estimación de profundidad es un proceso fundamental en la visión por computadora, que se encarga de calcular la distancia de cada píxel de un objeto en una escena con respecto a la cámara [25], relacionando los píxeles de una imagen de color, con la distancia a la que se encuentran los píxeles en el mundo real. Consiste, por tanto, en facilitar el entendimiento de escenas tridimensionales a partir de imágenes bidimensionales [26].

Los métodos de estimación de profundidad que se suelen utilizar en sensores comerciales, emplean tecnología activa que hacen que la generación de mapas de profundidad realmente precisos tengan un alto coste de recursos, tanto a nivel material como a nivel computacional. Algunas de estas tecnologías activas son el uso de láseres y luces que analizan como rebota la luz en los objetos de una escena, generando así mapas de profundidad densos.

Tradicionalmente, para estimar la profundidad de las imágenes, se ha dependido de cámaras estéreo. Estos sistemas trabajan calculando la disparidad entre dos imágenes bidimensionales que son capturadas desde dos puntos de vista distintos, imitando comúnmente la disposición de los ojos. Posteriormente se aplican técnicas de coincidencia y calibración para producir los mapas de profundidad. Este enfoque es por tanto más costoso ya que se parte de necesitar dos cámaras, además de presentar dificultades computacionales [27].

Debido a los costes que tiene la estimación de profundidad tradicional, se ha avanzado hacia una técnica más novedosa y en pleno auge: la estimación de profundidad monocular. Su investigación continúa expandiendo a pasos agigantados las fronteras de este campo, permitiendo obtener nuevos resultados que abren las puertas a nuevas aplicaciones.

La estimación de profundidad monocular, MDE por sus siglas en inglés, es un proceso que captura una imagen o secuencia de vídeo sin la necesidad de utilizar equipos o técnicas complicadas. Únicamente se necesita una cámara y, debido a esto, está empezando a crecer su interés en distintos tipos de aplicaciones que abarcan desde la conducción autónoma hasta la robótica y medicina [28].

No obstante, la estimación de profundidad de una única imagen es un desafío complejo a nivel geométrico en el mundo de la visión por computadora, necesitando una comprensión profunda de la escena [29]. Por tanto, es evidente que el desarrollo de las técnicas de aprendizaje profundo han hecho progresar esta área de manera sobresalien-

te [30].

Para la realización de este documento, se han examinado los métodos más recientes y avanzados de estimación de profundidad monocular, destacando la evolución y el estado actual de esta tecnología. A través de una meticulosa investigación, se han podido obtener los modelos que están liderando esta rama de la estimación de profundidad. Entre ellos, se encuentran MiDas v3.1, ZoeDepth, PatchFusion y Depth Anything. Cada uno de estos modelos ofrece distintos enfoques para afrontar las limitaciones que conlleva la estimación de profundidad.

A continuación, se hará un estudio detallado de las características distintivas, las metodologías, fortalezas e innovaciones de cada uno de los modelos mencionados anteriormente. Además, se hará una comparativa con las imágenes de la Figura 15 en los distintos modelos del estado del arte de estimación de profundidad monocular que se van a estudiar.

(a) Imagen de color 1

(b) Imagen de color 2

Figura 15: Imágenes utilizadas en los modelos del estado del arte

2.4.1. Midas v3.1

Midas v3.1 [31] representa un avance significativo con respecto a sus modelos anteriores, que solo usaban *transformers*. Se puede ver un ejemplo en la Figura 16

Este modelo emplea una arquitectura de codificador-decodificador para la predicción de profundidad, donde el codificador se basa en redes de clasificación de imágenes y el decodificador está especializado en profundidad. De manera innovadora, MiDaS v3.1 integra codificadores basados en *transformers* como BEiT [32], Swin [33] y Next-ViT [34], que mejoran la estimación de profundidad al procesar características globales de la imagen.

Para fortalecer su generalización, el modelo se entrena con una variedad de datasets. Esto es esencial para asegurar un rendimiento consistente bajo diferentes condiciones visuales. Además, utiliza una función de pérdida que es invariante a cambios de escala y desplazamiento, abordando así las diferencias en las etiquetas de profundidad.

Se ha visto, que con las innovaciones mencionadas, MiDas v3.1 ha conseguido mejo-

rar significativamente la estimación de profundidad con respecto a su versión anterior, además de optimizarse los modelos para un uso en tiempo real. Finalmente, cabe destacar que está diseñado de tal manera que la integración de versiones futuras será muy sencilla.

(a) Imagen de color 1

(b) Imagen de color 2

(c) Imagen de profundidad 1

(d) Imagen de profundidad 2

Figura 16: Imágenes de profundidad generadas por Midas v3.1 [35]

2.4.2. ZoeDepth

ZoeDepth [36] es un modelo que aborda la estimación de profundidad monocular mediante la combinación de la estimación de profundidad relativa y métrica en un único marco. Este enfoque no solo logra una generalización superior sino que también mantiene la precisión métrica en diversas aplicaciones.

El entrenamiento que realiza ZoeDepth sigue una estrategia de entrenamiento que podemos dividir en dos fases: inicialmente, se pre-entrena utilizando distintos conjuntos de datos que tratan la profundidad relativa. Posteriormente, se realiza un ajuste fino en conjuntos de datos específicos para optimizar la precisión de la profundidad métrica.

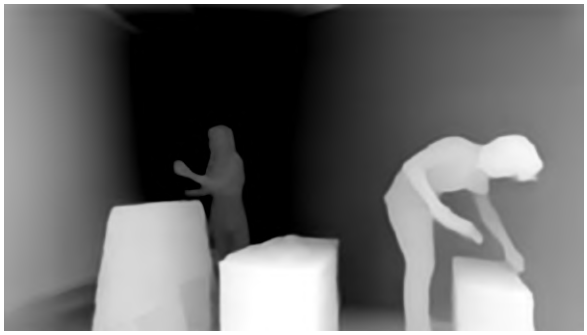
Además, introduce una innovación con su módulo de *bins* métricos, que se encarga de ajustar dinámicamente los centros de los *bins* de profundidad en la fase de decodificación, consiguiendo una mejora significativa en la precisión de las estimaciones.

Una característica distintiva de ZoeDepth es su alta configurabilidad, lo que permite adaptar el modelo a una gran cantidad de escenarios. Además, el modelo, visto en la Figura 17, ha demostrado su capacidad para efectuar transferencias sin experiencia pre-

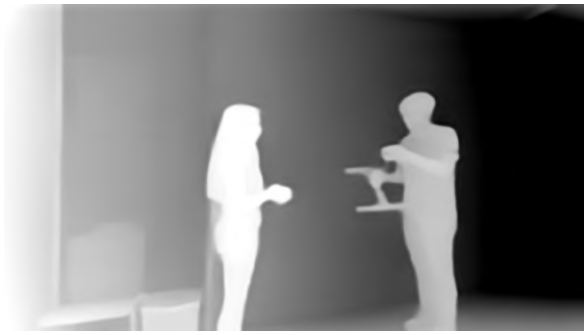
via entre datos desconocidos anteriormente, tanto en espacios internos como externos, gracias a su sólida estructura de entrenamiento y su habilidad para generalizar.

(a) Imagen de color 1

(b) Imagen de color 2



(c) Imagen de profundidad 1



(d) Imagen de profundidad 2

Figura 17: Imágenes de profundidad generadas por ZoeDepth [36]

2.4.3. PatchFusion

PatchFusion [37], que ha demostrado una innovación significativa en la estimación de profundidad en imágenes de alta resolución, utiliza una red de fusión de parches que combina predicciones consistentes a nivel global, con predicciones localmente más detalladas, pero inicialmente inconsistentes. Este proceso se ve reforzado por un módulo Global-a-Local (G2L) que mejora el contexto dentro de la red de fusión de parches mencionada, eliminando así la necesidad de emplear estrategias para la selección de parches. Este enfoque se complementa con un entrenamiento e interpretación consciente de la consistencia, asegurando así la uniformidad en la superposición de parches y eliminando una necesidad de post-procesado.

Debido a todas las innovadoras características mencionadas, PatchFusion destaca frente a otros modelos gracias a su gran capacidad para tratar con imágenes de alta resolución, mejorando principalmente la precisión en los bordes y detalles finos de la imagen, como se ve en la Figura 18.

No obstante, a pesar de sus avances, PatchFusion todavía enfrenta distintos desafíos, pues tiene limitaciones, como la eficiencia computacional. Además, destacan una falta de datos de alta resolución para poder realizar mejores entrenamientos de cara a mejorar

todavía más el rendimiento.

(a) Imagen de color 1

(b) Imagen de color 2

(c) Imagen de profundidad 1

(d) Imagen de profundidad 2

Figura 18: Imágenes de profundidad generadas por PatchFusion [37]

2.4.4. Depth Anything

Depth Anything [38] presenta una innovadora metodología para la estimación de profundidad monocular, de la se destaca principalmente el uso de datos no etiquetados a gran escala, ya que el modelo está entrenado con aproximadamente 62 millones de imágenes no etiquetadas y de distintos tipos, permitiendo mejorar la capacidad de generalización del modelo. No obstante, este modelo no solo aprende de imágenes no etiquetadas, si no también de imágenes pseudo-etiquetadas y también de imágenes etiquetadas (aproximadamente 1,5 millones).

El modelo, cuyo ejemplo se muestra en la Figura 19, emplea una combinación de estrategias para manejar los datos a tratar, partiendo de aumentar considerablemente los datos, impulsando al modelo a buscar conocimiento visual adicional y así poder adquirir representaciones más robustas. Además, incorpora una supervisión adicional para que el modelo aproveche los conocimientos adquiridos por codificadores pre-entrenados como DINOv2 [39], mejorando la comprensión semántica de la escena que se vaya a procesar.

En cuanto a la generalización mencionada anteriormente, Depth Anything ha demostrado que su capacidad con datos capturados públicamente y otras imágenes aleatorias es impresionante. Además, con el ajuste de profundidad que realiza utilizando conjuntos de datos como KITTI [40] y NYUv2 [41], este modelo establece un estado del arte con res-

pecto a otros modelos estudiados anteriormente en cuando a estimación de profundidad monocular se refiere.

(a) Imagen de color 1

(b) Imagen de color 2

(c) Imagen de profundidad 1

(d) Imagen de profundidad 2

Figura 19: Imágenes de profundidad generadas por Depth Anything [38]

3. Captura de contenido para FVV Live

3.1. Introducción

Como se ha visto en la Subsección 2.4, Depth Anything ofrece una muy buena calidad en sus resultados, no obstante, las imágenes de profundidad que este modelo genera no tienen escala, es decir, no existe una relación entre las dimensiones de la imagen de profundidad generada por el modelo y las dimensiones reales de la escena capturada. Es debido a esto, que se realizará la captura de imágenes de profundidad e imágenes de color multivista con dos objetivos principales: mejorar las imágenes de profundidad combinando las imágenes profundidad obtenidas durante la captura, con las imágenes de profundidad generadas por Depth Anything y realizar la síntesis de vistas virtuales con el *FVV Live* a partir de las imágenes de profundidad mejoradas en el primer objetivo. Además, se evaluarán las mejoras de la profundidad comparando tanto las imágenes de profundidad como las vistas virtuales sintetizadas con el *FVV Live*.

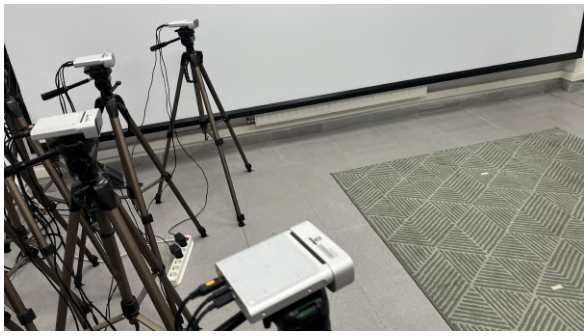
Para lograr esto, se va a realizar una primera estimación de profundidad utilizando las cámaras Azure Kinect, que son capaces de obtener la geometría de una escena en forma de imagen de profundidad utilizando su método activo de tiempo de vuelto (ToF), como se ha explicado en la Subsubsección 2.2.1.

Durante este proceso, ilustrado en la Figura 20, se partirá de colocar adecuadamente las cámaras. Estas cámaras deben estar colocadas de manera que cubran correctamente la escena que se quiera capturar, maximizando así la calidad de los datos obtenidos. Posteriormente, se utilizará un software que permitirá la calibración de las mismas, para que las imágenes capturadas sean precisas y estén alineadas. Una vez realizada la calibración, se realizará la captura de la escena. A continuación, se realizará la extracción de los cuadros obtenidos durante la captura, que dará como resultado imágenes de color e imágenes de profundidad de las secuencias capturadas, además de aplicar la codificación de vídeo utilizada por el sistema *FVV Live*. Finalmente, se sintetizarán vistas virtuales sobre el contenido grabado.

Figura 20: Proceso para la captura de imágenes con las Azure Kinect

3.2. Hardware de captura

Para la grabación de las secuencias se han utilizado cuatro cámaras Azure Kinect, que se dispusieron a unos setenta centímetros unas de otras. Además, se colocaron en forma de arco, estando las dos cámaras centrales unos cuarenta centímetros más atrás que las cámaras de las esquinas, permitiendo así cubrir toda la escena desde cuatro puntos de vista diferentes.



(a) Disposición de las 4 cámaras

(b) Vista de la escena

Figura 21: Preparación del escenario para la grabación de las secuencias

Como se puede ver en la parte trasera de las cámaras de las imágenes de la Figura 21, y más específicamente en la Figura 22, hay distintos cables con distintas funciones:

- Un cable de energía, que alimenta la cámara para que esta pueda encenderse y funcionar. Se indicará que está encendida cuando se encienda un LED blanco en la parte trasera de la misma, como se puede ver en la Figura 22.
- Un cable de datos, de tipo *USB-C*, que se encarga de enviar toda la información capturada por la cámara al ordenador en uso.
- Un cable (para las cámaras situadas en las esquinas) o dos cables (para las cámaras centrales) de sincronización, de tipo *mini Jack*, que están interconectados entre las distintas cámaras, y que permiten realizar las capturas de las secuencias de manera sincronizada.

(a) Cámara en una esquina

(b) Cámara central

Figura 22: Parte trasera de las cámaras con sus cables

Una vez todas estas cámaras estén adecuadamente colocadas y conectadas, el siguiente paso será comprobar la detección y el funcionamiento de las mismas a través del ordenador en uso. Para esto, se utilizará el software Azure Kinect Viewer [42], que permitirá no solo confirmar que todos los sensores operan correctamente, sino también asegurarse de que cada cámara esté bien posicionada y localizada. Finalmente, este software facilitará experimentar con las distintas configuraciones que estas cámaras ofrecen, como los distintos modos de profundidad, las resoluciones o las tasas de refresco de cuadro, entre otros aspectos.

3.3. Software de captura

Para la captura de las secuencias, se ha utilizado el software desarrollado en [43], que permite grabar secuencias utilizando las cuatro cámaras Azure Kinect de manera sincronizada.

Este software permite, mediante distintos modos y para todas las cámaras, obtener tanto las imágenes RGB como las imágenes de profundidad asociadas a cada imagen de color. A continuación, se comentarán los distintos modos utilizados durante el proceso.

3.3.1. Modo de calibración

Una vez se haya comprobado que las cámaras están correctamente colocadas y funcionan adecuadamente, se inicia el proceso de calibración. Este se llevará a cabo simultáneamente para las cuatro cámaras que se estén utilizando.

El proceso de calibración consiste en relacionar coordenadas del mundo real en un espacio tridimensional, con la proyección de dichas coordenadas en el plano de la cámara, es decir, un espacio bidimensional. Para esto, se necesitará utilizar los parámetros extrínsecos, es decir, la rotación y traslación de la cámara con respecto a un sistema de coordenadas fijo. Posteriormente, utilizando los parámetros intrínsecos, como son la distancia focal y el centro óptico, se podrá proyectar con precisión el punto del espacio tridimensional con el que se esté trabajando, en el plano de la imagen bidimensional.

Las cámaras Azure Kinect incluyen un sensor RGB con una calibración de intrínsecos y unos parámetros de distorsión previos, que se utilizan en ese proceso. Por tanto, solo se necesitaría obtener los parámetros extrínsecos. Para ello, se utilizará un tablero con damero blanco y negro (patrón de ajedrez) y se tomarán múltiples imágenes con las cuatro cámaras, moviendo el tablero a través de distintas posiciones y ángulos. El software que se está utilizando se encargará de detectar los puntos de intersección entre cuadrados, y cada intersección detectada por una cámara se relacionará con las detectadas por el resto de cámaras, obteniendo así una correspondencia entre las cuatro imágenes.

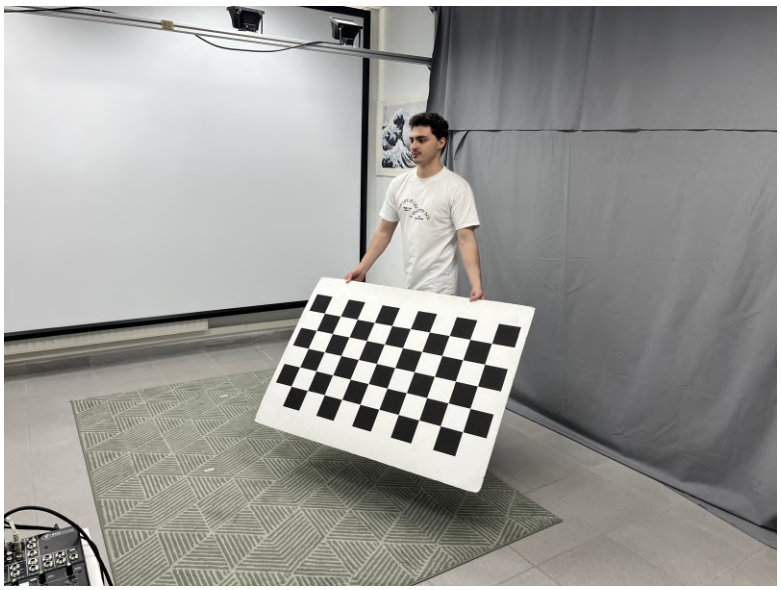
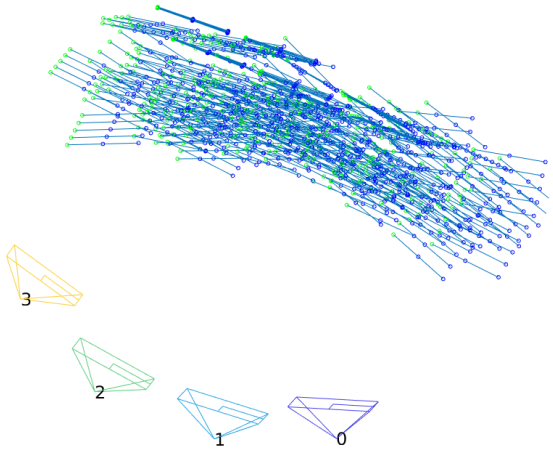
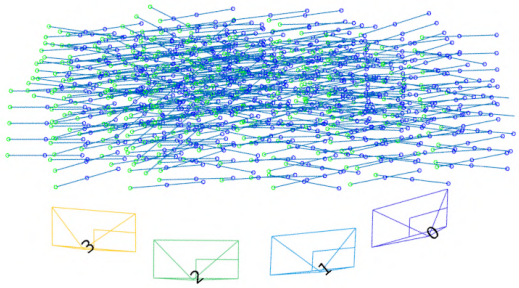


Figura 23: Tablero con damero usado durante la calibración

Como resultado de todo el proceso anterior, se obtendrá un fichero con formato *YAML*, que contendrá los intrínsecos y parámetros de distorsión para cada una de las cuatro cámaras y los extrínsecos correspondientes a la calibración. Además, se obtendrán los gráficos de la Figura 24, donde se pueden ver las 4 cámaras (numeradas de 0 a 3) desde arriba y desde detrás de forma esquemática y justo delante de ellas, se podrá ver una gran nube de puntos, que es el resultado de la posición del tablero en cada una de las imágenes tomadas por las cuatro cámaras, proyectadas en 3D.



(a) Vista de la calibración desde arriba de las cámaras



(b) Vista de la calibración desde detrás de las cámaras

Figura 24: Resultado gráfico de la calibración

Finalmente, se comentarán los parámetros relevantes que se introdujeron al realizar la calibración:

- **Tablero con damero blanco y negro**: se utilizará un tablero de 6x11 cuadrados de colores blanco y negro con 10,6 centímetros por lado. Esto implica que hay 5x10 intersecciones, que es el número de intersecciones con el que el software opera correctamente.
- **Cuadros a calibrar**: se configura para 20 cuadros, ya que esta información es suficiente para la calibración.
- **Periodo de cuadro**: es el tiempo que pasa entre la captura de dos imágenes es de 1500 milisegundos, ya que tiene que dar tiempo a cambiar la posición y rotación del tablero entre imágenes.
- **Tiempo de salida de la calibración**: es un tiempo, que se ha establecido en 300 segundos (5 minutos), y que cuando se llega a él, la calibración se detiene con el objetivo de que las cámaras no se queden capturando infinitamente.
- **Tiempo de exposición**: por defecto se utilizará un tiempo de 8000 microsegundos.

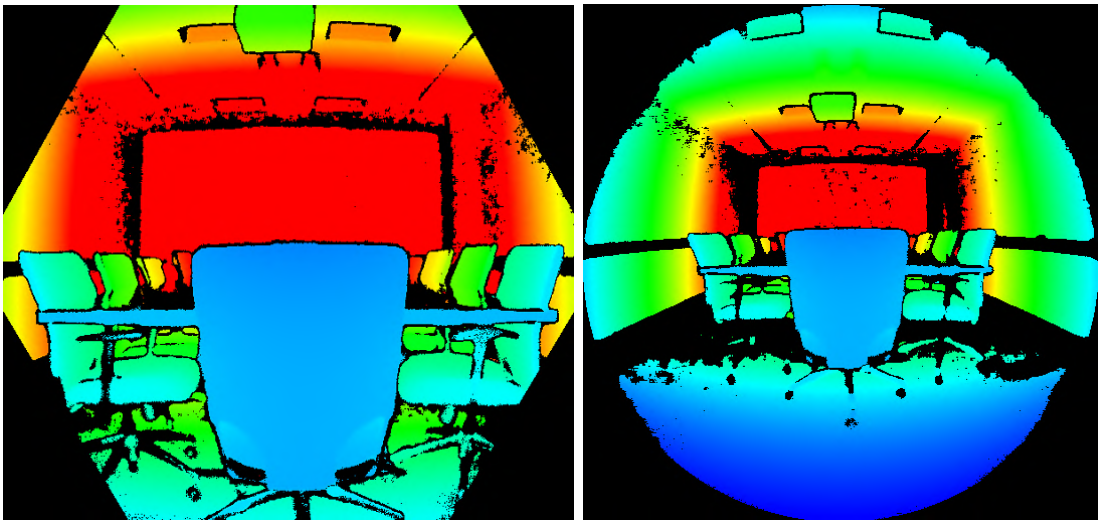
3.3.2. Modo de captura

La captura de las secuencias se realiza una vez queda hecha la previa calibración. En este modo se realizará la captura de los vídeos y se almacenarán en formato *MKV*. Para ello, es de vital importancia que las cámaras hagan las capturas en el mismo instante de tiempo. Como se ha mencionado anteriormente, las Azure Kinect tienen un sistema de sincronización entre cámaras basado en cables *mini Jack*. Cada cámara tiene un puerto de sincronismo de entrada y otro de salida y además, una de las cuatro cámaras (en este caso la *cámara 0*) actúa como una cámara "maestra", sincronizando la captura de imágenes con las otras cámaras conectadas.

A continuación, se comentarán los parámetros de entrada que se han utilizado a la hora de hacer la captura de las secuencias:

- **Número de cuadros a grabar**: por cuestión de espacio, se grabarán 450 cuadros a 15 cuadros por segundo, obteniendo una grabación de 30 segundos.
- **Tasa de cuadros por segundo**: se utilizarán 15 cuadros por segundo para facilitar la obtención de imágenes de profundidad.
- **Modo de las imágenes de color**: en este modo se podrán elegir las distintas resoluciones y formatos. Aunque las cámaras son capaces de alcanzar resoluciones de 3840x2160 (4K) en 16:9 y 4096x3072 en 4:3, se utilizará una resolución de 1280x720, consiguiendo así reducir la tasa binaria y la carga computacional del procesado.

- **Modo de las imágenes de profundidad**: en cuanto a los modos de profundidad, se pueden distinguir varios modos, como pueden ser: *Narrow FOV* (campo de visión estrecho) o *Wide FOV* (campo de visión ancho). El modo *Narrow FOV* es más útil para escenas en las que hay más información en el eje vertical que en los laterales, generando así imágenes hexagonales. En cuanto al modo *Wide FOV*, generará una imagen tipo "ojo de pez" obteniendo así un campo de visión más amplio de la escena. Además, existen las funciones "*binned*" o "*unbinned*" para ambos tipos de campos de visión. Si se utiliza "*binned*" mejorará la medición de la profundidad a costa de reducir la resolución a la mitad. En este caso, se decide utilizar el campo de visión ancho en modo "*unbinned*", buscando la máxima resolución.



(a) *Narrow FOV* (b) *Wide FOV*

Figura 25: Comparación de profundidad de campo de visión ancho y estrecho [10]

- **Tiempo de exposición**: por defecto se utilizará un tiempo de 8000 microsegundos.

3.3.3. Modo de extracción y codificación

Una vez realizada la calibración, de la cual se obtiene el fichero de calibración ya mencionado, y la captura de las secuencias, ya se puede comenzar con la extracción de los cuadros de cada una de las grabaciones. Como ya se ha mencionado antes, la extracción será de 450 cuadros de imágenes color y otros 450 cuadros de imágenes de profundidad para cada una de las cámaras. El sensor de profundidad no coincide con el sensor de color, por lo que se aplica una transformación (*warping*) a las imágenes de profundidad para que coincidan con la imagen de color. El resultado es, por cada una de las secuencias, cuatro carpetas de imágenes de color y cuatro carpetas de imágenes de profundidad.

Finalmente, y para terminar con el proceso, se realizará la codificación de vídeo, donde se prepararán estas imágenes de color y de profundidad para que sean compatibles

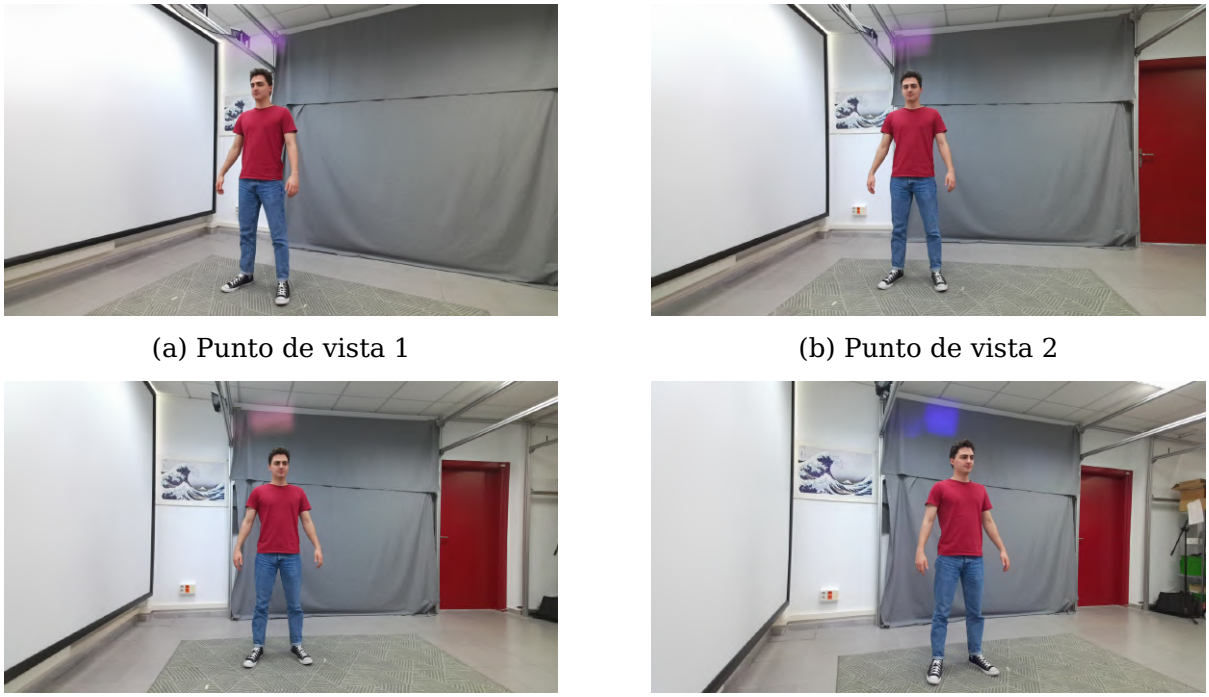
con el sistema *FVV Live*.

3.4. Grabación de las secuencias

Para la grabación de las secuencias con las cuatro cámaras Azure Kinect, se idearon cuatro escenas diferentes: dos individuales, una en pareja y una última con cuatro sujetos en la escena. Estos escenarios van aumentando su complejidad en función del aumento del número de sujetos que hay en la escena, el movimiento que estos hacen o los distintos objetos que se van colocando en la misma. Durante el proceso de grabación, se grabó a cuatro voluntarios, dos del género femenino y dos del género masculino, durante un tiempo de 30 segundos por cada secuencia.

A continuación, se exponen las diferentes secuencias capturadas y sus características desde la menos compleja a la más compleja:

- ***Standing Person***: esta es la secuencia individual, la más simple de todas las realizadas. En esta secuencia, se puede ver a un sujeto prácticamente quieto, mirando al frente.



(a) Punto de vista 1

(b) Punto de vista 2

(c) Punto de vista 3

(d) Punto de vista 4

Figura 26: Muestras de la secuencia *Standing Person*

- ***Moving Person***: en esta secuencia, también individual, se puede observar al mismo sujeto que en la secuencia *Standing Person*. No obstante, se consigue ganar complejidad en la escena mediante el movimiento del mismo, que mueve principalmente los brazos en varias direcciones mediante movimientos lentos y pautados.

(a) Punto de vista 1

(b) Punto de vista 2

(c) Punto de vista 3

(d) Punto de vista 4

Figura 27: Muestras de la secuencia *Moving Person*

- ***Standing People***: esta secuencia muestra a dos sujetos en una escena en la que se han añadido objetos de distintos tipos y de gran variedad de colores. Durante la escena, los sujetos, interactúan entre si, hablando y manteniéndose cerca de los distintos objetos e incluso interponiéndose delante de los mismos.

(a) Punto de vista 1

(b) Punto de vista 2

(c) Punto de vista 3

(d) Punto de vista 4

Figura 28: Muestras de la secuencia *Standing People*

- ***Moving People***: en esta última secuencia, se utilizaron a cuatro sujetos voluntarios. Durante la secuencia se intentó que hubiera movimiento por parte de todos los sujetos, moviéndose alrededor de la escena, buscando así una complejidad superior.

(a) Punto de vista 1

(b) Punto de vista 2

(c) Punto de vista 3

(d) Punto de vista 4

Figura 29: Muestras de la secuencia *Moving People*

4. Estimación de profundidad monocular con aprendizaje profundo

4.1. Planteamiento, equipo y entorno de desarrollo

Tras el estudio de los distintos modelos de estimación de profundidad del estado del arte, Depth Anything [38] ha demostrado una gran capacidad de generalización, obteniendo además imágenes de profundidad (Figura 19) de alta precisión. Por tanto, este trabajo basará sus desarrollos en dicho modelo de estimación de profundidad monocular, ya que actualmente no se han visto resultados mejores que los generados por este modelo.

En el campo del aprendizaje profundo, la mayoría de los desarrollos se realizan utilizando el lenguaje de programación Python [44]. Por tanto, debido a este factor y a que Depth Anything también lo utiliza, el software con el que se trabajará se desarrollará en este lenguaje, buscando así la máxima compatibilidad con este modelo.

Además, durante este trabajo se ha utilizado Git [45] como sistema de control de versiones y Anaconda [46] como gestor de paquetes y entornos en Python, ya que permite la creación de entornos cerrados que evitan conflictos entre librerías y dependencias, facilitando por tanto el uso de librerías complejas.

Al estar trabajando con redes neuronales, es necesario mencionar el uso de tensores. Con la continua evolución de este campo, han nacido distintos tipos de librerías que facilitan el trabajo con tensores, ya que no solo están optimizadas para un correcto funcionamiento en GPU, si no que también contienen implementaciones para la mayor parte de las operaciones requeridas por las redes neuronales.

En el caso de este trabajo, las implementaciones de aprendizaje profundo se han llevado a cabo mediante PyTorch [47], una librería de código abierto de Python desarrollada principalmente por el laboratorio de inteligencia artificial de Facebook. PyTorch contiene numerosos módulos con una gran cantidad de funcionalidades, que van desde el trabajo con tensores y operaciones con los mismos, hasta la construcción de redes neuronales e incluso un módulo de visión por ordenador llamado Torchvision.

Debido a que los modelos de estimación de profundidad monocular están basados en aprendizaje profundo, y que la complejidad de los algoritmos de este campo es elevada, además de que tratan con cantidades de información grandes, es común que sea necesario utilizar ordenadores de altas prestaciones, que por cuestión de fluidez y su capacidad de trabajar en paralelo, deben incluir GPUs. Durante el desarrollo de este proyecto, se ha tenido acceso a dos ordenadores, cuyas características se pueden ver en la Tabla 1.

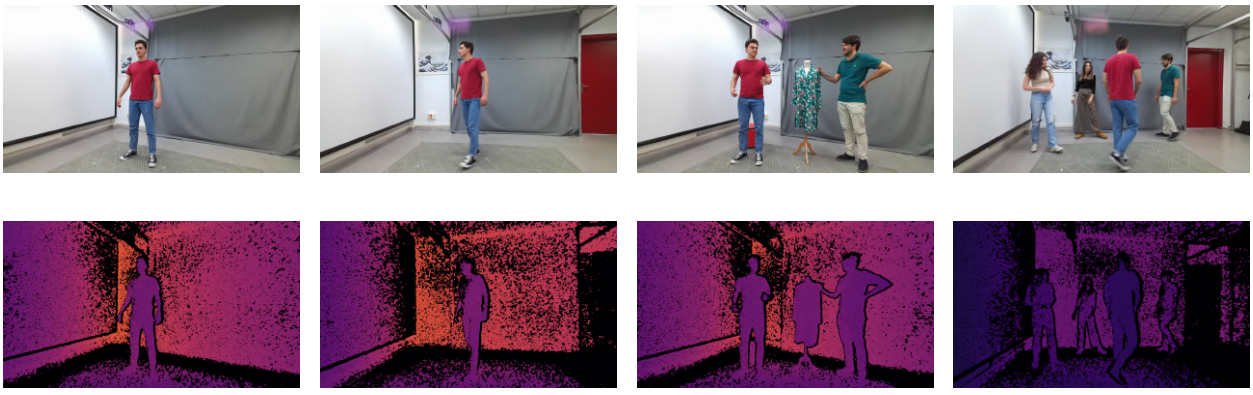
Tabla 1: Ordenadores usados durante el trabajo

Ordenador	Sistema operativo	Memoria RAM	Procesador	Tarjeta gráfica
carpanta	Ubuntu 18.04.6 64 bit	31.3 GiB	Intel Core i7-4790 3.60 GHz x 8	NVIDIA TITAN X
argos04	Ubuntu 16.04 64 bit	62.7 GiB	Intel Core i7-6850K 3.60 GHz x 12	NVIDIA GeForce GTX 1080 y Quadro P4000

Para el afrontar el procesado con el modelo de estimación de profundidad monocular elegido, se ha optado por seguir dos líneas similares, pero con cambios que ofrecen diferencias y que serán explicados en los próximos puntos.

4.2. Procesado con Depth Anything

Como método de unión entre los datos obtenidos por las cámaras Azure Kinect, cuyos resultados se pueden ver en la Figura 30, el modelo de estimación de profundidad elegido y la técnica de aprendizaje profundo que se va a usar, se han desarrollado en Python dos códigos que en un principio comparten características comunes. Sin embargo, a medida que se avanza en cada uno de ellos, se introducen cambios con el objetivo de analizar las diferencias en los resultados obtenidos. A continuación, se explicarán las similitudes entre ambos procesos.



(a) *Standing Person* (b) *Moving Person* (c) *Standing People* (d) *Moving People*

Figura 30: Imágenes obetenidas por las Azure Kinect

En ambos procesos, las imágenes RGB se introducirán en el modelo de estimación de profundidad monocular Depth Anything, que generarán a su salida imágenes de profundidad sin escala a partir de la imágenes RGB introducidas. Estas imágenes sin escala son una representación de la profundidad relativa de la escena que no proporcionan medidas de distancia absolutas. Estos procesos se han desarrollado para poder recuperar la escala que se había obtenido con las cámaras Azure Kinect.

En estos procesos, que están representados tanto en la Figura 31 como en la Figura 35, se parte de realizar las grabaciones con las cuatro cámaras Azure Kinect tal y como se explicó en la Sección 3 de este trabajo. Estas cámaras, generan dos tipos de imágenes que serán esenciales para estos procesos: imágenes RGB, que son una representación a color de la escena, e imágenes de profundidad, que proporcionan información sobre las distancias de los objetos de la escena con respecto a la cámaras que se están utilizando. Después de extraer los cuadros de cada una de las grabaciones realizadas, se obtienen como resultado 450 cuadros de color y 450 cuadros de profundidad por cámara en formato PNG, que son guardados en distintas carpetas.

Finalmente, y como preparación para el futuro uso del modelo de estimación de profundidad, se carga en ambos códigos el modelo Depth Anything, además, también se activa la GPU para facilitar la fluidez durante el procesado. Es a partir de este punto, donde comienzan las diferencias entre ambos procedimientos, ya que, como se ha dicho, se implementan diversas técnicas y ajustes para cada uno de ellos que serán explicados a continuación.

4.2.1. Procesado monovista con Depth Anything

Durante este procesado, una vez que se tienen todas las imágenes correctamente preparadas para su uso, se utiliza una función de carga de las mismas, llamada *imageLoader*, que se encarga de tomar todas las imágenes de color y de profundidad de una única cámara, comenzando por la primera cámara, y cargarlas cuadro a cuadro.

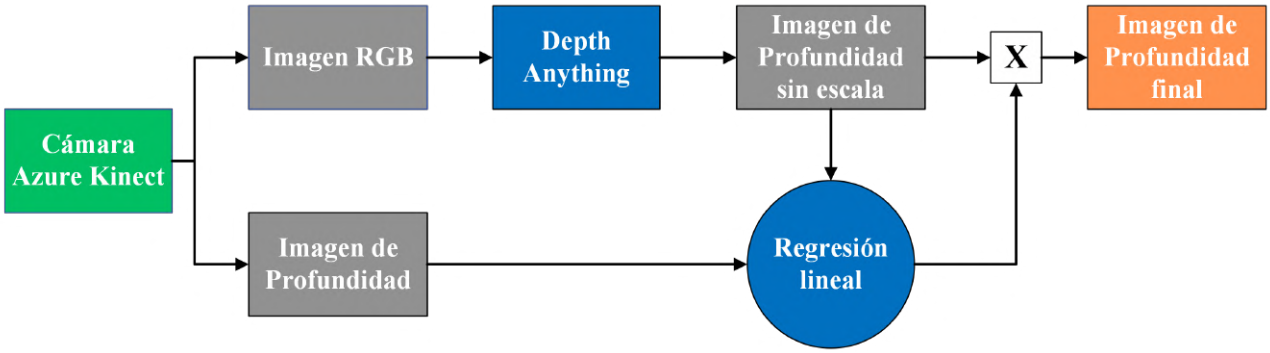


Figura 31: Esquema de procesado monovista con Depth Anything

A continuación, como se puede ver en la Figura 31, se tomará una imagen RGB y una imagen de profundidad, empezando por el primer cuadro de ambas. Como se necesita convertir la imagen de profundidad sin escala obtenida, en una imagen de profundidad con medidas absolutas, se ha utilizado una técnica de regresión lineal de grado uno, de las que se obtienen para cada iteración una pendiente y una ordenada en el origen (Figura 33). El uso de esta técnica de regresión lineal viene motivado porque al realizar el estudio de los distintos datos, se obtienen gráficas como la que se pueden ver en la

Figura 32, y estas presentan una tendencia aproximadamente lineal. La regresión lineal se entrena utilizando las imágenes de profundidad capturadas por las cámaras Azure Kinect, que contienen información de profundidad calibrada y precisa, con las imágenes de profundidad sin escala mencionadas. De esta manera se escalarán adecuadamente las imágenes de profundidad de Depth Anything.

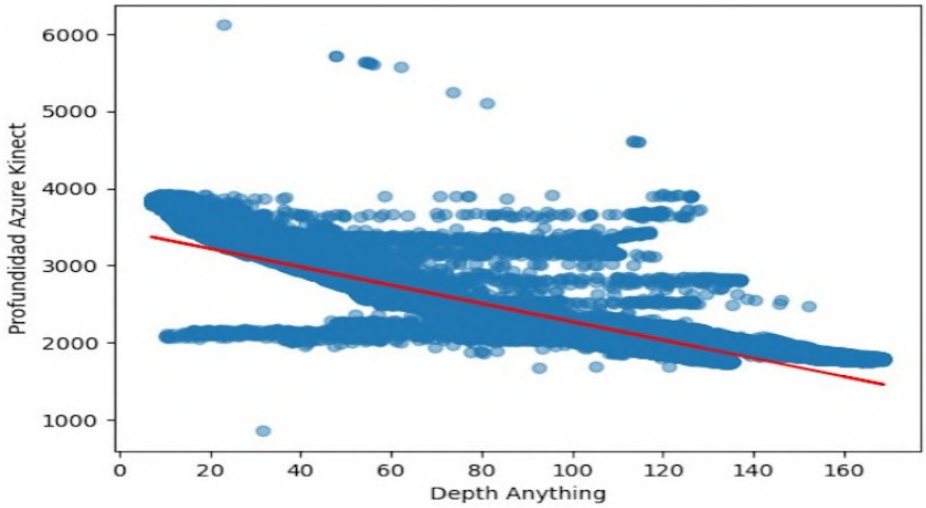
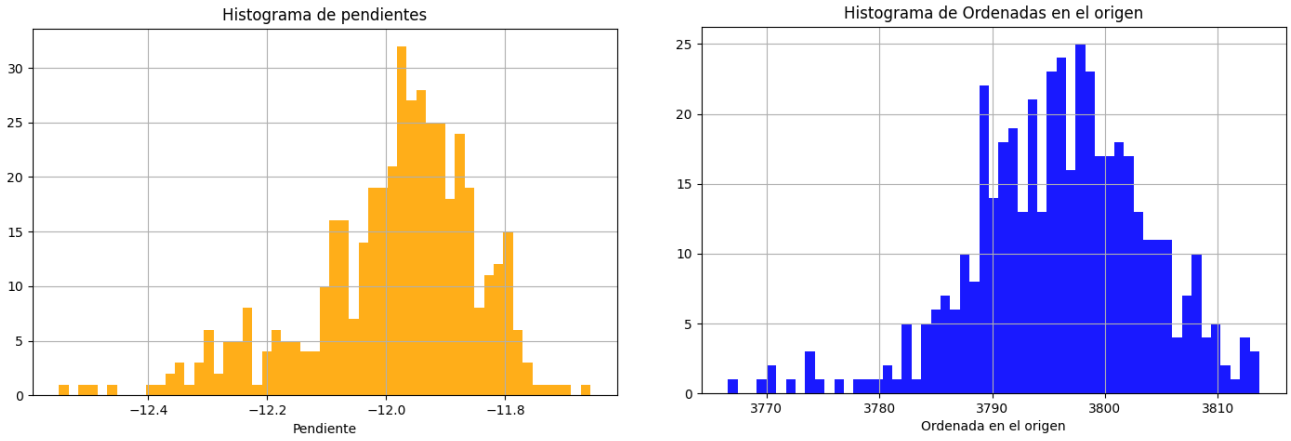


Figura 32: Ejemplo de relación entre profundidad de Depth Anything y de Azure Kinect

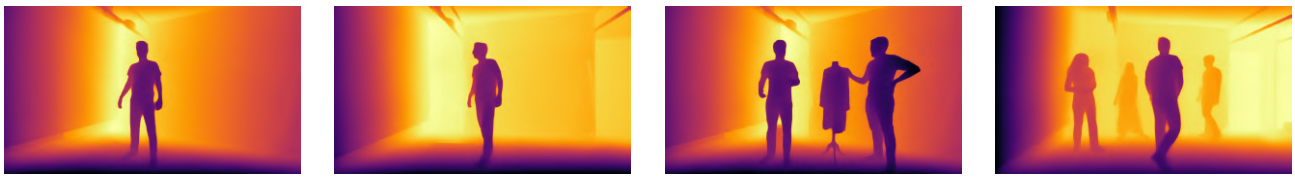


(a) Pendientes

(b) Ordenadas en el origen

Figura 33: Histograma de resultados de la regresión lineal monovista

Una vez realizado este proceso, se obtendrá una imagen de profundidad final (Figura 34), que ya estará escalada adecuadamente y que habrá mejorado su calidad y datos de profundidad con respecto a las obtenidas con las cámaras Azure Kinect. A continuación, este proceso se repetirá para el resto de las imágenes de una secuencia, es decir, para las 449 imágenes restantes, y para cada una de las cuatro cámaras. Este proceso, por tanto, se realizará un total de 1800 veces en cada una de las cuatro secuencias grabadas.



(a) *Standing Person* (b) *Moving Person* (c) *Standing People* (d) *Moving People*

Figura 34: Imágenes de profundidad tras el proceso monovista

4.2.2. Procesado multivista con Depth Anything

Para realizar el procesado multivista, se tiene que cambiar el enfoque. Se parte igualmente de tener todas las imágenes correctamente preparadas, pero en este caso, la función de carga de imágenes *imageLoader* cargará a la vez los cuadros de las cuatro cámaras utilizadas. Como se puede ver en la Figura 35, en cada iteración se estarán cargando cuatro imágenes RGB, una por cada cámara, correspondientes al mismo cuadro pero visto desde distintos puntos de vista, y cuatro imágenes de profundidad de la misma manera.

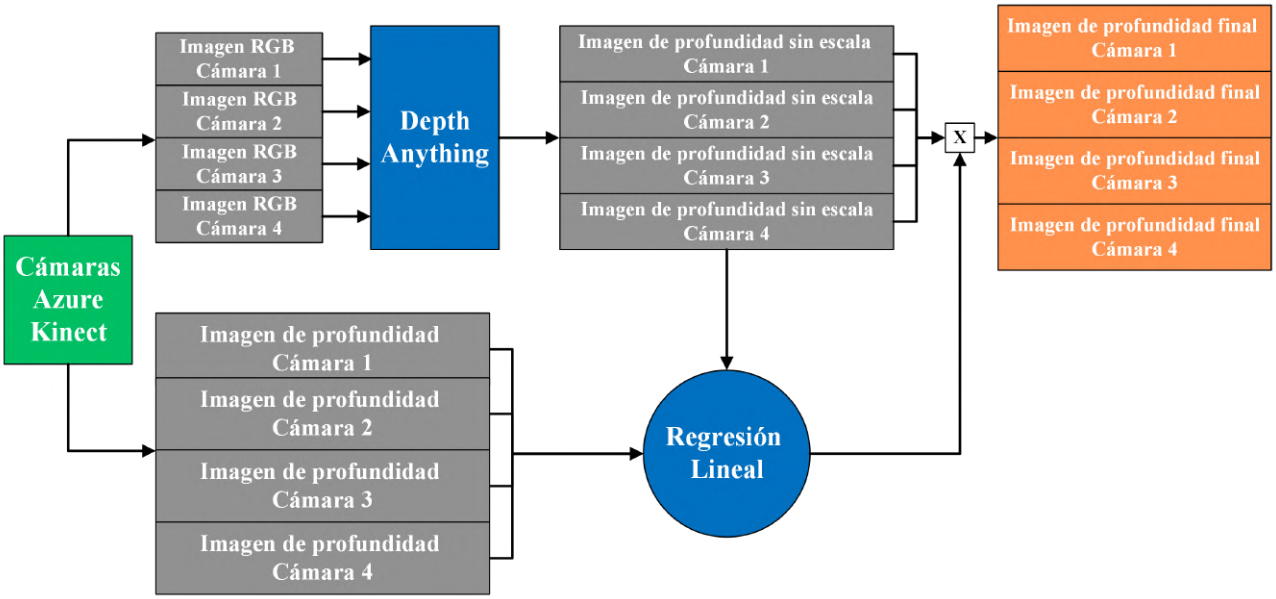
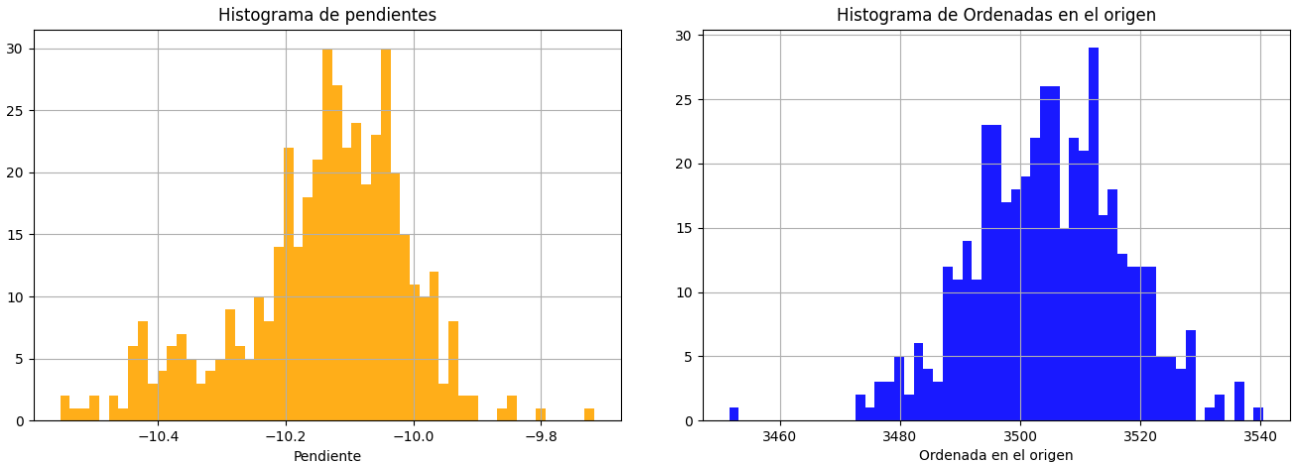


Figura 35: Esquema de procesado multivista con Depth Anything

A continuación, las cuatro imágenes RGB se introducirán en el modelo de estimación de profundidad Depth Anything, obteniendo a la salida cuatro imágenes de profundidad sin escala. Como ocurría en el proceso monovista, esta es una profundidad sin escala, por tanto, necesitamos realizar una técnica de regresión lineal junto con las imágenes de profundidad de las Azure Kinect para poder obtener las imágenes escaladas correctamente. En este caso, la regresión lineal se realizará al mismo tiempo para las cuatro imágenes correspondientes al mismo cuadro, obteniendo una pendiente y una ordenada en el origen por cada cuatro imágenes (Figura 36), relacionando así durante el entrena-

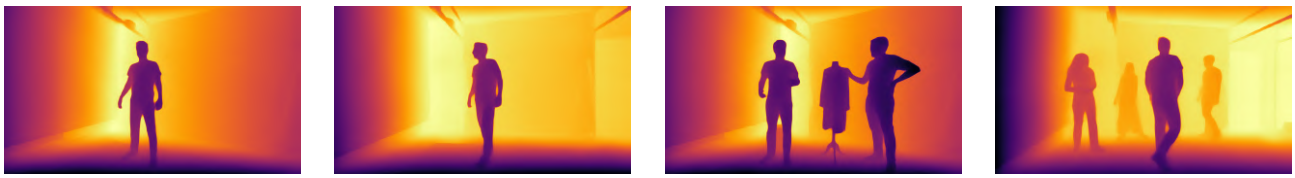
miento de la regresión lineal las cuatro imágenes entre sí, buscando mejores resultados que haciéndolo para cada punto de vista por separado.



(a) Pendientes (b) Ordenadas en el origen

Figura 36: Histograma de resultados de la regresión lineal multivista

Una vez realizada la regresión lineal para el primer grupo de cuatro imágenes correspondientes al primer cuadro, se obtendrán cuatro imágenes de profundidad ya escaladas y con las distancias de profundidad correctas (Figura 37). A partir de este punto, se pasará al siguiente grupo de imágenes correspondientes al siguiente cuadro hasta completar el proceso. En este caso, como el proceso se realiza para las cuatro cámaras a la vez buscando la relación entre ellas, para cada secuencia grabada se realizará un total de 450 veces.



(a) *Standing Person* (b) *Moving Person* (c) *Standing People* (d) *Moving People*

Figura 37: Imágenes de profundidad tras el proceso multivista

Cabe mencionar que el tiempo de ejecución entre el procesado monovista y el multivista es distinto, siendo más rápido este último, puesto que el proceso se tiene que realizar cuatro veces menos, aunque la duración de cada iteración es mayor en el procesado multivista que en el monovista.

Una vez realizados ambos procesados, se pasará a codificar todos los cuadros pertenecientes a cada una de las cámaras, obteniendo vídeos de profundidad en formato H.264. Estos vídeos serán los que se introducirán juntos con los cuadros de color codificados en el sistema *FVV Live*, obteniendo finalmente resultados.

4.3. Resultados

Para la obtención de los resultados de este trabajo, se introducen en el sistema *FVV Live* los vídeos que previamente se han generado, y se utilizan como base para sintetizar vistas virtuales. Se sintetiza un vídeo de 450 frames del *FVV Live* desde el punto de vista de cada una de las cuatro cámaras usadas y para todas las secuencias.

Se obtendrá para cada secuencia por tanto: cuatro vídeos que utilizan las imágenes de profundidad de las cámaras Azure Kinect, otros cuatro vídeos utilizando las imágenes de profundidad obtenidas con el procesado monovista con Depth Anything y cuatro vídeos utilizando las imágenes de profundidad obtenidas con el procesado multivista con Depth Anything. En total, para todas las secuencias, se obtendrán 48 vídeos en formato H.264.

A continuación, se decodificarán todos los vídeos obtenidos, recuperando los cuadros correspondientes a las secuencias obtenidas en el sistema *FVV Live*. Es en este punto donde se pueden empezar a ver las primeras diferencias, como se muestra en la Figura 38, la Figura 39 y en la Figura 40.

(a) Standing Person

(b) Moving Person

(c) Standing People

(d) Moving People

Figura 38: Resultados obtenidos con las cámaras Azure Kinect

Se observa que el resultado obtenido utilizando únicamente la profundidad de las cámaras Azure Kinect incluye partes de la escena que no han sido detectadas por los sensores de profundidad de las cámaras, dejando espacios sin rellenar, destacando su empeoramiento a la hora de captar profundidad a medida que los sujetos se alejan de las cámaras, apareciendo prácticamente borrados.

En la Figura 39 y la Figura 40, se mostrarán los resultados obtenidos tras aplicar los procesados con Depth Anything:

(a) Standing Person

(b) Moving Person

(c) Standing People

(d) Moving People

Figura 39: Resultados obtenidos con el procesado con Depth Anything monovista

(a) Standing Person

(b) Moving Person

(c) Standing People

(d) Moving People

Figura 40: Resultados obtenidos con el procesado con Depth Anything multivista

Como se puede ver en estos casos, las diferencias respecto a los resultados generados únicamente con las cámaras Azure Kinect son significativas. Pero a simple vista no se pueden observar diferencias en los resultados obtenidos entre ambos procesos. Es por esto que se realizarán distintas pruebas, entre ellas el cálculo del Error Cuadrático Medio (MSE) y la Relación señal-ruido de Pico (PSNR).

Para ello, se calculará la diferencia entre los cuadros sintetizados por el sistema *FVV Live* y los cuadros obtenidos con la grabación a color inicial, es decir la imagen original. Esto se realizará para cada una de las secuencias y sus correspondientes cámaras. No obstante, los resultados de la Tabla 2, muestran las medias de los resultados obtenidos para las cuatro cámaras. Además, como referencia, se utilizará la diferencia obtenida entre los cuadros obtenidos tras la decodificación de los vídeos a color originales ya codificados, obteniendo los valores que se tomarán como el mejor caso (referencia en la tabla).

Tabla 2: Resultados de las pruebas

Secuencia	**Método**	***MSE***	***MSE Desviación típica***	***PSNR (dB)***	***PSNR Desviación típica***
Standing Person	Referencia	8,171	0,156	39,019	0,086
	Azure	258,169	40,214	24,174	0,580
	Depth Anything monovista	31,995	0,181	33,184	0,027
	Depth Anything multivista	32,023	0,183	33,180	0,027
Moving Person	Referencia	8,186	0,225	39,011	0,120
	Azure	221,769	44,129	24,760	0,750
	Depth Anything monovista	31,438	0,677	33,251	0,094
	Depth Anything multivista	31,484	0,689	33,246	0,095
Standing People	Referencia	11,483	0,171	37,533	0,065
	Azure	638,736	81,941	20,145	0,512
	Depth Anything monovista	46,779	0,530	31,489	0,050
	Depth Anything multivista	46,649	0,596	31,495	0,056
Moving People	Referencia	9,324	0,366	38,442	0,175
	Azure	939,760	260,780	18,601	1,288
	Depth Anything monovista	38,810	14,590	32,477	0,936
	Depth Anything multivista	38,847	13,796	32,502	0,860

De los resultados obtenidos en la tabla superior, se destaca la gran diferencia entre los resultados obtenidos con las cámaras Azure Kinect, y los resultados obtenidos tras los procesados con Depth Anything.

Para el caso de las cámaras Azure Kinect, se trata de resultados razonables, puesto que se podía ver la gran cantidad de errores que estos resultados tenían a simple vista. Además, cabe mencionar que hay una gran diferencia entre las secuencias *Standing Person* y *Moving Person*, en las que la complejidad es menor, puesto que solo aparece una persona, y las secuencias *Standing People* y *Moving People*, en las que aparecen varias personas además de distintos objetos.

En cuánto a los resultados obtenidos tras procesar con Depth Anything, se puede ver que los resultados son muy parecidos, sobre todo en términos de PSNR, obteniendo unos resultados mínimamente mejores para el caso del procesado con Depth Anything multivista, en las secuencias más complejas, como son *Standing People* y *Moving People*. En cuanto a las secuencias *Standing Person* y *Moving Person*, se obtienen mejores resultados (una vez más, por una mínima diferencia) con el procesado con Depth Anything monovista.

Pese a los buenos resultados obtenidos utilizando cámaras desde los puntos fijos, se ha de mencionar que a medida que se va cambiando el punto de vista de la escena, hay inconsistencias en los puntos medios entre cámaras, generándose imágenes dobles como se puede ver en la Figura 41. Esto probablemente pueda deberse a problemas de calibración entre cámaras debido a que están bastante separadas unas de otras.

(a) Inconsistencia en *Standing Person*

(b) Inconsistencia en *Moving Person*

(c) Inconsistencia en *Standing People*

(d) Inconsistencia en *Moving People*

Figura 41: Resultados de las inconsistencias entre imágenes de profundidad

Finalmente, cabe destacar que a medida que aumenta la complejidad de cada una de las secuencias, los resultados que obtenemos para todos los métodos calculados van empeorando. Esto puede deberse a que se introducen movimientos, objetos y más personas a lo largo de las secuencias. No obstante, se ve una vez más el gran funcionamiento de los procesos de estimación de profundidad desarrollados, manteniéndose con una PSNR unos 6-7 dB por debajo de la PSNR de referencia para cada una de las secuencias.

5. Conclusiones y trabajo futuro

5.1. Conclusiones

El objetivo de este proyecto consistía en desarrollar un sistema que mejorara mediante modelos de estimación de profundidad monocular y aprendizaje profundo la calidad de las imágenes de profundidad generadas por las cámaras Azure Kinect utilizadas en este proyecto, y conseguir así optimizar el entorno virtual generado por el sistema *FVV Live*.

Para alcanzar esta meta, se ha pasado por diferentes puntos, como son la realización de las grabaciones de las distintas escenas con las cámaras Azure Kinect. Un objetivo que supuso un reto debido a que se tuvo que pensar minuciosamente la posición de cada una de las cámaras en la escena, las distintas grabaciones y el aumento de complejidad que iba a haber en cada una de ellas. Además, también ha supuesto un gran reto el aprendizaje del uso de los códigos de grabación y calibración de las cámaras, explicado en la Subsección 3.3.

Durante el estudio de los distintos modelos de estimación de profundidad monocular, se han realizado pruebas sobre el funcionamiento de cada uno de los modelos, llegando finalmente a la conclusión de que el mejor modelo de estimación de profundidad monocular actualmente es Depth Anything.

Trabajando con el procesado con Depth Anything monovista, se han relacionado las imágenes de profundidad obtenidas con el modelo Depth Anything con las de profundidad de las cámaras Azure Kinect para cada una de las cámaras por separado, y se ha comprobado que los resultados obtenidos con respecto a estas cámaras son muy superiores. Sobre todo para los casos en los que hay menos complejidad en las escenas.

Trabajando con el procesado con Depth Anything multivista, se han relacionado las imágenes de profundidad obtenidas con el modelo Depth Anything con las de profundidad de las cámaras Azure Kinect pero en este caso, para las cuatro cámaras de manera simultánea. Se ha observado que los resultados obtenidos son muy parecidos a los del procesado monovista, superando una vez más a los resultados obtenidos por las Azure Kinect. Además, se ha visto que este procesado funciona mejor para las escenas más complejas, ya sea por la cantidad de objetos o el movimiento.

No obstante, en ambos procesados se obtienen resultados muy prometedores en cuanto a la calidad de profundidad respecta, consiguiendo así unos buenos resultados a la hora de optimizar el entorno generado por el sistema *FVV Live*.

5.2. Líneas Futuras

Este proyecto abre varias líneas de cara a mejorar sus resultados. Como primera mejora, queda abierto a expandir el despliegue con un mayor número de cámaras, o desarrollar mecanismos que ofrezcan una mejor calibración de cara a las inconsistencias que en *FVV Live* se generaban.

En cuanto a los modelos de estimación de profundidad monocular, este proyecto queda abierto a la incorporación de nuevos modelos de estimación de profundidad monocular que puedan mejorar los resultados obtenidos con Depth Anything y, que incluso puedan mejorar las velocidades de procesado, obteniéndose nuevas mejoras para el sistema *FVV Live*.

Finalmente, se espera, como se ha mencionado al principio de este proyecto, integrar estos resultados, o los nuevos que se puedan obtener en el sistema *FVV Live* consiguiendo avanzar en la mejora continua de este sistema.

Anexos

A. Aspectos Éticos, Económicos, Sociales y Ambientales

A.1. Introducción

Este trabajo se enmarca dentro de la línea de investigación destinada al sistema de punto de vista libre *FVV Live*. El sistema *FVV Live* es un innovador sistema que permite navegar a un espectador libremente alrededor de una escena, creando experiencias agradables para los espectadores mediante un contenido más realista. Además, se destaca frente a otros sistemas de punto de vista libre por su capacidad para trabajar con un hardware de consumo asequible, de tal manera que pueda ser utilizado por un mayor número de personas.

A.2. Impactos relevantes relacionados con el proyecto

El desarrollo de innovadoras tecnologías de este tipo, abre las puertas a cuestiones de distintos aspectos. A nivel social, ya que a día de hoy las comunicaciones por vídeo tienen una gran importancia, tanto en ambientes personales como ambientes de trabajo. A nivel económico, el hecho de ser tecnologías innovadoras crea nuevas oportunidades de negocio, ya que las empresas están continuamente buscando nuevas áreas de creación de contenidos de cara a dar publicidad de propios productos.

En un nivel ético, y relacionándolo con los nuevos modelos de aprendizaje de máquina, es muy importante desarrollar sistemas igualitarios, que no discriminen a las personas, ni tengan preferencias de unas sobre otras. Además, dentro de este mismo entorno de aprendizaje de máquina, es necesario destacar las consecuencias medioambientales que estás innovadoras técnicas presentan, ya que el trabajo de estas con grandes volúmenes de datos implica un gran gasto energético que genera un impacto en la huella de carbono.

A.3. Análisis detallado del impacto social y ambiental

En los últimos años, ha quedado clara la gran importancia a nivel social de las comunicaciones a través de vídeo entre distintas personas. Con este tipo de sistemas, no solo se permite a las personas poder comunicarse a distancia de una manera más cercana y agradable, sino también abre las puertas a nuevas maneras de enseñanza y entretenimiento, siendo estas más interactivas y realistas que antes.

En cuanto al uso de aprendizaje profundo, redes neuronales y otras técnicas de este

campo, se ha avanzado mucho, consiguiendo resultados mucho mejores que los que se podían obtener hacer unos años. No obstante, el uso de estas técnicas trae consecuencias a nivel ambiental, debido a que, comúnmente utilizan numerosas GPUs y dispositivos que consumen grandes cantidades de energía para procesar los grandes volúmenes de datos con los que se trabaja. Esto conlleva por tanto a un gran impacto en la huella de carbono. Además, el aumento en el uso de distintos dispositivos hardware genera más residuos electrónicos, que actualmente son difíciles de reciclar al final de su vida útil. Es por esto, que ha surgido la necesidad de desarrollar nuevos métodos para optimizar el diseño o el funcionamiento de los recursos que se utilizan, para así reducir el impacto medioambiental.

A.4. Conclusiones

Actualmente, la necesidad de las personas de estar conectadas unas con otras es cada vez mayor. Es por ello, que las redes sociales y las nuevas técnicas de generación de contenidos multimedia tienen un importante papel en distintos ámbitos. En el ámbito social y económico, serán importantes para conectar a personas y generar nuevos modelos de negocio. Además, es importante llevar a cabo el desarrollo de estos sistemas de una manera respetuosa y sostenible, de cara a enriquecer la experiencias de cualquier usuario y siempre tratando de minimizar el impacto de la huella de carbono.

B. Presupuesto Económico

Tabla 3: Presupuesto económico

		Horas	**Precio/hora**	**Total**
Mano de obra (CD)		300	20,00 €	**6.000,00 €**
Recursos Materiales (CD)	**Precio de compra**	**Uso en meses**	**Amortización en años**	**Total**
Ordenador 1	2.000,00 €	6	5	200,00 €
Ordenador 2	3.000,00 €	6	5	300,00 €
4 cámaras Azure Kinect	1.600,00 €	6	3	266,67 €
4 extensores USB	32,00 €	6	1	16,00 €
3 cables de audio Jack	24,00 €	6	1	12,00 €
Total				**794,67 €**
Gastos Generales (CI)	25 %	sobre CD		**1.698,66 €**
Beneficio Industrial	6 %	sobre CD + CI		**509,60 €**
Subtotal Presupuesto				**9.002.93 €**
IVA Aplicable			21 %	**1.890,62 €**
TOTAL PRESUPUESTO				**10.893,55 €**

Referencias

[1] P. Carballeira, C. Carmona, C. Díaz, D. Berjón, D. Corregidor, J. Cabrera, F. Morán, C. Doblado, S. Arnaldo, M. d. M. Martín, and N. García, "Fvv live: A real-time free-viewpoint video system with consumer electronics hardware," *IEEE Transactions on Multimedia*, vol. 24, pp. 2378–2391, 2022.

[2] C.-C. Lee, A. Tabatabai, and K. Tashiro, "Free viewpoint video (fvv) survey and future research direction," *APSIPA Transactions on Signal and Information Processing*, vol. 4, p. e15, 2015.

[3] "4d replay," 2024. [Online]. Available: https://4dreplay.com.

[4] "Technology - frontiers of innovation," 2024. [Online]. Available: https://global.canon/en/technology/frontier18.html.

[5] "Intel true view technology," 2024. [Online]. Available: https://www.intel.com/content/www/us/en/sports/technology/true-view.html.

[6] "Cómo el sistema de vídeo free viewpoint de canon ofrece una nueva perspectiva de la acción de la copa del mundo de rugby 2019," 2024. [Online]. Available: https://www.canon.es/pro/stories/free-viewpoint-video-rugby-world-cup/.

[7] D. Berjón, P. Carballeira, J. Cabrera, C. Carmona, D. Corregidor, C. Díaz, F. Morán, and N. García, "Fvv live: Real-time, low-cost, free viewpoint video," in *2020 IEEE International Conference on Multimedia Expo Workshops (ICMEW)*, 2020, pp. 1–2.

[8] P. Pérez, D. Corregidor, E. Garrido, I. Benito, E. González-Sosa, J. Cabrera, D. Berjón, C. Díaz, F. Morán, N. García, J. Igual, and J. Ruiz, "Live free-viewpoint video in immersive media production over 5g networks," *IEEE Transactions on Broadcasting*, vol. 68, no. 2, pp. 439–450, 2022.

[9] "Azure kinect dk," 2024. [Online]. Available: https://azure.microsoft.com/es-es/products/kinect-dk/#x9f3fc470efe64022984c96b2509db6ac.

[10] "Azure kinect dk depth camera," 2024. [Online]. Available: https://learn.microsoft.com/en-us/azure/kinect-dk/depth-camera.

[11] F. Izaurieta and C. Saavedra, "Redes neuronales artificiales," *Departamento de Física, Universidad de Concepción Chile*, 2000.

[12] D. J. Matich, "Redes neuronales: Conceptos básicos y aplicaciones," *Universidad Tecnológica Nacional, México*, vol. 41, pp. 12–16, 2001.

[13] A. Dongare, R. Kharde, A. D. Kachare *et al.*, "Introduction to artificial neural network," *International Journal of Engineering and Innovative Technology (IJEIT)*, vol. 2, no. 1, pp. 189–194, 2012.

[14] S. Karsoliya, "Approximating number of hidden layer neurons in multiple hidden layer bpnn architecture," *International Journal of Engineering Trends and Technology*, vol. 3, no. 6, pp. 714–717, 2012.

[15] S. Karthigai and K. M. Sundaram, "Categorization of lung carcinoma using multilayer perceptron in output layer," *ICTACT Journal on Soft Computing*, vol. 10, no. 2, pp. 2035–2039, 2020.

[16] K.-L. Du and M. Swamy, *Perceptrons*, 12 2014, pp. 67–81.

[17] S. M. Kasongo, "A deep learning technique for intrusion detection system using a recurrent neural networks based framework," *Computer Communications*, vol. 199, pp. 113–125, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0140366422004601

[18] M. G. M. Abdolrasol, S. M. S. Hussain, T. S. Ustun, M. R. Sarker, M. A. Hannan, R. Mohamed, J. A. Ali, S. Mekhilef, and A. Milad, "Artificial neural networks based optimization techniques: A review," *Electronics*, vol. 10, no. 21, 2021. [Online]. Available: https://www.mdpi.com/2079-9292/10/21/2689

[19] J. Maurício, I. Domingues, and J. Bernardino, "Comparing vision transformers and convolutional neural networks for image classification: A literature review," *Applied Sciences*, vol. 13, no. 9, 2023. [Online]. Available: https://www.mdpi.com/2076-3417/13/9/5521

[20] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. Scao, S. Gugger, and A. Rush, "Transformers: State-of-the-art natural language processing," 01 2020, pp. 38–45.

[21] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," 2023.

[22] R. Szeliski, *Computer Vision: Algorithms and Applications*. Wiley, 2021. [Online]. Available: https://szeliski.org/Book

[23] C. L. González Gutiérrez, "Sistema de navegación autónoma para robot pepper basado en aprendizaje por refuerzo," 2022.

[24] J. Sierra-García and M. Santos, “Redes neuronales y aprendizaje por refuerzo en el control de turbinas eólicas,” *Revista Iberoamericana de Automática e Informática industrial*, vol. 18, no. 4, pp. 327–335, 2021.

[25] M. Mousavi, A. Khanal, and R. Estrada, “Ai playground: Unreal engine based data ablation tool for deep learning,” 12 2020.

[26] M. S. Junayed, A. Sadeghzadeh, M. B. Islam, L.-K. Wong, and T. Aydin, “Himode: A hybrid monocular omnidirectional depth estimation model,” 2022.

[27] Y. Ming, X. Meng, C. Fan, and H. Yu, “Deep learning for monocular depth estimation: A review,” *Neurocomputing*, vol. 438, pp. 14–33, 2021.

[28] X. Dong, M. A. Garratt, S. G. Anavatti, and H. A. Abbass, “Towards real-time monocular depth estimation for robotics: A survey,” 2021.

[29] F. Khan, S. Salahuddin, and H. Javidnia, “Deep learning-based monocular depth estimation methods—a state-of-the-art review,” *Sensors*, vol. 20, no. 8, 2020. [Online]. Available: https://www.mdpi.com/1424-8220/20/8/2272

[30] B. Ke, A. Obukhov, S. Huang, N. Metzger, R. C. Daudt, and K. Schindler, “Repurposing diffusion-based image generators for monocular depth estimation,” 2023.

[31] R. Birkl, D. Wofk, and M. Müller, “Midas v3.1 – a model zoo for robust monocular relative depth estimation,” 2023.

[32] H. Bao, L. Dong, S. Piao, and F. Wei, “Beit: Bert pre-training of image transformers,” 2022.

[33] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, “Swin transformer: Hierarchical vision transformer using shifted windows,” 2021.

[34] J. Li, X. Xia, W. Li, H. Li, X. Wang, X. Xiao, R. Wang, M. Zheng, and X. Pan, “Next-vit: Next generation vision transformer for efficient deployment in realistic industrial scenarios,” 2022.

[35] R. Ranftl, K. Lasinger, D. Hafner, K. Schindler, and V. Koltun, “Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 3, 2022.

[36] S. F. Bhat, R. Birkl, D. Wofk, P. Wonka, and M. Müller, “Zoedepth: Zero-shot transfer by combining relative and metric depth,” 2023.

[37] Z. Li, S. F. Bhat, and P. Wonka, “Patchfusion: An end-to-end tile-based framework for high-resolution monocular metric depth estimation,” 2023.

[38] L. Yang, B. Kang, Z. Huang, X. Xu, J. Feng, and H. Zhao, "Depth anything: Unleashing the power of large-scale unlabeled data," *arXiv:2401.10891*, 2024.

[39] M. Oquab, T. Darcet, T. Moutakanni, H. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby, M. Assran, N. Ballas, W. Galuba, R. Howes, P.-Y. Huang, S.-W. Li, I. Misra, M. Rabbat, V. Sharma, G. Synnaeve, H. Xu, H. Jegou, J. Mairal, P. Labatut, A. Joulin, and P. Bojanowski, "Dinov2: Learning robust visual features without supervision," 2024.

[40] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The KITTI dataset," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231–1237, Aug. 2013. [Online]. Available: https://doi.org/10.1177%2F0278364913491297

[41] P. K. Nathan Silberman, Derek Hoiem and R. Fergus, "Indoor segmentation and support inference from rgbd images," in *ECCV*, 2012.

[42] "Azure kinect viewer," 2024. [Online]. Available: https://learn.microsoft.com/es-es/azure/kinect-dk/azure-kinect-viewer.

[43] J. Usón, J. Cabrera, D. Corregidor, and N. García, "Analysing foreground segmentation in deep learning based depth estimation on free-viewpoint video systems," in *2022 IEEE 12th International Conference on Consumer Electronics (ICCE-Berlin)*, 2022, pp. 1–5.

[44] "Python," 2024. [Online]. Available: https://www.python.org/.

[45] "Git," 2024. [Online]. Available: https://git-scm.com/.

[46] "Anaconda," 2024. [Online]. Available: https://www.anaconda.com/.

[47] "Pytorch," 2024. [Online]. Available: https://pytorch.org.