

## **1. Introducción.**

En este proyecto se ha desarrollado un sistema orientado a la captura y procesamiento de información espacial en tiempo real, combinando visión artificial y computación embebida. El objetivo principal es generar un mapa del entorno en forma de nube de puntos, aprovechando sensores de profundidad y una plataforma hardware optimizada para tareas de procesamiento intensivo, como es la Kria KV260.

Para ello, se ha trabajado con una arquitectura completa que abarca desde la Kria, pasando por un sistema operativo, hasta la cámara de profundidad. A lo largo del proyecto se integran distintos componentes que permiten adquirir la información del entorno, procesarla eficientemente y obtener resultados útiles que pueden aplicarse en distintos ámbitos. En los siguientes apartados se explicarán los dispositivos utilizados, cómo se ha estructurado el sistema con Yocto, el proceso de generación de la nube de puntos y, finalmente, los resultados obtenidos y las posibles aplicaciones del proyecto.

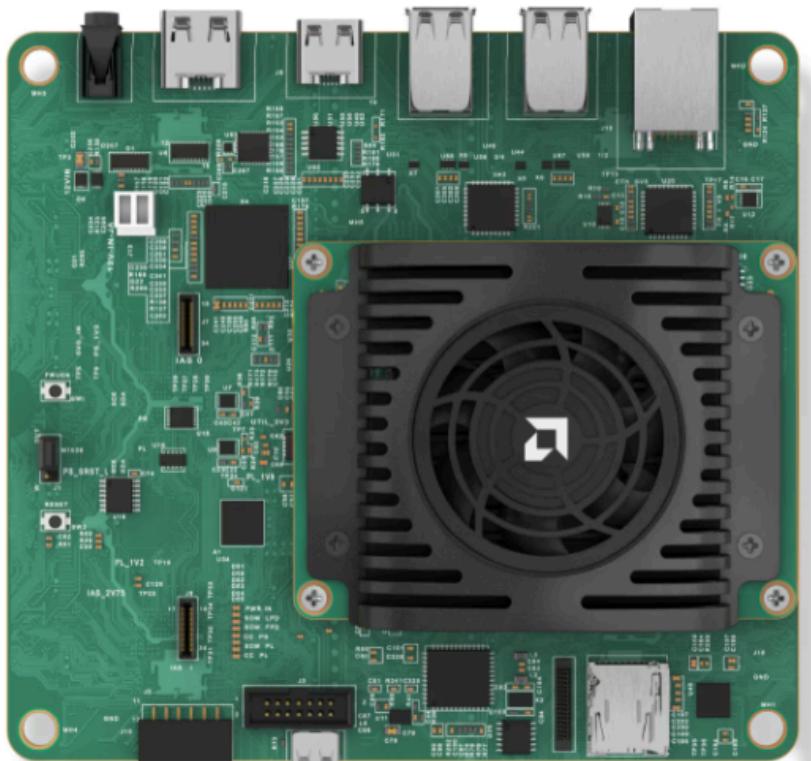
## **2. Dispositivos utilizados**

Como ya se ha mencionado, los dispositivos utilizados son la Kria KV260 y una cámara Realsense D415:

- Kria KV260:

La plataforma de desarrollo SoM (Sistema en modulo) AMD Kria™ KV260 está diseñada para el desarrollo de aplicaciones de visión avanzada sin requerir conocimientos complejos de diseño de hardware. Se trata de una plataforma de visión artificial y aplicaciones de IA en tiempo real, diseñada para facilitar el trabajo con modelos de machine learning y procesamiento de video sin necesidad de conocimientos complejos de diseño de hardware.

La placa integra un SoM en el procesador Zynq™ UltraScale+™ MPSoC, que combina CPU, GPU y lógica programable FPGA en una sola unidad, permitiendo así que pueda procesar tareas de visión y procesamiento de vídeo, entre otros.



#### Enlaces:

[https://www.amd.com/en/products/system-on-modules/kria/k26/kv260-vision-starter-kit.html?utm\\_source](https://www.amd.com/en/products/system-on-modules/kria/k26/kv260-vision-starter-kit.html?utm_source)

- RealSense D435:

Se trata de un sistema de cámaras de profundidad, creado por intel. Dentro de su arquitectura incluye un módulo estéreo de profundidad y un procesador de visión con conectividad via cable USB 2.0/USB 3.1.

#### Características

- Sistema de cámara de profundidad estéreo de segunda generación.
- Procesador de visión Intel® RealSense™ D4 de segunda generación con algoritmos avanzados.
- Sistema de proyector láser infrarrojo (IR) (Clase 1).
- Sensores de imagen con resolución Full HD 1920x1080 en RGB.
- Gestión activa de energía.
- Selección de módulos de profundidad estéreo para adaptarse a los requisitos de uso.
- Vista hasta 90°.
- Alcance de 0,2 m hasta más de 10 m.



Enlace: [Intel-RealSense-D400-Series-Datasheet](#)

### 3. Yocto.

Yocto Project es un proyecto de código abierto que permite crear distribuciones Linux embebidas personalizadas, adaptadas a un hardware y aplicación concretos.

PetaLinux es una distribución y un conjunto de herramientas construidas sobre Yocto Project, específicamente orientadas a plataformas Xilinx / AMD.

En Yocto, una \*\*Layer\*\* es un repositorio que contiene metadatos (recetas, configuraciones, clases) que le dicen al sistema de construcción (Bitbake) cómo construir el software. Las layers se apilan unas sobre otras, permitiendo modularidad y reutilización.

A continuación, se describen las layers configuradas en tu `bblayers.conf` .

- **Layers del Core (Poky)**

Estas layers provienen del repositorio principal de Yocto (Poky) y proporcionan la base del sistema.

***poky/meta (OpenEmbedded-Core)***

Es el corazón de Yocto. Contiene las recetas fundamentales para construir un sistema Linux básico .

Es obligatoria para cualquier build de Yocto.

***poky/meta-poky***

Define la "distribución" de referencia llamada Poky. Proporciona políticas de configuración por defecto y configuraciones específicas de la distro (como políticas de versiones).

- **Layers de Dependencias (Meta-OpenEmbedded)**

Estas layers extienden las funcionalidades básicas con una gran cantidad de software común de código abierto.

***meta-openembedded/meta-oe***

La capa base de OpenEmbedded. Contiene muchas librerías y utilidades que no caben en el core ('meta') pero son dependencias comunes para otros paquetes.

Importancia: Alta. Muchas recetas de otras capas dependen de librerías que están aquí.

#### ***meta-openembedded/meta-python***

Función: Proporciona soporte para Python y una inmensa cantidad de módulos de Python

- **Layers de Xilinx (Hardware Support)**

Estas layers son específicas para trabajar con FPGAs y SoCs de Xilinx (AMD).

#### ***meta-xilinx/meta-xilinx-core***

Contiene lo esencial para el soporte de Xilinx: definiciones de arquitectura (Zynq, ZynqMP, Versal), kernels de Linux de Xilinx, y bootloaders (U-Boot) modificados por Xilinx.

Fundamental. Aquí reside el kernel que arrancará tu placa.

#### ***meta-xilinx/meta-xilinx-bsp***

Proporciona configuraciones de máquina específicas (Machine Configuration) para las placas de Xilinx (evaluación, custom). Define qué drivers y configuraciones usar para una placa concreta (en tu caso, relacionado con las Kria).

#### ***meta-xilinx-tools***

Permite la integración con herramientas propietarias de Xilinx como Vivado o Vitis. Permite, por ejemplo, empaquetar un bitstream (firmware FPGA) dentro de la imagen o usar XSCT.

#### ***meta-kria***

Esta capa suele ser específica para las SOM (System on Modules) Kria. Contiene configuraciones específicas para el gestor de arranque de Kria ('xmutil'), gestión de \*firmware overlays\*, y configuraciones de la \*carrier card\*.

Crítica para que funcionen las características únicas de la Kria, como la carga dinámica de aplicaciones FPGA.

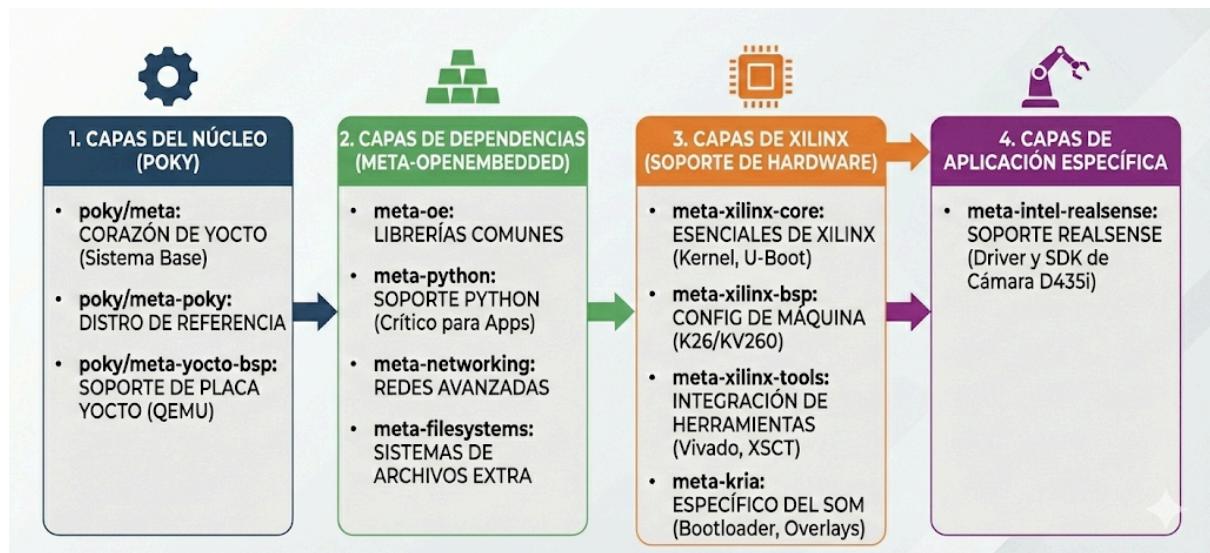
- **Layers de Aplicación Específica**

Layers añadidas para tu caso de uso concreto.

#### ***meta-intel-realsense***

Proporciona las recetas para instalar `librealsense2` (el driver y SDK de las cámaras RealSense) y sus utilidades.

Esta capa compila las librerías necesarias para que tu aplicación Python pueda hablar con la cámara D435i.



Enlace: <https://emlix.com/en/linux-engineering/yocto-engineering/>

#### 4. Generación de la nube de puntos.

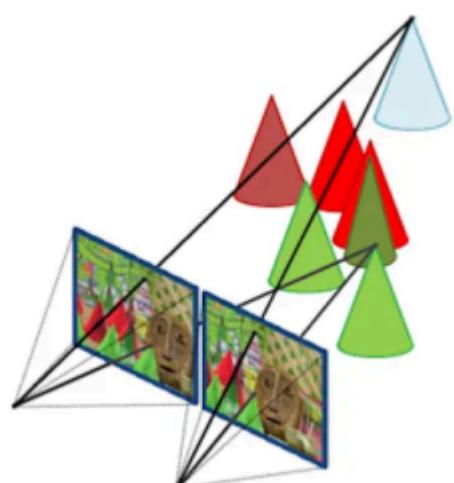
La generación de la nube de puntos se basa en el uso de la visión estéreo, permite obtener información de profundidad utilizando dos cámaras colocadas a una distancia conocida, en nuestro caso, las dos cámaras se encuentran en los lados de la RealSense. Al capturar la misma escena desde dos puntos de vista distintos, es posible calcular la posición tridimensional de cada punto mediante triangulación.

La implementación de la visión estéreo consiste en un sensor izquierdo, un sensor derecho y un proyector infrarrojo opcional. El proyector infrarrojo proyecta un patrón IR (infrarrojo) estático no visible para mejorar la precisión de la profundidad en escenas con baja textura.

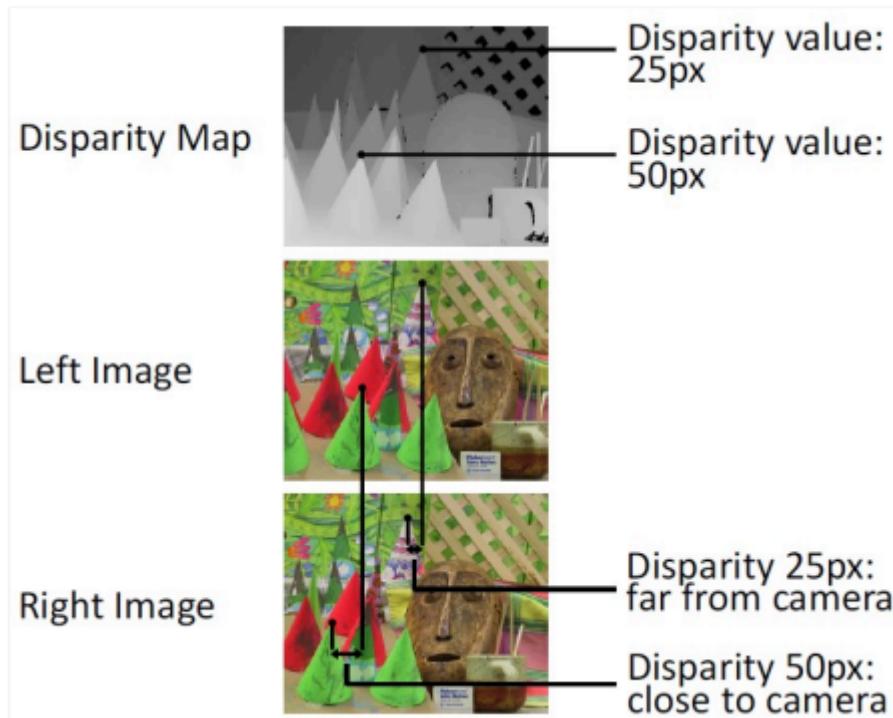
Los sensores izquierdo y derecho capturan la escena y envían los datos de imagen al procesador de imagen (visión), que calcula los valores de profundidad para cada píxel de la imagen correlacionando puntos de la imagen izquierda con la imagen derecha mediante un desplazamiento (disparidad) entre un punto de la imagen izquierda y su correspondiente en la imagen derecha.

Los valores de profundidad de cada píxel se procesan para generar un frame de profundidad. Los frames de profundidad sucesivos crean un flujo de vídeo de profundidad.

En este proyecto, el sistema ya proporciona internamente los parámetros necesarios de calibración. De manera interna, la RealSense realiza el procesado a través del procesador de visión D4 de Intel.



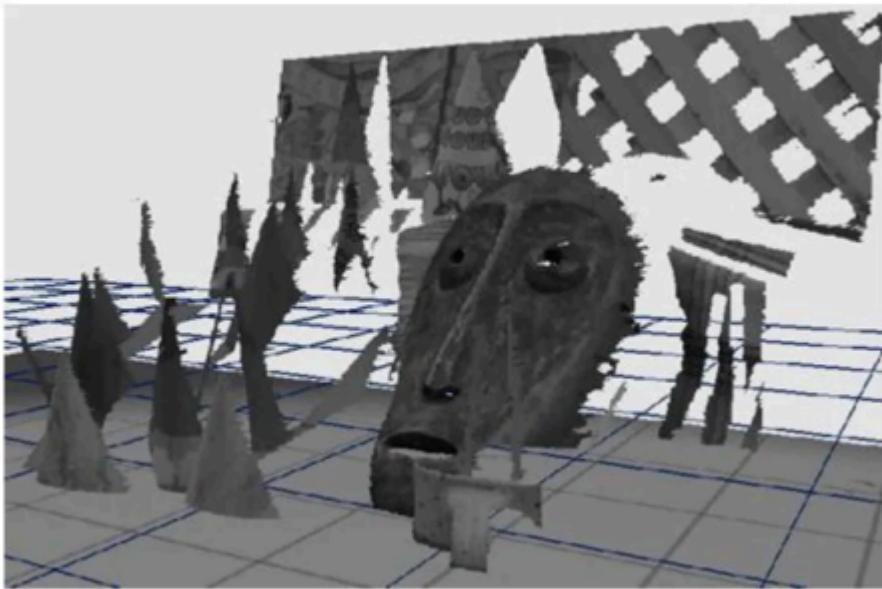
El siguiente paso, es la correspondencia estéreo, donde se identifican los píxeles de ambas imágenes que representan el mismo punto de la escena. A partir de esta correspondencia se obtiene un mapa de disparidad, que indica el desplazamiento entre los puntos de las dos imágenes. Finalmente, mediante un proceso de reproyección, estos valores de disparidad se transforman en coordenadas reales en el espacio, generando así la nube de puntos 3D que representa el entorno.



Disparidad.

Enlace:

<https://es.ids-imaging.com/files/downloads/knowledgebase/pdf/whitepaper/es/whitepaper-depth-information-3d-images.pdf>



Reproyección.

## 5. Demostración.

### 6. Aplicaciones reales

- **Arquitectura, ingeniería y construcción (AEC)**

El mapeo 3D permite capturar el estado real de un edificio o terreno con precisión geométrica. Esto resulta crítico en proyectos donde la información previa es incompleta o poco fiable.

Desde un punto de vista técnico y operativo:

Se trabaja sobre geometría real, no idealizada.

Se reducen errores de diseño y ejecución.

Se facilita la integración con entornos BIM para coordinación multidisciplinar.

El resultado es una mejora directa en costes, plazos y control del riesgo.

- **Robótica y vehículos autónomos**

Los sistemas autónomos requieren una comprensión espacial detallada del entorno para operar de forma segura. El mapeo 3D aporta información volumétrica que va más allá de la simple localización en un plano.

Su valor reside en que:

Permite navegación robusta en entornos dinámicos.

Mejora la planificación de trayectorias y la evasión de obstáculos.

Aumenta la fiabilidad de algoritmos de percepción como SLAM 3D.

Por ello, es un componente esencial en robótica industrial, logística y movilidad autónoma.

- **Inspección de infraestructuras y mantenimiento**

El mapeo 3D facilita la inspección remota de infraestructuras complejas o de difícil acceso, reduciendo costes y riesgos humanos.

Desde un enfoque técnico:

Permite mediciones precisas de deformaciones y desgaste.

Habilita comparaciones temporales entre modelos 3D.

Sustenta estrategias de mantenimiento predictivo.

Esto convierte al mapeo 3D en una herramienta clave para la gestión de activos críticos.

- **Realidad virtual, aumentada y contenidos digitales**

La creación de entornos virtuales realistas se beneficia enormemente del mapeo 3D de escenas reales, especialmente cuando se requiere fidelidad espacial.

El mapeo 3D:

Reduce el tiempo de modelado manual.

Aumenta el realismo geométrico.

Garantiza coherencia entre el espacio físico y el virtual.

Esto es especialmente relevante en simuladores de entrenamiento, videojuegos y aplicaciones de realidad aumentada.

- **Agricultura de Precisión**

En agricultura, el mapeo 3D permite caracterizar el terreno y los cultivos de forma detallada, integrando información geométrica con datos agronómicos.

Su aporte clave es que:

Mejora la estimación de pendientes, drenaje y microtopografía.

Permite análisis de volumen y altura de cultivos.

Facilita la aplicación localizada de riego, fertilizantes y tratamientos.

- **Salud y Ciencias de la Vida**

El mapeo 3D es fundamental en aplicaciones médicas y biomédicas donde la precisión espacial es crítica.

Desde un punto de vista clínico y científico:

Permite modelar anatomía específica de cada paciente.

Mejora la planificación quirúrgica y la simulación de procedimientos.

Facilita el diseño de prótesis e implantes personalizados.

En este contexto, el mapeo 3D contribuye directamente a una medicina más precisa, segura y personalizada.