**Seguimiento de objetos y detección de gestos en tiempo real**

#### **1. Introducción**

Este proyecto presenta la implementación de un sistema de visión 3D en tiempo real, basado en cámaras estéreo calibradas ZED, para interpretar interacciones humanas y coordinar acciones de robots simulados en RoboDK. El sistema detecta el número de dedos que muestra una persona y localiza una pelota de tenis en el espacio tridimensional, enviando esta información a través del protocolo MQTT para que uno o varios robots realicen una acción de recogida en un entorno virtual.

#### **2. Calibración y Triangulación**

Para asegurar la precisión espacial, se realiza una doble calibración del sistema: primero, calibración individual de cada cámara para obtener sus parámetros intrínsecos (matriz de cámara y distorsión); luego, calibración estéreo para obtener la relación geométrica entre ambas cámaras (rotación y traslación). Esta información es crucial para la **triangulación estéreo**, que permite calcular la posición tridimensional de un objeto (como una pelota) a partir de sus coordenadas en las imágenes de ambas cámaras. Esta triangulación se implementa mediante la función cv2.triangulatePoints.

#### **3. Obtención de Imágenes con Cámaras ZED**

Se utilizan dos cámaras ZED2 calibradas para capturar imágenes en tiempo real desde dos perspectivas diferentes. Estas imágenes, correspondientes a las vistas izquierda y derecha, se guardan de forma sincronizada en directorios separados. Un hilo específico del programa principal se encarga exclusivamente de esta tarea de captura continua, permitiendo así una operación paralela con el procesamiento de datos.

#### **4. Detección de Dedos**

El sistema utiliza la biblioteca **MediaPipe** para detectar manos y calcular el número de dedos extendidos. A través del análisis de puntos clave (landmarks) de la mano, se evalúa si cada dedo está levantado según su geometría relativa. Este dato determina cuántos robots deben actuar haciendo un “pick” sobre la pelota. La detección se realiza en tiempo real sobre imágenes capturadas por la cámara izquierda.

#### **5. Detección de la Pelota con YOLOv8**

La detección de la pelota de tenis se basa en el uso del modelo de red neuronal **YOLOv8**, entrenado para reconocer objetos clasificados como “sports ball”. El sistema analiza ambas imágenes (izquierda y derecha), identifica el centroide de la pelota en cada una, y posteriormente aplica triangulación para obtener su posición 3D. La clase BallDetector encapsula todo el flujo de detección, carga de modelo, proyección y escalado de coordenadas para su uso posterior.

#### **6. Funcionamiento del Bucle Principal**

El archivo main.py gestiona el ciclo completo de ejecución del sistema. Se estructura mediante una **clase principal** que contiene la configuración de la cámara, la inicialización de los detectores (pelota y dedos), y el cliente MQTT. Esta clase define dos funciones fundamentales:

* capture\_images: se encarga de capturar en tiempo real las imágenes sincronizadas de la cámara ZED (izquierda y derecha), y guardarlas con nombres secuenciales en directorios predefinidos.
* process: busca continuamente pares de imágenes sincronizadas ya capturadas. Luego, realiza la detección de dedos (con MediaPipe) y la detección de pelota (con YOLO), aplica la triangulación y, si ambos elementos son válidos, publica los resultados por MQTT.

Estas dos funciones se ejecutan **en paralelo** usando **hilos (threads)**, lo que permite que el sistema capture imágenes y procese datos simultáneamente sin bloqueo. Esta arquitectura multihilo asegura que el procesamiento y la publicación se mantengan en tiempo real mientras la cámara sigue generando nuevas imágenes.

#### **7. Comunicación y Publicación por MQTT**

Una vez obtenidos los datos de interés (número de dedos y posición 3D de la pelota), estos se publican en tiempo real en dos tópicos MQTT:

* v3d/finger para el número de dedos
* v3d/position para la posición tridimensional de la pelota

#### **7. Acción Robótica en RoboDK**

En RoboDK, un script en Python recibe los datos enviados por MQTT y ejecuta las acciones correspondientes. Se activan el mismo número de robots que dedos detectados y los robots activados realizan una operación de “pick” sobre la pelota, la cual se genera en una posición equivalente en el sistema del mundo virtual, dicha posición se calcula mediante homografía.

#### **8. Conclusión**

El sistema desarrollado logra la integración fluida entre percepción visual 3D, interpretación de gestos, tracking de objetos y control de robots simulados, todo en tiempo real. A través de una arquitectura modular basada en calibración precisa, triangulación estéreo, redes neuronales y comunicación por MQTT, el proyecto demuestra una solución robusta y adaptable para aplicaciones de simulación, automatización y colaboración humano-robot.

#### **9. Reparto de tareas**

* **Gonzalo Albelda Barona**: Responsable del entorno virtual en RoboDK y de la comunicaciónMQTT entre el sistema de visión y los robots.
* **Jose Luis Galán Avilés**: Encargado de la calibraciónestéreo, la triangulación para obtener la posición 3D de la pelota y apoyo en la deteccióndededos y pelota.
* **Alberto Andrés**: Desarrollador del bucleprincipal **(main.py)**, incluyendo la capturadeimágenes, uso de hilos, y la estructura general del proyecto.
* **Gonzalo Martín Peñalba**: Responsable de la detección de dedos con MediaPipe y de la detección de la pelota con YOLOv8.