

Implementación de sistema de visión artificial usando webcam de bajo coste para manipulación robótica en aplicaciones educativas

Yepez-Figueroa, Johnny J.*, Oña, Edwin D., Balaguer, Carlos, Jardón, Alberto

Robotics Lab, Dpto de Ingeniería de Sistemas y Automática, Universidad Carlos III de Madrid, Avda. de la Universidad, 30, 28911, Leganés, España.

Resumen

Este proyecto presenta el desarrollo e implementación de un sistema de visión artificial de bajo coste, orientado a entornos educativos, con capacidad para ejecutar tareas de manipulación robótica, como operaciones de Pick & Place, utilizando un robot colaborativo ABB CRB 15000. El sistema emplea una cámara web calibrada mediante un patrón de tablero de ajedrez, permitiendo obtener parámetros intrínsecos y de distorsión óptica con suficiente precisión para aplicaciones prácticas. La detección de objetos se basa en un modelo personalizado de YOLOv8, entrenado para identificar tres tipos de frutas: manzana, banana y naranja. Una vez detectadas, las coordenadas en píxeles se transforman a coordenadas reales mediante una matriz de homografía, y se envían al robot a través de una conexión TCP/IP para ejecutar la acción correspondiente. El sistema sigue una configuración Eyes-to-Hand, facilitando una visión estable del área de trabajo y simplificando la calibración. La validación experimental demostró una precisión de localización con errores relativos entre 0.1 % y 0.6 %, y una respuesta robusta ante objetos reales. Finalmente, se proponen líneas de mejora como la estimación de pose 3D, el uso de sensores de profundidad y la integración con Unity para visualización aumentada, ampliando así su aplicabilidad en formación técnica y automatización flexible.

Palabras clave: Robótica Colaborativa, Visión por Computador, Calibración de Cámara, Manipulación Robótica, YOLO

Implementation of artificial vision system using low-cost webcam for robotic manipulation in educational applications

Abstract

This project presents the development and implementation of a low-cost artificial vision system, oriented to educational environments, capable of executing Pick & Place tasks using an ABB CRB 15000 collaborative robot. The system employs a webcam calibrated by means of a checkerboard pattern, allowing to obtain intrinsic and optical distortion parameters with sufficient accuracy for practical applications. Object detection is based on a customized YOLOv8 model, trained to identify three types of fruit: apple, banana and orange. Once detected, the pixel coordinates are transformed to real coordinates using a homography matrix, and sent to the robot through a TCP/IP connection to execute the corresponding action. The system follows an Eyes-to-Hand configuration, facilitating a stable view of the working area and simplifying calibration. Experimental validation showed a localization accuracy with relative errors between 0.1 and 0.6 %, and a robust response to real objects. Finally, lines of improvement are proposed such as 3D pose estimation, the use of depth sensors and integration with Unity for augmented visualization, thus extending its applicability in technical training and flexible automation.

Keywords: Collaborative robotics, Computer Vision, Camera Calibration, Pick & Place, YOLO

1. Introducción

En los entornos educativos enfocados en la robótica colaborativa, existe un creciente interés por acercar a los estudiantes a tecnologías utilizadas en entornos industriales reales, como los sistemas de visión artificial integrados con modelos de apren-

dizaje profundo. Aplicaciones como es la manipulación robótica, como operaciones de Pick & Place, comúnmente vistas en líneas de producción automatizadas, pueden ser replicadas a menor escala mediante soluciones de bajo coste que combinan cámaras convencionales y algoritmos de detección en tiempo real (Automation (2023)).

*Autor para correspondencia: johnnyjavier.yepez@uc3m.es
Attribution-NonCommercial-ShareAlike 4.0 International (CC BY-NC-SA 4.0)

La visión artificial ha recorrido un largo camino desde sus inicios, pasando de ser una tecnología complementaria a convertirse en una herramienta clave en la automatización industrial moderna. Impulsada por avances en el Deep Learning, ha permitido un nivel de precisión y adaptabilidad que sería impensable con métodos tradicionales. Esta evolución ha sido fundamental para transformar procesos tradicionales y aumentar la eficiencia en diversos entornos industriales (Torres Abril et al. (2023)).

El presente artículo tiene como objetivo demostrar la viabilidad de implementar un sistema de visión artificial de bajo coste, orientado a aplicaciones cuasi industriales en entornos educativos. Para ello, se emplea una cámara web accesible —específicamente el modelo Trust Trino HD— cuya calibración se realiza mediante un patrón de tablero de ajedrez. El sistema busca validar que, a pesar de sus limitaciones frente a cámaras industriales, esta solución es capaz de ofrecer una precisión suficiente para ejecutar tareas representativas de automatización, como la detección de objetos y operaciones de Pick & Place.

En el diseño de sistemas de visión para robótica, una decisión clave es la configuración de la cámara respecto al robot, por lo tanto, se ha optado por una disposición Eyes-to-Hand, en la cual la cámara permanece fija en el entorno, observando el área de trabajo desde una posición estable. Esta elección, en contraste con el enfoque Eye-in-Hand —donde la cámara se monta directamente sobre el efector final del robot— (Chen and Hu (2023)), ofrece importantes ventajas para un entorno educativo: permite mantener una calibración más consistente, reduce la complejidad del sistema al no requerir actualizaciones constantes de la posición de la cámara y proporciona una visión más amplia del espacio de trabajo, incluyendo tanto los objetos a manipular como el propio robot (Enebuse et al. (2021)).

Este proyecto contempla la detección de frutas —específicamente manzanas, naranjas y bananas— mediante un modelo previamente entrenado con YOLO. Una vez detectadas, las coordenadas de la fruta seleccionada se transforman al sistema de referencia del robot colaborativo ABB CRB 15000, el cual ejecuta el proceso de recogida y posterior envasado.

Mediante esta iniciativa, se pretende sentar las bases para que en el futuro se puedan desarrollar prácticas de pick & place utilizando sistemas de visión artificial en entornos de educación en automática. Se busca así ofrecer una alternativa accesible y funcional para que los estudiantes puedan experimentar con tecnologías modernas de automatización sin necesidad de una gran inversión en hardware especializado.

1.1. Técnicas de visión

La estimación precisa de la posición de objetos en el espacio es fundamental en sistemas de visión por computadora aplicados a la robótica, especialmente en escenarios de teleoperación asistidos por realidad aumentada. Para lograr esta estimación, se han desarrollado distintas técnicas de configuración del sistema de cámaras respecto al robot, entre las cuales destacan “Eyes in Hand” y “Eyes to Hand”.

1. **Eyes-in-Hand:** Consiste en montar la cámara directamente en el efector final del robot, lo que permite una vista dinámica del entorno desde la perspectiva del manipulador. Esta configuración proporciona una alta flexibilidad y precisión local, ya que la cámara se mueve junto

con el robot (Balanji et al. (2022)). Sin embargo, su campo de visión puede ser limitado y depende de la posición del brazo.

2. **Eyes-to-Hand:** Adoptada en este trabajo, ubica la cámara fija en una posición externa al robot, orientada hacia su espacio de trabajo. Esta configuración permite una vista general y constante del entorno, facilitando el seguimiento de múltiples objetos dentro del área de operación (Balanji et al. (2022)). No obstante, requiere una cuidadosa calibración entre el sistema de visión y el sistema de coordenadas del robot.

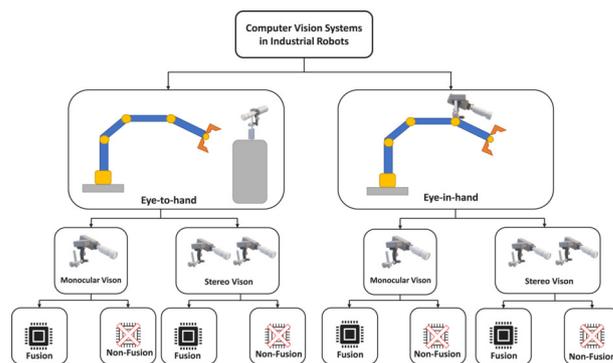


Figura 1: Application of computer vision systems in industrial robots Balanji et al. (2022).

1.2. Técnicas de calibración

En cuanto a las técnicas de calibración de cámaras, estas son esenciales para obtener una estimación precisa de la posición y orientación de los objetos. Una de las técnicas más utilizadas es la calibración mediante un tablero de ajedrez, donde se capturan múltiples imágenes del patrón desde diferentes ángulos. Utilizando librerías como “OpenCV”, es posible extraer los parámetros intrínsecos de la cámara, como la matriz de calibración y los coeficientes de distorsión (Kaiser et al. (2024)).

Otra técnica comúnmente empleada en aplicaciones de realidad aumentada es el uso de marcadores “ArUco”, que permiten una detección rápida y robusta de la pose de objetos en el espacio. Los marcadores ArUco codifican información en un patrón binario que puede ser identificado fácilmente por algoritmos de visión (Ošćádal et al. (2020)).

Ambas técnicas presentan ventajas según el tipo de aplicación. Por un lado, los patrones de ajedrez se han consolidado como una de las herramientas más precisas para la calibración inicial de sistemas de visión, especialmente en configuraciones multivisión, donde es fundamental minimizar errores de alineación. Estudios recientes han demostrado que este enfoque puede lograr errores mínimos, con valores de RMSE tan bajos como 0.02 mm, lo cual valida su idoneidad incluso en contextos clínicos y de alta exigencia metrológica (Kaiser et al. (2024)). Por otro lado, los marcadores ArUco destacan por su capacidad de ser detectados de forma rápida y robusta en tiempo real, lo que los convierte en una solución ideal para la estimación dinámica de pose y la referencia de objetos dentro del entorno, especialmente en aplicaciones de realidad aumentada y robótica interactiva.

En este trabajo, se opta por el enfoque Eyes-to-Hand junto con calibración con tablero de ajedrez debido a su exactitud

y compatibilidad con la metodología de posicionamiento desarrollada.

2. Arquitectura del Sistema de visión

El sistema de visión artificial desarrollado en este proyecto se basa en una arquitectura modular que integra componentes de percepción, calibración y comunicación para automatizar tareas de Pick & Place en un entorno educativo (Figura 2). La configuración adoptada es *Eyes-to-Hand*, donde una cámara web modelo Trust Trino HD permanece fija observando el área de trabajo, lo que permite mantener constante la referencia espacial con respecto al entorno. La calibración de la cámara se realiza mediante un patrón de tablero de ajedrez, utilizando la librería *OpenCV*, lo que permite estimar con precisión los parámetros intrínsecos y corregir distorsiones ópticas.

Para el reconocimiento de objetos, se entrena un modelo de detección basado en *YOLO* (You Only Look Once), mediante la librería *ultralytics*, adaptado para identificar tres tipos de frutas: manzana, banana y naranja. La programación del sistema de visión se lleva a cabo en Python, utilizando librerías como *cv2*, *numpy*, *socket*, *select* y *time*, que permiten gestionar el procesamiento de imágenes, las estructuras de datos y la comunicación entre procesos. La detección se ejecuta en tiempo real, y las coordenadas de los objetos detectados son transformadas a coordenadas del mundo real y enviadas al robot colaborativo ABB CRB 15000 mediante una conexión TCP/IP.

El robot está programado en el lenguaje RAPID, propio de RobotStudio, lo que permite recibir datos desde el sistema de visión y ejecutar acciones coordinadas de toma y colocación de objetos, siguiendo una arquitectura distribuida donde el procesamiento visual y el control del robot operan de forma integrada pero independiente.

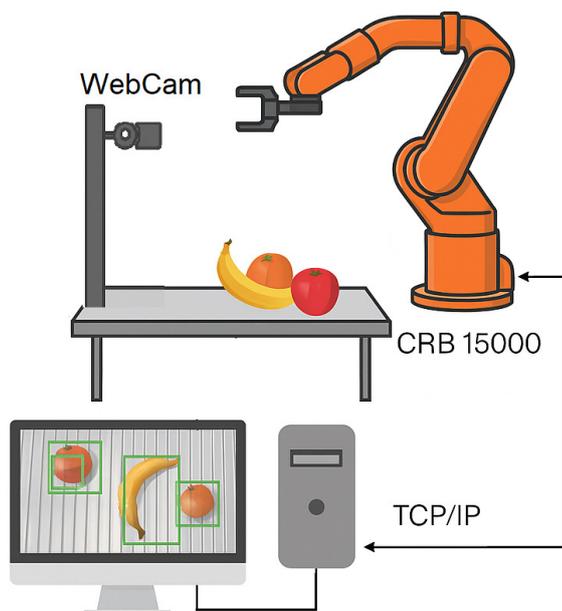


Figura 2: Arquitectura del sistema de visión.

2.1. Entrenamiento del modelo YOLO

Para permitir la detección automatizada de frutas en el entorno de trabajo del robot, se entrena un modelo de detección de

objetos basado en la arquitectura YOLOv8, utilizando la implementación proporcionada por la librería *ultralytics*. El objetivo es generar un modelo personalizado capaz de identificar tres clases: manzana, naranja y banana, a partir de imágenes capturadas en condiciones reales del laboratorio.

A continuación se describen los pasos necesarios para entrenar y generar el archivo *my_model.pt*:

1. **Recolección y etiquetado de datos:** El primer paso consiste en capturar un conjunto de imágenes en las que aparezcan claramente las frutas objetivo. Es importante incluir variaciones de iluminación, fondo y ángulo de cámara para que el modelo pueda generalizar mejor. Una vez obtenidas las imágenes, se procede a etiquetar manualmente cada una de ellas, es decir, indicar la ubicación y la clase de cada fruta presente. Esto puede realizarse con herramientas gráficas de etiquetado que exporten en formato YOLO, donde se guarda por cada imagen un archivo de texto con la información de los objetos detectados.
2. **Organización del dataset:** Las imágenes y sus etiquetas deben organizarse en carpetas separadas para el entrenamiento y la validación del modelo. Además, es necesario definir un archivo de configuración (*data.yaml*) que indique las rutas del dataset, el número de clases y los nombres asignados a cada clase (manzana, naranja, banana).
3. **Configuración del entorno:** Para ejecutar el entrenamiento se utiliza el lenguaje Python, junto con la librería *ultralytics*, que facilita la implementación y entrenamiento de modelos YOLO. Se deben instalar todas las dependencias necesarias, incluyendo *OpenCV*, *numpy* y *ultralytics*.
4. **Entrenamiento del modelo:** El modelo se entrena usando un conjunto base (por ejemplo, YOLOv8n) y se ajusta a las características del conjunto de datos recolectado. Durante el entrenamiento, el sistema ajusta internamente los pesos del modelo para que pueda reconocer correctamente las clases de frutas en imágenes nuevas. Este proceso puede llevarse a cabo durante varias iteraciones (épocas), y el resultado es un modelo personalizado capaz de detectar frutas con alta precisión.
5. **Generación del modelo final:** Una vez terminado el entrenamiento, se selecciona el mejor modelo generado (por ejemplo, *best.pt*) y se guarda como *my_model.pt*. Este archivo es el que se utilizará posteriormente en el sistema de visión para realizar la detección en tiempo real.

Para validar el desempeño del modelo entrenado, se realizaron pruebas utilizando imágenes no incluidas en el conjunto de entrenamiento ni en el de validación. Estas pruebas permiten evaluar la capacidad del modelo para generalizar y detectar correctamente las frutas en condiciones reales de operación. En particular, se analizaron escenarios con múltiples objetos, variaciones de iluminación y posiciones aleatorias sobre la superficie de trabajo. En la Figura 3, se presenta un ejemplo representativo en el que el modelo identifica correctamente una manzana, una banana y una naranja, demostrando una segmentación precisa de las frutas en la escena y una alta confianza en las predicciones.

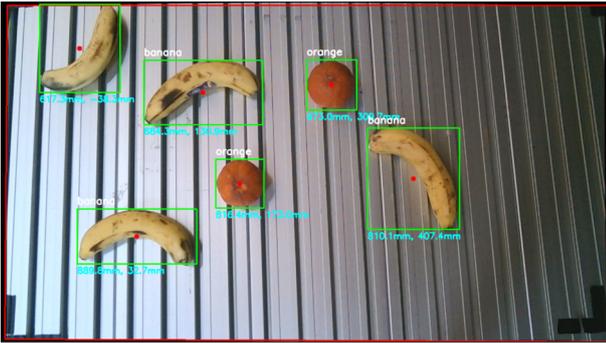


Figura 3: Resultado de la validación del modelo YOLOv8 entrenado.

2.2. Calibración de la cámara

La calibración de la cámara web es un proceso esencial para obtener una representación precisa del entorno en coordenadas reales, lo cual es fundamental en aplicaciones de robótica guiada por visión.

Para este fin, se utiliza un patrón de tablero de ajedrez impreso con una distribución de 9x6 esquinas internas, donde cada cuadrado tiene un tamaño físico conocido de 37 mm. Este patrón facilita la detección de puntos de referencia geométricos consistentes mediante el algoritmo *findChessboardCorners* de la librería OpenCV, que identifica automáticamente las intersecciones entre los cuadros blancos y negros en la imagen.

Se capturan un total de 30 imágenes, ubicando el tablero sobre una mesa (plano XY), manteniendo fija su altura (eje Z constante), pero variando su posición y orientación respecto a la cámara. Esto garantiza una cobertura espacial suficiente para una calibración robusta.

A partir de las imágenes capturadas desde distintos ángulos, se refinan las ubicaciones de las esquinas con el algoritmo *cornerSubPix* para mejorar la precisión subpíxel. Posteriormente, se emplea el algoritmo *calibrateCamera* para estimar la matriz intrínseca de la cámara (*mtx*), que contiene parámetros como la distancia focal y el punto principal, así como los coeficientes de distorsión óptica (*dist*) que corrigen deformaciones introducidas por la lente.

En la Figura 4, se muestra una de las imágenes empleadas durante este proceso, donde se visualizan las esquinas detectadas superpuestas sobre el tablero de ajedrez.



Figura 4: Proceso de calibración mediante el tablero de ajedrez.

Finalmente, los parámetros calculados se guardan para su utilización posterior en la corrección de las imágenes durante la etapa de análisis descrita en la sección de Resultados 3.

2.3. Comunicación y detección en tiempo real

Una vez calibrada la cámara y entrenado el modelo de detección, se desarrolla una interfaz de comunicación en tiempo real entre el sistema de visión artificial y el robot colaborativo ABB CRB 15000. Esta comunicación se implementa mediante el protocolo *TCP/IP*, permitiendo el envío de coordenadas en milímetros desde el sistema de visión al controlador del robot programado en *RAPID*.

La cámara web, previamente calibrada, captura imágenes del entorno. Estas imágenes son corregidas por distorsión óptica utilizando los parámetros almacenados en el archivo *parameter_calibration_robot41_720.npz*. A continuación, se aplica un modelo de detección previamente entrenado con YOLOv8 (*my_model.pt*), utilizando la librería *ultralytics*. El modelo reconoce las frutas objetivo (manzana, banana, naranja) y obtiene las coordenadas en píxeles de sus centros.

Para convertir estas coordenadas de píxeles a posiciones reales en milímetros, se emplea una matriz de homografía, calculada previamente a partir de referencias conocidas en el área de trabajo. Además, se aplican offsets físicos que corrigen la diferencia entre el origen de la cámara y el origen del sistema de coordenadas del robot.

El sistema permite al usuario ingresar, a través de terminal, la fruta que desea manipular. Una vez detectadas todas las instancias de dicha fruta en el campo de visión, se selecciona automáticamente la más cercana al robot, utilizando una métrica de distancia euclidiana, y se envía su posición al robot.

La comunicación se mantiene activa y cíclica. El robot, al recibir el mensaje "ready", solicita una nueva posición, la cual es calculada y enviada. Al finalizar una acción, se puede reiniciar el ciclo mediante el mensaje "finish", permitiendo al usuario cambiar la fruta objetivo en tiempo real sin reiniciar el sistema.

Esta arquitectura distribuida permite que el procesamiento visual se ejecute de forma continua, independiente del ciclo del robot, garantizando una operación fluida y flexible del sistema de pick & place.

El código de este sistema está disponible en: https://github.com/YEPEZ25/UC3M-Computer_Vision_EyesToHand.git.

3. Resultados

Tras realizar el proceso de calibración descrito en el capítulo anterior, se obtuvieron los parámetros intrínsecos y coeficientes de distorsión de la cámara. Estos parámetros son fundamentales para corregir las deformaciones presentes en las imágenes y permitir un análisis geométrico preciso.

Los resultados de la calibración son los siguientes:

- Resolución de la cámara:** La cámara utilizada para la calibración es una webcam con una resolución de 720p, lo que implica una resolución de 1280x720 píxeles.
- Tamaño de la rejilla del tablero:** El patrón de ajedrez utilizado tiene 9 columnas y 6 filas de cuadrados, con cada cuadrícula de 37 mm de lado. Este tamaño es fundamental para calcular la distancia focal y otros parámetros intrínsecos.

- **Matriz de la cámara:** La matriz de la cámara obtenida mediante el proceso de calibración es:

$$K = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 512,360 & 0,000 & 637,155 \\ 0,000 & 502,776 & 425,970 \\ 0,000 & 0,000 & 1,000 \end{bmatrix}$$

Donde:

- f_x y f_y son las distancias focales en píxeles,
- c_x y c_y son las coordenadas del centro óptico.

Esta matriz indica los parámetros intrínsecos de la cámara, como la distancia focal en píxeles y la ubicación del punto principal.

- **Coefficientes de distorsión:** Los coeficientes de distorsión calculados son los siguientes:

$$\begin{bmatrix} k_1 \\ k_2 \\ p_1 \\ p_2 \\ k_3 \end{bmatrix} = \begin{bmatrix} 0,03898881 \\ -0,02534507 \\ -0,00068034 \\ 0,00148653 \\ 0,00719257 \end{bmatrix}$$

Estos coeficientes corresponden a las distorsiones radiales k_1, k_2, k_3 y tangenciales p_1, p_2 que se utilizan para corregir las imágenes capturadas por la cámara antes de aplicar los algoritmos de detección.

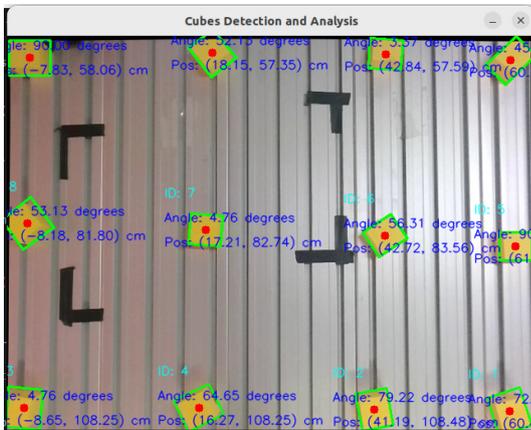


Figura 5: Medidas detectadas por el sistema de visión.

Se evalúan las desviaciones de las mediciones obtenidas por el sistema de visión con respecto a las mediciones reales. Para ello, se realizaron varias pruebas comparando las posiciones y orientaciones detectadas de los cubos en el entorno físico con sus valores reales medidos manualmente. Esta evaluación es crucial para determinar la precisión del sistema de visión y ajustar en consecuencia los algoritmos de calibración.

- **Configuración del experimento:** Se colocaron cubos amarillos en posiciones específicas dentro del campo de visión de la cámara. Las posiciones y orientaciones reales de estos cubos se midieron con herramientas de precisión.

- **Medición automática:** Las mismas posiciones y orientaciones de los cubos se detectaron mediante el sistema de visión (Figura 5), que procesa las imágenes capturadas para obtener las coordenadas y ángulos de cada cubo.

- **Cálculo de las desviaciones:** Las desviaciones en las posiciones (en centímetros) y los ángulos (en grados) se calcularon comparando los valores medidos automáticamente con los valores reales.

A continuación se presenta una tabla comparativa que muestra las desviaciones encontradas en una serie de mediciones (Figura 6):

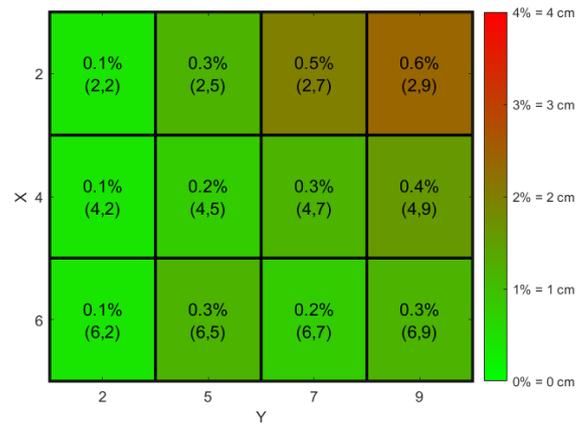


Figura 6: Comparación de las mediciones reales y las detectadas por el sistema de visión.

Los resultados muestran que las desviaciones entre las mediciones reales y las detectadas por el sistema de visión son mínimas, lo que indica una gran precisión en la detección de las posiciones y ángulos de los cubos. No obstante, se observan pequeñas variaciones que podrían deberse a factores como la distorsión óptica, errores en la calibración de la cámara o limitaciones en la resolución del sistema de visión.

Además del análisis con objetos patrón, se validó el sistema utilizando frutas reales dispuestas aleatoriamente. La cámara calibrada permitió estimar con precisión sus posiciones en milímetros, como se muestra en la Figura 7. Estos resultados confirman la capacidad del sistema para trabajar con objetos reales y refuerzan su aplicabilidad en entornos educativos y de automatización ligera.



Figura 7: Aplicación de Pick & Place utilizando la cámara calibrada.

4. Trabajos Futuros

Como proyección de este trabajo, se identifican varias oportunidades para extender y mejorar el sistema de visión artificial desarrollado, tanto en términos de precisión como de aplicabilidad a entornos más complejos o realistas. A continuación, se destacan algunas posibles líneas de trabajo futuro:

- Estimación de pose 3D y sensores de profundidad: Para mejorar la capacidad del sistema, se plantea incorporar estimación de orientación en 3D (pose 6DoF) y sensores de profundidad como cámaras RGB-D o Intel RealSense™. Esto permitiría al robot manipular objetos con orientación variable y obtener coordenadas espaciales más precisas sin depender únicamente de transformaciones planas.
- Ampliación del conjunto de objetos reconocibles: Entrenar el modelo de YOLO para detectar una mayor variedad de objetos industriales o educativos, e incluso permitir su reentrenamiento en laboratorio por parte de los estudiantes, reforzaría el valor pedagógico del sistema.
- Interfaz gráfica y realidad aumentada: Se propone desarrollar una interfaz gráfica que permita al usuario seleccionar objetos, visualizar su posición y monitorear la conexión con el robot. Su integración con plataformas de realidad aumentada como Unity facilitaría la interacción y permitiría superponer información visual sobre el entorno real, potenciando su uso en aplicaciones educativas y demostrativas.

Estas líneas de trabajo permitirían no solo mejorar el rendimiento del sistema, sino también explorar nuevas funcionalidades orientadas a aplicaciones educativas, de investigación o incluso industriales, manteniendo como principio el uso de tecnologías accesibles y replicables.

5. Discusiones y conclusiones

Los resultados obtenidos en el proceso de calibración y validación del sistema de visión artificial permiten identificar distintos factores que afectan la precisión espacial en la reconstrucción y visualización de objetos mediante realidad aumentada.

En primer lugar, los parámetros intrínsecos y extrínsecos derivados de la calibración con un patrón de ajedrez permitieron obtener la matriz de cámara y los coeficientes de distorsión necesarios para la reconstrucción tridimensional. Sin embargo, al realizar la validación empírica con la colocación de 12 cubos en posiciones reales conocidas dentro del campo de visión, se observaron errores relativos de posicionamiento que oscilan entre 0.1 % y 0.6 %, con una distribución no homogénea. Como se evidencia en la Figura 6, los errores tienden a incrementarse hacia las zonas periféricas del área visual, especialmente en la esquina superior derecha.

Este comportamiento es consistente con fenómenos típicos en calibraciones ópticas, donde las distorsiones radiales y tangenciales no corregidas de forma precisa pueden afectar la fidelidad geométrica en las regiones alejadas del centro. Además, una cobertura insuficiente del patrón de calibración sobre el total del campo visual puede limitar la precisión en la estimación de los parámetros de distorsión en dichas zonas.

Otro factor relevante es la dependencia del sistema respecto a las condiciones de iluminación, el enfoque y resolución de la cámara. Pequeñas variaciones en estos factores pueden alterar significativamente la calidad de detección de los puntos del patrón, afectando la robustez de la calibración y su generalización.

Con base en los resultados obtenidos, se proponen diversas mejoras para aumentar la precisión del sistema: utilizar patrones de calibración con mayor densidad de puntos y capturar más imágenes desde distintas perspectivas, especialmente en las esquinas del campo visual; asegurar una iluminación homogénea para evitar sombras o reflejos; validar cuantitativamente los resultados mediante mediciones físicas de referencia; aplicar técnicas avanzadas como bundle adjustment para reducir el error global de reproyección; e incorporar, si es necesario, transformaciones correctivas locales para compensar errores en zonas específicas del área visual.

En conjunto, estas acciones permitirían mejorar la precisión global del sistema, reforzando su utilidad para aplicaciones de realidad aumentada con requerimientos espaciales estrictos.

Agradecimientos

La investigación que ha conducido a este resultado ha recibido financiación de los proyectos “iRoboCity2030-CM, Robótica inteligente para ciudades sostenibles (TEC-2024/TEC-62) y “FotoArt5.0-CM, Laboratorios inteligentes para la ciencia del futuro”, (TEC-2024/TEC-308) financiados por los “Programas de Actividades I+D en Tecnologías de la Comunidad de Madrid”; y del proyecto “iREHAB: AI-powered Robotic Personalized Rehabilitation”, (DTS22/00105), financiado por el Instituto de Salud Carlos III (ISCIII) y cofinanciado con fondos FEDER de la UE.

Referencias

- Automation, A., 2023. Visión artificial aplicada a la industria. Accedido el 6 de mayo de 2025.
URL: https://www.aer-automation.com/wp-content/uploads/2023/01/Vision-Artificial_AERPaper.pdf
- Balanji, H. M., Turgut, A. E., Tunc, L. T., 2022. A novel vision-based calibration framework for industrial robotic manipulators. *Robotics and Computer-Integrated Manufacturing* 73, 102248.
- Chen, C.-S., Hu, N.-T., 2023. Eye-in-hand robotic arm gripping system based on machine learning and state delay optimization. *Sensors* 23 (3).
DOI: 10.3390/s23031076
- Enebase, I., Foo, M., Ibrahim, B. S. K. K., Ahmed, H., Supmak, F., Eyobu, O. S., 2021. A comparative review of hand-eye calibration techniques for vision guided robots. *IEEE Access* 9, 113143–113155.
DOI: 10.1109/ACCESS.2021.3104514
- Kaiser, M., Brusa, T., Bertsch, M., Wyss, M., Čuković, S., Meixner, G., Koch, V. M., 2024. Extrinsic calibration for a modular 3d scanning quality validation platform with a 3d checkerboard. *Sensors* 24 (5).
DOI: 10.3390/s24051575
- Oščádal, P., Heczko, D., Vysocký, A., Mlotek, J., Novák, P., Virgala, I., Sukop, M., Bobovský, Z., 2020. Improved pose estimation of aruco tags using a novel 3d placement strategy. *Sensors* 20 (17).
DOI: 10.3390/s20174825
- Torres Abril, P. C., Jara Moya, S. D., Torres Valverde, L. D., Arias Martínez, D. R., 2023. Systematic review of state-of-the-art applications in the field of computer vision. *Ciencia Digital* 7 (4), 26–53.
DOI: 10.33262/cienciadigital.v7i4.2710