

Jornadas de Automática

Sistema integral de percepción y control en tiempo real para la manipulación flexible con robots industriales

Pedro Martín-Martos, Ricardo Vázquez-Martín*

*Departamento de Ingeniería de Sistemas y Automática, Instituto de Investigación en Ingeniería Mecatrónica y Sistemas Ciberfísicos (IMECH.UMA),
Universidad de Málaga, Arquitecto Francisco Peñalosa, nº 6, 29071. Málaga, España.*

To cite this article: Martín-Martos, Pedro, Vázquez-Martín, Ricardo. 2026. Real-time perception and control system for flexible handling with industrial robots. *Jornadas de Automática*, 47.
<https://doi.org/10.17979/ja-cea.2026.47.13705>

Resumen

Este trabajo presenta un sistema de manipulación robótica flexible basado en visión artificial, orientado a aplicaciones industriales. El entorno propuesto consiste en un sistema de visión capaz de identificar objetos, localizarlos mediante la segmentación de instancias y estimar automáticamente la *pose* (posición y orientación) de la herramienta, lo que permite manipular el objeto mediante un brazo robótico. El sistema se estructura en una arquitectura modular sobre ROS 2, que integra componentes de percepción y de control del manipulador, así como una interfaz de usuario. El sistema de visión se basa en una cámara estereoscópica para capturar imágenes RGB-D desde la herramienta del manipulador y en un modelo YOLO de segmentación de instancias para identificar objetos y generar sus correspondientes máscaras, que, junto con la información de profundidad, permiten calcular de forma robusta un punto de agarre adecuado. Finalmente, el sistema incorpora un mecanismo de verificación posagarre. Los resultados obtenidos muestran un comportamiento robusto y en tiempo real, lo que valida el uso de técnicas de visión avanzada en entornos industriales no estructurados. El código se pone a disposición de la comunidad en el repositorio (<https://github.com/PedroMM03/semantic-segmentation-adaptive-manipulation>).

Palabras clave: Percepción y detección, Tecnología robótica, Manipuladores robóticos, Sistemas robóticos autónomos.

Real-time perception and control system for flexible handling with industrial robots

Abstract

This work presents the development of a flexible robotic manipulation system based on computer vision for industrial applications. The main goal is to identify and locate objects through instance segmentation and automatically compute their pose (position and orientation) to enable robotic-arm manipulation. The system follows a modular architecture built on the ROS 2 ecosystem, integrating visual perception components, manipulator control, and a graphical interface. A ZED Mini stereo camera captures RGB-D images, and a YOLO-based instance segmentation model identifies objects and generates corresponding masks. These, together with depth information, allow a robust grasping point to be computed. Finally, the proposed system incorporates a verification post-grasping mechanism. The results demonstrate robust and real-time performance, validating the application of advanced vision techniques in unstructured industrial environments. The code is publicly available in the repository (<https://github.com/PedroMM03/semantic-segmentation-adaptive-manipulation>).

Keywords: Perception and sensing, Robotics technology, Robot manipulators, Autonomous robotic systems

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

1. Introducción

En las últimas décadas, la industria ha evolucionado desde una producción en serie con características fijas y bien definidas hasta un entorno de fabricación de productos con una diversidad de bienes y modelos, lo que ha obligado a las líneas de producción a adaptarse a distintos tipos de productos y a cambios continuos en los procesos de fabricación. En este contexto, los sistemas de automatización deben perseguir objetivos de flexibilidad y de adaptación a las condiciones cambiantes del proceso productivo y del entorno. Un ejemplo centrado en las aplicaciones de robótica industrial es la necesidad de manipular distintos tipos de objetos ubicados en entornos desordenados, un problema conocido como *bin-picking* (recogida de objetos en contenedores) (Zhuang et al., 2021).

En la literatura se puede distinguir entre sistemas que estiman el agarre directamente sobre nubes de puntos, habitualmente orientados a pinzas o manos multidedo como efectores finales (ten Pas et al., 2017), o métodos geométricos (Zapata-Impata et al., 2019), y los basados en máscaras de segmentación, versátiles tanto para pinzas como para ventosas Liu et al. (2025). Asimismo, existen enfoques basados en características visuales para determinar la pose del objeto (Zhuang et al., 2021) frente a otros que emplean modelos neuronales para estimar directamente la pose del efector final (Utomo et al., 2021; Zhong et al., 2025) o coordinan acciones complejas de interacción en entornos desordenados (Zhao et al., 2025). A diferencia de estos, nuestra propuesta combina la precisión de la segmentación de instancias con un refinamiento geométrico mediante SVD para optimizar el agarre por succión, e integra, además, una verificación posagarre en una arquitectura modular en ROS 2, empleando un equipo embebido NVIDIA AGX Xavier para la manipulación en tiempo real.

Este artículo se organiza de la siguiente forma: en la Sección 2 se describe el equipamiento hardware y software empleado para implementar la solución propuesta, en la Sección 3 se detalla el sistema de visión que permite la detección de los objetos y cómo se resuelven los problemas de percepción acerca de la manipulación, en la Sección 4 se describe la tarea de manipulación y la interfaz de usuario. En la Sección 5 se muestran los experimentos realizados que validan el sistema propuesto y, finalmente, en la Sección 6 se resumen las conclusiones y las propuestas de trabajo futuro.

2. Materiales y método

2.1. Arquitectura de hardware

El núcleo computacional del sistema reside en una plataforma NVIDIA Jetson AGX Xavier, un equipo que ofrece un rendimiento de hasta 32 TeraOPS. Las prestaciones de este computador permiten ejecutar simultáneamente la inferencia de la red neuronal, el procesamiento de visión y la integración de la lógica de control del manipulador.

El subsistema de manipulación física consta de un brazo robótico colaborativo xArm6. Se trata de un manipulador de seis grados de libertad con una precisión de repetibilidad de $\pm 0,1$ mm y una carga útil de 5 kg. Para la interacción con los objetos, el robot incorpora una ventosa eléctrica oficial del fabricante, lo que permite prescindir de sistemas neumáticos externos.



Figura 1: Configuración *eye-in-hand* del efector final con cámara ZED Mini y ventosa eléctrica.

La percepción visual se obtiene mediante una cámara estéreo pasiva ZED Mini de Stereolabs, capaz de generar mapas de profundidad densos y capturar imágenes RGB de alta resolución. Para integrar el sensor y el actuador, se ha diseñado e impreso en 3D un soporte específico que permite acoplar ambos elementos en el extremo del manipulador en una configuración *eye-in-hand* (Figura 1). Esta disposición permite minimizar las oclusiones durante la maniobra de aproximación y garantizar una captura precisa del mapa de profundidad en el instante previo al agarre.

2.2. Entorno software y modelo de visión artificial

El ecosistema de software se ha implementado íntegramente sobre *Robot Operating System* (ROS 2 Foxy), lo que proporciona una arquitectura distribuida basada en nodos y en comunicaciones asíncronas. Para garantizar el aislamiento de procesos, la gestión de dependencias y la portabilidad del sistema, el espacio de trabajo se ha encapsulado en un contenedor Docker construido a partir de la imagen base oficial de NVIDIA JetPack 5.1.2. En este entorno se gestionan librerías críticas como CUDA, TensorRT y OpenCV.

La extracción de características semánticas de la imagen no se limita a la obtención de cajas delimitadoras (*bounding boxes*), sino que también requiere generar máscaras para calcular el punto de agarre en la superficie real del objeto. Por ello, se ha integrado el modelo Ultralytics YOLO11-seg en su versión *large*, una arquitectura de tipo *single-shot* que evoluciona el modelo fundacional (Redmon et al., 2016) mediante optimizaciones estructurales de su undécima versión (Khanam and Hussain, 2024). Se empleó el modelo preentrenado de YOLO11-seg proporcionado por la propia Ultralytics, que lo entrena sobre el conjunto de datos COCO.

Este modelo se seleccionó por su excelente compromiso entre precisión métrica (mAP_{50-95}) y baja latencia de inferencia, características críticas para operar en tiempo real en hard-

ware embebido. Además, se habilitó la función de segmentación con seguimiento (*tracking*), asignando un identificador único a cada instancia detectada. Esta capacidad resulta imprescindible para el rastreo temporal del objetivo a lo largo de las distintas etapas del ciclo de manipulación.

3. Sistema de percepción

El subsistema de percepción transforma la información sensorial *RGB-D* en consignas de posición y orientación precisas para el manipulador. Este nodo desarrolla tres tareas principales: segmentación de instancias, estimación de la pose de agarre y validación de la maniobra.

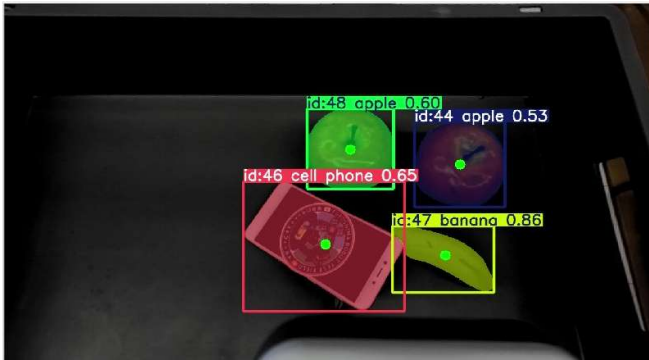


Figura 2: Captura de la ventana de procesamiento de imagen.

3.1. Preprocesamiento y segmentación de instancias

Para mejorar la robustez frente a condiciones lumínicas variables, la imagen capturada se transpone al espacio de color *HSV*. Se aplica una atenuación en el canal *V* (brillo), lo que mitiga los reflejos especulares en objetos metálicos o plásticos que podrían degradar la inferencia.

Posteriormente, el modelo genera máscaras a nivel de píxel. A diferencia de la detección convencional, este enfoque permite discriminar la geometría exacta del objeto objetivo incluso en situaciones de solape parcial (Figura 2). Mediante la tarea de seguimiento (*tracking*), el sistema asigna un identificador único a cada instancia, garantizando la persistencia temporal de los datos a lo largo del ciclo de trabajo.

Para cada instancia detectada se calcula su distancia respecto a la cámara a partir de la información del centroide de su máscara. Los objetos se almacenan en una lista ordenada por dicha distancia, lo que permite, en la manipulación posterior, priorizar los elementos más próximos cuando pertenecen a la clase objetivo.

3.2. Cálculo de la pose de agarre

La posición de agarre se determina proyectando el centroide geométrico de la máscara segmentada en el espacio tridimensional mediante el modelo de cámara *pinhole* (Hartley and Zisserman, 2004), utilizando los parámetros intrínsecos proporcionados por la calibración de fábrica de la cámara ZED, accesibles a través de su SDK. No obstante, dicho centroide puede corresponder a un punto de contacto inválido al situarse fuera del objeto o en una región de profundidad incoherente.

Algoritmo 1 Búsqueda local del centroide

Require: centroide c , mapa de etiquetas L

Ensure: centroide refinado c

```

1: if  $L(c) \neq \text{cluster\_mas\_cercano}$  then
2:    $P \leftarrow$  píxeles del cluster_mas_cercano
3:   for cada punto  $p \in P$  do
4:      $d(p) \leftarrow$  distancia euclídea entre  $p$  y  $c$ 
5:   end for
6:    $i \leftarrow$  índice de la menor distancia
7:    $c \leftarrow P(i)$ 
8: end if
9: return  $c$ 

```

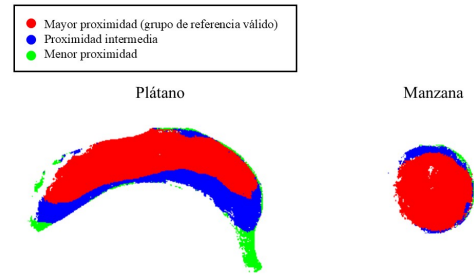


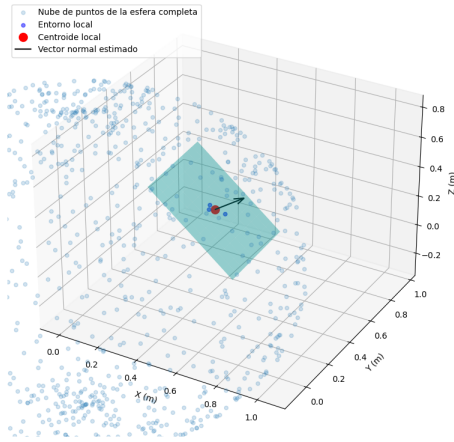
Figura 3: Agrupación por proximidad mediante el algoritmo K-Medias de los píxeles del mapa de profundidad.

Para evitar este problema, se implementa un algoritmo de refinamiento que verifica si el centroide pertenece a la máscara y al clúster de menor profundidad obtenido mediante *K-Medias*. En caso contrario, se aplica el procedimiento mostrado en el Algoritmo 1, que selecciona el píxel válido más cercano al centroide y lo utiliza para estimar el plano local y calcular la normal.

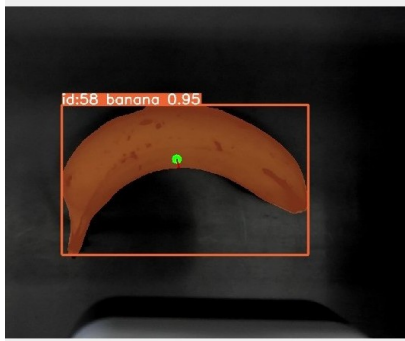
Posteriormente, para determinar la orientación, se analiza la nube de puntos local asociada al clúster validado (Figura 3). Sobre este conjunto se realiza una Descomposición en Valores Singulares (*SVD*), donde el vector singular asociado al valor de menor magnitud define la normal a la superficie en el punto de interés (Figura 4(a)). Dicha normal rige la aproximación perpendicular de la ventosa para maximizar la efectividad del vacío durante el contacto, obteniéndose la pose final que publica el nodo de visión (Figura 4(b)).

3.3. Verificación de la maniobra de agarre

Dada la configuración *eye-in-hand*, la cámara pierde la visibilidad del objeto durante la fase de contacto. Por ello, el sistema implementa una verificación *a posteriori*. Tras el ascenso del brazo, el nodo de visión analiza una Región de Interés (*ROI*) centrada en la ubicación previa del objeto. Si la segmentación confirma la ausencia de la clase seleccionada en dicha *ROI*, el agarre se clasifica como exitoso; en caso contrario, se activa un protocolo de reintento con ajuste de profundidad.



(a) Representación conceptual del cálculo del plano local y vector normal mediante SVD.



(b) Pose de agarre estimada sobre la escena segmentada.

Figura 4: Estimación de la orientación de agarre.

4. Arquitectura de control e interacción con el usuario

En esta sección se detalla la integración del sistema mediante una arquitectura distribuida basada en ROS 2. El control se gestiona mediante una máquina de estados finitos (*Finite State Machine*, FSM) que coordina de forma determinista los módulos de percepción y actuador.

4.1. Lógica de manipulación y transformación cinemática

El flujo de trabajo se estructura en una FSM que garantiza que el robot solo avance tras confirmaciones asíncronas de éxito en las tareas de visión o movimiento de cada etapa.

El control del manipulador se ejecuta en el espacio cartesiano de la base, pero los objetos se localizan inicialmente respecto al sistema de referencia de la cámara. Para la posición, se ha implementado una matriz de transformación homogénea estática T_{Cam}^{TCP} que relaciona el sistema de coordenadas de la cámara con el extremo del robot TCP (*Tool Center Point*). Según el montaje físico y las dimensiones del fabricante, la matriz resultante se define en la ecuación (1):

$$T_{Cam}^{TCP} = \begin{pmatrix} 0 & -1 & 0 & 0,071 \\ 1 & 0 & 0 & -0,023 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (1)$$

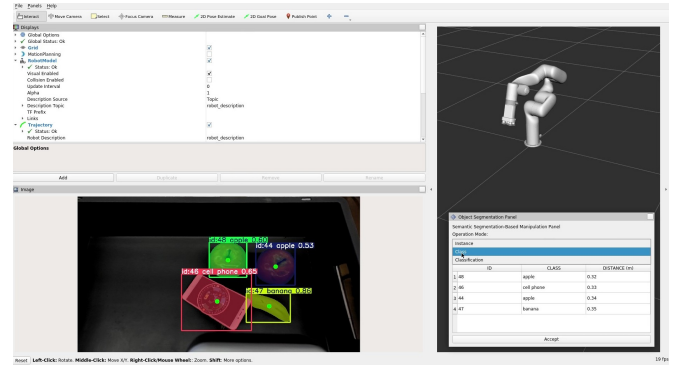


Figura 5: Interfaz de monitorización en RViz2 (modelo 3D del manipulador xArm6 sincronizado en tiempo real, imagen anotada de la cámara y panel de control).

Asimismo, la transformación del TCP a la base se obtiene a partir del árbol de transformadas en ROS 2 proporcionado por el nodo del controlador del robot. En cuanto a la orientación, se construye un sistema de referencia auxiliar a partir de la normal del objeto, que especifica la dirección de agarre. Esta información se convierte a ángulos de Euler XYZ para que el controlador del xArm6 ejecute los comandos de movimiento.

4.2. Interfaz de usuario e interoperabilidad de nodos

Como núcleo de la interacción con el usuario, se ha desarrollado un panel de control basado en el *framework* Qt, integrado como un *plugin* en RViz2 (Figura 5). Esta interfaz actúa como el orquestador del sistema y permite al usuario tanto monitorizar el estado del manipulador como seleccionar entre tres niveles de operatividad: *Instancia* (objeto único), *Clase* (todos los objetos de una misma clase) y *Clasificación Automática* (segregación autónoma en zonas predefinidas).

La arquitectura de software y el flujo de ejecución secuencial se detallan en la Figura 6. La interoperabilidad se fundamenta en el intercambio de mensajes estandarizados, en el que la interfaz coordina la activación de los nodos de percepción y control. Se emplean tipos de datos *geometry_msgs/msg/PointStamped* para la transmisión de coordenadas y *std_msgs/msg/Int16* para la sincronización de la máquina de estados (FSM). Mientras que las tareas críticas de movimiento se gestionan mediante servicios síncronos para garantizar la confirmación de recepción, los datos de monitorización utilizan tópicos con políticas de Calidad de Servicio (*QoS*) de tipo *Best Effort* para minimizar la latencia.

5. Experimentación y resultados

5.1. Procedimiento experimental

Se diseñó un escenario de clasificación automática de objetos para validar el desempeño del sistema integrado. Se seleccionaron elementos con geometrías diversas, presentes en el conjunto de datos de entrenamiento del modelo YOLO: una manzana (geometría curva y cuasi-esférica), un plátano (geometría cóncava) y un teléfono móvil (superficie plana con pendiente variable). El experimento permite evaluar la integridad de la arquitectura de software y la precisión en la estimación de la pose de agarre (Figura 7).

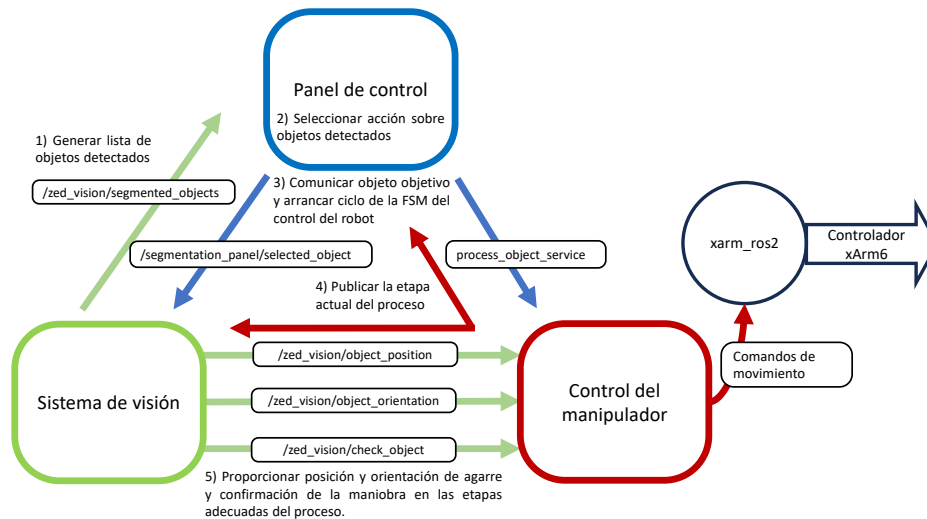


Figura 6: Arquitectura de comunicaciones en ROS 2 y diagrama de flujo del ciclo de trabajo.



Figura 7: Escenario de validación del sistema desarrollado.

El sistema se configuró en modo de clasificación automática. En la fase inicial, el manipulador se desplaza a una pose de inicio que permite obtener una visión global del área de trabajo; en esta etapa se identifican las instancias y se ordenan por proximidad a la cámara. El ciclo operativo se ejecutó de forma secuencial siguiendo las etapas de aproximación, cálculo de pose final, agarre y depósito en la zona asignada según la clase detectada (Figura 8).

Durante las pruebas, se verificó que la implementación de políticas de Calidad de Servicio (QoS) en ROS 2 garantiza una comunicación fluida y previene la pérdida de mensajes entre los nodos de percepción y control, incluso ante publicaciones simultáneas de imágenes y datos de profundidad.

5.2. Análisis de resultados

Los resultados obtenidos confirman que el algoritmo basado en SVD genera una normal de superficie estable en las tres geometrías analizadas. Asimismo, el mecanismo de verificación visual, basado en el análisis de la ROI tras la maniobra de elevación, demostró ser eficaz para la detección indirecta de fallos de agarre (Figura 9), permitiendo la activación correcta de reintentos automáticos.

Sin embargo, se ha observado que el desempeño del sistema es sensible a las limitaciones inherentes a la visión estereoscópica. Se identificó que la precisión del cálculo de la profundidad es menor en superficies homogéneas, como la carcasa lisa del teléfono móvil. En estos casos, la ausencia de características visuales dificulta la ejecución del algoritmo de correspondencia estéreo, como se aprecia en la Figura 10. Por último, aunque el preprocesamiento en el espacio de color HSV logra mitigar la presencia de reflejos, la sobreexposición intensa en entornos industriales puede degradar la calidad de las máscaras de segmentación, afectando colateralmente a la precisión en el cálculo del centroide del objeto.

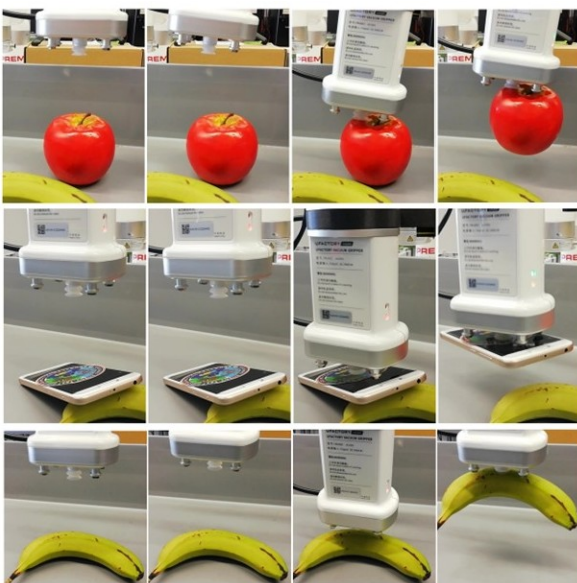


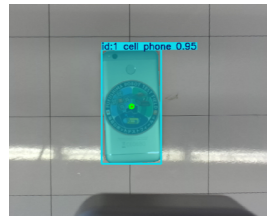
Figura 8: Secuencia de la maniobra de agarre para objetos de diversas clases.



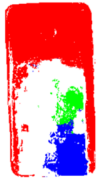
Figura 9: Sistema de verificación del agarre basado en ROI: la ausencia del objeto indica una captura correcta.



(a) Teléfono móvil con carcasa de color homogéneo.



(b) Teléfono móvil con adhesivo con textura.



(c) Píxeles agrupados del mapa de profundidad con fondo homogéneo.



(d) Píxeles agrupados del mapa de profundidad del teléfono móvil con textura.

Figura 10: Problema de cálculo de profundidad para superficies homogéneas.

6. Conclusiones

En este trabajo se ha presentado un sistema integral de manipulación robótica flexible que combina la segmentación de instancias y el control basado en visión en ROS 2. La arquitectura desarrollada permite la identificación y localización de objetos en tiempo real y la integración de modelos de aprendizaje profundo con algoritmos de ajuste geométrico, y ha demostrado ser una solución eficaz para el guiado de manipuladores industriales en entornos no estructurados. El uso de SVD para estimar la normal de la superficie se ha mostrado como un método robusto para garantizar la efectividad del agarre por vacío, incluso en objetos con geometrías complejas.

La validación experimental confirma la estabilidad del sistema frente al ruido en el mapa de profundidad y durante la manipulación de objetos no convexos. La lógica de corrección de centroide y el filtrado mediante K-Medias han permitido reducir significativamente la tasa de error en la selección de puntos de contacto. Asimismo, la implementación de un mecanismo de verificación visual basado en la ROI ha dotado al sistema de capacidades de monitorización indirecta, lo que facilita la detección de fallos y la ejecución automática

de reintentos. Esto ha incrementado la fiabilidad del ciclo de trabajo en tareas de *bin-picking*.

Como trabajo futuro, se plantea mejorar la adquisición de datos mediante la sustitución del sensor estéreo por cámaras de luz estructurada o infrarrojas para mitigar las limitaciones asociadas a superficies sin textura. En el ámbito del control, se prevé integrar *MoveIt2* para la planificación de trayectorias con evasión dinámica de obstáculos. Desde el punto de vista de la percepción, se explorará el uso de técnicas de realce local, como *CLAHE*, así como la incorporación de modelos *SAM* para una segmentación agnóstica de objetos. Finalmente, se estudiará el potencial de los modelos de lenguaje y visión de gran escala (*LVL*) para dotar al sistema de capacidades de aprendizaje incremental y razonamiento visual ante nuevas clases de objetos.

Agradecimientos

Este trabajo ha sido realizado parcialmente gracias al apoyo del Ministerio de Asuntos Económicos y Transformación Digital del Gobierno de España, en el marco del proyecto 5GVEC, con referencia TSI-065100-2022-001.

Referencias

- Hartley, R., Zisserman, A., 2004. Multiple View Geometry in Computer Vision, 2nd Edition. Cambridge University Press.
- Khanam, R., Hussain, M., 2024. YOLOv11: An overview of the key architectural enhancements. arXiv preprint arXiv:2408.00714.
- Liu, Y., Graf, C., Spies, M., Keuper, M., 2025. Segment any repeated object. In: International Conference on Robotics and Automation. pp. 16042–16049. DOI: 10.1109/ICRA55743.2025.11128131
- Redmon, J., Divvala, S., Girshick, R., Farhadi, A., 2016. You only look once: Unified, real-time object detection. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 779–788. DOI: 10.1109/CVPR.2016.91
- ten Pas, A., Gualtieri, M., Saenko, K., Platt, R., 2017. Grasp pose detection in point clouds. The International Journal of Robotics Research 36 (13-14), 1455–1473. DOI: 10.1177/0278364917735594
- Utomo, T. W., Cahyadi, A. I., Ardiyanto, I., 2021. Suction-based grasp point estimation in cluttered environment for robotic manipulator using deep learning-based affordance map. International Journal of Automation and Computing 18 (2), 277–287. DOI: <https://doi.org/10.1007/s11633-020-1260-1>
- Zapata-Impata, B. S., Gil, P., Pomares, J., Torres, F., 2019. Fast geometry-based computation of grasping points on three-dimensional point clouds. International Journal of Advanced Robotic Systems 16 (1), 1729881419831846. DOI: 10.1177/1729881419831846
- Zhao, M., Zuo, G., Yu, S., Luo, Y., Liu, C., Gong, D., 2025. Language-guided category push-grasp synergy learning in clutter by efficiently perceiving object manipulation space. IEEE Transactions on Industrial Informatics 21 (2), 1783–1792. DOI: 10.1109/TII.2024.3488774
- Zhong, X., Gong, T., Zhong, X., Liu, Q., Hu, H., 2025. Region-aware 6D grasping for industrial bin-picking: A Sim2Real label self-generation and hybrid evaluation framework. In: International Conference on Intelligent Robots and Systems. pp. 9448–9455. DOI: 10.1109/IRoS60139.2025.11247129
- Zhuang, C., Wang, Z., Zhao, H., Ding, H., 2021. Semantic part segmentation method based 3D object pose estimation with RGB-D images for bin-picking. Robotics and Computer-Integrated Manufacturing 68, 102086. DOI: <https://doi.org/10.1016/j.rcim.2020.102086>