Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Cost-Effective Vision-Based Bin Picking System for Cluttered Environments

Daniel Charles Garza

A00836561@tec.mx

Tecnologico de Monterrey https://orcid.org/0009-0008-6561-7876

Ruben Morales-Menendez

Tecnologico de Monterrey https://orcid.org/0000-0003-0498-1566

Antonio Vallejo Guevara

Tecnologico de Monterrey

Federico Guedea Elizalde

Tecnologico de Monterrey

Ana Gabriela Sepúlveda Garza

Tecnologico de Monterrey

Research Article

Keywords:

Posted Date: November 26th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11587292/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Cost-Effective Vision-Based Bin Picking System for Cluttered Environments

Daniel Charles Garza[1], Rubén Morales-Menéndez[1*], Antonio Vallejo Guevara[1], Federico Guedea Elizalde[1], Ana Gabriela Sepúlveda Garza[1]

[1*]School of Engineering and Sciences, Tecnológico de Monterrey, Av. Eugenio Garza Sada 2501 Sur, Col. Tecnológico, Monterrey, 64849, Nuevo León, Mexico.

Contributing authors: a00836561@tec.mx;
*Corresponding author(s). E-mail(s): rmm@tec.mx;

Abstract

Picking elongated, reflective industrial parts from cluttered bins remains difficult for robots, and existing hybrid vision systems require additional sensors or a three-dimensional (3D) model of each part. This work presents a vision system that estimates the position and orientation of aluminium profiles in a cluttered bin from a single colour-and-depth image, without a 3D model of the part. A deep network first segments the parts, and classical geometric methods then estimate the pose of the selected part. Comparing these methods for each pose component showed that none was the most accurate on all of them. Therefore, the proposed hybrid estimator takes each component from the method that estimates it best. The hybrid matched or outperformed every baseline on each component and roughly halved the composite error relative to the best alternative. Robot picking and parts of other shapes and dimensions remain to be evaluated. This per-component comparison can serve as a practical guide for selecting geometric methods for similar industrial parts.

Keywords: Bin picking, 6D pose estimation, Instance segmentation, Point clouds, Principal component analysis, Random sample consensus

1 Introduction

Robotic bin-picking — the retrieval of randomly arranged parts from unstructured containers — remains an open challenge in industrial automation: its deployments are not yet routine and widespread [3], and vision-based grasping methods still show clear limitations in cluttered, occluded scenes [7]. The task requires estimating the six-degrees-of-freedom (6-DoF) pose of each target object in the presence of clutter, partial occlusion, and inter-part contact. In the BOP benchmark [13, 14], object symmetries and similarities remain open problems [13], and the industrial ITODD dataset, whose glossy parts are not always captured in the depth channel, yielded the lowest score of the top-ranked method in the 2020 challenge [14]. Industrial parts — machined brackets, extruded profiles, stamped sheet components — share homogeneous surfaces with little discriminative appearance information, the regime targeted by texture-less pose estimation methods [12, 27]; their elongated geometries introduce rotational near-symmetries, and strongly reflective surfaces complicate synthetic-to-real transfer [27]. Estimating the pose of such parts with a single commodity RGB-D sensor, without a 3D

object model or object-specific pose networks, is the problem this work addresses.

Two research trajectories dominate the state of the art. In end-to-end deep learning (DL), an artificial neural network (ANN) estimates the 6-DoF pose, either by regressing it directly [29, 31] or through intermediate keypoints [23]; these methods require object-specific training data, either pose-annotated images or synthetic renderings of a 3D model [27], and must be retrained for each new part. Classical geometric estimators on point clouds — principal component analysis (PCA), plane fitting by random sample consensus (RANSAC) [9], point pair features (PPF) [6, 28], and iterative closest point (ICP) refinement [1] — require no per-object training, but differ in the pose components they constrain and in the inputs they require. PCA is commonly employed by model-free bin-picking approaches to estimate the 3D translation and the in-plane orientation [5]; a RANSAC plane normal recovers the surface tilt but carries no information about the rotation about itself; and PPF matches the scene against a 3D model of the object [6]. Hybrid frameworks pair a deep segmentation front-end with a single classical pose estimator [20, 21], and point-cloud instance segmentation followed by a learned pose network has also been applied to industrial bin-picking [32].

These pipelines treat pose estimation monolithically: a single estimator produces all 6-DoF and is evaluated through aggregate metrics such as ADD [12], the 5 cm/5° criterion [26], which counts a pose as correct when its translation and rotation errors are below 5 cm and 5°, respectively, or VSD [13]. Aggregate metrics obscure *where* the error originates, although the geometry of each primitive suggests a division of labour for elongated parts: the dominant axis recovered by PCA constrains the in-plane rotation, whereas the normal of a RANSAC plane fit to the top face constrains the tilt. A per-axis decomposition of pose error across competing classical estimators, on real cluttered industrial data, has not, to the best of the authors' knowledge, been reported. In addition, prior hybrid systems require additional sensing or 3D models: [21] relies on two cameras, a robot approach motion, and scanned 3D models, and [20, 32] require a 3D model of each part.

This paper proposes a cost-effective hybrid framework that addresses both gaps with a single fixed Intel RealSense D456 RGB-D camera; the target manipulator is a Fairino FR10, and the evaluation ends at the pose of the selected part in the camera frame. The contributions are as follows. *First*, a per-axis benchmark of six pose estimation pipelines — PCA, RANSAC, PPF, and their ICP-refined counterparts — on real cluttered bin-picking data. *Second*, an *axis-decoupled* hybrid estimator that takes the translation as the centroid of the cleaned point cloud, R_z from PCA, and R_x, R_y from a RANSAC plane fit, assigning each pose component to the method with the lowest error for it in the baseline comparison. *Third*, a comparison of five segmentation architectures: Mask R-CNN [10], Mask R-CNN with PointRend [19], and the YOLOv8, YOLO11, and YOLO26 segmentation variants [15–17]. *Fourth*, a single-sensor pipeline, from one RGB-D frame to the pose of the selected part, that requires neither a 3D object model nor a robot approach motion and runs on sensing and compute hardware with an approximate cost of USD 1,600, excluding the manipulator (Table 2).

Section 2 reviews related work, Section 3 describes the proposed pipeline, Section 4 presents the experimental setup, Section 5 reports and discusses the results, and Section 6 concludes this research.

2 Related Work

Related work is reviewed along four lines: instance segmentation, classical geometric pose estimation, DL pose estimation, and hybrid frameworks.

2.1 Instance Segmentation for Industrial Vision

Instance segmentation provides per-object pixel masks that can be back-projected to extract isolated point clouds, and it serves as the front-end of hybrid bin-picking systems [20, 21]. Two architectural families dominate. The two-stage family is represented by Mask R-CNN [10], which extends the Faster R-CNN detector with a mask branch that runs in parallel with box regression, and by PointRend [19], which re-predicts

Table 1 Acronyms used throughout this work.

Acronym	Definition
6-DoF	Six Degrees of Freedom
ADD	Average Distance of model points
ADI	Average Distance for objects with Indistinguishable views
AP	Average Precision
BOP	Benchmark for 6D Object Pose estimation
COCO	Common Objects in Context
CPU	Central Processing Unit
GPU	Graphics Processing Unit
ICP	Iterative Closest Point
IoU	Intersection over Union
LR	Learning Rate
mAP	mean Average Precision
MAE	Mean Absolute Error
PCA	Principal Component Analysis
PPF	Point Pair Features
RANSAC	Random Sample Consensus
RGB	Red, Green, Blue
RGB-D	Red, Green, Blue plus Depth
SDK	Software Development Kit
SGD	Stochastic Gradient Descent
SOR	Statistical Outlier Removal
VSD	Visible Surface Discrepancy
YOLO	You Only Look Once

mask labels at adaptively selected, uncertainty-driven points through iterative subdivision, concentrating computation near object boundaries. The single-stage YOLO family, in its segmentation variants YOLOv8 [16], YOLO11 [15], and YOLO26 [17], predicts per-instance mask coefficients over a shared set of prototype masks from dense single-stage heads; YOLOv8 uses an anchor-free head, and YOLO26 further removes non-maximum suppression and the distribution focal loss for end-to-end inference. Whether the single-stage design trades mask precision for inference speed on industrial imagery is examined in Section 5.1. Also, while these architectures have been benchmarked extensively on the COCO dataset of natural images [22], to the best of the authors' knowledge, no comparison spanning Mask R-CNN, PointRend, and the YOLOv8, YOLO11, and YOLO26 segmentation variants has been reported on cluttered bin-picking imagery of weak-texture elongated industrial parts. The present work conducts that comparison as an enabling step toward the pose stage.

2.2 Classical Geometric Pose Estimation on Point Clouds

Classical geometric estimators recover object pose directly from point cloud data, without object-specific training, and four primitives are considered in this work. PCA, applied to the masked point cloud, recovers the dominant geometric axis of the part through the eigendecomposition of the covariance matrix; in bin-picking, it is commonly employed by model-free approaches to estimate a 4D pose, composed of the 3D translation and the in-plane orientation [5]. Plane fitting by RANSAC [9], a robust model-fitting paradigm tolerant to a large fraction of gross errors, recovers the surface tilt of a flat or quasi-planar part through the normal of the fitted plane; however, a plane normal carries no information about rotation about itself, so the in-plane rotation remains unconstrained. PPF, introduced by [6] and improved by [28], match scene point pairs against a hash table built from a 3D model of the object through a Hough-like voting scheme; the method is strong on free-form rigid objects, but its recognition rate and run time depend on the sampling distance chosen for model and scene [6]. ICP [1] is used as a refinement stage after a coarse estimator [20, 32]; because the original algorithm converges monotonically to the nearest local minimum of a mean-square distance metric [1], its result depends on the quality of the initial pose supplied by the coarse estimator. However, the hybrid frameworks reviewed in Section 2.4 report pose accuracy through aggregate metrics such as ADD [12], 5 cm/5° [26], and VSD [13], which summarize translational and rotational errors in a single score and do not indicate which estimator is best suited to each degree of freedom.

2.3 Deep Learning for 6D Pose Estimation

A parallel line of research trains ANNs to estimate the 6-DoF pose from RGB or RGB-D input. Direct-regression [31], keypoint-based [11, 23], dense-fusion [29], and implicit-orientation [27] approaches all report strong performance on public benchmarks, ranging from textured household objects (YCB-Video) to texture-poor objects (LineMOD, Occlusion LineMOD) and, in the

case of [27], the texture-less objects of T-LESS. However, three structural limitations make them unsuitable for the present industrial weak-texture regime: they require per-object annotated datasets or synthetic renderings of an accurate 3D model [11, 27]; their accuracy on reflective metallic parts outside public benchmarks has not been established, and synthetic-to-real transfer on strongly reflective objects is itself reported as challenging, as is the handling of long, thin objects [27]; and they impose a retraining cost whenever a new part type is introduced. These limitations motivate keeping the pose-estimation stage classical and interpretable, while leveraging DL only at the segmentation front-end.

2.4 Hybrid Frameworks and the Per-Axis Gap

The hybrid trajectory pairs a deep segmentation front-end with a pose estimator that operates on the segmented point cloud, typically a classical geometric one, inheriting the robustness of the learned segmentation while retaining the interpretability of the geometric back-end. [20] combined instance segmentation (Mask R-CNN or RetinaMask) with PPF voting on the segmented depth points, ranking second overall and first among methods averaging below 1 s per image in the BOP Challenge 2020 [14]. [21] integrated Mask R-CNN with the improved PPF of [28], achieving 97.37% on the 5 cm/5° metric across the nine objects of a custom dataset. [32] segmented the scene point cloud with a PointNet++-based instance segmentation network and predicted each instance pose with a convolutional network applied to projected depth and normal feature maps, followed by ICP refinement; the network was trained on a physically simulated synthetic dataset built from scanned models of four industrial parts. Two limitations are common to all three. First, pose estimation is treated *monolithically*: a single estimator produces all 6-DoF, and methods are compared through aggregate metrics. Although [21] also reports the distribution of the per-axis errors of its own method, none of these works compares estimators per pose component or assigns different estimators to different DoF. Second, each of these systems requires additional sensing or a 3D object model: [20] matches a 3D model of each object; [21] requires a global top-view camera plus an eye-in-hand camera and a robot approach motion, together with scanned 3D models of the objects; and [32] requires high-precision scanned 3D models of the parts to synthesize its training data, and its accuracy is reported only through the aggregate ADI metric. Whether a single fixed RGB-D sensor, without a 3D object model, combined with an axis-decoupled hybrid estimator can recover the pose of weak-texture elongated parts is the question this work addresses.

3 System Architecture

The proposed bin-picking system is described following the data flow from a single RGB-D capture to the 6-DoF pose of a pickable part. The hardware is deliberately minimal—a fixed RGB-D camera, a commodity laptop, and a collaborative manipulator—and the pipeline operates on a single capture, without multi-frame fusion, additional sensors, or robot approach motions.

3.1 System Overview

The proposed system is built around a single fixed Intel RealSense D456 RGB-D camera, an IP65-rated variant of the D455 within the D400 active-stereo family [18, 24], mounted at a height of 2 m above the floor and centered above a 1 m × 1 m × 1 m container that holds the target parts. The container is open at the top and on two of its sides, so that the 1150 mm profiles extend beyond its walls. A Fairino FR10 collaborative manipulator is the target manipulator of the system, Fig. 1; however, robot execution is outside the experimental scope of this work, which ends at the estimated pose of the selected part. No second camera is used and no eye-in-hand approach motion is performed; the pose estimator operates on the same fixed top-down view that drives segmentation.

The processing pipeline comprises five sequential stages, summarized in Algorithm 1. First, a single aligned RGB-D frame is captured and its depth holes are filled by nearest-neighbour inpainting. Second, an instance segmentation network is run on the RGB image, producing a set of candidate object masks with associated confidence scores. Third, these candidates are filtered through a three-criterion cascade, and a

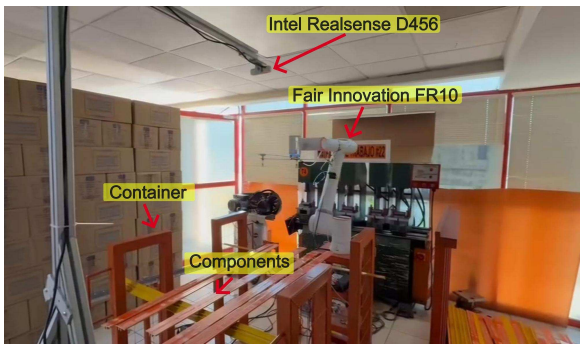


Fig. 1 Hardware setup: RealSense D456 camera at 2 m, container open at the top and on two sides, and Fairino FR10 manipulator.

single pickable instance is selected via a top-of-pile heuristic. Fourth, the selected mask is back-projected through the depth map to recover an isolated 3D point cloud, from which residual outliers are removed. Fifth, the 6-DoF pose of the object is estimated with the proposed axis-decoupled hybrid scheme, in which the translation is taken as the centroid of the cleaned point cloud, the in-plane rotation R_z is obtained from PCA on the same cloud, and the out-of-plane rotations R_x, R_y are obtained from a RANSAC plane fit. In deployment, a calibrated rigid transform maps the camera-frame pose to the robot base frame. The remainder of this section describes each stage in detail; the hyperparameters used throughout are summarized in Table 3, and the sensing-and-compute hardware configuration is reported in Table 2.

3.2 RGB-D Acquisition and Calibration

The RealSense D456 is operated through the manufacturer SDK at a resolution of 1280×720 for both the RGB and depth streams, and the depth frame is aligned to the RGB frame through the alignment function of the SDK, so that each depth sample is placed in the RGB image coordinate frame. A single frame is captured per estimation cycle; no temporal averaging or multi-frame fusion is applied. Reflective metallic surfaces and inter-part occlusion produce localized depth dropouts in the raw frame, which appear as zero-valued pixels inside otherwise valid object regions. These holes are filled by a nearest-neighbour inpainting step based on the Euclidean distance transform, in which each invalid pixel is assigned the depth value of its nearest valid neighbour; the filling is global across the image and runs in a single pass. Camera intrinsics are read from the factory calibration available through the SDK, and no per-deployment recalibration is applied. For deployment, the system requires a fixed rigid transform between the camera frame and the robot base frame, established once at installation through an extrinsic camera-to-robot calibration; because both the camera and the bin are stationary in the world frame, this transform remains valid for all subsequent picks. However, no robot picks were executed in this work, so the transform is not part of the reported evaluation, and all poses are reported in the camera frame.

3.3 Instance Segmentation Stage

The segmentation stage is treated as a pluggable component, and five candidate architectures are compared in this work: Mask R-CNN [10], Mask R-CNN augmented with the PointRend mask head [19], and three single-stage YOLO segmentation variants: YOLOv8m-seg [16], YOLO11m-seg [15], and YOLO26m-seg [17]. All five are trained on the same dataset: a master split is performed on the source images first, and six augmentation pipelines are applied only to the training partition, so that all models share identical validation and test sets. Also, all five share COCO-pretrained initialization, a batch size of 24, a maximum of 300 epochs, and an early-stopping patience of 30 epochs, while the optimizer, the learning-rate schedule, and the input resolution follow the conventions of each framework. The dataset construction, the augmentation policy, and the training configuration of each model are detailed in Section 4. Regardless of architecture, the segmentation stage outputs a set of N candidate masks $\{M_i\}_{i=1}^{N}$ with associated confidence scores $\{s_i\}$, of which only one is ultimately consumed by the pose stage. In the pose experiments of this work, the masks are produced by Mask R-CNN, the model selected in Section 5.1, and all pose estimation methods operate on the same selected mask.

The reduction of this candidate set to a single pickable instance is performed by a filtering cascade of three sequential criteria. First, masks

Table 2 Sensing and compute hardware with approximate purchase cost (USD), excluding the manipulator.

Component	Specification	Approx. cost
Depth camera	Intel RealSense D456	$600
Laptop	MSI Cyborg 15	$1,000
GPU	NVIDIA RTX 4060 Laptop (8 GB)	incl.
CPU	Intel Core i7, 12th gen.	incl.
RAM	32 GB	incl.
OS	Ubuntu 24.04 (WSL2)	Free
Total (sensing + compute)		**∼$1,600**

with confidence below $\tau_{\text{conf}} = 0.7$ are discarded to remove weakly supported detections. Second, masks with pixel area below $A_{\min} = 15{,}000$ pixels are rejected; this criterion is intended to suppress fragments of severely occluded parts and far-field clutter. Third, masks with solidity below $\sigma_{\min} = 0.7$ — where solidity is defined as the ratio of the area enclosed by the outer contour of the mask to the area of its convex hull — are rejected; this criterion targets elongated profiles in dense piles, where partial overlap between adjacent instances produces masks with concave bites that could distort the principal axis estimated by PCA in the point-cloud stage. Among the masks surviving the cascade, a single instance is selected via a top-of-pile heuristic: the candidate with the lowest mean depth across its mask region is chosen, which corresponds to the part physically closest to the camera and therefore on top of the pile. A similar criterion, the highest mean height of the instance, is used to select the topmost part in a recent industrial bin-picking system [32].

3.4 Point Cloud Extraction and Preprocessing

The selected mask is back-projected through the aligned depth map to recover a 3D point cloud restricted to the chosen instance. For each pixel (u, v) inside the mask with depth value d, the corresponding camera-frame point is computed using the standard pinhole model with the intrinsics (f_x, f_y, c_x, c_y) read from the SDK:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} (u - c_x)\, d/f_x \\ (v - c_y)\, d/f_y \\ d \end{bmatrix}. \tag{1}$$

The resulting raw cloud $\mathcal{P}^{\text{raw}}$ contains residual artefacts from two sources: edge-bleed at mask boundaries, where the mask covers pixels whose depth values belong to neighbouring objects or to the bin floor; and isolated noise spikes on the reflective metallic surfaces. Both are removed with a k-nearest-neighbour Statistical Outlier Removal (SOR) step, as implemented in the Point Cloud Library [25] and re-implemented here with a k-nearest-neighbour search in scikit-learn. For each point, the mean Euclidean distance to its k nearest neighbours is computed, and points whose mean distance exceeds $\mu + \alpha\sigma$ — where μ and σ are the global mean and standard deviation of these per-point distances and α is a fixed multiplier — are discarded. The implementation uses $k = 20$ and $\alpha = 2.0$. No voxel downsampling or surface smoothing is applied; the cleaned cloud $\mathcal{P}$ is passed directly to the pose estimation stage.

3.5 Axis-Decoupled Hybrid Pose Estimation

The pose stage estimates the full 6-DoF pose $(t_x, t_y, t_z, R_x, R_y, R_z)$ of the selected instance by assigning each component to the geometric primitive that yields the lowest error for it in the per-axis comparison of Section 5.2; as described below, this assignment is also consistent with the geometry of each primitive.

Translation.

The translation (t_x, t_y, t_z) is taken as the centroid of the cleaned cloud $\mathcal{P}$, that is, the mean of its points, and therefore coincides with the translation of the PCA estimator. Because $\mathcal{P}$ is restricted to the selected instance and cleaned by SOR, its centroid is less affected by the edge-bleed and noise artefacts described in Section 3.4; since the cloud covers only the visible surface of the part,

the centroid corresponds to the centre of that visible surface. Among the six baseline methods, PCA yields the lowest translation errors (Table 7).

In-plane rotation R_z via PCA.

The principal component analysis of $\mathcal{P}$ yields the eigendecomposition of its covariance matrix; the eigenvector $\mathbf{v}_1$ associated with the largest eigenvalue is the dominant geometric axis of the part, which for elongated profiles aligns with the part's length. The Euler angles are extracted from the resulting rotation matrix with a fixed convention: the axis of the matrix with the largest component along the camera optical axis is taken as the surface normal, from which R_x and R_y are computed via $\arctan 2$; the long axis is the remaining axis best aligned with the dominant axis of the cloud, and R_z is the angle of its projection onto the image XY plane, normalised to $[0°, 180°)$. The latter convention is required because elongated profiles exhibit an exact 180° rotational symmetry about the z-axis: the head and tail of the part are physically interchangeable for the picking task. Accordingly, all R_z errors reported in this work use the symmetry-aware metric

$$\Delta R_z = \min\big(|R_z^{\text{est}} - R_z^{\text{gt}}|,\ 180° - |R_z^{\text{est}} - R_z^{\text{gt}}|\big), \tag{2}$$

which avoids penalizing flips that have no operational consequence.

Out-of-plane rotations R_x, R_y via RANSAC.

The surface tilt of the part is recovered from a RANSAC plane fit [9] to the cleaned cloud $\mathcal{P}$, with a distance threshold of 1.5 mm, three points per sample, and 1000 iterations (Table 3). The unit normal $\hat{\mathbf{n}}$ of the fitted plane is taken as the surface normal, from which R_x and R_y are extracted with the same $\arctan 2$ convention. RANSAC is assigned to this sub-task because the part presents a near-planar top face to the camera, and the consensus step of the plane fit discards the residual outliers that survive the SOR stage. However, a plane normal carries no information about the rotation about itself. In a pure-RANSAC pipeline, the in-plane axes of the estimated frame are constructed from a fixed auxiliary vector, so the resulting R_z is restricted to the image axes (approximately 0° or 90°) regardless of the actual part orientation; in the pose evaluation set, it returned values close to 0°/180° for every retained instance (Section 5.2). This is precisely the deficit that PCA repairs in the proposed scheme, and it is the structural motivation for axis decoupling.

Algorithm 1 Axis-decoupled hybrid pose estimation pipeline.

Require: RGB image I, aligned depth map D, intrinsics $\mathbf{K}$
Ensure: 6-DoF pose $(t_x, t_y, t_z, R_x, R_y, R_z)$ of the selected instance in the camera frame, or $\varnothing$ if no valid instance is found

$D \leftarrow \textsc{DepthHoleFill}(D)$ ▷ nearest-neighbour inpainting
$\{(M_i, s_i)\} \leftarrow \textsc{Segment}(I)$ ▷ instance segmentation
$\mathcal{C} \leftarrow \{M_i : s_i \geq \tau_{\text{conf}},\ |M_i| \geq A_{\min},\ \text{solidity}(M_i) \geq \sigma_{\min}\}$
if $\mathcal{C} = \emptyset$ **then**
  return $\varnothing$
end if
$M^* \leftarrow \arg\min_{M \in \mathcal{C}} \overline{D(M)}$ ▷ top-of-pile selection
$\mathcal{P}^{\text{raw}} \leftarrow \textsc{Backproject}(M^*, D, \mathbf{K})$
$\mathcal{P} \leftarrow \text{SOR}(\mathcal{P}^{\text{raw}},\ k{=}20,\ \alpha{=}2.0)$
$(t_x, t_y, t_z) \leftarrow \textsc{Mean}(\mathcal{P})$ ▷ cloud centroid
$\mathbf{v}_1 \leftarrow \text{PCA}(\mathcal{P})$ ▷ dominant eigenvector
$R_z \leftarrow \textsc{ProjectToXY}(\mathbf{v}_1)$ ▷ normalised to $[0°, 180°)$
$\hat{\mathbf{n}} \leftarrow \text{RANSACPlane}(\mathcal{P})$
$(R_x, R_y) \leftarrow \textsc{TiltFromNormal}(\hat{\mathbf{n}})$
return $(t_x, t_y, t_z, R_x, R_y, R_z)$ ▷ mapped by $T_{c \to r}$ in deployment

Composition and execution.

The three independent estimates yield a complete pose $(t_x, t_y, t_z, R_x, R_y, R_z)$ in the camera frame, which is evaluated per axis in this work. For deployment, the rotation is composed as

$$R_{\text{est}} = R_z(\theta_{\text{PCA}})\, R_y(\beta_{\text{RANSAC}})\, R_x(\alpha_{\text{RANSAC}}), \tag{3}$$

and the calibrated rigid transform of Section 3.2 converts the resulting pose into the robot base frame, producing the gripper target consumed by the manipulator controller. The full pipeline is summarized as Algorithm 1.

Table 3 Pipeline hyperparameters.

Stage	Parameter	Value
Acquisition	RGB / depth resolution	1280×720
Acquisition	Camera height above floor	2.0 m
Acquisition	Bin dimensions	1.0 m × 1.0 m × 1.0 m
Segmentation	Confidence threshold τ_{conf}	0.70
Segmentation	Minimum area $A_{\min}$	15,000 px
Segmentation	Minimum solidity $\sigma_{\min}$	0.70
Point cloud	SOR neighbours k	20
Point cloud	SOR std multiplier α	2.0
Pose	RANSAC distance threshold	1.5 mm
Pose	RANSAC points per sample	3
Pose	RANSAC iterations	1000

4 Experimental Setup

This section presents the experimental setup used to evaluate the system described in Section 3.

4.1 Dataset

A custom dataset of 487 RGB-D frames was collected with the fixed RealSense D456 described in Section 3.2, capturing cluttered piles of elongated aluminium profiles (nominal dimensions 57 × 13 × 1150 mm) with a coloured, glossy finish that retains a metallic appearance, inside the 1 m × 1 m × 1 m container at the configured camera height of 2 m. Instance masks were drawn manually in COCO polygon format by a domain expert; the number of visible instances per frame depends on pile density and inter-part occlusion, and the 50 test images contain 773 annotated instances (15.5 per image on average). The dataset was partitioned into training, validation, and test sets at the source-image level, prior to any augmentation, using an approximately 70/20/10 split that yielded 340 training images, 97 validation images, and 50 test images. Performing the split before augmentation is essential to prevent visually similar augmented variants of the same source image from leaking across partitions, an error that would otherwise inflate validation and test metrics by an unknown factor.

Augmentation was applied exclusively to the training partition, leaving validation and test sets in their resize-only form. Six controlled augmentation pipelines were applied once to each training image, producing seven training variants per source image (one resize-only baseline plus six augmented variants). Each pipeline targets a distinct industrial scenario relevant to bin-picking of rigid metallic parts: *Geometry_Flip* applies horizontal flip, vertical flip, and discrete 90° rotations to enforce SE(2) pose invariance; *Geometry_Rotate* adds continuous rotations in $[-20°, 20°]$ for fine orientation coverage; *Light_Hard* applies brightness and contrast perturbations together with contrast limited adaptive histogram equalization (CLAHE) to simulate harsh overhead lighting and specular reflections on metallic surfaces; *Camera_Noise* introduces Gaussian sensor noise and motion blur to simulate sensor and motion artefacts; *Occlusion* applies coarse rectangular dropout to simulate inter-part overlap; and *Combo_Hard* combines moderate rotation, brightness/contrast variation, and light noise as a realistic mixed-degradation scenario. Transformations that violate the rigid-body assumptions of metallic parts were deliberately excluded: `ElasticTransform` and `GridDistortion` were removed because metal does not deform elastically under the imaging geometry; aggressive perspective warps and shear transforms were removed because they introduce non-rigid artefacts inconsistent with rigid-body imaging; `ToGray` was removed because it destroys the residual surface-defect colour cues that aid segmentation on otherwise homogeneous metal; and `HueSaturationValue` was removed because the resulting colour shifts are unrealistic for metallic surfaces under fixed industrial lighting. All augmentations were implemented using the Albumentations library [4].

In addition to the per-image augmentation pipelines, 70 mosaic images were generated by

tiling four randomly selected training source images into a single 2×2 canvas [2], with polygon labels analytically scaled and translated to their target quadrant to avoid mask rasterization error. Mosaics increase per-image instance density and are particularly relevant to dense bin-picking scenes where a typical frame contains many parts in close proximity. All output images, augmented and mosaic alike, were standardized to 1280×720 pixels. The final dataset composition is summarized in Table 4.

In addition to the segmentation dataset, an independent pose evaluation set of 100 RGB-D frames was captured with the same camera, container, and parts. Its instance masks were drawn manually in COCO polygon format by the same annotator, and every visible part in each frame was annotated. These frames are not part of the 487-frame segmentation dataset, and they are used exclusively for the pose evaluation described in Section 4.4.

4.2 Ground-Truth Pose Extraction

Robot-verified ground truth, in which a calibrated jig or teach-pendant readout supplies a reference pose for every test instance, is impractical for elongated 1.15 m profiles in a cluttered bin: each pose would require a manual jog of the manipulator to a verified contact point, and the part length combined with inter-part occlusion makes such measurements both slow and prone to operator error. This work therefore adopts an *annotation-consistent* ground-truth methodology, in which the hand-annotated COCO masks of the pose evaluation set are treated as the source of truth for instance localization, and the same processing pipeline used at inference (mask filtering, top-of-pile selection, back-projection, SOR) is applied to the annotation mask rather than to a model prediction. The resulting reference pose $(t_x^*, t_y^*, t_z^*, R_x^*, R_y^*, R_z^*)$ is derived per axis as follows: (t_x^*, t_y^*) from the centroid of the annotation mask back-projected through depth; t_z^* from the filtered mean depth of the annotation-masked point cloud after SOR; R_z^* from `cv2.minAreaRect` applied to the 2D annotation mask, with the long-axis angle extracted and normalised to $[0°, 180°)$; and R_x^*, R_y^* from a RANSAC plane fit [9] on the SOR-cleaned annotation point cloud.

This methodology has known uncertainty bounds that must be acknowledged. The reference t_z^* inherits the depth-sensor noise of the RealSense D456, whose depth error is specified as below 2% up to 4 m [24] and, for stereo triangulation, grows quadratically with range [18]; specular reflections on the metallic profiles are expected to increase it further. The references R_x^* and R_y^* inherit this depth uncertainty through the plane fit, and the reference R_z^* is limited by the discretization of the annotation mask at the part edges; no numerical bound is claimed for these references. All pose errors reported in this work are therefore meaningful only as *relative* comparisons across pose estimation methods, all of which are evaluated against the same reference and operate on the same point cloud geometry. Absolute error values of the order of the reference uncertainty should not be interpreted as a method's true accuracy.

4.3 Training Protocol

All five segmentation architectures — Mask R-CNN [10], Mask R-CNN with PointRend [19], YOLOv8m-seg [16], YOLO11m-seg [15], and YOLO26m-seg [17] — were trained on the 2450-image training set and validated on the 97 original validation images, without augmentation. The comparison shares a common set of settings: COCO-pretrained initialization, a batch size of 24, a maximum of 300 epochs, and an early-stopping patience of 30 epochs. To ensure hardware uniformity across the comparison, every model was trained on a single Google Colab instance equipped with an NVIDIA A100 80 GB GPU and high-RAM configuration. Also, checkpoint resumption was used to handle Colab session disconnections without restarting training. However, the optimizer, the learning-rate schedule, and the input resolution follow the conventions of each framework, as summarized in Table 5.

The Mask R-CNN and PointRend models were trained with the Detectron2 framework and its SGD optimizer, using a custom cosine learning-rate schedule that decays from $1{\times}10^{-2}$ to $1{\times}10^{-3}$ after a linear warm-up. Input images were rescaled by the multi-scale shorter-side resizing of Detectron2, with the shorter side set to 800 pixels and the longer side capped at 1333 pixels at inference. Early stopping was monitored on the validation mask $\mathrm{mAP}_{50:95}$. Results show that Mask R-CNN

Table 4 Dataset composition. Augmentation is applied only to the training partition.

Partition	Source images	Final images
Training	340	2450
baseline (resize-only)	—	340
6 augmentation pipelines	—	2040
mosaic	—	70
Validation	97	97
Test	50	50
Total	**487**	**2597**

stopped at epoch 79, with its best validation score at epoch 49, and that PointRend stopped at epoch 121, with its best validation score at epoch 91.

The three YOLO variants were trained with Ultralytics 8.4.14 under identical arguments that differ only in the model definition. The optimizer was selected automatically by the framework, which for the length of this training run resolved to MuSGD [17] with an initial learning rate of 1×10^{-2} and a momentum of 0.9; the learning rate decays linearly to 1×10^{-3}. Images were processed at an input size of 1280 pixels on the long side, and gradient accumulation over three batches of 24 images yields an effective batch size of 72. In addition, Ultralytics applies online mosaic augmentation, disabled during the last 10 epochs, on top of the 70 offline mosaics described in Section 4.1; this constitutes a protocol difference with respect to the Detectron2 models. Also, early stopping was never triggered for the YOLO variants: all three completed the 300 epochs, with their best validation epochs at 300 (YOLOv8m-seg and YOLO11m-seg) and 298 (YOLO26m-seg). In all cases, the checkpoint with the best validation score was retained for the evaluation on the test set.

4.4 Evaluation Metrics

The segmentation stage is evaluated exclusively on the 50 original test images, without augmentation, and the pose stage on the independent 100-frame pose evaluation set described in Section 4.1, so that all reported metrics reflect performance on sensor data the system would encounter during deployment. Augmented copies are not included in any final reporting figure. In the pose evaluation, a frame is excluded when the centroid estimated by PCA lies more than 50 mm from the reference centroid, which indicates that the top-of-pile selection on the predicted masks retained a different part from the one selected on the annotation masks. Because all pose methods operate on the same selected mask, the exclusion applies equally to every method; 8 of the 100 frames were excluded on this basis, leaving 92 instances for the pose comparison.

Segmentation performance is reported using the standard COCO evaluation protocol [22] with mask-based IoU: average precision averaged over ten IoU thresholds from 0.50 to 0.95 in steps of 0.05 ($\text{mAP}_{50:95}$), average precision at IoU thresholds of 0.50 (mAP_{50}) and 0.75 (AP_{75}), and average recall given at most 100 detections per image (AR_{100}). All segmentation metrics are computed via `pycocotools` on the test set predictions.

Pose estimation performance is reported in two complementary forms. The first is the per-axis decomposition that constitutes this work's principal evaluation contribution: the absolute translational errors $|\Delta t_x|$, $|\Delta t_y|$, $|\Delta t_z|$ in millimetres, the Euclidean translation error $|\Delta t_{\text{pos}}|$, and the absolute rotational errors $|\Delta R_x|$, $|\Delta R_y|$, $|\Delta R_z|$ in degrees, with $|\Delta R_z|$ computed using the symmetry-aware metric of Equation (2) to account for the 180° rotational equivalence of elongated profiles. Per-axis errors are reported as mean absolute errors (MAE) over the 92 retained instances. The second form is an aggregate success rate. A pose is counted as correct when $|\Delta t_x|$, $|\Delta t_y|$, and $|\Delta t_z|$ are each below 5 cm and $|\Delta R_x|$, $|\Delta R_y|$, and $|\Delta R_z|$ are each below 5°. This is a per-axis variant of the 5 cm/5° criterion originally introduced for RGB-D camera relocalization [26] and adopted for hybrid bin-picking by [21]; because the thresholds are applied per axis rather than to the Euclidean translation and total rotation errors, it is less strict than the original formulation.

Table 5 Training configuration of the segmentation models.

Setting	Mask R-CNN & PointRend	YOLOv8m / YOLO11m / YOLO26m
Framework	Detectron2	Ultralytics 8.4.14
Optimizer	SGD	MuSGD (automatic selection)
LR schedule	Cosine, linear warm-up	Linear
Initial / final LR	10^{-2} / 10^{-3}	
Batch size	24	24 (effective 72)
Input resolution	Multi-scale; 800 (short side), 1333 max. (long side) at test	1280 (long side)
Online mosaic	No	Yes (off in last 10 epochs)
Max epochs	300	
Early-stopping patience	30	
Epochs run (best)	79 (49) / 121 (91)	300 (300) / 300 (300) / 300 (298)
Initialization	COCO-pretrained	
GPU	A100 80 GB	

To summarize the comparison in a single figure, a composite score is computed from four MAE metrics: the Euclidean translation error $|\Delta t_{\text{pos}}|$ and the rotation errors $|\Delta R_x|$, $|\Delta R_y|$, and $|\Delta R_z|$. Each MAE is divided by the largest value obtained for that metric among the seven methods, so that the worst method scores 1 on every metric and every other method keeps a non-zero score proportional to its error; the composite score is the unweighted mean of the four normalized values, and a lower score indicates a better method. Because this normalization depends on the set of compared methods, the ranking was also computed with min–max normalization and with the two PPF-based methods excluded from the normalization. The order of the methods was unchanged in every case. The runtime of the pose stage is reported as the median wall-clock time per instance on the laptop CPU, computed over all 100 frames of the pose evaluation set, and the runtime of the segmentation stage is reported as the inference time per test image on the laptop GPU.

4.5 Implementation Details

The complete software stack is reproducible from open-source components. The Mask R-CNN and PointRend models were trained and evaluated using Detectron2; the three YOLO variants used the Ultralytics framework. Point-cloud processing employed a custom k-nearest-neighbour SOR implementation built on `scikit-learn`'s `NearestNeighbors`; depth hole-filling used SciPy's distance-transform-based nearest-neighbour inpainting. The 2D mask analysis (centroid, contour, `minAreaRect`, solidity) used OpenCV-contrib. Three-dimensional visualizations were produced using Plotly rather than the Open3D GUI, to accommodate the WSL2 display constraints of the local development machine without affecting any reported metric. The offline augmentation is generated once: each training source image passes through the seven variants exactly once, with no per-epoch resampling, so that every model is trained on the same fixed set of 2450 images.

5 Results and Discussion

All inference, point-cloud processing, pose estimation, and evaluation were performed on the laptop described in Table 2; only model training (Section 4.3) used the Google Colab A100 platform.

5.1 Segmentation Performance

Results indicate that the two-stage architectures (Mask R-CNN and PointRend) outperform the single-stage YOLO family on mask quality at strict IoU thresholds, Table 6. The five architectures were evaluated under the COCO evaluation protocol on the 50 original test images (773 annotated instances), with inference on the laptop GPU (NVIDIA RTX 4060 Laptop, 8 GB). Mask R-CNN achieved the highest $\text{mAP}_{50:95}$ (63.13) and the highest AP_{75} (71.46). PointRend, despite a boundary-refinement design intended to recover

crisp object contours that are oversmoothed by standard mask heads [19], performed comparably to the standard Mask R-CNN (62.28 versus 63.13 $mAP_{50:95}$). Its authors note that IoU-based mask metrics are biased towards object-interior pixels and relatively insensitive to boundary improvements [19], which may partly explain the absence of a measurable gain. Also, the three YOLO variants lie within 0.85 points of one another (45.15–46.00 $mAP_{50:95}$), so the successive architectures from YOLOv8 to YOLO26 produced no measurable difference on this single-class dataset. At an IoU threshold of 0.5, which counts a predicted mask as correct when its intersection over union with the annotation is at least 0.5, all five models exceed 83 AP; the gap between the two families therefore appears at the higher IoU thresholds, where mask-boundary precision is decisive. For reference, Mask R-CNN with the ResNet-50-FPN backbone used in this work reaches a mask AP of 33.6 (AP_{50} of 55.2) on the multi-category COCO benchmark [10]; these values are not directly comparable with the single-class results of this work, but they illustrate that $mAP_{50:95}$, which averages precision over IoU thresholds up to 0.95, is considerably lower than mAP_{50} for the same model. Also, on a bin-picking dataset of thin, low-solidity industrial parts, with training on synthetic images and testing on real scenes, a re-trained Mask R-CNN reached an average mask AP_{50} of 86 and AP_{75} of 56 [8]; although the datasets differ, the values obtained in this work (89.33 and 71.46) are of the same order or higher.

From the size-stratified COCO metrics, it can be said that the gap between the two families is largest for small masks. On instances in the COCO small-area range, Mask R-CNN reaches an $AP_{50:95}$ of 36.35, against 11.32–12.49 for the three YOLO variants; on instances in the large-area range, the corresponding values are 75.18 and 56.25–57.38. In addition, the inference times in Table 6 are the mean wall-clock time per image of the evaluation loop, which includes image loading, the transfer of the predictions to the CPU, and the run-length encoding of every predicted mask for the COCO evaluation. Because the Detectron2 models were evaluated without a score threshold and return 100 masks per image (5000 in total), whereas the YOLO variants returned 912–1052 masks in total, this overhead is larger for the two-stage models. Within this measurement, YOLO11m-seg was the fastest model (106.86 ms per image) and PointRend the slowest (349.55 ms). Mask R-CNN was selected as the segmentation front-end of the pipeline (Section 3.3) because mask quality is the selection criterion of this component: the pose stage consumes a single mask, and the boundary of that mask determines which depth pixels enter the point cloud. Its additional 206 ms per image relative to YOLO11m-seg is the cost of this choice, and PointRend was not retained because it offers no mask-quality gain at a higher inference time. Fig. 2 shows the masks predicted by Mask R-CNN on an image of a cluttered bin, before the filtering cascade of Section 3.3 discards those with confidence below 0.7.



Fig. 2 Instance masks and confidence scores predicted by Mask R-CNN on a cluttered bin of aluminium profiles.

5.2 Per-Axis Pose Estimation: Baseline Methods

Results indicate that each classical primitive yields its lowest errors on a different subset of the pose components, Table 7. The six baselines (PCA, RANSAC, PPF, and their ICP-refined counterparts) were evaluated on the 92 retained instances of the pose evaluation set against the annotation-consistent reference defined in Section 4.2. All methods consume the same SOR-cleaned point cloud, derived from the same Mask R-CNN mask; the ICP-refined methods and the two PPF methods additionally use a 3D model of the part, whereas PCA and RANSAC do not. PPF follows the original formulation of [6]. The ICP-refined methods apply point-to-plane ICP from an initial alignment of the model to the cloud centroid, to the dominant PCA axis of the cloud, and to a reference normal, which is the camera axis for PCA+ICP and the RANSAC plane normal for

Table 6 Segmentation results on the 50 original test images; inference is the mean time per image on the laptop GPU. Best per column in bold.

Model	$\mathbf{mAP}_{50:95}$	$\mathbf{mAP}_{50}$	$\mathbf{AP}_{75}$	$\mathbf{AR}_{100}$	Inference (ms)
Mask R-CNN	**63.13**	89.33	**71.46**	**69.07**	313.25
PointRend	62.28	**89.42**	70.26	68.80	349.55
YOLOv8m-seg	45.15	83.31	46.00	51.47	132.10
YOLO11m-seg	45.33	83.35	44.54	52.26	**106.86**
YOLO26m-seg	46.00	83.24	45.20	52.83	109.82

RANSAC+ICP; for the model-based methods, the reported translation is the position of the model origin, placed at the centre of its top face. The results fall into the four regimes described below.

PCA yields the lowest translation and in-plane rotation errors.

PCA without refinement achieved the lowest $|\Delta R_z|$ among the six baselines (0.42°), more than 5 times lower than RANSAC (2.35°) and more than 17 times lower than PPF (7.35°). The dominant eigenvector of the cloud covariance follows the long axis of the elongated profile, as described in Section 3.5. However, all 92 reference orientations lie within 4.75° of the 0°/180° image axis (mean 2.30°), because the 1150 mm profiles extend beyond the open sides of the container (Section 3.1); the advantage of PCA in R_z is therefore established only within this narrow range. Also, PCA produced the lowest errors on every translation component ($|\Delta t_{\text{pos}}|$ of 6.25 mm), with the translation taken as the centroid of the cleaned point cloud.

RANSAC yields the lowest out-of-plane rotation errors.

RANSAC without refinement achieved the lowest $|\Delta R_x|$ (1.19°) and $|\Delta R_y|$ (0.08°) among the six baselines, approximately 2.5 and 2.0 times lower than the next best method, PCA+ICP (2.99° and 0.15°). The plane fit recovers the tilt of the near-planar top face of the part. However, RANSAC does not estimate R_z: the in-plane axes of its frame are constructed from a fixed auxiliary vector (Section 3.5), and its R_z remained within 0.60° of the 0°/180° image axis in all 92 instances. Its $|\Delta R_z|$ of 2.35° is therefore close to the mean deviation of the reference orientations from that axis (2.30°), and it reflects the narrow orientation range of the evaluation set rather than an estimate of R_z. In addition, its translation, taken as the centroid of the plane inliers, has the largest error among the methods that do not use PPF (29.26 mm).

PPF fails on the geometry of the part.

PPF and PPF+ICP exhibited markedly larger errors than the other four baselines: vanilla PPF exceeded 110 mm in $|\Delta t_{\text{pos}}|$ and 18° in $|\Delta R_x|$, and PPF+ICP diverged on $|\Delta t_z|$ (92.72 mm MAE versus 0.25 mm for PCA), which is consistent with ICP being initialized from a PPF coarse pose far from the correct one, since the result of ICP depends on its initial pose [1]. This failure mode is consistent with a documented property of point pair feature matching: geometry-only PPF voting often returns false positives when clutter is shaped like the object, which is common for objects with large planar sides that are then found in background planes [20].

ICP refinement is conditionally beneficial.

Compared with PCA, PCA+ICP lowers $|\Delta R_x|$ (5.29° → 2.99°) but raises $|\Delta t_{\text{pos}}|$ (6.25 mm → 13.97 mm). Compared with RANSAC, RANSAC+ICP lowers $|\Delta t_{\text{pos}}|$ (29.26 mm → 13.38 mm) and $|\Delta R_z|$ (2.35° → 0.51°), the latter because its initial alignment takes the long axis from PCA, but raises $|\Delta R_x|$ (1.19° → 3.10°). Also, PPF+ICP raises $|\Delta t_{\text{pos}}|$ with respect to PPF (110.63 mm → 134.78 mm, driven by t_z). Results indicate that ICP refinement is not a free upgrade: its effect depends on both the coarse stage and the pose component.

The composite score of Section 4.4 placed PCA+ICP first among the six baselines (0.095), followed closely by RANSAC+ICP (0.096) and PCA (0.114). However, no single baseline achieved both the lowest translation and R_z errors (PCA)

and the lowest R_x and R_y errors (RANSAC). This per-axis complementarity is the basis of the component assignment of the hybrid estimator, Section 3.5.

5.3 Hybrid Method Validation

Results indicate that the proposed hybrid matches or improves on the best baseline on every pose component, Table 7. The hybrid combines the translation and R_z of PCA with the R_x and R_y of RANSAC, as specified in Section 3.5; by construction, its per-axis errors equal those of PCA on the translation and on R_z and those of RANSAC on R_x and R_y, which are the lowest values among the six baselines on each of these components. As a result, the hybrid achieves a composite score of 0.047, ranking first among the seven methods and improving by approximately 50% on the runner-up, PCA+ICP (0.095). The result is visualized in Fig. 3 (per-axis comparison) and Fig. 4 (composite ranking).

The mechanism behind this result is a selection over existing estimators rather than a new estimator: each primitive is used only on the components on which it yields the lowest error in the baseline comparison. The R_x of PCA (5.29°) is replaced by that of RANSAC (1.19°), the R_z of RANSAC (2.35°) by that of PCA (0.42°), and the translation of PCA (6.25 mm) is kept unchanged, at the cost of running both estimators on the same point cloud (Section 5.4). The authors therefore present the hybrid not as a new pose estimator in the algorithmic sense, but as a per-axis selection rule; the per-axis decomposition of the error, and the component assignment it supports for elongated near-planar parts, constitute the methodological contribution of this work. Because the assignment was chosen from the same evaluation set on which it is reported, it is also supported by the geometry of each primitive (Section 3.5), and this point is discussed as a limitation in Section 5.5. This decomposition can serve as a practical guide for selecting, per pose component, the classical primitive best suited to a given part geometry.

In addition, the hybrid places all 92 retained instances within the per-axis 5 cm/5° criterion (100%), against 81.5% for PCA+ICP and RANSAC+ICP, 80.4% for RANSAC, 72.8% for PCA, 34.8% for PPF+ICP, and 2.2% for PPF; replacing the per-axis translation thresholds with the Euclidean translation error changes only the PPF+ICP rate (33.7%). Also, because this criterion saturates for the hybrid, a stricter 2 cm/2° threshold is reported: the hybrid retains 85.9% of the instances, against 52.2% for PCA, 34.8% for RANSAC+ICP, 29.3% for PCA+ICP, 10.9% for RANSAC and PPF+ICP, and 0% for PPF. These rates measure pose accuracy against the reference of Section 4.2; they are not grasp success rates, since no robot picks were executed in this work.

To assess whether the difference with the runner-up is systematic, a per-instance composite error was computed by applying the normalization of the composite score to the four absolute errors of each instance ($|\Delta t_{\text{pos}}|$, $|\Delta R_x|$, $|\Delta R_y|$, and $|\Delta R_z|$) and averaging them. A paired, two-sided Wilcoxon signed-rank test [30] on these per-instance composite errors between the hybrid and PCA+ICP confirms that the difference is statistically significant ($p < 0.001$, $n = 92$); the hybrid yields the lower per-instance composite error in 82 of the 92 instances. Also, the hybrid bounds the worst case: its maximum errors are 4.24° on $|\Delta R_x|$ and 1.08° on $|\Delta R_z|$, against 8.96° and 2.08° for PCA+ICP. The tail behavior of the hybrid is therefore controlled, on each axis, by the component method with the lower error, a property that the composite score does not directly reward. As noted in Section 5.2, the R_z results hold within the narrow orientation range of the evaluation set.

Comparison with similar hybrid frameworks.

Table 8 summarizes the key architectural and evaluation differences between the proposed system and its three closest precursors: [20], [21], and [32]. Two distinctions are decisive. First, all three prior works require either multiple sensors, a 3D object model, or both, while the present system operates from a single fixed RGB-D camera without any object model. Second, none of the prior works compares pose estimators per pose component or assigns different estimators to different degrees of freedom; [21] reports the distribution of the per-axis errors of its own method, but not a per-axis comparison of estimators. The per-axis comparison of estimators introduced here, and the component assignment it supports, are, to the best of the authors' knowledge, the first reported for a

Table 7 Per-axis MAE of the six baselines and the proposed hybrid over the 92 retained instances of the pose evaluation set. Best per column in bold (ties included); second-best underlined.

Method	$\lvert\Delta t_x\rvert$ (mm)	$\lvert\Delta t_y\rvert$ (mm)	$\lvert\Delta t_z\rvert$ (mm)	$\lvert\Delta t_{\text{pos}}\rvert$ (mm)	$\lvert\Delta R_x\rvert$ (°)	$\lvert\Delta R_y\rvert$ (°)	$\lvert\Delta R_z\rvert$ (°)	Composite	Rank
PCA	**5.92**	**1.05**	**0.25**	**6.25**	5.29	0.24	**0.42**	0.114	4
PCA+ICP	11.38	5.35	2.53	13.97	2.99	0.15	0.54	0.095	2
RANSAC	28.89	2.38	0.89	29.26	**1.19**	**0.08**	2.35	0.156	5
RANSAC+ICP	10.72	5.54	2.56	13.38	3.10	0.17	0.51	0.096	3
PPF	78.99	52.74	20.92	110.63	18.75	3.35	7.35	0.955	7
PPF+ICP	61.13	29.46	92.72	134.78	5.37	0.47	2.42	0.439	6
Hybrid (proposed)	**5.92**	**1.05**	**0.25**	**6.25**	**1.19**	**0.08**	**0.42**	**0.047**	**1**

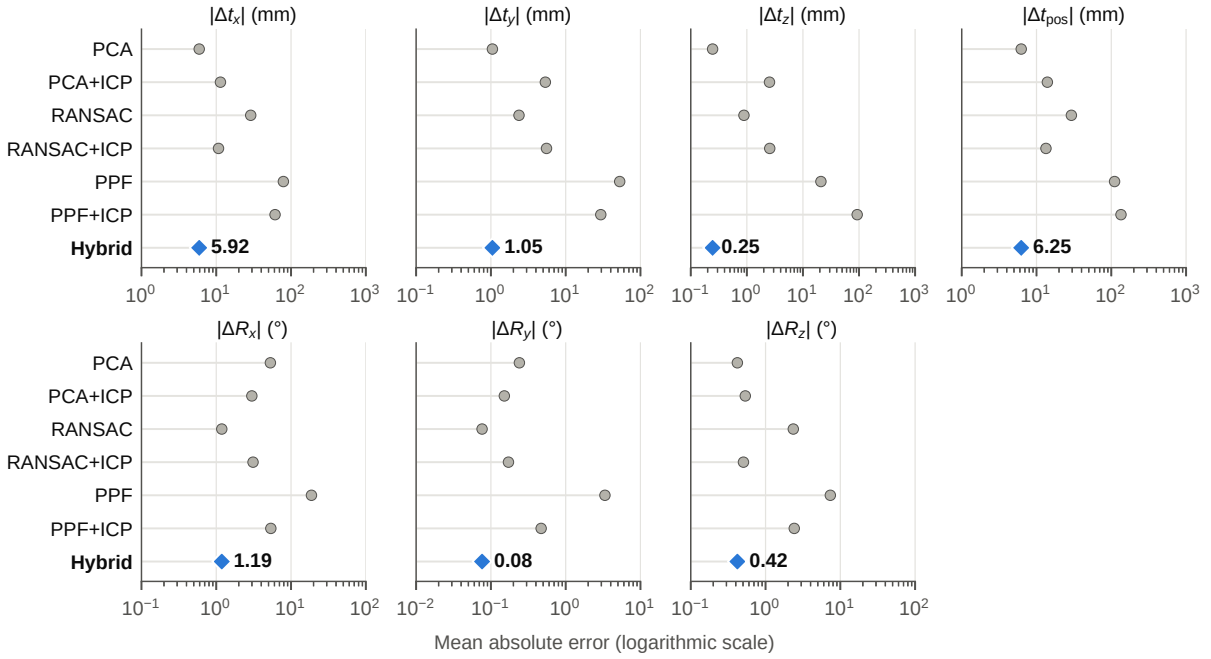


Fig. 3 Per-axis MAE of the seven pose estimation methods over the 92 retained instances of the pose evaluation set (logarithmic scale). Translation in mm, rotation in degrees.

hybrid bin-picking framework. However, the headline results of Table 8 are obtained on different datasets and with different metrics, so they are not directly comparable. Also, [32] reports grasp success in physical experiments, whereas the evaluation of the present work ends at the estimated pose of the selected part.

5.4 Computational Cost and Pipeline Timing

Results indicate that the pose stage of the hybrid adds a small fraction of the segmentation time, Table 9. The median wall-clock time per instance of each pose method was measured on the laptop CPU (Intel Core i7, 12th generation; no GPU acceleration) over the 100 frames of the pose evaluation set. The hybrid runs in 17.12 ms per instance, a time that includes PCA, the RANSAC plane fit, and the extraction of the angles; pure RANSAC runs in 16.09 ms and PCA in 1.72 ms. Also, the hybrid is approximately 15 times faster than the fastest ICP-refined method (PCA+ICP, 257.25 ms). PPF and PPF+ICP, which exceed 23 s per instance on this dataset, are unsuitable for real-time operation regardless of accuracy — a finding consistent with prior hybrid pipelines that

Table 8 Comparison with the three closest hybrid bin-picking precursors.

Property	[20]	[21]	[32]	**This work**
Sensors	Single RGB-D	Two cameras (global + eye-in-hand)	Single 3D scanner (PhoXi)	Single RGB-D
Robot approach motion	No	Yes (one-stage move)	No	No
Segmentation front-end	Mask R-CNN / RetinaMask	Mask R-CNN	Point-cloud instance seg.	Mask R-CNN (5 compared)
Pose back-end	PPF + ICP	PPF (Vidal variant)	Convolutional network on depth/normal maps + ICP	Axis-decoupled PCA + RANSAC
3D object model required	Yes	Yes (scanned)	Yes (scanned)	No
Per-axis error evaluation	No	Own method only	No	Yes (7 methods compared)
Headline result	BOP 2020: 2nd overall, best <1 s	97.4% 5 cm/5°	94.2–97.1% ADI (synthetic); 100% grasp success (real)	Composite 0.047, rank 1 of 7

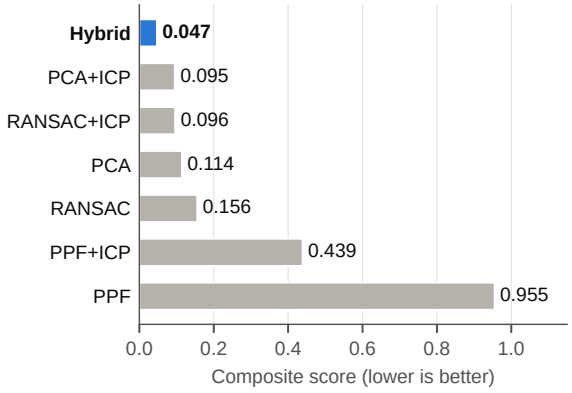


Fig. 4 Composite score of the seven pose estimation methods (lower is better).

restrict PPF voting to segmented instances, which made PPF-based pose estimation over 12 times faster than voting on the complete scene [20].

The computation from an RGB-D frame to a pose comprises the segmentation stage (laptop GPU), the intermediate stages (mask filtering, top-of-pile selection, depth hole-filling, back-projection, and SOR; laptop CPU), and the pose stage (laptop CPU). The intermediate stages were not instrumented during the experiments, so Table 10 reports only the two measured stages, for Mask R-CNN, the selected front-end, and for YOLO11m-seg, the fastest alternative. Results indicate that segmentation and hybrid pose estimation together take approximately 330 ms per frame with Mask R-CNN and 124 ms with YOLO11m-seg, and that segmentation accounts for most of this time. These sums combine a mean segmentation time, which includes the evaluation overhead described in Section 5.1, with a median pose time; they indicate the order of magnitude of the per-frame computation rather than a full cycle time, which would also include the intermediate stages and the robot

Table 9 Median runtime of the pose stage per instance on the laptop CPU (100 frames of the pose evaluation set).

Method	**Median runtime (ms)**
PCA	1.72
RANSAC	16.09
Hybrid (proposed)	**17.12**
PCA+ICP	257.25
RANSAC+ICP	282.77
PPF	23,209.86
PPF+ICP	23,254.52

motion, both outside the measurements of this work.

5.5 Limitations and Failure Modes

Six limitations of the present work merit explicit discussion. First, the reference poses are annotation-consistent rather than robot-verified (Section 4.2), and absolute pose errors of the order of the reference uncertainty described there should be interpreted with this in mind; the per-method comparisons remain valid because all methods are evaluated against the same reference, which computes (t_x^*, t_y^*) from the annotation-mask centroid and R_z^* from the minimum-area rectangle, differently from the hybrid. Second, all 92 reference orientations lie within 4.75° of the 0°/180° image axis, because the profiles extend beyond the open sides of the container; the R_z results of PCA and of the hybrid are therefore established only within this range, and a set with a wider range of in-plane orientations is required to extend them. Third, the component assignment of the hybrid was chosen from the per-axis comparison on the same evaluation set on which it is reported; although it is consistent with the geometry of each primitive (Section 3.5), its confirmation on an independent set remains pending. Fourth, the RANSAC plane fit is stochastic: two runs on the same point clouds with different random sampling yielded R_x estimates whose per-instance difference has a standard deviation of 1.36°, of the same order as its MAE. The hybrid reported here takes R_x and R_y from the same RANSAC run as the RANSAC baseline, and its rates at the stricter thresholds, which depend on R_x, would vary between runs.

Fifth, the dataset comprises a single object class, the elongated metallic profile of dimensions $57 \times 13 \times 1150$ mm. The axis-decoupling principle is expected to apply to parts whose in-plane orientation is characterized by a principal direction and whose tilt is characterized by a planar face, such as brackets, plates, and machined panels, but this has not been evaluated; it does not apply to free-form rigid objects with distinctive surface geometry, for which PPF [28] or learning-based estimators remain the appropriate choice. Sixth, no robot picks were executed: the evaluation ends at the estimated pose of the selected part in the camera frame, the camera-to-robot transform is not part of the evaluation, and no grasp success rate is reported. Also, each frame was captured from a static pile, so dynamic scenes with shifting piles between captures are not addressed.

Two failure modes were observed in the pose evaluation set and are reported here as guidance for deployment. The first is a selection mismatch: in 8 of the 100 frames, the top-of-pile selection on the predicted masks retained a different part from the one selected on the annotation masks, with 55 to 633 mm between the two centroids. These frames yield the pose of another part rather than an erroneous pose of the intended one, and they are excluded from the comparison (Section 4.4); the pipeline produced a pose in all 100 frames. The second concerns R_z: the error of the hybrid grows with the tilt of the part, $\sqrt{R_x^{*2} + R_y^{*2}}$, which ranges from 0.04° to 16.35° in the evaluation set. Across the four tilt quartiles, the mean $|\Delta R_z|$ increases from 0.12° to 0.35°, 0.58°, and 0.63° (Spearman's $\rho = 0.70$), and it also increases as the part lies closer to the camera ($\rho = -0.78$ with the mean depth of the part). Results indicate that the errors of the hybrid remain bounded: over the 92 retained instances, its maximum errors are 36.58 mm in $|\Delta t_{\text{pos}}|$, 4.24° in $|\Delta R_x|$, 0.41° in $|\Delta R_y|$, and 1.08° in $|\Delta R_z|$, and its t_z error has a mean of -0.06 mm and a maximum absolute value of 1.94 mm.

6 Conclusions

This work has presented a cost-effective hybrid framework for 6D pose estimation of weak-texture industrial parts in cluttered bins. A deep instance segmentation front-end is combined with an axis-decoupled classical pose estimator: the translation is taken as the centroid of the cleaned point cloud, the in-plane rotation R_z is recovered by PCA, and the out-of-plane rotations R_x and R_y by a RANSAC plane fit. The pipeline uses a single fixed RGB-D camera, without a second sensor, a robot approach motion, or a 3D object model, on hardware with an approximate cost of USD 1,600, excluding the manipulator (Table 2).

Results indicate that no single geometric primitive yields the lowest error on every pose component: PCA achieved the lowest translation and R_z errors (6.25 mm and 0.42°), and RANSAC the lowest R_x and R_y errors (1.19° and 0.08°); to the

Table 10 Measured computation time per frame on the laptop. The intermediate stages (mask filtering to SOR) were not instrumented and are not included.

Stage	Mask R-CNN	YOLO11m-seg
Segmentation (mean, GPU)	313.25 ms	106.86 ms
Hybrid pose (median, CPU)	17.12 ms	17.12 ms
Segmentation + pose	≈330 ms	≈124 ms

best of the authors' knowledge, this is the first per-axis comparison of pose estimators reported for a hybrid bin-picking framework. As a result, the hybrid matched or improved on the best baseline on every component, achieved a composite score of 0.047, approximately 50% lower than PCA+ICP (0.095), and placed 100% and 85.9% of the 92 retained instances within the per-axis 5 cm/5° and 2 cm/2° criteria; these are pose accuracy rates, not grasp success rates. Its pose stage runs in 17.12 ms per instance on the laptop CPU, approximately 15 times faster than PCA+ICP. Also, Mask R-CNN achieved the highest mask quality among the five segmentation architectures ($mAP_{50:95}$ of 63.13).

These results hold for a single part class, within the narrow range of in-plane orientations evaluated, and without robot picks (Section 5.5). Future work will extend the evaluation to other part classes and to a wider range of orientations, and will integrate the pipeline with a suction-cup end-effector on the Fairino FR10 to report the grasp success rate and the full cycle time. The single-sensor pipeline can serve as a practical starting point for vision-based bin-picking of weak-texture elongated parts.

Statements and Declarations

Funding. The authors declare that no funds, grants, or other support were received during the preparation of this manuscript.

Competing interests. The authors have no competing interests to declare that are relevant to the content of this article.

Ethics approval and consent. Not applicable.

Data and code availability. The datasets and code generated and analysed during the current study are not publicly available but are available from the first author, Daniel Charles Garza (a00836561@tec.mx), on reasonable request.

Author contributions. Daniel Charles Garza: conceptualization, methodology, software, investigation, data curation, formal analysis, visualization, writing – original draft. Rubén Morales-Menéndez: conceptualization, supervision, project administration, writing – review and editing. Antonio Vallejo Guevara: writing – review and editing. Federico Guedea Elizalde: writing – review and editing. Ana Gabriela Sepúlveda Garza: methodology, software. All authors read and approved the final manuscript.

ORCID. Daniel Charles Garza: 0009-0008-6561-7876; Rubén Morales-Menéndez: 0000-0003-0498-1566; Antonio Vallejo Guevara: 0000-0003-3738-4661; Federico Guedea Elizalde: 0000-0001-7959-0573; Ana Gabriela Sepúlveda Garza: 0009-0009-8281-4246.

References

[1] Besl PJ, McKay ND (1992) A method for registration of 3-D shapes. IEEE Transactions on Pattern Analysis and Machine Intelligence 14(2):239–256. https://doi.org/10.1109/34.121791

[2] Bochkovskiy A, Wang CY, Liao HYM (2020) YOLOv4: Optimal speed and accuracy of object detection. https://doi.org/10.48550/arXiv.2004.10934, arXiv:2004.10934

[3] Bogue R (2023) Bin picking: A review of recent developments. Industrial Robot 50(6):873–877. https://doi.org/10.1108/IR-07-2023-0167

[4] Buslaev A, Iglovikov VI, Khvedchenya E, et al (2020) Albumentations: Fast and flexible image augmentations. Information 11(2):125. https://doi.org/10.3390/info11020125

[5] Cordeiro A, Rocha LF, Boaventura-Cunha J, et al (2026) A review of visual perception for robotic bin-picking. Robotics and Autonomous Systems 196:105236. https://doi.org/10.1016/j.robot.2025.105236

[6] Drost B, Ulrich M, Navab N, et al (2010) Model globally, match locally: Efficient and robust 3D object recognition. In: Proc of the IEEE Conf on Computer Vision and Pattern Recognition (CVPR), pp 998–1005

[7] Du G, Wang K, Lian S, et al (2021) Vision-based robotic grasping from object localization, object pose estimation to grasp estimation for parallel grippers: A review. Artificial Intelligence Review 54(3):1677–1734. https://doi.org/10.1007/s10462-020-09888-5

[8] Feng Y, Yang B, Li X, et al (2022) Towards robust part-aware instance segmentation for industrial bin picking. In: Proc of the IEEE Int Conf on Robotics and Automation (ICRA), Philadelphia, PA, USA, pp 405–411, https://doi.org/10.1109/ICRA46639.2022.9811728

[9] Fischler MA, Bolles RC (1981) Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM 24(6):381–395. https://doi.org/10.1145/358669.358692

[10] He K, Gkioxari G, Dollár P, et al (2017) Mask R-CNN. In: Proc of the IEEE Int Conf on Computer Vision (ICCV), pp 2961–2969, https://doi.org/10.1109/ICCV.2017.322

[11] He Y, Sun W, Huang H, et al (2020) PVN3D: A deep point-wise 3D keypoints voting network for 6DoF pose estimation. In: Proc of the IEEE/CVF Conf on Computer Vision and Pattern Recognition (CVPR), pp 11632–11641, https://doi.org/10.1109/CVPR42600.2020.01165

[12] Hinterstoisser S, Lepetit V, Ilic S, et al (2012) Model based training, detection and pose estimation of texture-less 3D objects in heavily cluttered scenes. In: Computer Vision – ACCV 2012, Lecture Notes in Computer Science, vol 7724. Springer, pp 548–562, https://doi.org/10.1007/978-3-642-37331-2_42

[13] Hodaň T, Michel F, Brachmann E, et al (2018) BOP: Benchmark for 6D object pose estimation. In: Proc of the European Conf on Computer Vision (ECCV). Springer, Lecture Notes in Computer Science, pp 19–34, https://doi.org/10.1007/978-3-030-01249-6_2

[14] Hodaň T, Sundermeyer M, Drost B, et al (2020) BOP challenge 2020 on 6D object localization. In: Computer Vision – ECCV 2020 Workshops, Lecture Notes in Computer Science, vol 12536. Springer, pp 577–594, https://doi.org/10.1007/978-3-030-66096-3_39

[15] Jocher G, Qiu J (2024) Ultralytics YOLO11. URL https://github.com/ultralytics/ultralytics, version 11.0.0, AGPL-3.0

[16] Jocher G, Chaurasia A, Qiu J (2023) Ultralytics YOLOv8. URL https://github.com/ultralytics/ultralytics, version 8.0.0, AGPL-3.0. Accessed April 2026

[17] Jocher G, Qiu J, Liu M, et al (2026) Ultralytics YOLO26: Unified real-time end-to-end vision models. https://doi.org/10.48550/arXiv.2606.03748, arXiv:2606.03748

[18] Keselman L, Iselin Woodfill J, Grunnet-Jepsen A, et al (2017) Intel(R) RealSense(TM) stereoscopic depth cameras. In: Proc of the IEEE Conf on Computer Vision and Pattern Recognition Workshops (CVPRW), pp 1267–1276, https://doi.org/10.1109/CVPRW.2017.167

[19] Kirillov A, Wu Y, He K, et al (2020) PointRend: Image segmentation as rendering. In: Proc of the IEEE/CVF Conf on Computer Vision and Pattern Recognition (CVPR), pp 9799–9808

[20] König R, Drost B (2020) A hybrid approach for 6DoF pose estimation. In: Computer Vision – ECCV 2020 Workshops, Lecture Notes in Computer Science, vol 12536. Springer, pp 700–706, https://doi.org/10.

1007/978-3-030-66096-3_46

[21] Le TT, Le TS, Chen YR, et al (2021) 6D pose estimation with combined deep learning and 3D vision techniques for a fast and accurate object grasping. Robotics and Autonomous Systems 141:103775. https://doi.org/10.1016/j.robot.2021.103775

[22] Lin TY, Maire M, Belongie S, et al (2014) Microsoft COCO: Common objects in context. In: Computer Vision – ECCV 2014, Lecture Notes in Computer Science, vol 8693. Springer, pp 740–755, https://doi.org/10.1007/978-3-319-10602-1_48

[23] Peng S, Liu Y, Huang Q, et al (2019) PVNet: Pixel-wise voting network for 6DoF pose estimation. In: Proc of the IEEE/CVF Conf on Computer Vision and Pattern Recognition (CVPR), pp 4561–4570, https://doi.org/10.1109/CVPR.2019.00469

[24] RealSense (2026) Depth camera D456 – product specifications. https://www.realsenseai.com/products/depth-camera-d456/, accessed: 2026-10-14

[25] Rusu RB, Cousins S (2011) 3D is here: Point Cloud Library (PCL). In: Proc of the IEEE IntConf on Robotics and Automation (ICRA), Shanghai, China, pp 1–4, https://doi.org/10.1109/ICRA.2011.5980567

[26] Shotton J, Glocker B, Zach C, et al (2013) Scene coordinate regression forests for camera relocalization in RGB-D images. In: Proc of the IEEE Conf on Computer Vision and Pattern Recognition (CVPR), pp 2930–2937, https://doi.org/10.1109/CVPR.2013.377

[27] Sundermeyer M, Marton ZC, Durner M, et al (2020) Augmented autoencoders: Implicit 3D orientation learning for 6D object detection. Int J of Computer Vision 128:714–729. https://doi.org/10.1007/s11263-019-01243-8

[28] Vidal J, Lin CY, Lladó X, et al (2018) A method for 6D pose estimation of free-form rigid objects using point pair features on range data. Sensors 18(8):2678. https://doi.org/10.3390/s18082678

[29] Wang C, Xu D, Zhu Y, et al (2019) DenseFusion: 6D object pose estimation by iterative dense fusion. In: Proc of the IEEE/CVF Conf on Computer Vision and Pattern Recognition (CVPR), pp 3343–3352, https://doi.org/10.1109/CVPR.2019.00346

[30] Wilcoxon F (1945) Individual comparisons by ranking methods. Biometrics Bulletin 1(6):80–83. https://doi.org/10.2307/3001968

[31] Xiang Y, Schmidt T, Narayanan V, et al (2018) PoseCNN: A convolutional neural network for 6D object pose estimation in cluttered scenes. In: Proc of Robotics: Science and Systems (RSS), https://doi.org/10.15607/RSS.2018.XIV.019

[32] Zhuang C, Li S, Ding H (2023) Instance segmentation based 6D pose estimation of industrial objects using point clouds for robotic bin-picking. Robotics and Computer-Integrated Manufacturing 82:102541. https://doi.org/10.1016/j.rcim.2023.102541