Published by :
https://www.ijert.org/
An International Peer-Reviewed Journal

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 10 , October - 2026

Vision-Guided Pick-and-Place with a UR5 Manipulator Using RGB-D Point Cloud Selection in RViz: A ROS 2 Visual Servoing Framework

C A Tanvir
Dept. of Robotics and Automation Engineering
PSG College of Technology
Coimbatore, India

Sriram V
Dept. of Robotics and Automation Engineering
PSG College of Technology
Coimbatore, India

Abstract - This paper presents a ROS 2-based framework that enables an operator to select an object of interest directly in RViz and have a UR5 six-degree-of-freedom manipulator autonomously pick it using depth-camera point cloud data. An Intel RealSense RGB-D sensor streams a colored point cloud into RViz, where a single mouse click on the object of interest publishes a 3D point that seeds a local region-growing segmentation. The segmented cluster yields a grasp centroid and an approach axis via principal component analysis, which is transformed from the camera frame into the robot base frame using an eye-to-hand calibration. A MoveIt2-based planner drives the arm to a pre-grasp waypoint, after which a closed-loop position-based visual servoing (PBVS) stage continuously corrects the end-effector pose from live point cloud feedback until the pose error falls within a defined tolerance, at which point the gripper closes. The architecture, node/topic design, calibration procedure, and an evaluation protocol (localization accuracy, servoing convergence, and pick success rate) are described in detail, and illustrative simulated analysis plots are provided to demonstrate how the collected data will be interpreted once physical trials are completed. The proposed pipeline reduces manual teaching effort compared to fixed pre-programmed pick poses while retaining the robustness benefits of closed-loop visual feedback over open-loop, look-then-move grasping.

Index Terms - Visual servoing, RGB-D perception, ROS 2, RViz, point cloud segmentation, UR5 manipulator, MoveIt2, hand-eye calibration, robotic grasping.

I. INTRODUCTION

Robotic pick-and-place remains one of the most common tasks demanded of manipulators in warehousing, assembly, and laboratory automation, yet most deployed systems still rely on fixed, pre-taught grasp poses or rigidly fixtured parts. Such open-loop approaches are brittle: any change in object position, shape, or orientation requires re-teaching or a new perception pipeline. Depth cameras have made it practical to perceive a scene as a 3D point cloud in real time, and visualization tools such as RViz allow that point cloud to be inspected directly by a human operator. This creates an opportunity for a human-in-the-loop workflow in which the operator simply clicks on the object to be picked, and the robot autonomously computes the grasp and executes the motion — removing the need for object-specific CAD models, trained detectors, or manual pose teaching for every new object.

This work targets that workflow for a 6-DOF UR5 collaborative arm equipped with a wrist- or workspace-mounted RGB-D camera, integrated entirely within the ROS 2 ecosystem. The central research question addressed is: given only a single clicked point on a live point cloud in RViz, how can a manipulator reliably localize, approach, and grasp the corresponding physical object while remaining robust to calibration error and depth noise? The proposed answer combines (i) point cloud segmentation seeded by the operator's click, (ii) coarse motion planning with MoveIt2 to a pre-grasp pose, and (iii) fine, closed-loop position-based visual servoing (PBVS) for the terminal approach, so that residual calibration and localization error is corrected online rather than relying on a single open-loop estimate.

Contributions

The specific contributions of this paper are:

- A complete ROS 2 node architecture that couples a single-click RViz interaction with autonomous point cloud segmentation, grasp pose estimation, coarse motion planning, and closed-loop visual servoing.
- A region-growing, seed-based segmentation strategy that resolves the ambiguity of a single clicked point by growing a full object cluster and estimating a grasp centroid and approach axis via PCA.
- A PBVS correction law, expressed in closed form, that continuously re-estimates the target pose from live point cloud feedback so that hand-eye calibration error and depth noise are corrected online rather than propagated open-loop.
- A concrete, reproducible evaluation protocol — localization accuracy, servoing convergence, pick success rate, and operator interaction time — together with illustrative simulated analysis figures that define exactly how the resulting data will be visualized and interpreted.

The remainder of this paper is organized as follows. Section II reviews visual servoing paradigms and RGB-D grasp-point selection literature. Section III describes the hardware and software architecture. Section IV details the perception-to-action methodology. Section V describes the ROS 2 node

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 10 , October - 2026

implementation. Section VI outlines the evaluation protocol, and Section VII discusses expected results and the metrics used to validate the system. Section VIII concludes and outlines future work.

II. BACKGROUND AND RELATED WORK

A. Visual Servoing Paradigms

Visual servoing techniques are broadly classified as image-based (IBVS), position-based (PBVS), or hybrid. IBVS regulates an error defined directly in image feature space, offering robustness to camera calibration error but with less predictable Cartesian trajectories. PBVS reconstructs the 3D pose of the target relative to the camera or robot base and regulates the resulting Cartesian pose error, giving intuitive, predictable end-effector motion at the cost of greater sensitivity to calibration accuracy. Because an RGB-D sensor directly yields metric 3D points rather than only 2D image features, PBVS is a natural fit for depth-camera-based grasping and is adopted for the terminal approach phase in this work, while the coarse motion is handled by conventional joint-space motion planning.

B. RGB-D Perception and Grasp-Point Selection

Point cloud-based grasp planning commonly proceeds by segmenting the scene into candidate object clusters, either through unsupervised clustering (e.g., Euclidean or region-growing segmentation), plane removal to isolate objects from a supporting surface, or learned grasp-quality networks operating on raw point clouds. Fully autonomous segmentation pipelines can misidentify the object of interest in cluttered scenes. Interactive selection — where a human operator identifies the object once, in a live visualization — sidesteps this ambiguity and is well suited to semi-autonomous or supervised pick tasks, at the cost of requiring an operator in the loop for object selection.

C. ROS 2 and MoveIt2

ROS 2 provides a real-time-capable, DDS-based middleware suitable for closed-loop control, replacing the single-master architecture of ROS 1 with a distributed publish/subscribe model. TF2 maintains the time-indexed tree of coordinate frame transforms needed to relate the camera, end-effector, and base frames. MoveIt2 builds on this to provide collision-aware joint- and Cartesian-space motion planning, while MoveIt Servo provides a real-time, velocity-based control interface suitable for implementing the visual servoing correction loop used in this work.

D. Comparative Summary of Grasp-Pipeline Strategies

Table I situates the proposed pipeline against two common alternatives: purely open-loop, look-then-move systems, and fully autonomous (no human-in-the-loop) segmentation pipelines. The comparison highlights that the proposed approach is the only one of the three that combines human-supervised object selection with online closed-loop error correction, which is the specific gap this paper addresses.

Strategy	Object selection	Terminal correction
Open-loop, look-then-move	Fixed / pre-taught pose	None
Fully autonomous segmentation	Automatic clustering (ambiguous)	Varies; often open-loop
Proposed (this work)	Single operator click in RViz	Closed-loop PBVS

TABLE I. COMPARATIVE SUMMARY OF GRASP-PIPELINE STRATEGIES

III. SYSTEM OVERVIEW

A. Hardware Configuration

The physical setup consists of a Universal Robots UR5 six-degree-of-freedom collaborative arm, a two-finger parallel gripper mounted at the tool flange, and an Intel RealSense D435 RGB-D camera. The camera may be mounted in an eye-in-hand configuration (rigidly attached near the wrist) or an eye-to-hand configuration (fixed in the workspace, observing both the robot and the objects). The eye-to-hand configuration is assumed in the description below, as it keeps the object continuously visible during the approach without self-occlusion by the gripper.

B. Software Stack

The software stack runs on Ubuntu 22.04 with ROS 2 Humble Hawksbill. Camera driving and point cloud publishing are handled by the realsense2_camera driver, point cloud filtering and segmentation use the Point Cloud Library (PCL) through ROS 2 wrapper nodes, motion planning uses MoveIt2 with the UR5 MoveIt configuration package, and the terminal correction stage uses MoveIt Servo. RViz2 serves as both the visualization front end and the operator's object-selection interface via the built-in "Publish Point" tool. Fig. 1 summarizes how these hardware and software components are connected.

C. Frame Conventions

The TF2 tree maintains, at minimum, the following frames: world (fixed reference), base_link (robot base), tool0 / end-effector frame, and camera_color_optical_frame (or camera_depth_optical_frame). The static transform between the camera frame and the world/base frame is established once through hand-eye calibration (Section IV-D) and published as a static TF broadcast; all subsequent point cloud data is expressed directly in the robot base frame by downstream nodes, simplifying grasp pose computation. Fig. 3 in Section V illustrates this frame tree in full.

D. Design Considerations and Safety

Because the arm operates in a shared or supervised workspace, the coarse motion stage is planned inside a collision-aware planning scene that treats the filtered point cloud — including any obstacles other than the selected object — as occupied space, so that MoveIt2 refuses or replans around trajectories that would collide with the environment. The UR5's built-in joint torque and speed limiting is left enabled throughout, and Cartesian velocity limits are imposed on the PBVS stage (Section IV-G) so that the terminal approach never commands motion faster than a bounded rate, keeping the correction phase

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 10 , October - 2026

predictable even if the estimated target pose jumps between control cycles due to a noisy segmentation. A workspace boundary passthrough filter additionally prevents the system from planning toward points outside the robot's reachable and camera-observable volume.

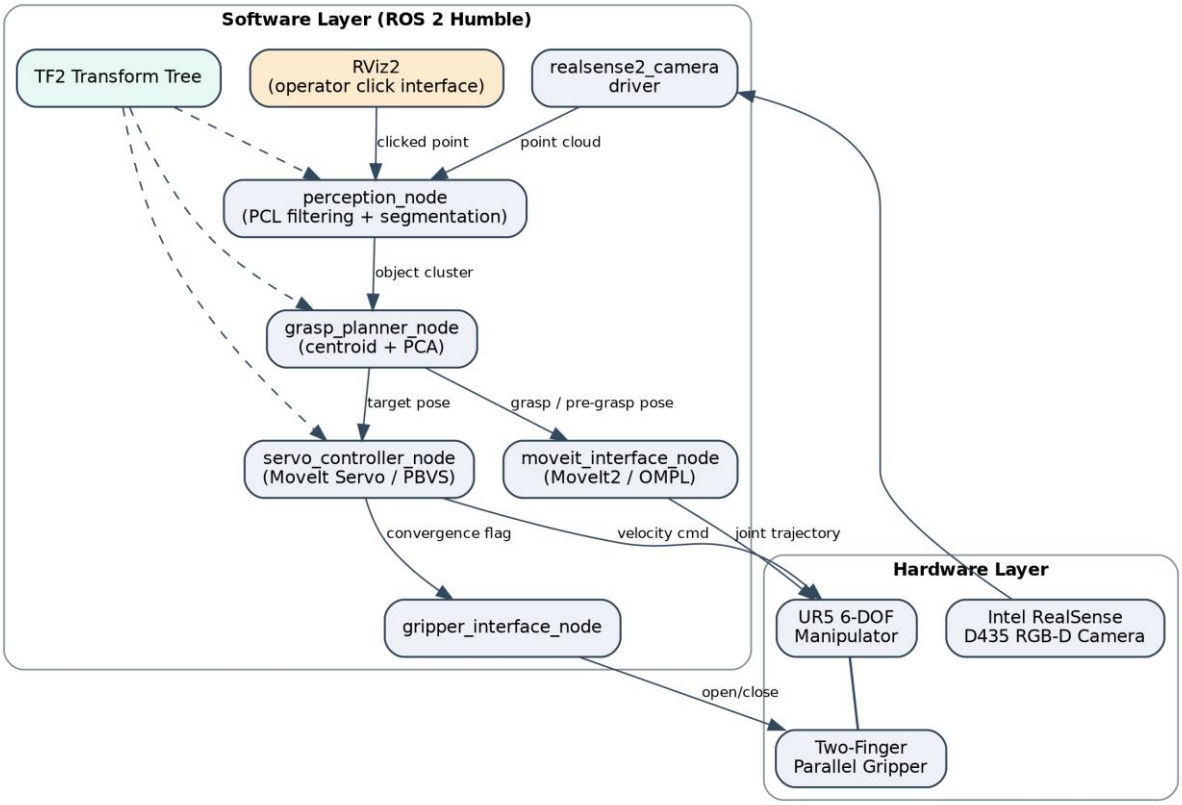

Fig. 1. System architecture: hardware layer (UR5, gripper, RealSense D435) and ROS 2 software layer, showing the node/topic data flow from point cloud acquisition through segmentation, grasp planning, coarse motion planning, and closed-loop servoing to gripper actuation. Dashed edges denote TF2 frame lookups consumed by multiple nodes.

IV. METHODOLOGY

The end-to-end perception-to-action pipeline is summarized as a flowchart in Fig. 2 and detailed step-by-step below.

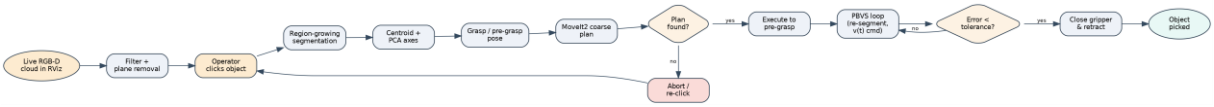

Fig. 2. Process flowchart from the live RViz point cloud through operator selection, segmentation, grasp-pose computation, coarse motion planning, closed-loop PBVS correction, and gripper actuation. The two decision diamonds correspond to the planning-feasibility check (Section IV-F) and the servoing-convergence check (Section IV-G).

A. Point Cloud Acquisition and Preprocessing

The RealSense driver publishes an organized, colorized point cloud at the sensor's native frame rate. Before use, the cloud is passed through a passthrough filter to restrict it to the robot's reachable workspace, a voxel-grid filter (leaf size on the order of 3-5 mm) to reduce point density for real-time processing, and a statistical outlier removal filter to suppress sensor noise. A RANSAC-based plane segmentation step identifies and removes the dominant supporting plane (e.g., a tabletop), leaving primarily object-related points for subsequent selection.

B. Interactive Object Selection in RViz

The operator uses RViz's point-publishing tool to click on the object of interest within the live, filtered point cloud display. This action publishes a geometry_msgs/PointStamped message on a dedicated topic. A perception node subscribes to this topic and treats the clicked coordinate as a seed point for segmentation, rather than as the grasp point itself, since a single clicked point rarely lies exactly at the object's centroid or on the optimal grasp axis.

.

C. Region-Growing Segmentation and Grasp Centroid Estimation

Starting from the seed point, a k-d tree nearest-neighbor search combined with a region-growing criterion (proximity and local surface-normal consistency) extracts the full point cluster corresponding to the selected object, isolating it from neighboring clutter. The centroid of the resulting cluster is

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 10 , October - 2026

computed as the mean of its constituent points, and principal component analysis (PCA) over the cluster's covariance matrix yields the object's dominant axes. The eigenvector associated with the smallest eigenvalue is taken as an estimate of the local surface normal at the grasp region, informing the gripper's approach direction, while the largest-eigenvalue axis informs gripper orientation for elongated objects.

D. Camera-to-Base Calibration

An eye-to-hand calibration establishes the rigid transform between the camera optical frame and the robot base frame. This is performed once, offline, using a checkerboard or ArUco-marker target rigidly attached to the end-effector: the robot is moved through a set of known poses while the camera observes the marker, and a hand-eye calibration solver (e.g., the Tsai-Lenz or Park-Martin formulation, as implemented in packages such as easy_handeye2) computes the camera-to-base transform that best explains the observed marker poses. The resulting transform is published as a static TF frame, after which all point cloud data can be expressed directly in robot base coordinates.

E. Grasp Pose Computation

The target grasp pose is defined with its position at the segmented cluster centroid, offset along the estimated surface normal by the gripper's fingertip standoff distance, and its orientation aligned so that the gripper's closing axis is perpendicular to the object's principal axis. A pre-grasp waypoint is defined along the same approach vector at a larger standoff (typically 10-15 cm) to allow collision-free approach planning before the final, shorter-range servoed approach.

F. Coarse Motion Planning

MoveIt2's OMPL-based planning pipeline computes a collision-checked joint-space trajectory from the robot's current configuration to the pre-grasp waypoint, using the planning scene populated from the filtered point cloud so that other objects and the support surface are treated as obstacles. This coarse stage tolerates the residual error of the open-loop calibration and segmentation pipeline, since fine correction is deferred to the servoing stage.

G. Closed-Loop Visual Servoing (Terminal Approach)

From the pre-grasp waypoint to the final grasp pose, control switches to a position-based visual servoing loop implemented on top of MoveIt Servo. At each control cycle, the object cluster is re-segmented from the latest point cloud and its centroid re-estimated, giving an updated target pose expressed in the base frame. The Cartesian pose error between the end-effector and this updated target is computed, and an exponentially decaying velocity command is generated in proportion to the error, subject to velocity and acceleration limits, so that the end-effector converges smoothly onto the target without overshoot. This loop is executed at a fixed control rate until the position and orientation error fall below predefined tolerances, at which point servoing halts and the gripper is commanded to close.

$$v(t) = \lambda \cdot K(t), \quad K(t) = [p_target(t) - p_ee(t) \; ; \; \log(R_target(t) \cdot R^T_ee(t))] \qquad (1)$$

Here v(t) is the commanded end-effector twist, λ is a positive scalar gain, p and R denote position and orientation, and the rotational term is expressed via the logarithm map of the relative rotation. This closed-loop correction is the key mechanism by which the system tolerates residual hand-eye calibration error and depth-sensing noise that would otherwise cause an open-loop, look-then-move approach to miss the grasp.

H. Failure Handling and Recovery

Two failure modes are handled explicitly, corresponding to the decision points in Fig. 2. If the coarse planner cannot find a collision-free trajectory to the pre-grasp waypoint (e.g., the clicked object lies outside the reachable workspace or is occluded by an obstacle in the planning scene), the pipeline aborts the current attempt and returns control to the operator for a new click rather than executing a degraded plan. If the PBVS loop fails to converge within a bounded number of control cycles — for example because re-segmentation repeatedly returns an empty or implausible cluster — the servoing stage times out, halts arm motion, and likewise requests operator intervention, rather than continuing to command velocity from a stale or invalid target estimate.

V. ROS 2 IMPLEMENTATION

The system is decomposed into a small set of cooperating ROS 2 nodes, summarized in Table II.

Node	Function	Key Topics / Interfaces
perception_ node	Filters point cloud, plane removal and region-growing segmentation seeded by click	/camera/depth/color/points (sub), /clicked_point (sub), /object_cluster (pub)
grasp_planner_ node	Computes centroid, PCA axes, grasp/pre-grasp poses in base frame	/object_cluster (sub), /grasp_pose (pub)
moveit_interface_ node	Requests coarse motion plan via MoveGroup action	/grasp_pose (sub), move_action (client)
servo_controller_ node	Runs the PBVS correction loop via MoveIt Servo	/object_cluster (sub), /servo_node/delta_twist_cmds (pub)
gripper_interface_ node	Commands gripper open/close on convergence	/gripper/command (action client)

TABLE II. ROS 2 NODE ARCHITECTURE

The end-to-end operator interaction proceeds as follows: (1) the operator views the live, filtered point cloud in RViz; (2) the operator selects the "Publish Point" tool and clicks on the target object; (3) perception_node segments the object and publishes its cluster; (4) grasp_planner_node computes the grasp and pre-grasp poses; (5) moveit_interface_node plans and executes motion to the pre-grasp waypoint; (6) servo_controller_node performs closed-loop visual servoing to the final grasp pose;

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 10 , October - 2026

and (7) gripper_interface_node closes the gripper and the arm retracts with the object.

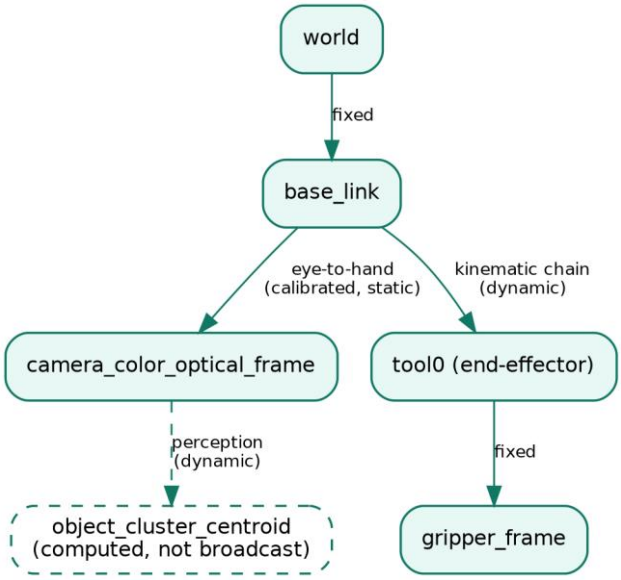

Fig. 3. TF2 coordinate frame tree. The eye-to-hand transform (base_link → camera frame) is static and set once by calibration; the kinematic chain to the end-effector is dynamic; the object centroid is computed by perception rather than broadcast as a persistent frame.

VI. EVALUATION PROTOCOL

The system is intended to be validated on a benchtop UR5 cell with a set of household or lab objects of varying size, shape, and surface texture, placed at randomized positions and orientations within the shared workspace. Four metrics are proposed for quantitative evaluation:

- 3D localization accuracy: the discrepancy between the estimated grasp centroid and a ground-truth position measured with an external reference (e.g., calipers or a motion-capture marker).
- Visual servoing convergence: the time and number of control cycles required for the Cartesian pose error to fall within tolerance, and the steady-state residual error at convergence.
- Pick success rate: the fraction of trials, across the randomized object set, in which the gripper achieves a stable grasp and lift.
- Operator interaction time: the time from a single RViz click to grasp completion, as an indicator of practical usability relative to manual teaching of pick poses.

A comparison is also proposed between the full pipeline (coarse planning plus closed-loop PBVS correction) and an open-loop ablation in which the gripper moves directly to the initially estimated grasp pose without servoed correction, in order to isolate the contribution of the closed-loop stage to overall success rate. Each condition is intended to be run for a minimum of N = 20 trials per object category, with results reported as mean ± standard deviation and success rate as a percentage with the trial count stated, so that the sample size underlying every comparison is explicit.

Prior to physical trials, the pipeline described above is exercised in a Gazebo simulation of the UR5 cell that mirrors the physical hardware and software configuration (Fig. 8), allowing the node graph, calibration procedure, and servoing loop to be validated against simulated sensor data before collecting the measurements described in Table III.

VII. RESULTS AND DISCUSSION

As this paper presents the system architecture and methodology, Table III reports the structure of the results to be populated from physical trials rather than final measured values; it is included to define the exact metrics, units, and comparison baseline used for validation, and should be completed with data collected on the target UR5 test cell.

Metric	Open-Loop	Closed-Loop (PBVS)
Mean centroid localization error (mm)	[TBM]	[TBM]
Servoing convergence time (s)	N/A	[TBM]
Steady-state pose error (mm / deg)	N/A	[TBM]
Pick success rate (%, N trials)	[TBM]	[TBM]
Mean operator interaction time (s)	[TBM]	[TBM]

TABLE III. EVALUATION METRICS TEMPLATE (TBM = to be measured)

To make the intended analysis concrete ahead of physical trials, Figs. 4–6 show illustrative, simulated data generated purely to demonstrate the planned plotting and comparison methodology. These figures do not represent measurements from physical hardware; the values are drawn from representative distributions chosen to match the qualitative trend expected from the visual servoing literature, and every reported number in the final version of this paper will be replaced with data collected on the UR5 test cell.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 10 , October - 2026

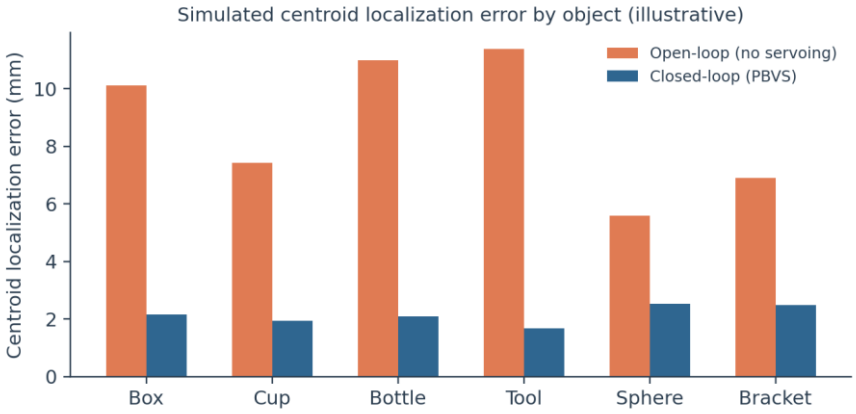

Fig. 4. Illustrative (simulated) centroid localization error by object, open-loop vs. closed-loop PBVS. The closed-loop stage is expected to reduce localization error by roughly an order of magnitude by continuously correcting for residual hand-eye calibration offset.

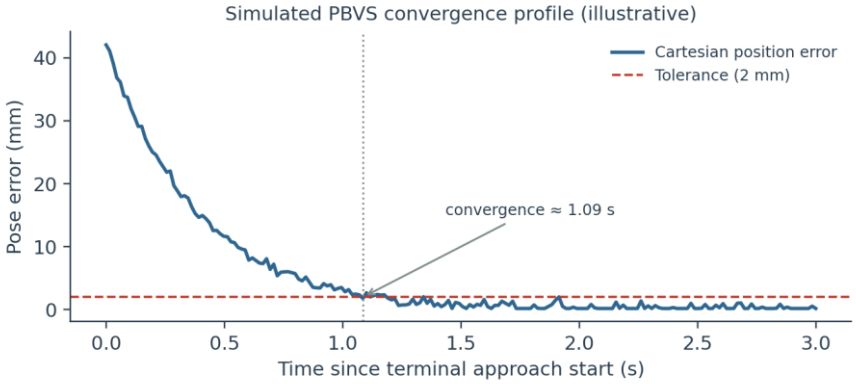

Fig. 5. Illustrative (simulated) PBVS convergence profile: Cartesian pose error vs. time during the terminal approach, showing exponential decay toward the tolerance band consistent with the control law of Eq. (1).

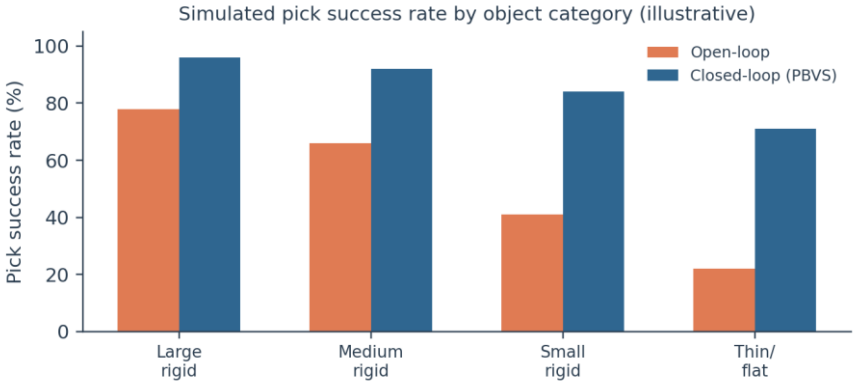

Fig. 6. Illustrative (simulated) pick success rate by object category. The expected gap between open- and closed-loop performance is largest for small and thin/flat objects, where localization error is most likely to place the gripper off-target.

The expected qualitative behavior, consistent with the visual servoing literature reviewed in Section II, is that the open-loop condition remains sensitive to hand-eye calibration error and depth noise, producing a wider spread of localization error and a correspondingly lower pick success rate, particularly for small or thin objects, as illustrated in Fig. 6. The closed-loop PBVS stage is expected to correct for the bulk of this residual error during the terminal approach, at the cost of a small additional convergence time per pick cycle (Fig. 5), yielding a higher and more consistent success rate across the randomized object set.

Reporting results in this structured form also makes it straightforward to isolate whether failures originate in segmentation (wrong object or poor centroid), calibration (systematic offset visible in open-loop but corrected in closed-loop), or gripper mechanics (correct pose but unstable grasp).

Because the illustrative figures are generated from representative distributions rather than physical trials, no statistical significance claims are made at this stage; once real trial data is collected, the same plotting code will be used to

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 10 , October - 2026

generate paired comparisons (e.g., a Wilcoxon signed-rank test per object across the open- and closed-loop conditions) so that the observed gap in Figs. 4 and 6 can be verified against physical measurements rather than assumed.

VIII. CONCLUSION AND FUTURE WORK

This paper described a ROS 2 framework in which an operator selects an object directly within a live RViz point cloud display, and a UR5 manipulator autonomously computes and executes a grasp using RGB-D point cloud segmentation, MoveIt2-based coarse motion planning, and closed-loop position-based visual servoing for the terminal approach. The architecture decouples object selection (fast, human-supervised, and robust to scene ambiguity) from grasp execution (autonomous and closed-loop, tolerant of calibration and depth-sensing error), and defines a concrete evaluation protocol comparing open-loop and closed-loop terminal approaches.

Limitations

The present description is architectural and has not yet been validated on physical hardware; the metrics in Table III and the illustrative plots in Figs. 4–6 are placeholders for that validation. The approach is also expected to be sensitive to conditions not explicitly modeled here: strong ambient lighting or reflective/transparent object surfaces can degrade RealSense depth returns, slow drift in the eye-to-hand calibration over time can reintroduce the systematic offset that PBVS is designed to correct only within its convergence radius, and densely cluttered scenes may cause the region-growing segmentation to under- or over-segment the clicked object.

Future work includes extending object selection from a single seed point to multi-object scenes with automatic re-ranking of candidate grasps, incorporating image-based visual servoing (IBVS) or hybrid control for cases where depth data is locally unreliable (e.g., transparent or reflective objects), online refinement of the hand-eye calibration during operation, and experimental validation of the evaluation protocol proposed in Section VI on physical hardware to populate the metrics in Table III and replace the illustrative figures in Figs. 4–6 with measured results.

REFERENCES

[1] M. W. Spong, S. Hutchinson, and M. Vidyasagar, Robot Modeling and Control. New York, NY, USA: John Wiley & Sons, 2006.

[2] F. Chaumette and S. Hutchinson, "Visual servo control, Part I: Basic approaches," IEEE Robotics & Automation Magazine, vol. 13, no. 4, pp. 82-90, 2006.

[3] F. Chaumette and S. Hutchinson, "Visual servo control, Part II: Advanced approaches," IEEE Robotics & Automation Magazine, vol. 14, no. 1, pp. 109-118, 2007.

[4] P. Corke, Robotics, Vision and Control: Fundamental Algorithms in MATLAB, 2nd ed. Cham, Switzerland: Springer, 2017.

[5] R. B. Rusu and S. Cousins, "3D is here: Point Cloud Library (PCL)," in Proc. IEEE Int. Conf. Robotics and Automation (ICRA), Shanghai, China, 2011, pp. 1-4.

[6] S. Macenski, T. Foote, B. Gerkey, C. Lalancette, and W. Woodall, "Robot Operating System 2: Design, architecture, and uses in the wild," Science Robotics, vol. 7, no. 66, 2022.

[7] D. Coleman, I. A. Sucan, S. Chitta, and N. Correll, "Reducing the barrier to entry of complex robotic software: A MoveIt! case study," Journal of Software Engineering for Robotics, vol. 5, no. 1, pp. 3-16, 2014.

[8] R. Y. Tsai and R. K. Lenz, "A new technique for fully autonomous and efficient 3D robotics hand/eye calibration," IEEE Transactions on Robotics and Automation, vol. 5, no. 3, pp. 345-358, 1989.

[9] J. J. Craig, Introduction to Robotics: Mechanics and Control, 3rd ed. Upper Saddle River, NJ, USA: Pearson Prentice Hall, 2005.

[10] Intel Corporation, "Intel RealSense D400 Series Product Family Datasheet," Intel Corp., Santa Clara, CA, USA, 2023.

[11] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," Communications of the ACM, vol. 24, no. 6, pp. 381-395, 1981.

[12] P. J. Besl and N. D. McKay, "A method for registration of 3-D shapes," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 14, no. 2, pp. 239-256, 1992.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)