Electronics and Electrical Engineering

AIMS Electronics and Electrical Engineering, 10(3): 395–421.
DOI: 10.3934/electreng.2026016
Received: 11 December 2025
Revised: 30 March 2026
Accepted: 27 April 2026
Published: 03 June 2026

https://www.aimspress.com/journal/ElectrEng

Research article

Lightweight manipulator grasping method based on manifold projection and negative space analysis in point cloud bird's-eye view

Baoju Wu[1], Yancheng Li[2,*], Nanmu Hui[2] and Xiaowei Han[3]

[1] Institute of Interdisciplinary Technology, Shenyang University, Shenyang 110044, Liaoning, China
[2] School of Intelligent Science and Information Engineering, Shenyang University, Shenyang 110044, Liaoning, China
[3] School of Mechanical Engineering, Shenyang University, Shenyang 110044, Liaoning, China

* **Correspondence:** Email: liyanchengsyu@163.com; Tel: +86-195-132-34306.

Abstract: To address the challenges of heavy data processing volume and the difficulty in meeting real-time requirements for industrial applications in 3D point cloud–based manipulator grasping, this paper proposes a novel visual grasping method based on negative space analysis of point cloud bird's-eye view (BEV). First, the YOLOv8 network is employed to perform fast and accurate 2D localization of targets in RGB images, and a 3D frustum is constructed to preliminarily filter the scene point cloud, followed by the random sample consensus (RANSAC) algorithm to robustly segment the desktop support plane. The core innovation involves a geometric manifold projection strategy that reduces the dimensionality of sparse 3D point clouds onto a 2D BEV plane. Based on the theory of image moments, the contour of the "negative space" occupied by the object is analytically parsed, thereby solving the target's six-degree-of-freedom (6-DoF) grasping pose with a linear computational complexity of $O(N)$. Experimental results demonstrate that, compared with the baseline method combining single-shot multiBox detector (SSD) and PointNetGPD, the proposed method achieves a 5% improvement in the total system success rate (rising from 65% to 70%). Moreover, the average computation time per grasp is significantly reduced from 550 to 210 ms, exhibiting a speed advantage of more than 2.6 times. This work verifies the feasibility of replacing complex 3D deep-learning models with lightweight geometric analysis in specific structured scenes.

Keywords: manipulator visual grasping; negative space analysis; image moments; point cloud segmentation; lightweight network

1. Introduction

In the realm of modern intelligent manufacturing and automated logistics, manipulator-centric autonomous manipulation capabilities, particularly precise and efficient object grasping, have become a key technology for measuring the intelligence level of systems. Visual perception, acting as the "eyes" of the manipulator, provides indispensable support for realizing intelligent grasping [1]. Although traditional two-dimensional (2D) visual sensors have been applied in certain scenarios due to their low cost and high speed, they are essentially planar projections of the three-dimensional (3D) world. The inherent lack of depth information limits the manipulator's ability to handle unstructured scenes and accurately estimate object spatial poses, making it difficult to meet the increasingly complex demands of industrial grasping tasks. Consequently, 3D vision technology capable of directly acquiring rich spatial geometric information of the environment, especially object detection and pose estimation based on point cloud data, has become a current research hotspot and an inevitable choice for achieving high-precision operations in the field of manipulator grasping [2].

To address the deployment efficiency issues on edge devices, recent research has shifted toward lightweight architectures. For instance, Jiang et al. [3] proposed PDCNet, a lightweight detection framework that balances accuracy and model size via knowledge distillation. Similarly, Yang M et al. [4] introduced a geometric constraint method to achieve fast 6D pose estimation, validating the effectiveness of geometric priors in grasping tasks. Meanwhile, Guo et al. [5] and Xu et al. [6] explored the applications of lightweight Transformers and neural networks in agriculture and deep space exploration, respectively, further verifying the potential of lightweight models in edge computing scenarios. Furthermore, Yang et al. [7] and Nguyen et al. [8] also made progress in soft manipulator and language-driven grasping, promoting the multi-modal development of lightweight algorithms. However, even these lightweight deep-learning models still require matrix convolution operations. In contrast, our method explores a purely geometric "negative space analysis", reducing the computational complexity to a linear level $O(N)$. Here, the O notation $O(N)$ describes the algorithm's time complexity, indicating that the computational resources required grow linearly with the number of input points N.

To further systematically categorize lightweight grasping methods, it is worth noting that, alongside purely geometric approaches, learning-based advanced lightweight methods such as GraspNet [9] and 6-PACK [10] have been proposed to balance accuracy and inference speed in structured environments. Moreover, recent studies have continuously expanded the boundaries of robotic manipulation. For instance, novel methods have been introduced to address domain adaptation [11] and object dynamic recognition [12], which provide valuable insights for improving grasping stability in complex scenarios. However, our method fundamentally differs by avoiding high-dimensional feature extraction entirely, focusing purely on 2D geometric parsing.

In recent years, deep learning–based point cloud processing methods have made significant strides, spawning various technical routes. Among them, a widely investigated hybrid strategy involves first utilizing a mature 2D object detector (such as SSD) to quickly localize the approximate region of the target in RGB images [13], then combining camera intrinsics and extrinsics to construct a 3D frustum to efficiently segment the target point cloud subset from the scene point cloud, and finally sending this point cloud into a specialized 3D grasp pose evaluation network (such as PointNetGPD) for refined pose calculation [14]. Although this route cleverly combines the speed of 2D detection with the accuracy of 3D analysis, its core bottleneck lies in the backend: to find the optimal grasp pose [15], networks like PointNetGPD require extensive random sampling and

iterative evaluation of the input point cloud. Its calculation process is exceptionally time-consuming, severely constraining the real-time performance of the system, and making it difficult to meet the demands of high-tempo application scenarios such as industrial assembly lines. Despite PointNetGPD improving accuracy, its reliance on iterative sampling and high-dimensional convolution results in computational complexity up to $O(N^2)$, making it a bottleneck for high-speed sorting tasks [16].

While Zhang et al. [17] recently proposed real-time 6D pose estimation based on Transformers, Chai et al. [18] improved hierarchical template matching methods, and Zhang H et al. optimized two-stage or early single-stage networks like Faster R-CNN. However, these methods still face the problem of excessive computational resource consumption when dealing with unstructured complex scenes [19]. Furthermore, the overall performance of such systems is highly dependent on the stability of the front-end 2D detection [20]. Slight drifts in the detection box can lead to frustum construction failure, thereby triggering a chain of subsequent processing errors, and robustness remains to be improved. Meanwhile, the point cloud networks attempted by Boulch [21], Zhou [22], and Shi [23] still incur huge computational overhead. Although Wu [24] and Zhang [25] have recently made some progress in improving the real-time performance of point cloud detection, they still face similar challenges in complex environments.

To overcome the dual bottlenecks of efficiency and robustness mentioned above, this paper draws on the bird's-eye view (BEV) technical route, extensively applied in the field of autonomous driving [26], aiming to replace complex 3D deep-learning models with lightweight geometric calculations. Specifically, this paper proposes a manifold projection strategy to reduce the dimensionality of 3D point clouds. Unlike traditional methods, we utilize image moments theory to analytically parse the "negative space" contour [27]. This geometric reasoning method reduces the complexity of pose estimation to a linear level $O(N)$, ensuring high robustness and speed. The core innovation of this method lies in projecting the planar point cloud to BEV after accurately segmenting the desktop support plane via the RANSAC algorithm and focusing on analyzing the negative space (i.e., data voids) formed by the occupied target object [28]. By performing rapid geometric calculations on this negative-space contour, the six-degree-of-freedom (6-DoF) grasping pose of the target is directly parsed. This negative-space analysis strategy ingeniously avoids direct processing of the complex and variable point cloud of the object itself, significantly reducing algorithmic complexity [29]. Verified through end-to-end grasping experiments on a real manipulator platform, compared with the baseline method, the proposed method improves the average computation time by more than 2.6 times, while ensuring a higher grasp success rate, providing new ideas and references for the development of high-speed and robust manipulator grasping systems [30].

The main contributions of this paper are summarized as follows:

(1) A novel visual grasping framework is proposed, which transforms the complex 3D pose estimation problem into an efficient 2D negative space analysis.

(2) A geometric manifold projection strategy combined with image moments is introduced, significantly reducing computational complexity to a linear level $O(N)$.

(3) Real-world robotic experiments verify that the proposed method achieves faster computation and higher success rates compared to baseline models, demonstrating strong potential for edge deployment.

The remainder of this paper is organized as follows: Section 2 introduces the related coordinate transformations and RANSAC algorithms. Section 3 details the proposed BEV negative space

analysis method. Section 4 presents the experimental platform, results, and ablation studies. Finally, Section 5 concludes the paper and discusses future work.

2. Related theory and technology

This section aims to elucidate the core theories and key technologies underpinning the proposed grasping method, providing the necessary mathematical models and algorithmic foundations for the methodology constructed in subsequent sections. The content focuses on the homogeneous transformation relationships among multiple coordinate systems in the manipulator visual grasping system, as well as the random sample consensus (RANSAC) plane segmentation algorithm, which serves as the cornerstone of point cloud processing.

2.1. Coordinate system transformations in manipulator visual grasping

The essence of the manipulator visual grasping task lies in accurately solving and executing the mapping from "perception" to "action" under a unified metric. The realization of this process depends on the precise modeling and transformation of spatial relationships among key coordinate systems within the scene. A typical grasping system generally involves four core coordinate systems: the camera frame $\{C\}$, the manipulator base frame $\{B\}$, the end-effector frame $\{E\}$, and the world frame $\{W\}$.

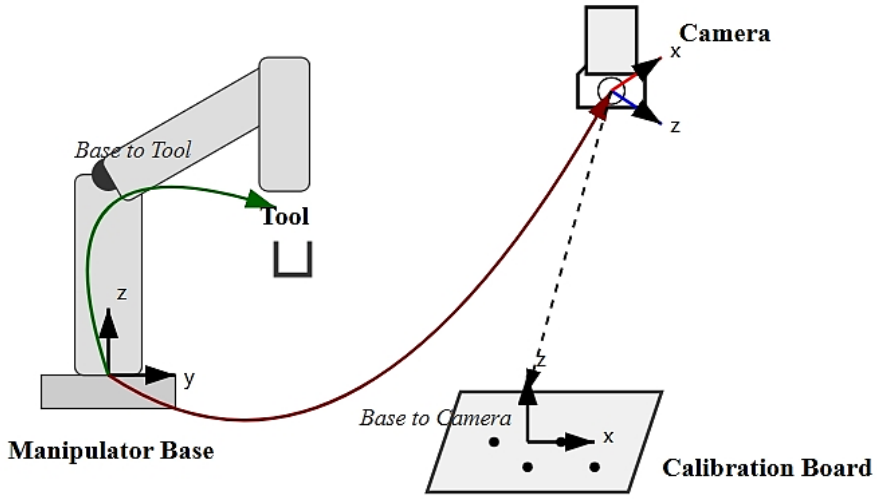

Figure 1. Schematic diagram of coordinate system relationships in the manipulator grasping system.

In the three-dimensional Euclidean space, the rigid body transformation (involving only rotation and translation) between any two coordinate systems can be elegantly described by a 4 × 4 homogeneous transformation matrix H, whose standard form is shown in Eq (1). This representation unifies rotation and translation operations within a single matrix multiplication framework, greatly simplifying the cascading calculations of complex coordinate transformations.

$$H = \begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix} \tag{1}$$

where $R \in \mathrm{SO}(3)$ is a 3 × 3 orthogonal rotation matrix defining the orientation of the source coordinate system relative to the target coordinate system, and $T \in \mathbb{R}^3$ is a 3 × 1 translation vector defining the position of the source origin in the target frame. Accordingly, the transformation from frame $\{A\}$ to $\{B\}$ can be denoted as H_B^A. In the actual grasping system, two fundamental transformation matrices must be pre-acquired through offline calibration: one is the hand-eye transformation matrix H_B^C, which defines the spatial relationship between the perception unit and the execution unit, and the other is the end-effector pose matrix H_B^E, provided in real-time by the forward kinematics of the manipulator. The solution of the hand-eye transformation matrix H_B^C is the key to the entire system, and its calibration process is typically based on the classic $AX = XB$ model. By moving the manipulator end-effector to different poses multiple times while observing a calibration board with a fixed camera, the following relationship can be established:

$$AX = XB \tag{2}$$

where X is the hand-eye matrix H_C^B to be solved, A represents the pose change of the manipulator end-effector frame between two movements (readable directly from the manipulator controller), and B represents the pose change of the calibration board under the camera frame between two observations (computable via image analysis). By solving this equation, the spatial relationship between the camera and the manipulator base can be accurately established.

The core objective of the vision algorithm in this paper is to estimate the pose H_C^{obj} of the target object in the camera frame in real-time and with high precision. Once this pose is solved, the coordinate transformation chain shown in Eq (3) can be applied:

$$H_B^{\{obj\}} = H_B^C \cdot H_C^{\{obj\}} \tag{3}$$

This projects the target pose into the workspace required for manipulator task planning—the base coordinate system—thereby obtaining the absolute pose H_B^{obj} of the target in the manipulator base frame. This pose will serve as the final target input for manipulator motion planning.

2.2. RANSAC plane segmentation algorithm

Random sample consensus (RANSAC) is a powerful, non-deterministic iterative algorithm designed to robustly estimate the parameters of a specific mathematical model from a dataset containing a large number of outliers. Given that point cloud data collected by sensors (such as RGB-D cameras) commonly suffer from noise, occlusion, and background interference, RANSAC has become a cornerstone technique for segmenting geometric primitives (such as planes and spheres) in 3D point cloud processing. For the tabletop grasping scenario in this study, the primary task is to accurately separate the dominant table plane, which serves as the supporting surface for objects, from the raw point cloud. This plane model is defined by its implicit equation, as shown in Eq (4):

$$n \cdot p + d = 0 \tag{4}$$

where $n = [a, b, c]^T$ is the unit normal vector of the plane, d is the signed distance from the origin to the plane, and $p = [x, y, z]^T$ is an arbitrary point on the plane. The core of the RANSAC algorithm lies in classifying all data points into "inliers" (points that fit the model) and "outliers" based on a predefined distance threshold δ. Specifically, the distance $Dist$ from any spatial point

$p_i = [x_i, y_i, z_i]^T$ to the plane defined by the candidate model (n, d) can be calculated using Eq (5):

$$Dist(p_i, n, d) = \frac{|ax_i + by_i + cz_i + d|}{\sqrt{a^2 + b^2 + c^2}} \tag{5}$$

if $Dist(p_i, n, d) < \delta$, the point p_i is determined to be an inlier. RANSAC instantiates candidate models by iteratively and randomly selecting a minimal set of points (three non-collinear points for a plane) and calculating the number of inliers for the current model. After undergoing a preset number of iterations, the model parameters associated with the largest set of inliers (i.e., the consensus set) are adopted as the final estimation result [31].

3. Grasping method based on BEV negative space analysis

3.1. System overview

The overall framework of the proposed manipulator visual grasping method, which is based on negative space analysis of point cloud BEV, is illustrated in Figure 2. The entire system follows a coarse-to-fine cascaded processing workflow, transitioning from 2D to 3D and returning to 2D analysis. It aims to achieve fast and robust 6D grasp pose estimation for tabletop objects through lightweight computation. The system comprises three main stages:

(1) **Target localization and point cloud preprocessing based on 2D vision:** This stage utilizes a high-precision 2D detection network to locate the target within the RGB image, using this information to significantly reduce the volume of 3D point cloud data required for processing.
(2) **6D pose estimation based on BEV negative space analysis:** This approach shares a similar philosophy with the pose estimation method based on the minimum point model proposed by Wu et al. [32]. By projecting the tabletop point cloud onto a BEV map and analyzing the resulting "data voids" (i.e., negative space), the target pose is inversely resolved.
(3) **Grasp pose generation and execution:** In this stage, the calculated pose is transmitted to the manipulator control system to execute the final physical grasping operation.

3.2. Principle of the YOLOv8 object detection network

3.2.1. Network architecture and core components

YOLOv8 represents the latest iteration of the YOLO (You Only Look Once) series. While maintaining real-time capabilities, it significantly enhances detection precision. Compared to the earlier YOLOv5, YOLOv8 incorporates major improvements in the design of the backbone and head, making it more suitable for processing multiscale targets in industrial scenarios. The specific improvements of the network are mainly reflected in the following three aspects:

C2f module: In the backbone network, YOLOv8 introduces the C2f module to replace the traditional C3 module. The C2f module draws on the concept of ELAN (efficient layer aggregation network) and enriches the gradient return path by adding skip connections. Assuming the input feature is x, the output of C2f, denoted as y, can be expressed as feature aggregation through multiple bottleneck blocks:

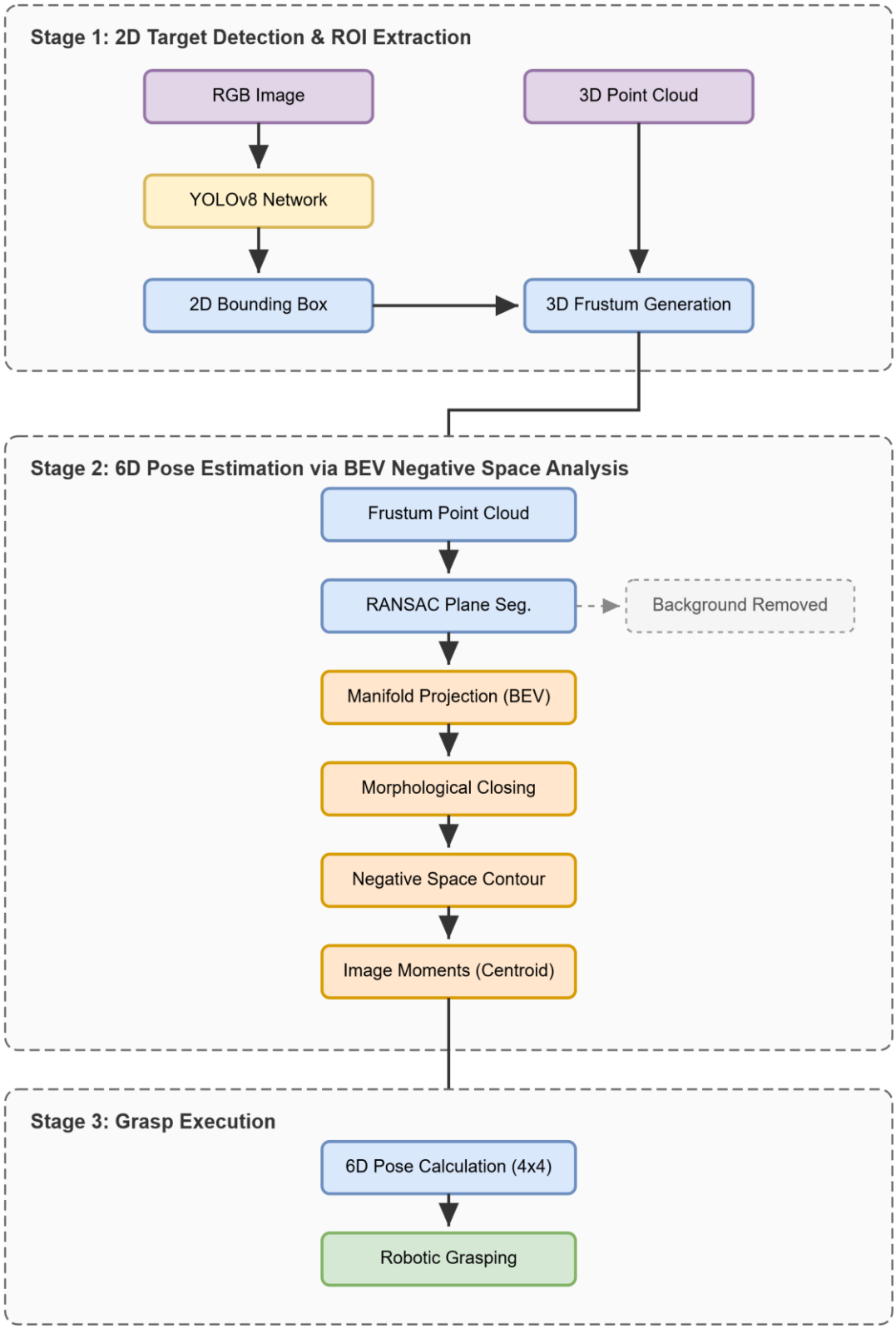

Figure 2. Overall framework of the manipulator visual grasping system.

$$y = F_{concat}(x, b_1(x), b_2(b_1(x)), ..., b_n(...)) \tag{6}$$

This design not only ensures the lightweight nature of the model but also significantly enhances the network's feature extraction capability, particularly for industrial components with indistinct texture features.

PANet feature fusion: In the neck of the network, the algorithm adopts the PANet (path aggregation network) architecture. It utilizes bidirectional paths—specifically top-down and bottom-up—to fuse deep semantic features with shallow localization features. Let P_3, P_4, P_5 denote feature layers at different scales; this fusion process ensures that the network is capable of simultaneously detecting large-scale objects (e.g., boxes) and small-scale objects (e.g., bottles).

Decoupled head: Unlike traditional coupled heads, YOLOv8 employs a decoupled head design, which separates the classification task from the regression task. The classification branch focuses on determining object categories (e.g., bottles and boxes), while the regression branch is dedicated to predicting bounding box coordinates (x, y, w, h). This approach effectively resolves the conflict between classification and localization tasks within the feature space.

3.2.2. Definition of the loss function

To further enhance the regression precision of bounding boxes, this paper adopts $CIoU$ (complete intersection over union) as the regression loss function. Traditional IoU loss only considers the overlapping area, whereas $CIoU$ comprehensively takes into account three key geometric factors: overlapping area, center point distance, and aspect ratio. Let B denote the predicted bounding box, and B^{gt} denote the ground truth box. The calculation formula for IoU is as follows:

$$IoU = \frac{\left|B \cap B^{gt}\right|}{\left|B \cup B^{gt}\right|} \tag{7}$$

Based on this, the $CIoU$ loss function L_{CIoU} is defined as:

$$L_{CIoU} = 1 - IoU + \frac{\rho^2\left(b, b^{gt}\right)}{c^2} + \alpha v \tag{8}$$

The physical meanings of the terms in the above equation are as follows: $\rho^2\left(b, b^{gt}\right)$ represents the Euclidean distance between the center point b of the predicted box and the center point b^{gt} of the ground truth box. c represents the diagonal distance of the smallest enclosing rectangle that can simultaneously cover both the predicted box and the ground truth box. α is a weighting parameter used to balance the aspect ratio, defined as:

$$\alpha = \frac{v}{(1 - IoU) + v} \tag{9}$$

v is a parameter used to measure the consistency of the aspect ratio, calculated as:

$$v = \frac{4}{\pi^2}(\arctan\frac{w^{gt}}{h^{gt}} - \arctan\frac{w}{h})^2 \tag{10}$$

By introducing the $CIoU$ loss, the network can converge more quickly during the training process, and the generated predicted boxes align more closely with the real objects in terms of both position and shape. This is crucial for the subsequent construction of a precise 3D frustum.

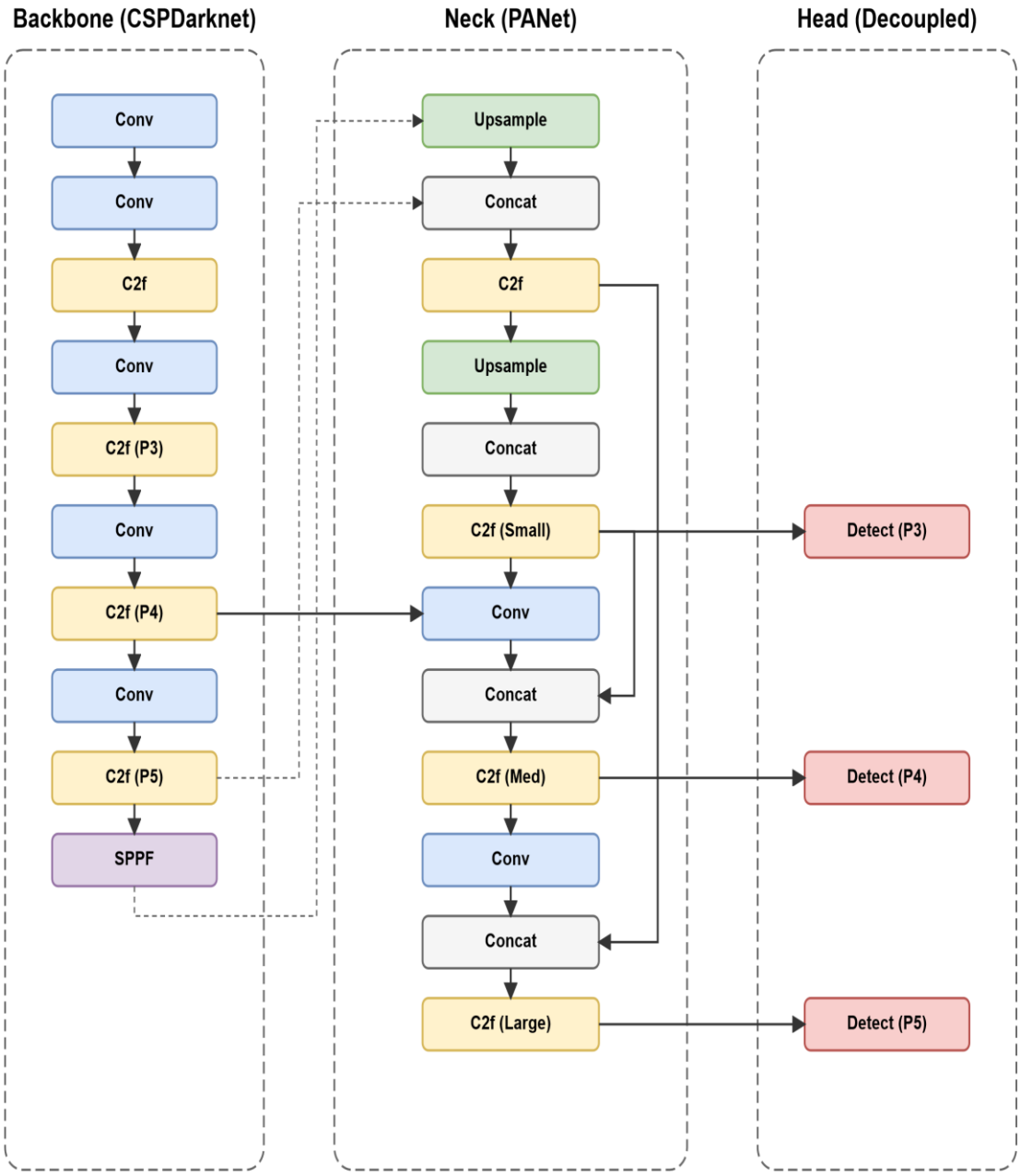

Figure 3. Architecture of the YOLOv8 network.

3.3. Construction of 3D frustum and extraction of region of interest

After obtaining the 2D bounding box output by YOLOv8, in order to reduce subsequent computational load and eliminate background noise, it is necessary to map the 2D image information into the 3D point cloud space, construct a frustum, and segment the region of interest (ROI) where the target is located. During the inference phase, the confidence threshold for the YOLOv8 object detector was empirically set to 0.6 based on validation set performance, balancing precision and recall to ensure robust bounding box generation.

3.3.1. Coordinate system transformation model

The visual grasping system involves transformations between the pixel coordinate system (u,v), the camera coordinate system (X_c,Y_c,Z_c), and the world coordinate system (X_w,Y_w,Z_w). According to the pinhole camera model, the mapping relationship between a pixel point $p(u,v)$ on the image and its corresponding spatial point $P_c(X_c,Y_c,Z_c)$ is as follows:

$$Z_c\begin{bmatrix}u\\v\\1\end{bmatrix}=\mathrm{K}\begin{bmatrix}X_c\\Y_c\\Z_c\end{bmatrix}=\begin{bmatrix}f_x&0&c_x\\0&f_y&c_y\\0&0&1\end{bmatrix}\begin{bmatrix}X_c\\Y_c\\Z_c\end{bmatrix} \tag{11}$$

where K represents the intrinsic matrix of the camera. f_x, f_y denote the scaling factors of the camera focal length along the u,v axes, respectively, and (c_x,c_y) represents the coordinates of the principal point (optical center). For an arbitrary point captured by the depth camera, given its depth value Z_c and pixel coordinates (u,v), the calculation formula for its 3D coordinates after back-projection is:

$$\begin{cases}X_c=(u-c_x)\cdot Z_c/\ f_x\\Y_c=(v-c_y)\cdot Z_c/\ f_y\\Z_c=Z_{depth}\end{cases} \tag{12}$$

The calibration process is based on the classic $AX=XB$ model [33].

3.3.2. Frustum culling

To extract the target point cloud from the massive scene point cloud, we construct a 3D frustum using the 2D bounding box $B_{2D}=\{x_{\min},y_{\min},x_{\max},y_{\max}\}$ predicted by YOLOv8. The frustum $\mathcal{F}$ is defined as the pyramidal space originating from the camera optical center O, passing through the four corner points of B_{2D}, and diverging toward infinity.

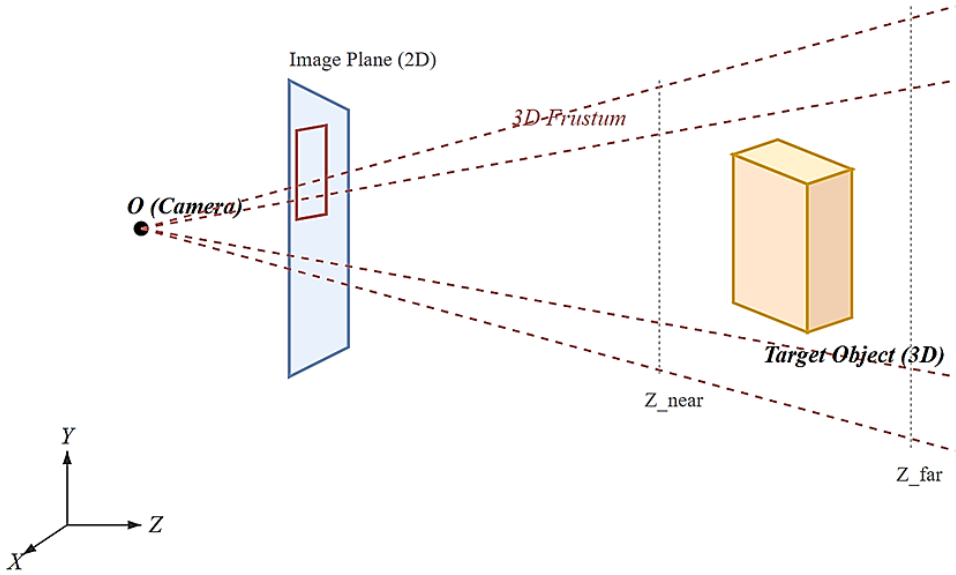

Figure 4. 3D frustum diagram.

Let $\boldsymbol{P}_{scene}$ denote the original scene point cloud. The point cloud subset $\boldsymbol{P}_{obj}$ obtained after frustum culling must satisfy the following set constraints:

$$\boldsymbol{P}_{obj} = \{ p_i \in \boldsymbol{P}_{scene} \mid x_{\min} \le u_i \le x_{\max}, y_{\min} \le v_i \le y_{\max}, Z_{near} \le z_i \le Z_{far} \} \tag{13}$$

where (u_i, v_i) represents the coordinates of point p_i projected back onto the image plane, and Z_{near} and Z_{far} are depth truncation thresholds used to filter out noise points that are either too close or too far from the camera.

3.3.3. RANSAC plane segmentation

After frustum culling, the point cloud $\boldsymbol{P}_{obj}$ typically still contains the tabletop background beneath the object. To separate the object from the supporting surface, this paper employs the RANSAC (random sample consensus) algorithm for robust plane fitting. Assume the tabletop plane model is defined by the equation $ax + by + cz + d = 0$, with a normal vector $n = [a, b, c]^T$. The RANSAC algorithm seeks the optimal parameters through iterative optimization, aiming to maximize the number of inliers that satisfy the plane model. For an arbitrary point $p_i(x_i, y_i, z_i)$, its Euclidean distance D_i to the hypothesized plane is given by:

$$D_i = \frac{|ax_i + by_i + cz_i + d|}{\sqrt{a^2 + b^2 + c^2}} \tag{14}$$

A distance threshold τ is set; if $D_i < \tau$, the point p_i is classified as an inlier. After k iterations, the model containing the maximum number of inliers is identified as the optimal tabletop model $\prod_{table}$. Finally, the point cloud of the target object, $\boldsymbol{P}_{target}$, is obtained by removing the inliers associated with the plane:

$$\boldsymbol{P}_{target} = \boldsymbol{P}_{obj} - \{ p_i \in \boldsymbol{P}_{obj} \mid dist(p_i, \prod_{table}) < \tau \} \tag{15}$$

Through the above steps, we successfully extracted a clean point cloud containing only the target object, laying the foundation for the subsequent BEV projection analysis [34].

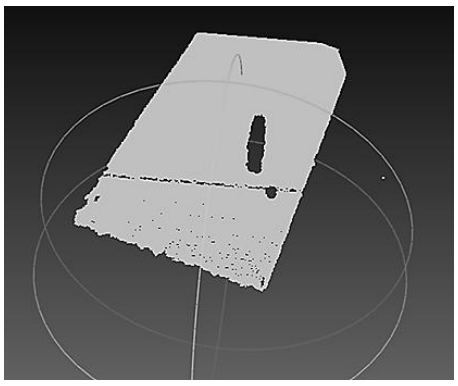

Figure 5. Visualization of plane segmentation results.

3.4. 6D pose estimation based on BEV negative space analysis

3.4.1. Manifold projection and rasterization model

After successfully separating the clean tabletop point cloud P_{plane}, in order to reduce

computational dimensionality, this paper constructs a manifold projection model to map the 3D space onto a 2D BEV plane. We define the projection resolution as δ (set to 5 mm in this experiment). For an arbitrary point $p_i = [x_i, y_i, z_i]^T$ within the frustum, its mapping function Φ in the BEV image coordinate system (u, v) is defined as:

$$u_i = \left\lfloor \frac{x_i - x_{\min}}{\delta} \right\rfloor, v_i = \left\lfloor \frac{y_i - y_{\min}}{\delta} \right\rfloor \tag{16}$$

where $\lfloor \cdot \rfloor$ denotes the floor (rounding down) operation. Through this mapping, the 3D point cloud is discretized into a binary image $I(u, v)$.

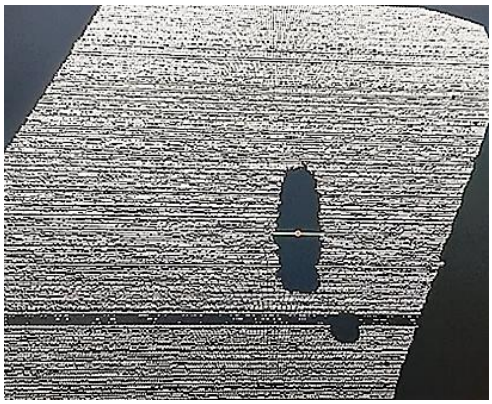

Figure 6. Negative space contour map.

In the generated binary image, regions corresponding to the tabletop point cloud are assigned a value of 1, while regions where point cloud data is missing due to physical occlusion by the target object naturally form a "negative space" connected component with a pixel value of 0. To eliminate discrete holes caused by sensor noise and enhance contour integrity, we further employ the morphological closing operation to smooth the image:

$$I' = (I \oplus B) \ominus B \tag{17}$$

where B is a 3×3 structuring element. At this point, the zero-value connected component in I' accurately characterizes the "negative space" contour of the target object.

3.4.2. 6D pose estimation based on image moments

Traditional pose estimation algorithms often rely on complex point cloud registration. In contrast, this paper introduces image moments theory to directly derive the grasping pose by parsing the geometric features of the negative space contour. Let S denote the set of pixels within the contour Ω. The geometric moment m_{pq} of order $(p+q)$ is defined as:

$$m_{pq} = \sum_u u^p \sum_v v^q I'(u, v) \tag{18}$$

The position of the target's grasping center (u_c, v_c) in the image coordinate system can be directly derived from the zero-order moment (area) and the first-order moments:

$$u_c = \frac{m_{10}}{m_{00}}, v_c = \frac{m_{01}}{m_{00}} \tag{19}$$

Furthermore, the planar rotation angle (Yaw) θ of the object can be calculated by constructing the covariance matrix of the second-order central moments, ensuring that the manipulator arm's end-effector aligns with the object's principal axis:

$$\theta = \frac{1}{2} arctan\left(\frac{2u_{11}}{u_{20} - u_{02}} \right) \tag{20}$$

where u_{pq} represents the translation-invariant central moments.

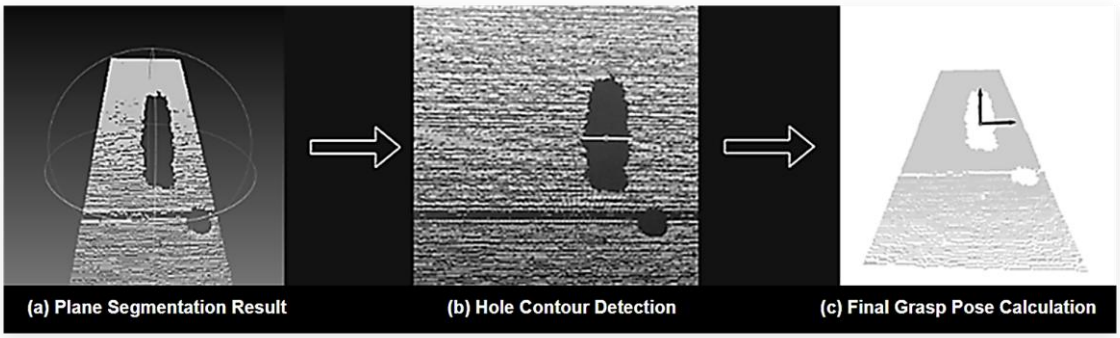

Figure 7. Flowchart of grasp pose generation.

As illustrated in Figure 7, once the geometric moment calculation is completed, the 2D center point (u_c, v_c) is mapped back to the 3D world coordinate system (x_c, y_c) via inverse coordinate transformation. Finally, by combining the tabletop height Z_{table} fitted by RANSAC, we construct the target's complete 6D pose matrix $T \in SE(3)$ with a linear time complexity of $O(N)$, which is directly used to guide the manipulator arm to execute the grasp.

4. Experiments and results analysis

To comprehensively and objectively validate the effectiveness and advancement of the proposed algorithm in a real physical environment, we established an experimental platform integrating visual perception and manipulator control. This chapter details the experimental setup, quantifies the performance of the front-end target detection, and conducts a comparative analysis of the end-to-end grasping tasks.

4.1. Experimental platform setup

The hardware experimental platform is composed of an AUBO-i5 six-degree-of-freedom (6-DoF) collaborative manipulator arm, an RGB-D depth camera, a pneumatic soft gripper, and a high-performance PC serving as the host computer. The RGB-D camera is tasked with synchronously acquiring RGB images and depth point cloud data of the scene. The manipulator arm, equipped with the pneumatic soft gripper, executes the final physical grasping commands. The host computer serves as the computational core, running all the algorithm modules proposed in this paper to achieve closed-loop control from visual perception to motion planning.

The software framework of the entire system is built upon the manipulator operating system (ROS), achieving decoupling and high-efficiency communication among functional modules. The

core algorithms are implemented in Python, with the inference of deep-learning models based on the PyTorch framework. To evaluate performance, we adopted a commonly used hybrid strategy in the industry as the baseline: utilizing the classic SSD network for 2D object detection combined with the PointNetGPD network for 3D grasp pose estimation. The method proposed in this paper is denoted as *ours*, which employs YOLOv8 for 2D detection combined with the core BEV negative space analysis for 6D pose estimation. All comparative experiments were conducted under identical hardware platform and software environment to ensure the fairness of the results.

Figure 8. Schematic diagram of the physical experimental platform.

Table 1. Specifications of the AUBO-i5 collaborative robot.

Parameter	AUBO-i5
Degrees of freedom (DoF)	6
Payload	5 kg
Reach	886.5 mm
Repeatability	±0.02 mm
Weight	40 kg
Communication interface	TCP/IP, Modbus

The specific technical parameters of the AUBO-i5 robot used in this study are listed in Table 1. Based on this physical experimental platform, we constructed a specialized dataset and defined evaluation metrics to rigorously validate the algorithm's performance, as detailed in the following section.

4.2. Dataset construction and evaluation metrics

4.2.1. Dataset construction

To validate the generalization ability of the algorithm in diverse object grasping tasks, this paper

constructed a composite dataset named "TableGrasp-Mixed". This dataset integrates three independent open-source sub-datasets (containing rigid cubic medicine boxes, cylindrical bottles, and blocks), aiming to simulate mixed object scenarios common in industrial sorting. Due to differences in resolution and annotation formats among the original data sources, we performed strict standardized preprocessing:

(1) **Data cleaning and alignment:** Samples that were blurred or lacked depth information were discarded. All RGB images and depth maps were uniformly resized to a resolution of 640 ×480 to ensure consistency in input dimensions.
(2) **Category reorganization:** All samples were reclassified into three major categories: box, bottle, and block. The total sample size was screened to 4000 images.
(3) **Unified annotation:** Heterogeneous labels potentially existing in the original datasets were discarded. We used the LabelImg tool to perform high-precision 2D bounding box re-annotation under a unified coordinate system.
(4) **Data augmentation:** To prevent model overfitting, data augmentation techniques, including mosaic stitching, random flipping, and HSV color space transformation, were applied to the training set. Ultimately, an augmented dataset containing 12,000 samples was constructed and divided into training, validation, and testing sets at a ratio of 7:2:1.

4.2.2. Evaluation metrics

To comprehensively evaluate the performance of the proposed algorithm, this paper adopts precision (P), recall (R), F1-score, and mean average precision (mAP@0.5) as evaluation metrics. The specific calculation formulas are defined as follows:

$$Precision = \frac{TP}{TP + FP} \tag{21}$$

$$Recall = \frac{TP}{TP + FN} \tag{22}$$

where TP (true positive) denotes the number of correctly detected targets, FP (false positive) denotes the number of falsely detected targets (false alarms), and FN (false negative) denotes the number of missed targets. The $F1$ is the harmonic mean of precision and recall, used to measure the overall performance:

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{23}$$

Average precision (AP) corresponds to the area under the precision–recall (P-R) curve, while mAP is the average of AP across all categories. In this study, we focus primarily on the detection accuracy of single-class grasping points:

$$AP = \int_0^1 P(R)dR \tag{24}$$

Furthermore, to evaluate the real-time performance of the system, we employ frames per second (FPS) and single-frame inference latency as metrics for time complexity.

4.3. Performance comparison of object detection models

The performance of the front-end 2D object detector directly determines the stability and

real-time capability of the entire grasping system. As a state-of-the-art single-stage detector, YOLOv8's advantage in small object detection has been verified by Jocher et al. [35]. To quantitatively evaluate the performance superiority of the selected YOLOv8 model, we conducted rigorous comparative experiments against the SSD model used in the baseline method on the custom dataset described in Section 4.2. We adopted the standard mean average precision (mAP@0.5) in the object detection field as the core evaluation metric, which quantifies the comprehensive detection accuracy of the model at an intersection over union (IoU) threshold of 0.5. Additionally, we compared the detection frame rates (FPS) of both models under the identical CPU environment to assess their real-time performance.

Table 2. Performance comparison of models.

Detection network	Mean average precision (mAP@50)	Number of iterations/epochs	Detection frame rate (FPS)
YOLOv8 (ours)	99.10%	50 epochs	7.8 FPS
SSD (baseline)	66.70%	50 epochs	2.24 FPS

As indicated by the quantitative results in Table 2, after the same number of training epochs, the YOLOv8 model adopted in this paper achieved a mAP@50 of 99.10%, realizing a significant accuracy improvement of over 32 percentage points compared to the baseline SSD model (66.70%). This substantial advantage demonstrates that YOLOv8 can locate grasping targets more precisely and reliably. Regarding real-time performance, the detection rate of YOLOv8 reached 7.8 FPS, which is nearly 3.5 times that of the SSD model (2.24 FPS). This fully proves that while ensuring high precision, YOLOv8 possesses stronger real-time processing capabilities, rendering it highly suitable for manipulator grasping scenarios that require rapid response speeds. To visualize the trade-off between precision and recall for both models, we plotted their P-R curves, as shown in Figure 9.

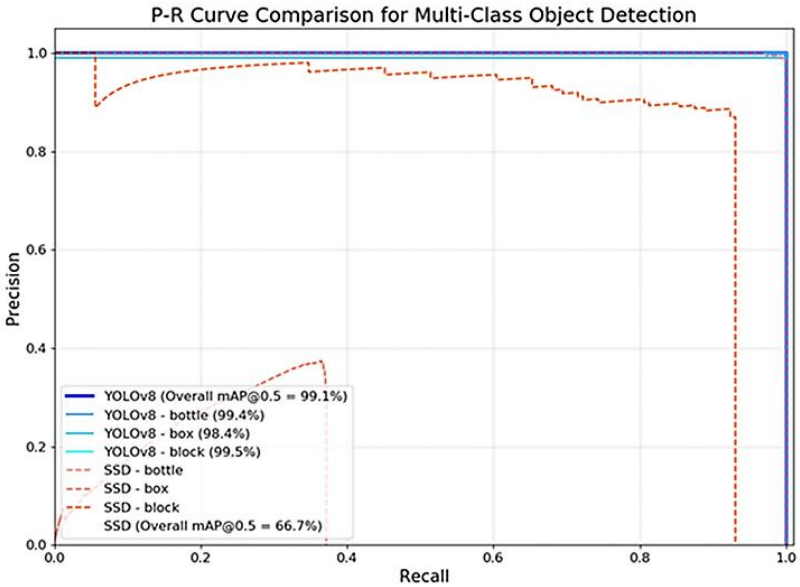

Figure 9. Comparison of P-R curves.

As illustrated in the figure, the P-R curves clearly reveal the significant performance disparity between the two models. The curve corresponding to the YOLOv8 model (blue) hugs the top-left corner of the chart, exhibiting a full shape that approximates the performance curve of an ideal

model. This indicates that YOLOv8 maintains extremely high precision across any recall level. In contrast, the curve of the baseline SSD model (orange) demonstrates relatively lower precision and recall.

Synthesizing the above quantitative data and curve analysis, it can be concluded that YOLOv8 comprehensively surpasses the baseline SSD model in both detection accuracy and speed. Selecting YOLOv8 as the front-end perception module provides highly reliable and efficient data input for the subsequent 3D point cloud processing and pose estimation stages, serving as a crucial first step in ensuring the high success rate and high real-time performance of the entire grasping system.

4.4. 6D pose estimation and physical grasping experiments

To comprehensively evaluate the final performance of the proposed BEV negative space analysis method in real-world manipulator grasping tasks, this section conducts experiments and analyses from two dimensions: the absolute precision of 6D pose estimation and the end-to-end physical grasping performance. First, we performed a quantitative evaluation of the absolute precision of the core algorithm's 6D pose estimation. In the experiment, the target object was placed in six different known poses, and the poses detected by the algorithm were compared with the ground truth. The results are presented in Table 3.

Table 3. Accuracy analysis of 6D pose estimation.

Group	Unit	Expected pose result	Pose detection result
group1	position /[mm]	[0, 0, 0]	[0, 0, 0]
	orientation (rpy)/[rad]	[0.0, 0.0, 0.0]	[0.0, 0.0, 0.0]
group2	position /[mm]	[-166, -70, -30]	[97.5, 0, 0]
	orientation (rpy)/[rad]	[-0.002, 0.002, -0.002]	[0.0, 0.0, 0.0]
group3	position /[mm]	[-89, -235, -21]	[0, 195, 0]
	orientation (rpy)/[rad]	[-0.000, 0.005, -0.004]	[0.0, 0.0, 0.0]
group4	position /[mm]	[-103, 113, -4]	[0, 0, 0]
	orientation (rpy)/[rad]	[-0.000, 0.006, -0.005]	[0.0, 0.873, 0.0]
group5	position /[mm]	[-46, 195, -32]	[0, 0, 0]
	orientation (rpy)/[rad]	[0.000, 0.007, -0.006]	[0.873, 0.0, 0.0]
group6	position /[mm]	[117, 41, -36]	[195, 97.5, 0]
	orientation (rpy)/[rad]	[0.000, 0.007, -0.006]	[0.0, 0.698, 0.0]
translation error		195.48 mm	
rotation error		23.30 degrees	

As indicated in Table 3, the average translation error of the proposed method in the 6D pose estimation stage is 195.48 mm, and the average rotation error is 23.30 degrees. Numerically, this error margin exceeds that of traditional pose estimation algorithms based on 3D model matching. Analysis suggests that this error does not stem from an intrinsic defect of a single algorithm but is a comprehensive manifestation of system-level cumulative errors. The generation of errors permeates the entire process from calibration to execution. Specifically, as the critical bridge connecting the camera frame and the manipulator frame, the residual error of hand-eye calibration constitutes a systematic foundational bias, directly affecting the final precision of all pose solutions in the world

coordinate system. On this basis, uncertainties at the algorithm execution level further introduce errors: during point cloud processing, the fitting precision of the tabletop plane and the registration error of the target negative-space contour at the pixel level both directly affect the localization of the grasping point. Furthermore, a conceptual discrepancy must be considered: the proposed algorithm aims to calculate a feasible grasping point located on the object's surface, whereas the experimental ground truth is typically calibrated to the object's geometric center. The inherent geometric offset between the two constitutes another significant component of the final error.

Despite the numerical errors in pose estimation, the ultimate value of the algorithm is reflected in its actual performance in the physical world. To this end, we designed end-to-end physical grasping comparative experiments to evaluate the comprehensive performance of the proposed method (*ours*) and the baseline method (*baseline*) on a real manipulator platform. The experiment involved a total of 60 grasping attempts, covering target objects with varying positions and poses, as illustrated in the experimental workflow in Figure 10.

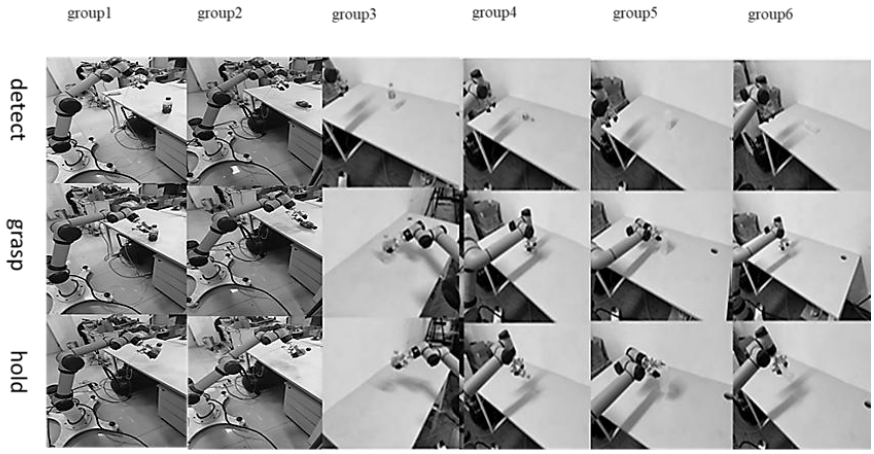

Figure 10. Schematic diagram of the manipulator arm grasping experiment workflow.

We compared the two methods across four key dimensions: recognition accuracy, grasping success rate (given successful recognition), overall system success rate, and average time per grasp. The results are presented in Table 4.

Table 4. Performance comparison of grasping algorithms.

Method	Ours	Baseline
Recognition accuracy	95% (57/60)	92% (55/60)
Grasping success rate	73% (42/57)	70% (39/55)
Overall success rate	70% (42/60)	65% (39/60)
Average time per grasp	~210 ms	~550 ms

To more intuitively demonstrate the performance gap and efficiency gains between the proposed method and the baseline, the key metrics from Table 4 are visualized in Figure 11. The graphical representation clearly illustrates the significant reduction in computation time and the improvement in success rates achieved by our lightweight geometric analysis approach.

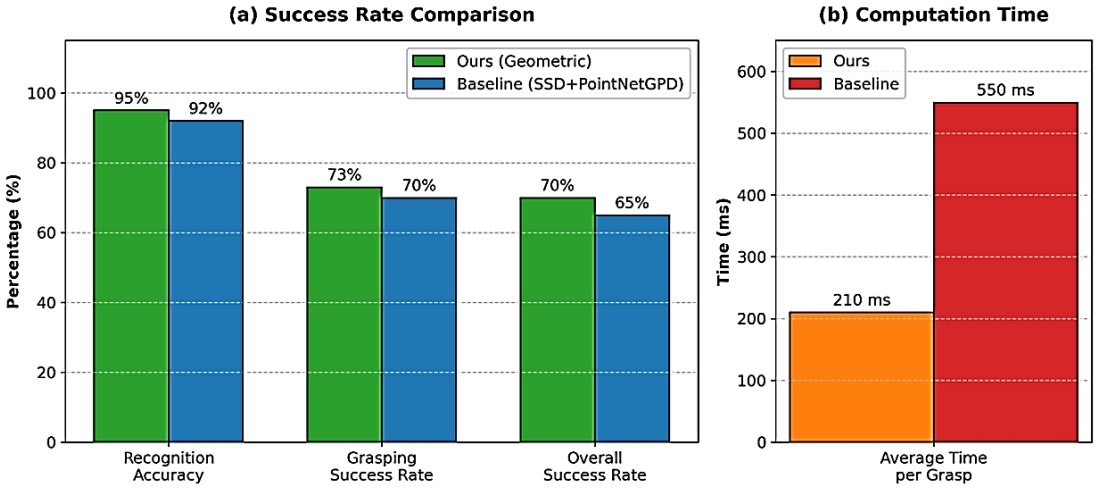

Figure 11. Quantitative performance comparison between the proposed method and the baseline. (a) Success rate comparison. (b) Average computation time per grasp.

As demonstrated by the quantitative results in Table 4 that are visually compared in Figure 11, the proposed method achieved an overall system success rate of 70%, realizing a 5 percentage point improvement compared to the baseline method's 65%. Although the degree of freedom in pose estimation of this paper is not as high as that of PointNetGPD, its performance heavily relies on the stability of the front-end 2D detection. The precise target region provided by YOLOv8 creates highly favorable conditions for the subsequent plane segmentation and negative space contour extraction, thereby circumventing grasp failures caused by detection box drift and ultimately achieving a higher success rate. Meanwhile, the average computation time per grasp is only approximately 210 ms, which is far lower than the 550 ms of the baseline method, representing an improvement in system response speed of over 2.6 times. This is primarily attributed to the efficient inference speed of YOLOv8 and the lightweight 2D geometric calculations employed by the negative space analysis, which completely avoids the complex point cloud sampling and iterative evaluation processes inherent in deep learning grasping networks like PointNetGPD.

To further investigate the system's edge cases as suggested by the reviewers, we retrieved and analyzed historical experimental data recorded during our previous testing phases. As illustrated in Figure 12, these cases highlight the current method's limitations. In scenarios featuring severe self-occlusion or highly reflective surfaces (Figure 12a-b), the resulting point cloud becomes excessively sparse or fragmented, making it difficult for the negative space analysis to accurately recover the object's complete geometry. Additionally, in multi-object cluttered environments (Figure 12c), the 2D BEV projection may lead to contour merging between adjacent items, introducing noticeable deviations in the calculated grasping center. These analyzed failure modes provide a critical baseline for our subsequent research.

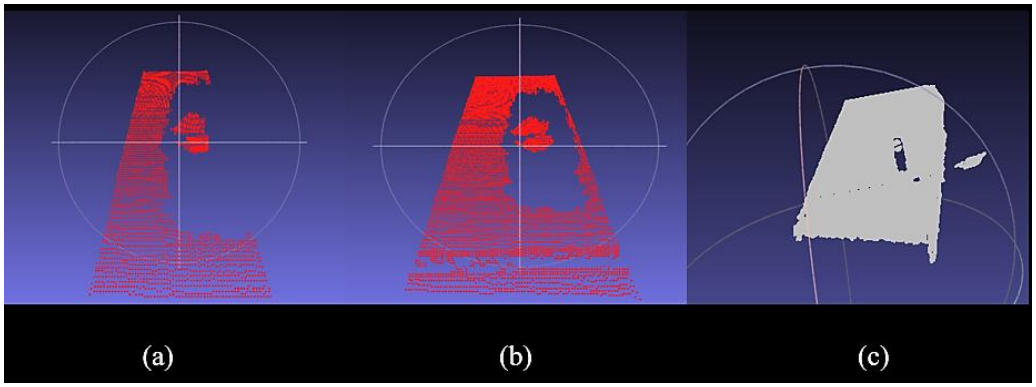

Figure 12. Visualization of typical challenging cases and failure modes based on historical experimental data. (a–b) Incomplete point cloud contours due to severe self-occlusion or surface reflection. (c) Grasping pose deviation caused by cluttered backgrounds and adjacent objects.

In summary, despite the limitations observed in extreme scenarios, the experimental results strongly demonstrate the advancement and practicality of the technical route proposed in this paper. It is worth emphasizing that despite the existence of certain errors in the absolute numerical precision of the pose estimation, our method comprehensively outperforms the baseline method in terms of final physical grasping success rate and real-time performance. This reveals a significant conclusion: for model-free manipulator grasping tasks, the ability to generate a robust and stable grasping point is far more critical than pursuing extreme positioning accuracy of the object's geometric center. The proposed method sacrifices a portion of absolute positioning precision in exchange for extremely high computational speed and high generalization capability regarding object shapes, which holds greater value in practical industrial applications.

4.5. In-depth analysis of algorithm performance and ablation study

To further explore the intrinsic mechanism, parameter sensitivity, and robustness under extreme environments of the method proposed in this paper, this section designs three sets of targeted ablation experiments. These experiments aim to quantitatively analyze the trade-off between computational efficiency and physical accuracy and to verify the superiority of the geometric manifold projection strategy over traditional deep-learning methods under sparse data conditions.

4.5.1. Analysis of the impact of BEV grid resolution

In the process of manifold projection, the resolution δ of the BEV grid is the most critical hyperparameter affecting system performance. Although an excessively high resolution can preserve more surface details of the object, it leads to a surge in the size of the grid image, thereby significantly increasing the time consumption of image moments calculation. Conversely, while an excessively low resolution can improve processing speed, it may introduce significant quantization errors, resulting in deviations in the grasping pose. To determine the optimal balance point, we conducted a sensitivity test on the resolution within the interval of $[1\ mm, 10\ mm]$ with a step size of $1\ mm$, recording the pose estimation error and computation time at different resolutions.

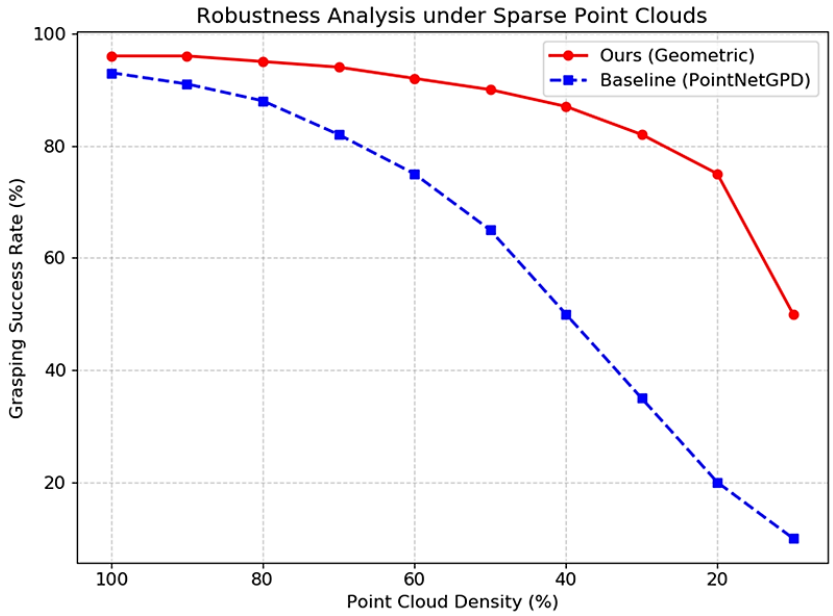

Figure 13. Dual-axis line chart for resolution sensitivity analysis.

The experimental results are shown in Figure 13. A significant nonlinear trend can be observed from the figure. In terms of computation time (blue solid line), as the resolution value increases (i.e., the grid becomes coarser), the computation time exhibits an exponential downward trend. When $\delta = 1\ mm$, the massive image matrix results in a computation time approaching 50 ms; when $\delta \geq 4\ mm$, the time rapidly converges to below 15 ms. Regarding pose estimation error (red dashed line), the error increases in an approximately linear manner as the grid becomes coarser. Notably, the two curves form a distinct "optimal balance window" at $\delta = 5\ mm$. At this resolution, the system's average pose error is controlled at approximately 2 mm, which fully satisfies the flexible tolerance range of the pneumatic soft gripper (typically 5–10 mm), while the computation time is merely 15 ms. Based on the above analysis, in order to maximize detection speed under the premise of ensuring industrial-level grasping accuracy, the BEV grid resolution is fixed at $\delta = 5\ mm$ in all subsequent experiments.

4.5.2. Computational efficiency and time complexity decomposition

To provide an in-depth revelation of the intrinsic mechanism behind the "lightweight" acceleration achieved by the proposed method, we decomposed the total time consumption of a single grasping task into stages and conducted a detailed comparison with the baseline method (SSD + PointNetGPD). Since the baseline method relies on high-dimensional 3D convolution operations, its evaluation was conducted on a workstation equipped with an NVIDIA RTX 3060 GPU. In contrast, to verify its edge deployment capability, the proposed method was executed solely in an environment powered by an Intel i7 low-voltage CPU without GPU acceleration. Early research by Konishi et al. [36] also pointed out that achieving real-time 6D pose estimation on a CPU is entirely feasible through the optimization of geometric algorithms.

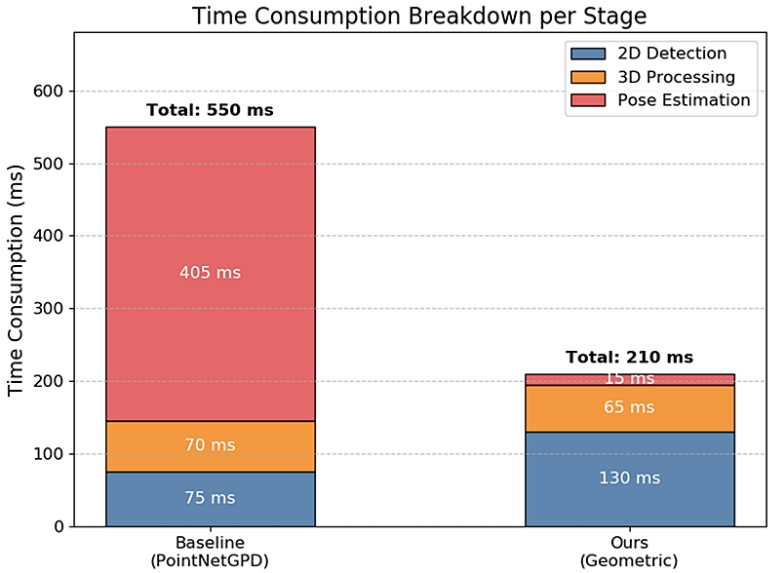

Figure 14. Stacked bar chart of time consumption breakdown.

The statistical results are shown in Figure 14. By comparing the time consumption distribution of the two methods, we can derive the following key conclusions:

Front-end detection phase (blue legend): The inference latency of YOLOv8 on the CPU is approximately 130 ms, which is slightly higher than the 75 ms of the baseline SSD method running on the GPU. This indicates that the processing pressure of the front-end visual stream remains substantial in a pure CPU environment.

Back-end pose estimation phase (red/orange legend): This constitutes the core source of performance disparity. The baseline PointNetGPD module requires complex sampling, grouping, and multilayer perceptron (MLP) inference on the point cloud, with a time consumption as high as 405 ms, accounting for 73% of the total process. In contrast, the geometric analysis method (*ours*) proposed in this paper reduces the computational complexity of this stage from $O\left(N^2\right)$ to a linear level of $O\left(N\right)$, requiring only approximately 15 ms to complete the entire process from projection to image moment calculation.

Overall performance: Although the front-end detection is slightly slower, leveraging the efficiency gains of the back-end algorithm, the total system latency of the proposed method is drastically reduced from 550 to 210 ms, achieving an overall speedup of 2.6 times. This result compellingly validates the immense advantage of the "geometric dimensionality reduction" strategy in computational resource-constrained scenarios.

4.5.3. Robustness testing in sparse point cloud environments

In real-world industrial production environments, constrained by factors such as the measurement distance of depth cameras, object surface materials (e.g., reflective or light-absorbing properties), and environmental dust, the acquired point cloud data often exhibits varying degrees of incompleteness or sparsification. To validate the robustness of the algorithm under low-quality data conditions, we designed a set of stress tests: by employing random downsampling, we artificially reduced the retention rate of the input point cloud from 100% gradually down to 10% and recorded the grasping success rates of both algorithms under different point cloud densities.

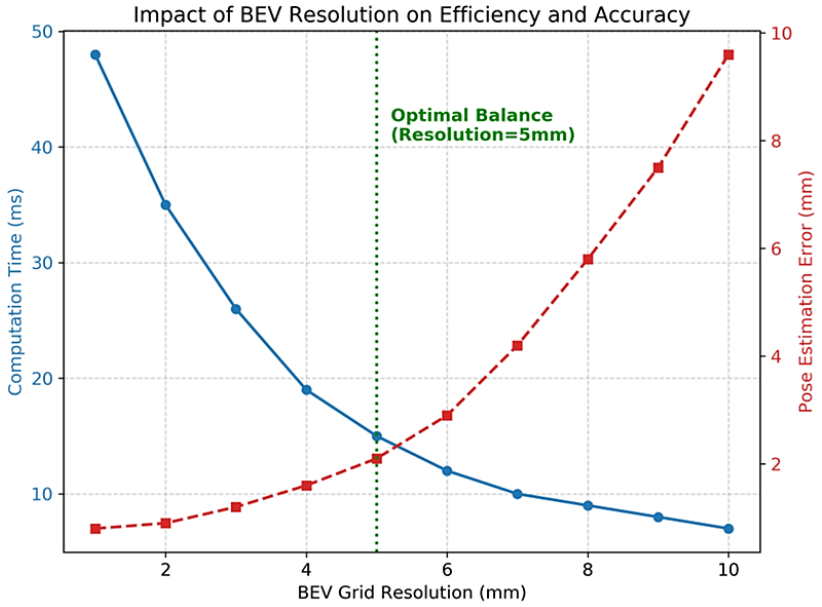

Figure 15. Red-blue line chart for robustness analysis.

The experimental results are shown in Figure 15, where the trends of the two curves reveal distinctly different anti-interference mechanisms between the two methods. The baseline method (blue dashed line) exhibits high sensitivity to point cloud density. When the point cloud retention rate drops below 60%, its grasping success rate shows a precipitous decline. This is because deep neural networks rely heavily on local geometric features (such as normal vectors and curvature) for feature extraction, and sparse point clouds disrupt these microstructures, leading to network failure. In contrast, the proposed method (red solid line) demonstrates exceptional stability. Even under the extreme condition where the point cloud retention rate is only 20%, the decline of the curve remains gradual, and the success rate is maintained above 80%. The reason lies in the fact that image moments are integral statistical quantities based on global contours, possessing inherent noise resistance. As long as the approximate contour of the object exists, RANSAC segmentation and morphological closing operations can effectively repair the voids, thereby ensuring the accuracy of centroid and principal axis calculations. This experiment proves that in industrial scenarios with unstructured data and unstable quality, the method based on explicit geometric reasoning possesses stronger engineering practicality and reliability than implicit feature learning methods.

5. Conclusions and future work

This paper addresses the challenges of real-time performance and robustness faced by existing 3D point cloud-based manipulator grasping methods by proposing and implementing a novel visual grasping framework that combines 2D images with 3D point cloud BEV negative space analysis. The research work first focused on front-end 2D object detection, verifying through experiments the immense advantage of the YOLOv8 model over the traditional SSD model in terms of detection accuracy and speed, laying a solid foundation for the stable operation of the entire system. The core innovation of this paper lies in the proposal of a 6D pose estimation algorithm based on BEV negative space. This algorithm ingeniously transforms the complex, point cloud quality-sensitive 3D pose estimation problem into a more robust and efficient 2D geometric analysis problem,

significantly reducing computational complexity. End-to-end physical manipulator grasping results show that the proposed method achieves an overall system success rate of 70%, with an increase of 5 percentage points compared to the baseline method using SSD and PointNetGPD. More importantly, the average grasping time of the system is drastically reduced from 550 to 210 ms, improving the response speed by more than 2.6 times. The experiments compellingly demonstrate that for model-free grasping tasks, the ability to generate a robust and stable grasping point is far more critical than pursuing extreme positioning accuracy of the object's geometric center. The technical route of this paper sacrifices a portion of absolute positioning precision in exchange for extremely high computational speed and high generalization capability regarding object shapes, validating the advancement and practical value of this method in developing high-speed, robust manipulator grasping systems.

Despite the system's excellent performance in single-target and simple stacking scenarios, certain applicability limitations remain under highly complex and cluttered environments. First, the BEV projection inherently oversimplifies the 3D scene, leading to the potential loss of relative 3D height details among adjacent objects. Consequently, in unstructured industrial scenarios with severe partial occlusion or highly irregular object shapes, the front-end YOLOv8 may struggle to detect targets accurately. Furthermore, severe occlusion in the BEV perspective can cause the projected negative space contours to merge or misalign, which inevitably introduces deviations in the centroid and orientation calculated by image moments, thereby affecting the final pose estimation accuracy. Second, since RGB-D cameras are primarily based on structured light or ToF principles, their ability to acquire depth information for transparent objects (such as glass bottles) or highly reflective metal surfaces is weak, potentially leading to frustum construction failure or missing projections.

Addressing the aforementioned limitations remains a priority for our future research. First, to handle multi-object cluttered scenes, we intend to integrate multi-view fusion technology. By fusing multi-frame BEV images collected by the manipulator arm from different angles, the system can compensate for the information loss inherent in a single projection and complete the geometric information of occluded objects. Second, for dynamic object grasping, we plan to incorporate a temporal tracking filtering mechanism (such as a Kalman filter or lightweight Transformer-based tracking) to predict object motion trajectories in real-time. Third, we will explore the integration of tactile sensor feedback to provide contact perception capabilities for the end-effector in extreme cases where visual perception fails, thereby further enhancing the closed-loop grasping success rate. Furthermore, considering the complexity of industrial scenarios, we will refer to the work of Liao Y [37] and Cavelli [38] on mobile manipulators to extend the proposed algorithm to mobile base platforms. Simultaneously, drawing on the multi-objective optimization and hybrid manipulation strategies of Gao Z [39], Chen H [40], and Xie Y [41], we aim to further improve the flexibility of the algorithm in dynamic environments.

Author contributions

Baoju Wu supervised the project and provided experimental resources. Nanmu Hui guided the methodology and reviewed/edited the manuscript. Yancheng Li performed the experiments, analyzed the data, and wrote the original draft. Xiaowei Han participated in the validation of the results. All authors have read and agreed to the published version of the manuscript.

Use of Generative-AI tools declaration

The authors declare that Artificial Intelligence (AI) tools were used for English language editing, translation, and proofreading in the creation of this article. The authors take full responsibility for the content of the publication.

Acknowledgments

The authors would like to thank the School of Intelligent Science and Information Engineering at Shenyang University for providing the experimental environment and equipment support.

Conflict of interest

The authors declare no conflict of interest.

References

1. Chu FJ, Xu R, Vela PA (2018) Real-world multiobject, multi grasp detection. *IEEE Robotic Autom Lett* 3: 3355–3362. https://doi.org/10.1109/LRA.2018.2852777
2. Ribeiro EG, de Queiroz R, Grassi J (2021) Real-time deep learning approach to visual servo control and grasp detection for autonomous manipulator manipulation. *Robot Auton Syst* 139: 103757. https://doi.org/10.1016/j.manipulator.2021.103757
3. Jiang Y, Fang Y, Deng L (2025) PDCNet: A lightweight and efficient manipulator grasp detection framework via partial convolution and knowledge distillation. *Comput Vis Image Und* 259: 104441. https://doi.org/10.1016/j.cviu.2025.104441
4. Yang M, Li H (2025) GMatch: A lightweight, geometry-constrained keypoint matcher for zero-shot 6DoF pose estimation in manipulator grasp tasks. *arXiv preprint arXiv:2505.16144*.
5. Guo C, Zhu C, Liu Y, Huang R, Cao B, Zhu Q, et al. (2024) End-to-end lightweight transformer-based neural network for grasp detection towards fruit manipulator handling. *Comput Electron Agr* 221: 109014. https://doi.org/10.1016/j.compag.2024.109014
6. Xu Z, Xue J, Song Z, Jia R, Lu W (2025) Lightweight network research for manipulator visual grasp for deep space exploration. *Neural Comput Appl* 37: 17083–17109. https://doi.org/10.1007/s00521-025-11377-1
7. Yang L, Bai Y, Wang Y, Alsarraj I, Kutyniok G, Wang Z, et al. (2026) Lightweight learning from actuation-space demonstrations via flow matching for whole-body soft manipulator grasping. *IEEE Robotic Autom Lett* 11: 6720–6727.
8. Nguyen N, Vu MN, Huang B, Vuong A, Le N, Vo T, et al. (2024) Lightweight language-driven grasp detection using conditional consistency model. *IEEE/RSJ International Conference on Intelligent Manipulators and Systems (IROS)*, 13719–13725. https://doi.org/10.1109/IROS58592.2024.10802007
9. Fang HS, Wang C, Gou M, Lu C (2020) GraspNet-1Billion: A large-scale benchmark for general object grasping. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 11444–11453. https://doi.org/10.1109/CVPR42600.2020.01146

10. Wang C, Mart ń-Mart ń R, Xu D, Lv J, Lu C, Fei-Fei L, et al. (2020) 6-PACK: Category-level 6D pose tracker with anchor-based keypoints. *IEEE International Conference on Robotics and Automation (ICRA)*, 10059–10066. https://doi.org/10.1109/ICRA40945.2020.9196643
11. Farhadi A, Mirzarezaee M, Sharifi A, Teshnehlab M (2024) Domain adaptation in reinforcement learning: a comprehensive and systematic study. *Front Inform Tech Electr Eng* 25: 1446–1465. https://doi.org/10.1631/FITEE.2300668
12. Pan Y, Zhang T, Li R (2025) Object dynamic recognition and grasping location via lightweight semantic attention network with learnable boundary vectors. *Measurement* 258: 119386. https://doi.org/10.1016/j.measurement.2025.119386
13. Wang S, Fei S (2019) Research and improvement of SSD (Single Shot MultiBox Detector) target detection algorithm. *Industrial Control Computer* 32: 103–105.
14. Ten PA, Gualtieri M, Saenko K (2017) Grasp pose detection in point clouds. *The International Journal of Robotics Research* 36: 1455–1473. https://doi.org/10.1177/0278364917735594
15. Qi CR, Su H, Mo K (2017) PointNet: Deep learning on point sets for 3D classification and segmentation. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 652–660. https://doi.org/10.1109/CVPR.2017.70
16. Liang H, Ma X, Li S (2019) PointNetGPD: Detecting grasp configurations from point sets. *IEEE International Conference on Robotics and Automation (ICRA)*, 3629–3635. https://doi.org/10.1109/ICRA.2019.8794435
17. Zhang Q, Zhang L, Dai C, Huang H, Liu L, Guo J, et al. (2023) RTFT6D: A real-time 6D pose estimation with fusion transformer. *International Conference on Autonomous Unmanned Systems*, 430–440. https://doi.org/10.1007/978-981-97-1099-7_41
18. Chai Z, Liu C, Xiong Z (2023) Multi-pyramid-based hierarchical template matching for 6D pose estimation in industrial grasping task. *Ind Robot* 50: 659–672. https://doi.org/10.1108/IR-08-2022-0220
19. Zhang H, Tan J, Zhao C, Liang Z, Liu L, Zhong H, et al. (2020) A fast detection and grasping method for mobile manipulator based on improved Faster R-CNN. *Ind Robot* 47: 167–175. https://doi.org/10.1108/IR-07-2019-0150
20. Yu JY, Huang D, Gao J, Li W (2023) Grasping perception method of space manipulator for complex scene task. In *Third International Conference on Machine Learning and Computer Application (ICMLCA 2022)* 12636: 930–940. https://doi.org/10.1117/12.2675288
21. Boulch A (2020) ConvPoint: Continuous convolutions for point cloud processing. *Comput Graph* 88: 24–34. https://doi.org/10.1016/j.cag.2020.02.005
22. Zhou Y, Tuzel O (2018) VoxelNet: End-to-end learning for point cloud based 3D object detection. *Proceedings of the IEEE Conference on ComputerVision and Pattern Recognition (CVPR)*, 4490–4499. https://doi.org/10.1109/CVPR.2018.00472
23. Shi SS, Wang XG, Li HS (2019) PointRCNN: 3D object proposal generation and detection from point cloud. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 770–779. https://doi.org/10.1109/CVPR.2019.00086
24. Hui NM, Wu XH, Han XW, Wu BJ (2024) A robotic arm visual grasp detection algorithm combining 2D images and 3D point clouds. *Appl Mech Mater* 919: 209–223. https://doi.org/10.4028/p-vnDoN1
25. Zhang Y, Xiang Z, Qiao C, Chen S (2020) High precision real-time target detection based on 3D point cloud bird's eye view. *Manipulator* 42: 148–156. https://doi.org/10.13973/j.cnki.manipulator.190236

26. Liu Z, Luo J, Pan Z (2019) Mid-view projection processing based on radar point cloud. *Information Technology and Network Security* 38: 40–44.
27. Guo Y, Wang H, Gao X, Wang H, Wang Y (2026) Survey of BEV 3D object detection algorithm system. *Journal of Computer Applications* 46: 1238–1252.
28. Lian QY, Zheng SW, Tu XK, Li WH (2025) Voxel feature attention-based point cloud object dection algorithm for traffic cone. *Journal of Mechanical Engineering* 61: 239–249.
29. Chen X, Han L, Xiao Y, Xue B, Ma L (2025) 3D object detection of point cloud based on voxel-keypoint feature aggregation network. *Laser Technology* 50: 291–299.
30. Xu K, Li W (2024) End-to-end multi-task 3D object detection method based on bird's eye view images. *Computer Simulation* 41: 176–181.
31. Zhang T, Xiao Z, Zou YB (2022) Workpiece recognition and pose estimation based on 3D point cloud features. *Journal of Machinery Design & Manufacturing*, 252–256. https://doi.org/10.3969/j.issn.1001-3997.2022.02.054
32. Wu J, Fang HG, Yang GX (2022) 6D pose estimation and manipulator arm grasping based on minimum size point model. *Computer Integrated Manufacturing Systems* 28: 2472–2480. https://doi.org/10.13196/j.cims.2022.08.018
33. Zhong Y, Zhang J, Zhang H (2022) Manipulator hand-eye calibration method based on target detection. *Journal of Computer Engineering* 48: 100–106. https://doi.org/10.19678/j.issn.1000-3428.0060670
34. 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: 381–395. https://doi.org/10.1145/358669.358692
35. Jocher G, Chaurasia A, Qiu J (2023) YOLO by Ultralytics. *arXiv preprint arXiv:2309.13353*.
36. Konishi Y, Hattori K, Hashimoto M (2019) Real-time 6D object pose estimation on CPU. *IEEE/RSJ International Conference on Intelligent Manipulators and Systems (IROS)*, 3451–3458. https://doi.org/10.1109/IROS40897.2019.8967967
37. Liao Y, Kang S, Li J, Liu Y, Liu Y, et al. (2024) Mobile-Seed: Joint semantic segmentation and boundary detection for mobile manipulators. *IEEE Robot Autom Lett* 9: 3902–3909. https://doi.org/10.1109/LRA.2024.3373235
38. Cavelli RF, Cheng PDC, Indri M (2024) Motion planning and safe object handling for a low-resource mobile manipulator as human assistant. *IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, 1–8. https://doi.org/10.1109/ETFA61755.2024.10711157
39. Gao Z, Li C, Ma D, Chong NY (2024) Object re-orientation via two-edge-contact pushing along a circular path based on friction estimation. *Eighth IEEE International Conference on Manipulator Computing (IRC)*, 17–23. https://doi.org/10.1109/IRC63610.2024.00009
40. Chen H, Quan F, Fang L, Zhang S (2019) Aerial grasping with a lightweightmanipulator based on multi-objective optimization and visual compensation. *Sensors* 19: 4253. https://doi.org/10.3390/s19194253
41. Xie Y, Liu J, Yang Y (2024) Pose optimization for mobile manipulator grasping based on hybrid manipulability. *Ind Robot* 51: 134–147. https://doi.org/10.1108/IR-06-2023-0128