Robotica (2026), **44**, pp. 252–274
doi:10.1017/S0263574726103154

CAMBRIDGE UNIVERSITY PRESS

RESEARCH ARTICLE

Red green blue-depth multimodal learning for vision-guided grasping in Delta robots

Puxuan Ning[1], Jiangping Mei[1], Jinlu Ni[2], Peichen Liu[3], Tianlin Wei[1] and Xiaochen Su[1]

[1]Key Laboratory of Mechanism Theory and Equipment Design of Ministry of Education, Tianjin University, China
[2]State Grid Electric Power Engineering Research Institute Co., LTD., China
[3]Charles V Schaefer Jr School of Engineering and Science, Stevens Institute of Technology, USA
Corresponding author: Puxuan Ning; E-mail: ningpuxuan@tju.edu.cn

Received: 16 April 2025; **Revised:** 19 December 2025; **Accepted:** 31 December 2025;
First published online: 13 February 2026

Keywords: visual servoing; Mask R-CNN; Delta robot

Abstract

In unstructured environments, Delta robots face challenges in achieving high vision-guided grasping precision due to dynamic lighting conditions and workpiece diversity. This paper designs an integrated solution that combines RGB-D multimodal learning with an enhanced Mask R-CNN framework. Initially, a dual-stream ResNet50-FPN backbone network is designed to achieve cross-modal adaptive alignment via hierarchical feature fusion. Subsequently, a depth-guided attention module is incorporated to bolster robustness against material ambiguity and reflective interference. Moreover, a dynamic depth estimation algorithm is employed to significantly improve target localization accuracy and stability. Finally, real-time trajectory tracking is realized by integrating PD control with Jacobian mapping. Experimental results validate the efficacy of the proposed method, offering an efficient and reliable approach for industrial robotic applications.

Abbreviations

EE	End-Effector
MMF	Multimodal Fusion
FPN	Feature Pyramid Network
RPN	Region Proposal Network
mAP	Mean Average Precision
FCN	Fully Connected Network
ROI	Regions of Interest
DGA	Depth-Guided Attention
IoU	Intersection over Union
CLS	Curriculum Learning Strategy
FW	Fixed Weights
DDE	Dynamic Depth Estimation
SD	Standard Deviation
MAE	Mean Absolute Error
FOV	Field of View
PD	Proportional-Derivative
TSR	Task Success Rate

1. Introduction

During real-world operational tasks, Delta robots often encounter visual information errors due to variations in lighting conditions and differences in workpiece materials. The accurate and real-time capture,

© The Author(s), 2026. Published by Cambridge University Press.

https://doi.org/10.1017/S0263574726103154 Published online by Cambridge University Press

processing, and feedback of visual data is a critical prerequisite for ensuring precise positioning and efficient operation. To address these challenges, researchers worldwide have conducted extensive studies on visual servoing systems based on image-based visual servoing (IBVS), pose-based visual servoing (PBVS), and deep learning-based methods.

The IBVS method constructs feedback control signals by directly extracting feature information from images, thereby adjusting robot motion in real time to minimize image errors. Zhu et al. [1] connected visual features with the robot's control inputs by utilizing principal components analysis on 2D contour subspace projections, while simultaneously generating object representations (e.g., feature vectors) and interaction models. Lagneau et al. [2] combined geometric B-spline models with sequential importance resampling particle filtering techniques, while also introducing an adaptive model-free shape servo control method, enabling dual-arm robots to manipulate linear objects without relying on mechanical properties or physical markers. Ghasemi et al. [3] proposed an innovative adaptive switching IBVS controller that decouples the rotation and translation motion of the camera, dividing the control process into three distinct phases. By employing variable gains, the controller enhances response speed and tracking accuracy. Briot et al. [4] analyzed three types of visual servo controllers: leg-direction-based visual servoing, line-based visual servoing, and image moment visual servoing. By establishing a localization error model that includes camera observation errors, they addressed the singularity issues of the controllers, thereby formulating an optimization design problem. Wang et al. [5] proposed a mobile robot visual servoing method based on virtual-goal-guided rapidly exploring random tree, utilizing a two-level framework of trajectory planning and tracking. Fried et al. [6] propose an adaptive IBVS method that does not require camera calibration nor direct measurement of image feature velocities, employing a cascaded control structure: the upper layer uses indirect adaptive motion control to handle camera-kinematic uncertainties and depth-related parameters, while the lower layer employs Slotine–Li type direct adaptive dynamic control to compensate for robot dynamic uncertainties. Simultaneously, a partial state observer is designed to estimate the necessary quantities. Wang et al. [7] utilize visual errors in real time to drive robot motion while simultaneously identifying the unknown depth from image points to the camera online. To address speed saturation limits, the controller incorporates a saturation function and compensation term in its design, ensuring the output speed operates within physical constraints, while adjusting the depth estimation through an adaptive law to eliminate model uncertainties caused by unknown depth. Under the virtual camera model, Chen et al. [8] constrain the image feature error through error transformation and a prescribed performance function. Then, during the contact interaction phase, they introduce a robust adaptive stiffness force-admittance controller that adjusts the stiffness coefficient and estimates the disturbance upper bound in real time based on image error and contact force error, generating smooth and disturbance-resistant commanded image features to ensure the stability and compliance of the contact force. At the kinematic level, Wu et al. [9] employ an iterative linear quadratic regulator within a predictive control framework to design a velocity controller and generate reference velocities. At the dynamic level, they introduce a Gaussian process-based (GP) data-driven predictive controller to estimate unknown dynamics and design a torque control law, while also providing an upper bound on the GP prediction error. Kamath et al. [10] construct an enhanced set of states by coupling the kinematic relationships of camera image features with the dynamics model of the aircraft, including image features, their derivatives, and aircraft dynamic variables. In the controller design, unknown depth-related quantities are handled through system augmentation and observers, resulting in a control law that relies solely on image measurements without the need for direct depth measurements. The aforementioned methods rely on local or simplified IBVS models, as well as manually set intermediate targets and switching conditions. This leads to challenges in ensuring global convergence and real-time performance, while also exhibiting high sensitivity to online data acquisition, sensor accuracy, and environmental noise. Consequently, this limits their robustness and control precision in complex scenarios.

The PBVS method is a technique that utilizes three-dimensional pose data obtained from a camera to achieve precise control of a robot's spatial position and orientation through geometric reconstruction. Daraviña et al. [11] used the Eye-to-Hand configuration in a 3-RRR parallel manipulator, optimized the sensor orientation, and combined PBVS and Kalman filtering to predict the trajectory of the object

https://doi.org/10.1017/S0263574726103154 Published online by Cambridge University Press

based on the initial parameters when the object was occluded, achieving continuous and precise tracking. Xie et al. [12] combined PBVS and adaptive sliding mode control to enhance the trajectory tracking control of a 6-RSS parallel robot. They utilized the C-Track 780 optical measurement sensor to measure the pose of the robot end-effector (EE) in real time. The accuracy of pose estimation was improved through an adaptive Kalman filter, and a highly robust sliding mode controller was designed to address system uncertainties. Additionally, a radial basis function neural network was introduced for the automatic adjustment of control gains. Silva et al. [13] combined PBVS with strain-based feedback control to enhance the vibration suppression capabilities of parallel manipulators equipped with flexible links. The sliding mode control strain feedback loop effectively reduces overshoot and position errors during operation, thereby significantly improving trajectory tracking performance. In conjunction with the requirements for precise positioning and trajectory tracking of parallel robots using the PBVS method, the aforementioned research adopted adaptive or hybrid control strategies to address complex motions. However, due to insufficient data sampling and inadequate noise handling strategies, inconsistencies arose in the estimation of key parameters (such as the $\boldsymbol{Q}$ matrix) in inverse kinematics. This issue is analogous to the error accumulation caused by an excessive reliance on local features in PBVS, ultimately affecting the stability and efficiency of the PBVS control framework in industrial applications.

The deep learning-based visual servoing method is a technique that automatically extracts and learns complex image features by constructing and training neural network models, enabling intelligent prediction and control of target states in dynamic environments. Ribeiro et al. [14] utilized the Cornell Grasping Dataset to employ a CNN for simultaneous recognition of object pose and the selection of stable grasping contact points. Al-Shanoon and Lang [15] proposed a framework called deep visual servoing, which utilizes a monocular RGB camera to overcome the limitations of traditional methods in real-world scenarios, such as lighting and occlusion. Han et al. [16] proposed a spectral adaptive law for online neural network training by replacing the traditional F-norm term in the Lyapunov function with a spectral norm regularization term. This approach enhances the robustness of closed-loop systems, reduces sensitivity to disturbances, and ensures that the network weights remain bounded through a mapping algorithm. Additionally, they utilized a neural network estimator for the online identification of the system's composite Jacobian matrix, which not only avoids the time-consuming process of camera calibration but also allows for the design of a dynamic visual servo controller based on the identification results. Jiang et al. [17] proposed a deep learning method based on depth images for vision-guided robotic box picking, targeting efficient grasping of textureless planar objects. By utilizing a deep convolutional neural network to directly predict grasping points, the complex object segmentation process is avoided. Yu et al. [18] proposed a robot grasping framework based on the transformer model, designed for the safe handling of deformable objects such as fruits. By integrating visual and tactile information, the framework overcomes the limitations of traditional methods in dynamic contact interactions. Utilizing two exploratory actions (pinching and sliding), the model effectively extracts the physical characteristics of objects and employs deep learning techniques to predict safe grasping forces. Odira et al. [19] adopt a two-stage training approach that first uses data generated by a physical model for pretraining, followed by fine-tuning with a small amount of measured data to improve path tracking accuracy and robustness. This method does not rely on image velocity measurements, cascade observers, or adaptive laws, and it is proven to ensure global stability and error convergence even in the presence of unmodeled dynamics. Hong et al. [20] proposed combining artificial neural networks with nonlinear model predictive control for visual path tracking of mobile robots, and employed a two-stage training strategy to enhance the generalization capability of the data-driven model. Kamath et al. [21] proposed a method that combines physics-informed neural networks (PINN) with dynamics-centered visual servoing: by learning an enhanced dynamics model through PINN, the changes in image pixels are directly mapped to the thrust and torque inputs of a multirotor. Simultaneously, PINN is used to model uncertainties in camera and vehicle parameters as well as dynamics modeling errors, thereby merging image noise and system disturbances into a unified disturbance term to improve robustness. The generalization ability of deep learning methods for robotic grasping and dynamic manipulation is crucial. However,

https://doi.org/10.1017/S0263574726103154 Published online by Cambridge University Press

relying solely on limited RGB or synthetic data, while neglecting depth information, object segmentation, and tuning for control gains, noise, and adaptability to complex objects, can increase grasping errors, energy consumption, and reduce positioning accuracy. Additionally, offline training and high computational complexity result in slow system responses and insufficient robustness, thereby limiting its industrial applications.

This paper considers the material differences of workpieces in unstructured environments based on Delta robot conditions and designs an improved Mask R-CNN instance segmentation model for segmenting target workpieces and the EE. By integrating a dynamic depth estimation algorithm, it enables precise and rapid object localization. The method is suitable for unstructured environments, which is crucial for the generalization of robotic systems. Meanwhile, it reduces localization errors while ensuring accuracy. Additionally, it enables real-time feedback processing, meeting the requirements of high-speed parallel robotic systems.

The contributions of our work are summarized as follows.

1. We design a hierarchical multimodal fusion module built on a dual-stream ResNet50–FPN backbone that aligns multi-scale RGB and depth features, improving mAP@0.5 from 84.7 to 95.2% while increasing parameters by only approximately 5.0%.
2. We introduce a depth-guided attention module that cross-modally reweights RGB-depth features spatially and channel-wise, boosting instance segmentation IoU from 0.68 to 0.81 while keeping mask inference latency at roughly 8.3 ms.
3. We propose a dynamic depth estimation method that uses noise–distance adaptive grids, within-grid median filtering, and inverse variance weighted median fusion, reducing depth MAE from 12.3 to 5.7 mm. Integrated with PD + Jacobian control, the full pipeline operates at roughly 40 Hz.

2. Multimodal learning Mask R-CNN

2.1. RGB-D multimodal fusion module

This network utilizes RGB-D multimodal fusion module (MMF) input to achieve heterogeneous feature extraction through parallel data streams. The RGB image stream receives 8-bit color data at a resolution of $1{,}280 \times 720 \times 3$, processed through pixel-level normalization, as shown in Eq. (1):

$$\boldsymbol{I}_{\text{norm}} = \frac{\boldsymbol{I}_{\text{raw}}}{255} \tag{1}$$

where $\boldsymbol{I}_{\text{norm}}$ represents the normalized image, while $\boldsymbol{I}_{\text{raw}}$ is the original image, mapped to the range of [0,1]. The depth image stream processes $1{,}280 \times 720$ depth data in Z16 format, with the depth value intercepted in the range of [0, 2000] and then normalized, as shown in Eq. (2):

$$\boldsymbol{D}_{\text{norm}} = 2 \cdot \frac{\boldsymbol{D}_{\text{raw}}}{2000} - 1 \tag{2}$$

where $\boldsymbol{D}_{\text{norm}}$ represents the normalized depth image, and $\boldsymbol{D}_{\text{raw}}$ is the original depth image. This generates a normalized tensor aligned with the RGB modality, which is then normalized to the range of $[-1, 1]$ using min–max scaling. This preprocessing strategy effectively eliminates noise from the depth sensor while preserving millimeter-level geometric accuracy.

The network has designed a hierarchical feature fusion mechanism, as depicted in Figure 1. Both streams utilize an improved ResNet50-FPN backbone network with shared parameters, which includes:

1. Shallow feature sharing: Weight sharing between Conv1-Conv3 layers to extract common edge and texture features.
2. Deep modality differentiation: Conv4-Conv5 layers differentiate into dedicated paths for RGB and depth, respectively, extracting color semantics and geometric structure features.

https://doi.org/10.1017/S0263574726103154 Published online by Cambridge University Press

Table I. *Ablation study on MMF module (mAP@0.5).*

Fusion strategy	mAP@0.5 (%)	Inference time (ms)	Parameter number (M)
Baseline (RGB)	84.7	38.2	44.1
Late fusion	87.1	41.5	45.8
MMF (ours)	92.3	39.1	46.3

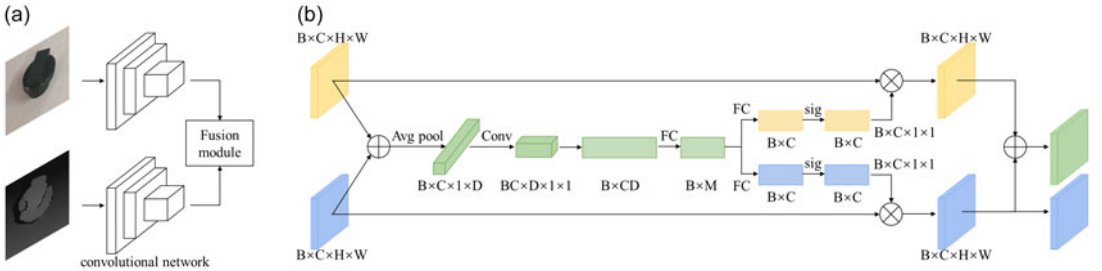


Figure 1. *MMF input preprocessing, (a) hierarchical feature fusion, (b) structure of the MMF module.*

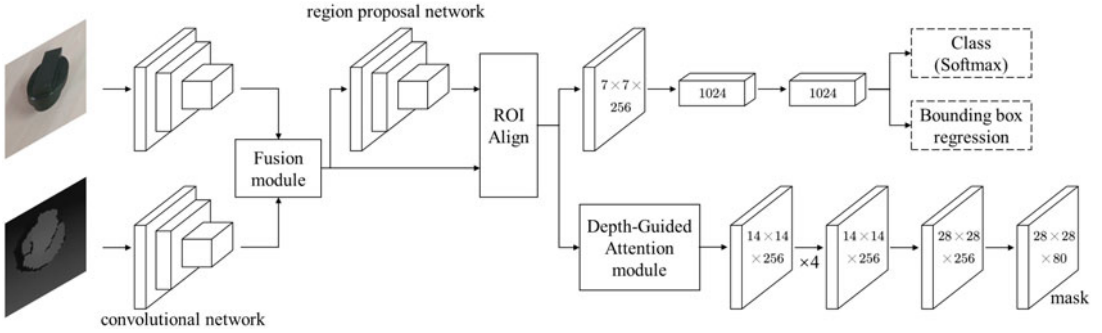


Figure 2. *Mask R-CNN network architecture based on RGB-D multimodal learning.*

3. Multi-scale fusion: At the P3-P5 levels of the feature pyramid network (FPN) output, feature fusion is achieved through adaptive channel alignment and element-wise summation, as follows in Eq. (3):

$$\boldsymbol{F}^{i}_{\text{fusion}} = \text{Conv}_{1\times1}\left(\boldsymbol{F}^{i}_{\text{color}}\right) + \text{Conv}_{1\times1}\left(\boldsymbol{F}^{i}_{\text{depth}}\right) \ (i \in \{3, 4, 5\}) \tag{3}$$

where $\boldsymbol{F}^{i}_{\text{fusion}}$ represents the fused feature map of the layer i, $\boldsymbol{F}^{i}_{\text{color}}$ denotes the feature map extracted from the RGB modality at the layer i, and $\boldsymbol{F}^{i}_{\text{depth}}$ denotes the feature map extracted from the depth modality at the layer i.

The 1×1 convolutional layer is used to unify the channel dimension ($256\rightarrow256$), and the fused feature tensor is input into the subsequent region proposal network (RPN). This design significantly enhances the representation capability for occluded and reflective objects while reducing the number of parameters, as detailed in Table I.

This layered fusion strategy retains spatial details at lower levels while integrating semantic context at higher levels. The output size of the fusion module is $14\times14\times256$, and classification and bounding box regression are performed subsequently through the fully connected network (FCN). Additionally, a Softmax layer is used for the final class prediction.

The network retains the architecture of Mask R-CNN and has made key improvements for robotic grasping scenarios, as shown in Figure 2: the RPN uses five anchor scales of {32, 64, 128, 256, 512} and three aspect ratios of 0.5, 1, and 2 to optimize the prior box distribution for robotic tasks; the regions

https://doi.org/10.1017/S0263574726103154 Published online by Cambridge University Press

Table II. *Coordinate alignment error experiment.*

Expected ROI center	Extracted ROI center	X Direction error	Y Direction error	Euclidean error
(100.0, 150.0)	(100.2, 150.1)	0.20 pixel	0.10 pixel	0.22 pixel
(200.0, 250.0)	(199.8, 250.3)	0.20 pixel	0.30 pixel	0.36 pixel
(300.0, 350.0)	(300.4, 349.8)	0.40 pixel	0.20 pixel	0.45 pixel
(400.0, 450.0)	(400.1, 449.6)	0.10 pixel	0.40 pixel	0.41 pixel
(500.0, 550.0)	(500.3, 550.2)	0.30 pixel	0.20 pixel	0.36 pixel

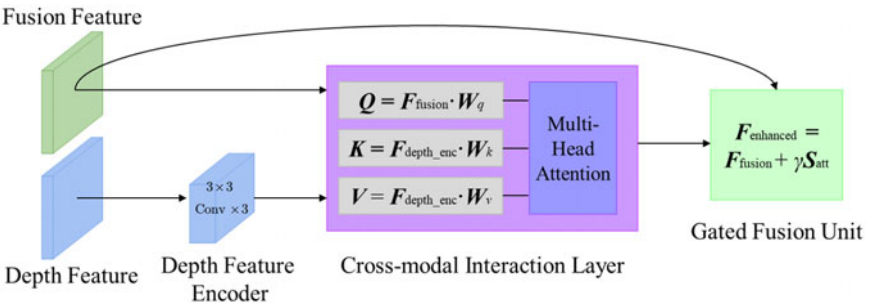


Figure 3. *DGA module.*

of interest (ROI) align layer extracts 7×7 region features from the fused feature map, with a coordinate alignment error of less than 0.5 pixel, as detailed in Table II; the dual-task head consists of a classification-box regression head (Faster R-CNN) and a mask prediction head, which operate in parallel, with the latter incorporating a depth-guided attention mechanism, as discussed in Section 2.2.

The total number of parameters in the network is 46.3 million, which is an increase of only 4.8% compared to the baseline model.

2.2. Depth-guided attention module

In the Delta robot grasping task, the material of the target objects (such as metal and glass) and complex scenes (such as occlusion and reflection) pose significant challenges to mask prediction accuracy. The mask head of the traditional Mask R-CNN only relies on RGB features, which makes it difficult to effectively distinguish objects with similar geometry but different materials. To address this, this paper designs the depth-guided attention mechanism (DGA), which dynamically integrates information from the depth modality to enhance the robustness of mask predictions. The core idea can be expressed as Eq. (4):

$$\boldsymbol{A}_{\text{att}} = \sigma\left(\boldsymbol{W}_a \cdot \left[\boldsymbol{F}_{\text{fusion}}; \boldsymbol{F}_{\text{depth}}\right]\right) \odot \boldsymbol{F}_{\text{fusion}} \tag{4}$$

where $\boldsymbol{A}_{\text{att}}$ represents the attention weight matrix, σ denotes the Sigmoid activation function, $\boldsymbol{W}_a$ is the learnable weight parameter, $[\cdot\,;\cdot]$ indicates channel concatenation, and $\odot$ signifies element-wise multiplication. This mechanism adaptively weights the fused features in both spatial and channel dimensions using deep features.

The DGA module consists of a deep feature encoder, a cross-modal interaction layer, and a gated fusion unit submodule, as shown in Figure 3. The deep feature encoder encodes the depth features $\boldsymbol{F}_{\text{depth}}$ output by the FPN into a tensor with the same resolution as the fused features through three layers of 3×3 convolution. The cross-modal interaction layer computes the correlation between the fused features and the depth features using cross-attention, as follows in Eqs. (5) and (6):

https://doi.org/10.1017/S0263574726103154 Published online by Cambridge University Press

Table III. *Ablation study on DGA module (IoU).*

Configuration	Metallic object	Glass object	Obscured scene
Baseline (no DGA)	0.72	0.61	0.68
Spatial attention	0.78	0.69	0.73
DGA (ours)	0.84	0.79	0.81

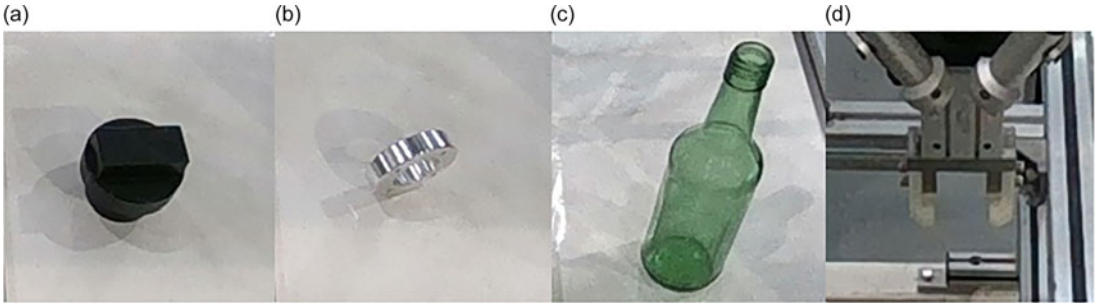


Figure 4. *Workpieces, (a) plastic workpiece, (b) metal workpiece, (c) glass workpiece, (d) EE.*

$$\boldsymbol{Q} = \boldsymbol{F}_{\text{fusion}}\boldsymbol{W}_q, \quad \boldsymbol{K} = \boldsymbol{F}_{\text{depth}}\boldsymbol{W}_k, \quad \boldsymbol{V} = \boldsymbol{F}_{\text{depth}}\boldsymbol{W}_v \tag{5}$$

$$\boldsymbol{S}_{\text{att}} = \text{Softmax}\left(\frac{\boldsymbol{Q}\boldsymbol{K}^T}{\sqrt{d_k}}\right)\boldsymbol{V} \tag{6}$$

where $\boldsymbol{W}_q$, $\boldsymbol{W}_k$, and $\boldsymbol{W}_v$ represent the learnable weight parameters for $\boldsymbol{Q}$, $\boldsymbol{K}$, and $\boldsymbol{V}$, respectively, while $\boldsymbol{S}_{\text{att}}$ denotes the attention map. The gated fusion unit further integrates the attention map $\boldsymbol{S}_{\text{att}}$ with the original fused features, using Eq. (7):

$$\boldsymbol{F}_{\text{enhanced}} = \boldsymbol{F}_{\text{fusion}} + \gamma \cdot \boldsymbol{S}_{\text{att}} \tag{7}$$

where γ is a learnable scaling factor, with an initial value of 0.1.

Ablation experiments on the synthetic dataset indicate that the DGA module significantly enhances segmentation accuracy in complex scenarios, as detailed in Table III.

To meet the real-time requirements of the Delta robot, the DGA module employs a lightweight design: channel compression reduces the number of attention channels from 256 to 64, with only a 0.8% loss in accuracy; sparse computation applies the DGA only at the P4 level of the FPN, resulting in a 22% increase in inference speed. On the Advantech EPC-P3086 industrial computer platform, the inference time for the mask prediction head that includes the DGA is 8.3 ms.

2.3. Fine tuning

To enhance the task adaptability of Mask R-CNN under multimodal inputs, this section employs a fine-tuning framework based on category reduction and domain adaptation. It reduces the 80 target categories from general scenes to 4 target categories specific to industrial scenarios, optimizing model parameters for dynamic grasping scenarios involving Delta robots. The tuning process combines the characteristics of the MMF module and the DGA module and realizes the migration of the model from general scenarios to industrial scenarios through a hierarchical parameter optimization strategy.

Using the Mask R-CNN pre-trained on the COCO dataset as the base model retains its general feature extraction capabilities. For four types of targets in industrial scenarios: PLA plastic workpieces, metal workpieces, glass workpieces, and EE, as shown in Figure 4. Remap the classification head by replacing the 80-class classification head of COCO with a 4-class classification head. The parameters of the MMF module and the DGA module are randomly initialized to avoid feature bias caused by modal differences.

https://doi.org/10.1017/S0263574726103154 Published online by Cambridge University Press

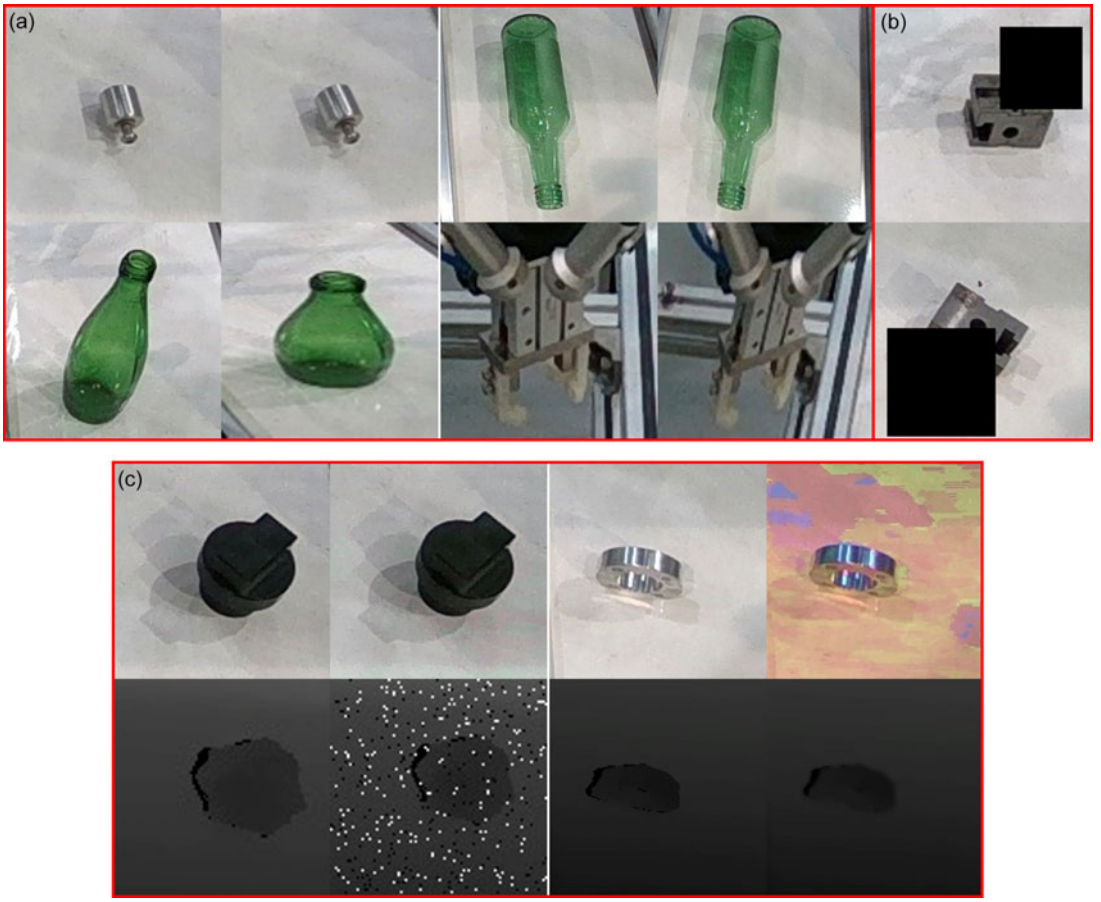


Figure 5. *Data augmentation, (a) geometric perturbation, (top) 10° counterclockwise and 15° clockwise rotation, and (bottom) affine transformation, (b) synthetic occlusion, (top) occlusion 15%, and (bottom) occlusion 30%, (c) modality-specific noise, (top left) $\sigma_G = 0.05$ $\Delta H = -0.1$, (top right) $\sigma_G = 0.05$ $\Delta S = 0.3$, (bottom left) 5% pulse noise, and (bottom right) bilateral filtering.*

Utilizing a layered parameter unfreezing mechanism to freeze the backbone network, the weights of the Conv1-Conv3 layers in ResNet50-FPN are fixed, preserving the ability to extract low-level edge and texture features. Modality-specific layer tuning is conducted, unfreezing the RGB and depth branch parameters in the Conv4-Conv5 layers to enhance the adaptability of the model to multimodal features in industrial scenarios. During the dynamic training of the DGA module, the cross-modal interaction layer and the gated fusion unit are actively involved throughout the training process, achieving adaptive fusion of geometric and semantic features through end-to-end learning.

In addition, to address variations in lighting conditions and sensor noise in industrial environments, a joint enhancement of 12,000 pairs of labeled RGB-D images was performed, as shown in Figure 5. First, geometric perturbations were applied, introducing random rotations within a $\pm$ 15° range, scaling between 0.8 and 1.2 times, and affine transformations to both the RGB and depth images. Next, modality-specific noise was added, with Gaussian noise of $\sigma_G = 0.05$ introduced to the RGB channel, along with color disturbances of $\Delta H = \pm 0.1$ and $\Delta S = \pm 0.3$; the depth channel received simulated impulse noise at a density of 5% and bilateral filtering for denoising, enhancing the robustness of the model against anomalies in depth sensors. Finally, synthetic occlusions were added, randomly inserting occlusion samples covering 15–25% of the area in 20% of the training batches to improve the attention focusing capability of the DGA module.

https://doi.org/10.1017/S0263574726103154 Published online by Cambridge University Press

Table IV. *Curriculum learning strategy.*

Training strategy	mAP@0.5 (%)	IoU	Epochs
Fixed weights	92.1	0.83	120
CLS (ours)	95.2	0.87	100

2.4. Training

For this network model, optimize the loss function and define a multi-task joint loss function, as follows in Eq. (8):

$$\mathcal{L}_{\text{total}} = \lambda_1 \mathcal{L}_{\text{cls}} + \lambda_2 \mathcal{L}_{\text{box}} + \lambda_3 \mathcal{L}_{\text{mask}} + \lambda_4 \mathcal{L}_{\text{dga}} \tag{8}$$

where $\mathcal{L}_{\text{cls}}, \mathcal{L}_{\text{box}}, \mathcal{L}_{\text{mask}}$ are the classification loss, bounding box regression loss, and mask segmentation loss, respectively, and $\mathcal{L}_{\text{dga}}$ is the attention supervision loss, as follows in Eq. (9):

$$\mathcal{L}_{\text{dga}} = ||\boldsymbol{S}_{\text{att}} - \boldsymbol{Y}_{\text{edge}}||_2 \tag{9}$$

where $\boldsymbol{Y}_{\text{edge}}$ is the object edges extracted from the depth map using the Canny operator as prior knowledge. The weight coefficients λ_1 to λ_4 adopt a curriculum learning strategy (CLS), as follows in Eq. (10):

$$\lambda_i(t) = \lambda_i^{\text{start}} + \left(\lambda_i^{\text{end}} - \lambda_i^{\text{start}}\right) \cdot \frac{t}{T} \tag{10}$$

where the initial weights are denoted as λ_i^{start} and the final weights as λ_i^{end}, with initial values of [1.0, 1.0, 0.5, 0.2]. These values are linearly adjusted over the training epochs to [0.8, 0.8, 1.0, 0.5] to balance the convergence speed of each task. The classification loss λ_1 and localization loss λ_2 decrease from 1.0 to 0.8 to reduce reliance on the foundational tasks, while the segmentation loss λ_3 increases from 0.5 to 1.0 to enhance attention to mask details. The attention supervision loss λ_4 rises from 0.2 to 0.5 to guide the model in learning geometric features in the later stages. For further details, see Table IV.

The CLS reduces the number of training iterations by 17%, as adjustments to task weights prevent early oscillations. The intersection over union (IoU) for the target object improves by 4.8%, demonstrating the effectiveness of the enhanced attention mechanism in the later stages. This approach avoids conflicts driven by tasks, allowing early high-weight classification and localization tasks to converge quickly, thereby providing reliable candidate ROI for subsequent segmentation and preventing mask predictions from failing due to localization errors. It alleviates optimization imbalances; if weights are fixed throughout (e.g., $\lambda_3 = 1.0$), complex tasks may interfere with foundational tasks due to excessively large gradient magnitudes. The phased adjustments enable a smoother transition, enhancing overall convergence. The effectiveness of the attention mechanism is strengthened, as increasing λ_4 in the later stages compels the model to refine segmentation edges using deep geometric information once the foundational tasks stabilize, such as the reflective areas of metal workpieces and the translucent areas of glass workpieces.

A custom dataset containing 12,000 pairs of RGB-D images, including workpieces such as PLA plastic, metal, and glass materials, is used for training and validation. The multimodal learning-based model in this paper achieves 95.2% mAP@0.5 and 87% IoU, exceeding the baseline model 84.7% mAP@0.5 and 73% IoU, exceeding the late feature fusion model 87.1% mAP@0.5 and the spatial attention model 80% IoU, respectively. As shown in Figures 6(a) and 7(a), the training curve demonstrates rapid convergence. The designed method achieves 90% mAP@0.5 and 80% IoU within 40 epochs. The performance comparison of the MMF module and the DGA module designed in this paper with other methods under fixed weights (FW) and CLS approaches are shown in Figures 6(b) and 7(b).

The model achieves a real-time inference speed of approximately 50 FPS on the Advantech EPC-P3086 industrial computer platform, ensuring that visual information can be processed and fed back in about 20 milliseconds.

https://doi.org/10.1017/S0263574726103154 Published online by Cambridge University Press

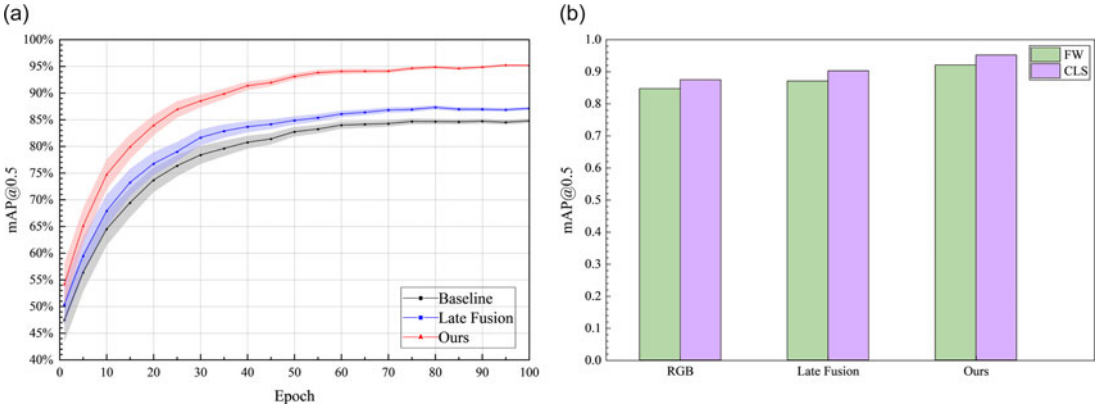


Figure 6. *The MMF module, (a) training curve, (b) performance comparison.*

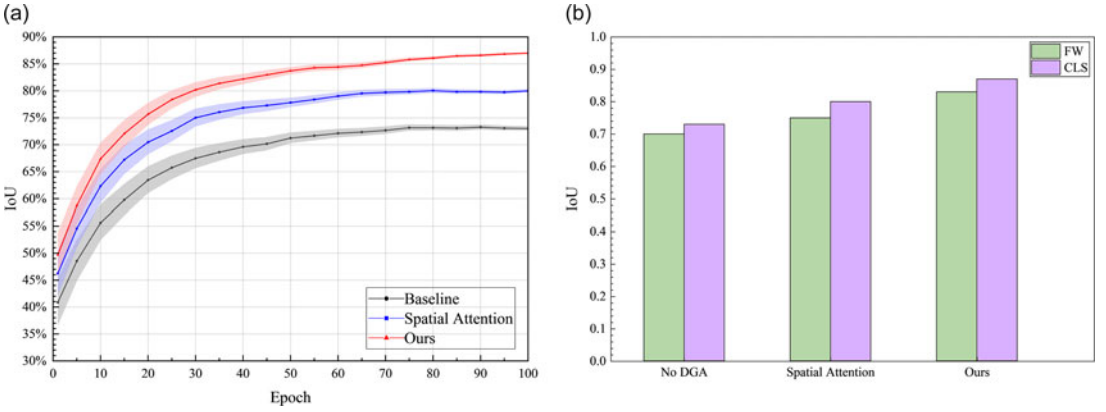


Figure 7. *The DGA module, (a) training curve, (b) performance comparison.*

(a)

1605	1609	1600	3786
1598	1612	1607	1602
1593	59	1603	1612
1613	1606	1608	1597

(b)

1374	1506	1468	1530
1418	1405	1500	1420
1512	1300	1450	1380
1340	1415	1498	1442

Figure 8. *Common issues with depth images, (a) local outlier, (b) depth fluctuation.*

3. Dynamic depth estimation

In the depth images generated by the Intel RealSense D455, there are also some common and significant issues, which can be summarized in the following two aspects:

One is that the depth image contains erroneous pixel values that cannot accurately represent depth. For example, the pixel segment in the depth image shown in Figure 8(a) corresponds to an object at a depth of 1600 mm, and the values of the erroneous pixels (marked in red) differ significantly from the actual depth of the object.

https://doi.org/10.1017/S0263574726103154 Published online by Cambridge University Press

The other is that the depth values of the same object in the depth image are different and fluctuate. For example, the pixel segment is shown in Figure 8(b). These pixels correspond to a plane object with a depth of 1,400 mm. The pixel segment shows that the pixels on the same plane fluctuate by about 150 mm.

In order to solve the above problems, the following dynamic depth estimation (DDE) algorithm is designed, see Algorithm 1 for details. The main idea of the algorithm is: first take the instance segmentation mask in the depth image and divide it into n grid cells; then calculate the local depth Z_{grid} of each grid cell to eliminate local outliers; then weight these local depths by inverse variance to obtain Z_{cand} to suppress depth fluctuations; finally, concatenate the target mask centroid (u_c, v_c) and the object depth Z_{final}, and output the target three-dimensional coordinates $\boldsymbol{P}_{3D}$ through the camera model, see Section 4 for details.

Considering that the noise intensity σ_{int} and the object distance d directly determine the statistical characteristics and reliability of depth data and are highly coupled with the physical model of the sensor, the influence of other variables (such as lighting and material) has been effectively isolated by the preliminary part of the model (MMF module, DGA module). This ensures the focus and efficiency of the depth estimation part; thus, the relationship between the two and the number of grid cells N is established.

The noise model of the depth sensor is usually Gaussian noise, and the noise intensity generally varies exponentially with the distance of the object from the sensor, as shown in Eq. (11):

$$\sigma_{int}(d) = \sigma_0 \cdot \left(\frac{d}{d_0}\right)^{\alpha} \tag{11}$$

where $\sigma_{int}(d)$ is the noise intensity at the object distance d, σ_0 is the reference noise intensity, d_0 is the reference distance, which is generally the typical working distance of the sensor, and α is the relationship exponent, usually $\alpha \approx 1$. In general, the greater the noise, the more grid cells N are needed. By increasing the number of grid cells, multiple local depth values can be used to reduce the impact of noise and improve the accuracy of depth estimation, as shown in Eq. (12):

$$N(\sigma_{int}) = \left[N_0 \cdot \left(\frac{\sigma_{int}}{\sigma_0}\right)^{\beta}\right] \tag{12}$$

where N_0 is the number of reference grid cells, β is the adjustment factor used to control the relationship between noise intensity and the number of grid cells, usually within [1, 2]. Substituting (11) into (12) can be expressed as Eq. (13):

$$N(d) = \left[N_0 \cdot \left(\frac{d}{d_0}\right)^{\alpha\beta}\right] \tag{13}$$

This relationship implies that as the distance d of an object increases, the noise σ_{int} also increases, and the number of grid cells N correspondingly increases, thereby enhancing the stability and robustness of depth estimation.

When calculating the independent depth value of each grid cell, the minimum valid pixel threshold is set, and only when the number of valid pixels in the grid $\eta \geq 5$ is involved in the calculation, the interference of sparse data is avoided and adaptive filtering is realized. Through the spatial consensus mechanism, the local depth values of the valid grid cells are weighted by inverse variance.

Let the local depth values in the effective grid cell $\boldsymbol{G}_{ij}$ be $Z_{cand} = \{z_1, z_2, \ldots, z_k\}$, with each value z_i corresponding to a variance of ${\sigma_i}^2$, as follows in Eq. (14):

$$\sigma_i^2 = \frac{1}{n_i - 1} \sum_{(u,v)\in \boldsymbol{G}_{ij}} (D(u,v) - \mu_i)^2 \tag{14}$$

https://doi.org/10.1017/S0263574726103154 Published online by Cambridge University Press

Algorithm 1. Dynamic Depth Estimation.

Input:
$\boldsymbol{M}$: Binary mask from Mask R-CNN ($H\times W$)
$\boldsymbol{D}_{raw}$: Raw depth map ($H\times W$)
$\boldsymbol{K}$: Camera intrinsic matrix
d_{max}: Maximum valid depth (e.g., 2000 mm)
d_0: Reference distance (e.g., 1000 mm)
σ_0: Reference noise intensity at d_0
α: Noise-distance exponent (default: 1)
β: Grid scaling factor (default: 1.5)
N_0: Base grid size (e.g., 8)
Output:
$\boldsymbol{P}_{3D}$: Estimated 3D centroid coordinates (X, Y, Z)
1: // Step 1: Mask-based depth extraction
2: $\boldsymbol{D}_{obj} \leftarrow \boldsymbol{D}_{raw} \odot \boldsymbol{M}$
3: $\boldsymbol{D}_{valid} \leftarrow \{D_{obj}(u,v) \mid 0 < D_{obj}(u,v) < d_{max}\}$
4: // Step 2: Estimate object distance and noise intensity
5: $d_{init} \leftarrow$ **Median** ($\boldsymbol{D}_{valid}$)
6: $\sigma_{int}(d) \leftarrow \sigma_0 \cdot (d_{init}/d_0)\text{^}\alpha$
7: // Step 3: Dynamic grid size adjustment
8: $N \leftarrow \lfloor N_0 \cdot (d_{init}/d_0)\text{^}\alpha\beta \rfloor$
9: Divide $\boldsymbol{M}$ into $N \times N$ grids
10: // Step 4: Grid-wise adaptive filtering
11: Initialize $Z_{grid}[N][N]$
12: **for** each grid $\boldsymbol{G}_{ij}$ **do**
13: $\quad \boldsymbol{D}_{patch} \leftarrow \{D_{valid}(u,v) \mid (u,v) \in \boldsymbol{G}_{ij}\}$
14: $\quad$ **if** $|\boldsymbol{D}_{patch}| > \eta$ **then** // $\eta = 5$ (minimum valid pixels)
15: $\quad\quad Z_{grid}[i][j] \leftarrow$ **Median** ($\boldsymbol{D}_{patch}$)
16: $\quad$ **else**
17: $\quad\quad Z_{grid}[i][j] \leftarrow$ **NULL**
18: $\quad$ **end if**
19: **end for**
20: // Step 5: Spatial consensus
21: $\boldsymbol{Z}_{cand} \leftarrow \{z \mid z \in Z_{grid}, z \neq \textbf{NULL}\}$
22: $Z_{final} \leftarrow$ **WeightedMedian** ($\boldsymbol{Z}_{cand}$, $\omega_i = 1/\sigma_i\text{^}2$)
23: // Step 6: 3D coordinate calculation
24: $(u_c, v_c) \leftarrow$ **Centroid** ($\boldsymbol{M}$)
25: $\boldsymbol{P}_{3D} \leftarrow \boldsymbol{K}\text{^}(-1) \cdot [u_c, v_c, 1]\text{^}T \cdot Z_{final}$
26: **return** $\boldsymbol{P}_{3D}$

where $D(u,v)$ is the depth value in the grid cell G_{ij}, n_i is the number of valid pixels in it, and μ_i is the mean of the depth value in it, as follows in Eq. (15):

$$\mu_i = \frac{1}{n_i} \sum_{(u,v)\in \boldsymbol{G}_{ij}} D(u,v) \tag{15}$$

Sort Z_{cand} by the median z_i in ascending order to get the ordered sequence $z_{(1)}, z_{(2)}, \ldots, z_{(k)}$; calculate the cumulative weight for each median, as follows in Eq. (16):

$$W_j = \sum_{i=1}^{j} \omega_{(i)} \tag{16}$$

https://doi.org/10.1017/S0263574726103154 Published online by Cambridge University Press

Table V. *Performance of regional depth estimation.*

Method	MAE (mm)	SD (mm)	Runtime (ms)
Fixed grids ($N = 8$)	12.3	28.0	2.1
Depth-image fusion	13.9	29.8	41.2
VoxDepth	10.4	25.6	37.1
Dynamic grids (ours)	5.7	10.4	3.4

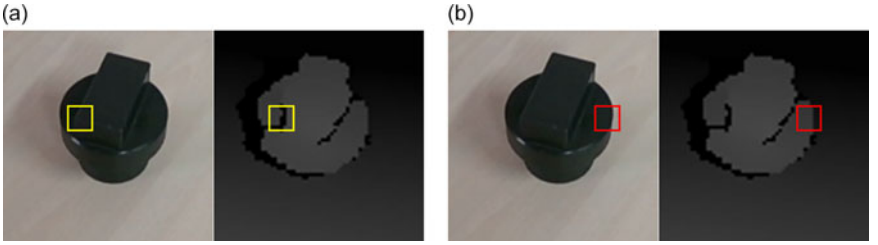


Figure 9. *Objects' measured areas, (a) uneven surface, (b) object's edge.*

where $\omega_{(i)}$ is shown in Eq. (17):

$$\omega_{(i)} = \frac{1}{\sigma_i^2} \tag{17}$$

Find the minimum j that satisfies $W_j \geq 1/2W_k$, to determine the median, then $Z_{final} = z_{(j)}$. The smaller the variance ${\sigma_i}^2$, the more stable the depth value of the grid, the larger the weight ω_i, and the greater the contribution to the final result. By weighting the median, the influence of high-variance grids that are severely affected by noise on the final depth estimation is weakened, thereby suppressing depth fluctuations. The measurement area selected is shown in Figure 9. Compared with the fixed grid strategy, Perezyabov et al. [22] Depth-image fusion method, and Chakrabarty et al. [23] VoxDepth method, the designed method reduces the depth fluctuation error by 54, 59, and 45%, respectively. For a plastic workpiece at a distance of 1,500 mm, the standard deviation (SD) of the depth estimation decreases from ±28 mm with the fixed grid, ±29.8 mm with the Depth-image fusion method, and ±25.6 mm with the VoxDepth method to ±10.4 mm, as detailed in Table V. Furthermore, testing has shown that although the LiDAR-visual-inertial SLAM system integrating visual point-line features of Hui et al. [24] and the optical flow-based stereo visual odometry of Liu et al. [25] achieve high accuracy, their processing times are 168.3 ms and 90.9 ms, respectively, which do not meet the requirements for real-time performance. Depth estimation experiments were conducted at distances of 1, 1.5, and 2 m from the workpiece, respectively. The MAE and SD statistical indicators for the four methods are shown in Figures 10 and 11.

4. Visual servo control

In order to facilitate the analysis of the motion rule of the robotic visual servo system, according to the research of Ni et al. [26], the kinematic model of the Delta robot is first established, and its mechanism is simplified to the structure shown in Figure 12. The dimensional parameters are detailed in Table VI.

As shown in Figure 12, the moving platform is simplified to a mass point O', and the coordinate system O-xyz is established at the center of the static platform. In this coordinate system, the position vector $\boldsymbol{r} = (x\ y\ z)^{\mathrm{T}}$ of the moving platform reference point O' can be expressed as Eq. (18):

$$\boldsymbol{r} = \boldsymbol{e}_i + l_1\boldsymbol{u}_i + l_2\boldsymbol{w}_i, \quad i = 1, 2, 3 \tag{18}$$

https://doi.org/10.1017/S0263574726103154 Published online by Cambridge University Press

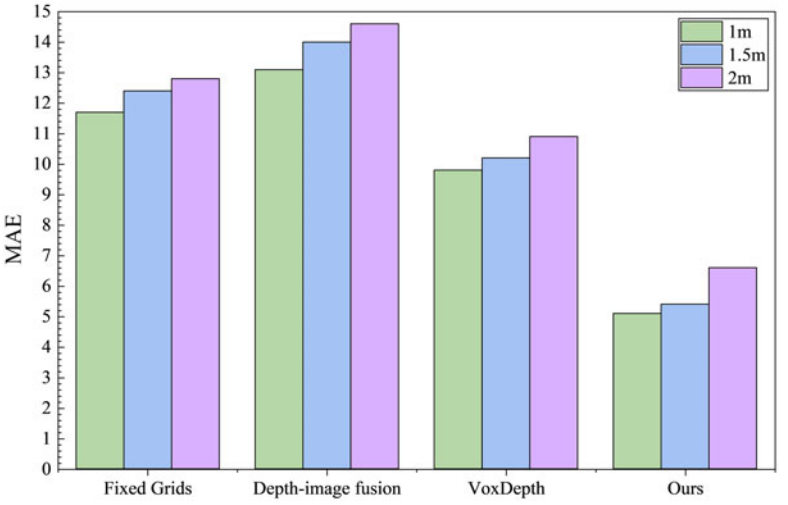


Figure 10. *MAE of depth estimation methods.*

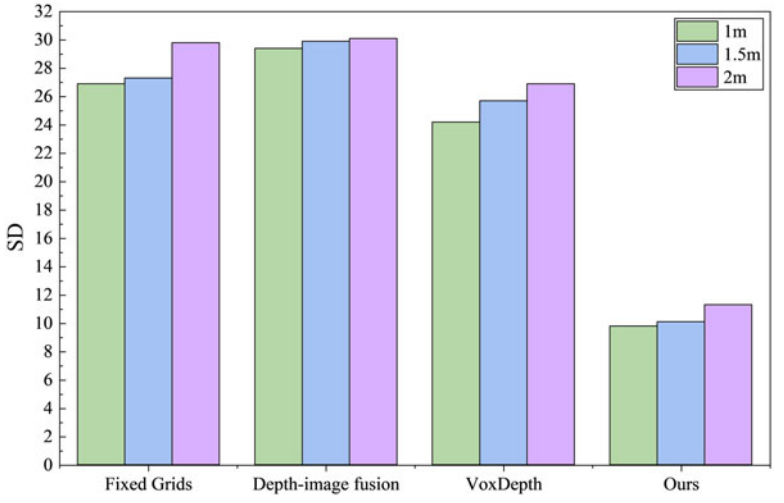


Figure 11. *SD of depth estimation methods.*

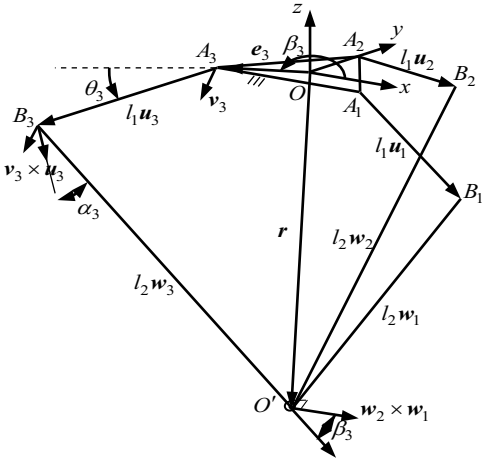


Figure 12. *Mechanism schematic diagram of Delta robot.*

https://doi.org/10.1017/S0263574726103154 Published online by Cambridge University Press

Table VI. *Dimensional parameters of Delta robot.*

Parameter	Active limb l_1	Passive limb l_2	Static platform bias e
Value	375 mm	950 mm	113 mm

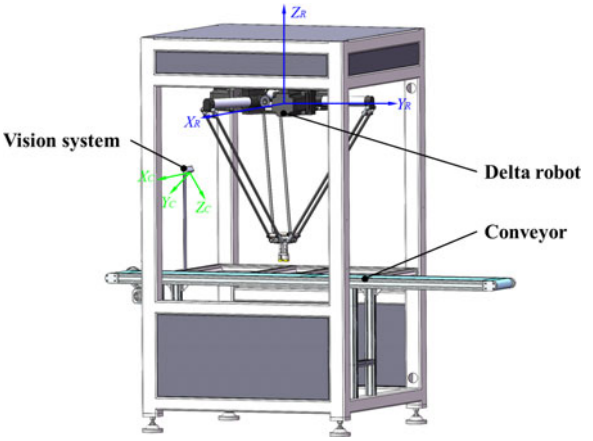


Figure 13. *Layout of robot visual servo system.*

where the modulus of $\boldsymbol{e}_i = e(\cos\beta_i, \sin\beta_i, 0)$ $(i = 1, 2, 3)$ represents the difference in radius of the circumcircle between the moving platform and the static platform, and in Figure 12, it indicates the vector from O to A_i; β_i represents the structural angle of the static platform, with $\beta_i = (i-1)2\pi/3 - \pi/6$; l_1, l_2, $\boldsymbol{u}_i$, and $\boldsymbol{w}_i$ represent the limb lengths and unit vectors of the active and passive limbs in branch i, respectively, and $\boldsymbol{u}_i$ is shown in Eq. (19):

$$\boldsymbol{u}_i = (\cos\beta_i \cos\theta_i \quad \sin\beta_i \cos\theta_i \quad -\sin\theta_i)^{\mathrm{T}} \tag{19}$$

where θ_i represents the rotation angle of the active limb i $(i = 1, 2, 3)$. By identity transformation, we get Eq. (20):

$$\boldsymbol{w}_i = \frac{1}{l_2}(\boldsymbol{r} - \boldsymbol{e}_i - l_1\boldsymbol{u}_i) \tag{20}$$

Taking the derivative of Eq. (18), we get Eq. (21):

$$\dot{\boldsymbol{r}} = l_1\dot{\theta}_i(\boldsymbol{v}_i \times \boldsymbol{u}_i) + l_2\boldsymbol{\omega}_i \times \boldsymbol{w}_i, \quad i = 1, 2, 3 \tag{21}$$

where $\dot{\boldsymbol{r}}$ represents the velocity vector of point O'; $\dot{\theta}_i$ represents the angular velocity vector of the active limb i; ω_i represents the angular velocity of the passive limb i; $\boldsymbol{v}_i$ represents the unit vector perpendicular to the plane formed by $\boldsymbol{e}_i$ and $\boldsymbol{u}_i$, and $\boldsymbol{v}_i$ is shown in Eq. (22):

$$\boldsymbol{v}_i = (-\sin\beta_i \quad \cos\beta_i \quad 0)^{\mathrm{T}} \tag{22}$$

The identity transformation yields Eq. (23):

$$\dot{\boldsymbol{\theta}} = \boldsymbol{J}_q^{-1}\boldsymbol{J}_x\dot{\boldsymbol{r}} = \boldsymbol{J}\dot{\boldsymbol{r}} \tag{23}$$

where $\boldsymbol{J}_q = \mathrm{diag}[l_1\boldsymbol{w}_i^{\mathrm{T}}(\boldsymbol{v}_i \times \boldsymbol{u}_i)]$ represents the direct Jacobian matrix, $\boldsymbol{J}_x = [\boldsymbol{w}_1\ \boldsymbol{w}_2\ \boldsymbol{w}_3]^{\mathrm{T}}$ represents the indirect Jacobian matrix, and $\boldsymbol{J}$ represents the Jacobian matrix.

Then the camera model is established, and its relationship with the robotic system layout is shown in Figure 13 and marked the robot coordinate system O-$X_RY_RZ_R$ and the camera coordinate system O_C-$X_CY_CZ_C$. The target can be described by the transformation relationship between the robot coordinate system and the pixel coordinate system, which involves the robot coordinate system, camera coordinate system, image coordinate system, and pixel coordinate system. Eq. (24) represents the transformation relationship between these coordinate systems.

https://doi.org/10.1017/S0263574726103154 Published online by Cambridge University Press

Table VII. *Intel RealSense D455 intrinsic and key parameters.*

	f (mm)	f_x (pixel)	f_y (pixel)	u_0 (pixel)	v_0 (pixel)	Resolution	FOV
Left imager	1.93	647.44	643.99	640.14	360.46	1,280 × 720	H87, V58
		323.50	429.22	320.57	240.23	640 × 480	H75, V62
Right imager	1.93	647.81	643.72	639.95	360.35	1,280 × 720	H87, V58
		323.49	429.17	320.46	240.32	640 × 480	H75, V62
Color camera	1.93	642.59	712.64	640.37	400.16	1,280 × 800	H90, V65
		321.25	427.39	320.55	240.11	640 × 480	

$$Z_C \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{1}{dx} & 0 & u_0 \\ 0 & \frac{1}{dy} & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{R} & \boldsymbol{T} \\ \boldsymbol{O} & 1 \end{bmatrix} \begin{bmatrix} X_R \\ Y_R \\ Z_R \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & u_0 & 0 \\ 0 & f_y & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{R} & \boldsymbol{T} \\ \boldsymbol{O} & 1 \end{bmatrix} \begin{bmatrix} X_R \\ Y_R \\ Z_R \\ 1 \end{bmatrix} \tag{24}$$

where Z_C represents the depth distance measured by D455, u and v represent the horizontal and vertical coordinates in the pixel coordinate system. The matrix $\begin{bmatrix} \frac{1}{dx} & 0 & u_0 \\ 0 & \frac{1}{dy} & v_0 \\ 0 & 0 & 1 \end{bmatrix}$ represents the transformation relationship matrix between the pixel coordinate system and the image coordinate system, d_x and d_y represent the width and height of a single pixel, u_0 and v_0 represent the pixel coordinates of the center point of the image. The matrix $\begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$ represents the transformation relationship matrix between the image coordinate system and the camera coordinate system, and f represents the focal length. The matrix $\begin{bmatrix} \boldsymbol{R} & \boldsymbol{T} \\ \boldsymbol{O} & 1 \end{bmatrix}$ represents the transformation relationship matrix between the camera coordinate system and the robot coordinate system, which is also the extrinsic parameter matrix. $\boldsymbol{R}$ represents the rotation matrix, with values of $\begin{bmatrix} 1 & 0 & 0 \\ 0 & -0.86 & -0.52 \\ 0 & -0.52 & -0.86 \end{bmatrix}$, $\boldsymbol{T}$ represents the translation matrix, with values of $[85.00 \ -582.68 \ -406.27]^{\mathrm{T}}$, and $\boldsymbol{O}$ represents the zero matrix. X_R, Y_R, and Z_R represent the three-dimensional coordinate values of the target in the robot coordinate system, respectively. In the second equation, $\begin{bmatrix} f_x & 0 & u_0 & 0 \\ 0 & f_y & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$ represents the intrinsic parameter matrix, and f_x and f_y represent the equivalent focal length in the x-axis and y-axis directions, in pixels. After camera calibration, the intrinsic parameters of each imaging module of the Intel RealSense D455 were measured, and the key parameters obtained from the official technical documents [27] are shown in Table VII. In addition, the baseline distance between the left and right imaging modules is 95 mm, and the field of view (FOV) of the infrared projector is H90° V63°.

Through the robot kinematic model and camera model, the three-dimensional coordinate information of the EE and the target workpiece can be obtained. Based on this, this paper designs a simple and efficient visual servo control strategy to achieve precise positioning and smooth movement of the Delta robot in a dynamic environment.

Let the current position of the EE be $\boldsymbol{P}_{\text{current}} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$, $\boldsymbol{R}_{\text{current}} \in \mathbb{R}^{3\times3}$, and the target position be $\boldsymbol{P}_{\text{desired}} = \begin{bmatrix} x_d \\ y_d \\ z_d \end{bmatrix}$, $\boldsymbol{R}_{\text{desired}} \in \mathbb{R}^{3\times3}$. Define the position error as $\boldsymbol{e}_p = \boldsymbol{P}_{\text{desired}} - \boldsymbol{P}_{\text{current}}$. To ensure that the system has

https://doi.org/10.1017/S0263574726103154 Published online by Cambridge University Press

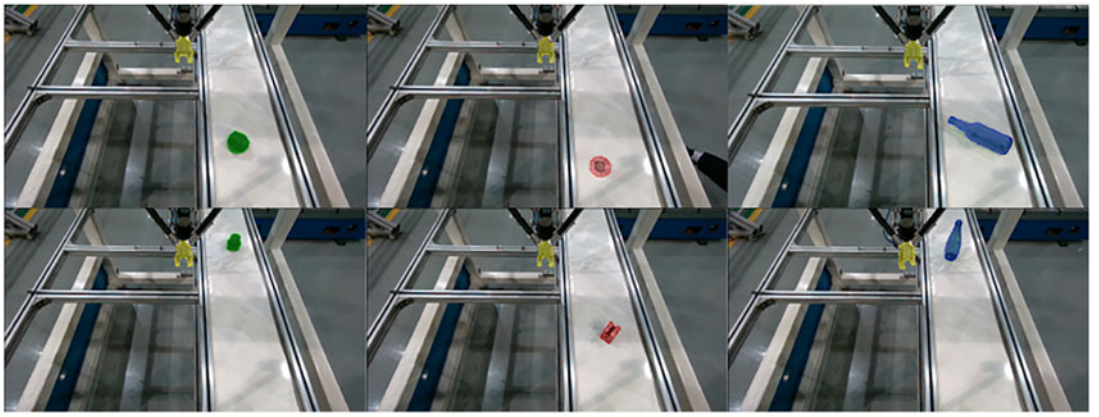


Figure 14. *Instance segmentation.*

a quick response and good robustness in a dynamic environment, this paper employs a proportional-derivative (PD) control law. Let the control input be the desired velocity of the EE $\dot{\boldsymbol{r}}_d$; then the control law is designed as follows in Eq. (25):

$$\dot{\boldsymbol{r}}_d = \boldsymbol{K}_p \boldsymbol{e}_p + \boldsymbol{K}_d \dot{\boldsymbol{e}}_p \tag{25}$$

where $\boldsymbol{K}_p$ and $\boldsymbol{K}_d$ are positive definite gain matrices. This control law not only ensures the rapid attenuation of the error but also takes into account the dynamic response and anti-disturbance performance of the system. In order to obtain the joint control command, the relationship between the desired velocity $\dot{\boldsymbol{r}}_d$ of the EE in the operation space and the desired velocity $\dot{\boldsymbol{\theta}}_d$ of the joint space is also similar to Eq. (23), that is shown in Eq. (26):

$$\dot{\boldsymbol{\theta}} = \boldsymbol{J}(\boldsymbol{\theta}) \left(\boldsymbol{K}_p \boldsymbol{e}_p + \boldsymbol{K}_d \dot{\boldsymbol{e}}_p \right) \tag{26}$$

In order to verify the effectiveness of the control law, this paper uses the Lyapunov function method to perform stability analysis. Assume that the candidate Lyapunov function is shown in Eq. (27):

$$V = \frac{1}{2} \boldsymbol{e}_p^{\mathrm{T}} \boldsymbol{e}_p \tag{27}$$

By selecting an appropriate PD gain, it can be satisfied that $\dot{V} < 0$, thus ensuring the global asymptotic stability of the closed-loop system.

5. Experiment

The system is deployed on a Delta robot platform equipped with an Intel RealSense D455 depth camera and Advantech EPC-P3086 industrial computer. The robot control frequency is set to 40 Hz to meet the requirements of high-speed operation. As shown in Figure 14, the model can achieve accurate segmentation on plastic workpieces, metal workpieces, glass workpieces, and EE.

As shown in Figure 15(a), the 3D trajectory of the EE while performing the grasping task. Figure 15(b) illustrates that the EE velocity smoothly converges to the desired value within 0.25 s, demonstrating the quick response and stability of the system, while also meeting practical production demands of 60−80 times per minute. The end-to-end latency, which includes perception at 17.2 ms, depth estimation at 3.4 ms, and control computation at 1.2 ms, totals 21.8 ms, satisfying the real-time requirement of 40 Hz.

Record the endpoint pose of an experiment as $\hat{\boldsymbol{T}} = (\hat{\boldsymbol{p}}, \hat{\boldsymbol{R}})$, with the ground truth target pose $\boldsymbol{T}^* = (\boldsymbol{p}^*, \boldsymbol{R}^*)$. Where $\hat{\boldsymbol{p}}$, $\boldsymbol{p}^*$, $\hat{\boldsymbol{R}}$, and $\boldsymbol{R}^*$ represent the actual and target position vectors and rotation matrices, respectively. The position error and the orientation error are defined as Eqs. (28) and (29):

https://doi.org/10.1017/S0263574726103154 Published online by Cambridge University Press

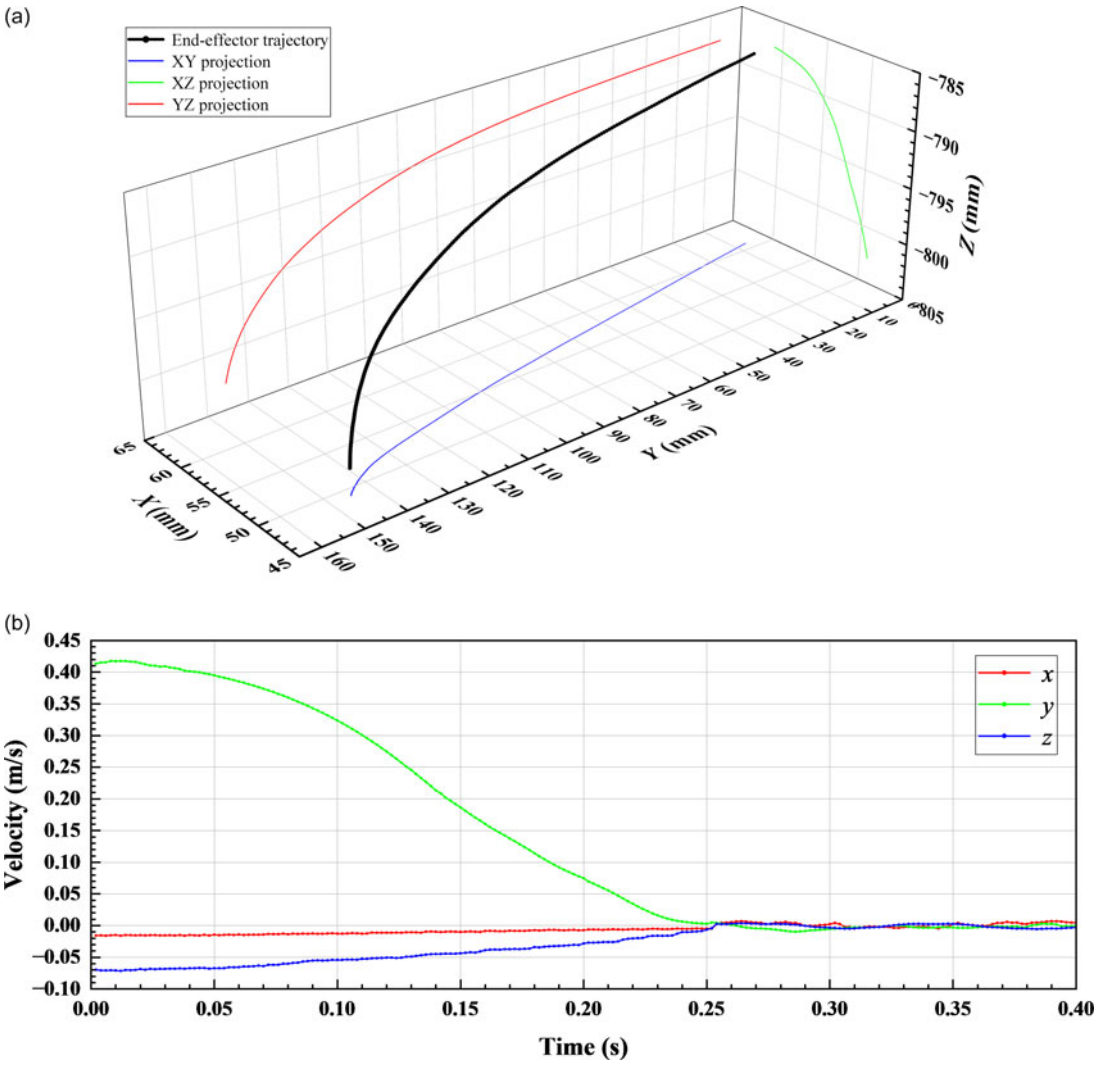


Figure 15. *EE grasping, (a) trajectory, (b) velocity.*

$$\boldsymbol{e}_p = \left\| \hat{\boldsymbol{p}} - \boldsymbol{p}^* \right\| \tag{28}$$

$$\boldsymbol{e}_\theta = \arccos\left(\frac{\text{trace}\left(\boldsymbol{R}^{*\mathrm{T}} \hat{\boldsymbol{R}} \right) - 1}{2} \right) \tag{29}$$

Let the task completion time be t_{done}. The time from the start until the first moment the condition is met and remains stable for more than t_{hold} seconds must be less than $T_{\max}$. A single experiment is judged as Eq. (30):

$$\text{success} = \begin{cases} 1 & \text{if } \boldsymbol{e}_p \leqslant d_{th} \wedge \boldsymbol{e}_\theta \leqslant \theta_{th} \wedge t_{done} \leqslant T_{\max} \\ 0 & \text{otherwise} \end{cases} \tag{30}$$

https://doi.org/10.1017/S0263574726103154 Published online by Cambridge University Press

Table VIII. *Comparison of visual servoing grasping methods.*

Methods	Success rate (%)	Positioning error (mm)	Control frequency (Hz)
HDLVS	87.5	8.3	25
DVSFN	91.4	5.2	39
GT/LT-RICP + LK	92.3	5.0	10
Ours	95.2	5.7	40

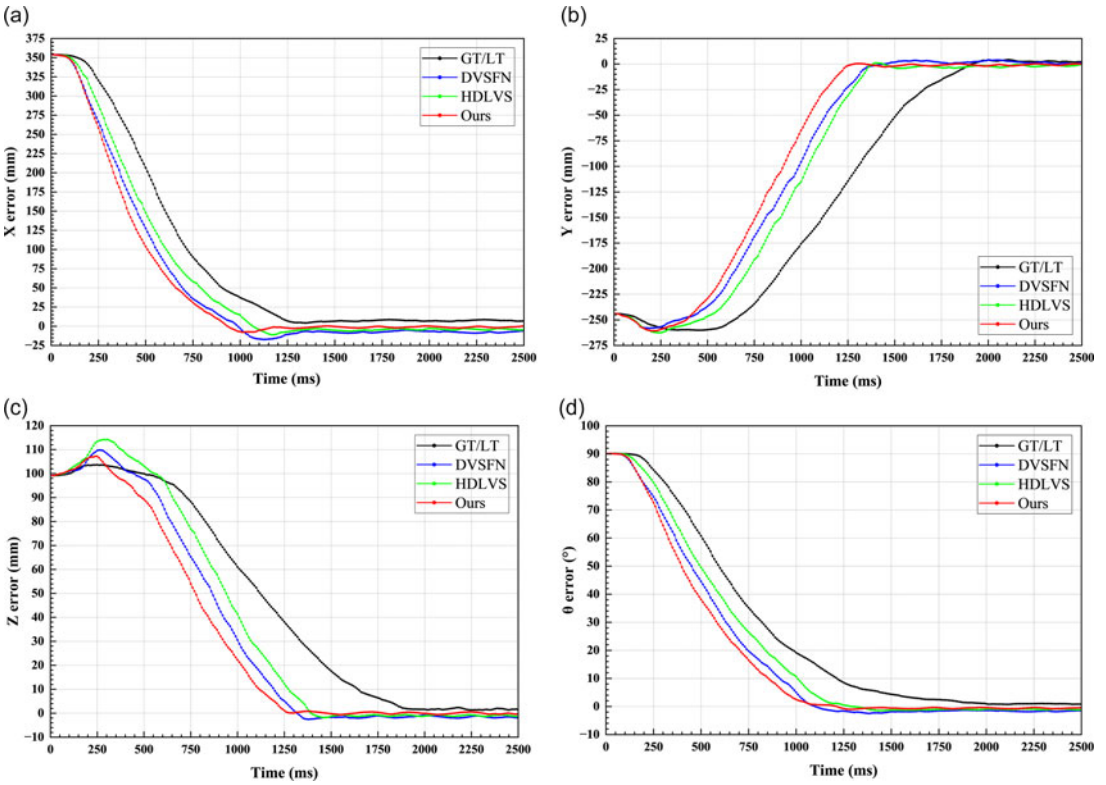


Figure 16. *Convergence curves of grasping error, (a) X error, (b) Y error, (c) Z error, (d) θ error.*

The task success rate (TSR) is calculated by Eq. (31):

$$\mathrm{TSR} = \frac{1}{N}\sum_{i=1}^{N} \mathrm{success}_i \times 100\% \tag{31}$$

where the position threshold $d_{th} = 0.01$ m, the angle threshold $\theta_{th} = 5°$, the time limit $T_{\max} = 5$ s, and the stable holding time $t_{\mathrm{hold}} = 1$ s.

Furthermore, our method is directly compared with other existing visual servoing grasping approaches, as detailed in Table VIII. It can be seen that although the hybrid deep learning visual servoing (HDLVS) method proposed by Botezatu et al. [28] can meet real-time requirements, its success rate and positioning accuracy need to be improved. The deep visual servoing feature network (DVSFN) method proposed by Luo et al. [29] performs similarly to our method in terms of real-time capability and positioning accuracy, but still has a lower success rate. The GT/LT-RICP + LK method designed by Ruan et al. [30] combines the global template/local template-based RANSAC and iterative closest point unified algorithm (GT/LT-RICP) with Lucas-Kanade optical flow estimation. Although it has a

https://doi.org/10.1017/S0263574726103154 Published online by Cambridge University Press

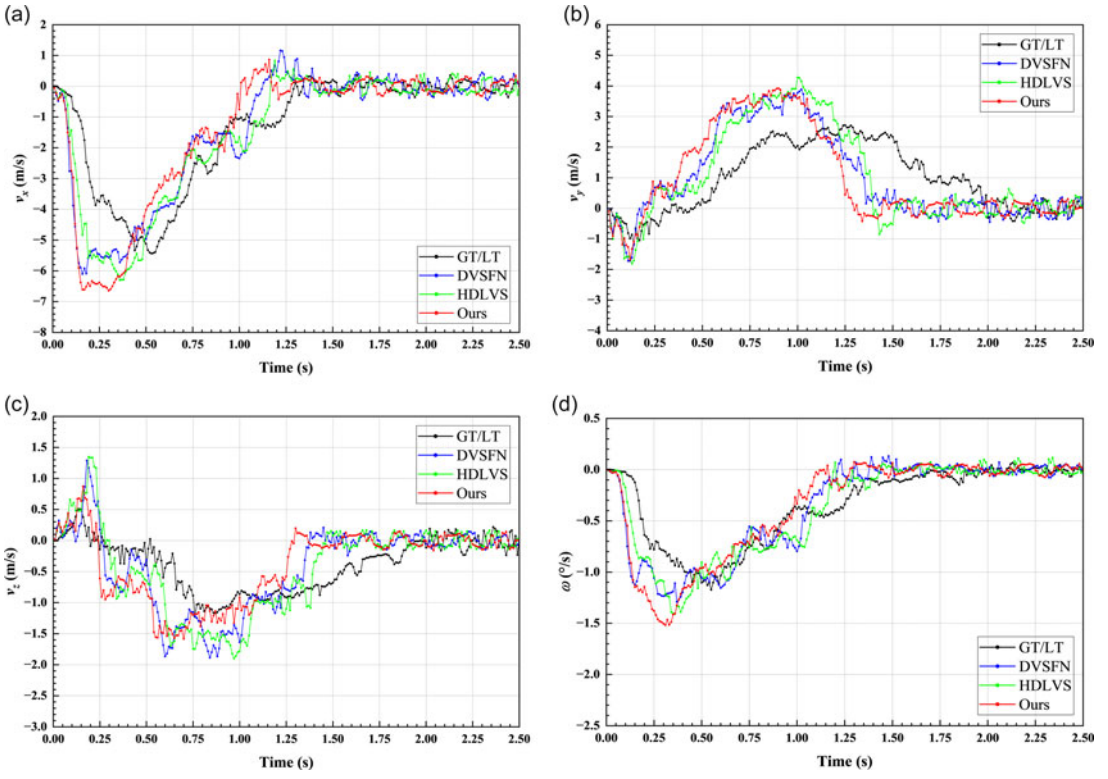


Figure 17. *Convergence curves of velocity error, (a) v_x error, (b) v_y error, (c) v_z error, (d) ω error.*

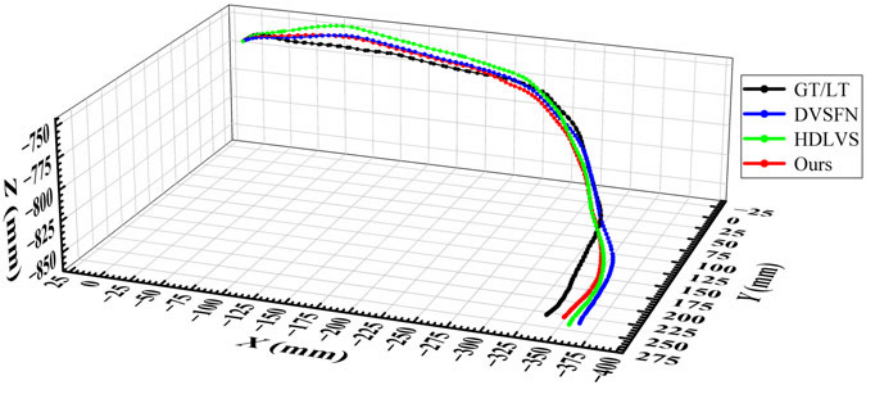


Figure 18. *Trajectory comparison.*

high success rate and positioning accuracy, its control frequency makes it difficult to meet real-time requirements.

The displacement error curves, velocity error curves, and 3D trajectory comparison diagrams obtained from testing the above four methods are shown in Figures 16, 17, and 18, respectively. Conduct 50 parallel trials for each of the four methods, respectively, and obtain the TSR as shown in Figure 19. As shown in Figure 20, the image sequence of the robot grasping plastic, metal, and glass workpieces, respectively.

https://doi.org/10.1017/S0263574726103154 Published online by Cambridge University Press

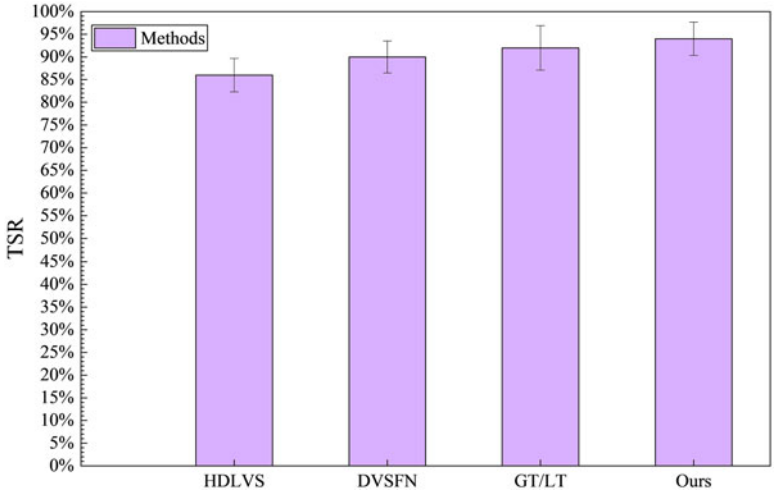


Figure 19. *Four types of vision servo TSR.*

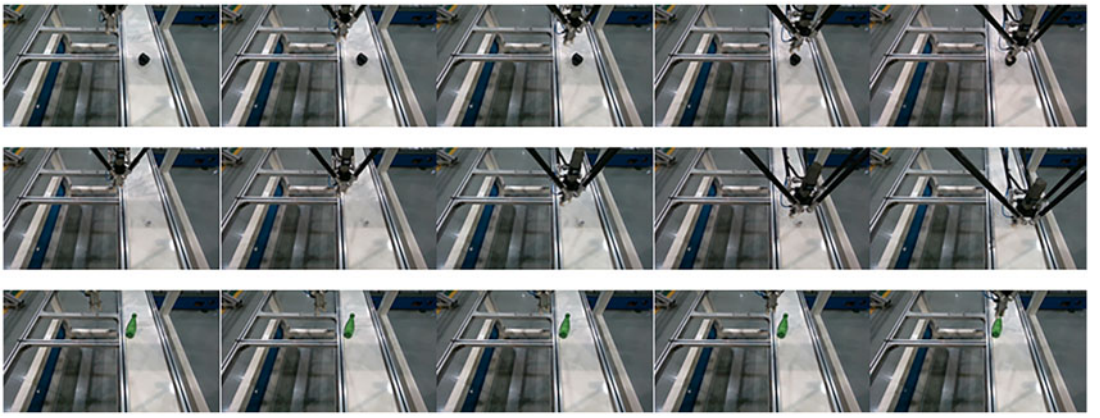


Figure 20. *Grasping image sequences.*

6. Conclusions

This article addresses the vision-guided grasping tasks of Delta robots in unstructured environments and presents a comprehensive solution based on RGB-D multimodal learning. By integrating the heterogeneous features of RGB images and depth images, and combining an improved Mask R-CNN architecture with the DGA mechanism, it achieves high-precision instance segmentation and robust depth estimation. The key contributions are as follows:

1. MMF Architecture: A dual-stream ResNet50-FPN backbone network is designed, achieving cross-modal feature adaptive fusion at the P3-P5 levels through shallow feature sharing and deep modality differentiation. Experiments show that this strategy increases the segmentation mAP@0.5 to 95.2%, an improvement of 10.5% over the single-modal baseline, significantly enhancing the recognition capability for plastic, metal, and glass objects.
2. DGA Mechanism: Introducing the cross-modal interaction layer and the gated fusion unit, dynamically weighting geometric and semantic features. Ablation experiments show that the DGA module improves the segmentation IoU from 0.68 to 0.81, effectively alleviating issues of material ambiguity and reflective interference.

https://doi.org/10.1017/S0263574726103154 Published online by Cambridge University Press

3. DDE Algorithm: Employing the noise-distance adaptive grid partitioning strategy, the MAE of depth estimation was reduced from 12.3 to 5.7 mm, with the SD decreasing by 63%. The use of inverse variance weighted median filtering helped to suppress the effects of sensor noise and depth fluctuations.
4. Real-time Visual Servo Control: Combining the PD control law with Jacobian kinematic mapping, stable tracking of the EE trajectory is achieved. The system has an end-to-end latency of 21.8 ms and supports high-frequency control at 40 Hz, meeting the real-time requirements of industrial scenarios.

This study provides reliable technical support for the autonomous operation of robots in the intelligent manufacturing industry, enhancing the practical applicability of visual servo methods in robotics and promoting their large-scale application in scenarios such as electronic assembly and logistics sorting. Although the methods presented in this paper have achieved significant improvements in accuracy and efficiency, future exploration could focus on compressing model parameters through neural architecture search to create lightweight algorithms; integrating tactile/force feedback to construct a visual-force control closed loop, thereby enhancing the safety of grasping deformable objects; and studying temporal feature fusion and predictive control algorithms for high-speed moving targets to overcome existing frame rate limitations.

Author contribution. Puxuan Ning developed the MMF module, DGA mechanism, DDE algorithm, and real-time control for the robot, implemented the applications, and wrote the article. Jiangping Mei contributed to supervision, technical correction, and verification of the robot's algorithms and control, as well as the analysis of the results. Jinlu Ni conceived and led the robot development project and secured funding. Peichen Liu contributed to the revision of the article. Tianlin Wei and Xiaochen Su set up the experimental platform and assisted with experiments. All authors have read and approved the final version of the manuscript.

Financial support. This work was supported by the National Natural Science Foundation of China (Grant No.52375507).

Competing interests. The authors declare no conflicts of interest.

Ethical approval. Not applicable.

References

[1] J. Zhu, D. Navarro-Alarcon, R. Passama and A. Cherubini, "Vision-based manipulation of deformable and rigid objects using subspace projections of 2D contours," *Robot. Auton. Syst.* **142**, 103798 (2021). doi: 10.1016/j.robot.2021.103798.
[2] R. Lagneau, A. Krupa and M. Marchal, "Automatic shape control of deformable wires based on model-free visual servoing," *IEEE Robot. Autom. Lett.* **5**(4), 5252–5259 (2020). doi: 10.1109/LRA.2020.3007114.
[3] A. Ghasemi, P. Li and W.-F. Xie, "Adaptive switch image-based visual servoing for industrial robots," *Int. J. Control Autom. Syst.* **18**(5), 1324–1334 (2020). doi: 10.1007/s12555-018-0753-y.
[4] M. Zhu, S. Briot and A. Chriette, "Sensor-based design of a Delta parallel robot," *Mechatronics* **87**, 102893 (2022). doi: 10.1016/j.mechatronics.2022.102893.
[5] R. Wang, X. Zhang, Y. Fang and B. Li, "Virtual-goal-guided RRT for visual servoing of mobile robots with FOV constraint," *IEEE Trans. Syst. Man Cybern., Syst.* **52**(4), 2073–2083 (2022). doi: 10.1109/TSMC.2020.3044347.
[6] J. Fried, A. C. Leite and F. Lizarralde, "Uncalibrated image-based visual servoing approach for translational trajectory tracking with an uncertain robot manipulator," *Control Eng. Pract.* **130**, 105363 (2023). doi: 10.1016/j.conengprac.2022.105363.
[7] R. Wang, Z. Zhu, X. Zhang, Y. Fang and B. Li, "Visual servoing trajectory tracking and depth identification for mobile robots with velocity saturation constraints," *IEEE Trans. Ind. Electron.* **71**(6), 5950–5959 (2024). doi: 10.1109/TIE.2023.3290254.
[8] Y. Chen, L. Lan, X. Liu, G. Zeng, C. Shang, Z. Miao, H. Wang, Y. Wang and Q. Shen, "Adaptive stiffness visual servoing for unmanned aerial manipulators with prescribed performance," *IEEE Trans. Ind. Electron.* **71**(9), 11028–11038 (2024). doi: 10.1109/TIE.2023.3344827.
[9] J. Wu, Z. Jin, A. Liu, L. Yu and F. Yang, "A hierarchical data-driven predictive control of image-based visual servoing systems with unknown dynamics," *IEEE Trans. Cybern.* **54**(5), 3160–3173 (2024). doi: 10.1109/TCYB.2022.3228123.
[10] A. K. Kamath, Z. Zheng and M. Feroskhan, "Depth-independent augmented dynamics visual servoing of multirotors," *IEEE Trans. Aerosp. Electron. Syst.* **61**(2), 5444–5453 (2025). doi: 10.1109/TAES.2024.3496418.
[11] G. C. Daraviña, J. L. Valencia, G. A. Holguin, H. F. Quintero, E. A. Ariza and D. Vergara, "Visual servoing and kalman filter applied to parallel manipulator 3-RRR," *Electronics* **13**(14), 2703 (2024). doi: 10.3390/electronics13142703.

https://doi.org/10.1017/S0263574726103154 Published online by Cambridge University Press

[12] N. Zhu, W.-F. Xie and H. Shen, "Position-based visual servoing of a 6-RSS parallel robot using adaptive sliding mode control," *ISA Trans.* **144**, 398–408 (2024). doi: 10.1016/j.isatra.2023.10.029.
[13] M. M. Da Silva, F. T. Colombo, G. C. De Oliveira and L. P. R. De Oliveira, "Hybrid vision/strain-based control strategy for a parallel manipulator with flexible links," *Mech. Mach. Theory* **191**, 105508 (2024). doi: 10.1016/j.mechmachtheory.2023.105508.
[14] E. G. Ribeiro, R. De Queiroz Mendes and V. Grassi, "Real-time deep learning approach to visual servo control and grasp detection for autonomous robotic manipulation," *Robot. Auton. Syst.* **139**, 103757 (2021). doi: 10.1016/j.robot.2021.103757.
[15] A. Al-Shanoon and H. Lang, "Robotic manipulation based on 3-D visual servoing and deep neural networks," *Robot. Auton. Syst.* **152**, 104041 (2022). doi: 10.1016/j.robot.2022.104041.
[16] N. Han, X. Ren and D. Zheng, "Visual servoing control of robotics with a neural network estimator based on spectral adaptive law," *IEEE Trans. Ind. Electron.* **70**(12), 12586–12595 (2023). doi: 10.1109/TIE.2023.3237881.
[17] P. Jiang, Y. Ishihara, N. Sugiyama, J. Oaki, S. Tokura, A. Sugahara and A. Ogawa, "Depth image–Based deep learning of grasp planning for textureless planar-faced objects in vision-guided robotic bin-picking," *Sensors* **20**(3), 706 (2020). doi: 10.3390/s20030706.
[18] Y. Han, K. Yu, R. Batra, N. Boyd, C. Mehta, T. Zhao, Y. She, S. Hutchinson and Y. Zhao, "Learning generalizable vision-tactile robotic grasping strategy for deformable objects via transformer," *IEEE/ASME Trans. Mechatron.* **30**(1), 554–566 (2025). doi: 10.1109/TMECH.2024.3400789.
[19] J. K. Odira, N. Andreff, L. Petiet and J. B. Byiringiro, "Visual servoing of a laser beam through a mirror," *Robotica* **40**(9), 3157–3177 (2022). doi: 10.1017/S026357472200011X.
[20] S. H. Hong, B. Albia, T. Kyzer, J. Cornelius, E. R. Mark, A. J. Hall and Y. Wang, "Artificial neural network-based model predictive visual servoing for mobile robots," *Robotica* **42**(8), 2825–2847 (2024). doi: 10.1017/S0263574724001176.
[21] A. K. Kamath, S. G. Anavatti and M. Feroskhan, "A physics-informed neural network approach to augmented dynamics visual servoing of multirotors," *IEEE Trans. Cybern.* **54**(11), 6319–6332 (2024). doi: 10.1109/TCYB.2024.3413072.
[22] O. Perezyabov, M. Gavrilenkov and I. Afanasyev. "Depth and Image Fusion for Road Obstacle Detection Using Stereo Camera," **In:** *31st Conference of Open Innovations Association (FRUCT)* (IEEE, Helsinki, 2022) pp. 237−244. doi: 10.23919/FRUCT54823.2022.9770927.
[23] Y. Chakrabarty, A. Dixit and S. R. Sarangi, "VoxDepth: Rectification of depth images on edge devices," *ACM Trans. Embed. Comput. Syst.* **24**, 3763793 (2025). doi: 10.1145/3763793.
[24] F. Hui, Z. Zhou and Y. Liu, "PL-LVI: A liDAR-visual-inertial SLAM system integrating visual point-line features," *IEEE Trans. Automat. Sci. Eng.* **22**, 1 (2025). doi: 10.1109/TASE.2025.3559668.
[25] Y. Liu and Z. Zhou, "Optical flow-based stereo visual odometry with dynamic object detection," *IEEE Trans. Comput. Soc. Syst.* **10**(6), 3556–3568 (2023). doi: 10.1109/TCSS.2022.3205015.
[26] J. Ni, J. Mei and W. Hu, "A hierarchical approach for rigid-body dynamics model simplification of a high-speed parallel robot by considering kinematics performance," *Sci. Prog.* **104**, 3685042110630 (2021). doi: 10.1177/00368504211063072.
[27] I. Corporation, "Intel® RealSenseTM Camera D400 series product family datasheet," *Revision* **019**, 337029–337017 (2024). https://www.intel.com/design/literature.htm
[28] A.-P. Botezatu, A.-I. Iancu and A. Burlacu, "Hybrid deep learning framework for eye-in-hand visual control systems," *Robotics* **14**(5), 66 (2025). doi: 10.20944/preprints202503.1552.v1.
[29] J. Luo, Z. Zhang, Y. Wang and R. Feng, "Robot closed-loop grasping based on deep visual servoing feature network," *Actuators* **14**(1), 25 (2025). doi: 10.3390/act14010025.
[30] Y. Luo, S. Ruan, Y. Li and D. Chen, "Hand–eye coordinated grasping method for textured targets in unstructured dynamic scenes," *Chin. J. Mech. Eng.* **38**(1), 160 (2025). doi: 10.1186/s10033-025-01344-1.

Cite this article: P. Ning, J. Mei, J. Ni, P. Liu, T. Wei and X. Su (2026). "Red green blue-depth multimodal learning for vision-guided grasping in Delta robots", Robotica **44**, 252–274. https://doi.org/10.1017/S0263574726103154

https://doi.org/10.1017/S0263574726103154 Published online by Cambridge University Press