Applying Visual Servoing for Industrial Robots Through Interactive Segmentation and Robot-Computer Communication

Zeinel Momynkulov[1], Moldir Kuatova[2], Amandyk Tuleshov[3]
Joldasbekov Institute of Mechanics and Engineering, Kazakhstan[1, 2, 3]
International Information Technology University, Kazakhstan[1]

Abstract—**Industrial robots are increasingly required to operate in flexible manufacturing environments where conventional offline programming and manually defined working positions limit production adaptability. This study presents a visual servoing framework that integrates interactive image segmentation with industrial robot control through robot-computer communication. The proposed system enables an operator to specify the desired manipulation region directly on an RGB-D camera image, eliminating the need for predefined object models or manually generated trajectories. A fine-tuned Segment Anything Model (SAM) is employed to obtain accurate segmentation masks under industrial conditions, while depth information and hand-eye calibration are used to transform the selected image region into robot coordinates. The generated positions are transmitted to an industrial robot through a TCP/IP communication interface, where they are executed by the robot controller. A graphical user interface was developed to combine image acquisition, interactive segmentation, trajectory visualization, and communication monitoring within a single application. Experimental evaluation demonstrated reliable robot-computer communication and stable segmentation performance in the presence of illumination changes, metallic reflections, and partial occlusions. The proposed segmentation module achieved an average confidence of approximately 91% with an average inference time of 46 ms, enabling near real-time operation for industrial visual servoing. The obtained results indicate that the proposed framework simplifies robot deployment while maintaining accurate target localization and compatibility with existing industrial robotic platforms, making it a practical solution for flexible manufacturing applications.**

Keywords—Industrial robotics; interactive segmentation; human-robot interaction; robot-computer communication; industrial automation

I. Introduction

Industrial automation has continuously evolved alongside the growing demand for higher productivity, consistent product quality, and safer manufacturing environments. The introduction of the first industrial robots in the early 1960s marked a significant milestone in modern production, allowing repetitive and hazardous operations to be performed with remarkable precision [1]. These robotic systems were primarily designed for highly structured environments, where every movement was explicitly programmed and repeated with minimal variation. Although such an approach significantly improved manufacturing efficiency, it also limited the ability of robots to adapt to changes in workspaces or production layouts [2]. As manufacturing processes became increasingly diversified, the need for more flexible robotic systems gradually emerged, encouraging researchers to investigate new approaches to robot programming and control [3].

One of the major directions in this evolution was the integration of perception into robotic systems. Instead of executing predefined trajectories without considering the surrounding environment, robots gradually became capable of receiving sensory feedback from cameras and other measurement devices. This transition led to the development of visual servoing, where visual information is directly incorporated into the robot control loop [4].

Unlike conventional offline programming, visual servoing continuously updates robot motion according to observed image features, allowing the manipulator to compensate for positioning errors, calibration inaccuracies, and object displacement during task execution [5]. Because of these advantages, visual servoing has been successfully applied in robotic assembly, inspection, object manipulation, and precision manufacturing for several decades, becoming one of the fundamental approaches in vision-based robot control [6].

The concept of intelligent manufacturing expanded considerably with the emergence of Industry 4.0 [7]. Modern production facilities increasingly rely on interconnected cyber-physical systems, cloud computing, industrial Internet of Things technologies, and digital twins to improve operational efficiency and production flexibility. In these environments, industrial robots are no longer isolated machines executing predefined programs. Instead, they operate as components of distributed intelligent systems that continuously exchange information with sensors, production equipment, and supervisory software [8]. This transformation has shifted industrial robotics toward adaptive manufacturing, where rapid reconfiguration and autonomous decision-making have become essential requirements rather than optional features.

The rapid progress of artificial intelligence has accelerated this transition even further. During the last decade, deep learning techniques have significantly improved the ability of robots to perceive and interpret complex industrial scenes. Modern neural networks demonstrate remarkable performance in object detection, semantic segmentation, pose estimation, and scene understanding, often outperforming traditional computer vision algorithms based on handcrafted image features [9]. These capabilities allow robotic systems to operate more reliably under varying illumination conditions, partial occlusions, and cluttered industrial environments that frequently occur in real production facilities. As a result, AI-based perception has become an important component of intelligent robotic automation across numerous industrial applications [10].

Recently, foundation vision models have introduced a new level of flexibility for industrial perception. Unlike conventional deep learning models that require extensive task-specific datasets and repeated training, these large-scale models are capable of generalizing across a wide variety of previously unseen objects [11]. Interactive segmentation further simplifies human-robot interaction by allowing an operator to indicate the desired object or working region using only a few visual prompts [12]. Such an approach considerably reduces the effort required to deploy robotic systems in new production scenarios while preserving accurate object localization [13]. Consequently, interactive segmentation has attracted increasing attention as a practical solution for flexible industrial robotics, particularly in environments characterized by small production batches and frequent product changes.

Despite these technological advances, many industrial robotic systems are still programmed through manually specified positions or fixed working zones established during system commissioning. Even relatively small changes in product geometry or workspace configuration may require additional calibration procedures, modifications of robot programs, or retraining of perception models. These limitations increase deployment time and reduce production flexibility, especially in manufacturing processes where new products are introduced regularly.

To address these challenges, this study proposes a visual servoing framework that combines interactive image segmentation with industrial robot control. Instead of manually defining manipulation coordinates or preparing dedicated datasets for every new object, the proposed approach enables an operator to specify the desired working region directly on the camera image. The segmented region is subsequently transformed into robot coordinates and utilized for robot motion generation. By combining interactive perception with computer vision, the proposed framework aims to simplify robot deployment while maintaining compatibility with existing industrial robotic platforms and conventional manufacturing workflows.

II. Related Works

Visual servoing remains an important approach for improving the positioning and manipulation capabilities of robotic systems operating in changing environments. Classical visual servoing methods are generally divided into IBVS and PBVS. In image-based visual servoing, robot motion is calculated directly from image features, whereas position-based methods estimate the three-dimensional pose of the target before generating robot commands. Although these approaches provide accurate closed-loop positioning, their performance can be affected by calibration errors, feature loss, camera motion, image noise, and changes in the observed scene. Wu et al. reviewed both conventional and learning-based visual servoing methods and showed that recent research increasingly applies neural networks, model learning, and optimization techniques to reduce the dependence on manually designed image features [14]. Adaptive control has also been investigated as a way to improve visual servoing under uncertain robot and camera parameters. Lai et al. developed an uncalibrated adaptive visual servoing method for manipulators with uncertainties in both kinematics and dynamics [15]. The controller estimates unknown parameters online and does not require complete prior knowledge of the camera model. This improves adaptability, although the design and tuning of the controller remain relatively complex. Muthusamy et al. introduced a neuromorphic eye-in-hand visual servoing system based on an event camera [16]. Instead of processing complete image frames, the camera records changes in pixel intensity, allowing the system to respond quickly to object motion and illumination variation. However, event-based cameras require specialized hardware and processing algorithms that are not yet common in conventional industrial workcells. The development of deep learning has considerably changed the role of computer vision in robotics. Modern robotic perception systems increasingly use convolutional neural networks and transformer-based architectures for object recognition, pose estimation, grasping, navigation, and human-robot interaction. Shahria et al. reviewed vision-based robotic applications and identified object detection, scene interpretation, calibration, depth estimation, and real-time processing as important components of intelligent robotic systems [17]. Their review also showed that practical deployment remains limited by computational requirements, sensitivity to environmental conditions, and difficulties in transferring perception models between different workspaces. The introduction of the Segment Anything Model provided a new approach to interactive object selection. Kirillov et al. proposed a promptable segmentation model that generates object masks from points, bounding boxes, or automatically produced prompts [18]. SAM demonstrates strong generalization to previously unseen objects and reduces the need to train an individual segmentation model for every industrial component. This property is particularly useful for operator-assisted robot control because an operator can specify a target region using only a small number of clicks. Earlier studies of interactive image segmentation similarly showed that point, stroke, and region-based inputs can substantially reduce manual annotation effort while preserving accurate object boundaries [19]. Nevertheless, general-purpose segmentation models are not always reliable under the imaging conditions found in industrial environments. Blur, noise, weak illumination, reflective surfaces, low contrast, and partial occlusion may reduce mask quality. Chen et al. introduced RobustSAM to improve the performance of SAM on degraded images while preserving its prompt-based and zero-shot capabilities [20]. The model produced more stable masks under several types of image degradation, but it still requires additional model components and greater computational resources than the original SAM architecture. Several studies have also focused on improving object boundaries and extending interactive segmentation beyond conventional binary masks. Fan et al. proposed Segment and Matte Anything, which combines segmentation and image matting in a unified promptable model

[21]. The method improves the representation of fine boundaries, thin structures, and semi-transparent regions. Such boundary refinement can be valuable for precise robotic manipulation, although the model was not designed specifically for real-time robot control. In video processing, Yao et al. reviewed object segmentation and tracking methods and identified temporal consistency, appearance variation, occlusion, and computational efficiency as major challenges [22]. These problems are also relevant to visual servoing because the selected target should remain correctly identified while the robot or camera is moving. A more direct integration of semantic segmentation and robot control was presented by Jiang et al. [23]. Their approach combines referring image segmentation with an eye-in-hand visual servoing system. A target object is specified through a language expression, segmented in the camera image, and represented using geometric constraints that are converted into robot actions. The method demonstrates that semantic masks can replace some manually designed visual features used in classical visual servoing. However, the approach depends on language-based target descriptions and is intended mainly for eye-in-hand configurations. Robot-specific segmentation has recently become another important research direction. Mei et al. introduced RobotSeg, a foundation model and dataset developed for segmenting robots in images and videos [24]. The model addresses the difficulties caused by differences in robot embodiment, articulated structures, cluttered backgrounds, and rapid changes in robot shape during motion. RobotSeg can support visual servoing, real-to-simulation transfer, data augmentation, and safety monitoring. However, its main objective is to segment the robot itself rather than the external object or workspace region selected by an operator. Interactive perception has also been combined with mixed-reality robot programming. Yin et al. proposed a scene-centric programming framework that integrates mixed reality, SAM-assisted segmentation, three-dimensional reconstruction, and virtual object manipulation [25]. The operator specifies the intended operation by interacting with a digital representation of an object rather than manually defining every robot movement. This provides an intuitive programming process, but it requires mixed-reality equipment, three-dimensional scene reconstruction, and a more complex software architecture. Computer vision is also considered an important enabling technology for Industry 5.0. Tzampazaki et al. reviewed the transition of machine vision from Industry 4.0 to Industry 5.0 and emphasized its role in human-centered manufacturing, quality control, intelligent automation, and safe human-robot collaboration [26]. In this context, industrial vision systems should not only automate perception but should also provide clear and accessible interaction mechanisms for human operators. Recent segmentation research has further explored the connection between masks and semantic descriptions. Huang et al. proposed Segment and Caption Anything, which performs region segmentation together with localized image captioning [27]. The method provides richer semantic information than mask generation alone, but its architecture and computational cost are more suitable for general visual understanding than low-latency industrial control. Kweon and Yoon investigated the transfer of knowledge from SAM to weakly supervised semantic segmentation [28]. Their results demonstrate that knowledge encoded in segmentation foundation models can improve task-specific perception even when dense annotations are unavailable. However, the method requires task-oriented training and does not directly provide robot motion control. Robot communication is another essential component of an integrated industrial system. Lin and Hwang presented an architecture for connecting an industrial robot with Industrial Internet of Things services through OPC Unified Architecture [29]. Their work demonstrates the importance of standardized communication between robot controllers, computers, and external industrial services. However, communication and visual perception are treated as separate system functions rather than as parts of an operator-guided visual servoing workflow. Although existing studies have made significant progress in visual servoing, adaptive control, foundation segmentation models, and interactive robot programming, these components are usually investigated separately.

Visual servoing studies mainly concentrate on control stability and feature tracking, while segmentation studies primarily evaluate mask accuracy. Mixed-reality systems provide intuitive interaction but require additional hardware, and industrial communication studies focus mainly on data interoperability. Relatively few approaches combine operator-guided object selection, interactive segmentation, image-to-robot coordinate processing, and direct robot-computer command exchange within a single industrial framework.

The framework proposed in this study addresses this gap by allowing an operator to select a task-relevant region from the camera image using interactive segmentation. The selected mask is processed to determine the target region, after which the corresponding command is transferred from the computer to the industrial robot controller. This approach preserves human supervision while reducing the need for manual coordinate entry and task-specific object detection models. Table I shows a comparison of the most relevant visual servoing, segmentation, robot programming, and communication methods.

Therefore, the proposed work differs from previous research by combining interactive segmentation with real-time robot-computer communication for industrial visual servoing. Instead of requiring predefined object models or complex feature engineering, the operator simply selects the target region through interactive segmentation, after which the system converts the selected region into robot commands for industrial manipulation. This workflow aims to reduce programming complexity while preserving the flexibility required for modern manufacturing environments.

TABLE I. COMPARISON OF RECENT VISUAL SERVOING AND INTERACTIVE SEGMENTATION METHODS

Authors	Description	Strengths	Limitations
Wu et al. (2022)	Survey of learning-based visual servoing control	Systematic comparison of classical and learning-based controllers	Does not present a complete industrial implementation
Lai et al. (2023)	Uncalibrated adaptive visual servoing	Compensates for uncertain kinematics and robot dynamics	Complex adaptive controller design and parameter tuning
Muthusamy et al. (2021)	Neuromorphic eye-in-hand visual servoing	Fast response to motion and illumination changes	Requires an event camera and specialized processing
Shahria et al. (2022)	Review of vision-based robotic applications	Broad analysis of robotic perception components and applications	Does not propose a unified visual servoing framework
Kirillov et al. (2023)	Segment Anything Model	General-purpose promptable segmentation and zero-shot transfer	Not designed specifically for industrial robots
Ramadan et al. (2020)	Interactive image segmentation review	Covers point, stroke, boundary, and region-based interaction	Focuses mainly on image processing rather than robot control
Chen et al. (2024)	RobustSAM for degraded images	Improved segmentation under blur, noise, and low-quality imaging	Additional computational and training requirements
Fan et al. (2026)	Segment and Matte Anything	Improved fine-boundary segmentation and matting	No direct integration with robotic manipulation
Jiang et al. (2025)	Referring segmentation with visual servoing	Connects semantic target selection with real robot control	Depends mainly on language input and eye-in-hand vision
Mei et al. (2026)	RobotSeg	Intuitive object-level interaction and automatic action generation	Segments the robot rather than operator-selected workspace objects
Yin et al. (2026)	Mixed-reality scene-centric robot programming	Robot-oriented foundation segmentation model	Requires mixed-reality hardware and 3D reconstruction
Huang et al. (2024)	Segment and Caption Anything	Combines region masks with semantic descriptions	High computational cost for real-time robot guidance
Kweon et al. (2024)	SAM-assisted weakly supervised segmentation	Reduces the need for dense segmentation annotations	Requires task-specific training and does not control a robot
Lin et al. (2019)	Robot and IIoT communication through OPC UA	Supports standardized industrial data exchange	Does not integrate interactive segmentation or visual servoing
This work	Interactive segmentation with robot-computer communication	Real-time operator interaction, industrial visual servoing, simplified robot guidance	Currently limited to operator-assisted manipulation and the tested workcell

III. MATERIALS AND METHODS

The developed system combines an industrial robot, an external computer, a vision sensor, an interactive segmentation module, and a graphical user interface. The general workflow consists of five main stages. First, the operator receives an image of the robot workspace. Second, the required working region is selected through interactive segmentation. Third, the selected image points are converted into spatial coordinates. Fourth, the generated robot positions are transferred from the computer to the robot controller. Finally, the controller executes the received trajectory while the graphical interface displays the connection state and operation progress. The experimental system was divided into five functional components: the SIASUN industrial robot, the TCP/IP connection interface, the low-level robot-to-computer communication procedure, the segmentation module, and the operator-oriented graphical interface.

A. Robot Description

The experimental platform was based on the SIASUN SR25A-20/1.80 six-axis industrial robot. The manipulator has a nominal payload capacity of 20 kg, a maximum operating radius of approximately 1803 mm, and a repeatability of ±0.05 mm. The robot can be installed in both upright and inverted configurations, which allows it to be integrated into different industrial workcells. The SR25A-20/1.80 has also received European CE certification, indicating its applicability in industrial automation environments. The industrial robot employed in this work is shown in Fig. 1. The six revolute joints provide sufficient flexibility for positioning the end effector at different locations and orientations within the working area. Depending on the installed tool, the robot can be used for material handling, marking, assembly, welding, inspection, and other repetitive industrial operations.

Fig. 1. Image of the Siasun robot SR25A-20.

In the proposed system, the robot was used as an execution platform that receives target positions generated by the external vision and control software. The robot controller stores

Cartesian positions using position registers containing the translational coordinates as well as the orientation components. During operation, the target positions are transferred to these registers and then executed using joint-space or Cartesian motion instructions. The orientation values can either be obtained from previously taught positions or assigned according to the orientation required by the technological process. The main characteristics of the experimental robot are summarized in Table II.

TABLE II. MAIN CHARACTERISTICS OF SIASUN

Parameter	Value
Number of controlled axes	6
Maximum payload	20 kg
Maximum working radius	1803 mm
Position repeatability	0.05 mm
Controller	SIASUN SRC E5
Communication interface used in this study	Ethernet TCP/IP

B. Connection Interface

Communication between the robot controller and the external computer was implemented through an Ethernet network using the TCP/IP protocol. TCP provides connection-oriented and reliable byte-stream transmission, which is suitable for transferring robot coordinates, system states, and control messages between two network devices. The robot controller and the computer were connected to the same local network. Static IPv4 addresses were assigned to both devices, and the subnet mask was configured so that the controller and the computer belonged to the same subnet. Before starting the robot program, the network connection was verified using the controller web interface and basic network diagnostic tools. In the proposed configuration, the external computer operates as a TCP server, while the SIASUN controller acts as a TCP client. The computer continuously listens on a predefined IP address and port. When the corresponding robot instruction is executed, the controller creates a connection to the server and starts the data exchange.

Several practical difficulties were observed during the connection setup. First, an incorrect IP address or subnet mask prevented the devices from detecting each other, even when the physical Ethernet connection was active. Second, the firewall of the computer could block the selected TCP port. Third, the robot controller and the computer processed line endings differently. Therefore, each transmitted message was terminated by a predefined end-of-line character. Fourth, the TCP connection could be closed before the complete coordinate message was received. This required an explicit communication state and message confirmation mechanism.

Another difficulty was related to the difference between TCP packets and application-level messages. A single coordinate string could be divided into several received data blocks, or several short messages could be received together. For this reason, the computer did not treat each *recv()* operation as a complete command. Instead, received bytes were accumulated in a buffer until the selected message delimiter was detected.

The communication procedure was implemented at two levels. The high-level communication and data processing were performed by the computer application, while the low-level connection and motion instructions were executed inside the SIASUN controller program.

At the robot-controller level, the communication sequence consists of creating a TCP connection, sending a request or the current robot state, receiving a response from the computer, converting the received text into a position register, executing the robot movement, and closing or preserving the connection according to the selected operating mode. The simplified communication sequence is expressed as Eq. (1).

$$O_{\text{open}} => S_{\text{send}} => R_{\text{receive}} => C_{\text{convert}} => E_{\text{execute}} \tag{1}$$

where, O represents the establishment of the TCP connection, S is the message transmitted by the controller, R is the coordinate message returned by the computer, C is the conversion from a text string into a robot position, and E represents the execution of the movement command. The controller program uses low-level instructions such as TCPCREATE, TCPSEND, TCPRECV, and TCPCLOSE. The current Cartesian position can be read from the controller and stored in a position register. The register values are then converted into a character string and transmitted to the computer. Similarly, the coordinate string received from the computer is converted back into a position-register format before the movement instruction is executed. A typical transmitted target position has the following structure Eq. (2).

$$P_i = \left(X_i, Y_i, Z_i, R_{\{x,i\}}, R_{\{y,i\}}, R_{\{z,i\}}\right) \tag{2}$$

There is i is the trajectory-point index. A message delimiter is added to the end of each position to allow the receiving program to determine when the complete coordinate message has arrived. For trajectories containing several points, the computer first transmits the total number of positions N. The controller then requests or receives the positions sequentially Eq. (3).

$$T = \{P_1, P_2, \dots, P_N\} \tag{3}$$

Only one point is written into the active position register at each iteration. After the point has been successfully parsed, the robot executes the corresponding motion and proceeds to the next point. This approach avoids storing the complete trajectory in the limited set of controller registers and simplifies the validation of individual positions. The communication protocol also contains acknowledgment messages. After receiving and parsing a valid position, the controller or computer returns an *OK* response. When the coordinate string is incomplete, incorrectly formatted, or outside the expected range, an error response is generated, and robot motion is not started. One of the practical problems was that a position received as an ordinary string could not always be used directly by the robot motion instruction. Although the displayed coordinate values were correct, the controller required the complete internal position-register structure, including coordinate-system and orientation-related fields. Therefore, the received values were inserted into a previously initialized position register rather than being treated as an independent uninitialized position.

C. Segmentation Module

The segmentation module was designed to identify the workpiece or the working region selected by the operator. Instead of requiring the operator to manually specify all trajectory coordinates, the system allows the user to provide a small number of visual prompts, such as positive clicks inside the required object and negative clicks outside it. The initial mask was generated using an interactive segmentation model based on the Segment Anything approach. This type of model accepts points, bounding boxes, or masks as prompts and produces a pixel-level segmentation of the selected image region. However, the general-purpose pretrained model did not always provide stable masks for the industrial scene. The observed difficulties included low contrast between the workpiece and its background, reflected light from metallic surfaces, partial occlusion by the robot end effector, and the presence of visually similar objects. Therefore, the segmentation model was additionally adapted to the target industrial environment. The fine-tuning dataset consisted of images collected directly from the experimental workspace.

The images contained different robot positions, illumination conditions, viewing angles, target locations, and partial occlusions. Each image was annotated with the required working-region mask. Data augmentation was applied through changes in image brightness, contrast, scale, rotation, and geometric position. During fine-tuning, the pretrained visual representation was preserved, while the task-specific segmentation layers were adjusted using the collected industrial images. This reduced the number of training samples required compared with training a segmentation network from the beginning. For an input image *i* and an operator prompt *q*, the segmentation model produces a binary mask Eq. (4).

$$M = f_{\{\theta\}(I,q)} \tag{4}$$

where, $f_{\{\theta\}}$ is the segmentation model with parameters θ, and Eq. (5).

$$\mathrm{M(u,v)} = \begin{Bmatrix} 1, \text{if pixel } (\mathrm{u,v}) \text{belongs to the selected region} \\ 0, \text{otherwise} \end{Bmatrix} \tag{5}$$

The model was optimized using a combined segmentation loss, Eq. (6).

$$\mathrm{L_{seg}} = \lambda_1 \mathrm{L_{BCE}} + \lambda_2 \mathrm{L_{Dice}} \tag{6}$$

Here $\mathrm{L_{BCE}}$is the binary cross-entropy loss, $\mathrm{L_{Dice}}$ is the Dice loss, and $\lambda_1$and $\lambda_2$are weighting coefficients. After obtaining the mask, its contour was extracted and filtered to remove small isolated regions. The operator could accept the mask, add another positive point, or mark an incorrectly segmented region with a negative point. The corrected mask was then used to calculate the required image points. The depth value corresponding to a selected pixel *(u,v)* was used to reconstruct its position in the camera coordinate system Eq. (7).

$$X_c = \frac{(u - c_x)Z_c}{f_x}$$

$$\mathrm{Y_c} = \frac{(\mathrm{v} - \mathrm{c_y})\mathrm{Z_c}}{\mathrm{f_y}}$$

$$\mathrm{Z_c} = \mathrm{Depth(u,v)} \tag{7}$$

Function *Depth(u,v)* is the measured depth, f_x and f_y are the camera focal lengths, and C_x and $C_y$represent the principal point. The three-dimensional camera coordinates were subsequently converted into the robot-base coordinate system Eq. (8).

$$\mathrm{p_r} = \mathrm{R_{cr}} \mathrm{p_c} + \mathrm{t_{cr}} \tag{8}$$

where, $\mathrm{p_c}$is a point in the camera coordinate system, $\mathrm{p_r}$ is the corresponding point in the robot coordinate system, $\mathrm{R_{cr}}$ is the rotation matrix, and $\mathrm{t_{cr}}$ is the translation vector obtained during camera-to-robot calibration.

The novelty of the proposed segmentation module is not limited to the application of an existing segmentation network. The module combines interactive operator prompts, domain-specific fine-tuning, depth-based coordinate reconstruction, and direct generation of executable industrial robot positions. Thus, the segmentation result is not used only for visual recognition but becomes an active component of the robot-control loop.

D. Graphical User Interface

A graphical user interface was developed to integrate camera visualization, interactive segmentation, robot communication, and trajectory execution in a single application. The interface was implemented on the external computer and was designed for operators who may not have experience with the native controller language. The main GUI window contains the following functional areas:

- a live camera-image panel.
- an interactive segmentation panel.
- network-connection parameters.
- robot-motion parameters.
- a communication log.
- a three-dimensional robot-workspace view.
- trajectory-generation and execution controls.

The camera panel displays the current RGB image and allows the operator to select the required working region. Positive and negative points are shown using different markers. After every additional prompt, the segmentation mask is updated and overlaid on the original image. The connection panel contains the robot IP address, server address, TCP port, connection state, and the number of transmitted trajectory points. The connection status is displayed as disconnected, waiting, connected, transmitting, executing, completed, or error. Before starting robot motion, the interface checks whether the camera is available, a valid segmentation mask has been generated, the coordinate transformation has been loaded, and the TCP server is active. This prevents the trajectory from being transmitted when one of the required system components is unavailable.

The three-dimensional view visualizes the generated trajectory in the robot coordinate system. It is used to verify that the calculated points are located inside the expected working region. The operator can inspect the trajectory before transmitting it to the controller. The GUI does not directly control the robot servo drives. Instead, it generates and transmits target positions, while the final motion execution remains inside the controller. This architecture preserves the controller-level motion and safety functions while allowing computationally demanding vision and segmentation operations to be performed on the external computer. Fig. 2 presents the overall architecture of the developed system.

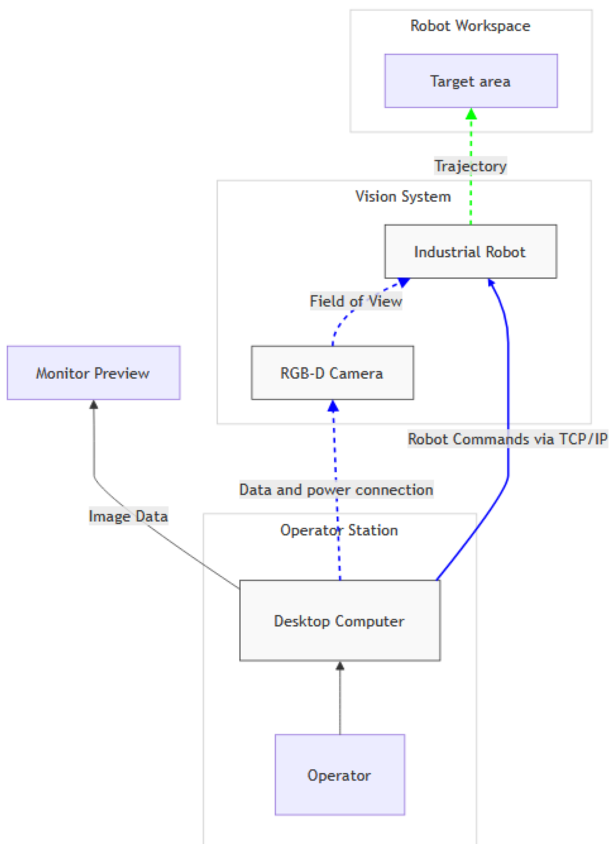

Fig. 2. Overall architecture of the proposed visual servoing framework for industrial robot control.

The image-processing and segmentation operations are performed on the computer, while the controller is responsible for the interpretation and execution of the received positions.

IV. Results

The proposed visual servoing framework was evaluated in an industrial workspace using a collaborative robot, an RGB-D camera, and the fine-tuned Segment Anything Model (SAM). The objective of the experiment was to verify whether the interactive segmentation approach can reliably identify operator-selected regions and generate stable target coordinates for subsequent robot motion. After fine-tuning on the collected industrial images, the segmentation model successfully distinguished the working area from the surrounding background despite variations in illumination, reflections from metallic surfaces, and partial occlusions. Unlike conventional threshold-based segmentation, the proposed approach preserved object boundaries and remained stable under different viewing conditions. The operator selected the desired working region using a single prompt, after which the SAM model generated the corresponding segmentation mask. The resulting mask was subsequently converted into a three-dimensional point using the depth information provided by the RGB-D camera. Fig. 3 presents examples of interactive segmentation obtained during the experiments.

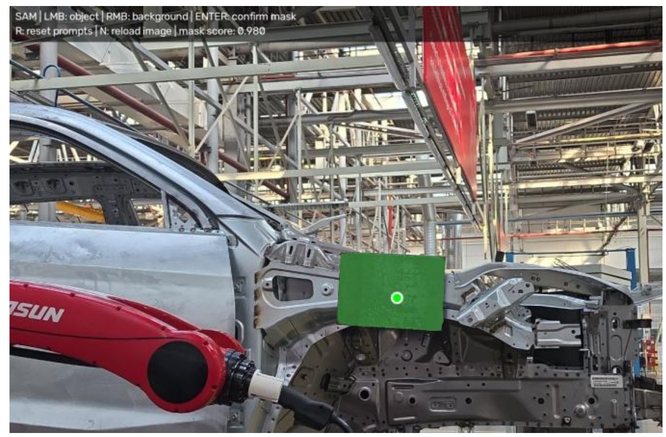

Fig. 3. Image of target segmentation.

This point was then transformed into the robot coordinate system through the previously obtained calibration matrix. The confidence score produced by the segmentation model remained consistently high throughout the experiments. Table III demonstrates the segmentation results. The average confidence on the target region reached approximately 91% for the selected industrial regions, indicating that the detected masks closely matched the intended target areas. The confidence score produced by the model reached 98% for the target region, while the robot arm and workspace were segmented with confidence scores of 90% and 85%, respectively, as shown in Fig. 4. High confidence values were observed even for objects with low texture and moderate illumination changes, demonstrating the robustness of the proposed approach.

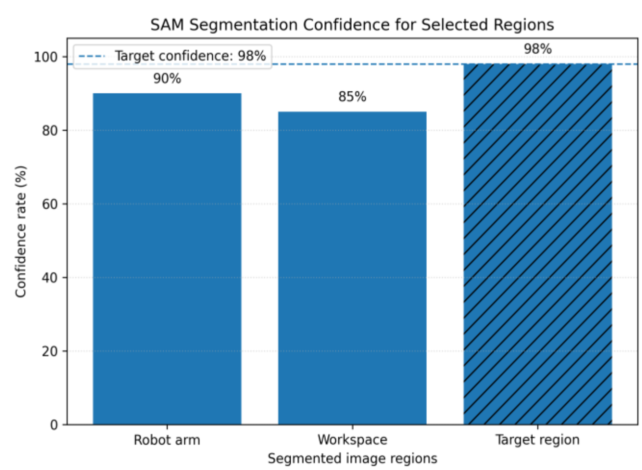

Fig. 4. Segmentation results.

TABLE III. SEGMENTATION PERFORMANCE OF THE PROPOSED METHOD

Metric	Value
Average confidence score target	91%
Average inference time	46 ms
Successful segmentations	95%
Failed detections	5%

The average inference time remained below 50 ms, allowing the segmentation process to operate close to real time. Consequently, the delay introduced by the vision subsystem did not noticeably affect the overall robot response during the visual servoing process. After the operator selected the desired region, the segmentation mask was generated by SAM, transformed into three-dimensional coordinates using the depth image, converted into the robot reference frame through hand-eye calibration, and finally transmitted to the robot controller. Throughout the experiments, the robot successfully reached the designated target positions without requiring manual correction. Overall, the experimental results demonstrate that integrating interactive segmentation with robot-computer communication provides a reliable method for industrial visual servoing. The high segmentation confidence, low inference latency, and stable coordinate transformation collectively confirm the applicability of the proposed system for practical industrial manipulation tasks.

V. DISCUSSION

The experimental findings demonstrate that integrating interactive image segmentation, RGB-D perception, coordinate transformation, and robot–computer communication can provide a practical visual servoing solution for flexible industrial manipulation. The proposed framework achieved an average target-region segmentation confidence of approximately 91%, a successful segmentation rate of 95%, and an average inference time of 46 ms. These results are particularly relevant because the segmentation output is not treated merely as a computer-vision result; instead, it is transformed into three-dimensional coordinates and subsequently transmitted to the industrial robot controller for physical execution. This integration reduces dependence on manually taught positions and enables operators to redefine task-relevant regions directly from camera images.

The obtained results should also be considered within the broader development of machine learning-based industrial robotics. Conventional visual servoing depends strongly on explicitly defined image features, calibration models, geometric relationships, and control rules. Machine learning provides an alternative in which relevant visual representations or control relationships can be learned from data. Learning-based robotic systems can therefore improve their response to variations in object appearance, workspace configuration, illumination, and operating conditions. For the proposed framework, machine learning-based solutions could complement the existing coordinate-transformation pipeline by learning residual calibration errors, estimating target pose, predicting segmentation reliability, or selecting appropriate manipulation points. Such capabilities would be particularly beneficial when extending the current system from operator-assisted manipulation toward increasingly autonomous industrial operation [30].

Deep learning offers additional advantages because convolutional neural networks, vision transformers, and hybrid architectures can extract hierarchical representations directly from industrial images. Rather than relying on handcrafted features, deep learning-based solutions can simultaneously learn texture, shape, boundary, and contextual information relevant to robotic perception [31]. This characteristic is important in the experimental environment considered in this study, where metallic reflections, partial occlusions, low-texture regions, and illumination variations may affect conventional image-processing algorithms. The fine-tuned SAM used in the proposed framework follows this broader transition toward data-driven perception, while domain-specific adaptation allows the general visual representation to be adjusted to industrial imagery. The adopted fine-tuning strategy also reduces the need to train a segmentation network completely from scratch.

Another important direction concerns the combination of segmentation with automatic object detection and pose estimation. Modern machine learning algorithms can identify candidate objects before segmentation, whereas deep neural networks can estimate object categories, orientations, grasping points, and spatial relationships [32]. Integrating these algorithms with the present architecture could progressively reduce the number of operator prompts required for manipulation. Nevertheless, complete automation is not necessarily desirable in every manufacturing scenario. Operator-guided segmentation retains human supervision and provides an intuitive mechanism for handling previously unseen components, unusual production configurations, and low-volume manufacturing tasks. Human-in-the-loop machine learning could therefore provide a useful compromise between fully manual robot programming and completely autonomous perception [33].

The computational performance of the proposed system is also significant for industrial deployment. An average inference time of 46 ms indicates that segmentation can operate close to real time without introducing a substantial delay into the visual servoing workflow. However, increasing the complexity of future deep learning models may create a trade-off between perception accuracy and response latency. Lightweight CNNs, compact vision transformers, knowledge distillation, pruning, quantization, and hardware-aware neural-network optimization could therefore be investigated to maintain low latency while improving robustness [34]. Edge-based inference would further reduce dependence on remote computational resources and could improve predictability for time-sensitive industrial tasks [35].

The modular separation between perception and robot execution represents another practical strength. Computationally demanding segmentation and image processing are performed on the external computer, whereas motion execution and controller-level safety remain within the industrial robot controller. This architecture makes it possible to update machine learning and deep learning algorithms without substantially modifying the robot hardware. It also creates opportunities for integrating reinforcement learning, particularly for trajectory optimization, grasp selection, and adaptive motion planning. Reinforcement learning-based algorithms could use visual observations, robot states, and task outcomes to optimize manipulation policies while conventional controller constraints preserve safe execution [36].

Several limitations should nevertheless be acknowledged. The current evaluation was performed within the tested industrial workcell and primarily considered operator-assisted target selection. Consequently, the reported segmentation performance does not establish equivalent robustness across different factories, cameras, robots, object materials, or highly dynamic environments. Domain shift remains an important challenge for deep learning-based industrial vision because changes in illumination, camera characteristics, backgrounds, and object appearance can reduce model generalization. Future experiments should therefore include larger multi-condition datasets, cross-workcell validation, more extensive quantitative localization measurements, and comparisons with conventional machine learning, CNN-based segmentation, transformer-based segmentation, and other foundation-model approaches [37].

Finally, future development could combine multi-camera perception, machine learning-based uncertainty estimation, automatic grasp planning, deep learning-based object understanding, and collaborative multi-robot coordination. Multiple viewpoints could reduce occlusion, while uncertainty-aware algorithms could prevent robot motion when segmentation or coordinate estimation is unreliable. Multi-robot systems could additionally share a centralized perception model and exchange task information through the existing communication architecture [38]. Overall, the results suggest that the proposed framework provides a suitable foundation for progressing from interactive visual servoing toward more autonomous and adaptive industrial robotics, while retaining the operator supervision and controller-level safeguards required for practical manufacturing deployment.

VI. Conclusion

The proposed visual servoing framework demonstrates that interactive segmentation can be effectively integrated with industrial robot control through a unified robot-computer communication architecture. Unlike conventional programming methods that require predefined robot trajectories, the proposed approach enables an operator to specify the desired working region directly from the camera image, while the system automatically converts the selected region into robot motion commands. Experimental evaluation confirmed reliable communication between the computer and the industrial robot and showed that the fine-tuned segmentation model achieved average confidence values of up to 91% for the target industrial region, providing sufficiently accurate visual information for manipulation tasks. The proposed framework is modular and can be adapted to different industrial robots and vision sensors with only minor modifications to the communication interface. Future work will focus on improving robustness under varying illumination conditions, extending the method to three-dimensional object understanding, integrating automatic grasp and path planning algorithms, and validating the framework in more complex industrial scenarios involving multiple objects and dynamic environments. Overall, the presented system represents a practical step toward more intuitive, flexible, and intelligent visual servoing solutions for modern industrial robotic applications.

Acknowledgment

This work was supported by the Science Committee of the Ministry of Higher Education and Science of the Republic of Kazakhstan within the framework of research program “BR24992947 – Development of Robots, Scientific, Technical, and Software for Flexible Robotization and Industrial Automation (RPA) in Automotive Industrial Enterprises in Kazakhstan Using Artificial Intelligence”.

References

[1] Nof S. Y., Ed. Handbook of Industrial Robotics, 2nd ed. New York, NY, USA: John Wiley & Sons, 1999.

[2] Siciliano B., and Khatib O., Eds. Springer Handbook of Robotics, 2nd ed. Cham, Switzerland: Springer, 2016.

[3] J. J. Craig, Introduction to Robotics: Mechanics and Control, 4th ed., Global ed. Harlow, UK: Pearson, 2017, ISBN 978-1-292-16493-9.

[4] Hutchinson S., Hager G. D., and Corke P. I., "A tutorial on visual servo control," IEEE Transactions on Robotics and Automation, vol. 12, no. 5, pp. 651-670, Oct. 1996.

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

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

[7] Omarov, B., Batyrbekov, A., Dalbekova, K., Abdulkarimova, G., Berkimbaeva, S., Kenzhegulova, S., ... & Omarov, B. (2020, December). Electronic stethoscope for heartbeat abnormality detection. In International Conference on Smart Computing and Communication (pp. 248-258). Cham: Springer International Publishing.

[8] Omarov, B., Tursynova, A., & Uzak, M. (2023). Deep learning-enhanced Internet of medical things to analyze brain computed tomography images of stroke patients. International Journal of Advanced Computer Science and Applications, 14(8).

[9] Raghunath, M. P., Deshmukh, S., Chaudhari, P., Bangare, S. L., Kasat, K., Awasthy, M., ... & Waghulde, R. R. (2025). PCA and PSO based optimized support vector machine for efficient intrusion detection in internet of things. Measurement: Sensors, 37, 101806.

[10] Brohan A., et al., "RT-1: Robotics Transformer for Real-World Control at Scale," in Proceedings of Robotics: Science and Systems (RSS), Daegu, Republic of Korea, 2023.

[11] Liu, S., Zeng, Z., Ren, T., Li, F., Zhang, H., Yang, J., Jiang, Q., Li, C., Yang, J., Su, H., Zhu, J., & Zhang, L. (2024). Grounding DINO: Marrying DINO with Grounded Pre-training for Open-Set Object Detection. Lecture Notes in Computer Science, 38–55.

[12] Lasi H., Fettke P., Kemper H. G., Feld T., and Hoffmann M., "Industry 4.0," Business & Information Systems Engineering, vol. 6, no. 4, pp. 239-242, Aug. 2014.

[13] Z. Liu, Q. Liu, W. Xu, L. Wang, and Z. Zhou, “Robot learning towards smart robotic manufacturing: A review,” Robotics and Computer-Integrated Manufacturing, vol. 77, Art. no. 102360, 2022.

[14] J. Wu, Z. Jin, A. Liu, L. Yu, and F. Yang, “A survey of learning-based control of robotic visual servoing systems,” Journal of the Franklin Institute, vol. 359, no. 1, pp. 556-577, 2022. DOI: 10.1016/j.jfranklin.2021.11.009.

[15] G. Lai, A. Liu, W. Yang, Y. Chen, and L. Zhao, “Uncalibrated adaptive visual servoing of robotic manipulators with uncertainties in kinematics and dynamics,” Actuators, vol. 12, no. 4, Art. no. 143, 2023. DOI: 10.3390/act12040143.

[16] R. Muthusamy et al., “Neuromorphic eye-in-hand visual servoing,” IEEE Access, vol. 9, pp. 55853-55870, 2021

[17] M. T. Shahria et al., “A comprehensive review of vision-based robotic applications: Current state, components, approaches, barriers, and potential solutions,” Robotics, vol. 11, no. 6, Art. no. 139, 2022.

[18] A. Kirillov et al., “Segment Anything,” in Proceedings of the IEEE/CVF International Conference on Computer Vision, Paris, France, 2023, pp. 4015-4026.

[19] H. Ramadan, C. Lachqar, and H. Tairi, “A survey of recent interactive image segmentation methods,” Computational Visual Media, vol. 6, pp. 355-384, 2020

[20] W.-T. Chen, Y.-J. Vong, S.-Y. Kuo, S. Ma, and J. Wang, “RobustSAM: Segment Anything robustly on degraded images,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 2024, pp. 4081-4091.

[21] Z. Fan, X. Li, T. Biswas, K. Nag, and K. Achan, “Segment and matte anything in a unified model,” Proceedings of the AAAI Conference on Artificial Intelligence, vol. 40, no. 5, pp. 3813-3821, 2026.

[22] R. Yao, G. Lin, S. Xia, J. Zhao, and Y. Zhou, “Video object segmentation and tracking: A survey,” ACM Transactions on Intelligent Systems and Technology, vol. 11, no. 4, pp. 1-47, 2020.

[23] C. Jiang, A. Wang, and M. Jägersand, “Robot manipulation in salient vision through referring image segmentation and geometric constraints,” in Proceedings of the 2025 IEEE International Conference on Robotics and Automation, Atlanta, GA, USA, 2025, pp. 11141-11147

[24] H. Mei, Q. Huang, H. Ci, and M. Z. Shou, “RobotSeg: A model and dataset for segmenting robots in image and video,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Denver, CO, USA, 2026, pp. 14346-14356.

[25] Y. Yin, J. Fan, A. Liu, and P. Zheng, “A mixed reality-assisted scene-centric robot programming approach for human-robot collaborative manufacturing,” Robotics and Computer-Integrated Manufacturing, vol. 98, Art. no. 103146, 2026.

[26] M. Tzampazaki, C. Zografos, E. Vrochidou, and G. A. Papakostas, “Machine vision: Moving from Industry 4.0 to Industry 5.0,” Applied Sciences, vol. 14, no. 4, Art. no. 1471, 2024.

[27] X. Huang et al., “Segment and caption anything,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 2024, pp. 13405-13417.

[28] H. Kweon and K.-J. Yoon, “From SAM to CAMs: Exploring Segment Anything Model for weakly supervised semantic segmentation,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 2024, pp. 19499-19509.

[29] H.-I. Lin and Y.-C. Hwang, “Integration of robot and IIoT over the OPC Unified Architecture,” in Proceedings of the 2019 International Automatic Control Conference, Keelung, Taiwan, 2019, Art. no. 9024732.

[30] Levine S., Pastor P., Krizhevsky A., Ibarz J., and Quillen D., "Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection," The International Journal of Robotics Research, vol. 37, no. 4-5, pp. 421-436, Apr. 2018.

[31] Omarov, B., Omarov, B., Rakhymzhanov, A., Niyazov, A., Sultan, D., & Baikuvekov, M. (2024). Development of an artificial intelligence-enabled non-invasive digital stethoscope for monitoring the heart condition of athletes in real-time. Retos, 60, 1169-1180.

[32] Lu Y., "Industry 4.0: A survey on technologies, applications and open research issues," Journal of Industrial Information Integration, vol. 6, pp. 1-10, Jun. 2017.

[33] Omarov, B., Baikuvekov, M., Sultan, D., Mukazhanov, N., Suleimenova, M., & Zhekambayeva, M. (2024). Ensemble approach combining deep residual networks and BiGRU with attention mechanism for classification of heart arrhythmias. Computers, Materials & Continua, 80(1), 341-359.

[34] Xaud, M. F., From, P. J., & Leite, A. C. (2025). Robust Visual Servoing and CNN-Based Thermal Imaging for Sugarcane Row Following with a Skid-Steering Mobile Robot. IEEE Access.

[35] Bajpai, A., & Srivastava, S. (2026). AI-Driven Real-Time Decision Making at the Edge: Overcoming Latency, Bandwidth, and Scalability Challenges for Smarter Data-Intensive Applications in Healthcare, Manufacturing, and Smart Cities. In AI at Edge: Transforming Edge Networks with Computational Intelligence (pp. 1-19). Cham: Springer Nature Switzerland.

[36] Khan, F., Feng, W., Wang, Z., Huang, T., Liu, X., Cui, Y., & Wang, W. (2026). Safe reinforcement learning for vision-based robotic manipulation in human-centered environments. International Journal of Intelligent Robotics and Applications, 10(1), 411-435.

[37] Omarov, B. (2025). Deep learning in biomedical image and signal processing: a survey. Computers, Materials & Continua, 85(2), 2195-2253.

[38] Boralessa, K., Adama, D., Lotfi, A., & Ferreira, J. F. (2026). Cooperative Perception for Multi-Robot Systems in Natural Outdoor Applications–A Survey. Journal of Intelligent & Robotic Systems.