CIRP Annals - Manufacturing Technology 73 (2024) 13–16

Contents lists available at ScienceDirect

CIRP Annals - Manufacturing Technology

journal homepage: https://www.editorialmanager.com/CIRP/default.aspx

Vision AI-based human-robot collaborative assembly driven by autonomous robots

Sichao Liu[a], Jianjing Zhang[b], Lihui Wang (1)[a,*], Robert X. Gao (1)[b]

[a] Department of Production Engineering, KTH Royal Institute of Technology, Stockholm, Sweden
[b] Department of Mechanical and Aerospace Engineering, Case Western Reserve University, Cleveland, OH, USA

ARTICLE INFO

Article history:
Available online 23 April 2024

Keywords:
Robot
Assembly
vision AI

ABSTRACT

Autonomous robots that understand human instructions can significantly enhance the efficiency in human-robot assembly operations where robotic support is needed to handle unknown objects and/or provide on-demand assistance. This paper introduces a vision AI-based method for human-robot collaborative (HRC) assembly, enabled by a large language model (LLM). Upon 3D object reconstruction and pose establishment through neural object field modelling, a visual servoing-based mobile robotic system performs object manipulation and navigation guidance to a mobile robot. The LLM model provides text-based logic reasoning and high-level control command generation for natural human-robot interactions. The effectiveness of the presented method is experimentally demonstrated.

1. Introduction

Autonomous robot-driven collaborative assembly promotes human-robot interaction and control for on-demand robot assistance, especially for personalised product assembly scenarios [1]. These scenarios involve operations that cannot be predefined but need to be dynamically adapted to. While it is natural for humans to instruct a robot to pick up a spare part when a component is broken or missed during assembly operations, having the robot to not only autonomously recognise the needed part but also parse and decompose human instructions into executable actions to provide assistance has remained a challenge. Establishing the 3D model and 6D pose of an object is the first step toward part recognition and manipulation [2]. In general, the existing methods rely on available CAD models and category /instance-level prior knowledge or known camera poses to create the object pose. This is unpractical for handling objects that are previously unknown [3]. In recent years, vision artificial intelligence (AI) has been introduced for 3D reconstruction of unknown objects and pose tracking [4]. Recently, neural rendering has been investigated for 3D modelling of unknown products [5], however, to achieve on-demand assistance, pose tracking will have to be integrated with 3D rendering and reconstruction techniques.

To assist a robot in understanding human language commands for assembly, natural language processing (NLP) models enable sentence parsing for cause-and-effect analysis [6]. However, traditional NLP models lack the ability for assembly contextual understanding and high-level text instruction analysis. Recently, large language models (LLMs) have demonstrated the capabilities of text understanding and reasoning [7]. As an example, Figure 01 humanoid robots powered by OpenAI's visual-language models can converse, reason and plan their actions as they work [8]. Also, leveraging LLMs in high-level planning, robot manipulation and code generation was investigated [9,10], but without exploring logic reasoning behind text instructions. With understanding of the text commands, autonomous mobile robots (AMRs) supported by vision and navigation capabilities can achieve motion control, object detection, and manipulation when executing assembly tasks [11]. These functions are critical to reliable assembly operations to enable handling the right objects in the right way [12].

This paper presents a vision AI-based HRC assembly technique supported by an LLM and AMR. A neural object field-based model is presented for accurate 3D reconstruction and 6D pose estimate of objects. The model enables a visual servoing-based autonomous mobile robotic system with object mapping capability to navigate around the assembly environment for object detection, tracking and manipulation. Finally, LLM-driven logic reasoning of text instructions and high-level robot control commands is presented for natural human-robot interactions in assembly.

2. 3D modelling and pose estimate of unknown objects

2.1. Vision AI-based HRC assembly

As shown in Fig. 1, the vision AI-based HRC assembly starts with RGB-D video collection of an object (e.g., a valve cover) along scanning paths, with the output being object frames and masks (a video frame includes a colour and a depth image). The frames and masks serve as the input for training a network to build the 3D model of the object with an optimised pose. The object is subsequently detected by a camera-driven visual servoing system installed on the AMR. Separately, a laser scanner (Lidar) creates a simultaneous localisation and mapping (SLAM) map of the assembly environment along the moving path of the robot, enabling it to navigate safely around the assembly environment. Since the robot does not know initially what objects to be acted upon, object mapping with labelling is taken as landmarks. To control the robot for task execution, new capabilities of the LLM are explored to reason and extract control logic steps behind text instructions issued by a human operator. Finally, high-level control commands with vocabulary-based object indexing and mapping are used for the robot motion control and assembly task execution.

2.2. Neural representation of unknown objects

The goal of neural representation of an unknown object is to build its optimal pose estimate for robotic manipulation when the CAD model and instance-

* Corresponding author.
E-mail address: lihui.wang@iip.kth.se (L. Wang).

https://doi.org/10.1016/j.cirp.2024.03.004

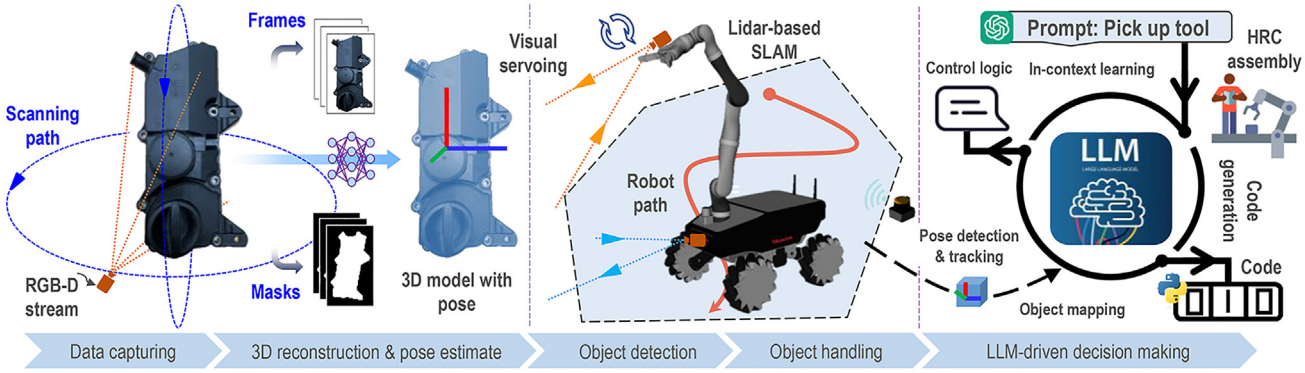

Fig. 1. Workflow of the vision AI-based HRC assembly process.

level prior information of the object as well as camera poses are not available [3]. As shown in Fig. 2, four modules have been developed to realise 3D reconstruction and 6D pose of the unknown objects [4]. Specifically, Module ❶ receives RGB-D video streams of an object from a depth camera and produces object frames and masks by using a segmentation network. Subsequently, pixel-wise dense match features between the current and previous frames, together with their masks, are extracted to generate a coarse pose of the current frame.

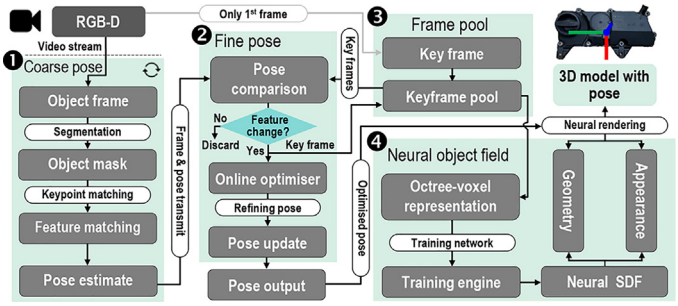

Fig. 2. Workflow of 3D reconstruction and 6D pose for unknown objects.

Meanwhile, the current frame and its coarse pose are transmitted to Module ❷ to perform pose comparison with a set of key frames provided by Module ❸, which is a frame pool storing frames with informative object features and adds the first frame to set a canonical coordinate system for next frames. If significant feature changes between the current frame and existing key frames in the pool are detected, online pose graph optimisation is performed to refine and update the pose, and the current frame is taken as a key frame and added into the frame pool. Otherwise, the current frame is discarded. By iterating each frame, key frames of all the frames and their pose estimates are obtained. Given that real-time neural processing of all the frames takes significant computational resources, only key frames are stored in Module ❸, while other frame information is discarded. Next, Module ❹ receives all the posed key frames from Module ❸ as inputs to the neural object field [5]. The training network learns to accumulate information into a consistent 3D representation that captures both the geometry and appearance of the object by using a neural signed distance function (Neural SDF). Finally, the 3D model of the object with a 6D pose is built for robotic manipulation.

2.3. Object reconstruction and manipulation with 6D pose

Fig. 3 shows the results of 3D reconstruction and 6D pose establishment of a valve cover with complex and textureless surfaces, based on its photographic image that serves as the ground truth. Here, RGB-D video streams of the valve cover are collected by a RealSense D435 camera with a 640×480 resolution at a sampling rate of 30 Hz. An object pose (P_k) is calculated by Eq. (1) [4], where k is the index of the frames. A coarse pose ($\tilde{P}_k$) of the object is computed between the frames $F(k)$ and $F(k-1)$, taking colour (I_c) and depth images (I_d), their mask (M_F), and intrinsic parameters (C) of the camera as inputs. Then, the frame $F(k)$ with its coarse pose ($\tilde{P}_k$) and a set of N key frames F_N^K from the frame pool are provided to the online pose optimiser f to refine the pose (P_k) if significant feature differences are detected. Meanwhile, the frame $F(k)$ as a key frame is added into the frame pool. Finally, all the key frames F^K in the pool are used to learn the neural object field (NF) depicted by Eq. (2), which is rendered by an object representation function (R). The function takes object's geometry (G) and appearance (A) as inputs to construct 3D shape and appearance of the object while adjusting the pose of key frames. In this study, a recorded video with 708 frames and 229 key frames are used to obtain the 3D model and pose of a valve cover, implemented in a CUDA environment. The outcome of the 3D reconstruction of the valve cover is shown in Figs. 3(b) and (c). The mesh models rendered by point clouds (front & back) provide precise 3D neural rendering with detailed representations of the object's complex structure (i.e., reflected by a red circle in Fig 3(c)).

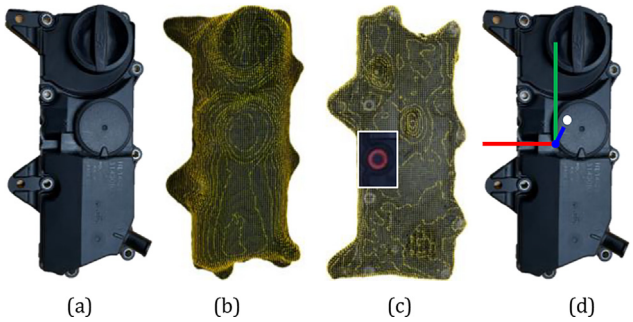

Fig. 3. 3D reconstruction and pose estimate for an object (valve cover): (a) object image as ground truth; (b) & (c): mesh models rendered by point cloud (front & back); (d) object's 6D pose with a grasping point (white dot).

$$P_k = f\left(\tilde{P}_k(I_c, I_d, M,\ C), F(k), F_N^K\right)\ :k\in\mathbb{N},\ P_0 = null \tag{1}$$

$$NF = R(G, A, F^K,\ k)\ :k\in\mathbb{N} \tag{2}$$

With the 3D model created, its fine 6D pose is built simultaneously as shown in Fig. 3(d), which provides the location and orientation of the object as inputs to robotic grasping. Since the object's pose centre is computed from the visible point cloud and may not represent an appropriate grasping point, the centre of its re-defined coordinate frame, which is created by taking the centre and oriented box of the mesh model of the object's surface structure model and geometry, is selected as the grasping point (indicated by a white dot in Fig. 3 (d)). Finally, object's 6D pose with a proper grasping point is identified and tacked for object grasping.

3. LLM-driven assembly execution

3.1. SLAM map for robot motion control and object mapping

An AMR assisting humans in assembly tasks (e.g., material handling) needs to know where the tasks are to be executed in the workspace, what tools/parts are needed, and when to deliver them to the operator at what spatial coordinates. By using Lidar data of scanning work environments, a Lidar-based SLAM system is developed to enable the robot in building a spatial map and localise itself on the map. Combining the robot's position and pose with the location data of the object, the robot will know the obstacle's spatial location and its geometrical profile, as illustrated in Fig. 4.

To further define the object properties (e.g., table vs shelf) and assist the robot in understanding text-based input commands (e.g., place a *tool* on the *table*), object indexing and mapping from an RGB-D camera (as shown in the inset at the upper-right corner of Fig. 4) is performed to associate real objects

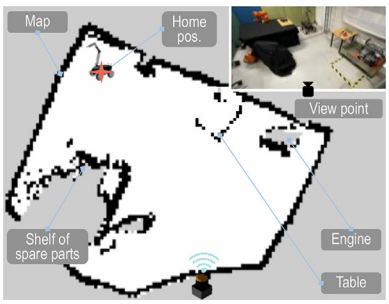

Fig. 4. Lidar-based SLAM map for robot navigation and object indexing.

(e.g., table, engine, and shelf) with the detected obstacles, which serve as the landmarks for robot navigation. Since the home position of the robot is defined by the centre of the robot coordinate system at the initial time-stamp (t_0) on the map (see Fig. 4), the specifics of the objects are known to the robot for indexing.

3.2. Visual servoing-based autonomous mobile robot system

For motion control of the mobile robot in object manipulation, a visual servoing-based closed-loop control scheme is developed. As shown in Fig. 5, two cameras are installed on the robot, with the top one for observing assembly operation whereas the bottom one for workspace scanning at the ground level to assist the robot in navigation. The 6D pose of the target object (Obj) is built by a pose estimate algorithm, once the object is detected by the bottom camera. Meanwhile, relative position and orientation of the object in the robotic coordinate system (i.e., robot odometry (Odom)) are continuously calculated by the Obj2Odom transformation as the robot moves to the object. The position information is simultaneously sent to the robot controller and the top camera for robot navigation and object pose calculation in the Odom based on the kinematics of the robot (i.e., End2Odom transformation).

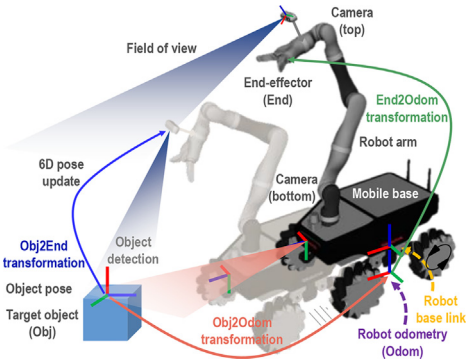

Fig. 5. Visual servoing-based object manipulation and robot control.

From the top camera view, the object pose is tracked in real time and passed to an ROS (robot operating system)-based motion planner that generates the robot trajectories. The robot arm is controlled to grasp the object at the customised grasping point. This forms a closed-loop of visual servoing to establish object-camera-robot data streams for robot control and object manipulation. It further provides the capability of handling dynamic situations (e.g., moving objects).

3.3. LLM-driven logic reasoning of texts and command generation

To assist the robot in understanding text-based human commands for assembly, new capabilities of the LLM are investigated to build logic reasoning behind text input as commands and generate high-level control codes for natural human-robot interactions (see Fig. 6). Prompting with exact protocols is a crucial component to generate the desired behaviours in LLMs. It starts with creating a scenario content that describes assembly scenarios, components, and possible assembly schemes in the form of prompts. This enables the LLM to gain an initial understanding of assembly situations and provide uniform format outputs through in-context examples. Scenarios with a higher level of detail and grounding with additional contexts will result in more accurate extraction of context-aware control logic. After in-context learning, the fine-tuned model is saved and used to interpret text commands and obtain task plans in the form of code generation by skills grounding, with human

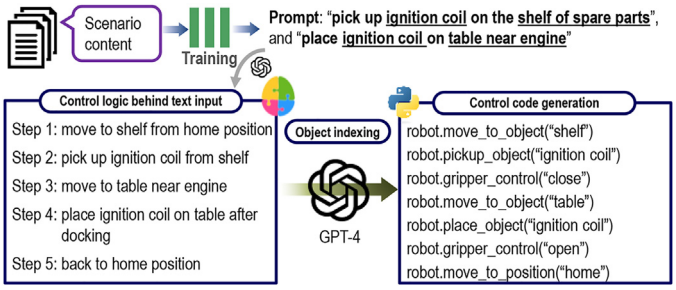

Fig. 6. LLM-driven logic reasoning of texts and high-level control codes.

intervention as necessary. Here, the operator's audio commands are served as input to a voice transcriber that outputs texts, executed by GPT-4.

As an example, if the text commands are given by 'pick up *ignition coil* on the *shelf of spare parts*' and then 'place *ignition coil* on *table near engine*' or similar texts, it is divided into small actionable steps for the robot to execute. Firstly, the control logic is reasoned, given the text into the fine-tuned model, and the text objects (e.g., ignition coil) are extracted by language reasoning that associates object names with text descriptions and categories. Next, the control logic is formulated as 1) '*move to shelf of spare parts*', 2) '*pick up ignition coil*', 3) '*move to table*', 4) '*place ignition coil on table*'. The outcome is depicted by the five steps in Fig. 6 (left side).

The extracted text objects are subsequently used for object indexing of high-level control codes, where the format is defined by the scenario content. The control code for object manipulation is represented by 'robot.move_to_object' or 'robot.pickup_object', which is determined by the types of tasks and manipulation actions. The parts that the robot manipulates are defined as the indexed objects in the form of vocabularies. As shown in Fig. 6, GPT-4 performs object indexing of the control logic extracted from the text and defines 'shelf' as the object of the control codes (robot.move_to_object("shelf")). These high-level control codes include physical information of the indexed object (e.g., position and height) and are then mapped to low-level robot control commands for robot movement and gripper control. As a result, the robot knows where the object is located in the robotic coordinate system, and finds a path towards the target.

4. System implementation

The performance of the developed system is evaluated in an experiment of component assembly of an engine block. As shown in Fig. 7, an operator instructs a mobile robot (Robotnik Summit-XL Gen with a Kinova arm) to handover spare parts for replacing a broken ignition coil and a missed valve cover when assembling a fuel injector tube and ignition coil A in parallel, followed by securing the components. The system is controlled by an open architecture ROS-integrated computer connected to OpenAI GPT-4. The RGB-D steams for visual servoing are taken by a top-down camera system.

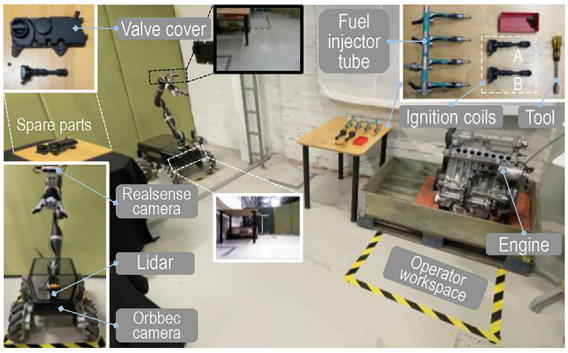

Fig. 7. Experimental setup.

The HRC assembly process is shown in Fig. 8 and includes six control steps. It starts with assembly component check given the assembly plan in Step ①, where the operator identified a broken ignition coil and a missed valve cover. Next, the operator's voice instruction to the robot, "pick up *ignition coil* on the *shelf of spare parts*", triggers the pre-trained LLM, and then it performs logic reasoning of the text and generates control steps, followed by outputting uniformed control codes with the indexed object as shown in Fig. 6. The indexed vocabulary of '*shelf*' is mapped to the built SLAM map (in Fig. 4), to load its location to the robot for docking to the object ('shelf'). The visual

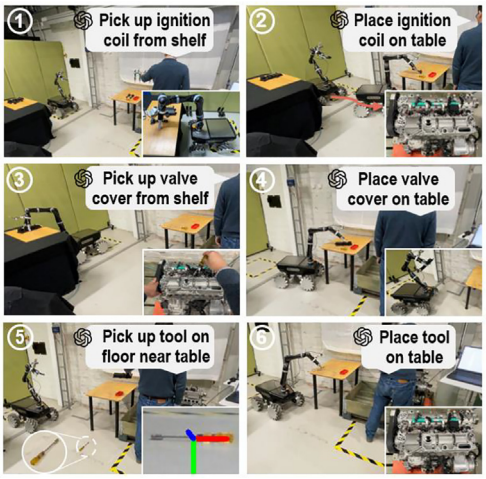

Fig. 8. Assembly process and control steps.

servoing system of the robot simultaneously recognises the ignition coil and builds its pose by calling the part recognition and pose estimate algorithms as depicted in Fig. 2. It should be noted that no rigidly defined format is needed for the operator to define a control command for the robot. What is shown in the case study is a representative example only.

The robot arm is finally controlled to grasp and handover the spare part. The text input 'place *ignition coil* on *table*' indexes the table location, and navigates the robot to the table for placing the object by following an ROS-based motion planner-generated path (marked by a red line) in Step ②. When executing these two steps, the operator works in parallel to finish the assembly of the fuel injector tube and ignition coil A as shown in the inset. Step ③ is to control the robot to handover the missed valve cover by a voice command of 'pick up *valve cover* on the *shelf of spare parts*'. In Step ④, the robot is instructed to 'place *valve cover* on *table*'.

While assembling the valve cover in Step ⑤, the operator accidentally dropped the tool on the floor, which is an unexpected change, and asked the robot to 'pick up *tool* on the *floor* near *table*'. The indexed vocabularies of 'table' and 'floor' to the LLM generate high-level control codes to navigate the robot towards the table and call the bottom camera system to recognise '*tool*' and estimate its pose (as shown in the inset). During navigation towards the tool's position, the robot arm is controlled to adjust its position to make the tool visible from the top camera when it is out of the field of view of the bottom camera. The built pose of the recognised tool is sent to the robot arm to pick it up. During the assembly process, the visual servoing system tracks the tool at a sampling rate of 30 Hz, ensuring a robust operation under dynamic situations.

In Step ⑥, the input text of 'place *tool* on *table*' only indexes the table's position on the map. The generated commands control the robot in *placing* actions triggered by a preset stop distance of 150 mm of the mobile base between the bottom camera and the table for collision-free robot navigation. The valve cover assembly operation is completed when the operator has secured the objects. The role of geometry in HRC assembly is comprehensively reflected in the presented study, including control, manipulation, etc. Handling different tasks during the process illustrates the model's ability in generalisation to new tasks in assembly.

5. Conclusions and future work

This paper presents a novel vision AI-based approach to human-robot collaborative assembly supported by an autonomous mobile robot and a large language model implemented in GPT-4. Specific contributions of this work include:

- Developed a method for the neural 3D representation and 6D pose estimate of objects that are unknown to the robot in advance to enable visual servoing-based object manipulation to assist in the assembly operation.
- Developed a large language model-driven reasoning method for text-based assembly task description and robot control commands interpretation and execution.
- Demonstrated effective communication and interaction between a human operator and an autonomous mobile robot through natural language input.

The developed techniques expand the boundary of human interactions with robots beyond the traditional "silent" communication mode. Natural language-based human commands to a robot that are accurately interpreted by the LLM enable the robot to more flexibly respond to real-world assembly scenarios where not every part and/or tool can be assumed to have been seen in advance. Such a natural mode of communication can be highly desirable in a batch and/or personalised production environment where an increased level of situation awareness, on-demand response, and adaptability to changing workflows is called. Such scenarios may be increasingly encountered as on-site repair and remanufacture are playing an increasingly important role to enhance sustainability in manufacturing. However, human-in-the-loop checking and intervention in case of unexpected behaviours generated by LLMs is necessary to ensure the accuracy and robustness of task execution. Future effort will focus on lightweight development by investigating the 1-bit quantisation technique, more autonomous and proactive robot cooperations, LLM-driven low-level robot programming and continuous feedback-based action correction as well as model generalisation.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

The authors acknowledge support from the National Science Foundation under awards CMMI-1830295 and EEC-2133630 (ERC HAMMER), and Vetenskapsrådet under award 2023-00493.

References

[1] Wang L, Gao R, Váncza J, Krüger J, Wang XV, Makris S, Chryssolouris G (2019) Symbiotic Human-Robot Collaborative Assembly. *CIRP Annals* 68 (2):701–726.
[2] Oba Y, Weaver K, Parwal A, Nagasue H, Fujishima M (2021) High-Accuracy Pose Estimation Method for Workpiece Exchange Automation by a Mobile Manipulator. *CIRP Annals* 70(1):357–360.
[3] Wen B, Mitash C, Soorian S, Kimmel A, Sintov A, Bekris KE (2020) Robust, Occlusion-Aware Pose Estimation for Objects Grasped by Adaptive Hands. *2020 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, Paris, France, 6210–6217.
[4] Wen B, Tremblay J, Blukis V, Tyree S, Müller T, Evans A, Fox D, Kautz J, Birchfield S (2023) BundleSDF: Neural 6-DoF Tracking and 3D Reconstruction of Unknown Objects. *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Canada606–617.
[5] Zhang J, Liu S, Gao RX, Wang L (2023) Neural Rendering-Enabled 3D Modeling for Rapid Digitization of In-Service Products. *CIRP Annals* 72(1):93–96.
[6] Addepalli S, Weyde T, Namoano B, Oyedeji OA, Wang T, Erkoyuncu JA, Roy R (2023) Automation of Knowledge Extraction for Degradation Analysis. *CIRP Annals* 72(1):33–36.
[7] Hagendorff T, Fabi S, Kosinski M (2023) Human-Like Intuitive Behavior and Reasoning Biases Emerged in Large Language Models but Disappeared in ChatGPT. *Nature Computational Science* 3(10):833–838.
[8] Figure. *Figure 01 + OpenAI: Speech-to-Speech Reasoning and End-TO-End Neural Network*, https://www.figure.ai/.
[9] Hu Y, Xie Q, Jain V, et al. (2023) *Toward General-Purpose Robots Via Foundation Models: A Survey and Meta-Analysis*, .
[10] Vemprala S, Bonatti R, Bucker A, Kapoor A (2023) *ChatGPT for Robotics: Design Principles and Model Abilities*, .
[11] Schmitt T, Hanek R, Beetz M, Buck S, Radig B (2002) Cooperative Probabilistic State Estimation for Vision-Based Autonomous Mobile Robots. *IEEE Transactions on Robotics and Automation* 18(5):670–684.
[12] Yin H, Varava A, Kragic D (2021) Modeling, Learning, Perception, and Control Methods for Deformable Object Manipulation. *Science Robotics* 6(54): eabd8803.