2023 IEEE International Conference on Robotics and Automation (ICRA 2023)
May 29 - June 2, 2023. London, UK

Vision-based Six-Dimensional Peg-in-Hole for Practical Connector Insertion

Kun Zhang, Chen Wang, Hua Chen, Jia Pan, Michael Yu Wang, Wei Zhang

***Abstract*— We study six-dimensional (6D) perceptive peg-in-hole problem for practical connector insertion task in this paper. To enable the manipulator system to handle different types of pegs in complex environment, we develop a perceptive robotic assembly system that utilizes an in-hand RGB-D camera for peg-in-hole with multiple types of pegs. The proposed framework addresses the critical hole detection and pose estimation problem through combining the learning-based detection with model-based pose estimation strategies. By exploiting the structure of the peg-in-hole task, we consider a rectangle-shape based characterization for modeling the candidate socket. Such a characterization allows us to design simple learning-based methods to detect and estimate the 6D pose of the target socket that balances between processing speed and accuracy. To validate our method, we test the performance of the proposed perceptive peg-in-hole solution using a KUKA iiwa7 robotic arm to accomplish the socket insertion task with two types of practical sockets (RJ45/HDMI). Without the need of additional search, our method achieves an acceptable success rate in the connector insertion tasks. The results confirm the reliability of our method and show that our method is suitable for real world application.**

I. Introduction

As one of the most promising applications of robotic manipulation, autonomous robotic assembly has attracted considerable research attention during the past several decades [1]–[6]. Such a problem focuses on operating a robotic arm to assemble different parts of an object to acquire its full functionality. This problem has great potentials in various practical application scenarios such as household service [7], industrial manufacturing [6], among others.

This work is supported by the Hong Kong Innovation and Technology Fund (ITF) under Grant ITS/036/21FP and the project of Hetao Shenzhen-Hong Kong Science and Technology Innovation Cooperation Zone (HZQBKCZYB-2020083). And also supported in part by National Natural Science Foundation of China under Grant No. 62073159 and Grant No. 62003155, in part by the Shenzhen Science and Technology Program under Grant No. JCYJ20200109141601708, and in part by the Science, Technology and Innovation Commission of Shenzhen Municipality under grant no.ZDSYS20200811143601004

Kun Zhang is with the Robotics Institute, Hong Kong University of Science and Technology, Clear Water Bay, Hong Kong. (Email: kun.zhang@connect.ust.hk)

Chen Wang and Jia Pan are with the Department of Computer Science, the University of Hong Kong, Hong Kong SAR. Chen Wang is also with the Department of Mechanical and Energy Engineering, Southern University of Science and Technology, Shenzhen, China.(Emails: cwang5@cs.hku.hk, jpan@cs.hku.hk)

Hua Chen and Wei Zhang are with the Department of Mechanical and Energy Engineering, Southern University of Science and Technology, Shenzhen, China. (Emails: chenh6@sustech.edu.cn, zhangw3@sustech.edu.cn)

Michael Yu Wang is with the HKUST Shenzhen-Hong Kong Collaborative Innovation Research Institute, Futian, Shenzhen, China, and also with the Monash University, Clayton VIC 3800, Australia. (Email: Michael.Y.Wang@monash.edu)

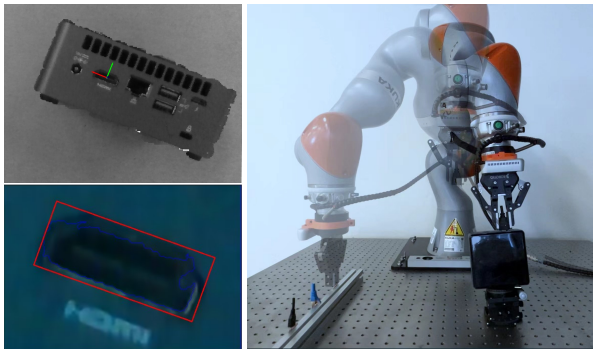


Fig. 1: Vision-based 6D connector insertion. The target socket is detected and its full 6D pose is estimated with the in-hand RGB-D camera. Given the estimated target socket's 6D pose, the robot insert the connector without additional search step.

Among the literature of robotic manipulation, different methods based on force or vision sensors are proposed to address the hole localization problem. When the positional uncertainty of the target hole is small, force sensor based methods can be used to align the peg with the hole [3], [4], [8], [9]. For the force sensor based methods, the peg is pressed against the hole surface and the hole is located based on the contact force. In [8], the spiral search method is proposed in which the peg is moved following a pre-defined spiral search path. Force and position readings are used to determine whether the peg is aligned with the hole. Rather than using the force sensing only as a stop condition for searching with a predefined trajectory, later works [3], [4] interpret the collected force sensing data and infer the hole's position. New types of force sensor are also adopted. More recently, in [9] the authors use Gelsight [10] gripper to get richer contact information and use it to estimate the contact line. However, the force sensor based methods are limited to small positional uncertainty and the orientational uncertainty is usually neglected or limited to 1 dimension.

As opposed to force sensor based hole localization strategies, vision sensors can handle large pose uncertainty and do not require physical contact between the peg and hole. The vision based method can be generally categorized into model-based method and learning based method. Commonly, the model-based method requires template of the hole for the pose estimation. In [11], CAD models are used as the template to extract the 2D position of the hole. With the advancement of deep learning in computer vision, learning based methods are also proposed [12]–[15]. Despite the impressive achievements of learning-based approaches, existing works only partially address the challenges. One limitation in most existing works

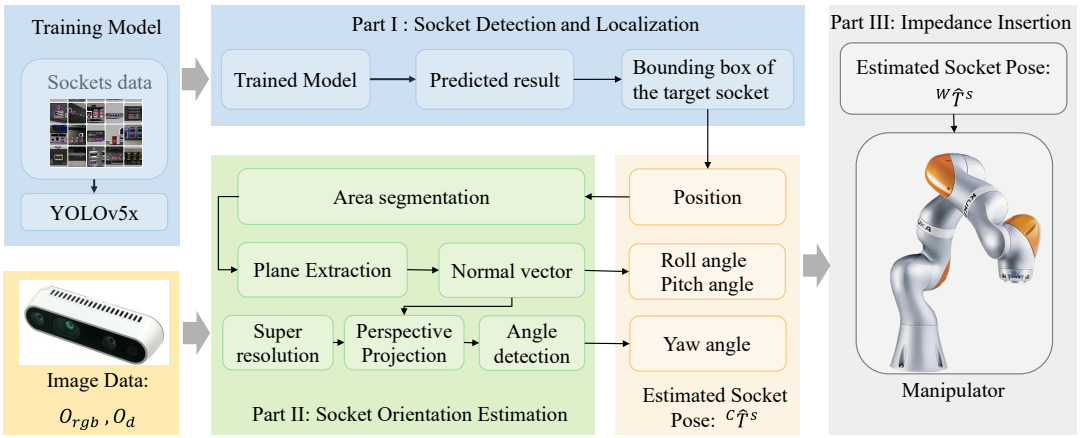


Fig. 2: Schematics of the proposed perceptive peg-in-hole system.

is that they do not estimate the full 6D pose of the hole. Works [12], [14] only consider predicting the 2D planar position of a round hole. While [13], [15] also considers the orientation, they do not consider the full orientation dimension in their experiments. Another limitation is that these methods do not handle the case in which multiple candidate holes are presented and the robot has to choose the correct one. Moreover, the estimated hole's pose may still has large error [12] and method like spiral search may still be needed to reduce the positional uncertainty. Recent advances in reinforcement learning (RL) brings new possibility into the Peg-in-Hole task. The trained policy can directly combine the multi-modal sensor data and output the control command [6], [16]. However, they also do not consider the full 6D pose of the hole and the required training data for RL method is quiet large.

In this paper, we aim to tackle the Peg-in-Hole problem with multiple candidate holes/sockets of different types within the same background as shown in Fig. 1, which appears ubiquitously in practical assembly scenarios. We mainly focus on the 6D hole pose estimation problem of the connectors insertion task. We assume that both the position and the orientation of the target socket is not known and we need to estimate the full 6D pose which is then used for the insertion. It should be noted that there are existing works focusing on the 6D object pose estimation, both in the object level [17] and the category level [18]. The difference between their works and ours is that they try to estimate the pose of an object while we try to estimate the pose of a hole/socket, which is a small part on a larger object. The relative pose of the socket with the larger object is usually not known and we try to directly estimate the 6D position of the socket. This means that we need to involve segmentation method to separate the socket from the whole object, meanwhile we can not use the convex geometry information to estimate the 6D pose.

Compared with existing works, the main contributions of this paper are summarized as follows. First, we present a socket detection and localization framework that can fast and reliably estimate the target socket's full 6D pose among multiple candidate sockets. The proposed framework combines learning-based method with traditional visual processing methods to tackle the 6D peg-in-hole problem that has not been adequately addressed in the literature. Second, our proposed method is data efficient and easy to deploy in practical scenarios. Unlike the RL based methods which require a large amount of training data, our method only use 100 annotated images to achieve accurate pose estimations of the candidate sockets. Third, the accurate target socket pose estimation frees us from additional search step during the insertion, which reduces the cycle time significantly. Last, we conduct extensive hardware experiments with daily used connectors and sockets, effectively demonstrating the applicability of the proposed approach in practical autonomous robotic assembly tasks.

II. Problem Description and Overview of the Proposed Framework

For the Peg-in-Hole problem studied in this paper, we consider a robotic arm together with a gripper and a depth camera that are installed on the robot's end. The frames that we consider in this work are W the world frame, s the socket frame, G the gripper frame and C the camera frame which are shown in Fig. 4. The poses of the gripper and the camera with respect to the world frame can be obtained by forward kinematics and are denoted by ${}^{W}T^{G}(\mathbf{q})$ and ${}^{W}T^{C}(\mathbf{q})$, where $\mathbf{q}$ is the joint positions of the robot arm. The camera outputs $\{O_{rgb}, O_d\}$, where $O_{rgb} \in \mathbb{R}^{N_u \times N_v \times 3}$ is the RGB image and $O_d \in \mathbb{R}^{N_u \times N_v}$ is the depth image. The output images are of width N_u and height N_v, both measured in pixel.

Given the above setup, the 6D multi-type connector insertion problem can be divided into the problem of estimating the target socket's 6D pose ${}^{W}T^{s}$ given the camera's outputs $\{O_{rgb}, O_d\}$ and the problem of planning insertion based on the estimated target socket's pose ${}^{W}\hat{T}^{s}$. In this work, we will focus on solving the first problem. For the practical problem that we consider in this paper, there are several key features in typical applications that allow for special treatments and thus achieving a satisfactory balance among speed, efficiency, and accuracy. First, despite the fact that the background is often cluttered in practical scenarios, the candidate socket's appearances and shapes are typically known and the data set for socket detection can be established. Second, the socket

ports are normally with a flat plane around. Therefore, the roll and pitch of the target socket can be extracted by estimating the orientation of the flat plane where it is located at.

The multi-connector insertion problem studied in this paper differs from many classical Peg-in-Hole problems in various aspects. The ability of dealing with different types of pegs with various noises is the most important feature. Fig. 2 gives an overview of the proposed framework. The overall estimation problem is decoupled into detection, localization and orientation estimation. We first train a learning-based object detection component. Given a target peg type, the candidate socket of the same type is first detected using the RGB reading O_{rgb} via the learning-based object detection component. Combined with the depth image O_d, we can get the target socket's pose in the camera frame ${}^{C}\hat{T}^{s}$. After coordinate transformation, the estimated target socket's pose in the world frame ${}^{W}\hat{T}^{s}$ can be obtained. With the estimated target socket's pose, the robot will directly insert the connector into the target socket with Cartesian impedance control without additional search.

III. Methodology

By virtue of the key features mentioned in previous section, we propose to decompose the estimation of target socket pose into three steps: detection, position estimation, and orientation estimation. By exploiting the first feature mentioned in the last paragraph, we adopt a learning-based approach to detect different types of sockets. Then we use the detected contours information to obtain the target socket's position. This leaves only the orientation of the target socket unknown. For connectors insertion task, the sockets are approximately of shape rectangle. However, unlike rectangle which has 2 mirror symmetry axes, the practical sockets usually only have 1 mirror symmetry axis. To simplify the problem, we assume that the uncertainty in the socket's orientation is less than 180 degrees and represent the target socket with a rectangle. We then extract the orientation of the target socket from the image data with the help of the rectangular representation.

A. Detection and Localization of the Target Socket

Thanks to the advancement of deep learning in computer vision, there are mature object detection algorithms based on convolutional neural network (CNN) which we can adopt for our usage. In particular, we need to implement an socket detection component $f_d(O_{rgb}, \theta)$ with neural network weights θ that takes the RGB image O_{rgb} as input and outputs the bounding boxes of all the sockets in view together with their types, $\{(u^i_{\text{min}}, u^i_{\text{max}}, v^i_{\text{min}}, v^i_{\text{max}}, c^i)|i = 1, 2 \ldots, N\}$, where $u_{\text{min}}, u_{\text{max}}, v_{\text{min}}, v_{\text{max}}$ are pixel-wise coordinates of bounding box, $c \in \mathcal{S}$ is the type of socket and N is the number of socket detected in the image O_{rgb}. If no candidate connector is detected, we will move the camera along a predefined rectangular search trajectory above the table until the target hole is observed. To achieve data efficiency and fast detection of the candidate pegs, we select `YOLOv5x` as the network [19] .

With the bounding box of the target socket obtained, we can then calculate the center of the socket in the image pixel coordinate $(\hat{u}, \hat{v})$ thanks to the symmetry of the rectangle:

$$\hat{u} = (u_{\text{max}} + u_{\text{min}})/2 \tag{1a}$$

$$\hat{v} = (v_{\text{max}} + v_{\text{min}})/2 \tag{1b}$$

The next step would be transferring the target's socket's position from the pixel coordinate to the camera frame. With the intrinsic matrix $\mathbf{K}$ of the camera, the estimated position of the target hole center ${}^{C}\hat{\mathbf{p}}^{s}$ can be calculated as:

$${}^{C}\hat{\mathbf{p}}^{s} = \begin{bmatrix} {}^{C}x^{s} \\ {}^{C}y^{s} \\ {}^{C}z^{s} \end{bmatrix} = {}^{C}\hat{z}^{s}\mathbf{K}^{-1} \begin{bmatrix} \hat{u} \\ \hat{v} \\ 1 \end{bmatrix} \tag{2}$$

where ${}^{C}\hat{z}^{s}$ is the depth of the target socket center, which can be obtained from the depth image O_d. To deal with the artifact in O_d, we first process it with hole-filling filter $f_{\text{hf}}(O_d)$ to fill the holes in the depth image caused by imperfections of perception and the presence of sockets. The depth of the socket in the camera frame ${}^{C}\hat{z}^{s}$ is then calculated as the mean of depths of randomly selected pixels in the hole's bounding box to compensate for the noise.

B. Orientation Estimation of the Target Socket

The next step is estimating the socket's orientation. We separate the orientation estimation into two steps: we first estimate the socket's roll and pitch angle by plane fitting with the O_d and then estimate the socket's yaw angle with the texture information from O_{rgb}.

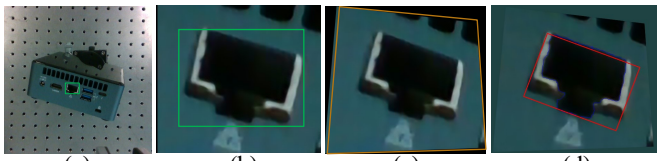
Algorithm 1: target socket orientation estimation

Input: RGB image O_{rgb},Depth image O_d, ${}^{C}\hat{p}^{s}$,camera intrinsic matrix K
Output: Socket Orientation $({}^{C}\alpha, {}^{C}\beta, {}^{C}\gamma)$
${}^{C}\mathbf{X} \leftarrow$ construct point cloud data using O_{rgb} and O_d;
${}^{C}\mathbf{X}^{\text{cropped}} \leftarrow$ crop point clouds around the target socket;
${}^{C}\mathbf{X}^{\text{filtered}} \leftarrow$ filter ${}^{C}\mathbf{X}^{\text{cropped}}$ with RANSAC;
$({}^{C}\alpha, {}^{C}\beta) \leftarrow$ apply PCA to ${}^{C}\mathbf{X}^{\text{filtered}}$;
$O^{\text{cropped}}_{rgb} \leftarrow$ crop O_{rgb};
$O^{\text{upscaled}}_{rgb} \leftarrow$ upscale O^{cropped}_{rgb} resolution by 4;
${}^{F}\mathbf{M}^{C} \leftarrow$ calculate the warp matrix using $({}^{C}\alpha, {}^{C}\beta)$;
$O^{front}_{rgb} \leftarrow$ do perspective transformation on $O^{\text{upscaled}}_{rgb}$ with ${}^{F}\mathbf{M}^{C}$;
${}^{C}\gamma \leftarrow$ find the rotation angle of minimum bounding rectangle in O^{front}_{rgb};
return $({}^{C}\alpha, {}^{C}\beta, {}^{C}\gamma)$;

1) Roll-pitch estimation for target socket: For typical applications such as computer connectors insertion or furniture assembly, the area around the target socket is usually a flat plane. Therefore, the roll and pitch angle of the target socket can be obtained by fitting the plane that it is located at. To this end, we first construct the point clouds ${}^{C}X$ using the depth image O_d and the RGB image image O_{rgb}. We do

not use the whole point cloud ${}^{C}X$ directly for two reasons. Firstly, the target socket is only located at a small surface area, there are noises in the converted point clouds ${}^{C}X$, especially for the whole object edges. Furthermore, using the whole point clouds in view ${}^{C}X$ has a high computational cost. To deal with these two problems, we use the partial point clouds ${}^{C}X^{\text{cropped}}$ cropped around the target socket center ${}^{C}\hat{\mathbf{p}}^{s}$ to fit the candidate plane containing the target socket. Since there could still be noises in ${}^{C}X^{\text{cropped}}$, we define a plane function and using RANSAC to filter out the outliers. With the filtered partial point clouds, ${}^{C}X^{\text{filtered}}$, we fit a plane with the Principal Component Analysis (PCA) method, which gives us the roll ${}^{C}\alpha$ and pitch ${}^{C}\beta$ angles of the target socket.

2) Yaw angle estimation for target socket: With the roll and pitch angles of the target socket obtained, we still need to estimate its yaw angle to obtain its full orientation. When the viewing axis of the camera and the z-axis of the socket are aligned, the rotation angle of the minimum rectangle that contains the socket in the RGB image O_{rgb} is the yaw angle of the target socket. However, since there are uncertainties in the roll and pitch angles of the target socket, the viewing angle and the z-axis of the socket are in general not aligned. Nonetheless, since we have already obtained the roll and pitch angles of the target socket, we can apply perspective transformation to transform the original RGB image O_{rgb} into the image with the viewing angle and the target socket's z-axis parallel. Details of how to obtain the yaw angles of the target sockets are given as follows.



(a) (b) (c) (d)

Fig. 3: Yaw angle estimation process:(a) the input RGB image O_{rgb} with the cropped image O_{rgb}^{cropped} shown in the green box;(b)the resolution-upscaled image $O_{rgb}^{\text{upscaled}}$;(c) the orthogonal front view image of the socket O_{rgb}^{front} after perspective transformation; (d) the red contour is the minimum rectangle containing the socket

We first crop the original RGB image to the small image O_{rgb}^{cropped} that only contains the target socket, as shown in Fig. 3 (a). Due to the limitation of the camera, the obtained image O_{rgb}^{cropped} is of low resolution and cannot be directly used. Therefore, we adopt the super-resolution algorithm ESPCN [20] to upscale its resolution by 4. The obtained high resolution image $O_{rgb}^{\text{upscaled}}$ is shown in Fig. 3 (b). The up-scaling does not change the orientation information of the original image. We then apply the perspective transformation [21] to the upscaled image. The perspective matrix $\mathbf{P}$ is defined as:

$$\mathbf{P} = \begin{bmatrix} \cot(f_v) & 0 & 0 & 0 \\ 0 & \cot(f_v) & 0 & 0 \\ 0 & 0 & -\frac{f+n}{f-n} & -\frac{2fn}{f-n} \\ 0 & 0 & -1 & 0 \end{bmatrix} \tag{3}$$

where f_v is the field of view, f is the far plane distance, and n is the near plane distance. The warp matrix used for the perspective image can be constructed as follows:

$${}^{F}\mathbf{M}^{C} = \mathbf{P}\mathbf{T}\mathbf{R}_{\beta}\mathbf{R}_{\alpha} \tag{4}$$

where $\mathbf{R}_{\alpha}$ is the rotation matrix around the x-axis, $\mathbf{R}_{\beta}$ is the rotation matrix around the y-axis, and $\mathbf{T}$ is the translation matrix along the z-axis. By applying the warp matrix on the upscaled image, we can get the RGB image O_{rgb}^{front} that contains the target socket and for which the viewing axis is co-linear with the target socket's z-axis. The RGB image O_{rgb}^{front} is shown in Fig. 3 (c).

With the orthogonal front view image of the target socket O_{rgb}^{front}, its yaw angle ${}^{C}\gamma$ can be obtained with the following steps. We perform gray-scale processing, Gaussian blurring, and black-white binary inversion to obtain better contour information. The resulting image can be used for extracting the rotation information. We normalize the contour of the hole with a minimum bounding rectangle (see Fig. 3 (d)) and then calculate the rotation of the minimum bounding rectangle to get ${}^{C}\gamma$. It should be noted that since we assume that the uncertainty in the possible rotation of the socket is less than 180 degrees, we do not need to consider the mirror symmetry problem in the orientation of the rectangle.

With the estimated socket's position and orientation in the camera frame obtained, we can then proceed to get the estimated socket's pose in the world frame ${}^{W}\hat{T}^{s}$:

$${}^{W}\hat{T}^{s} = {}^{W}T^{GG}T^{CC}\hat{T}^{s} \tag{5}$$

where ${}^{G}T^{C}$ is the pose of camera in the grippers frame and ${}^{W}T^{G}$ is the pose of gripper in the world frame. The camera is fixed in the wrist of the manipulator and we can get an accurate measurement of its pose. The pose of the gripper in the world frame can be obtained by forward kinematics of the robot arm.

C. Impedance Control-based Insertion

With the obtained target socket pose ${}^{W}\hat{T}^{s}$, we can proceed to do the insertion. To do the insertion, the robot first moves to a pre-insertion position ${}^{W}T^{above}$, which is selected to be $(0, 0, \Delta_{z,0})$ in the socket frame with $\Delta_{z,0} > 0$.

Despite the precision of our target socket pose estimation component, there may still be errors in insertion. To account for the error, we use Cartesian impedance control during the insertion process, which makes the end-effector of the robot arm behave like a mass-damper system, thus being compliant with the positional error.

Denote the pose of end-effector as a 6 dimension vector $\mathbf{x}_e = (\mathbf{p}_e, \boldsymbol{\varphi}_e)$ with $\boldsymbol{\varphi}_e$ being the Euler angles, the twist (translation velocity and rotation velocity) as $\mathbf{v}_e$ and the desired end-effector position as $\mathbf{x}_d$, we let

$$\mathbf{A}(\boldsymbol{\varphi}_e) \triangleq \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{T}(\boldsymbol{\varphi}_e) \end{bmatrix} \tag{6}$$

where $\mathbf{I}$ is the 3×3 identity matrix, $\mathbf{0}$ is the 3×3 zero matrix and $\mathbf{T}(\boldsymbol{\varphi}_e)$ maps the derivative of Euler angles to angular

velocity. Then effectively, the Cartesian impedance controller tries to render the closed-loop system to behave as follows:

$$\mathbf{\Lambda}(\mathbf{q})\dot{\mathbf{v}}_e+\mathbf{\Gamma}(\mathbf{q},\dot{\mathbf{q}})\mathbf{v}_e = \mathbf{A}^{-T}(\boldsymbol{\varphi}_e)\mathbf{K}_p(\mathbf{x}_d-\mathbf{x}_e)-\mathbf{K}_d\mathbf{v}_e+\mathbf{F}_{ext} \tag{7}$$

where $\mathbf{\Lambda}(\mathbf{q})$ is the effective mass, $\mathbf{\Gamma}(\mathbf{q},\dot{\mathbf{q}})\mathbf{v}_e$ characterizes the Coriolis and centripetal effect, $\mathbf{K}_p$ is the stiffness matrix and $\mathbf{K}_d$ is the damping matrix. For more details, the readers are recommended to refer to [22].

IV. Experimental Results

A. Experiment Platform

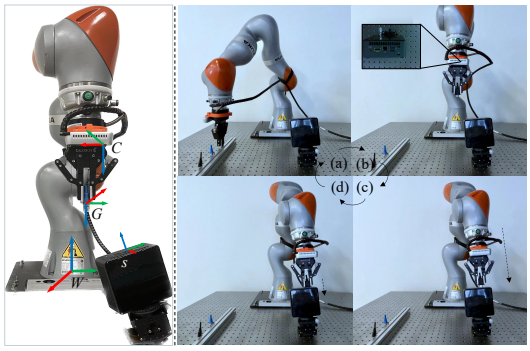


Fig. 4: The frames considered in this work are shown: W is the world frame, s is the socket frame, G is the gripper frame and C is the camera frame. Red, green and blue axes correspond to x, y and z axes respectively. An illustration of the insertion steps: (a) the robot moves to grasp the target connector with the grasp pose ${}^{W}T^{G_g}$ known; (b)the robot moves the camera to the detection pose to detect the corresponding socket; (c)with the target socket's pose estimated, the robot moves the connector to the pre-insertion pose above the target socket; (d) the robot inserts the connector into the target socket using impedance control

The manipulation system used for our experimental validation contains a KUKA iiwa7 R800 collaborative manipulator with 7 degrees of freedom, a Robotiq 2F-85 gripper, and an Intel Realsense D435 depth camera, see Fig. 4. The depth camera is rigidly mounted on the gripper via 3D printed support. In our tests, two types of connectors (HDMI and RJ45) are considered, which are grasped with the gripper. The sockets are all located on a NUC's back panel.

In Fig. 4, the snapshots of the insertion steps are given. Since we mainly focus on the detection and pose estimation of the target socket, we assume that the grasp pose ${}^{W}T^{G_g}$ for grasping the connector and the detection pose ${}^{W}T^{G_d}$ for which the target socket is in the camera's view are known. We use linear interpolation to get the robot's trajectory and use position control to track the trajectory. The remaining tasks are target socket detection, target socket pose estimation, and the final insertion. It should be noted that the insertion is sensitive to the pose error and therefore the pose estimation should be done with high accuracy which is challenging.

In order to obtain the ground truth value of the target socket's pose, all experiments are done with the NUC fixed on an optical multi-axis stage (see Fig. 4). The target socket's position can be measured from the grids on the table. Since the NUC is fixed on the optical stage, we can precisely control the ground true orientation of the target socket.

B. Dataset

We collect 100 images of random size from the Flickr website for training `YOLOv5x`. All images are under license of "commercial use & mods allowed". We hand-label the images with the bounding boxes and types. We use 72 annotated images as the training set, 18 images as the validating set, and 10 images as the testing set. A laptop with NVIDIA RTX3060 is used for the training, which takes about 1.5 hours to complete. The batch size is set as 2 and the training epoch is set as 300.

C. Metrics

We consider the following two metrics to evaluate the performance of the detection approach.

1) Pose Estimation Error(PEE): We measure the deviation of the estimated pose and the ground truth pose to calculate the pose error, which quantifies the accuracy of the proposed approach. Following the standard practice defined in [23], the position and orientation errors are defined as

$$\varepsilon_{pos} = \sqrt{({}^{W}p^{s} - {}^{W}p^{s}_{gt})^2} \tag{8a}$$

$$\varepsilon_{ori} = \arccos \frac{tr({}^{W}R^{s\,T}\,{}^{W}R^{s}_{gt}) - 1}{2} \tag{8b}$$

where ε_{pos} is the Euclidean distance between the estimated position of the target socket ${}^{W}p^{s}$ and the ground truth position ${}^{W}p^{s}_{gt}$. ε_{ori} is the minimum rotation angle required to align estimated rotation ${}^{W}R^{s}$ and the ground truth rotation ${}^{W}R^{s}_{gt}$.

2) Completion Time (CT): Completion time is the time used to complete one task cycle [24], which quantifies the efficiency of the proposed approach. Here we define the completion time T_{total} as

$$T_{total} = T_{detect} + T_{insert} \tag{9}$$

where T_{detect} is the time from the camera begins to acquire images to the system output of the target socket's pose. T_{insert} is the time from which the manipulator receives the target socket's pose to insert the connector successfully.

D. Results

1) Evaluation on different orientations and positions: In order to better verify the accuracy of our method, we divided the socket pose into position and orientation according to the evaluation metrics and measured them respectively.

Table I counts the mean orientation error for different states. For the orientation, we set each rotation angle as five states $(-10°, -5°, 0°, 5°, 10°)$. For each orientation state, we test 10 times and get the orientation error mean. For the position, we set all rotation angles as 0 and randomly placed the NUC back panel in a hundred positions. The ground truth positions are measured from the grids on the vibration isolation platform.

Fig. 5 shows the distribution of the orientation error in different rotation angles, which will greatly influence the final insertion success rate. Here we can see that the change of alpha angle has a greater impact on the orientation error. In contrast, the variation of beta and gamma angles has little effect on the error.

TABLE I: Pose estimation error under different orientations

$\varepsilon_{ori}{}^{\circ}$	α°																									
	10	5	0	-5	-10	10	5	0	-5	-10	10	5	0	-5	-10	10	5	0	-5	-10	10	5	0	-5	-10
RJ45																									
β° 10	1.62	1.63	1.49	0.89	0.79	1.57	2.62	1.43	1.56	1.91	0.53	1.98	1.62	1.13	1.92	2.56	1.97	0.25	2.02	1.23	2.47	1.90	1.10	1.87	1.30
5	2.51	1.84	1.38	0.65	1.68	2.10	2.18	1.50	1.20	1.42	2.52	1.96	1.38	0.79	0.92	2.51	1.33	1.26	0.70	1.29	2.58	1.39	1.15	1.22	1.41
0	2.22	1.45	0.81	0.43	1.42	2.47	1.60	0.55	0.96	0.98	2.51	2.12	0.32	0.71	1.62	2.27	2.32	1.28	1.29	1.26	1.89	2.49	1.05	1.27	2.12
-5	2.38	1.89	1.08	0.80	2.29	2.45	2.34	1.19	1.16	1.45	2.26	2.15	2.20	0.67	0.71	2.31	1.81	1.84	1.39	1.14	2.41	2.21	1.94	1.23	1.51
-10	1.76	1.08	0.99	1.67	1.23	2.00	1.57	1.13	1.45	1.27	2.25	1.74	0.70	1.62	1.43	2.26	1.80	1.23	1.56	0.76	2.42	1.66	1.14	2.55	1.06
HDMI																									
β° 10	1.75	1.18	1.64	0.79	2.06	2.12	1.54	0.86	1.07	1.68	2.19	2.52	1.90	0.72	2.31	2.21	2.19	0.56	1.74	1.56	2.47	2.41	0.62	1.66	2.14
5	1.87	1.65	1.38	1.41	2.51	1.24	1.45	1.64	1.45	2.26	2.13	0.89	1.48	1.43	1.85	2.36	1.25	1.32	1.12	1.51	2.53	2.35	1.95	1.01	1.64
0	1.05	0.89	1.36	1.77	1.88	0.63	1.26	0.76	1.28	1.03	1.21	2.31	1.02	2.09	2.45	2.62	1.02	0.21	1.83	2.36	2.38	2.41	0.49	2.15	2.16
-5	2.57	1.89	0.83	0.79	1.57	2.56	2.33	1.15	0.36	1.42	2.09	2.72	1.06	0.57	1.82	1.96	1.87	1.67	1.01	2.11	0.89	1.95	1.84	1.28	1.33
-10	2.32	0.83	1.50	1.38	1.17	1.78	1.74	0.52	1.74	1.23	1.61	1.68	1.01	1.13	0.73	2.39	2.01	1.71	1.24	1.60	2.57	2.11	1.37	1.48	1.33
	10					5					0					-5					-10					
γ°																									

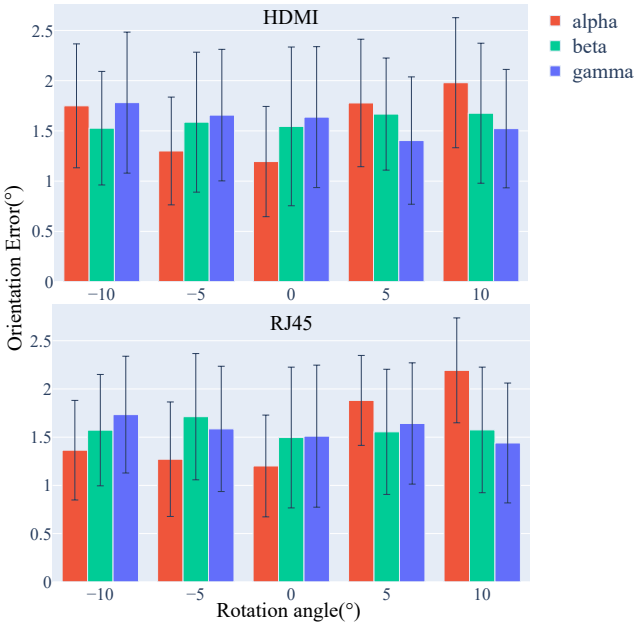


Fig. 5: Estimated orientation error distribution for HDMI and RJ45. The standard deviation is shown as the error bar.

2) 6D Peg Insertion: To verify the reliability of the proposed socket pose estimation method, we employ our method in a real peg-in-hole system. Here we consider two situations. One is for objects placed on a flat platform, the sockets are placed orthogonal to the camera coordinate system. This is common in real scenarios. For this situation we test 100 times for each type of socket. Another situation is more complicated. The orientation of the NUC back panel is not orthogonal to the camera coordinate system. We test 50 times for each type.

Fig. 6 gives the overall insertion performance for RJ45 and HDMI sockets. The detection time and total time are important for a task. The shorter they are, the better we want. For the orientation and position error, we give the mean errors here. The overall performance is similar, especially the pose errors are very close and the time costs are similar and acceptable. But the success insertion rate of HDMI 77% is much lower than the rj45's success insertion rate 94%. Therefore, the worse HDMI success insertion rate than RJ45 may due to the small tolerance of HDMI connector. The readers are kindly referred to the supplementary video for more experimental results.

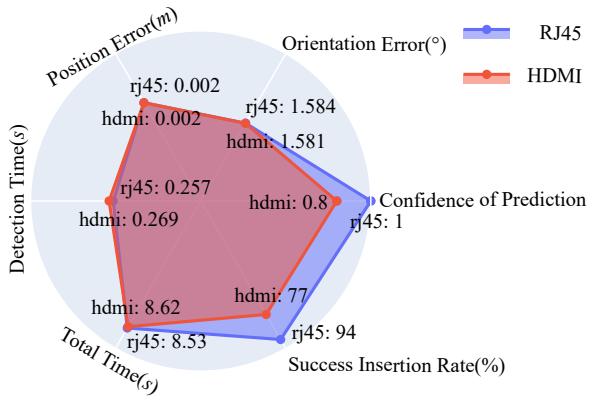


Fig. 6: Performance of our proposed framework for HDMI and RJ45

V. Conclusions

In this paper, we study the socket's pose estimation problem for the Peg-in-Hole task, which requires the proposed strategy to work with different types of small connectors with cluttered backgrounds. To tackle this complicated problem, the proposed strategy combines a vision-aided approach leveraging perception data for detecting and locating the target sockets and an impedance-based controller. By resorting to the perception data, the proposed approach effectively identifies and locates candidate sockets with high accuracy, freeing us from force-feedback based searching strategies that are not applicable to our case due to the cluttered background. The overall framework is validated with extensive experimental tests. The detection pose error is small. The overall success rate and completion time of inserting connectors into relevant sockets in a NUC back panel are acceptable.

In the current framework, the insertion strategy is decoupled from visual detection in the current solution. How to utilize data from both visual and force sensors to achieve real-time visual-force feedback in insertion control is an interesting problem to be further studied.

REFERENCES

[1] D. E. Whitney, "Quasi-Static Assembly of Compliantly Supported Rigid Parts," *Journal of Dynamic Systems, Measurement, and Control*, vol. 104, pp. 65–77, 03 1982.

[2] S. H. Drake, *Using compliance in lieu of sensory feedback for automatic assembly*. PhD thesis, Massachusetts Institute of Technology, 1978.

[3] W. Newman, Y. Zhao, and Y.-H. Pao, "Interpretation of force and moment signals for compliant peg-in-hole assembly," in *2001 IEEE International Conference on Robotics and Automation (ICRA)*, vol. 1, pp. 571–576 vol.1, 2001.

[4] Y.-L. Kim, B.-S. Kim, and J.-B. Song, "Hole detection algorithm for square peg-in-hole using force-based shape recognition," in *2012 IEEE International Conference on Automation Science and Engineering (CASE)*, pp. 1074–1079, 2012.

[5] T. Inoue, G. De Magistris, A. Munawar, T. Yokoya, and R. Tachibana, "Deep reinforcement learning for high precision assembly tasks," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 819–825, 2017.

[6] J. Luo, O. Sushkov, R. Pevceviciute, W. Lian, C. Su, M. Vecerik, N. Ye, S. Schaal, and J. Scholz, "Robust Multi-Modal Policies for Industrial Assembly via Reinforcement Learning and Demonstrations: A Large-Scale Study," in *Proceedings of Robotics: Science and Systems (RSS)*, (Virtual), July 2021.

[7] S. Levine, C. Finn, T. Darrell, and P. Abbeel, "End-to-end training of deep visuomotor policies," *The Journal of Machine Learning Research*, vol. 17, no. 1, pp. 1334–1373, 2016.

[8] S. Chhatpar and M. Branicky, "Search strategies for peg-in-hole assemblies with position uncertainty," in *2001 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, vol. 3, pp. 1465–1470 vol.3, 2001.

[9] S. Kim and A. Rodriguez, "Active extrinsic contact sensing: Application to general peg-in-hole insertion," in *2022 International Conference on Robotics and Automation (ICRA)*, pp. 10241–10247, 2022.

[10] I. H. Taylor, S. Dong, and A. Rodriguez, "Gelslim 3.0: High-resolution measurement of shape, force and slip in a compact tactile-sensing finger," in *2022 International Conference on Robotics and Automation (ICRA)*, pp. 10781–10787, 2022.

[11] H.-C. Song, Y.-L. Kim, and J.-B. Song, "Automated guidance of peg-in-hole assembly tasks for complex-shaped parts," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4517–4522, 2014.

[12] J. C. Triyonoputro, W. Wan, and K. Harada, "Quickly inserting pegs into uncertain holes using multi-view images and deep network trained on synthetic data," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5792–5799, 2019.

[13] M. Nigro, M. Sileo, F. Pierri, K. Genovese, D. D. Bloisi, and F. Caccavale, "Peg-in-hole using 3d workpiece reconstruction and cnn-based hole detection," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4235–4240, 2020.

[14] R. Haugaard, J. Langaa, C. Sloth, and A. Buch, "Fast robust peg-in-hole insertion with continuous visual servoing," in *Proceedings of the 2020 Conference on Robot Learning (CORL)* (J. Kober, F. Ramos, and C. Tomlin, eds.), vol. 155 of *Proceedings of Machine Learning Research*, pp. 1696–1705, PMLR, 16–18 Nov 2021.

[15] W. Gao and R. Tedrake, "kpam 2.0: Feedback control for category-level robotic manipulation," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 2962–2969, 2021.

[16] M. A. Lee, Y. Zhu, P. Zachares, M. Tan, K. Srinivasan, S. Savarese, L. Fei-Fei, A. Garg, and J. Bohg, "Making sense of vision and touch: Learning multimodal representations for contact-rich tasks," *IEEE Transactions on Robotics*, vol. 36, no. 3, pp. 582–596, 2020.

[17] Y. Xiang, T. Schmidt, V. Narayanan, and D. Fox, "Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes," 2018.

[18] H. Wang, S. Sridhar, J. Huang, J. Valentin, S. Song, and L. J. Guibas, "Normalized object coordinate space for category-level 6d object pose and size estimation," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[19] A. Bochkovskiy, C. Wang, and H. M. Liao, "Yolov4: Optimal speed and accuracy of object detection," *CoRR*, vol. abs/2004.10934, 2020.

[20] W. Shi, J. Caballero, F. Huszár, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, and Z. Wang, "Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1874–1883, 2016.

[21] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2 ed., 2004.

[22] L. Villani and J. De Schutter, "Force control," in *Springer handbook of robotics*, pp. 195–220, Springer, 2016.

[23] T. Sattler, W. Maddern, C. Toft, A. Torii, L. Hammarstrand, E. Stenborg, D. Safari, M. Okutomi, M. Pollefeys, J. Sivic, F. Kahl, and T. Pajdla, "Benchmarking 6DOF Outdoor Visual Localization in Changing Conditions," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 8601–8610, 2018.

[24] K. Kimble, K. Van Wyk, J. Falco, E. Messina, Y. Sun, M. Shibata, W. Uemura, and Y. Yokokohji, "Benchmarking Protocols for Evaluating Small Parts Robotic Assembly Systems," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 883–889, 2020.