actuators

MDPI

Article

Learning-Based Visual Servoing for High-Precision Peg-in-Hole Assembly

Yue Shen, Qingxuan Jia, Ruiquan Wang, Zeyuan Huang and Gang Chen *

School of Automation, Beijing University of Posts and Telecommunications, Beijing 100876, China
* Correspondence: chengang_zdh@bupt.edu.cn

Abstract: Visual servoing is widely used in the peg-in-hole assembly due to the uncertainty of pose. Humans can easily align the peg with the hole according to key visual points/edges. By imitating human behavior, we propose P2HNet, a learning-based neural network that can directly extract desired landmarks for visual servoing. To avoid collecting and annotating a large number of real images for training, we built a virtual assembly scene to generate many synthetic data for transfer learning. A multi-modal peg-in-hole strategy is then introduced to combine image-based search-and-force-based insertion. P2HNet-based visual servoing and spiral search are used to align the peg with the hole from coarse to fine. Force control is then used to complete the insertion. The strategy exploits the flexibility of neural networks and the stability of traditional methods. The effectiveness of the method was experimentally verified in the D-sub connector assembly with sub-millimeter clearance. The results show that the proposed method can achieve a higher success rate and efficiency than the baseline method in the high-precision peg-in-hole assembly.

Keywords: peg-in-hole; visual servoing; sim-to-real; assembly

Citation: Shen, Y.; Jia, Q.; Wang, R.; Huang, Z.; Chen, G. Learning-Based Visual Servoing for High-Precision Peg-in-Hole Assembly. *Actuators* **2023**, *12*, 144. https://doi.org/10.3390/act12040144

Academic Editors: Jing Wang, Zhijie Xu, Zhenyu Lu and Jonathan Gomez

Received: 15 February 2023
Revised: 25 March 2023
Accepted: 26 March 2023
Published: 27 March 2023

1. Introduction

Robotic peg-in-hole assembly is widely used in the industry, such as component assembly in the aviation field [1], automobile production lines [2], and 3C field [3]. At present, the peg-in-hole assembly is mostly applied in structured scenes, and the accurate pose between the peg and the hole is usually obtained by human demonstration. As the assembly line becomes more flexible and intelligent, robots are required to perform tasks with higher uncertainty. To handle the uncertainty of the hole position, visual and force feedbacks [4] are useful for the robots to find the hole. Methods such as visual servoing [5] and spiral search [6] are widely used. Visual servoing is the key part for eliminating the pose error because spiral search only works within a small contact range, especially in high-precision assembly tasks in the 3C field.

Visual servoing is a technique that uses visual feedback to control the robot, which can be categorized into position-based visual servoing (PBVS) and image-based visual servoing (IBVS). PBVS defines the position error in Euclidian space, and the position estimation is obtained from sensors such as stereo cameras or visual markers. Chang [7] reconstructed the 3D pose of the smartphone back shell using a corner detection algorithm and then adopted look-then-move and look-and-move control approaches to handle distal and proximal assembly tasks. Wang [8] introduced velocity feedforward to visual servoing, which can better track dynamic targets. Due to the accumulation of errors from pose estimation, robot–camera calibration, and robot positioning, it is difficult to achieve high accuracy with PBVS. IBVS defines the position error in image space and matches visual features such as SIFT [9], SURF [10], and ORB [11] between the current image and the target image. Gu [12] proposed an improved uncalibrated visual servo method based on projective homography, which can maintain robustness to image defects and noise. In peg-in-hole assembly tasks, IBVS can usually achieve a higher success rate than PBVS. However, IBVS is sensitive to camera occlusion.

Recently, deep learning has drawn wide attention in the community of robotic manipulation, especially in the field of robotic assembly [13,14]. Many learning-based visual servo methods have also been developed. Triyonoputro [15] trained a neural network to identify the hole location based on a concatenated image from two wrist cameras, but the visual servo is step-wise rather than continuous, making the assembly process inefficient. Haugaard [16] used multiple cameras to estimate the peg and hole positions simultaneously, enabling continuous visual servo. However, the above methods are only applicable to circular peg-in-hole assembly, not to other geometric shapes. To handle different shapes in real-world peg-in-hole assembly, Puang [17] presented a novel learning-based visual servoing method named KOVIS, which used one network to learn the keypoint representation from the image and another network to learn the robot motion based on the keypoints. However, KOVIS required the 3D model of objects for training in simulation, so there was a sim-to-real gap. Spector formulated the peg-in-hole assembly as a regression problem, and InsertionNet [18,19] was proposed with a fusion of vision and force to achieve daily insertion tasks. Inspired by human behavior for peg-in-hole, Xie [20] proposed the Seam Filling Net to fill the seam by gradually aligning the peg pose to the hole. However, most networks directly output the motion offset, and thus may face problems of scale and interpretability.

Humans can easily perform various peg-in-hole tasks for two main reasons: multi-modal sensors and an efficient strategy. Apart from flexible viewing angles and sensitive force perception, humans usually try to align the peg with the hole with peripheral geometric features, such as corners and edges. In the assembly process, humans need to focus on only a few landmarks for alignment, which is similar to face landmark detection. Motivated by the alignment strategy, we propose Peg-to-Hole Net (P2HNet) to extract specific landmarks of a workpiece. After training with synthetic data in a simulated environment, P2HNet can be quickly transferred to real-world assembly with a small amount of real data. The main contributions of this study are as follows:

(1) A neural network for workpiece landmark estimation is proposed, which can extract specific features from an image.
(2) A synthetic data generation method for peg-in-hole assembly is presented to achieve transfer learning for real-world assembly.
(3) A multi-modal peg-in-hole strategy is designed to combine learning-based visual servoing and force control.
(4) The results in real peg-in-hole experiments show that the proposed method has a higher success rate and efficiency compared to the baseline method.

The rest of this paper is structured as follows: Section 2 defines the robotic peg-in-hole assembly problem. Section 3 describes the proposed learning-based method. Section 4 presents the experiment and analysis. Section 5 summarizes the whole paper and discusses future research.

2. Problem Setup

The peg-in-hole assembly system mainly consists of three main parts: the workpiece, the manipulator, and the sensing system, as shown in Figure 1. The hole part is usually fixed on the workbench. The manipulator grips the peg part and is controlled to complete the assembly by visual feedback and force perception.

In real-world assembly tasks, the hole position error cannot be avoided due to mechanical errors and imperfect sensors. Therefore, the manipulator cannot directly insert the peg into the hole by pure motion planning, and contact in the assembly process is inevitable. Peg-in-hole assembly can be divided into two stages: hole search and insertion. In the hole search stage, the manipulator is controlled to align the peg with the hole using cameras and force sensors, where some pose errors are eliminated. In the insertion stage, the manipulator will eliminate the remaining pose errors to complete the assembly.

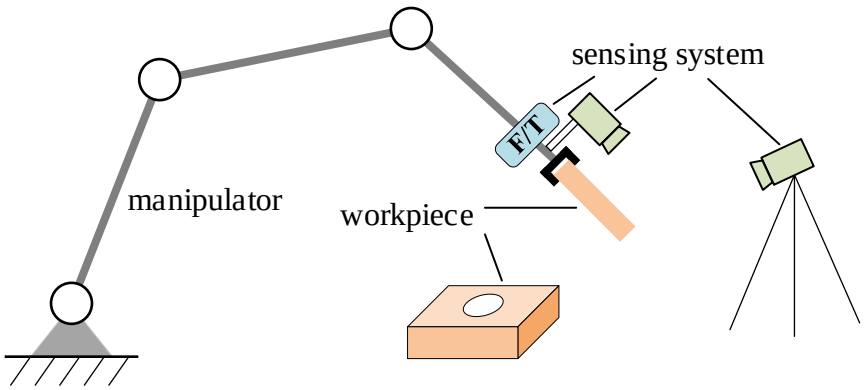


Figure 1. Peg-in-hole assembly system.

The difficulty of a peg-in-hole task depends on many factors such as workpiece shape and assembly tolerance. Unlike simple assembly tasks with circular or 3D-printed workpieces, we mainly focused on common connectors in the 3C field such as D-sub, USB, and RJ45. The connectors typically have irregular shapes and high precision with sub-millimeter tolerance, which makes it difficult to complete the assembly.

To simplify the problem, we assume that the peg is firmly gripped by the manipulator. The manipulator base and the hole are located on the workbench. Therefore, the hole pose error can be represented as a 4-DOF form:

$$e = [\Delta x, \Delta y, \Delta z, \Delta\theta] \tag{1}$$

where $\Delta x, \Delta y, \Delta z$ represent the position error, and $\Delta\theta$ is the orientation error in the z axis. In the hole search stage, the manipulator is required to eliminate the position error $\Delta x, \Delta y$ and the orientation error $\Delta\theta$. In the insertion stage, the position Δz is eliminated. To get a clear image, the wrist-mounted camera is adopted. The peg-in-hole assembly process is shown in Figure 2.

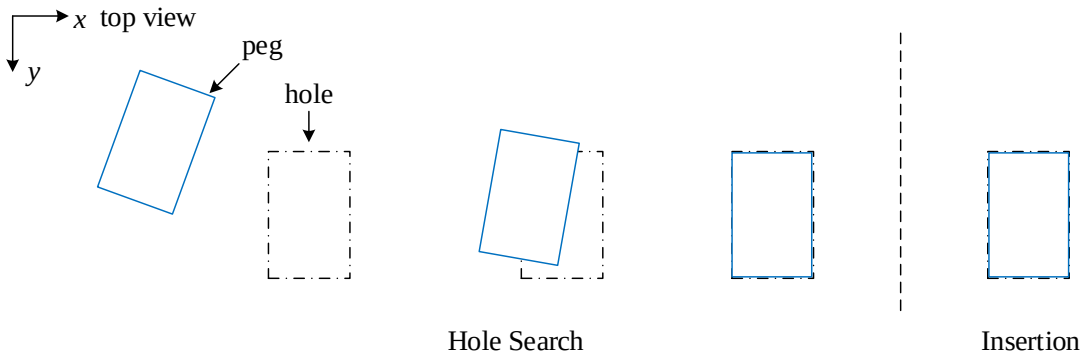


Figure 2. Peg-in-hole assembly process.

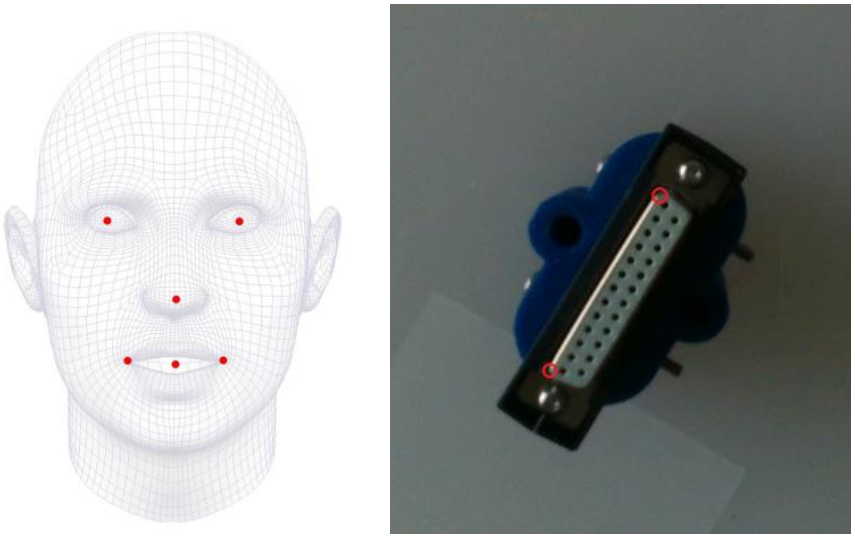
3. Method

As mentioned in Section 2, the peg-in-hole assembly comprises two stages: hole search and insertion. In our proposed method, the hole search has two steps: P2HNet-based visual servoing and spiral search. The visual servoing quickly moves the peg closer to the hole, while the spiral search is used to precisely align the peg with the hole. The assembly then switches to the insertion stage, which uses a constant force control strategy.

3.1. P2HNet

Humans can quickly align the peg with the hole with several landmarks. Inspired by this, we propose P2HNet to extract landmarks from a workpiece. Many neural networks can detect visual features well, but the number and distribution of key points are unclear.

Therefore, further feature matching is required, which is time-consuming. P2HNet can directly extract desired features to avoid feature matching. Similar to the 5 predefined landmarks in the face dataset AFW [21], we define 2 landmarks in the workpiece for peg-in-hole assembly. The workpiece landmarks can be considered as the application of the idea of face landmarks in the assembly field. The 2 workpiece landmarks are usually defined as the 2 endpoints of the contact edge during assembly. As for a D-sub connector, its landmarks are shown in Figure 3.



(**a**) (**b**)

Figure 3. Landmark definition. (**a**) Face landmarks in AFW. (**b**) Workpiece landmarks.

Direct output of pixel coordinates is widely used in many deep neural networks, but it loses the local relationship between adjacent pixels. P2HNet outputs heatmaps rather than pixel coordinates for better spatial generalization, similar to [22]. Let $p = (u, v)^T$ be the pixel coordinates of a point in image I. For a desired point p^*, its desired heatmap can be defined with a Gaussian kernel:

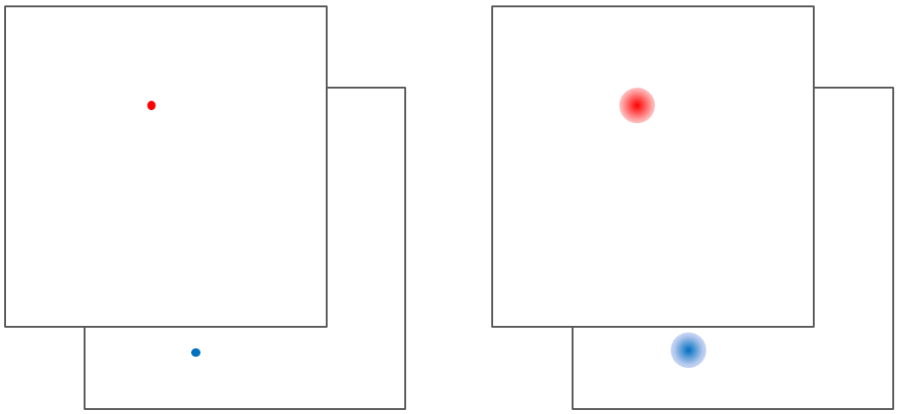
$$\Phi_{p^*} = \exp\left(-\frac{\|p - p^*\|^2}{2\sigma^2}\right) \tag{2}$$

where σ is the hyperparameter that controls the size of the active region in the heatmap. We set $\sigma = 3$ px for acceptable pixel error. The heatmap has the same resolution as the input image. To decouple 2 landmarks, the heatmap has 2 layers, and each layer represents a landmark. The landmarks and the heatmaps are shown in Figure 4.

Learning for a 2-layer heatmap can be modeled as an image regression problem, which can be estimated by a deep neural network f with trainable parameters φ:

$$\hat{\Phi} = f_\varphi(I) \tag{3}$$

The architecture of P2HNet based on U-Net [23] and ResNet [24] is shown in Figure 5, which is divided into the contracting path (left) and the expansive path (right). The contracting path gets a 3-channel RGB image, and then the expansive path outputs a 2-channel heatmap. A residual module is used in the contracting path for better training. The contracting path and the expansive path are directly connected in each block. The width of each rectangle in Figure 5 represents the number of channels in the image, while the height represents the image resolution. The meaning of the arrows can be found in the legend.



(**a**) (**b**)

Figure 4. Multi-layer heatmaps for workpiece landmarks. (**a**) Landmarks. (**b**) Heatmaps.

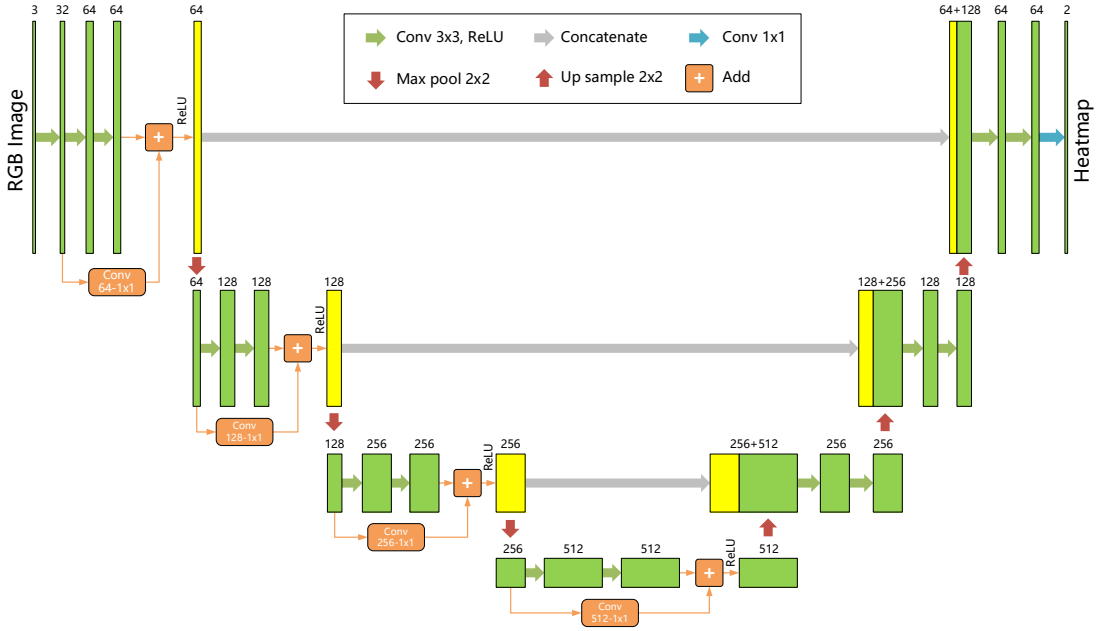


Figure 5. The architecture of P2HNet.

To train P2HNet, the loss is defined as the mean squared error between the desired and output heatmap:

$$\text{loss}(\Phi, \hat{\Phi}) = \frac{1}{W \times H} \sum_{i=0}^{H-1} \sum_{j=0}^{W-1} (y_{ij} - \hat{y}_{ij})^2 \tag{4}$$

where $W \times H$ is the image resolution, and $y_{ij}, \hat{y}_{ij}$ represent the pixel value in the coordinate (i, j) of the label and output heatmap.

The landmarks can then be extracted in the heatmap according to the maximum value:

$$\hat{p} = \underset{p}{\text{argmax}} \hat{\Phi}_p \tag{5}$$

3.2. Synthetic Data Generation

Collecting and annotating a real dataset is tedious and time-consuming. Moreover, the dataset cannot cover various cases due to different poses, varying lighting conditions, and diverse textures of workpieces. Therefore, we propose a method to generate synthetic data

in a simulated environment. The CAD model of the real workpiece is unnecessary because we can extract common features in a virtual dataset consisting of polygon primitives. The virtual data can be automatically collected and annotated, which can save a lot of time. After pre-training on the virtual dataset, a few real data are sufficient for transfer learning.

The virtual scene is built by Blender, an open-source software for 3D modeling. The assembly scene consists of a camera, a peg, and a hole, as shown in Figure 6. The hole is fixed on the plane and the peg is above the hole. To capture images from multiple perspectives, the camera is not attached to the peg and can move within a certain range of workspace. Considering that common connectors in the 3C field can be abstracted as simple geometric shapes, we designed three types of polygon primitives including trapezoid, square, and convex to represent them, as shown in Figure 7. Similar to the real connectors, the size of the virtual workpieces is 10~20 mm and the tolerance is 0.5 mm.

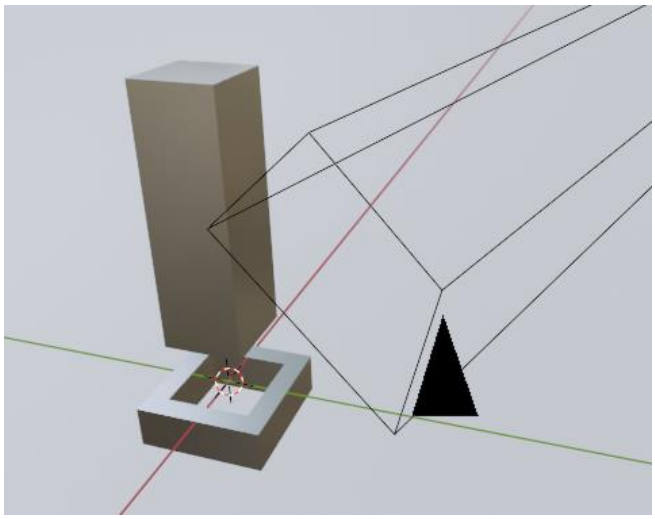


Figure 6. The virtual scene for peg-in-hole assembly.

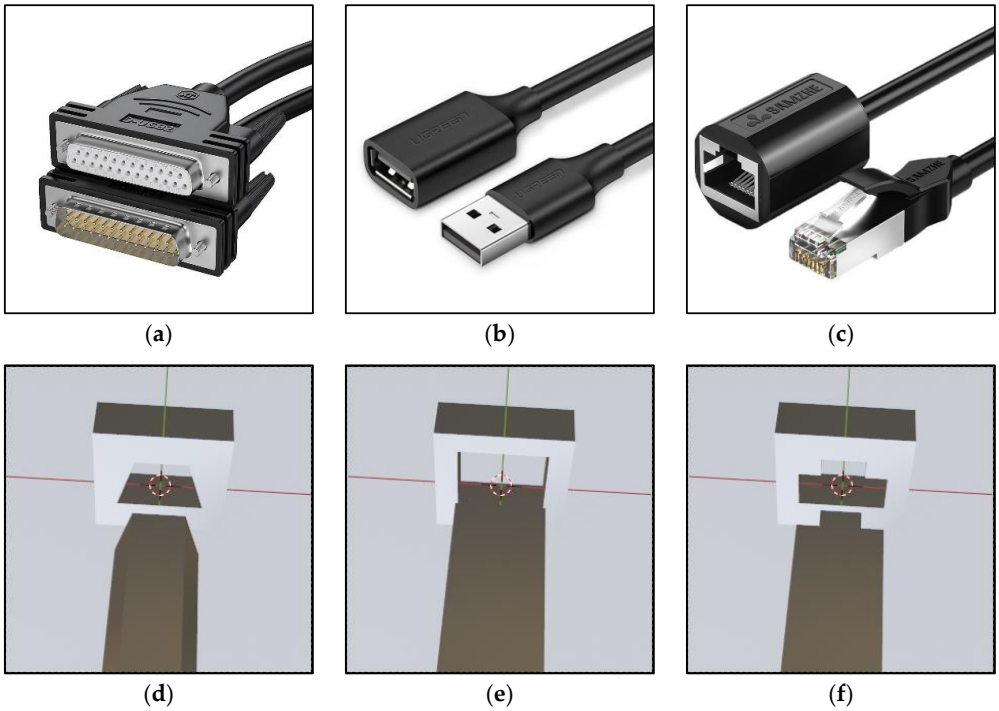


Figure 7. Common 3C connectors and polygon primitives. (**a**) DB-25. (**b**) USB. (**c**) RJ45. (**d**) Trapezoid. (**e**) Square. (**f**) Convex.

Due to the sim-to-real gap between virtual scenes and real tasks, domain randomization was adopted. The hole is fixed on the plane, while the peg position is uniformly sampled in a circle centered around the hole with a radius of 30 mm and a height relative to the hole between 5 and 15 mm. The peg orientation is sampled as an Euler angle in the Z axis between 0 and 5°. To generalize different camera views, the camera is not attached to the peg. When synthesizing an image, the distance from the camera to the hole is sampled between 120 and 150 mm. The angle between the optical axis of the camera and the plane is sampled between 50 and 60°, while the rotation angle around the optical axis is sampled between -5 and 5°. In addition to the pose randomization, we also introduced domain randomization of lighting, material, and texture to the virtual scene. For each type of workpiece, we generate 1000 rendered images. Some examples of the synthetic dataset are shown in Figure 8.

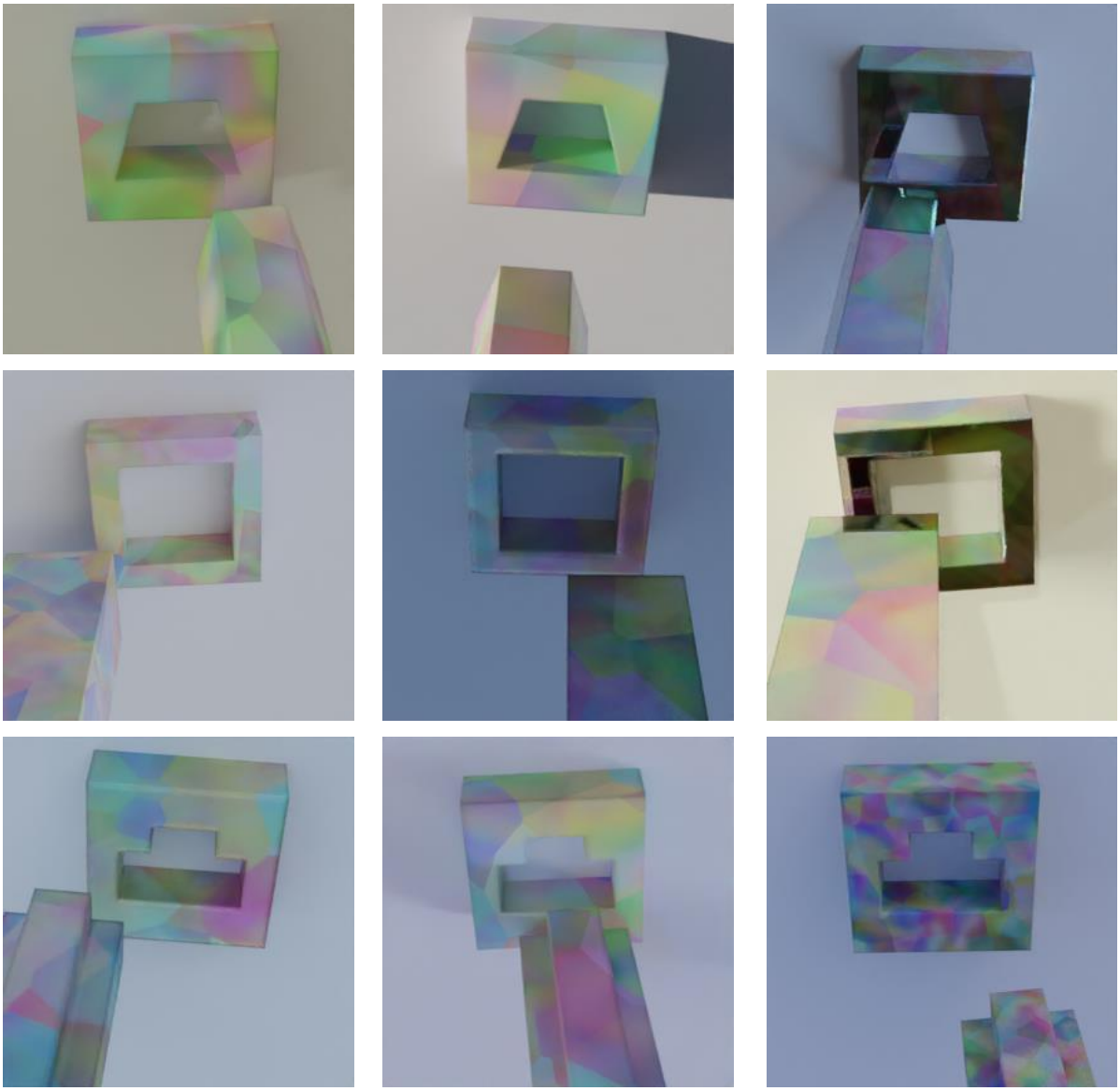


Figure 8. Synthetic data examples for different pegs and holes.

Camera parameters and workpiece positions can be accurately obtained in the virtual scene. Therefore, the pixel coordinates of landmarks can be automatically annotated without tedious manual work. Examples of landmark annotation are shown in Figure 9.

Figure 9. Workpiece landmarks for different holes.

3.3. Multi-Modal Peg-in-Hole Strategy

Considering that the peg is firmly gripped by the manipulator and the camera is attached to the end-effector, the pixel coordinates of the peg landmarks remain unchanged in the camera view. The peg landmarks can be obtained from the robot–camera calibration. The hole landmarks are extracted from P2HNet. The workpiece landmarks are shown in Figure 10.

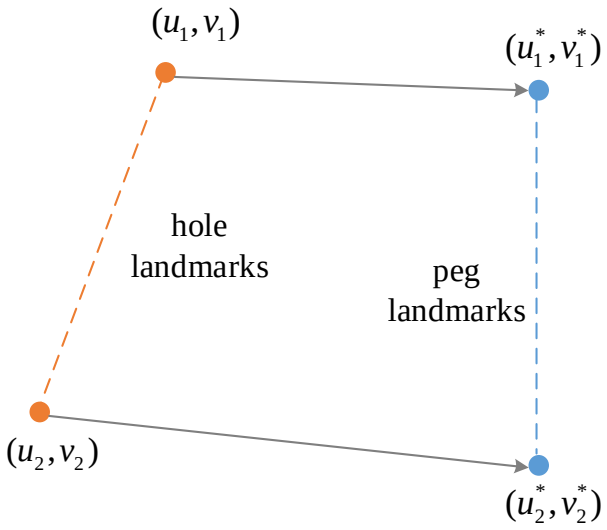


Figure 10. Workpiece landmarks for peg-in-hole.

The relationship between the motion of the camera and the pixels in the image is given by the image Jacobian matrix:

$$\begin{bmatrix} \dot{u} \\ \dot{v} \end{bmatrix} = \begin{bmatrix} -f/Z & 0 & u/Z & uv/f & -(f+u^2/f) & v \\ 0 & -f/Z & v/Z & f+v^2/f & -uv/f & -u \end{bmatrix} \begin{bmatrix} v_x \\ v_y \\ v_z \\ w_x \\ w_y \\ w_z \end{bmatrix} \tag{6}$$

where the velocity of a pixel can be expressed as the difference between the target position and the current position:

$$\begin{bmatrix} \dot{u} \\ \dot{v} \end{bmatrix} = \lambda \left(\begin{bmatrix} u^* \\ v^* \end{bmatrix} - \begin{bmatrix} u \\ v \end{bmatrix} \right) \tag{7}$$

In the hole search stage, the manipulator only needs to handle the pose error $\Delta x, \Delta y$ and $\Delta\theta$, so the motion relationship can be simplified:

$$\begin{bmatrix} \dot{u} \\ \dot{v} \end{bmatrix} = J(u, v, Z) \begin{bmatrix} v_x \\ v_y \\ w_z \end{bmatrix} \tag{8}$$

where $J(u,v,Z)$ is reduced as:

$$J(u, v, Z) = \begin{bmatrix} -f/Z & 0 & v \\ 0 & -f/Z & -u \end{bmatrix} \tag{9}$$

To calculate the required motion of the camera, the landmarks can be stacked with the same Z-height assumption.

$$\begin{bmatrix} \dot{u}_1 \\ \dot{v}_1 \\ \dot{u}_2 \\ \dot{v}_2 \end{bmatrix} = \begin{bmatrix} J(u_1, v_1, Z) \\ J(u_2, v_2, Z) \end{bmatrix} \begin{bmatrix} v_x \\ v_y \\ w_z \end{bmatrix} = J_s \begin{bmatrix} v_x \\ v_y \\ w_z \end{bmatrix} \tag{10}$$

Then, the motion of the camera can be calculated.

$$\begin{bmatrix} v_x \\ v_y \\ w_z \end{bmatrix} = \lambda J_s^{\dagger} \begin{bmatrix} u_1^* - u_1 \\ v_1^* - v_1 \\ u_2^* - u_2 \\ v_2^* - v_2 \end{bmatrix} \tag{11}$$

where $J_s^{\dagger}$ is the pseudo-inverse of J_s.

Depth information in the z direction can be obtained from a depth camera, and an image-based visual servo is performed based on Equation (8). When the camera view is occluded, it indicates that the peg has roughly reached the top of the hole. The manipulator is then controlled to move down in the z direction until the contact force is generated. A small spiral search is performed until the contact force changes abruptly in the z direction. Insertion can be completed with constant force control. The entire process is shown in Figure 11.

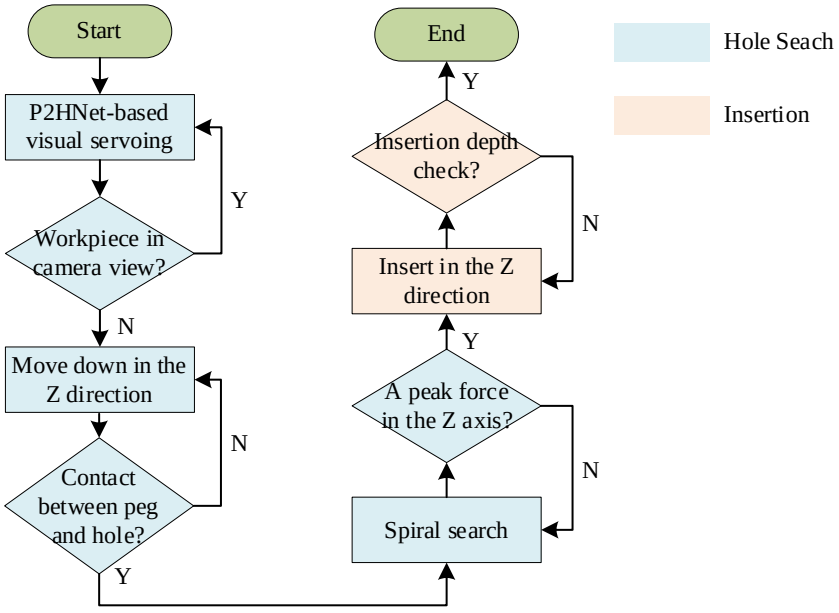


Figure 11. Multi-modal peg-in-hole strategy.

4. Experiment

In this section, we start with the experiment setup containing the necessary parameters for assembly. Then we evaluate the performance of P2HNet and transfer learning. In the end, we validate the multi-modal peg-in-hole strategy for the entire assembly process.

4.1. Experiment Setup

The peg-in-hole assembly platform is shown in Figure 12, which consists of a robot, a depth camera, and the NIST Assembly Task Board #1 [25].

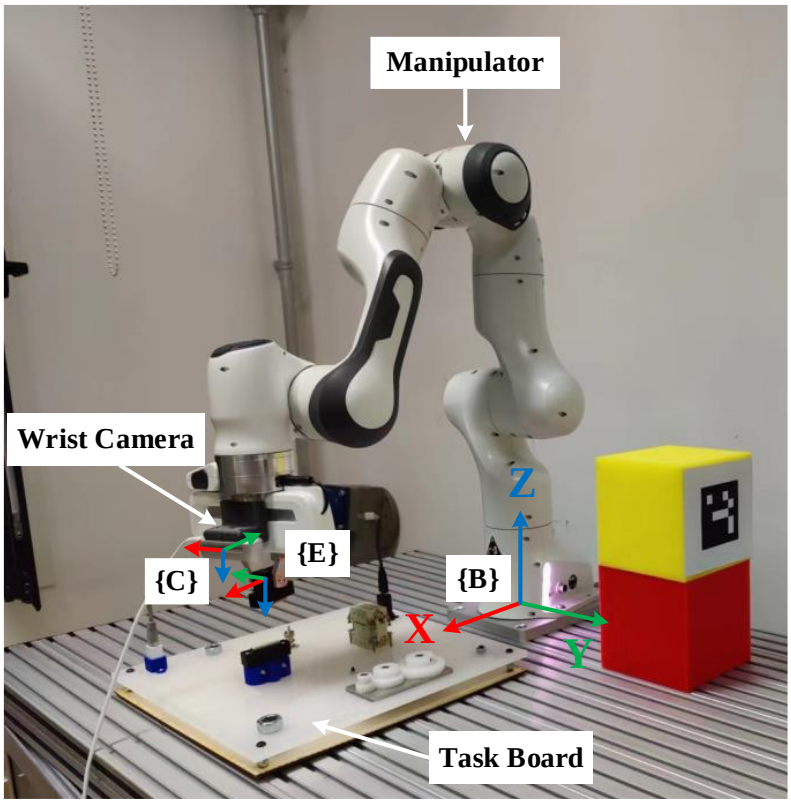


Figure 12. The peg-in-hole assembly platform.

We used the Franka Emika Panda robot, a 7-DoF torque-controlled robot. The pose, force, and torque estimation of the end-effector are accessible through the FCI interface. The Denavit–Hartenberg parameters of the robot are given in Table 1, and the pose repeatability is ± 0.1 mm.

Table 1. DH parameters of the Franka Emika Panda robot.

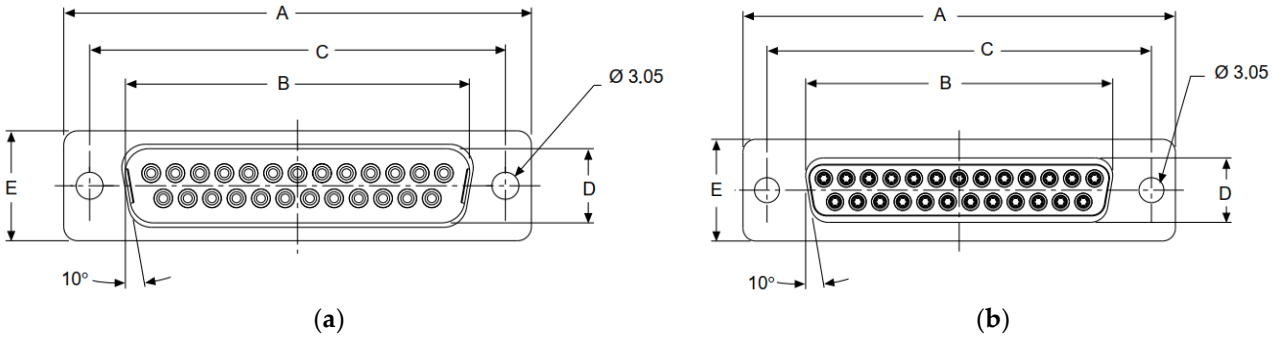
Link	a_{i-1} (mm)	α_{i-1} (°)	d_i (mm)	θ_i (°)
1	0	0	333	0
2	0	−90	0	0
3	0	90	316	0
4	82.5	90	0	0
5	−82.5	−90	384	0
6	0	90	0	0
7	88	90	0	0
Gripper	0	0	103.4	−45

The wrist camera, RealSense D435, was attached to the end-effector of the robot to observe the hole. The resolution of the camera was set to 640 × 480, which is suitable

for the assembly platform. The pose of the camera coordinate frame $\{C\}$ relative to the end-effector frame $\{E\}$ was calibrated.

$$ {}^{E}_{C}T = \begin{bmatrix} 0.0118 & -0.9999 & -0.0092 & 0.0512 \\ 0.9999 & 0.0117 & -0.0077 & -0.0354 \\ -0.0076 & 0.0093 & 0.9999 & -0.0359 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{12} $$

The NIST Assembly Task Board #1 includes 4 different 3C connectors (USB, RJ45, waterproof, and D-sub), ranging from easier single insertion to harder multi-pin alignment. The D-sub connector (DB-25) was selected as the workpiece in the following experiment. The mating face is shown in Figure 13, and the size/tolerance is given in Table 2. The minimum clearance between the peg and hole was 0.2 mm.



(**a**) (**b**)

Figure 13. The mating face of DB-25 connector. (**a**) Male (peg/plug). (**b**) Female (hole/receptacle).

Table 2. DB-25 size and tolerance.

DB-25	B	D
Male Size	38.96 ± 0.13 mm	8.36 ± 0.13 mm
Female Size	38.38 ± 0.13 mm	7.90 ± 0.13 mm
Fit Tolerance	0.52 mm	0.52 mm
Minimum Clearance	0.32 mm	0.20 mm

It should be noticed that although the motion accuracy of the robot is comparable to the minimum clearance of the DB-25, pure motion planning is inadequate to complete the assembly task due to the error in visual feedback.

All experiments were run on a computer with a 3.7GHz Intel(R) Core(TM) i9-10900X CPU and an RTX 2080Ti GPU.

4.2. Evaluation of P2HNet

4.2.1. Dataset

The virtual dataset was generated according to Section 3.2. To produce a real dataset, the robot is set to teach mode. Then, the end-effector of the robot is dragged by a human in the workspace. The images are collected continuously to cover different positions. The motion range of the robot during data collection is limited to Ω centered on the hole.

$$ \Omega: \ x \in [-0.1m, 0.1m], \ y \in [-0.1m, 0.1m], \ z \in [0.2m, 0.3m], \ \theta \in [-180^\circ, 180^\circ] \tag{13} $$

The DB-25 dataset consists of 420 images, and the image resolution is 640 × 480. Some examples are shown in Figure 14. LabelMe [26] was used to annotate the two landmarks of DB-25.

Figure 14. Dataset of D-sub Connector.

4.2.2. Train and Test

P2HNet was first trained on the virtual dataset containing 3000 images. The training set and the test set were divided by 9:1. To find the optimal learning rate, the one-cycle policy [27] was adopted in the training, and other hyperparameters are given in Table 3.

Table 3. Hyperparameters for training.

Hyperparameter	Value
Epoch	15
Optimizer	Adam
Maximum learning rate	0.001
Weight decay	0.0001
Mini-batch size	16

After training for 15 epochs, P2HNet was able to detect the workpiece landmarks, as shown in Figure 15.

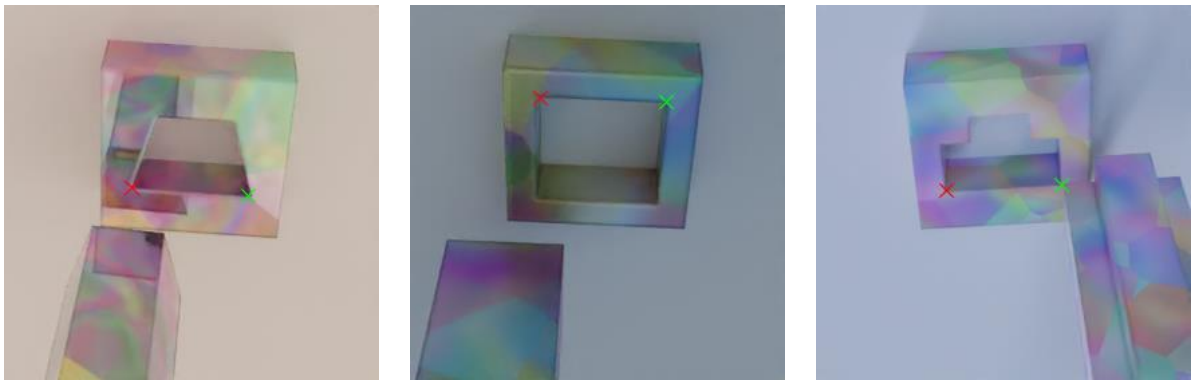


Figure 15. Workpiece landmark detection in the test set.

Based on the transfer learning method, P2HNet was initialized by the parameters trained on the virtual dataset, and then the real images were used for training. The learning curve with and without transfer learning is shown in Figure 16. P2HNet converges faster with transfer learning, and the final training loss is 20% less than that without transfer learning.

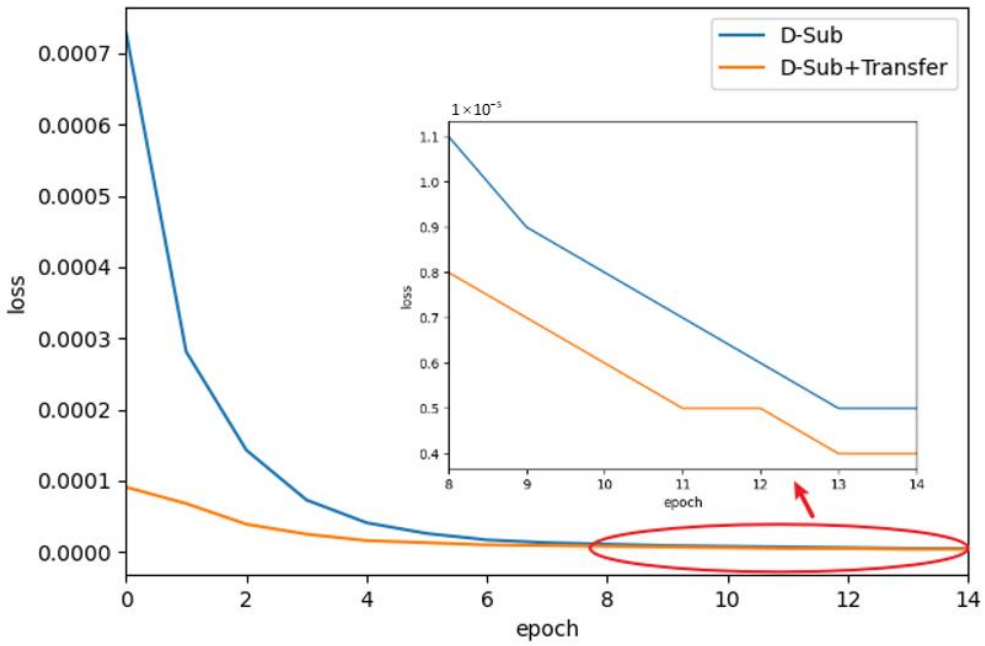


Figure 16. The learning curve on the real dataset.

To find out whether a smaller number of real images is sufficient for training with transfer learning, all/half/a quarter of the real dataset was used to train. The learning curve shown in Figure 17 indicates that more real images have less training loss. Moreover, half of the real dataset with transfer learning can achieve similar performance compared to the whole real dataset without transfer learning, which means much less tedious work of manual labeling.

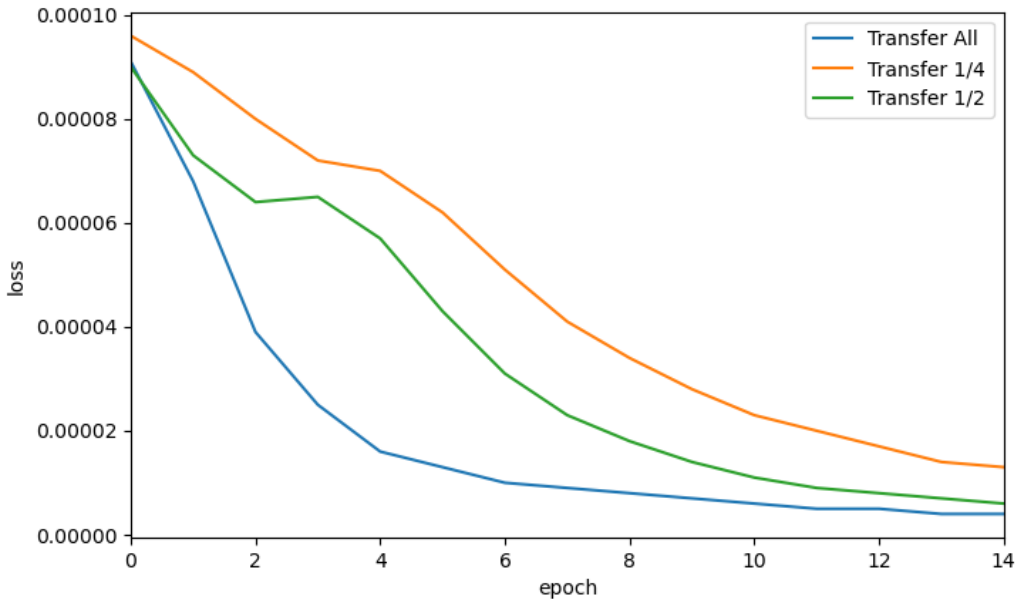


Figure 17. The learning curve on all/half/a quarter of the real dataset with transfer learning.

The heatmap predicted by P2HNet is shown in Figure 18. According to Equation (5), the workpiece landmarks are shown in Figure 19.

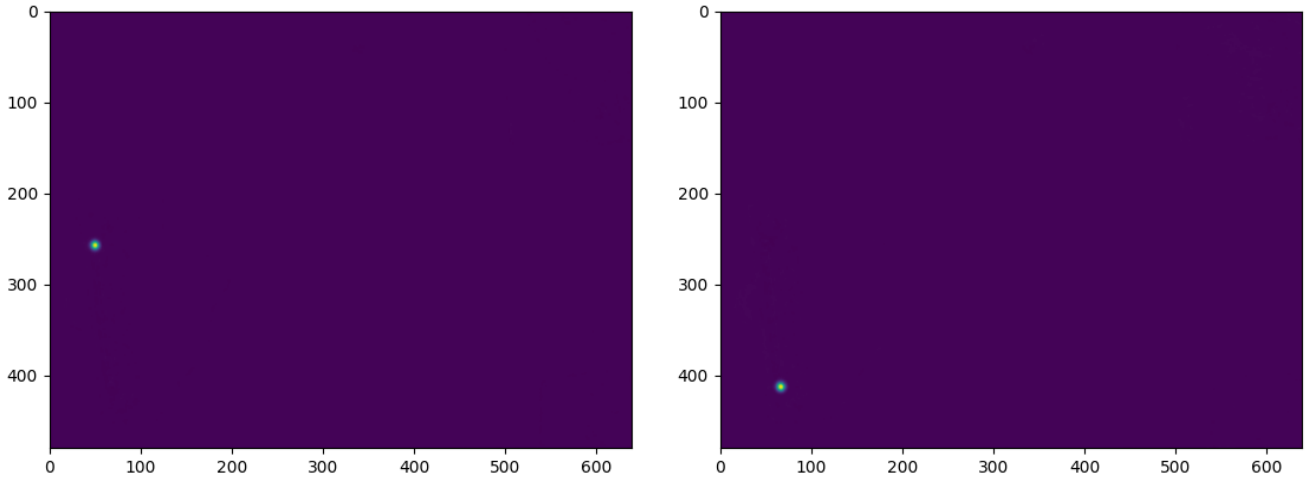


Figure 18. Heatmap of test image in DB-25 dataset.

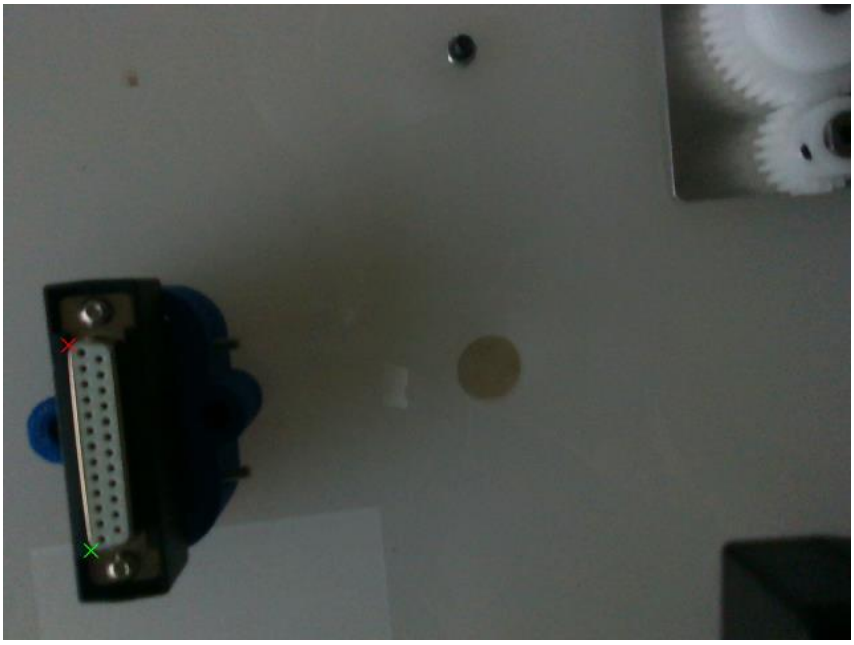


Figure 19. Workpiece landmark detection.

4.3. Peg-in-Hole Experiment

4.3.1. P2HNet-Based Visual Servo for Assembly

In the initial state, the peg is already firmly gripped by the manipulator, and the hole is in the view of the wrist camera. The visual servo control was performed based on P2HNet, followed by a spiral search. When the hole is found, a constant force control in the Z axis is used to finish the insertion. The motion process of the manipulator is shown in Figure 20. The end-effector trajectory of the manipulator is shown in Figure 21.

The external force profile is shown in Figure 22. According to the curve, the visual servoing for hole search was performed within 0~5 s. At 5.5 s, the external force in the z direction exhibited a peak change, and then the manipulator switched to spiral search after contact of the peg and the hole. At 6.8 s, the peak change of the external force in the z direction showed that the hole was found. Then, the peg was inserted into the hole by force control within the next 1 s.

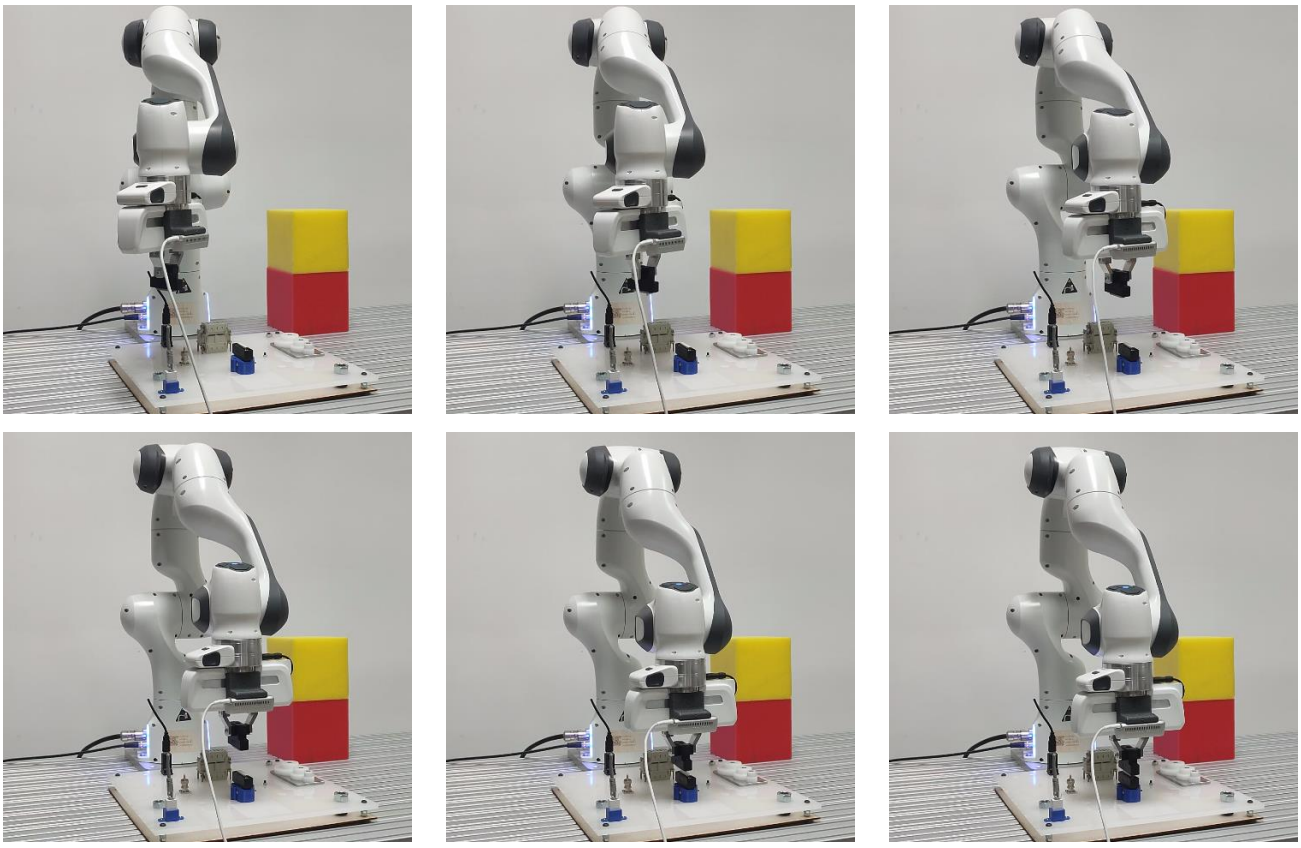


Figure 20. Snapshots of peg-in-hole experiment with DB-25.

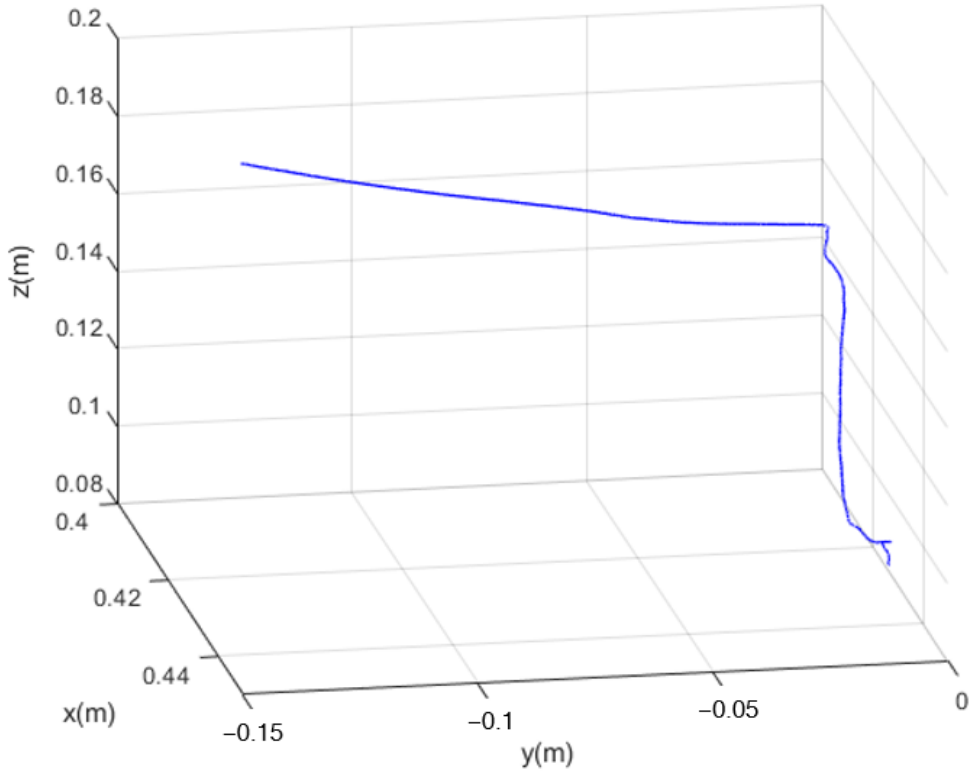


Figure 21. The manipulator end-effector trajectory.

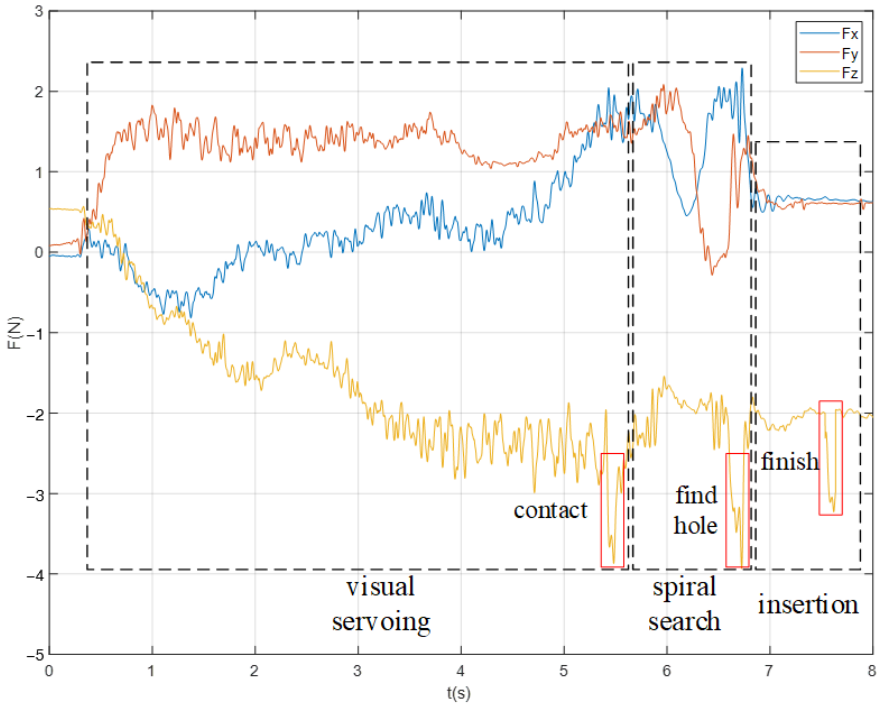


Figure 22. Force profile in the peg-in-hole assembly process.

4.3.2. Method Comparison

Because the contact area between the peg and the hole is limited in the mating face of DB-25, rather than a large plane in many other scenes, the success of the assembly mainly depends on the final pose of the peg after the visual servo. If the pose error is too large, the spiral search will fail to find the hole in such a limited contact area.

Given the same spiral search strategy, we compared the visual servo method based on P2HNet and ORB. ORB was chosen as the baseline method due to its accuracy and efficiency. The initial peg position was randomly sampled in a circle centered around the hole with a radius of 100 mm and a height of 100 mm relative to the hole. The initial peg orientation was randomly sampled between -5 and $5°$. The experiment was repeated 10 times.

Part of the visual servo process based on P2HNet is shown in Figure 23. The workpiece landmarks in red and green were accurately detected in the whole process, so a small final pose error can be eliminated by the spiral search.

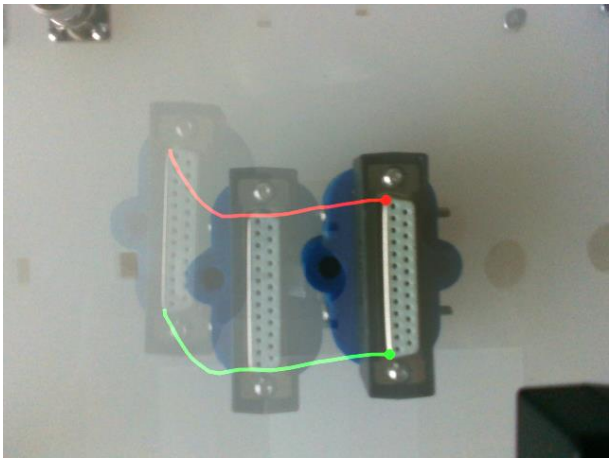


Figure 23. Visual servo process based on P2HNet.

In the ORB-based visual servo, the target image and pre-annotated workpiece landmarks are shown on the left in Figure 24. Based on feature detection and matching between the target image and the current image, a homography matrix H can be calculated. Then, the workpiece landmarks in the current image can be mapped from matrix H, as shown on the right in Figure 24. However, some wrong matches may occur due to similar features, as shown in Figure 25. The wrong matches will lead to a large pose error after the visual servo, so the spiral search may fail to find the hole.

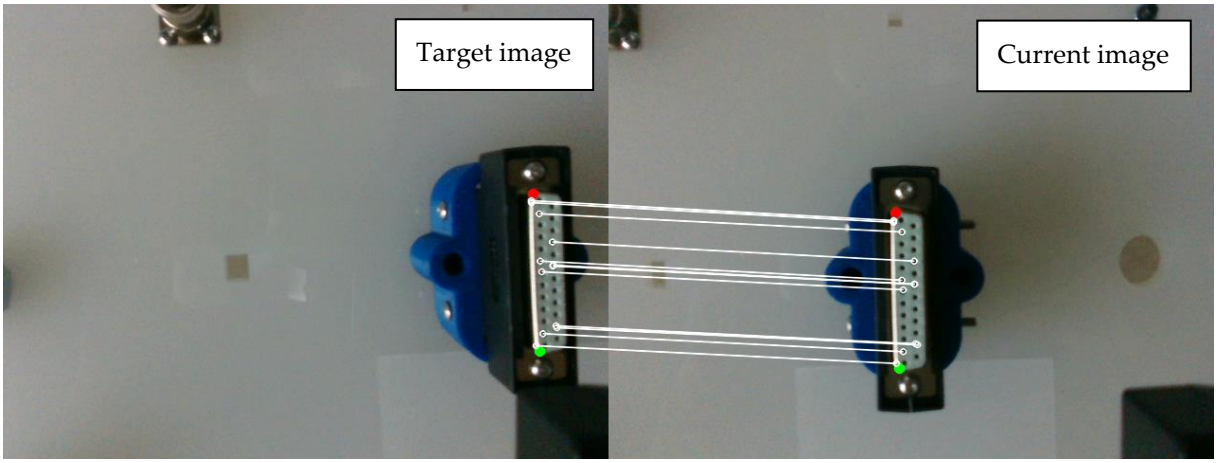


Figure 24. Visual servo based on ORB method (good match).

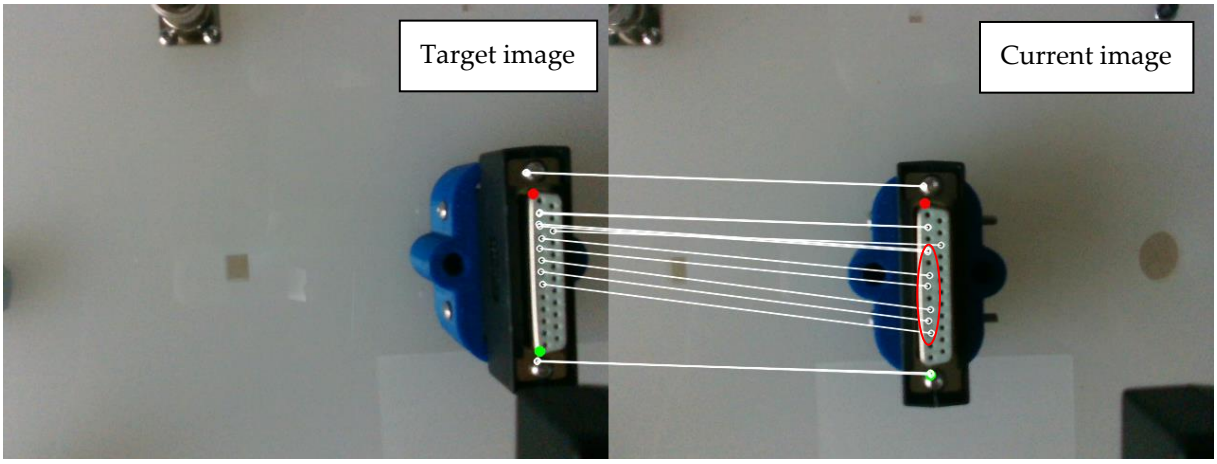


Figure 25. Visual servo based on ORB method (some wrong matches).

Further comparison between the ORB and P2HNet methods is shown in Table 4.

Table 4. Comparison between the ORB and P2HNet methods.

Comparison	ORB	P2HNet
Mean time for a vision servo step	43 ms	**15 ms**
Mean time for the hole search	8.5 s	**7.8 s**
Success rate of peg-in-hole	8/10	**10/10**

The bold indicates that P2HNet outperforms the ORB method.

The mean time for a visual servo step in P2HNet is 15 ms, much less than 43 ms in the ORB method. The primary reason for the difference is the pipeline of image processing. Compared with the ORB method, which requires feature detection, feature description, and feature matching, P2HNet directly extracts the desired landmarks through a neural

network without further feature matching. The mean time for the hole search time in both methods is close. The success rate of the P2HNet-based method outperforms the ORB-based method. Overall, the results show that the proposed method achieves a higher success rate and efficiency than the baseline method.

5. Conclusions

In this paper, a neural network named P2HNet is proposed, which can directly extract specified landmarks for visual servoing in the peg-in-hole assembly. To train P2HNet efficiently, a virtual assembly scene was established to generate a large number of labeled virtual images for transfer learning. Then, a multi-modal peg-in-hole strategy is proposed. Rough positioning was accomplished by P2HNet-based visual servoing, and then the alignment was completed by spiral search, followed by a force control to complete insertion. The method was validated on the peg-in-hole task of the D-sub connector with sub-millimeter clearance. The results confirmed that our method can achieve a 100% success rate with high efficiency. In the future, the generalization to different workpieces and a closer fusion between vision and force perception can be further investigated.

Author Contributions: Conceptualization, Y.S., Q.J. and G.C.; methodology, Y.S.; software, Y.S.; validation, Z.H. and R.W.; formal analysis, Y.S.; investigation, Y.S.; resources, G.C.; data curation, Y.S.; writing—original draft preparation, Y.S.; writing—review and editing, Q.J.; visualization, Y.S.; supervision, Q.J.; project administration, G.C.; funding acquisition, Q.J. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Major Project of the New Generation of Artificial Intelligence of China (No. 2018AAA0102904) and the National Natural Science Foundation of China (No. 51975059).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Mei, B.; Zhu, W. Accurate Positioning of a Drilling and Riveting Cell for Aircraft Assembly. *Robot. Comput.-Integr. Manuf.* **2021**, *69*, 102112. [CrossRef]
2. Hebecker, M.; Lambrecht, J.; Schmitz, M. Towards Real-World Force-Sensitive Robotic Assembly through Deep Reinforcement Learning in Simulations. In Proceedings of the 2021 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM), Delft, The Netherlands, 12–16 July 2021; pp. 1045–1051.
3. Haugaard, R.L.; Glent Buch, A.; Iversen, T.M. Self-Supervised Deep Visual Servoing for High Precision Peg-in-Hole Insertion. In Proceedings of the 2022 IEEE 18th International Conference on Automation Science and Engineering (CASE), Mexico City, Mexico, 22–26 August 2022; pp. 405–410.
4. Jiang, J.; Yao, L.; Huang, Z.; Yu, G.; Wang, L.; Bi, Z. The State of the Art of Search Strategies in Robotic Assembly. *J. Ind. Inf. Integr.* **2022**, *26*, 100259. [CrossRef]
5. Chaumette, F.; Hutchinson, S. Visual Servo Control. I. Basic Approaches. *IEEE Robot. Autom. Mag.* **2006**, *13*, 82–90. [CrossRef]
6. Park, H.; Park, J.; Lee, D.-H.; Park, J.-H.; Baeg, M.-H.; Bae, J.-H. Compliance-Based Robotic Peg-in-Hole Assembly Strategy Without Force Feedback. *IEEE Trans. Ind. Electron.* **2017**, *64*, 6299–6309. [CrossRef]
7. Chang, W.-C. Robotic Assembly of Smartphone Back Shells with Eye-in-Hand Visual Servoing. *Robot. Comput.-Integr. Manuf.* **2018**, *50*, 102–113. [CrossRef]
8. Wang, R.; Liang, C.; Pan, D.; Zhang, X.; Xin, P.; Du, X. Research on a Visual Servo Method of a Manipulator Based on Velocity Feedforward. *Space Sci. Technol.* **2021**, *2021*, 9763179. [CrossRef]
9. Niu, X.; Pu, J.; Zhang, C. An Improved SIFT Algorithm for Monocular Vision Positioning. *IOP Conf. Ser. Mater. Sci. Eng.* **2019**, *612*, 032124. [CrossRef]
10. Ding, G.; Liu, Y.; Zang, X.; Zhang, X.; Liu, G.; Zhao, J. A Task-Learning Strategy for Robotic Assembly Tasks from Human Demonstrations. *Sensors* **2020**, *20*, 5505. [CrossRef] [PubMed]
11. Kang, H.; Zang, Y.; Wang, X.; Chen, Y. Uncertainty-Driven Spiral Trajectory for Robotic Peg-in-Hole Assembly. *IEEE Robot. Autom. Lett.* **2022**, *7*, 6661–6668. [CrossRef]

12. Gu, J.; Zhu, M.; Cao, L.; Li, A.; Wang, W.; Xu, Z. Improved Uncalibrated Visual Servo Strategy for Hyper-Redundant Manipulators in On-Orbit Automatic Assembly. *Appl. Sci.* **2020**, *10*, 6968. [CrossRef]
13. Zou, P.; Zhu, Q.; Wu, J.; Xiong, R. Learning-Based Optimization Algorithms Combining Force Control Strategies for Peg-in-Hole Assembly. In Proceedings of the 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Las Vegas, NV, USA, 25–29 October 2020; pp. 7403–7410.
14. Spector, O.; Zacksenhouse, M. Learning Contact-Rich Assembly Skills Using Residual Admittance Policy. In Proceedings of the 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Prague, Czech Republic, 23–27 September 2021; pp. 6023–6030.
15. Triyonoputro, J.C.; Wan, W.; Harada, K. Quickly Inserting Pegs into Uncertain Holes Using Multi-View Images and Deep Network Trained on Synthetic Data. In Proceedings of the 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Macao, China, 4–8 November 2019; pp. 5792–5799.
16. Haugaard, R.L.; Sloth, C.; Langaa, J. Fast Robust Peg-in-Hole Insertion with Continuous Visual Servoing. In Proceedings of the CoRL, Cambridge, MA, USA, 16–18 November 2020; p. 10.
17. Puang, E.Y.; Peng Tee, K.; Jing, W. KOVIS: Keypoint-Based Visual Servoing with Zero-Shot Sim-to-Real Transfer for Robotics Manipulation. In Proceedings of the 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Las Vegas, NV, USA, 25–29 October 2020; pp. 7527–7533.
18. Spector, O.; Castro, D.D. InsertionNet-A Scalable Solution for Insertion. *IEEE Robot. Autom. Lett.* **2021**, *6*, 5509–5516. [CrossRef]
19. Spector, O.; Tchuiev, V.; Di Castro, D. InsertionNet 2.0: Minimal Contact Multi-Step Insertion Using Multimodal Multiview Sensory Input. In Proceedings of the 2022 International Conference on Robotics and Automation (ICRA), Philadelphia, PA, USA, 23–27 May 2022; pp. 6330–6336.
20. Xie, L.; Yu, H.; Zhao, Y.; Zhang, H.; Zhou, Z.; Wang, M.; Wang, Y.; Xiong, R. Learning to Fill the Seam by Vision: Sub-Millimeter Peg-in-Hole on Unseen Shapes in Real World. In Proceedings of the 2022 International Conference on Robotics and Automation (ICRA), Philadelphia, PA, USA, 23–27 May 2022; pp. 2982–2988.
21. Zhu, X.; Ramanan, D. Face Detection, Pose Estimation, and Landmark Localization in the Wild. In Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, 16–21 June 2012; pp. 2879–2886.
22. Sun, K.; Xiao, B.; Liu, D.; Wang, J. Deep High-Resolution Representation Learning for Human Pose Estimation. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition 2019, Long Beach, CA, USA, 15–19 June 2019; pp. 5693–5703.
23. Ronneberger, O.; Fischer, P.; Brox, T. U-Net: Convolutional Networks for Biomedical Image Segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015*; Navab, N., Hornegger, J., Wells, W.M., Frangi, A.F., Eds.; Lecture Notes in Computer Science; Springer International Publishing: Cham, Switzerland, 2015; Volume 9351, pp. 234–241, ISBN 978-3-319-24573-7.
24. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; IEEE: Las Vegas, NV, USA, 2016; pp. 770–778.
25. Lian, W.; Kelch, T.; Holz, D.; Norton, A.; Schaal, S. Benchmarking Off-The-Shelf Solutions to Robotic Assembly Tasks. In Proceedings of the 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Prague, Czech Republic, 1–27 September 2021; pp. 1046–1053.
26. Torralba, A.; Russell, B.C.; Yuen, J. LabelMe: Online Image Annotation and Applications. *Proc. IEEE* **2010**, *98*, 1467–1484. [CrossRef]
27. Smith, L.N.; Topin, N. Super-Convergence: Very Fast Training of Neural Networks Using Large Learning Rates. In Proceedings of the Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications, Baltimore, MD, USA, 10 May 2019; Volume 11006, pp. 369–386.