2025 IEEE/SICE International Symposium on System Integration (SII)
January 21-24, 2025, Munich, Germany

Vision-based Robotic Assembly from Novel Graphical Instructions

Chenxi Wang[1], Zhenting Wang[1], Takuya Kiyokawa[1], Weiwei Wan[1]
Natsuki Yamanobe[2], and Kensuke Harada[1,2*]

***Abstract*— For the purpose of performing robotic assembly from a novel graphical instruction, this paper proposes a new method for aligning assembly parts based on the visual information guided by the image in a graphical instruction manual. Our proposed method comprises two phases: We first detect an assembly part drawn in the instruction manual and then estimate its relative pose among multiple assembly parts to be assembled. For the detection of assembly parts, we build the matching algorithm based on fast-directional chamfer matching (FDCM) by utilizing multiple images of actual assembly parts captured from multiple angles. For the relative pose estimation, we collect the poses of the identified parts and match them with the assembly scenes in the graphical instruction manuals. We conducted collaborative assembly experiments between a human and a robot for a set of furniture parts. We confirmed that the captured images matched the graphics contained in the assembly manual well. In addition, we confirm that, with the help of a human, a robot can efficiently assemble furniture with a novel instruction manual.**

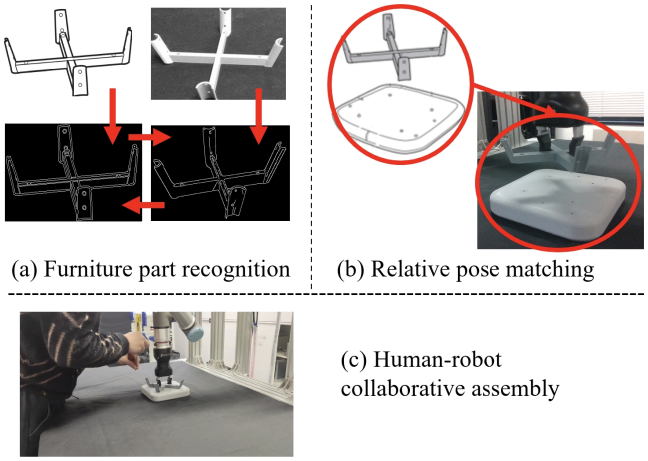


Fig. 1. Workflow of our proposed method

I. INTRODUCTION

In recent years, as the lifecycle of products has become shorter, manufacturing processes have begun to change; instead of mass-producing a single product, there is a shift towards mixed small-batch production. However, high-mix and low-volume manufacturing currently heavily relies on human labor. To enable robots to carry out such manufacturing tasks without the need for detailed task descriptions, robots must be able to automatically understand the details of the task and complete it based on simple or vague knowledge instructions. Written, oral, and graphical instructions are the most common types of task instructions widely used by humans. To adopt these types of task instructions on robots, it is necessary for robots to understand the meaning of abstract instructions and convert them into a form that robots can execute.

Among such task instructions, we focus on graphical instruction, which we commonly use to assemble furniture like IKEA [1] and NITORI [2]. So far, we have proposed an assembly task planning from a graphical instruction manual in which we complement the lacking information in the manual for a robot by building the ATSG (Assembly Task Sequence Graph) expressing a possible sequence of assembly task [3]. However, we have used the object recognition models trained by using the same instruction image from the same perspectives; this significantly limited the implementation of the assembly to novel instruction manuals.

In this paper, we propose a vision-based robotic assembly method from novel instruction manuals where our method does not use any prior information on the objects' 3D shapes for recognition. The overview of our proposed method is shown in Fig.1. Our proposed method realizes the assembly task by comparing the real image of the assembly parts with the image drawn in the instruction manual, just as a human does when he/she buys new furniture. Our proposed method comprises two phases: assembly part detection by matching the actual assembly part with its image in the instruction manual and relative pose estimation among multiple assembly parts by referring to the image drawn in the instruction manual.

For assembly part detection, we first extract the part images from the instruction manual to serve as templates. We then take the actual part's image from various angles. We also extract multiple features from the part images to robustly match the actual part's image with the image drawn in the instruction manual. Our matching algorithm is based on the Fast Directional Chamfer Matching (FDCM) [4].

For relative pose estimation, we obtain the pose of each assembly part by referring to the image of assembly parts drawn in the manual. By utilizing a camera mounted on the robot's wrist, we capture the image of the corresponding assembly part from multiple different poses. Image matching is performed based on FDCM. Once an image is successfully matched, we can obtain the corresponding pose of that object. After identifying the pose of each part to be assembled, we

[1]Department of Systems Innovation, Graduate School of Engineering Science, Osaka University, 1-3 Machikaneyama, Osaka, Japan. [2]National Institute of Advanced Industrial Science and Technology (AIST), Aomi 2-4-7, Tokyo, Japan. Contact: Kensuke Harada, harada@sys.es.osaka-u.ac.jp

can determine the relative poses of the two parts during assembly and apply this information to robotic assembly planning.

We evaluate the effectiveness of our proposed method through experiments. In the experiment of assembly part detection and relative pose estimation, we recognized eight parts from two furniture sets. The recognition accuracy and time were recorded, and the robustness of the method was analyzed. For the assembly experiment of a NITORI [2] chair, we focus on how to use manuals for path planning in pick and place tasks during robot-assisted furniture assembly, so we performed the human-robot collaborative assembly where the robot was responsible for joining parts, and the human was responsible for tightening screws. We first obtained the relative poses among the assembly parts from the furniture assembly scene. The robot then carried out the pick-and-place of furniture parts. Subsequently, the human completed the assembly. We recorded the similarity between the found relative postures and the postures on the furniture part manual and completed the assembly of the furniture parts.

II. RELATED WORK

As an application of AI algorithms, assembly planning problems have been focused [5], [6]. On the other hand, the primary target of this research is the robotic furniture assembly. Reinforcement learning is also hot in 3D part assembly [7], [8] of furniture. Lee et al. [9] developed a system for simulating furniture assembly over 80 different furniture models used for reinforcement learning. Yu et al. [10] also established a virtual assembly environment. Aslan et al. [11] has developed a system for assembly learning based on objects' point cloud. Knepper et al. [12] proposed a robot system that aims to assemble furniture using the part's CAD model. Similarly, there has been a wealth of research in determining grasp poses [13], [14] and human-machine interaction [15]–[17] for assembly tasks. For utilizing a graphical instruction manual, our previous research [3] has proposed a method named ATSG to generate assembly instructions based on the image in the manual; we build a deep-learning model for recognizing actual parts and generate available assembly sequence for the robot. Lee has also proposed a system for generating assembly planning based on manual [14]. But both these methods need to first train a deep-learning model for recognition, which will limit the application to novel instruction manuals.

Our proposed method uses object recognition and pose estimation from RGB images. The object recognition [18]–[20] and pose estimation [21], [22] have been extensively researched over the decades [23] based on deep learning. Our method for object recognition and pose estimation uses the feature-based method without learning to cope with a novel instruction manual. Imperoli et al al. [24] proposed a method to recognize 3-D parts named Directional Chamfer Matching [4]. The method matched the CAD 3-D model of the part with the real image based on the edge map by using a tensor-based optimization. For pose estimation, Li et al. in [21] focused on the category-level pose estimation for articulated objects from a single depth image.

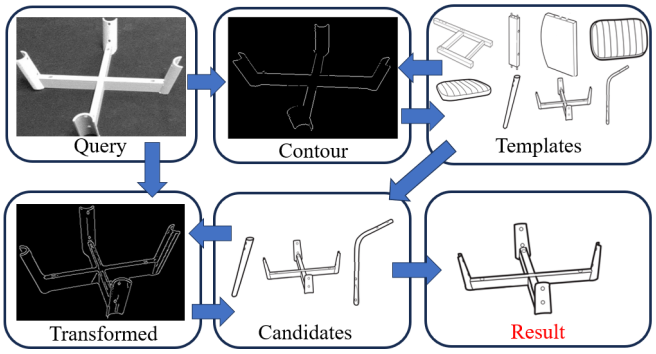


Fig. 2. The workflow of the part detection algorithm

On the other hand, this is the first trial of vision-based robotic assembly based on novel graphical instruction manuals.

III. ASSEMBLY PART DETECTION

The method we introduce now separates posture recognition from object recognition, but we have also considered combining the two. However, after multiple experiments and tests, we found that merging them into one would consume more time, as the image precision required for posture recognition is far greater than that for object recognition. Therefore, separating them into two processes is the most time-efficient approach. And as the first step of the proposed method, we explain our method for detecting the assembly part drawn in the instruction manual. We detect the assembly object drawn in the instruction manual from the real world by comparing the image of the real assembly part with the template image of the instruction manual. To this end, we set up cameras at multiple angles to take images of the object, assuming that the images we capture include the surfaces of the parts drawn in the instruction manual. The overview of the detection algorithm is shown in Fig.2. We first extract the contours of the assembly part's RGB image. Then, as we rotate the image, we compare it with the template's contour of the instruction manual. From the comparison results, we select multiple templates with a lower matching cost as candidates. The algorithm is detailed in the following.

A. Image Processing

The method for image processing for the assembly part detection is shown in Fig.3. We first set the query image of the assembly part taken from the RGB camera. Then, we extract the image of the assembly part from the instruction manual and set its size to be the same as the size of the query RGB image. To compare the query image with the image of the instruction manual, we first obtain the grayscale image of the query. Next, we apply the Gaussian filter for noise reduction, after that, we use the Canny operator [25] to extract the binary image. However, the binary image of the actual object obtained through Canny extraction is not as clear in its contour lines of the manual, such as light

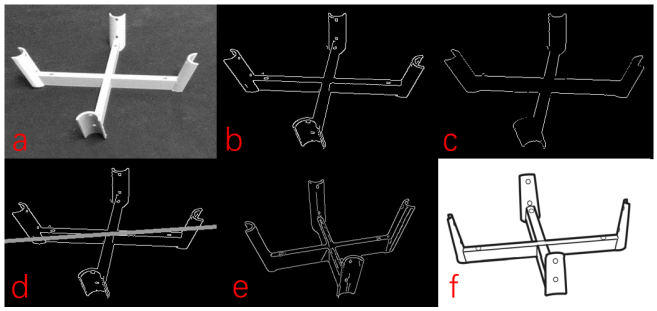


Fig. 3. The process of extracting features from actual part images where a) RGB image of the actual part, b)Extracted binary image, c) Extracted contour map, d) Fitting orientation in the image coordinate system, e) interpolated rotated image and f) the target template image.

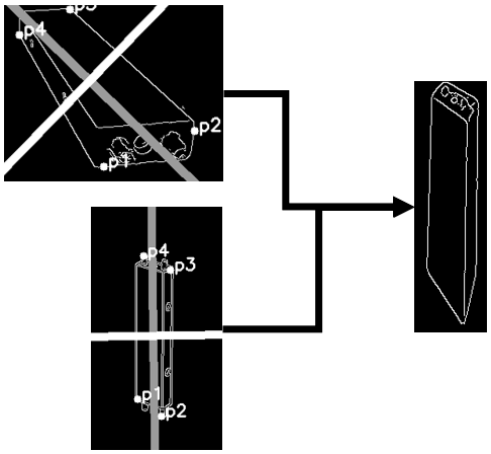


Fig. 4. Homography transformation (HT) of part's image. We first divided the image into four areas with fitted orientation of an object, then selected the corner point, and applied the HT algorithm to the query image to transform the pose.

reflections on the actual object, making the boundaries in the binary image discontinuous. To cope with such situations, we also implemented a simple algorithm to extract contours. We traverse each point in the image, and for each point, establishing several rays originating from that point, evenly distributed. If every ray has other points, then the point is judged as inside the contour; if at least one ray does not have any points from the image on it, then the point is judged as on the contour.

B. Detection Algorithm

We then scale down the input image proportionally, We perform a planar rotation of the query image's contour based on the difference in orientation between the object and the template in the image coordinate system, and compare the rotated contour with the template's contour. From the comparison results, we select multiple templates with a lower matching cost as candidates.

After that, we want to select the matched template in those candidates. We detect the assembly part by matching the query image with the instruction manual image. Here, we have to consider that the view pose of the query image is not always the same as the view pose of the instruction manual image. To compare the query image with the image of the instruction manual, we use the Homography Tramsnsformation(HT) [26] to modify the query image. The HT algorithm can calculate the planar perspective transformation matrix through corresponding points, and by applying the perspective transformation, we can adjust the posture on the query image to more closely resemble the posture depicted in the manual.

This significantly reduces the demands on the view pose of the query image to be precisely the same as the view pose of the image in the instruction manual. As for the selection of corresponding points required by the HT algorithm, we first extract corner points from the contours of the query image by using the Shi-Tomasi method [27].

First, we fit the center of the object, and using this center as the origin, we construct a Cartesian coordinate system based on the previously fitted orientation of the image layer and its perpendicular direction. This divides the entire image into four sections. For each previously extracted corner point, we search within the divided area for the point that is farthest from the origin to serve as the matching point, as shown in Fig.4.

However, since we apply a perspective transformation to the image after obtaining the transformation through the HT algorithm, it results in distortion of the image after the transformation, especially when there is a significant difference in posture between before and after the transformation. To cope with this problem, we considered the matching scores using both the contour image and the original image after applying the HT.

C. Image Similarity Evaluation

We then introduce the method for evaluating the similarity between two images. We choose Fast Directional Chamfer Matching(FDCM) to evaluate image similarity. This is because the FDCM is a method to match two images based on the chamfer distance, which works well on the line segment features. Since the image in the instruction manual is a binary image, it contains line segment information, which best suits this situation. In addition, compared with the original chamfer matching [28], FDCM not only matches the shapes but also takes into account the orientation of the shapes, resulting in more accurate matching results.

By translating the template image, the FDCM tries to minimize the chamfer distance. In addition to the original FDCM, we take the object size of the template image into consideration.

In summary, the similarity score between the input query image and template image in the instruction manual is calculated as follows. For the smaller-scaled rotated contour obtained from the query image, we apply the FDCM to compute the cost_1. By comparing cost_1, we can filter out a certain number of candidate templates with quantity limitations. In addition, we apply HT to transform the contour obtained from the query image. We apply the FDCM to this image and compute the cost cost_2. We will filter the matching results

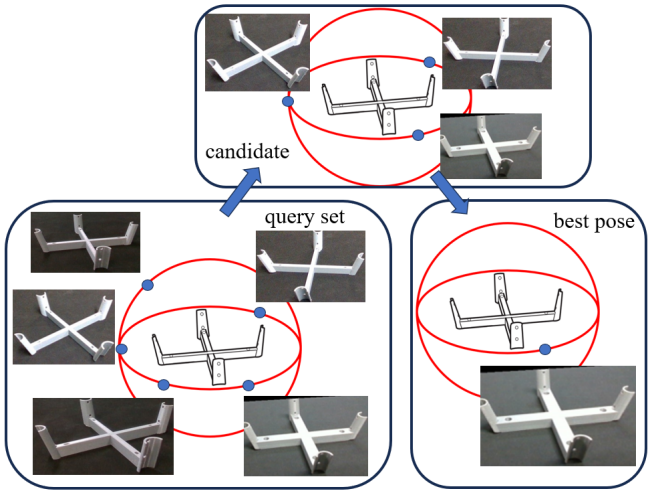


Fig. 5. The workflow of pose estimation of an assembly part. We first take images of the part from different angles. By comparing the similarity of the images, we filter out candidates and then select the best pose.

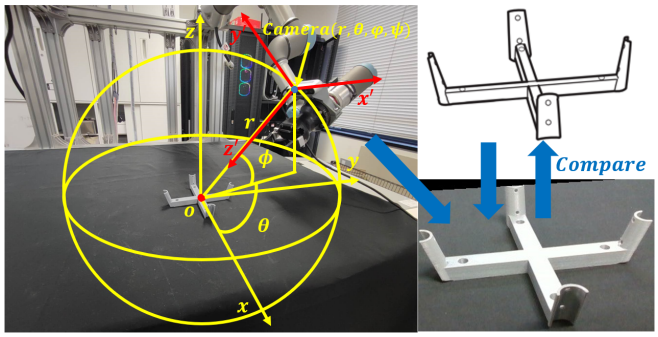


Fig. 6. Data collection for pose estimation where, with respect to the spherical coordinate system attached to the object, we generate random points with fixing the radius. We set the camera to face the origin of the spherical coordinate system. For each position, we take the image and record it with the camera's pose.

among the candidate templates using Equation 1.

$$\text{cost} = \alpha\ \text{cost}_1 + \beta\ \text{cost}_2, \tag{1}$$

where α and β are coefficients.

IV. Relative Pose Estimation

In this section, we explain a method for estimating the relative pose between two parts to be assembled from the instruction manual. We consider a situation in which two parts to be assembled are described in a single image of the instruction manual.

We first estimate the pose of each assembly part placed on the table by matching its image with the image drawn in the instruction manual, where its workflow is shown in Fig.5. Then, we identify the relative pose between two assembly parts. For the relative pose searching process, we assume that the initial poses of the components when unassembled are known. To search for the pose of an assembly part, we attach the spherical coordinate system to the assembly part. By fixing the radius, we generate uniformly distributed random points on the spherical surface. We obtain a set of RGB images assuming the camera is attached to each distributed point facing the center of the coordinate system as shown in Fig.6.

Here, as shown in Fig.6, we kept the camera's z' axis constantly pointing toward the sphere's center, and the camera's x' axis perpendicular to the zoz' plane for image sampling. Since assembly parts are usually drawn in a manual as placed on the horizontal plane, we can find the image taken by the camera, which is close to the image of assembly parts drawn in the instruction manual in most cases. However, there are some perspectives, e.g., a top-down view, where the image has to be rotated about the z' axis of the camera coordinate system. To cope with such a situation, we have added the rotation about the z' axis of the camera coordinate system to search for the pose of an assembly part. Therefore, when we successfully match the image taken at different locations on the sphere with the images in the instruction manual, we can get the angles the assembly part needs to rotate from its initial pose.

We now explain the algorithm for pose estimation. For the basic comparison algorithm, we still choose the FDCM algorithm. Even in cases of incorrect posture, the FDCM algorithm can still provide an evaluation of image similarity, and based on this, we can determine whether the correct relative posture has been found. Of course, this process requires search optimization. If we need to keep the error margin sufficiently small, we need to make our sampling interval small enough. However, this will lead to an increase in data volume and thus slow down the search process. To solve this problem, we mainly considered two solutions. The first idea is to use the gradient descent algorithm for the search. We uniformly deploy cameras on a sphere, mainly involving two variables: polar angle θ and azimuth angle ϕ. Besides, we also consider the rotation in the camera coordinate system; we give a variable ψ for the rotation in this flat. We perform gradient iteration for the cost calculated by FDCM in each variable direction to reach the final target. In practical implementation, this method has significant problems. First, parts have symmetry, so theoretically, there are multiple local minima and no global minima, which cause oscillation in the gradient descent process. Second, the cost calculated by FDCM does not change regularly with the variation of polar and azimuth angles; it is not linear, and even within a certain range without theoretical local minima, oscillation may occur. Based on these reasons, we adopted a combination of 'candidate' + 'voting system'. First, we use an object contour of a proportionally smaller size to compare with a template in the instruction manual. This step is much faster, and we can select a certain amount of relative posture candidates. Then, for each candidate's posture, we select a certain amount of points around it and the candidate itself, decide the weight based on distance, and conduct 'voting' to determine the final cost for each candidate. If we define the polar angle is θ, the azimuthal angle is ϕ, the camera coordinate system rotation angle to ψ, the amount of voting points to m, the distance between the voting point

Algorithm 1 Pose selection

initial candidate list C and the amount of posture is n.
the amount of candidate is $\sqrt{n}$, the amount of voting posture is m.
for each posture p **do**
  if $cost_p$ is the lowest $\sqrt{n}$ cost **then**
    Add posture p to candidate list C
  end if
end for
Initial $minCost \leftarrow \infty$
Initial $bestPose \leftarrow$ null
for each candidate posture $p \in C$ **do**
  Get the cost of the nearest m posture
  Get every point's cost by using Equation. 2 as $cost_{candidate}$
  if $cost_{candidate} < minCost$ **then**
    $minCost \leftarrow cost_{candidate}$
    $bestPose \leftarrow p$
  end if
end for

return $bestPose$

and candidate point is d, and $cost_s$ to represent the candidate point's FDCM cost, then the cost for the candidate can be represented like in Equation.2.

$$cost_d = \sum_{i=1}^{m} cost_i \cdot e^{-\lambda\sqrt{(\theta_s-\theta_i)^2+(\phi_s-\phi_i)^2+(\psi_s-\psi_i)^2}} + cost_s. \tag{2}$$

And the algorithm for searching is shown in the Algorithm 1. The greatest advantage of this algorithm is that it balances speed and accuracy. The candidate mechanism speeds up the search process, while the voting mechanism reduces the randomness of the FDCM algorithm results, making the algorithm robust.

After successfully matching the sampled photos with the component photos extracted from the assembly manual, we obtain the relative camera poses for that component in the given scene. Furthermore, we perform a corresponding sampling and matching process for the other components to obtain their respective camera relative poses in the scene. In this way, we have effectively quantified the relative poses of the components in the original assembly scene as previously mentioned. We achieved this by quantifying relative poses through image comparison. By comparing the differences in camera poses, we can determine how each component needs to change from its initial state to achieve the required pose for assembly. If after posture matching, the camera pose of each matched part is $(r, \theta_1, \phi_1, \psi_1)$ and $(r, \theta_2, \phi_2, \psi_2)$, then to achieve the relative pose described in the instruction manual the object need to make the rotation of $(\theta_1 - \theta_2, \phi_1 - \phi_2, \psi_1 - \psi_2)$. That's how we utilize the relative pose.

To evaluate the accuracy of the method, we primarily compare the differences between the actual images and the template images. We mainly use the Intersection over Union (IoU) to measure the differences in masks between the actual and template images, and use FDCM to measure the similarity of internal texture details. The reason we do not solely rely on masks for measurement is that our parts have rotational symmetry in their contours, whereas the assembly positions of the parts may not necessarily possess this property. Therefore, even if the contours are identical, it does not necessarily mean that the correct posture has been found, necessitating further evaluation by using FDCM.

V. EXPERIMENT

A. Assembly Parts Detection

We first verified the algorithm for assembly parts detection introduced in subsection III-B. The parts we used are shown in Fig.7. Cameras were set up at four different angles to take the image of the assembly part: three on the sides and one at the top, where examples of taken images are shown in Fig.8. For each assembly part, we collected twenty sets of image data to increase the diversity of the test data. We input each set of data including four images into the algorithm. Each set of images was tested for 100 times. We collected images with same conditions of lighting, temperature, and background. On the other hand, the camera parameters are remained constant. In addition, as well as the Gaussian noise reduction, we also applied the median filtering to remove the salt-and-pepper noise in the background. We set $\alpha = 0.6, \beta = 0.4$. Tables I and II show the statistical results of the parts detection without and with the candidate selection, respectively.

We conduct experiments in two scenarios. The first involves quickly selecting candidate templates and then performing a detailed analysis on them as mentioned in Section III-B. The second scenario involves conducting a FDCM+HT to the query edge directly. This comparison highlights the superiority of the rapid screening method we propose. We compiled statistics on accuracy, time, and robustness. For Table I, we use the image whose size is around 400x400 pixels, and without a candidate system, while for Table II, we scaled down the images proportionally to 150x150 pixels. And if the $cost$ of the matched object is defined as $cost_m$, and the $cost$ of the other templates smallest cost is $cost_o$, robustness is R, then the quantitative definition of robustness is as follows.

$$R = (\text{cost}_o - \text{cost}_m)/\text{cost}_m. \tag{3}$$

It was observed that the detection accuracy of part 7 is lower than others. This is because, as shown in Fig.7, the shape of part 7 is more complex than others. Since our algorithm is based on the chamfer distance, the matching cost is inherently higher for complex-shaped parts. In addition, there is a possibility that a simple image matches a part of a complex image. However, the other parts performed very well in our tests. At the same time, we can compare images

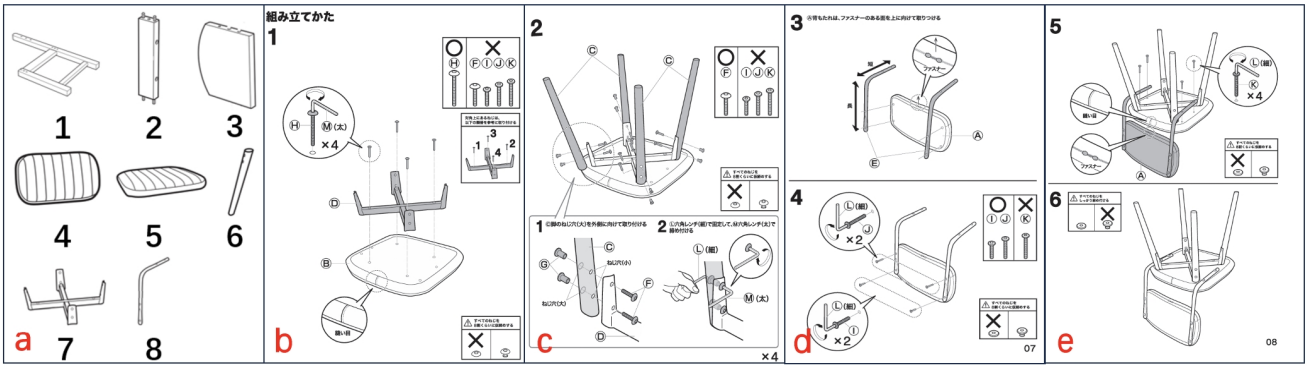


Fig. 7. These pictures are from a NITORI [2] chair instructions. Fig.a) described the parts we used in our detection experiment. Fig.b), c), d) and e) showcase the assembly issues we may encounter during the assembly process, including top-to-bottom assembly, assembly on inclined surfaces, and so on, posing significant challenges to assembly path planning.

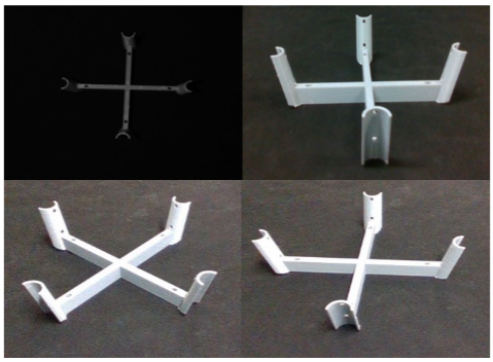


Fig. 8. The images taken from cameras which is set at different angle of the object.

TABLE I

DETECTION RESULT WITHOUT CANDIDATE SELECTION

Part	Accuracy	Robustness	Time(s)
1	94%	0.32	15.81
2	100%	1.06	15.85
3	98%	0.62	15.34
4	96%	0.61	15.35
5	100%	1.67	17.55
6	96%	0.99	16.67
7	92%	0.15	18.91
8	98%	0.84	16.33

of two different situations with a candidate system for selection or directly apply FDCM and HT to the template. Where with the candidate system, we use smaller-sized images for selecting candidates without losing much accuracy, greatly improving time efficiency. This is because the running speed of the FDCM algorithm we use is highly dependent on the size of the template and query.

For the object pose estimation experiment, we estimated the pose of assembly parts from a chair and utilized the obtained relative pose relationships to complete an assembly experiment. Since this process required sampling on a spherical surface, we mounted the necessary cameras on the robot's end effector and controlled the robot to achieve various poses for data collection.

TABLE II

DETECTION RESULT WITH CANDIDATE SELECTION

Part	Accuracy	Robustness	Time(s)
1	92%	0.44	4.12
2	100%	0.96	3.41
3	97%	0.44	2.23
4	95%	0.38	4.35
5	98%	1.24	3.55
6	97%	0.83	4.67
7	93%	0.71	3.61
8	96%	1.21	4.23

We used the UR3e robot. Due to the limitation of its motion range, we only generated camera position uniformly on one hemisphere of range $\theta \in [0, \pi]$ and $\phi \in [\pi/6, \pi/2]$. Practically, it is enough since the assembly parts used in this study have symmetrical shapes. We generated 90 uniform distributed positions. When measuring the accuracy, we calculated the values of IoU and FDCM cost for the selected images, where the IoU value ranges from $(0, 1)$, with larger values indicating closer proximity between the two, while the FDCM cost is preferred to be smaller.

In this experiment, we extracted the relative poses of two assembly parts from assembly scenes and applied them to actual assembly processes. We utilized these poses as intermediate ones for path planning. We conducted the experiment using a set of chair components as the assembly target.

To evaluate the experimental results, we directly compare the values of IoU of these two assembly parts after scaling them to the same size. In addition, we calculated the values of FDCM, to compare the similarity of the images.

We note that there still existed some discrepancies from the correct assembly poses during actual assembly processes. This is because increasing the amount of data collected would result in longer search times, and not all poses have inverse kinematic solutions for the robot.

B. Assembly Experiment

This experiment primarily showcases the practical application of the relative poses mentioned in Section V-A.

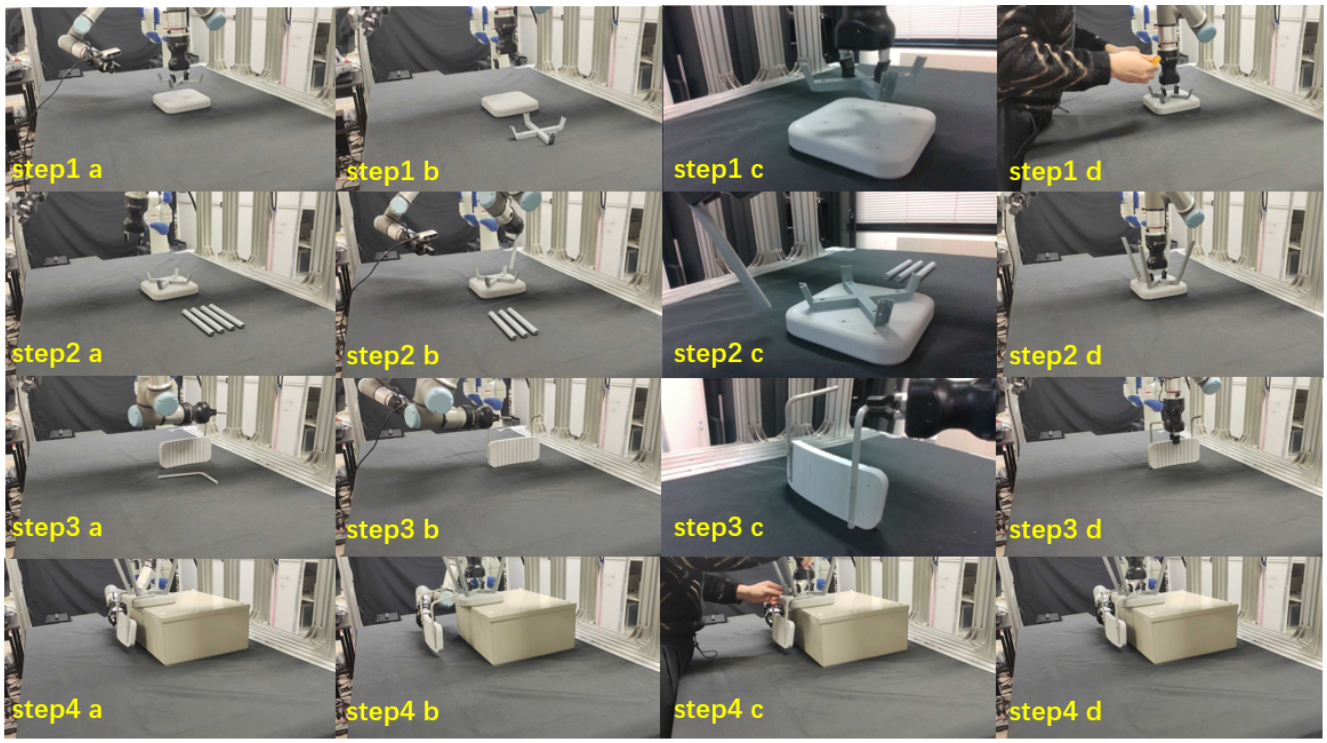
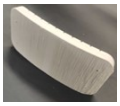
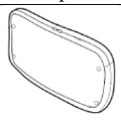
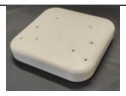
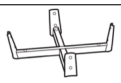


Fig. 9. The snapshot of collaborative assembly experiment between a human and a robot, where the robot was responsible for roughly moving the parts to the designated position, while a human was in charge of aligning the parts more accurately and assembling the screws. During this process, the robot continuously held the parts to keep them from moving, making it easier for humans to assemble the screws.

TABLE III
POSE ESTIMATION RESULT

Matched pose	Template	IoU	FDCM	Time(s)
		0.71	0.21	30.18
		0.64	0.26	32.61
		0.79	0.18	34.12
		0.78	0.17	28.87
		0.69	0.21	29.26

This is a human-robot collaboration assembly experiment where humans are responsible for screw assembly, while the robot is in charge of assembly planning and joining the parts together. In the assembly experiment, we applied the previously mentioned relative poses to the assembly process, utilizing this pose as an intermediate variable for assembly and performing path planning for the actual assembly process. In the collaborative assembly experiment, we assume that the grasping pose of an object is given.

However, relying solely on this pose is insufficient to obtain a plannable intermediate pose. For angles in the planning phase where the error with the actual planned pose exceeds 10 degrees, we conducted optimization. Finally, we used bounding boxes for the assembled objects based on their maximum width, length, and height for collision detection and intermediate path planning.

Once the intermediate relative poses were achieved, we could directly move in the corresponding direction based on the positional information obtained from the manual to complete the assembly.

Furthermore, in the experiment, the initial poses of real objects were known, and assembly hole information was obtained from a depth camera. We determined the final assembly poses based on the object's position and structural shape, as illustrated in Fig.9.

VI. CONCLUSION

In this study, we propose an method for aligning assembly parts based on the visual information guided by the image in a graphical instruction manual. Our approach involves the identification of furniture components and the utilization of relative poses in the assembly scenes depicted in the instructions. Our experiments validated the accuracy and robustness of the identification. Additionally, we successfully assembled a NITORI chair using the obtained relative poses.

For future research, we will focus on improving the accuracy of pose identification methods and testing the generality of the approach on a wider range of furniture items.

VII. ACKNOWLEDGEMENT

This paper is based on results obtained from a project, JPNP20006, commissioned by the New Energy and Industrial Technology Development Organization (NEDO).

REFERENCES

[1] IKEA, "IKEA furniture assembly environment - GitHub Pages." https://clvrai.github.io/furniture/.

[2] NITORI, "Nitori online chair instruction manual." https://www.nitori-net.jp/ecstatic/image/pdf/6620705.pdf.

[3] I. Sera, N. Yamanobe, I. G. Ramirez-Alpizar, Z. Wang, W. Wan, and K. Harada, "Assembly planning by recognizing a graphical instruction manual," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS)*, pp. 3138–3145, 2021.

[4] M.-Y. Liu, O. Tuzel, A. Veeraraghavan, and R. Chellappa, "Fast directional chamfer matching," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pp. 1696–1703, 2010.

[5] L. Ding, W. Jiang, Y. Zhou, C. Zhou, and S. Liu, "BIM-based task-level planning for robotic brick assembly through image-based 3D modeling," *Adv. Eng. Inform.*, vol. 43, p. 100993, 2020.

[6] J. Shu, W. Li, and Y. Gao, "Collision-free trajectory planning for robotic assembly of lightweight structures," *Autom. Constr.*, vol. 142, p. 104520, 2022.

[7] Y. Li, K. Mo, L. Shao, M. Sung, and L. Guibas, "Learning 3D part assembly from a single image," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, pp. 664–682, 2020.

[8] M. Heo, Y. Lee, D. Lee, and J. J. Lim, "FurnitureBench: Reproducible real-world benchmark for long-horizon complex manipulation," in *Proc. Robot.: Sci. Syst. (RSS)*, 2023.

[9] Y. Lee, E. S. Hu, and J. J. Lim, "IKEA furniture assembly environment for long-horizon complex manipulation tasks," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, pp. 6343–6349, 2021.

[10] M. Yu, L. Shao, Z. Chen, T. Wu, Q. Fan, K. Mo, and H. Dong, "RoboAssembly: Learning generalizable furniture assembly policy in a novel multi-robot contact-rich simulation environment," *arXiv:2112.10143*, 2021.

[11] O. Aslan, B. Bolat, B. Bal, T. Tumer, E. Sahin, and S. Kalkan, "AssembleRL: Learning to assemble furniture from their point clouds," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS)*, pp. 2748–2753, 2022.

[12] R. A. Knepper, T. Layton, J. Romanishin, and D. Rus, "IkeaBot: An autonomous multi-robot coordinated furniture assembly system," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, pp. 855–862, 2013.

[13] S. Park, J. Baek, S. Kim, and J. Park, "Rigid grasp candidate generation for assembly tasks," in *Proc. IEEE/ASME Int. Conf. Adv. Intell. Mecha. (AIM)*, pp. 589–594, 2020.

[14] S. Park, H. Lee, S. Kim, J. Baek, K. Jang, H. C. Kim, M. Kim, and J. Park, "Robotic furniture assembly: task abstraction, motion planning, and control," *Intelligent Service Robotics*, vol. 15, no. 4, pp. 441–457, 2022.

[15] E. Prati, V. Villani, M. Peruzzini, and L. Sabattini, "An approach based on vr to design industrial human-robot collaborative workstations," *Applied Sciences*, vol. 11, no. 24, p. 11773, 2021.

[16] F. I. Doğan, S. Gillet, E. J. Carter, and I. Leite, "The impact of adding perspective-taking to spatial referencing during human–robot interaction," *Robotics and Autonomous Systems*, vol. 134, p. 103654, 2020.

[17] A. Colim, C. Faria, J. Cunha, J. Oliveira, N. Sousa, and L. A. Rocha, "Physical ergonomic improvement and safe design of an assembly workstation through collaborative robotics," *Safety*, vol. 7, no. 1, p. 14, 2021.

[18] S. Qi, X. Ning, G. Yang, L. Zhang, P. Long, W. Cai, and W. Li, "Review of multi-view 3D object recognition methods based on deep learning," *Displays*, vol. 69, p. 102053, 2021.

[19] J. Tan, C. Wang, B. Li, Q. Li, W. Ouyang, C. Yin, and J. Yan, "Equalization loss for long-tailed object recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pp. 11662–11671, 2020.

[20] X. Sun, P. Wang, Z. Yan, F. Xu, R. Wang, W. Diao, J. Chen, J. Li, Y. Feng, T. Xu, *et al.*, "FAIR1M: A benchmark dataset for fine-grained object recognition in high-resolution remote sensing imagery," *ISPRS J. Photogramm. Remote Sens.*, vol. 184, pp. 116–130, 2022.

[21] X. Li, H. Wang, L. Yi, L. J. Guibas, A. L. Abbott, and S. Song, "Category-level articulated object pose estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pp. 3706–3715, 2020.

[22] F. Wei, X. Sun, H. Li, J. Wang, and S. Lin, "Point-set anchors for object detection, instance segmentation and pose estimation," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, pp. 527–544, Springer, 2020.

[23] G. Du, K. Wang, S. Lian, and K. Zhao, "Vision-based robotic grasping from object localization, object pose estimation to grasp estimation for parallel grippers: a review," *Artificial Intelligence Review*, vol. 54, no. 3, pp. 1677–1734, 2021.

[24] M. Imperoli and A. Pretto, "D^2CO: Fast and robust registration of 3D textureless objects using the directional chamfer distance," in *Int. Conf. Comput. Vis. Syst.*, pp. 316–328, 2015.

[25] P. Bao, L. Zhang, and X. Wu, "Canny edge detection enhancement by scale multiplication," *IEEE Trans. Patt. Ana. Mach. Intel. (TPAMI)*, vol. 27, no. 9, pp. 1485–1490, 2005.

[26] E. Dubrofsky, "Homography estimation," *Diplomová práce. Vancouver: Univerzita Britské Kolumbie*, vol. 5, 2009.

[27] J. Shi *et al.*, "Good features to track," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pp. 593–600, 1994.

[28] H. G. Barrow, J. M. Tenenbaum, R. C. Bolles, and H. C. Wolf, "Parametric correspondence and chamfer matching: Two new techniques for image matching," in *Proc.: Image Understanding Workshop*, pp. 21–27, 1977.