How AI and Robotics Can Build Furniture: A Case Study from the 2021 AI-Robot Assembly Challenge

Seongseop Yun, Myoung-Su Choi, Min-Young Cho, Keunhwan Kim, Dong-Hyuk Lee, Sewoong Jun, Ji-Hun Bae, and Dongjun Shin

Abstract—**The "Furniture Assembly AI-Robot Challenge 2021" is a competition that utilizes artificial intelligence (AI) and robots to assemble furniture and assess the quality of the assembly. To generate commands that a robot can execute for the assembly instructions, it is crucial to develop an AI-based algorithm to recognize and interpret the assembly process based on the provided instructions. The assembly robot must be dexterous and capable of safely executing assembly tasks without operator intervention. Before the assembly process, our team employed the Faster-region-based convolutional neural networks (Faster-RCNN) and the multi-object rectified attention network (MORAN) recognition methods to identify the assembly instructions, creating a connection relationship tree structure to interpret the recognized information. The robot utilized a developed multi-fingered gripper and a manipulation station to quickly and precisely complete the assembly task. Based on these exceptional results, our team was awarded first place, thus validating the adequacy of the proposed AI-robot system for complex furniture assembly tasks.**

I. Background

The field of robotics has been evolving owing to a rising demand for automated processes, but fully automated assembly remains a work in progress [1], [2]. Robots were initially introduced to unload aluminum casting machines on a Ford automobile production line. Gradually, robots have replaced humans in simple, repetitive tasks (i.e., pick and place, weld, paint) in dangerous environments due to increased degrees of freedom and payload. However, robots employed in assembly processes still require dexterous human assistance [3]. This is because realizing a fully automated assembly process with robots is challenging since parts must be placed in a specific location, making it necessary to have a designated environment.

As a partial solution to the challenge of atypical assembly environments, Nanyang Technological University (NTU) conducted a study on the automation of the assembly of IKEA's Stefan chair [4], [5]. The NTU team used two six-axis industrial robots and parallel grippers to connect and assemble wooden pins. They used bidirectional rapidly-exploring random tree technology to generate a collision-free assembly path and assembled the wooden pins using hybrid position/force control technology [5]. However, the NTU team only assembled a part of the chair using wooden pins, and the entire operation plan was hard-coded through engineering effort.

The "Furniture Assembly AI-Robot Challenge 2021" competition was held in Seoul, South Korea, to evaluate the performance of fully autonomous assembly robot systems. Two crucial assembly automation-related functions were evaluated. The first was the performance evaluation of the artificial intelligence (AI)-based algorithms that can automatically recognize and interpret assembly instructions to generate an assembly sequence without human intervention. The second was evaluating how well a robot system executed the entire assembly process without human assistance under atypical assembly conditions where many connector parts (pins, screw bolts, brackets) were cluttered.

Participating teams were assigned two missions to evaluate their assembly automation skills in the competition. The first mission was to assemble IKEA's Stefan chair from start to finish by following the original assembly instructions. This mission focused on judging whether the entire assembly process could be completed by manipulating the robot system without any human intervention. The second mission was to use the parts of IKEA's Stefan chair but assemble the Stefan chair in a deformed shape rather than the original shape. The parts were identical, but the assembly instructions were modified to create a deformed Stefan chair. The participants had to use an AI-based algorithm to understand the modified instructions and generate the correct assembly sequence for the robot system to execute.

Our team (Team SK^2Y) won first place in this competition due to our effective use of an AI-based algorithm and a novel robot system for the assembly process. The faster-region-based convolutional neural networks (Faster-RCNN) and the multi-object rectified attention network (MORAN) algorithms were applied to recognize the assembly instructions, and a connection relationship tree algorithm was developed to interpret the recognized commands. This AI-based algorithm quickly generated commands that the robot system could execute. Additionally, the implementation of multi-fingered grippers and a manipulation station provided additional advantages during the assembly process, improving assembly time and convenience. Owing to the use of these two unique technologies, the assembly process was successfully executed with high precision and safety, and our team outperformed the others with exceptional execution time.

II. Furniture Assembly AI–Robot Challenge 2021

A. Competition Schedule and Rules

The "Furniture Assembly AI-Robot Challenge 2021" evaluated the combine AI's recognition and interpretation ability with robot automation assembly technology. Four teams participated in the competition, and each team developed its own

Date of current version: 28 February 2023

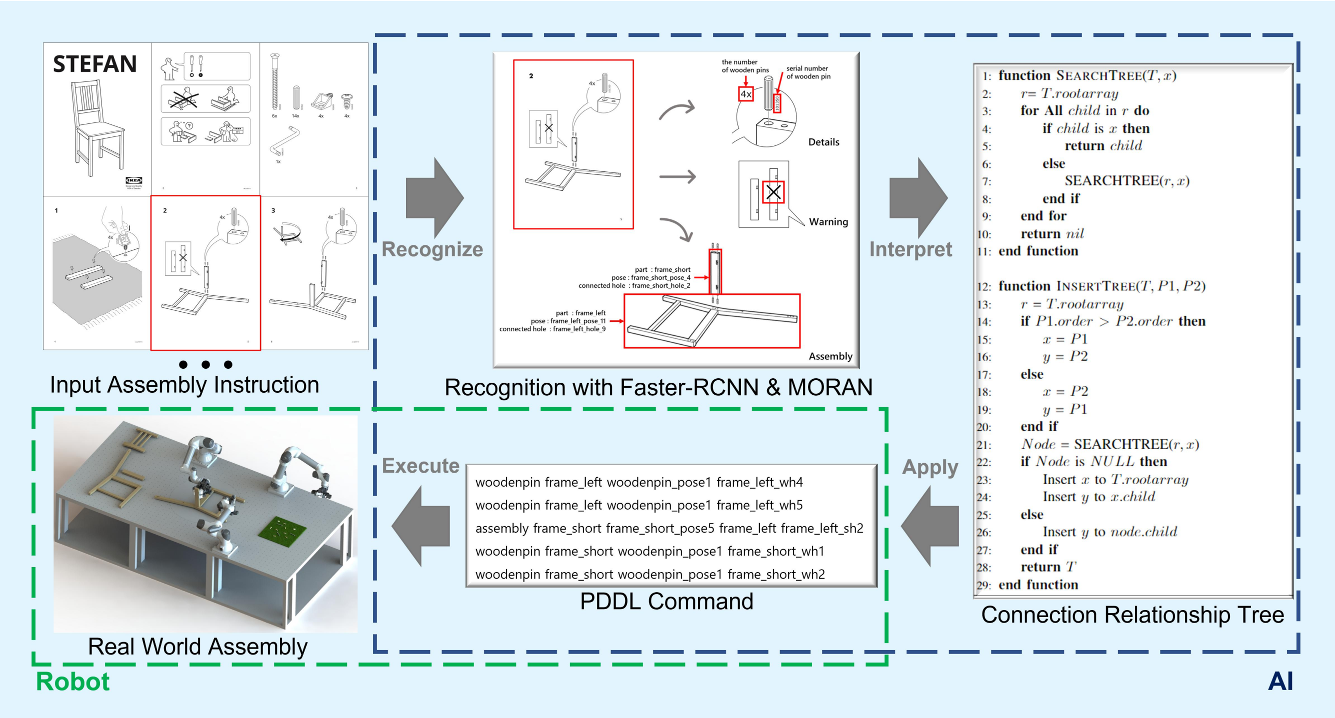

Fig. 1: Whole working process. We developed an artificial intelligent (AI) algorithm that can recognizes the essential information in the assembly instructions, creates a connection relationship tree, and then converts it into a planning domain definition language (PDDL) command. The robot assembly system received the PDDL command and executed the assembly task in the real world.

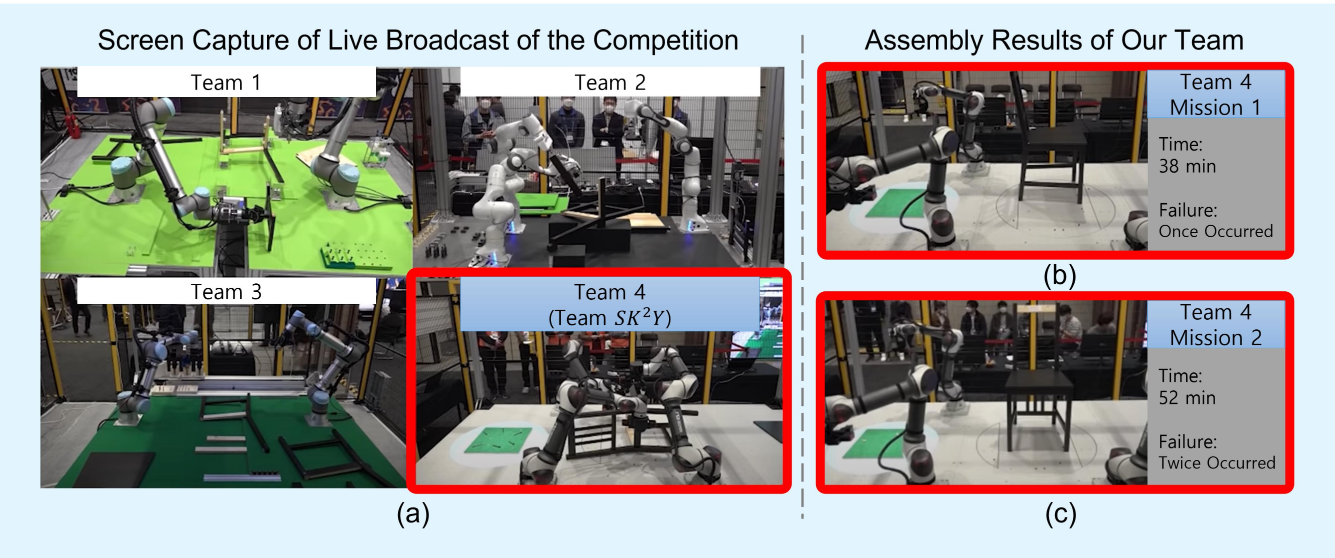

Fig. 2: (a) Screen capture of live broadcast of the competition (Team SK^2Y indicated by the red boxes). (b) Result of the first mission. Assembly was completed in 38 min without human intervention. (c) Result of the second mission. Assembly was completed in 52 min with only one intervention.

system during the two-year preparation period. The event was originally scheduled for 2020, but it was moved to March 2021 due to the COVID-19 pandemic. This competition was held at COEX in South Korea, and after a three-day installation and operation test period, it was held offline on the fourth day and was broadcast live online. Fig. 2(a) shows a screen capture from the live broadcast on competition day. The competition ran all day, with the first mission in the morning and the second in the afternoon.

The principal regulations related to the progress of the competition were as follows. The worktable on which the chair was to be assembled needed to be configured within an area of 2.5 $m \times$ 2.5 m. Two or more vertical articulated robots were to be used for assembly, and the type of robot could be freely selected. In addition, the camera required for recognizing the assembly process could be installed freely. Before assembly, the parts to be assembled were voluntarily placed by the team and inspected by the judges. A team that wants to gain additional kitting points applied a kitting action. The team that applied for the kitting action was given a 400 $mm \times$ 400 mm connecting part plate with connecting parts randomly placed by the judge. Before the mission began, a bitmap file with the assembly instructions was via USB, and immediately after that, the assembly instructions had to be recognized and assembled within 90 min.

The evaluation index of the competition consisted of 86 points, including parts fastness achievement (46 points), completeness (20 points), and speed (20 points). One judge was in charge of one team, checking the progress of each stage and scoring the parts fastening achievements. For the completeness parameter, the judges evaluated the assembly capability of the robot. If a defect occurred during assembly, the operator intervened and corrected it. The completeness score was subtracted in proportion to the number and duration of operator interventions. The speed score was evaluated for the assembly time, and the speed score was subtracted over time from 45 minutes from the start of assembly.

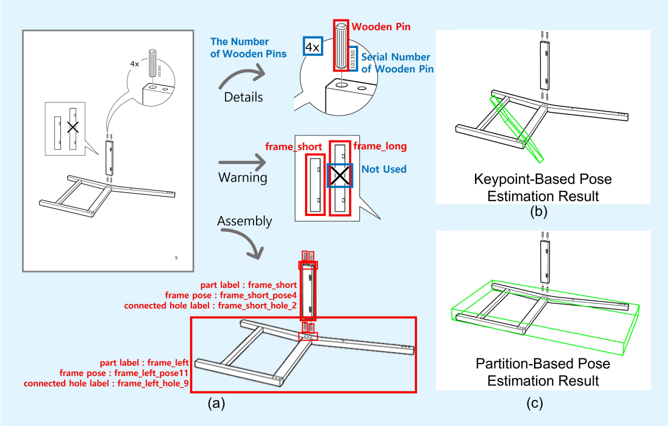

Fig. 3: (a) Recognition result of assembly instructions. We extracted the visual information of the assembly instructions using the faster-region-based convolutional neural networks (Faster-RCNN) and the multi-object rectified attention network (MORAN) recognition technology. In addition, we utilized a novel pattern-based pose estimator that predicts the pose of parts for the action estimation task. (b) Comparison results of pose estimators. The conventional keypoint-based pose estimator does not recognize the posture of the left frame, but the proposed estimator based on the partition correctly recognizes the posture of the left frame.

B. Competition Results

Our team (Team SK^2Y) was awarded first place to complete both missions with the least number of interventions among the four participating teams. The competition consisted of mission 1, in which the chair was to be assembled following the original assembly instructions and mission 2, in which the chair was to be assembled by following the modified assembly instructions. The competition results are shown in Figs. 2(b) and 2(c). In the first mission, all processes, except for the last short screw bolting, were performed without operator intervention, and the assembly was completed in 38 min. In the second mission, an additional intervention was required owing to a failure to recognize the position when fastening the screw bolts. The second mission was completed in 52 min.

III. AI-BASED ALGORITHM FOR READING AND COMMANDING ASSEMBLY INSTRUCTIONS

In Section III, we introduce an algorithm that converts the IKEA Stefan chair assembly instructions for humans into assembly commands for robots. The IKEA instructions describe the connections between the frame parts by using visual elements such as part diagrams, arrows, and highlights. These elements do not provide numerical data. Therefore, it is necessary to convert the visual information into instructions for robots.

We generate assembly commands for robots using AI-based algorithms. We divided the process into three stages: recognition, analysis, and application. In the recognition step, the visual instructions in two Dimensional assembly manual are extracted by describing information such as the poses of the parts and the relationships between parts. In the analysis step, a connection relationship tree is generated using the visual information on each instruction page. This tree contains the assembly information as a hierarchy of parts. Finally, in the application step, a planning domain definition language (PDDL) is generated using a connection relationship tree. Based on the hierarchy information presented in the connection relationship tree, the PDDL expresses an assembly process, such as the order of assembly of parts, connections of connectors, and assembly method between parts.

A. Recognition of Assembly Instructions

In this section, we introduce a recognition framework that can read and understand the required assembly actions from the assembly instruction. As illustrated in Fig. 3(a), we split the assembly instruction into three individual functional regions: details, warnings, and assembly regions. The details region contains items and explanatory text. The warning region helps with the selection of the correct parts when similar parts exist. The assembly region defines the assembly actions required for the current frame.

To understand an image, the proposed recognition framework is divided into two tasks: 1) Detection task: detect components and object parts and the number of components and parts; and 2) Action estimation task: estimate assembly actions between components and identify the parts required for the current frame. To achieve the detection task, we combined the object detection, text recognition, and detection correction approaches.

We adopted Faster-RCNN of the Detectron2 framework [6] to locate and label all items, including speech balloons, text, and cross signs [7]. The training and testing datasets were generated as follows: 1. We collected text and signs, diagrams, speech balloons, and cross signs from IKEA product manuals available on IKEA's homepage and augmented them through rotation, scaling, and flipping; 2. We rendered and automatically generated Stefan's parts and their labels from the 3D model created using Blender [8]. Consequently, to distinguish the items in the same category but having different serial numbers, we employed MORAN text recognition [9] and adopted the checkpoints trained by the authors to convert text regions of interest (ROI) into item serials. To correct the part labels, we examined the item shapes in the warning region.

For the action estimation task, poses of the parts in assembly regions are required and should be estimated. We used several keypoint-based pose estimation methods [10], [11], only to realize that these methods failed to recognize low-texture objects, such as drawing objects in IKEA's manuals. The pixel-wise information the IKEA manual is inadequate to recognize poses. To solve this issue, we have proposed a novel partition-based pose estimator to predict the poses of parts. It is possible to use structural information of the object rather than pixel-wise information because divided parts contain structural features, such as holes, outlines, and angular shapes, rather than keypoints of pixels. We detected several object partitions, which were easier to recognize compared to the corresponding keypoints. Fig. 3(b) shows a comparison of the results obtained using the conventional pose estimation method and the proposed novel partition-based pose estimation method. From the estimated poses, the locations of connectors on the parts could be calculated. In the last step, we examined all connector pairs and selected the pair with the minimum distance between connectors.

B. Connection Relationship Tree Inference

In the analysis step, we combined the relative connections of the recognition step into one object. For efficient inference, we proposed a connection relationship tree by using the recognition results. The visual elements on each page explain how to assemble the frame parts hierarchically. The tree structure is appropriate for expressing hierarchical characteristics because it is a set of nodes connected by parent-child relationships without cycles. These tree properties are advantageous for analyzing the instructions of assembly manuals. To generate a connection relationship tree, we used the recognition results, such as frame parts, poses, and connection positions. However, these data were inadequate to perform simulation in 3D space. To solve this problem, we used STEP files to increase the accuracy of 3D information. The recognition result indicated the pairs of frame parts to be assembled and the connection position. The connection positions for calculating the frame poses were inferred from the STEP files. The positions were limited to cylindrical holes in which connectors were inserted. The hole data included the position, radius, normal vector, and type of suitable connector. The position was estimated as the origin of the CIRCLE and CYLINDER coordination, and the radius and normal vector were calculated using the CIRCLE data from the STEP file. In addition, the type of connector part

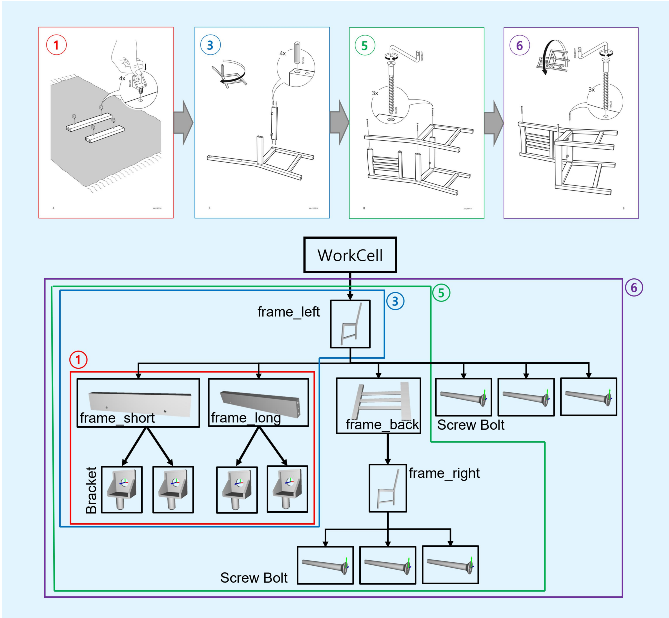

Fig. 4: Structure of generated connection relationship tree of Stefan chair. The AI-based algorithm expands the connection relationship tree structure by obtaining additional information from the assembly instruction manual.

was inferred from the hole radius. A connection relationship tree is generated by using the 2D manual and the STEP files. In this connection relationship tree, a node was defined for each part described in the instructions. All parts were added to the tree as nodes, except for the wooden pins (Fig. 4). A node contained its parent node, an affine matrix representing the parent node's coordinate system, and a set of child nodes. The connection relationship tree T_0 had only the workspace as the root node in the initial state. If connection information existed on the next page, the parts were added as nodes at T_0. The method of inserting nodes into T_n is described in Algorithm 1. The information for connecting Part A with Part B consisted of frame parts, frame poses, and connection positions of parts. First, large or assembled parts were assigned to the parent node and the other parts to child nodes. When tree T_n had a parent part as a node, the child node was inserted into the parent's child in the tree. If there was no parent node in the tree T_0, we added a parent part to the child of the root node. Then, the child node was added to the child of the parent. In addition, T_1 was completed by adding the relative 3D structure value between the current child part and the parent node to the child part of the parent part. The relative 3D information between the parent node and the child node was used to calculate the rotation and translation required when the child nodes were connected in the coordinate system of the parent node. This method converted the coordinate system of the child frame part to the connection position of the parent frame part. This relative 3D information was acquired using the hole connected to the parent part and the hole connected to the child part obtained from the recognition part. A rotation matrix was obtained using the normal vectors n_c, n_p of each hole, as in eqn. (3).

Algorithm 1 Insertion of parts in connection tree

function SEARCHTREE(T, x)
  r= $T.root$
  for All $child$ in r **do**
    if $child$ is x **then**
      return $child$
    else
      SEARCHTREE(r, x)
    end if
  end for
  return nil
end function

function INSERTTREE($T, P1, P2$)
  r = $T.root$
  if $P1.order > P2.order$ **then**
    x = $P1$
    y = $P2$
  else
    x = $P2$
    y = $P1$
  end if
  $Node_P$ = SEARCHTREE(r, x)
  if $Node_P$ is $NULL$ **then**
    Insert x to $T.root$
    Insert y to $x.child$
  else
    $Node_C$ = SEARCHTREE(r, y)
    if $Node_C$ is $NULL$ **then**
      Insert y to $Node_P.child$
    else
      Insert y to $x.child$
      Delete y in $r.child$
    end if
  end if
  return T
end function

$$v = -n_\mathrm{c} \times n_\mathrm{p} \tag{1}$$

$$[v]_\mathrm{x} \stackrel{\mathrm{def}}{=} \begin{bmatrix} 0 & -v_3 & v_2 \\ v_3 & 0 & -v_1 \\ -v_2 & v_1 & 0 \end{bmatrix} \tag{2}$$

$$R = I + [v]_\mathrm{x} + [v]_\mathrm{x}^2 \frac{1 - n_\mathrm{c} \cdot n_\mathrm{p}}{\|v\|^2} \tag{3}$$

After applying the rotation matrix to the child frame part, the translation was the difference between the position of the parent frame part hole and the position of the rotated child frame part hole. Although several subtrees were created, this process was applied to all steps to obtain one connection tree (Fig. 4) in which all frames were defined as parent–child relationships. By repeating the algorithm, we obtained a connection relationship tree using all parts. When the tree was complete, all of connection positions were converted to root coordinates for simulation. In addition, the position of the wooden pin, which was excluded from the analysis, was calculated from the completed tree. First, we calculated the distances of all hole pairs. If the distance of a hole pair was 0, the wooden pin was added as the child node of the parent.

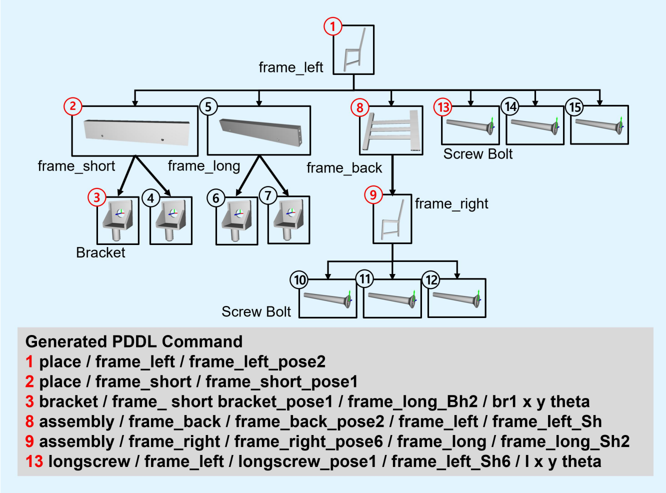

Fig. 5: Example of generated PDDL command. The AI algorithm developed herein generates an assembly sequence from the connection relationship tree.

C. Generation of PDDL commands

PDDL is a standard language for efficiently defining a robot's action plan, and it consists of objects and predicates [12]. In furniture assembly, robots repeatedly execute the assembly process with set motions. The repetitive motions of the robot should be effectively defined, and the assembly object should be accurately marked. For this purpose, PDDL was used to simplify the instructions and accurately transmit assembly information. In this study, commands transmitted to the robot were generated from the analyzed connection tree by using a modified form of PDDL.

The modified PDDL consisted of an object, including its posture and position, and a predicate classified into three actions. The object represents the name of the frame part with individual identity. The posture, a component, is displayed by dividing the posture in the three-dimensional space when viewed from the reference frame direction. The position indicates the hole label at the site where assembly occurs. The predicate indicates the action to be performed and consists of a place command to move the frame part, an assembly command to assemble two frame parts, and a connector command to join a connector part to the frame part.

In the connection relationship tree created in Section III-B, the connection relationships of all frames were divided into layers based on the top node. Therefore, the PDDL command is generated by searching from the upper layer to the lower layer (See Fig. 5). If there was a connector part in the child frame part, the connector command was executed first and then, the assembly command was executed. Commands at the same layer level took precedence over commands where the robot arm could assemble quickly based on the top frame

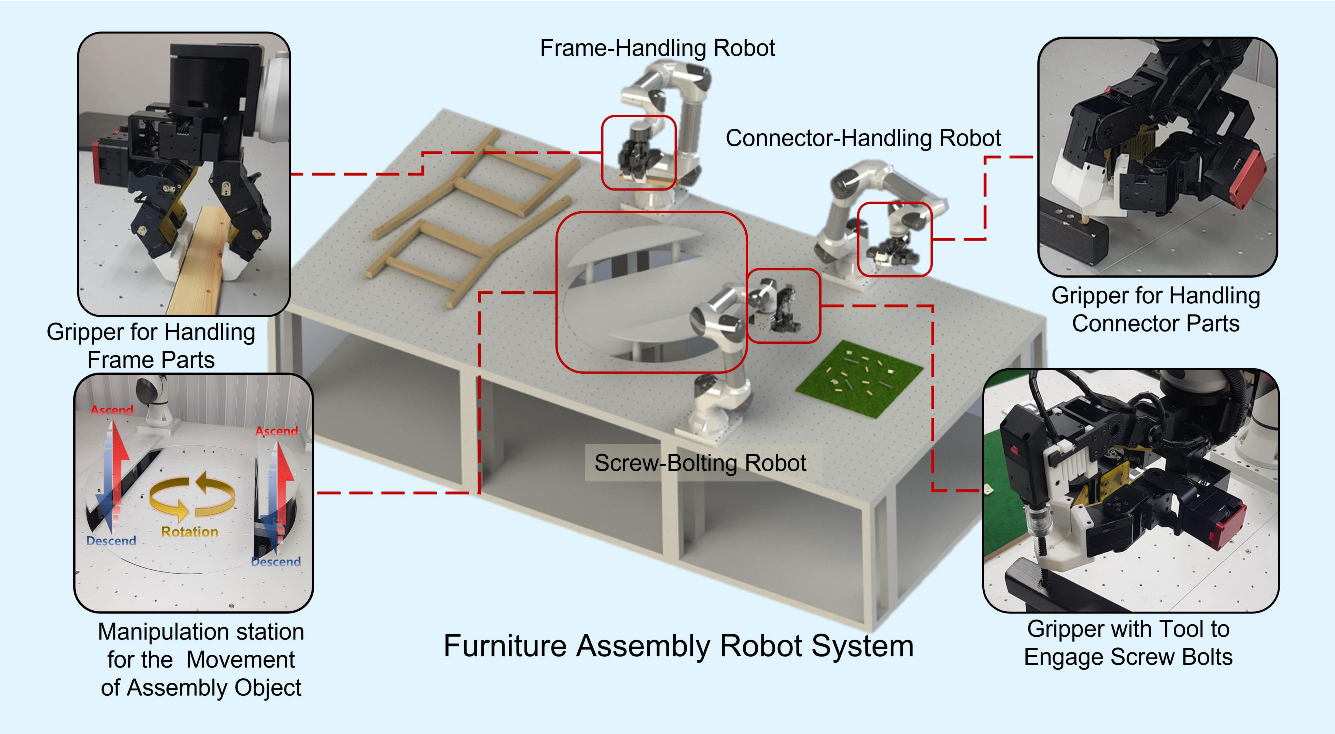

Fig. 6: Furniture assembly robot system. It consists of three robot manipulators, multi-fingered grippers, and a manipulation station. Multi-fingered grippers have different fingertip shapes depending on their roles. The manipulation station changes the position and rotation of the assembly, thus creating a good assembly posture.

coordinate system. Finally, the generated PDDL command was transmitted to the robot system to perform assembly work.

IV. Robot System Assembling Furniture

In this section, we introduce the robot's system and the technology used for furniture assembly. The specifications of the assembly system used in the competition and the overall assembly process are explained. Moreover, we introduce multi-fingered grippers and the manipulation station, which are key technologies for successful assembly. The multi-fingered gripper was used for handling the connector parts and frame parts dexterously. The manipulation station was used to move the assembly position, thus allowing the robot to perform assembly with a better posture.

A. Description of Furniture Assembly Robot System

We configured a robot system for assembling the furniture as one system on an assembly table (Fig. 6). We divided the assembly table into spaces in which frame parts were placed, assembly was executed, and connector parts were placed. The center of the assembly table consisted of three robot manipulators, multi-fingered grippers, and a manipulation station. The frame and connector parts were placed on both sides of the assembly table. We placed a heavy and bulky frame part on one side and installed a camera for recognition, which included a plate on which the connector parts were placed, on the other side. A detailed description of the drive mechanism used for assembly is as follows.

1) Robot manipulator: We used three RB5-850 robots (Rainbow Robotics Co., Ltd.) as macro manipulators. The RB5-850 robot comprises a rotation joint with six degrees of freedom, and it is controlled based on positional control. The payload of the RB5-850 is up to 5 kg. The RB5-850 robots were placed at the corners of the manipulation station according to their individual roles in the assembly.

2) Multi-fingered gripper: We developed a multi-fingered gripper that dexterously controls objects at the robot's end-effector by using XH430-V350 motors (Robotis Co.) [13]. The gripper has 11 degrees of freedom with three fingers. We used different types of grippers depending on the purpose of the task, including a gripper for handling furniture frames, one for fastening screw bolts, and one for handling connector parts such as wooden pins. The joint structures of all three grippers were identical, and only the fingertip shapes were different. For example, in the case of a fingertip designed to fasten screw bolts, the mechanical part and actuator necessary for fastening screw bolts were added [14]. The fingertips for grasping the frame parts were wide to realize high-friction gripping of heavy objects, and the fingertips for gripping connector parts had a groove at the tip to achieve superior gripping of small objects. We combined each gripper and robot according to their roles and called them frame-handling, connector-handling, and screw-bolting robots.

3) Manipulation station: At the center of the assembly table, we embedded a manipulation station with three degrees of freedom to control the assembly object. The diameter of the circular manipulation station was 800 mm; it had a rotational

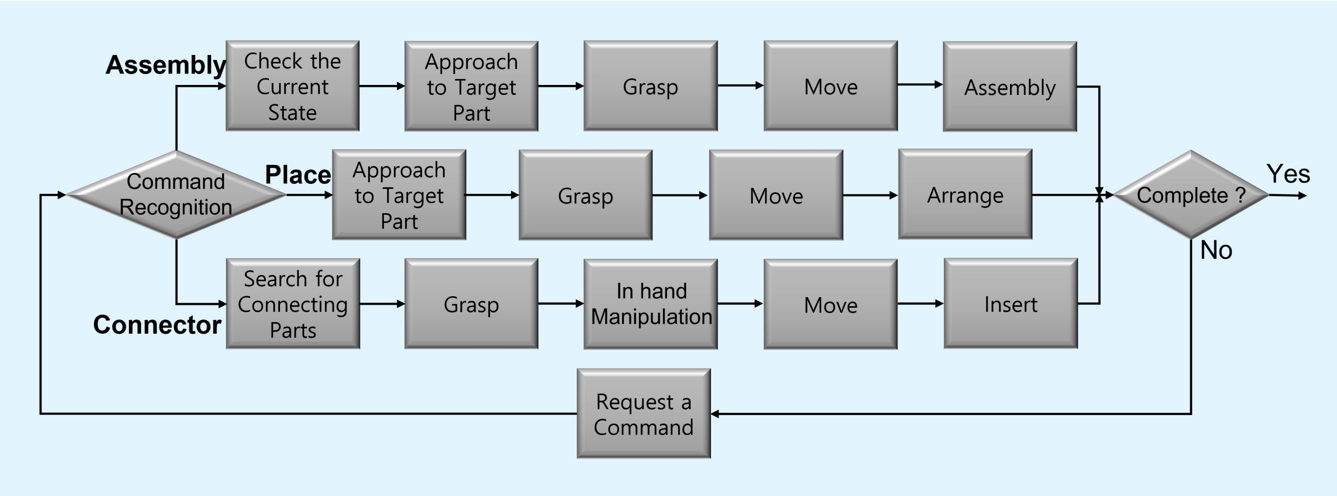

Fig. 7: Assembly flow chart. The robot system recognizes PDDL commands to distinguish assembly, place, and connector. Assemble is executed according to the control flow of each command.

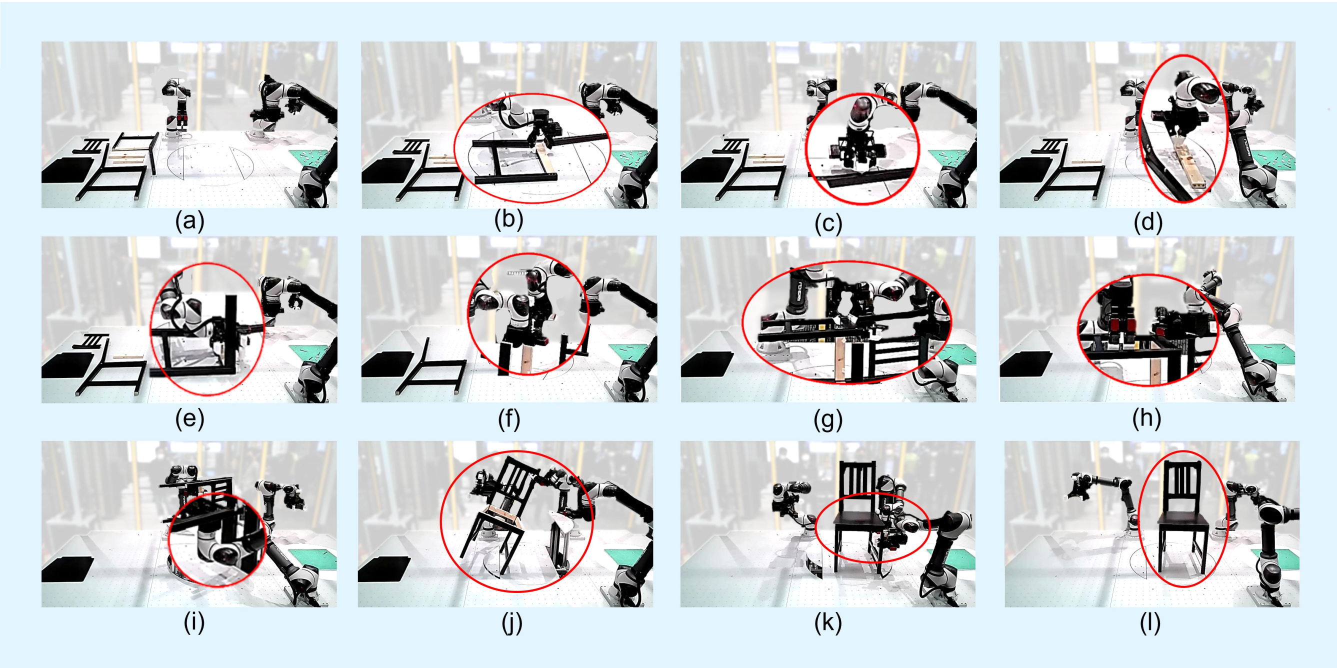

Fig. 8: Assembly process. (a) Initial state. (b) Move base part to the assembly workspace. (c) Wooden pin assembly. (d) Bracket assembly. (e) Frame assembly. (f) Wooden pin assembly. (g) Frame assembly. (h) Upper screw bolt assembly. (i) Lower crew bolt assembly. (j) Erect chair. (k) Fit seat frame. (l) Assembly is complete.

degree of freedom from $+180$ degrees to -180 degrees and vertical degrees of freedom on both sides. The manipulation station made it easier to handle large objects such as chairs. Moreover, the manipulation station allowed the robot to move into a position that was easier for performing assembly tasks, thus increasing the efficiency of assembly operations.

B. Assembly Flow and Process

1) Assembly flow: The assembly flow of each stage is outlined in Fig 7. The main controller of the furniture assembly robot system received PDDL commands from the AI-based algorithm. After recognizing the place, connector, and assembly commands corresponding to the predicates of the PDDL commands, the controller executed the process called by each command.

The place is a command related to movement of the frame part to the designated position. When the place command was input, the robot moved, grabbed the target frame, moved again, placed the frame at the designated location, and aligned it. The connector command inserted the connector parts such as wood pins, brackets, and screw bolts into the frame. The camera installed on the connector part plate acquired the position and rotation of the targeted connector part, and the connector-handling robot moved and grabbed the target part.

After aligning the parts through in-hand manipulation, the robot approached the target insertion location and executed the insertion operation. The assembly command drove the task of combining the frame parts. After receiving the assembly command, the controller checked the current assembly state, determined the location at which the assembly was to be performed, grabbed the frame, moved, and assembled it. Based on the assembly state, the three robots moved cooperatively to stably fix, align, and combine the frame. When the process underlying each command was completed, the controller checked the completion. If it was not yet complete, they requested the following command from the AI. In the case of mission 2, in which the assembly shape was different from that in mission 1, the assembly location was different, but the assembly flow was identical. Scheduling and robot control at each stage were performed manually. If a failure occurs, the operator stops the assembly operation, restores it, and proceeds with reassembly.

2) Assembly process: Because assembly was performed by receiving PDDL commands, the assembly process was determined based on the structure of the connection relationship tree generated by the AI-based algorithm. When commands were generated from the upper node of the connection relationship tree, in actual assembly, the assembly proceeded in the form of stacking up from the base frame. This assembly process was the same in missions 1 and 2. In mission 2 as well, where a deformed shape was assembled, the overall assembly process proceeded similarly, even though the assembly position or posture was slightly altered.

The entire assembly process proceeded in the same order as that in Fig. 8. The first action was to move the left frame, which became the base, to the workspace and align it (Fig. 8(b)). Thereafter, a wooden pin was inserted to keep assembly fixed in the temporarily assembled state in the left frame (Fig. 8(c)). After inserting the wooden pins, the movement and assembly operations proceeded for the long frame, short frame, and the back frame supporting the side (Figs. 8(d)–(f)). After inserting a wooden pin back into the frame supporting the side, the right frame was assembled, which was the opposite side of the base (Fig. 8(g)). This state was the temporary assembly state before the screw bolts were fastened.

Screw bolting was required to complete the assembly, and three screw bolts were fastened in the right and left frames (Fig. 8(h)–(i)). The screw-bolting robot applied force to the right frame from the top down, in the direction of gravity. The left frame was lifted by the manipulation station, and force was applied from the bottom up. Finally, the chair was erected, and the seat frame was fitted to complete the assembly (Fig. 8(j)–(l)). The robots and the manipulation station worked together quickly and stably in all assembly processes.

C. Vision Recognition and Connector Part Selection Method

As shown in Fig. 9, the connector parts were scattered in the designated space, and the robot could grip the connector parts necessary for each step. When the gripper grips the connector parts, it moves only by a certain depth to the connector parts plate. However, it is necessary to estimate the 2D position and orientation of the connector parts. To this end, the 2D positions and orientations of the connector parts are estimated using the camera mounted atop the assembly table [15]. The camera coordinate system was converted to the workspace coordinate system by using projective transformation so that the robot could grip the connector parts recognized by the camera. For projective transformation, ArUco markers [16] were installed on the outer corner of the connector part plate. The image was transformed to a size proportional to the actual distance based on the position of each marker. Then, by using the semantic segmentation algorithm mask regions with convolutional neural network (Mask R-CNN) [17], the connector parts required for each step were identified, and the position and orientation of connector parts were estimated.

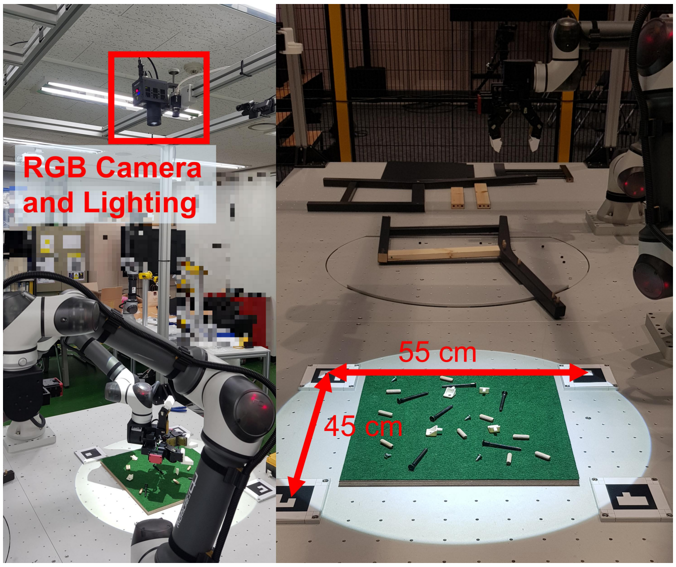

Fig. 9: Environment setup for vision camera and connector part plate.

The gripping position is the center point of each connector part mask, which is the output of Mask R-CNN. Because the image pixel coordinates in the projective transformed image are proportional to the workspace coordinate system, the gripping position can be used easily in the robot coordinate system. The connector parts such as wooden pins and screw bolts used in the furniture assembly are small and long. In addition, because the screw bolt is divided into the head and the body, the robot arm must be aligned in the correct gripping direction to hold the screw bolt and then perform the fastening operation. A principal component analysis (PCA) was performed to estimate the orientation of the connector parts. If the principal axis vector was estimated by applying PCA to the mask of each connector part, the shape characteristics of the connector parts were reflected, and the proper gripping orientation could be estimated.

By selecting a connector part that the gripper could grip easily, we were able to reduce assembly failures among the several scattered connector parts. To increase the gripping success rate, we applied the connector part selection method. The selection method started with selection of the part closest to the robot base (connector-handling robot or screw-bolting robot) that would grab the part. Thereafter, the method checked

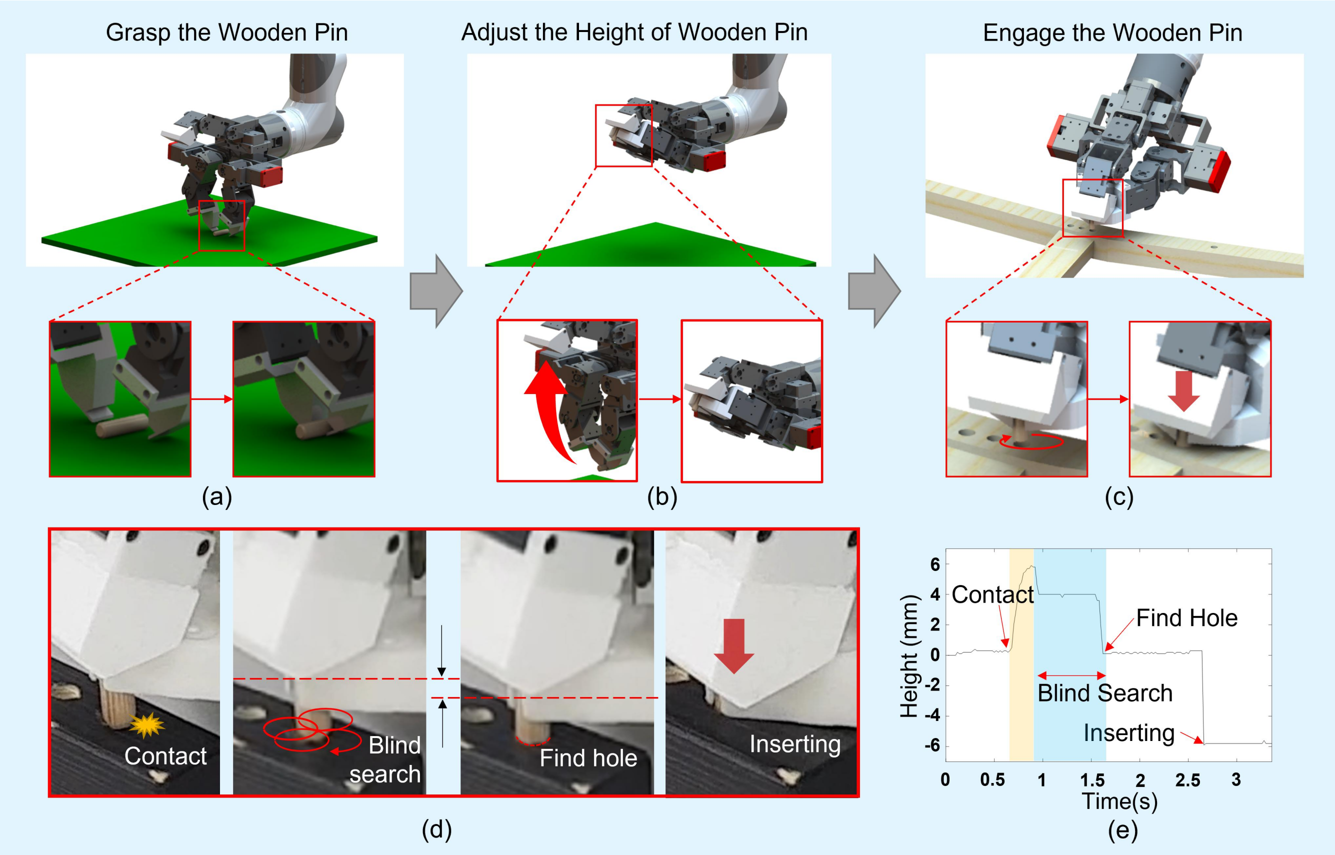

Fig. 10: Gripper strategy. (a)–(c) use of multi-fingered gripper to hold and align the wooden pin and proceed continuously until insertion. (d) Insertion process of (c) in detail. (e) A graph showing the change in the height of the middle finger during the insertion process.

whether the centers of the other connector parts were within a threshold radius from the center of the selected part. The threshold radius was selected such that it did not interfere with gripping, and if there were other parts within the radius, the probability of grip failure increased. If another part was within the radius, the method selected the next part and checked if it overlapped again. In addition, if there was no selectable part in the robot's workspace, the opposite robot moved the part into a position from where the connector part could be grabbed.

D. Gripper Motion for Kinesthetic Sensing-based Assembly

In this study, we used a multi-fingered gripper to rapidly assemble furniture parts. The multi-fingered gripper can perform in-hand manipulation on an aligned object without executing a kitting task, and the assembly state can be estimated by observing changes in finger movements.

In this paper, we describe an example of a wooden pin assembly. To perform this wooden pin assembly task, the robot must know the position of the wooden pin. To this end, a ceiling-mounted camera was used to recognize the position of the wooden pin. Moreover, to reduce positional uncertainty, it was necessary kitting task the wooden pins. However, a kitting task requires additional working time and working space. Herein, we aligned the wooden pins by means of in-hand manipulation, as shown in Figs. 10 (a)–(c), without performing a kitting task.

The process of aligning the position of the wooden pin with multiple fingers is as follows. First, the wooden pin is grasped with two fingers. Next, the height of the wooden pin is adjusted using the other finger. Because alignment is performed using only fingers, the wooden pins can always be aligned in the same position, even if the recognition result is inaccurate. After grasping the wooden pin, it is aligned with the hole position for assembly. At this time, contact occurs between the wooden pin and hole surface, and the height of the wooden pin calculated using the finger position increases marginally.

We conducted a blind search to find the hole position quickly. A blind search is a method of finding the hole position and moving an object along a predefined path until the object is combined [18]. In this paper, we found the hole position by creating a path that draws a circle once in each search. After all four searches, the diameter of the circle was gradually increased to continue the search. For this task, kinesthetic sensing [19] was used to find the moment at which the hole position was found. Kinesthetic sensing is a method of judging the assembly state by observing the movement of a finger (Fig. 10(d)). In this study, to identify the position of a hole by conducting a blind search for assembly, a wooden pin was pressed with a finger. As soon as the hole position was

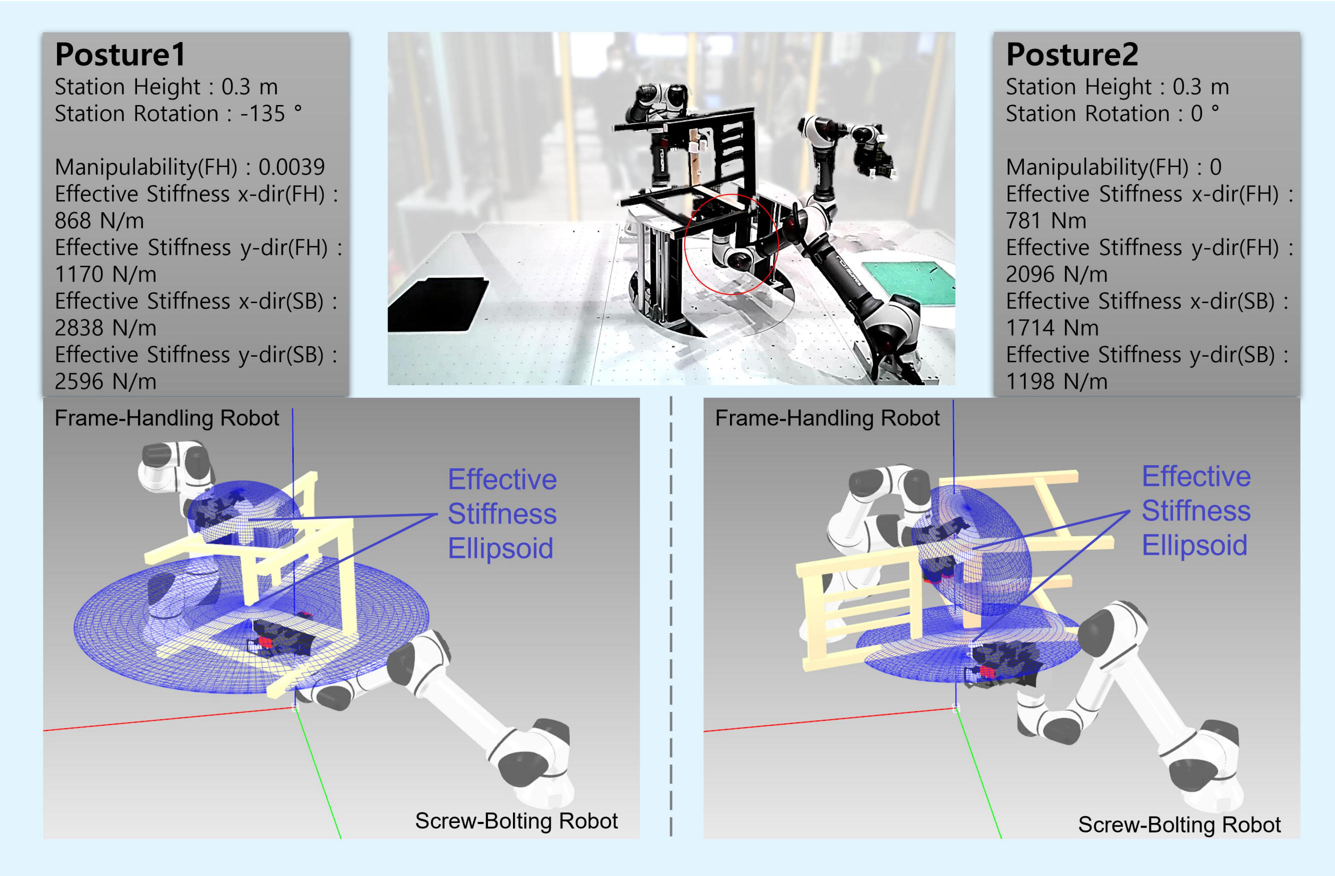

Fig. 11: Simulation of robot posture based on the angle of rotation of a manipulation station. We compared effective stiffness and manipulability in an assembly posture and proceeded with the assembly work in a good posture.

found in the blind search process, the wooden pin was partially inserted into the hole, and the height of the wooden pin was lowered. Therefore, the position of the finger that was pressing the wooden pin was lowered as well. That is, by observing changes in the finger position, it is possible to determine whether the hole position has been found. Changes in finger height at the moment in which the hole position is identified is displayed in the graph in Fig. 10(e). In this graph, the finger height decreases when the hole position is identified. When the hole position is identified, the search operation is stopped, and the assembly is completed by pressing in the direction of insertion of the wooden pin. At this time, the pressing force is higher than that when performing a blind search because an appropriate pressing force is required for the blind search, and a higher force is required to complete the assembly. Therefore, the height of the wooden pin is lower than that at the moment when the hole position is found.

E. Selection of Robot Posture in Assembly

Robot posture significantly influences the quality of assembly. A robot's ability to perform a task depends on its posture [20]. For example, when a robot is in a stretched posture, it can be difficult to apply additional force by using its ability to bear its own weight. Therefore, we configured the system such that the assembly position and the robot posture can be changed using a manipulation station. Therefore, we changed the assembly position and selected a good posture for assembly by considering the robot's dynamics.

In Fig. 11, the posture used in the competition is compared with other postures in the screw-bolting task. Posture 1 was the posture used in competition, and the manipulation station was rotated by $-135°$. Posture 2 was another posture that could be used to perform assembly without interference, and the manipulation station was rotated by $0°$.

We considered effective stiffness and manipulability as indices to evaluate posture, and the equation is as follows:

$$K_{xyz} = (JK_\theta^{-1}J^T)^{-1}, \tag{4}$$

$$\omega = \sqrt{det(JJ^T)}, \tag{5}$$

where, K_{xyz} and K_θ are effective stiffness and joint stiffness, respectively. J is a Jacobian matrix, and ω denotes manipulability.

We verified the effective stiffness of the robot to ensure that the robot did not shake during the bolting task, and the results indicated that the stiffness of the screw-bolting robot was higher in posture 1. The frame-handling robot was used to fix the frame in the bolting task. In posture 2, the y-directional stiffness of the frame-handling robot was high, but its manipulability was 0, resulting in a bad posture. In this manner, we selected the assembly posture by using an index that evaluates the robot posture within the possible assembly position, such that the robot can perform the assembly work stably with adequate performance. However, in actual assembly, we manually selected the robot's posture to additionally consider the possibility of collision between the robot and the assembly or whether unnecessary movements would be performed.

V. Conclusion

In this paper, we introduced the "Furniture Assembly AI-Robot Challenge 2021" and the technology of our research team, which won this challenge. The "Furniture Assembly AI-Robot Challenge 2021" was a competition in which robot assembly skills were evaluated based on recognition of assembly instructions. Our research team developed an AI-based algorithm that was able to recognize assembly instructions by using Faster-RCNN and create a connection relationship tree for interpreting the assembly instructions. We generated a PDDL command from the connection relationship tree and transmitted it to the robot system. The robot system consisted of three robots, multi-fingered grippers, and a manipulation station. The IKEA Stefan chair was assembled quickly and safely by using an assembly algorithm in conjunction with a multi-fingered gripper and manipulation station. By using the developed AI-based algorithm and robot system, our research team won the "Furniture Assembly AI-Robot Challenge 2021" competition with excellent performance, where the first mission was completed in 38 min and the second mission in 52 min.

Acknowledgment

The authors would like to thank Seonwoo Kim, Yong-Woo Shin, Trung Minh Bui, for their valuable time and effort that they have contributed throughout the competition. This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (NRF-2021R1A2C2008379), in part by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No.2019-0-01340, Development of AI Technology to Understand Assembly Manuals for People and Generate Assembly Work Plan), and in part by the Chung-Ang University Graduate Research Scholarship in 2020. Dongjun Shin is the corresponding author.

References

[1] J. Wallén, *The history of the industrial robot*. Linköping University Electronic Press, 2008.

[2] A. Gasparetto and L. Scalera, "A brief history of industrial robotics in the 20th century," *Adv. Hist. Stud.*, vol. 8, no. 1, pp. 24–35, 2019.

[3] A. Grau, M. Indri, L. L. Bello, and T. Sauter, "Industrial robotics in factory automation: From the early stage to the internet of things," in *Proc.: IECON 2017 - 43th Annu. Conf. IEEE Ind. Electron. Soc.* IEEE, 2017, pp. 6159–6164.

[4] F. Suárez-Ruiz and Q.-C. Pham, "A framework for fine robotic assembly," in *2016 IEEE Int. Conf. Robot. Autom. (ICRA)*. IEEE, 2016, pp. 421–426.

[5] F. Suárez-Ruiz, X. Zhou, and Q.-C. Pham, "Can robots assemble an ikea chair?" *Sci. Robot.*, vol. 3, no. 17, p. eaat6385, 2018.

[6] Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo, and R. Girshick, "Detectron2," https://github.com/facebookresearch/detectron2, 2019.

[7] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *Adv. Neural Inf. Process. Syst.*, vol. 28, pp. 91–99, 2015.

[8] B. O. Community, *Blender - a 3D modelling and rendering package*, Blender Foundation, Stichting Blender Foundation, Amsterdam, 2018. [Online]. Available: http://www.blender.org

[9] C. Luo, L. Jin, and Z. Sun, "Moran: A multi-object rectified attention network for scene text recognition," *Pattern Recognit.*, vol. 90, pp. 109–118, 2019.

[10] Y. Xiang, T. Schmidt, V. Narayanan, and D. Fox, "Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes," *arXiv preprint arXiv:1711.00199*, 2017.

[11] S. Peng, Y. Liu, Q. Huang, X. Zhou, and H. Bao, "Pvnet: Pixel-wise voting network for 6dof pose estimation," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 4561–4570.

[12] Z. Kootbally, C. Schlenoff, C. Lawler, T. Kramer, and S. K. Gupta, "Towards robust assembly with knowledge representation for the planning domain definition language (pddl)," *Robot. Comput.-Integr. Manuf.*, vol. 33, pp. 42–55, 2015.

[13] M.-S. Choi, D.-H. Lee, H. Park, Y.-J. Kim, G.-R. Jang, Y.-D. Shin, J.-H. Park, M.-H. Baeg, and J.-H. Bae, "Development of multi-purpose universal gripper," in *2017 56th Annu. Conf. Soc. Instrum. Control Eng. Jpn. (SICE)*. IEEE, 2017, pp. 1421–1424.

[14] S. Han, M.-S. Choi, Y.-W. Shin, G.-R. Jang, D.-H. Lee, J. Cho, J.-H. Park, and J.-H. Bae, "Screwdriving gripper that mimics human two-handed assembly tasks," *Robot.*, vol. 11, no. 1, p. 18, 2022.

[15] K. Kim, D. Y. Kim, S.-W. Jun, and J.-H. Hwang, "Estimation of the gripping position and orientation of fasteners in camera images," in *2021 18th Int. Conf. Ubiquitous Robots (UR)*. IEEE, 2021, pp. 190–193.

[16] S. Garrido-Jurado, R. Muñoz-Salinas, F. J. Madrid-Cuevas, and M. J. Marín-Jiménez, "Automatic generation and detection of highly reliable fiducial markers under occlusion," *Pattern Recognit.*, vol. 47, no. 6, pp. 2280–2292, 2014.

[17] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2961–2969.

[18] H. Park, J. Park, D.-H. Lee, J.-H. Park, M.-H. Baeg, and J.-H. Bae, "Compliance-based robotic peg-in-hole assembly strategy without force feedback," *IEEE Trans. Ind. Electron.*, vol. 64, no. 8, pp. 6299–6309, 2017.

[19] M.-S. Choi, Y.-W. Shin, G.-R. Jang, D.-H. Lee, J.-H. Park, and J.-H. Bae, "Kinesthetic sensing for peg-in-hole assembly based on in-hand manipulation," *IEEE Robot. Autom. Lett.*, vol. 6, no. 4, pp. 8418–8425, 2021.

[20] S. Kim, S. Yun, and D. Shin, "Numerical quantification of controllability in the null space for redundant manipulators," *Appl. Sci.*, vol. 11, no. 13, p. 6190, 2021.

Seongseop Yun Department of Mechanical Engineering, Yonsei University, Seoul, 03722, Republic of Korea. Email: ss.yun@yonsei.ac.kr & wsxde1234@cau.ac.kr.

Myoung-Su Choi Robotics R&D Department, Korea Institute of Industrial Technology, Ansan, 15588, Republic of Korea. Email: choims@kitech.re.kr.

Min-Young Cho Intelligent Robotics Research Center, Korea Electronics Technology, Bucheon, 14502, Republic of Korea. Email: mycho@keti.re.kr.

Keunhwan Kim Intelligent Robotics Research Center, Korea Electronics Technology, Bucheon, 14502, Republic of Korea. Email: khgapa@keti.re.kr.

Dong-Hyuk Lee Robotics R&D Department, Korea Institute of Industrial Technology, Ansan, 15588, Republic of Korea. Email: donghyuk@kitech.re.kr.

Sewoong Jun Intelligent Robotics Research Center, Korea Electronics Technology, Bucheon, 14502, Republic of Korea. Email: daniel@keti.re.kr.

Ji-Hun Bae Robotics R&D Department, Korea Institute of Industrial Technology, Ansan, 15588, Republic of Korea. Email: joseph@kitech.re.kr.

Dongjun Shin Department of Mechanical Engineering, Yonsei University, Seoul, 03722, Republic of Korea. Email: dj.shin@yonsei.ac.kr.