Proc. HRI-2013 Workshop on Collaborative Manipulation:
New Challenges for Robotics and HRI, Mar. 3, 2013.

Human-Robot Collaborative Assembly by On-line Human Action Recognition Based on an FSM Task Model

Hiraki Goto, Jun Miura, and Junichi Sugiyama
Department of Computer Science and Engineering
Toyohashi University of Technology

***Abstract*— This paper describes a human-robot collaboration in assembly tasks. When a robot works as an assistant to the user, timely supports from the robot is a key to realizing fluent collaboration. The robot, therefore, has to be able to recognize the current state of the assembly task and to choose assistive actions accordingly. We develop a finite state machine-based task model and a repertoire of visual routines for the task state and human action recognition for a collaborative assembly of a small table. The robot also generate verbal messages for keeping the user informed of the status of the robot. We successfully conducted collaboration experiments using our humanoid robot assistant.**

***Index Terms*— Human-robot collaboration, Task model-based collaboration, Assembly tasks, Vision-based state estimation.**

I. Introduction

Personal service robots are expected to help people in various scenes of their everyday life both in office and at home. People often collaborate to achieve several tasks such as jointly carrying a heavy item and constructing a large structure together. Future robots therefore have to have an ability to collaborate.

Suppose a robot helps a person who is doing a task which is not easy to complete by himself/herself. It is important for the robot to give its hands to him/her in a timely fashion. Such a timely assistance entails the robot's ability to understand the intention of the person. There are several ways of communicating intentions such as speech, gesture, and action [1] and such intentions need to be interpreted in the context of the collaborative task which they are conducting.

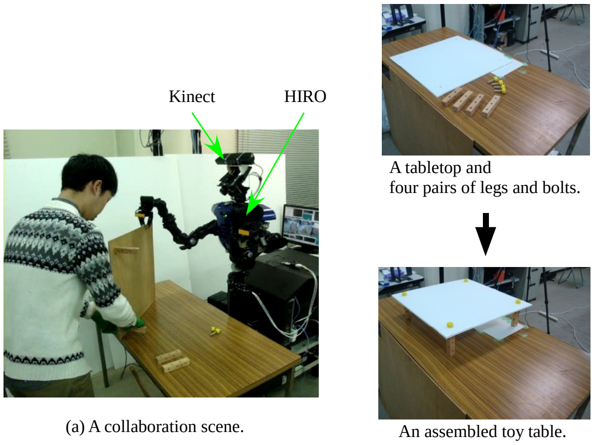
Kimura et al. [2] developed a system which analyzes a human demonstration to generate a set of assembly steps, each of which is described by preconditions, operations, and expected results. This knowledge is then used for invoking assistive robot operations by referring to the visual recognition result of the current state. Lens et al. [3] uses a similar rule-based expression of tasks.

Hanai et al. [4] developed a humanoid robot that can collaborate with human in a pick and place task. The task of a human and a robot is represented by a simple four-state state transition model. The robot uses this model for predicting human actions and generating an efficient, collision-free robot hand motion with an appropriate timing.

Dominey et al. [5] developed a framework of interactive task model construction and usage for anticipating human actions in a collaborative assembly task. A speech-controlled humanoid incrementally learns the sequences of verbal orders from the user and user's actions, which are used later for collaborative robot actions. A simple list is used for representing the task. They then extended the approach to task learning not by an explicit speech-based orders but by observing a collaborative task by human operators [6].

Hoffman and Breazeal [7] proposed a general framework of cooperation among a human-robot team, which emphasizes social aspects of collaboration such as negotiation and turn taking. They used a goal-oriented, hierarchical representation of tasks. Foster and Matheson [8] used an AND/OR-tree for representing tasks in a task model-based robotic verbal instruction for assembly. Nikolaidis and Shah [9] proposed to use Shared Mental Models (SMMs), which have been widely studied in human teamwork and coordination analysis, for information sharing among robots and humans in collaborative works. SMMs have a potential to be used for modeling complex collaborative tasks. Clark [10] considers the totality of communication including both gestural and verbal activities, which could be more important in fluent and effective human-robot interaction.

In this paper, we pursue a human-robot collaborative assembly based on human action recognition; a robot observes an action of the user to determine the status of the assembly task and generates an appropriate assistive action. We also consider the use of verbal messages for transferring the



(a) A collaboration scene.

A tabletop and
four pairs of legs and bolts.

An assembled toy table.

(b) Toy table assembly task.

Fig. 1. An assistive humanoid and the task.

robot's recognition results and intentions to the user for keeping him/her informed of the status of the robot.

Task representation is one of the important issues here. In order to cope with a variety of possible action sequences to achieve a goal, we adopt a finite state machine (FSM) for describing tasks. Each recognized user action is a trigger to cause a transition from one state to another, together with invoking an appropriate robotic observation or manipulation action. To construct the task model, we analyze human-human collaborations in an assembly task.

The task we deal with in this paper is the assembly of a small table. We chose this task because it is not easy to achieve it by a user alone and inherently requires a human-robot collaboration. Fig. 1 shows our assistive humanoid, HIRO (by Kawada Industry Co.) and the task. HIRO has two 6-dof arms with parallel grippers, a 1-dof waist, and a 2-dof neck mechanism. HIRO is also equipped with a Kinect sensor for recognizing parts states and human actions. Several visual recognition routines are developed for realizing this collaborative assembly.

The rest of the paper is organized as follows. Section II explains our task model representation and a concrete example for a small table assembly. Section III describes visual routines for recognizing the state as well as human actions. Section IV describes voice generation. Section V describes the experimental results. Section VI discusses the limitation of the current system and future work. Section VII summarizes the paper.

II. Task model

A task model describes how a task is achieved by a sequence of user and/or robot actions. A robot refers to it to generate timely assistive actions to the user. Since the assembly proceeds with changes of status of assembled parts, it is natural to describe a task by a set of *states* which is recognizable by the robot. We adopt a finite state machine (FSM) for describing tasks.

This paper does not deal with a learning aspect of task models. We instead develop a task model manually based on the observation of human-human collaborative assembly.

A. Observing human collaborations

We observed several sequences of assembly of a normal table by two persons. Fig. 2 shows one of the sequences. We asked two students to assemble a table and gave no specific instructions on how to collaborate, but they smoothly collaborated, with sometimes switching their roles, to achieve the task. Typical collaboration patterns observed are as follows.

- *Keep satisfying a necessary condition for continuing an action*. In Fig. 2(c), one (blue student) supports the tabletop while the other (white student) is attaching a leg to it. For the white student to continue to attach the leg, the tabletop should stand still in a certain pose; this condition is kept satisfied by the blue student.
- *Satisfy a precondition for starting an action*. In Fig. 2(b), one (blue student) pushes a leg to the other (white student) so that the white student can start attaching it to the tabletop.
- *Joint manipulation*. In Fig. 2(e), two students manipulate the table top in a coordinated way to rotate it.

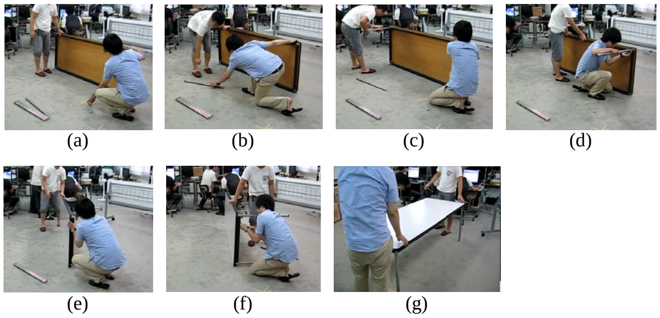

(a) (b) (c) (d) (e) (f) (g)

Fig. 2. An assembly sequence of a normal table by two persons (1st scenario). (a) They talk about their respective roles. (b) One pushes a leg to the other. (c) One is attaching a leg while the other holds the tabletop. (d) Switching the roles. (e) Turn the tabletop together. (f) Attach the other legs. (g) End of the task.

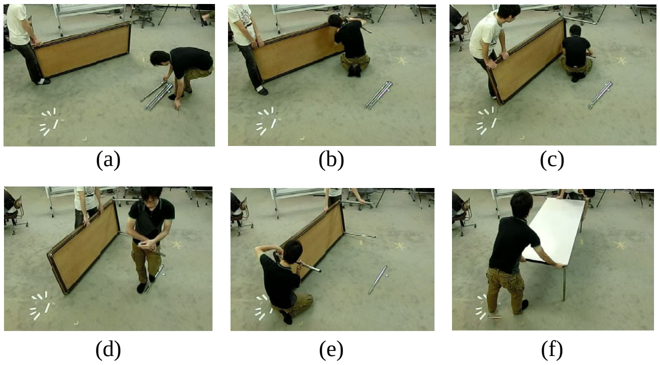

(a) (b) (c) (d) (e) (f)

Fig. 3. Another assembly sequence of a normal table by two persons (2nd scenario). (a) They talk about their respective roles. (b) One is attaching a leg while the other holds the tabletop. (c) Continue the same for the next leg. (d) Switching the positions while keeping the roles. (e) Attach the other legs. (e) End of the task.

Fig. 3 shows the sequence by another pair of students. In this sequence, the table is assembled in the same order as the previous one, but two students switch their positions while keeping their roles unchanged (see Fig. 3(d)).

Note that both sequences have a phase where two students negotiate about respective roles before starting the assembly. Although such a phase is really important for an effective and fluent collaboration, we do not deal with it in this paper.

Considering the robot's ability of object handling, we deal with the first two of the above collaboration patterns in task modeling.

B. Finite state machine representation

An FSM is defined by a set of states and state transitions. To model multiple possible consequences from one state, we adopt a Mealy machine, which is one type of finite state machines whose output values are determined both by its current state and input.

The task proceeds with state changes caused by user and/or robot actions. We consider two types of states: *static* and

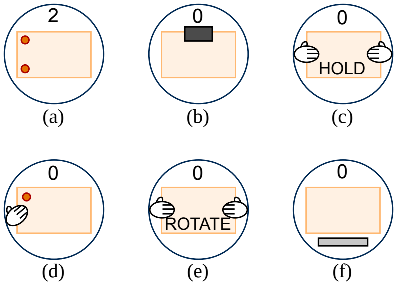


Fig. 4. Symbolic explanation of state descriptors. (a) The number of attached legs and positions. (b) Whether the robot holds the tabletop. (c) Whether the user holds the tabletop. (d) Whether the user is attaching a leg. (e) Whether the user is manipulating the tabletop. (f) Whether the robot is estimating the poses of the legs on the work table.

dynamic. Static states are the ones representing the state of the assembly, that is, the states of assembled parts and their relationships.

Dynamic states are the ones representing ongoing user or robot actions. This type of states is actually not a state of the assembly in a narrow sense, but is useful for recognizing the completion of user actions. Since the states of the assembled parts are sometimes hard to observe while the user is doing some action due to occlusion, detecting the end of such an action provides a good timing for calling the routine for recognizing the result (i.e., change of the static state) of the action.

C. Properties for describing tasks

State descriptors shown in Fig. 4 are prepared to represent each state of the task. There are three descriptors, for describing static states: (a) how many legs are attached to the tabletop and where, (b) whether the robot holds the tabletop, and (c) whether the user holds the tabletop. There are also three descriptors for describing dynamic states: (d) whether the user is attaching a leg, (e) whether the user is rotating the tabletop, and (f) whether the robot is estimating the poses of legs on the work table.

D. Task model for small table assembly

Fig. 5 shows a scenario of human-robot collaborative assembly of the small table, which is achievable by the current system and can be described using the above descriptors.

The assembly proceeds as follows: (a) the user put a tabletop on the work table and then the robot holds it; (b) the user attaches a leg while the robot is holding the tabletop; (c) after the user attaches two legs at one side of the tabletop, the robot releases the hand for the subsequent user action; (d) the user rotates the tabletop so that he can attach the remaining two legs, and the robot holds it again; (e) when a leg on the work table are far from the user, the robot pushes it to him; (f) the assembly finishes when all legs are properly attached.

Based on this scenario, we developed an FSM representing the small table assembly task. Fig. 6 shows the first part of the whole assembly graph. When the system in invoked, it starts recognizing the user's rotating action (which is actually an action of putting the tabletop on the work table). Once the action finishes (i.e., hand motion stops), the system starts to estimate the pose of the tabletop and calculate the grasp point. If the grasp point is obtained, the robot moves to grasp it. If not due to, for example, a bad position of the table top, the system notifies it to the user, and goes back to the recognition of the rotating action. If the grasp is successful, the system continues to do further actions. Otherwise, it goes back to the first state.

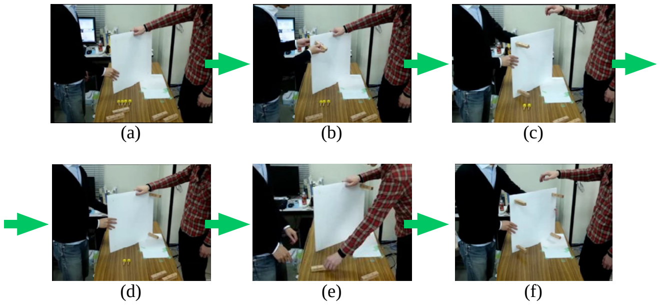


Fig. 5. Collaboration scenario. The red student takes the role of the robot. (a) The user put a tabletop and the robot holds it. (b) The user attaches a leg. (c) The robot releases the hand. (d) The user rotates the tabletop and the robot holds it again. (e) The robot passes a leg to the user. (f) The end of assembly.

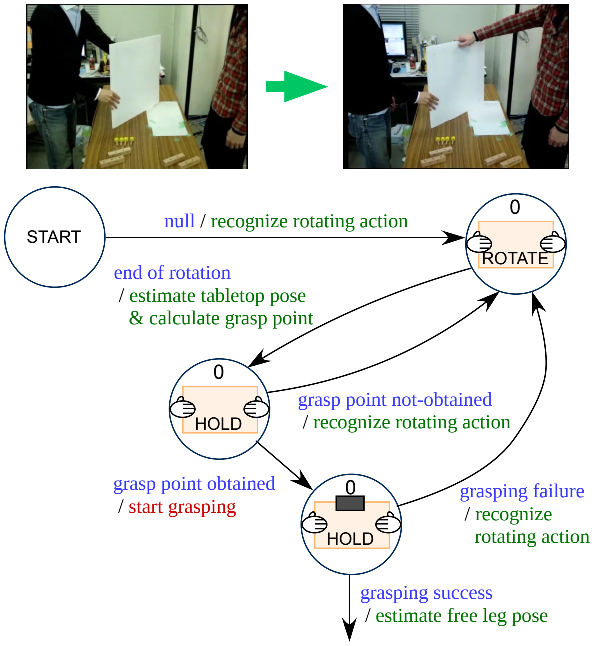


Fig. 6. A part of the small table assembly task model.

The whole task model has 36 states and 53 edges (see Fig. 7). Each state transition is specified by a pair of an input and an output. Each output leads to the invocation of a recognition or manipulation action of the robot, while each input corresponds to the completion of such an action.

Many of transitions (i.e., edges) are associated with the system's recognition of user actions. The details of the recognition procedures will be described in the next section.

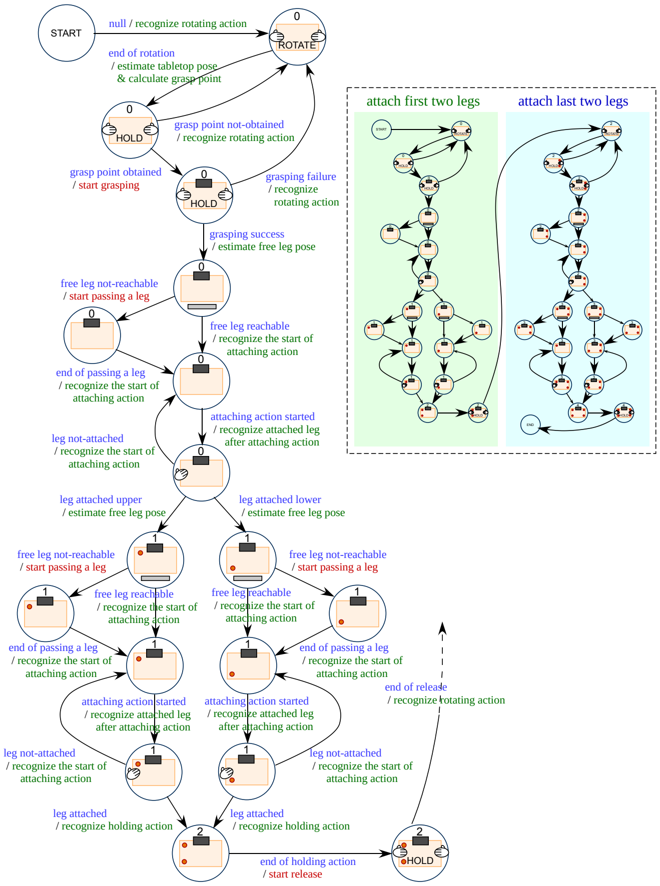
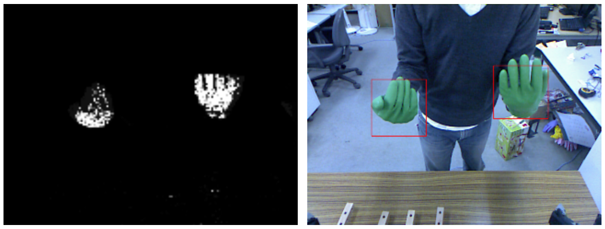


Fig. 7. The task model used. The complete model is shown inside the upper-right dashed box. The first half of the model (green part, attaching first two legs) is described in detail. Blue, green, and red descriptions indicate triggers (inputs) of state transition, invoked recognition actions, and invoked manipulation actions, respectively.



(a) High probability regions. (b) Tracked hand regions.

Fig. 8. Color-based hand tracking.

III. State Recognition Functions

A. Recognizing hands

Tracking hands is important for understanding user actions in collaborative assembly tasks. It provides information on whether a user action is going on and where it is.

We use a colored glove for simplifying the hand detection and tracking. An HSV color histogram is learned as a model of the hand color in advance and is used for tracking using the OpenCV [11] implementation of CAMSHIFT algorithm [12]. Tracked regions larger than some threshold are judged as hand regions. Fig. 8 shows a hand tracking result.

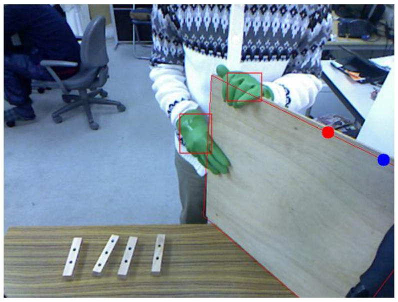


Fig. 9. Estimate the tabletop pose and calculate grasp points.

B. Recognizing a plane

Plane detection is used for recognizing the work table and for estimating the pose of the tabletop. The largest plane in the camera view is first detected from a point cloud using a PCL plane detection function [13].

In estimating the pose of the tabletop, The 3D points inside the detected plane is mapped onto the 2D image to find boundaries using a Hough transform. These detected boundaries are then backprojected to 3D to determine the 3D boundary lines, which are then used for pose estimation, combined with the knowledge of the shape and the size of the tabletop.

Candidates for grasp points are also given in advance. Feasible grasp points are calculated considering the kinematics of the robot and the hand position. If no feasible grasp point is obtained, this recognition module reports a failure so that the user will put the tabletop at another position (see Sec. II-D). Fig. 9 shows an example of tabletop pose estimation and grasp point calculation. The estimated tabletop pose is superimposed in red. Two feasible grasp points are calculated and the nearer one to the robot is selected (blue one).

C. Recognizing table legs

A general block recognition module is used for recognizing the legs of the table. This module applies an ICP (Iterative Closest Point) [14]-based pose estimation algorithm to a 3D point cluster extracted from the scene. This extraction utilizes the knowledge of the supporting plane, which is either the work table or the tabletop, depending on the current state.

The robot counts the number of legs on the work table, as a part of recognition of the user's attaching action; if the number is smaller by one from the previous state, the robot concludes that the user picked up one for the current attaching action. The work table is the supporting plane in this case (see Fig. 10, (a) and (b)).

To verify a leg is certainly attached to the tabletop after the user's attaching action, the block recognition module focuses on the area where the hand was moving in order to detect a leg there. In this case, the tabletop becomes the supporting plane (see Fig. 10, (c) and (d)).

IV. Generating Verbal Messages

Partner's status and intentions are important information sources for collaboration. Verbal communications are often

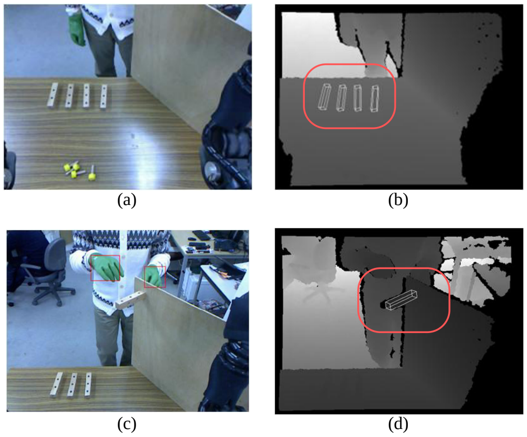

(a) (b)
(c) (d)

Fig. 10. Estimation of leg pose: (a)(b) legs on the work table. (c)(d) a leg attached to the tabletop.

used for transferring such information in human-human collaborations. We observed several instances of verbal communication in the analysis shown above (see Figs. 2 and 3).

It is, therefore, useful for the user to get verbal messages from the robot describing robot's status and intentions such as how it recognizes the scene (state description) and what it is going to do (action explanation).

State description messages are for describing the current state thereby telling the user that the assembly process is going on properly. Example messages are: "Two legs are attached," "I am holding the tabletop," and "You are attaching a leg."

Action explanation messages are for notifying the user what the robot is doing or is going to do, both in robot motion and scene recognition. Messages for robot motions are precautions to the user just before the robot moves; examples are "I'm going to hold the tabletop" and "I will pass a leg to you." Messages for scene recognition are for notifying the user what the robot is trying to recognize and how the recognition results are. Since recognition is a *silent* action and often takes some time, delivering such messages are useful for the user. It is also effective to ask the user to do some recovery action when the recognition fails. Example messages are: "I found legs on your side," and "I recognize the end of your attaching action. I will verify the attachment."

We use OpenHRI [15] as a speech synthesis engine.

V. Experimental results

We implemented the software modules explained above and other tools such as FSM management and robot control. These modules and tools are implemented using *RT-middleware* [16], which supports a modularized software development.

We performed collaborative assembly experiments several times, with a slightly different order of legs to attach. The user and the robot succeeded in completing the table assembly task. Figs. 11 and 12 show two example sequences of successful assembly.

It took about five and six minutes for completing the task in the first and the second sequence, respectively. The differences between the sequences are the order of attaching four legs and whether the robot generated an action for passing a leg to the user. These differences occur due to different user's plans or different initial states, and result in different paths to take in the finite state machine.

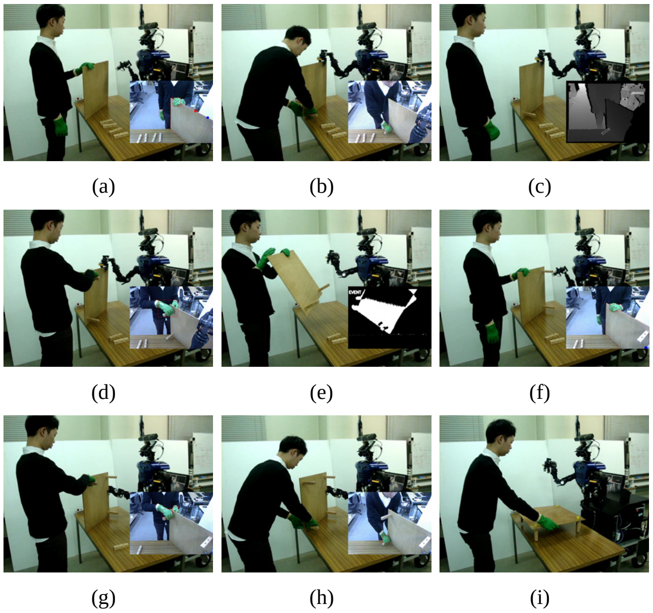

(a) (b) (c)
(d) (e) (f)
(g) (h) (i)

Fig. 11. Snapshots of sequence 1. (a) The user put the tabletop on a work table, and the robot estimates its pose and determines the grasp point. (b) The user is attaching a leg while the robot holds the top of the tabletop. (c) Pose estimation of the attached leg. (d) The user is attaching the second leg. (e) Th robot released the hand and the user is rotating the tabletop. (f) The robot selects the grasp point on the side of the tabletop. (g)(h) The user is attaching the third or the fourth leg. (i) The assembly is completed.

VI. Discussion

We have realized human-robot collaborative assembly of a small table using an FSM-based task model and scene recognition and action planning capabilities. The current system, however, has several problems/limitations which should be overcome when applied to more general and complex tasks.

The task model is currently hand-made. We analyzed the human-human collaborations, designed the work flow, and constructed the task model by considering the current ability of the robot. This way of task modeling is time consuming and is not scalable, and a constructed model can deal with only a limited portion of possible scenarios, even for a relatively simple task treated in this paper. The use must be careful to be within that portion. Programming by demonstration approaches (e.g., [17], [18]), which automatically generates task models by observing human demonstrations, can be adopted. When applied to robot-human collaborative tasks, however, more focus should be on what actions and communications are essential in establishing the collaboration.

The current scene recognition routines have also limitations. They usually take time to get the results, so the user has to wait for the end of a recognition action of the robot before taking the next action. The recognition ability itself should also be improved; recognition of various objects in various

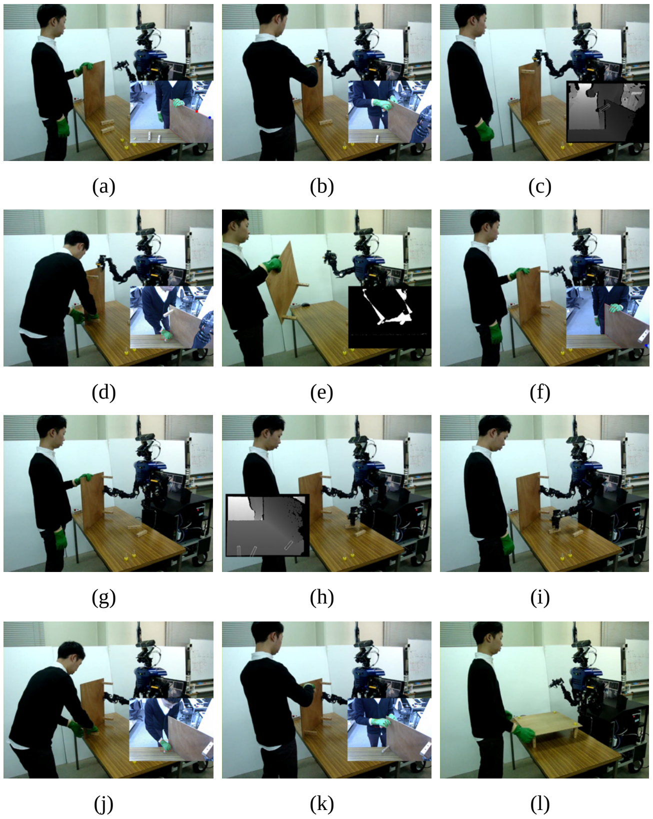

(a) (b) (c) (d) (e) (f) (g) (h) (i) (j) (k) (l)

Fig. 12. Snapshots of sequence 2. (a) The user put the tabletop on a work table, and the robot estimates its pose and determines the grasp point. (b) The user is attaching a leg while the robot holds the top of the tabletop. (c) Pose estimation of the attached leg. (d) The user is attaching the second leg. (e) Th robot released the hand and the user is rotating the tabletop. (f) The robot selects the grasp point on the side of the tabletop. (g) The robot holds the tabletop. (h) The robot estimates the pose of legs on the work table, and picks up one of them. (i) The robot places the leg at a near position to the user. (j)(k) The user is attaching the third and the fourth leg. (l) The assembly is completed.

conditions (e.g., viewpoints and illumination conditions) need to be realized to cope with a wider variety of tasks.

Integration of verbal and action-based communication should also be investigated. The timing and the contents of the current verbal messages are determined manually, without considering the status of the user. More dynamic, on-line generation of effective verbal messages are desirable. It is also important for the robot to actively inquire about knowledge on the task [19] and/or the user's status, for incrementally acquiring/refining task models, through verbal communication.

VII. Summary

Human-robot collaborative assembly is one of the interesting research fields in HRI. In this paper, we have described our assistive humanoid robot that can support the user in a collaborative way. The keys to realizing such a collaboration are an FSM-based task model and an elaborated set of visual recognition routines for state transition detection. Verbal messages are additionally used for notifying the user the status of the robot and the assembly task. We successfully conducted preliminary experiments on collaborative assembly of a small table. We have also discussed the current limitation of the system and future research directions.

Acknowledgment

This work is supported in part by Grant-in-Aid for Scientific Research (No. 23650096) from JSPS.

References

[1] A. Bauer, D. Wolherr, and M. Buss. Human-Robot Collaboration: A Survey. *Int. J. of Humanoid Robotics*, Vol. 5, No. 1, pp. 47–66, 2008.

[2] H. Kimura, T. Horiuchi, and K. Ikeuchi. Task-Model Based Human Robot Cooperation Using Vision. In *Proceedings 1999 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp. 701–706, 1999.

[3] C. Lenz, S. Nair, M. Rickert, A. Knoll, W. Rosel, J. Gast, A. Bannat, and F. Wallhoff. Joint-action for humans and industrial robots for assembly tasks. In *Proceedings of 17th IEEE Int. Symp. on Robot and Human Interactive Communication*, pp. 130–135, 2008.

[4] R. Hanai, R. Oya, T. Izawa, and M. Inaba. Motion Generation for Human-Robot Collaborative Pick and Place Based on Non-Obstruction Strategy. In *Proceedings of 2011 IEEE Int. Conf. on Robotics and Biomimetics*, pp. 20–25, 2011.

[5] P.F. Dominey, G. Metta, F. Nori, and L. Natale. Anticipation and Initiative in Human-Humanoid Collaboration. In *Proceedings of the 8th IEEE-RAS Int. Conf. on Humanoid Robots*, 2008.

[6] S. Lallee, F. Warneken, and P.F. Dominey. Learing to Collaborate by Observation. In *Proceedings of Humanoids 2009 Workshop on Developmental Psychology Contributions to Cooperative Human Robot Interaction*, 2009.

[7] G. Hoffman and C. Breazeal. Collaboration in Human-Robot Team. In *Proceedings of AIAA 1st Intelligent Systems Technical Conf.*, 2004.

[8] M.E. Foster and C. Matheson. Following Assembly Plans in Cooperative, Task-Based Human-Robot Dialogue. In *Proceedings of the 12th Workshop on the Semantics and Pragmatics of Dialogue*, 2008.

[9] S. Nikolaidis and J. Shah. Human-Robot Teaming using Shared Mental Models. In *Proceedings of HRI2012 Workshop on Human-Agent-Robot Teamwork*, 2012.

[10] H.H. Clark. *Using Language*. Cambridge University Press, 1996.

[11] OpenCV. `http://opencv.org/`.

[12] G.R. Bradski. Computer Vision Face Tracking for Use in a Perceptual User Interface. *Intel Technology Journal Q2*, 1998.

[13] Point Cloud Library. `http://pointclouds.org/`.

[14] K.S. Arun, T.S. Huang, and S.D. Blostein. Least-Squares Fitting of Two 3-D Point Sets. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 9, No. 5, pp. 698–700, 1987.

[15] OpenHRI. `http://openhri.net/`.

[16] N. Ando, T. Suehiro, K. Kitagaki, T. Kotoku, and W.-K. Yoon. RT-Middleware: Distributed Component Middleware for RT (Robot Technology). In *Proceedings of 2005 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp. 3555–3560, 2005.

[17] M. Ehrenmann, O. Rogalla, R. Zöllner, and R. Dillmann. Teaching Service Robots Complex Tasks: Programming by Demonstration for Workshop and Household Environments. In *Proceedings of 2001 Int. Conf on Field and Service Robots*, pp. 397–402, 2001.

[18] K. Ogawara, J. Takamatsu, H. Kimura, and K. Ikeuchi. Extraction of Essential Interactions Through Multiple Observations of Human Demonstrations. *IEEE Trans. on Industrial Electronics*, Vol. 50, No. 4, pp. 667–675, 2003.

[19] J. Miura and Y. Nishimura. Co-Development of Task Models through Robot-Human Interaction. In *Proceedings of the 2007 IEEE Int. Conf. on Robotics and Biomimetics*, pp. 640–645, 2007.