frontiers | Frontiers in Computer Science

TYPE Original Research
PUBLISHED 31 July 2024
DOI 10.3389/fcomp.2024.1235239

OPEN ACCESS

EDITED BY
Alessandro Bruno,
Universitá IULM, Italy

REVIEWED BY
Isak Karabegović,
University of Bihać, Bosnia and Herzegovina
Christian Tamantini,
Universitá Campus Bio-Medico di Roma, Italy

*CORRESPONDENCE
Katsushi Ikeuchi
katsushi.ikeuchi@outlook.jp

RECEIVED 06 June 2023
ACCEPTED 15 July 2024
PUBLISHED 31 July 2024

CITATION
Ikeuchi K, Takamatsu J, Sasabuchi K, Wake N and Kanehira A (2024) Applying learning-from-observation to household service robots: three task common-sense formulations. *Front. Comput. Sci.* 6:1235239. doi: 10.3389/fcomp.2024.1235239

Applying learning-from-observation to household service robots: three task common-sense formulations

Katsushi Ikeuchi*, Jun Takamatsu, Kazuhiro Sasabuchi, Naoki Wake and Atsushi Kanehira

Applied Robotics Research, Microsoft, Redmond, WA, United States

Utilizing a robot in a new application requires the robot to be programmed at each time. To reduce such programmings efforts, we have been developing "Learning-from-observation (LfO)" that automatically generates robot programs by observing human demonstrations. So far, our previous research has been in the industrial domain. From now on, we want to expand the application field to the household-service domain. One of the main issues with introducing this LfO system into the domain is the cluttered environments, which makes it difficult to discern which movements of the human body parts and their relationships with environment objects are crucial for task execution when observing demonstrations. To overcome this issue, it is necessary for the system to have task common-sense shared with the human demonstrator to focus on the demonstrator's specific movements. Here, task common-sense is defined as the movements humans take almost unconsciously to streamline or optimize the execution of a series of tasks. In this paper, we extract and define three types of task common-sense (semi-conscious movements) that should be focused on when observing demonstrations of household tasks and propose representations to describe them. Specifically, the paper proposes to use labanotation to describe the whole-body movements with respect to the environment, contact-webs to describe the hand-finger movements with respect to the tool for grasping, and physical and semantic constraints to describe the movements of the hand with the tool with respect to the environment. Based on these representations, the paper formulates task models, machine-independent robot programs, that indicate what-to-do and where-to-do. In this design process, the necessary and sufficient set of task models to be prepared in the task-model library are determined on the following criteria: for grasping tasks, according to the classification of contact-webs along the purpose of the grasping, and for manipulation tasks, corresponding to possible transitions between states defined by either physical constraints and semantic constraints. The skill-agent library is also prepared to collect skill-agents corresponding to tasks. The skill-agents in the library are pre-trained using reinforcement learning with the reward functions designed based on the physical and semantic constraints to execute the task when specific parameters are provided. Third, the paper explains the task encoder to obtain task models and task decoder to execute the task models on the robot hardware. The task encoder understands what-to-do from the verbal input and retrieves the corresponding task model in the library. Next, based on the knowledge of each task, the system focuses on specific parts of the demonstration to collect where-to-do parameters for executing the

task. The decoder constructs a sequence of skill-agents retrieving from the skill-agnet library corresponding and inserts those parameters obtained from the demonstration into these skill-agents, allowing the robot to perform task sequences with following the Labanotation postures. Finally, this paper presents how the system actually works through several example scenes.

KEYWORDS

learning-from-observation, task model, skill-agent library, grasp taxonomy, Labanotation, face contact equations, reinforcement learning

1 Introduction

One powerful means of acquiring human behavior is to observe and imitate the behavior of others. Humans go through a period of imitating their mother's behavior at one stage of development (Piaget, 2020). Even in adulthood, imitation from practice videos is often used in various sport practice such as golf practice and judo practice. Learning-from-observation, programming-by-demonstration and learning-by-watching aim to apply this behavior observation and learning paradigm to robots and to automatically generate robot programs from observation (Schaal, 1999; Schaal et al., 2003; Asfour et al., 2008; Billard et al., 2008; Dillmann et al., 2010; Akgun et al., 2012). The origin of this field lies in our research Ikeuchi and Reddy (1991) and Ikeuchi and Suehiro (1994) as well as Kuniyoshi et al. (1994). These two studies shared the common goal that they attempted to understand human behavior by viewing it under some framework and to make robots to perform the same behavior following the framework.

Later, in terms of the approach to obtain this framework for observation, the research field was split into two schools: the bottom-up and the top-down. The "bottom-up" school (Schaal, 1996; Samejima et al., 2006; Billard et al., 2008) has been attempting to acquire this framework from scratch through learning. The "top-down" school (Kang and Ikeuchi, 1997; Tsuda et al., 2000; Takamatsu et al., 2006) has been attempting to mathematically design a framework by utilizing the accumulated knowledge of robotics field to date. The "bottom-up" approach has been the mainstream due to the population of researchers and the rise of machine learning.

The authors, on the other hand, take the "top-down" position. In both human and robot learning, the physical structure, height, and weight of the student often differ from those of the teacher. Furthermore, environmental changes are also present. Therefore, trying to mimic the teacher's trajectory itself, as in the bottom-up approach, is difficult due to these differences in kinematics, dynamics, and environment. This difficulty is avoided by first extracting the essence of the behavior from the observation and then mapping this essence in the way adapted to the individual hardware and environment. For this purpose, we aim to design mathematically consistent abstracted behavioral models based on the knowledge accumulated in the robotics field, and to utilize these models to represent the essence of the demonstrations.

Going back to the history of robot programming, since its early days, some researchers have worked on automatic programming, to generate robot programs from assembly drawings or abstract concepts, referred to as task descriptions (Lozano-Perez, 1983; Lozano-Perez et al., 1984; De Mello and Sanderson, 1990). This trend, after about 30 years of research, encountered more difficult obstacles than imagined, as summarized in Raj Reddy's Turing Award talk "AI: To dream possible dreams" in 2000 (Reddy, 2007). In this talk, Reddy advocates 90% AI to overcome this barrier. In other words, for the relatively easy part of 90% or so, the solution is automatically obtained using the accumulated methods of automated programming to date. The remaining 10% or so of the problem, which cannot be solved by any means, should be solved with hints from humans.

Our "top-down" approach is based on this 90% AI approach. It attempts to overcome the difficulties of automated programming by designing a framework of understanding based on the theory accumulated in previous automated programming efforts in the field of robotics, and by gleaning hints from human demonstrations to actually make the system work.

This paper focuses on the "top-down" approach, provides some findings on this approach, and describes some ongoing projects to apply these findings to the development of household robots. In the next section, we review previous research along with the LfO paradigm. Section 3 describes the task common-sense that must be shared between human demonstrators and the observation systems in order to apply this paradigm to the household-service domain and discusses the formulation of this knowledge. Section 4 describes the grasping and the manipulation skill-agent libraries necessary to implement this formulated knowledge in robots. Sections 5 and 6 describe the system implementation and the system in action. Section 7 summarizes this paper.

The contribution of this paper are:

- to give an overview of the past LfO efforts,
- to enumerate the three task common-senses, semi-conscious human body-part movements, required to apply LfO to the household domains,
- to propose representations of the three task common-senses, and
- to show how these representations are used in LfO task-encoder and task-decoder.

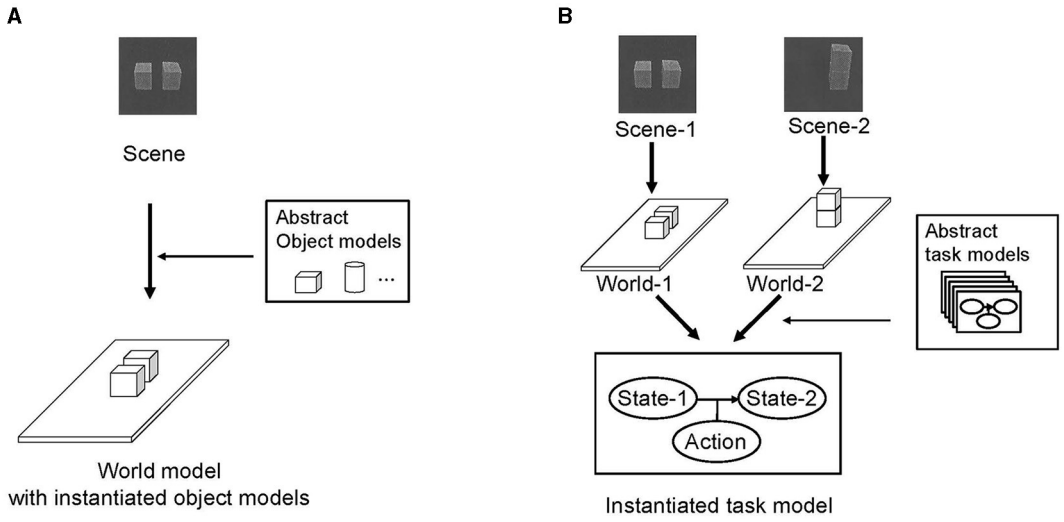

FIGURE 1
(A) Object recognition. **(B)** Task recognition.

2 LfO paradigm

The LfO paradigm can be considered as an extension of the Marr's paradigm of object recognition (Marr, 2010). First, let us review the Marr's paradigm, shown in Figure 1A. Marr defines the purpose of object recognition as creating a copy of the real world inside the brain or computer. For this purpose, an abstract object model is created inside the computer and then matching is performed between the internal model and the external image. The matching process involves two distinct sub-processes: indexing and localization. Indexing is the process of identifying which of the abstract object models corresponds to the observed one. Namely, indexing obtains the solution corresponding to what this is. Localization is the process of transforming the abstract object model into an instantiated object model with concrete dimensions and placing it in virtual copy. In other words, localization obtains a solution corresponding to where this is. In computer vision, these two operations are often performed sequentially. In the brain, these two sub-processes are carried out in separate and distinct circuits, with indexing proceeding in a new pathway through the visual cortex, and localization proceeding in an older pathway through the superior colliculus (Ramachandran and Blakeslee, 1998). In either case, a world model with instantiated object models, a copied world named by Marr, is eventually created in the computer or the brain (Marr, 2010).

Task recognition can be viewed as an extension of the Marr's object-recognition paradigm, as shown in Figure 1B (Ikeuchi and Suehiro, 1994). The purpose of task recognition is to be able to imagine in the brain or computer what task has been performed. An abstract task model is created that associates a state transition with an action required to cause such a transition. An abstract task model takes the form of Minsky (1988)'s frames, in which the slots for the skill parameters required to perform the action are also prepared. Object recognition is then performed on the input images to create two world models before and after an action. Task recognition instantiates a particular task model corresponding to the action based on the state transition extracted by comparing the pre- and post-action worlds. Namely, the what-to-do is obtained. Third, the where-to-do parameters, skill parameters for the action, such as the starting point, the end point, and the grasping point, are extracted from the images, and the instantiated task model is completed. The instantiated task model guides the robot to imitate human actions.

When designing task models, a divide-and-conquer strategy is used to address the different domains of human activity. It would be difficult, if not impossible, to design a set of abstract task models that completely covers all human activity domains. It would be inefficient, too. Therefore, we divide various human activities into specific domains and design a set of task models that satisfies the necessary and sufficient conditions for complete coverage of those domains. The domains we have dealt with so far include:

- Two-block domain (Ikeuchi and Reddy, 1991).
- Polyhedral object domain (translation only) (Ikeuchi and Suehiro, 1994).
- Polyhedral object domain (translation and rotation) (Takamatsu et al., 2007).
- Machine-parts domain including screw tightening, snap pushing, and rubber-ring hanging (Sato et al., 2002).
- Knot-tying domain including eight-knot and bowline-knot (Takamatsu et al., 2006).
- Folk dance domain (Nakaoka et al., 2007; Okamoto et al., 2014; Ikeuchi et al., 2018).

3 Task common-sense

When applying the LfO paradigm to the household-service domain in the human home environment, the main obstacle is

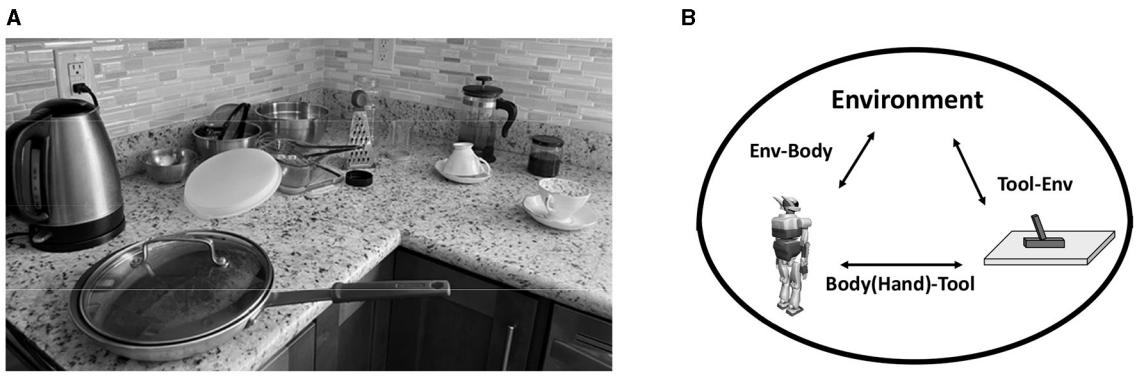

FIGURE 2
Service-robot domain. **(A)** Cluttered environment. **(B)** Three relations.

the overwhelming crowding in this domain (see Figure 2A). In the traditional industrial domain, the only objects present in the surroundings are those related to the task, to the exclusion of other miscellaneous objects. There are also not so many unrelated people in the surroundings. However, as seen in the Figure 2A, in the human home environment, there are unrelated objects on the table as well as various people moving around. Therefore, it is important to share the human task common-sense of what is the important for the demonstration between the LfO system and the demonstrator, and let the LfO system decide where to direct its attention to avoid the crowding. Here, task common-sense is defined as *the semi-unconscious movements of parts of the human body toward specific objects in the environment to streamline and optimize the execution of a serise of tasks.* We propose to focus on the following three relations as shown in Figure 2B to explicitly express task common-sense in household environmental tasks:

- **Environment and robot:** to represent human task common-sense regarding the postures a person should take with respect to the environment for smooth execution of a task sequence.
- **Robot and tool:** to represent human task common-sense regarding grasping strategies of objects (mainly tools) for robust execution of a task sequence.
- **Tool and environment:** to represent human task common-sense regarding the movement of the tools in relation to the environment for successful execution of a task sequence.

The following sections will discuss these relationships and describe task common-sense representations.

3.1 Environment-body relation

Our system is designed to mimic the approximated postures of the demonstrator. For example, when opening a refrigerator door, a robot with redundant degrees of freedom, as is common in recent humanoid robots, can perform this task in several different postures as shown in Figure 3. However, we prefer the robot to work in a human-like posture, as shown in Figure 3B, for the following reasons:

- When performing tasks, human-like postures are more predictable to bystanders and less likely to cause interpersonal accidents.
- Humans unconsciously adopt the optimal postures for the environment in order to perform next task in a task sequence. In the example above, the purpose of opening the refrigerator door is often to take out the items inside as the next task. The human-like posture shown in Figure 3B is considered more efficient for this purpose.

We need a representation method to describe approximate postures. The difference between human and robot mechanisms makes it difficult to achieve exactly the same postures by taking exactly the same joint angles and joint positions at each sampling time. Our approximate imitation does not require such precise representations. Rather, it is necessary to capture the essence of those postures that the bystanders perceive as nearly identical.

For this approximation, we will use Labanotation (Hutchinson-Guest, 1970), which is used by the dance community to record dance performances. The relationship between dance performances and Labanotation scores is similar to the relationship between music performances and music scores. Just as a piece of music can be performed from a music score, a piece of dance can be performed from a Labanotation score; just as a music score can be obtained from listening to a peice of music, a Labanotation score can be obtained from watching a piece of dance. More importantly, from the same Labanotation score, each dancer, with different heights and arm lengths, performs a piece of dance that appears the same to spectators. In other words, the Labanotation score is considered to capture the essence of the dance for the observers.

Labanotation is an approximation of human movement. Each dancer has different lengths in various parts of their body. However, it is known that when dancers follow Labanotation to dance the piece, their movements appear similar to spectators. When robots mimic human movement, due to the differing lengths of body parts between humans and robots, it is mechanically difficult, if

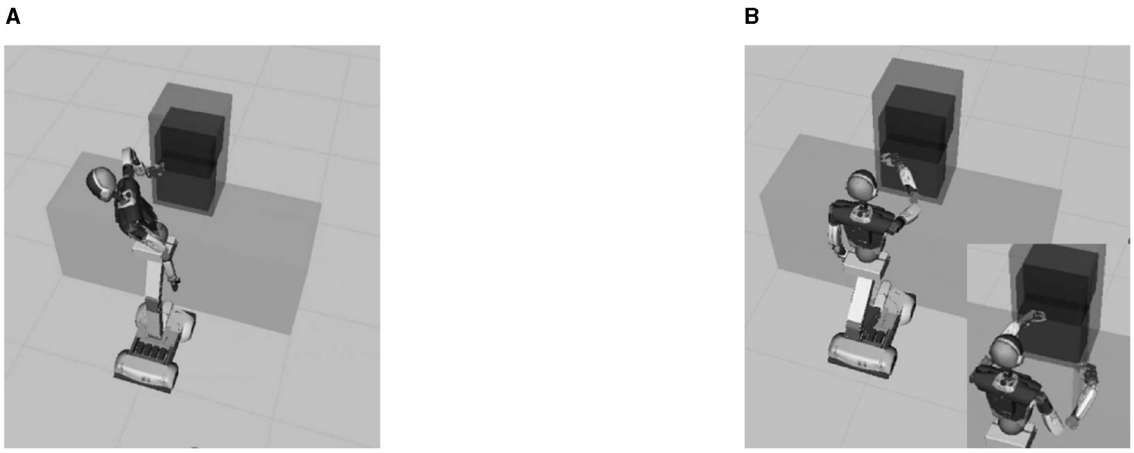

FIGURE 3
Two possible postures for opening the refrigerator door. **(A)** Inhuman-like posture. **(B)** Human-like posture.

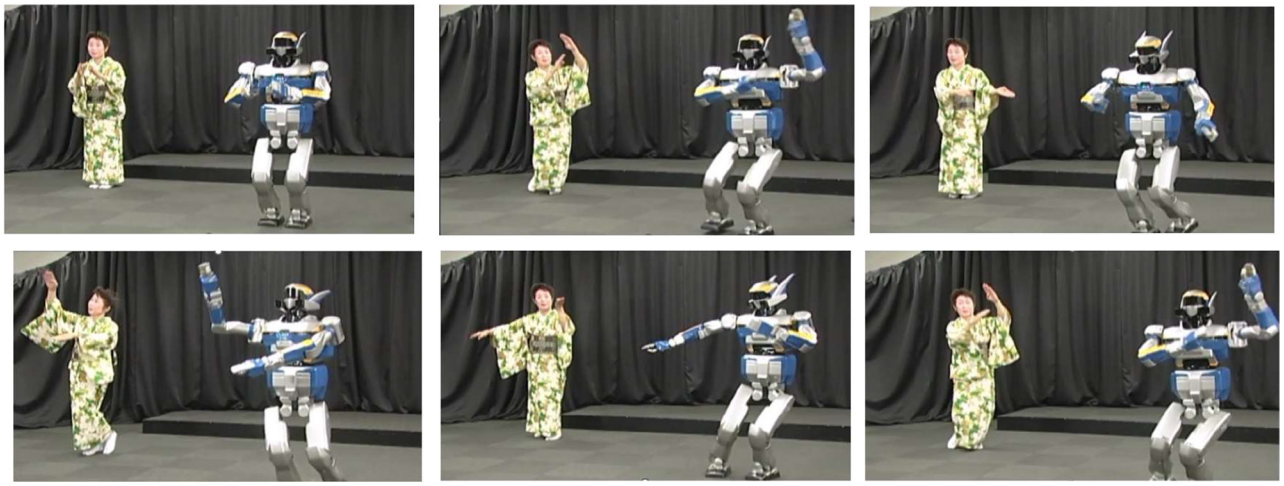

FIGURE 4
Dancing robot. See our YouTube video (https://youtu.be/PGKGXxwp6LM) for comparison. The pictures show the extracted frames corresponding to the key poses of the contentious dance motion in the YouTube video. Regarding the description of the lower body motions and the upper body motions, see Nakaoka et al. (2007) and Ikeuchi et al. (2018), respectively. The robot dance in the YouTube was originally demonstrated to the public in 2003 at the University of Tokyo's Institute of Industrial Science. The copyright of the video belongs to Ikeuchi laboratory, Institute of Industrial Science, The University of Tokyo (at that time).

not impossible, to take the same joint angles or joint positions at each sampling interval, and, even if it would be possible, there would be no guarantee that the movements would appear to be same to spectators. However, if a robot follows (or approximately follows) Labanotation at each key pose, it can achieve a posture flow that spectators perceive as almost the same. In fact, our dance robot creates movements based on Labanotation, even though each detailed movements are different from those of humans, they appear similar. See Figure 4 (Nakaoka et al., 2007) and YouTube video.[1] Therefore, we decided to use Labanotation as posture approximations for this service-robot domain.

In a music score, time flows from left to right, whereas in a Labanotation score, time flows from bottom to top. A simple example of Labanotation can be found around the middle of Figure 5. Labanotation follows this digitization in the time domain, i.e., each symbol in the score represents a posture at each brief stop. Each column of the Labanotation is used to represent the postures of one human part, such as an arm or an elbow. The length of each symbol represents the time it takes to move from the previous posture to that posture. The shorter the symbol, the faster the person moves the part; the longer the symbol, the slower the person moves the part. The shape of the symbol represents the digitized azimuth angles of the part in eight directions, such as east, west, north, and south, and the texture of the symbol represents the digitized zenith angles of the part in five directions, such as zenith, higher, level, lower, and nadir. Although the digitization of the eight azimuthal angles and the five zenithal angles seems somewhat coarse, it is consistent with Miller's theory of human

1 https://youtu.be/PGKGXxwp6LM

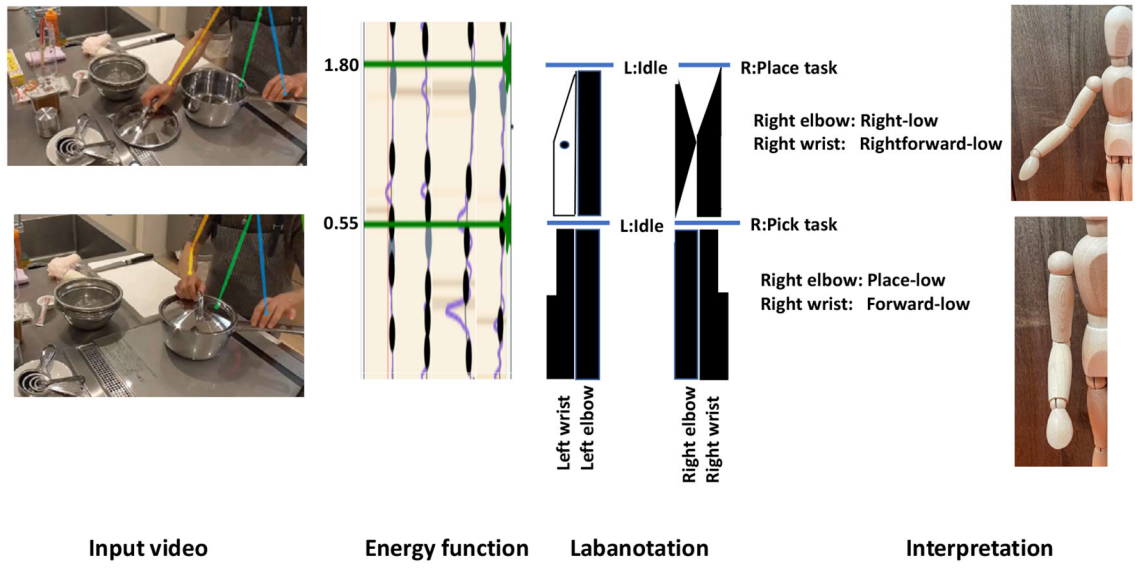

FIGURE 5
Applying LabanSuite to a cooking scene.

memory capacity (Miller, 1956), which is probably why the dance community has represented angles with this granularity.

LabanSuite (Ikeuchi et al., 2018)[2] was developed to detect short pauses, digitize the postures at the pauses, and obtain a Labanotation score from a human movement sequence. There are 2D and 3D versions: the 3D version uses a bone tracking obtained through the Kinect camera sensor and its default SDK,[3] while the 2D version uses a bone tracker based on OpenPose (Cao et al., 2017) and lifting (Rayat Imtiaz Hossain and Little, 2017) from video input. Both versions still extract short stops from the bone motion sequence and digitize postures at the granularity of 8 azimuthal and 5 zenithal angles according to the Labanotation rule.

Figure 5 shows the 2D version applied to a cooking scene. Brief pauses are extracted at the 0.55 and 1.80 seconds and the postures at these timings are described as Labanotation symbols by the LabanSuite. These correspond to the times when the right hand picks the lid of the pod (0.55) and places it on the table (1.80). The right hand at the pick timing is recorded as "Place low" for the right elbow and "Forward low" for the right wrist, and the right hand at the place timing is recorded as "Right low" for the right elbow and "Right forward low" for the right wrist. These postures are recorded in the task models and used for robot execution.

3.2 Body-tool relation

The relationship between the body and the tool when performing a task sequence, especially how the tool is grasped, is an important factor in the success of the task sequence. As shown in Figure 6, even when we grasp the same pen, we grasp it differently depending on the purpose of the task sequence. For example, when pushing, we grab the pen with the whole hand so that we can apply enough force to the pen (Figure 6A), and when pointing, we pick the pen with the fingertips so that we can freely manipulate the direction of the pen (Figure 6B). When writing, we hold it so that we can control the tip of the pen while exerting pressure on it, as shown in Figure 6C.

Various grasping taxonomies have been proposed in the robotics community, starting with Cutkosky's pioneering work (Cutkosky, 1989) to Felix's recent detailed taxonomy (Feix et al., 2015). For LfO, the concept of the contact web (Kang and Ikeuchi, 1997) was used to create a grasp taxonomy. Consistent with the application of the closure theory (discussed later), we use this contact web-based taxonomy and built a recognition system based on it, (a system that classifies an input image into a grasp type in the taxonomy using a CNN). To improve the performance, we used the prior distribution of grasp types (affordance) for each object (Wake et al., 2020).

3.3 Tool-environmental relation

We define a task as a transition in the contact relation between the grasped object and the environment as in Ikeuchi and Suehiro (1994). As an example of a state transition, let us consider a pick task. In Figure 7, the cup on the table is in surface contact with the table surface before the pick task. Surface contact constrains the range of directions an object can move. Let N be the normal direction of the tabletop surface and X be the direction in which the cup can move. Then, the range that X can take can be expressed by Equation (1):

$$N \cdot X \geq 0 \tag{1}$$

The only directions in which the cup can move are up in Figure 7 are upward, i.e., the northern hemisphere of the Gaussian

2 https://github.com/microsoft/LabanotationSuite
3 https://azure.microsoft.com/en-us/products/kinect-dk/

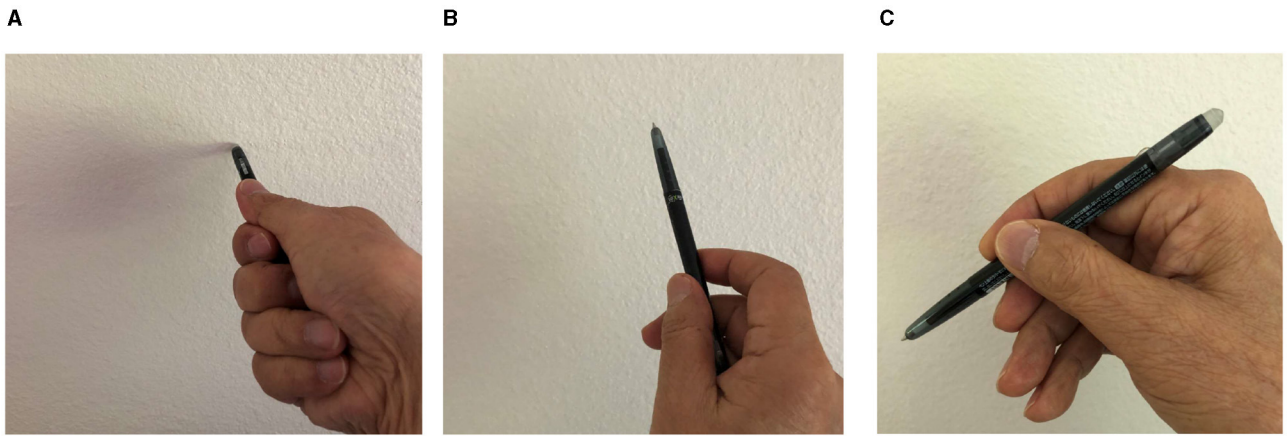

FIGURE 6
The grasping method depends on the purpose of the task sequence. **(A)** For applying power. **(B)** For precise control. **(C)** For versatility.

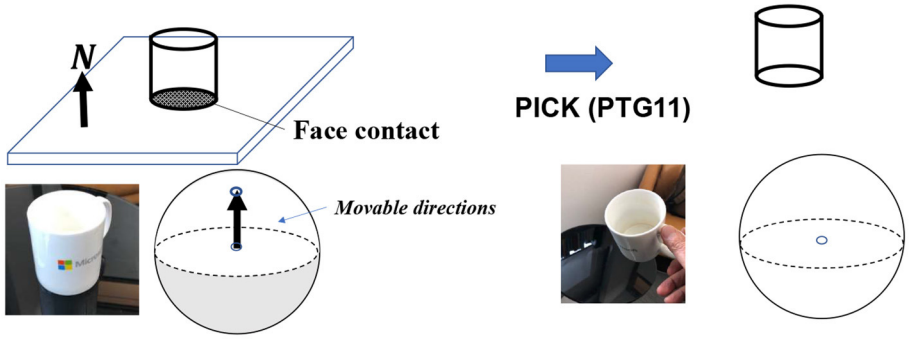

FIGURE 7
Pick task.

sphere, assuming the table surface is facing upward. The Gaussian sphere is used to illustrate the direction of movement; a unit directional vector is represented as a point on the Gaussian sphere. The starting point of the vector is located at the center of the sphere, and the end point on the sphere represents the directional vector. In the example of a cup on a table, the upper directions are the movable directions of the cup, which correspond to the northern hemisphere of the Gaussian sphere shown in white, assuming the surface normal of the table is represented as the north pole of the Gaussian sphere.

As a result of the pick task, the cup is lifted into the air and no longer has the surface contact with the table. As the result, the cup can move in all directions. The entire spherical surface becomes the movable region. This transition from one-directional contact to no contact is defined as the pick task (see Figure 7).

Let us enumerate the surface contact relations to count up how many transitions we have between them in general. For each additional surface contact, the range of motion is further constrained by an additional linear inequality equation corresponding to the surface contact. Adding an inequality equation one by one, the final range over which the object can move can be expressed as the solution of linear simultaneous inequalities given by the set of surface contacts. Using Kuhn-Tucker theory with respect to the solution space of linear simultaneous inequalities, we can count seven characteristic solution spaces, or contact states, for infinitesimal translational motion and seven characteristic solution spaces, or contact states, for infinitesimal rotational motion. Since there are seven possible initial states and seven possible final states, in principle, we have $7 \times 7 = 49$ transitions for translation and 49 transitions for rotation. However, by considering the physical constraints, the number becomes 13 transitions for translation and 14 transitions for rotation (Ikeuchi et al., 2024). Thus, the maximum number of tasks required would be 27 different tasks, assuming all possible transitions.

Further examinations in household tasks using YouTube video as well as our own-recorded cooking video reveal that only a few of them actually occur:

- **PTG1:Pick/Bring/Place**: This group consists of actions such as picking up and placing an object on the desk, which involves eliminating or generating surface contact between the desk and the object. Theoretically speaking, PTG11 (pick) is an action that eliminating surface contact through infinitesimal translation, transitioning the contact state in the direction of motion, from semi-freedom (surface contact) to complete freedom (floating). In the case of an actual action by a robot and a human, a translation motion with a finite interval is always observed immediately after the

infinitesimal translation, so this finite interval is also included in the definition of PTG11 (pick). During this infinitesimal and finite translation, in the direction orthogonal to the motion direction, the complete degree of freedom (floating) is maintained, allowing movement in this orthogonal direction. PTG13 (place) is the reverse of this. PTG12 (bring) involves only a finite translation interval in which surface contact transitions in the direction of motion do not occur at either end of the movement.

- **PTG3:Drawer-Open/Drawer-Adjust/Drawer-Close**: This group includes actions such as opening and closing a drawer, which involves eliminating or generating surface contact at the back of the drawer by moving linearly while being constrained in the orthogonal directions to the motion direction by its surroundings. Theoretically, similar to PTG11, in the direction of motion, the contact state of PTG31 (drawer-open) transitions from semi-freedom to full freedom due to the elimination of surface contact at the bottom of the drawer through infinitesimal translation. Additionally, finite translation is also observed in this case as well. The difference from PTG11 is that in the direction orthogonal to the motion, the contact state remains constrained with no degree of freedom due to the surrounding walls of the drawer, allowing no motion in this direction. PTG31 (drawer-close) is the reverse of this. PTG32 (drawer-adjust) involves only the finite transnational interval, in which contact state transitions in the motion direction do not occur at either end of the movement, while in the direction orthogonal to the motion, the constrained state is maintained.
- **PTG5:Door-Open/Door-Adjust/Door-Close**: In PTG5, the action involves rotation instead of translation, as opposed to PTG3, where surface contact is eliminated or generated through the rotation action in the rotation direction. Theoretically speaking, due to the infinitesimal rotation of PTG51 (door-open), the DOFs in the direction of rotation transits from semi-freedom (surface contact) to full freedom, while the rotation remains constrained in the direction perpendicular to the rotation direction. Similarly, a finite rotation is also included in this case. Conversely PTG53 (door-close) causes the opposite transition. PTG52 (door-adjust) involves only finite rotation, while being constrained perpendicular to the rotation direction, no changes in contact state occur at both ends in the direction of rotation.

Physical transitions other than these rarely occur in household activities, probably because they are difficult for the average unskilled person to perform. In this paper, these nine types of tasks have been prepared as the task models for the corresponding manipulation skill-agent library. Note that the remaining tasks can be added to the library, if needed in the future.

In household tasks, in addition to the physical tasks described above, we need to consider semantic tasks. For example, consider an action such as wiping a table surface with a sponge. Physically, the sponge on the table surface, is in contact with the surface in only one direction. This means that the sponge can be moved upward beside moving on the table surface. However, if the sponge is lifted off the surface of the table, it cannot wipe the surface of the table. In other words, under the task common-sense of wiping, the movement must be such that it always maintains contact with the surface. To express this, we can introduce a virtual surface that exists parallel to the surface of the table. The wiping motion can be described by considering that the sponge can only move in directions between this virtual surface and the original physical table surface. We will refer this virtual surface as a semantic constraint surface. By examining the actions in the household operations, the following five semantic constraints were obtained.

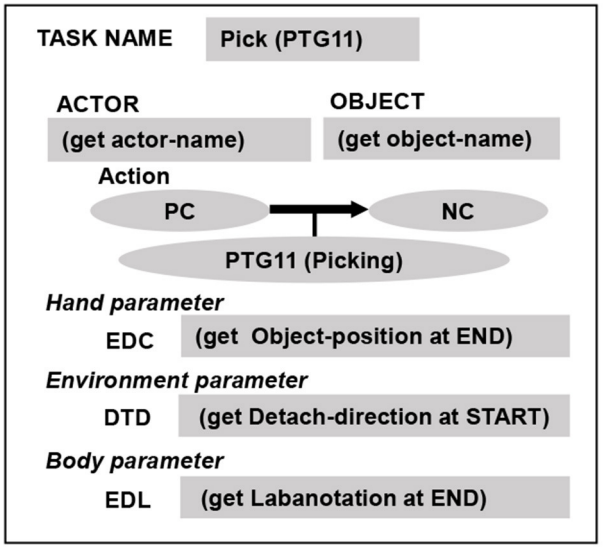

FIGURE 8
An illustrative example of pick (PTG11) task model.

- **SGT1: Semantic ping**: A task such as carrying a glass of juice can be described by a semantic ping. Physically, the glass can rotate about any axis in the air. However, in order to carry the juice without spilling it, the glass is only allowed to rotate about the axis along the direction of the gravity and not in any other axis direction. We assume that the semantic ping stands perpendicular to the surface of the glass. The object is only allowed to rotate around the semantic ping.
- **STG2: Semantic wall**: In a task such as wiping a table, only the directions of motion along the surface of the table, i.e., between the actual table surface and the semantic surface, is allowed.
- **STG3: Semantic tube**: Tasks such as peeling in cooking require translational motion along a specific trajectory on the object surface. This motion can be interpreted as the one along a semantic tube.
- **STG4: Semantic sphere**: In the STG2 semantic wall, motion was constrained between two planes. In the case of wiping spherical surfaces, motion is constrained between two spherical surfaces, a real sphere and a semantic sphere.
- **STG5: Semantic hing**: In a task such as pouring water from one pitcher into another, the motion of the pitcher must be a rotational motion around the spout, meaning rotation around the semantic hinge at the spout.

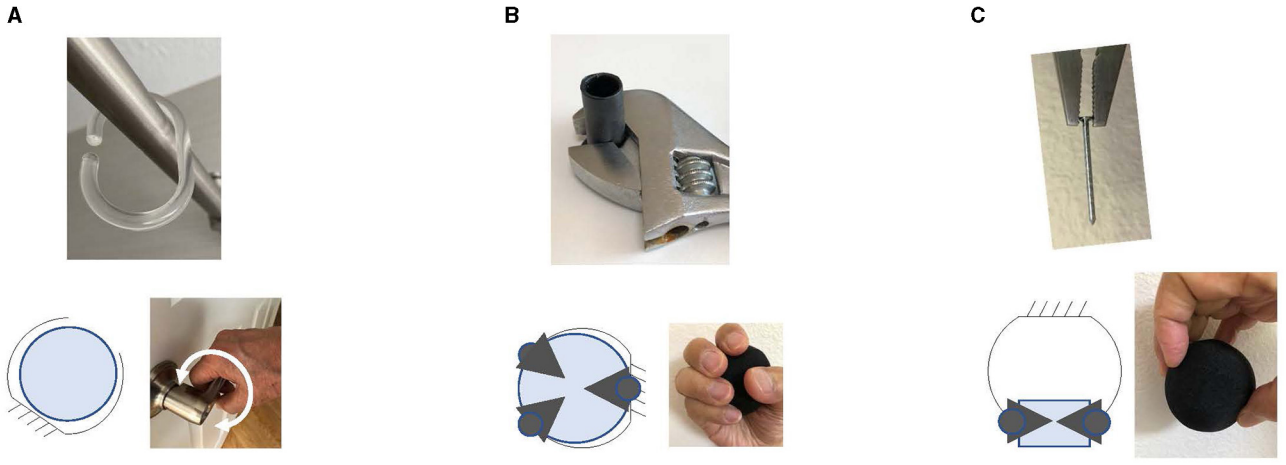

FIGURE 9
Three closures. **(A)** Passiveform closure. **(B)** Passiveforce closure. **(C)** Activeforce closure.

Task models are designed in the design phase, with the name and slots for the skill parameters essential for task execution, as in a Minsky's frame. Figure 8 shows Pick (PTG11) as an example task model. The task name is registered as Pick (PTG11), and the transition of surface contacts is described as PC (partial contact), i.e., one-directional contact, to NC (no contact). This transition information will not be used for execution, but is retained to help humans understand the transition states more easily. The actor slot describes whether to execute with the right or left hand. A slot to register the name of the object is also prepared, which is used for MS-Custom vision[4] to identify the location of the object in the image. Slots are also prepared for the first and last Labanotation, the initial position of the object, and the detaching direction and distance, in which direction and how much the object will be lifted up. Daemons are attached to these slots to observe and obtain these parameters from the demonstration.

During the teaching mode, the task encoder first recognizes which task it is from the verbal input and instantiates the corresponding task model, i.e., determining *what-to-do*. At the same time, it also obtains the name of the target object from the verbal input. Next, the instantiated task model collects the necessary skill parameters from human demonstration according to each daemons attached at the slots in the task model, i.e., determining *where-to-do*.

4 Skill-agent library

In order for a robot to be able to perform the corresponding motion from the recognized task model, an execution module corresponding to each task model, a module to know *how-to-do*, is necessary. In this section, we design the grasp skill library and the manipulation skill library which contain such execution modules, which we refer to as skill-agents. The core of each skill-agent consists of a control agent, Bonsai brain,[5] with a policy trained using Bonsai reinforcement learning system. Each skill-agent also includes interfaces to retrieve necessary skill parameters from the corresponding task model as well as to obtain additional parameters needed at run time.

4.1 Grasp-skill-agent library

4.1.1 Closure theory and contact web

In Robotics community, Feix et al. (2015) proposed a grasp taxonomy containing 33 grasp patterns. However, in terms of robot execution of task sequences, these 33 grasp patterns are redundant and can be aggregated using the closure theory. For example, Thumb-4 finger and Thumb-3 finger can achieve almost the same goal in the task sequence following the grasp. According to the closure theory, the grasping task can be classified into the following three objectives (Yoshikawa, 1999).

- **Passive form closure:** to maintain the constant position of a grasped object by bringing the hand into a particular shape without actively applying force on the grasped object. Examples include wheel bearing in a wheel (see Figure 9A).
- **Passive force closure:** to hold a grasped object in place without loosening its motion by applying force from all directions. Examples include a vice (see Figure 9B).
- **Active force closure:** to allow fingertip manipulation of the grasped object while grasping it (see Figure 9C).

To be able to perform these three types of closures, we assume a particular gripper and a particular shaped object, and computed contact-webs, the distribution of contact points between the object and the gripper. In this paper, Shadow-hand Lite[6] is used as the

4 https://www.customvision.ai/

5 Bonsai is a reinforcement learning package developed in Microsoft and Bonsai brain refers to an agent equipped with a trained policy, which works in the Bonsai environment.

6 https://www.shadowrobot.com/dexterous-hand-series/

gripper and the shape of an object is assumed to be able to be approximated as a superquadric surface. Three kinds of contact-webs are computed for each object shape: a passive-form contact web, a passive-force contact web, and an active-force contact web.

In order for a robot to perform a grasping task, the skill-agent must obtain the location of the contact webs from the visual information at run time to guide. As shown in Figure 10, the visual and grasping modules are designed as an integral part in the grasping skill-agent. The visual CNN determines the contact-web on the target object, and then the grasping brain relies on this contact-web location to control the hand. Such pipelines are prepared for each of the three contact-webs. At runtime, a specific grasp pipeline is activated by a grasp task model. Note that only three pipelines are prepared that are independent of object shape and size, since differences in shape and size are absorbed by the randomization in CNN training and reinforcement learning (Saito et al., 2022).

4.1.2 Contact-web localizer

The contact-web localizer consists of a multi-layer CNN that takes a depth image as input and outputs the contact-web locations. In order to obtain contact web locations without making a priori assumptions about the shape and size of an object, the depth image training data was generated by randomly sampling the size, shape, and viewing parameters of a superquadric surface, which can approximate a variety of objects by varying the shape and size parameters. Specifically, the shape parameters, the size parameters, the azimuth angle, and the zenith angle are randomly selected from the range of 0 to 1.0, 10 cm to 30 cm, between plus and minus 120 degrees from the front, and 0 to 90 degrees, respectively. For each closure, the positions of the contact-web were calculated analytically based on the superquadric equation of these parameters using the specific robot hand (in this paper, the Shadow-hand). The axis directions and the origin of the coordinate system are also given as true values so that the object-centered coordinate system of the superquadric surface can also be output.

4.1.3 Grasp skill-agents

Each grasp-skill agent contains a Bonsai brain, an agent with a policy for controlling the hand. The brain is pre-trained offline using the Bonsai reinforcement learning system to control the robot's hand movements. Note that for reusability at the hardware level, only the hand movements are trained, not the whole arm, to take advantage of the effectiveness of the role-division algorithm (Sasabuchi et al., 2021), which will be discussed later. The brain receives the positions of the contact-web from the contact-web localizer and the approach direction from the demonstration as hint information with respect to the object centered coordinates. The observable states include the current joint positions of the hand and the drag force derived from the effort at each joint. Since it is not practical to attach a force sensor to each fingertip, we decided to measure the drag force on a fingertip using the effort value instead. The effort is defined as the amount of electrical current required by the joint motor to move the joint to the target position. The greater the reaction force, the more current is required. The grasp task is considered complete when the drag force on the fingertip, as measured by the effort, is sufficient. The reward is the success or failure of the pick after the grasp.

4.2 Manipulation-skill-agent library

The manipulation skill-agents are also implemented as brains using the Bonsai reinforcement learning system. In the manipulation skill-agents, objects can be assumed to be already grasped. The hint information is the direction of motion of the grasped object by human demonstration. The observation states include the current joint positions and the drag force from the environment of the grasped object. The drag force is assumed to be obtained from a force sensor attached to the arm. The tasks of Place (PTG13), Drawer-closing (PTG33), and Door-closing (PTG53) are terminated when drag force from the wall or table surface is detected. The tasks of Pick (STG11), Door-opening (PTG31), and Drawer-opening (PTG51) ends when drag force is eliminated and the hand position given in the demonstration is reached. The reward for PTG13 is given by whether the object is stable when released, while the reward for the remaining tasks is given by whether the terminal condition is met.

5 Implementation

The system consists of two main modules:

- **Task encoder:** Recognizes tasks from verbal input, obtains skill parameters required for each task from visual input, and completes a sequence of task models.
- **Task decoder:** Based on the task model sequence, pick the corresponding skill-agents from the libraries and generates robot motions by the skill-agents.

5.1 Task encoder

An Azure Kinect sensor (see text footnote[3]) was positioned to provide a full view of the site through the entire demo, capturing RGBD images of the demonstrator and the object as well as the audio input of the demonstrator. The resolution of the images was $1,280 \times 720$ and the nominal sampling rates of the video and audio were 30 Hz and 48,000 Hz, respectively. AR markers were placed to align the orientation of the demonstration coordinates with the robot coordinates. Note that the role of the AR markers is to align the rough orientation of both coordinates, and the objects, human and robot positions do not need to be exactly the same during demonstration and execution.

One unit of robot teaching begins with the grasping task of an object, followed by several manipulation tasks of the object, and ends with the releasing task of the object. This sequence of grasping, manipulation, and releasing tasks is referred to as a GMR operation, and each GMR operation is assumed to manipulate the same object with the same hand.

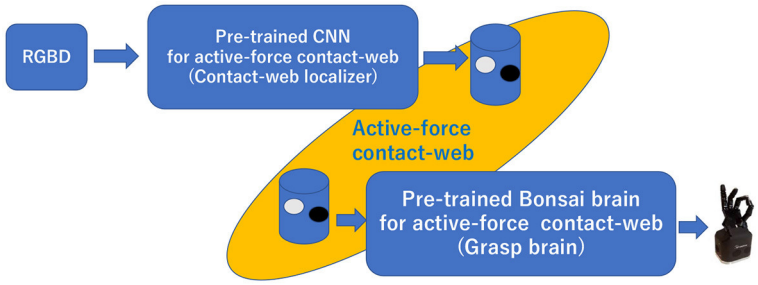

FIGURE 10
A grasp-pipe line. The observation module and the execution brain are designed as an integrated unit. The observation module consists of a CNN that determines the positions of the contact web from the scene, and the execution brain is trained using reinforcement learning to control the hand based on the positions of the contact web and the force feedback.

To facilitate teaching the granularity of tasks in one GMR operation, a stop-and-go method is used (Wake et al., 2022). That is, for each task, the user verbally indicates the task to be performed, and starts moving his/her hand. Then, when the demonstration of that task is completed, the user stops moving his/her hand and, after a brief stop, begins the next cycle of verbal instruction and visual demonstration corresponding to the next task. This series of verbal instructions and visual demonstrations is repeated until the GMR operation is completed. For example, the GMR operation "pick up a cup and carry it to the same table" consists of five cycles: grasp the cup (passive-force grasp), pick it up (PTG11), bring it carefullly (STG12), place it (PTG13), and release it (release).

Figure 11 shows an example of human demonstration. The video and audio are segmented at the timings when the hand stops (Figure 11A). To detect these timings from the input video, the brightness disturbance of the input video is characterized (Ikeuchi and Suehiro, 1994) (Figure 11B). For the calculation, the RGB image is converted to a YUV image and the Y channel is extracted as brightness. The brightness image is spatially filtered using a moving average of a 50 × 50 window, and the absolute pixel-by-pixel difference between adjacent frames is taken. The average of the differences is taken as the brightness perturbation at each timing. After removing outliers and low-pass filtering at 0.5 Hz, the local minima are extracted as the stop timings. The input video and audio are segmented based on these stop timings. This video-based segmentation is preferred over audio-based segmentation because, although human verbal instructions and hand demonstrations are roughly synchronized, their synchronization is not exact, and accurate video timings are needed for skill parameter extraction, such as hand waypoints (Figure 11C).

The segmented video and audio are processed in two steps for task encoding. The first step is task recognition based on the audio segments, and the second step is skill parameter extraction based on the video segments. The segmented audio is recognized using a cloud-based speech recognition service.[7] In addition, the fluctuations in the user's verbal instructions are absorbed by a learning system based on our own crowdsourced data. The segmented audio segments, corresponding video segments, and recognition results can be previewed and modified by the user via the GUI.

In the second step, the task-encoder instantiates a sequence of task models based on these recognition results. Each task model has a Minsky frame-like format with slots for storing skill parameters, which will be collected from the video segments. The skill parameters are mainly related to hand and body movements with respect to the target object. The hand movements are extracted using a 2D hand detector[8] and depth images, while the body movements are obtained by the LabanSuite.

The image of the operating hand in the last frame of the grasping video is given to the grasp recognition system, which determines the grasp type. From the hand movements immediately before the end of a task, the approach direction for a grasping task, the attaching direction for PTG13 (place), PTG33 (drawer-closing), and PTG53 (door-closing) are calculated in the object-centered coordinate system. The hand movements immediately after the start of a task also provide the departure direction for a releasing task and the detaching direction for PTG11 (pick), PTG13 (drawer-opening), and PTG15 (door-opening). Human body movements are also important skill parameters. From the first and last postures of a human arm, we estimate the 3D posture of the performer and convert them into Labanotation using LabanSuite. These skill parameters are stored at each corresponding slot of a task model for later task decoding.

5.2 Task decoder

We developed the TSS system, a task decoding platform that allows a robot to execute task models given by the encoder (Sasabuchi et al., 2023). The TSS system reads a set of task models from the task encoder, coordinates the parameters among the task models, calls the skill-agents with Bonsai brains corresponding to the task models, and passes the parameters stored in the task models to the skill-agents. TSS is also equipped with the Labanotation-based IK. Generally, humanoids have many degrees of freedom, and there are multiple IK solutions to bring the end effector to one position. To resolve this redundancy, an initial

7 https://azure.microsoft.com/en-us/products/congnitive-service/speech-to-text/

8 https://www.ultralytics.com/

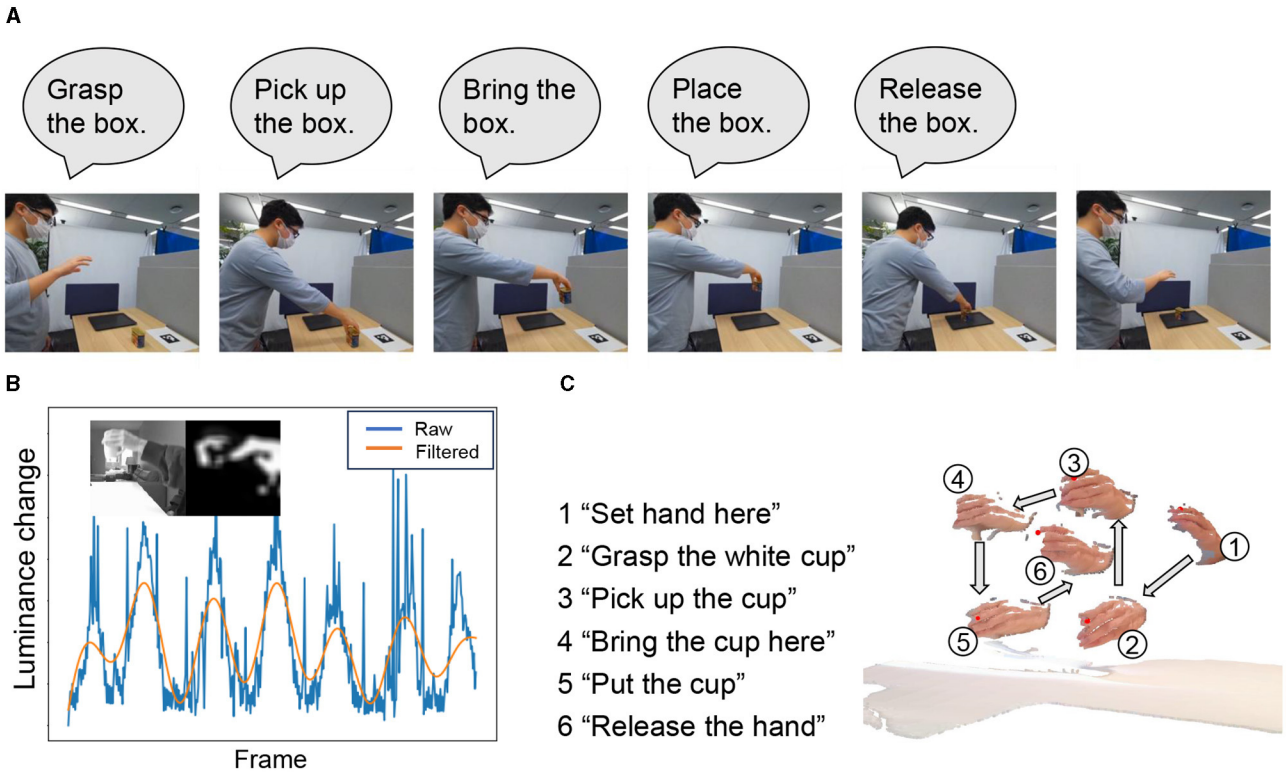

FIGURE 11
Example of the task encoder computation: **(A)** An example of a stop-and-go demonstration given to the system (Box-displacement demo). **(B)** Video segmentation based on the time-series luminance changes in the demonstration video. The top-left images show an example of the pixel-wise absolute difference between adjacent luminance images. **(C)** Extracted hand positions (Cup-relocation demo). Images were modified from Wake et al. (2022).

solution to IK, a rough arm shape, is determined based on the given Labanotation, and a neighboring solution is obtained using IK from the rough initial approximation given by the Labanotation. The TSS system can simulate movements with a virtual robot displayed in an unreal-engine environment or control a real robot via ROS.

The role-division algorithms in TSS improves reusability at the hardware level (Sasabuchi et al., 2021; Takamatsu et al., 2022). All skill-agents in the libraries used by TSS are those trained to control robot hand movements, not the movements of the entire robot as a whole. To realize these hand movements, the robot arm must form certain poses given by the inverse kinematics (IK) of the robot. The shape of the arm, on the other hand, must be as close as possible to the shape of the human demonstration as represented by the Labanotation score; the adjustment between the arm shape given from the Labanotation score and the arm shape required by the Bonsai brain is done using the Role-Division algorithm. Using this method, when introducing a new robot, only the Labanotation-based IK needs to be replaced, without need for retraining of skill-agents.

Three Bonsai brains in the three grasping skill-agents are directly connected to their vision modules to determine the contact-web locations and the local coordinate system, as described in the grasp skill-agent library section. This is because although the target objects are placed under nearly identical positions during the demonstration and execution, the robot's position is not exactly the same as the human demonstration position, so that the target object must be observed again during the execution in order to perform the robot grasp robustly. From the grasping task model, the decoder activates the RGBD sensor and detects the bounding box of the object using MS custom vision. Next, a plane fitting based on RANSAC (random sample consensus) (Fischler and Bolles, 1981) is applied to the depth image withing the bounding box so as to segment the object from the background table. The segmented depth image is then given to the trained CNN, which outputs the contact-web locations as well at the local coordinate system of the superquadric, from which the approach direction is re-calculated. Finally, the Bonsai brain receives those locations and the direction as hint information, and generates hand and finger movements based on force feedback.

Figure 12 shows the summary of the information flow to/from the Bonsai brain performing a grasping task. Each brain is designed to output hand movements using the demonstration parameters as hint values and the forces from the environment as state values. To minimize the differences between Sim2Real and between different sensors, the values from the force sensor are not used directly, but are processed to see if they exceed a certain threshold value. According to the hand position specified by the Bonsai brain, IK is solved by relying on human Labanotation. The robot's overall posture is also determined using the robot's mobile and lifting mechanisms as well to

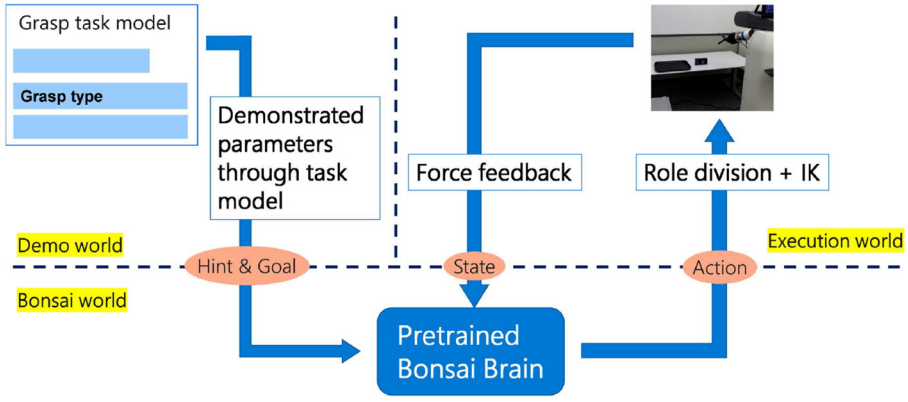

FIGURE 12
Bonsai brain and parameters.

satisfy the required shoulder position by the role-division algorithm.

6 Robot execution

To emphasize reusability at the hardware level (Takamatsu et al., 2022), two robots, Nextage and Fetch, both running in ROS, were used as testbeds. Nextage by Kawada Robotics has two arms, each with six degrees of freedom, with one degree of freedom (rotation around a vertical axis) at the waist. In this experiment, the robot did not use its left arm or waist, working only with its right arm, which is equipped with a Shadow Dextrous Hand Lite from Shadow Robotics as a robotic hand. Nextage is equipped with a stereo camera to observe the environment. The Fetch Mobile Manipulator has 7 DOFs in the arms with 1 DOF at the waist (vertical movement) and 2 DOFs in the mobile base. The Fetch robot is also equipped with a Shadow Dexterous Hand Lite. It is also equipped with a Primesense Carmine 1.09 RGB-D camera for environmental monitoring. In this paper, as an example of hardware-level reusabiliy, the same task-model sequence, generated from a demonstration, is given to the TSS to control the two robots, which share the same skill-agent libraries, with only different IKs. For videos of the robot executions, please refer to our website.[9]

6.1 Box-placement GMR operation

A box-displacement GMR operation is demonstrated in front of Azure Kinect with the verbal instructions, as shown in Figure 11A. The task model sequence, consisting of:

1. grasp the box (active-force closure),
2. pick up the box from the desk (PTG11),
3. bring-carefully the box (STG12),
4. place the box on a plate (PTG13), and
5. release the box (active-force closure).

9 https://www.microsoft.com/en-us/research/project/interactive-learning-from-observation/

was obtained and initiated from the verbal input, and then, corresponding skill parameters for each task models are obtained from the visual demonstration. The task model sequence was uploaded to Azure and, then down loaded to the local TSS at the Shinagawa site controlling Nextage in Shinagawa as shown in the upper raw of Figure 13. In the lower raw of Figure 13, the same task model sequence on Azure was downloaded to the local TSS at the Redmond site controlling Fetch at Redmond to perform the tasks. These two TSSs share the same skill-agent libraries, and differ only in the IKs to control the robots.

The success ratio during execution was approximately 90%. The main cause of the failures was due to errors in the positions of the contact web points obtained by the active-force grasp-agent at the time of execution. Another cause of failure was that the object was positioned too far from the edge of the table, making it impossible to solve the inverse kinematics (IK).

6.2 Shelf GMR operation

The task sequence was identified, from the verbal instruction shown in the upper raw of Figure 14, as:

1. grasp the cup (Passive-force closure),
2. pick up the cup (PTG11),
3. bring-carefully the cup (STG12),
4. bring-carefully the cup (STG12),
5. bring-carefully the cup (STG12),
6. place the cup (PTG13), and
7. release the cup (Passive-force closure).

Note that in this operation, bring-carefully STG12 is used three times in a row to teach the robots the trajectory to avoid collision with the shelf. In other words, the demonstrator, not the system, plans the collision avoidance path and teaches the collision avoidance path to the system according to Reddy's 90% AI rule (Reddy, 2007). See the lower two rows of Figure 14 for robot's execution.

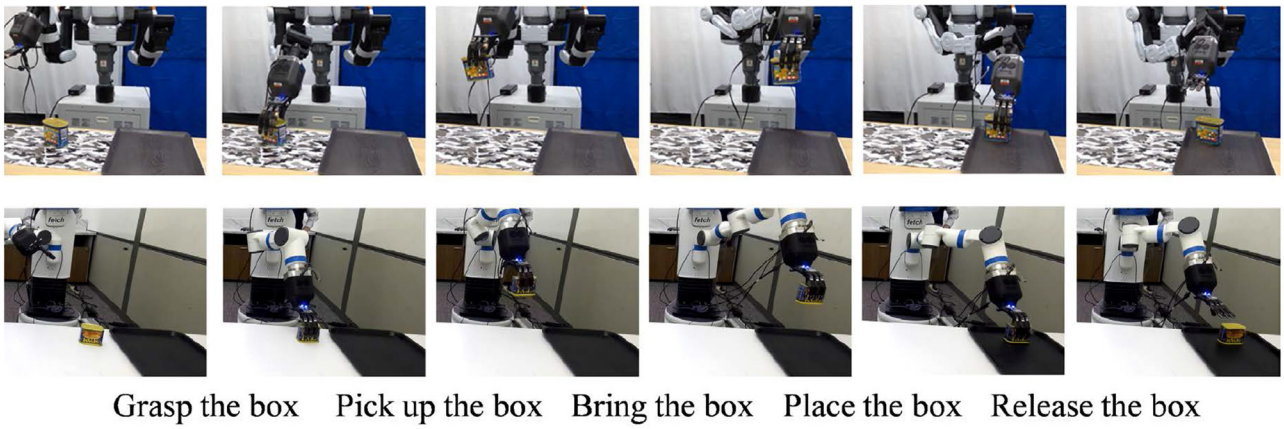

FIGURE 13
Executions of box-displacement GMR operation.

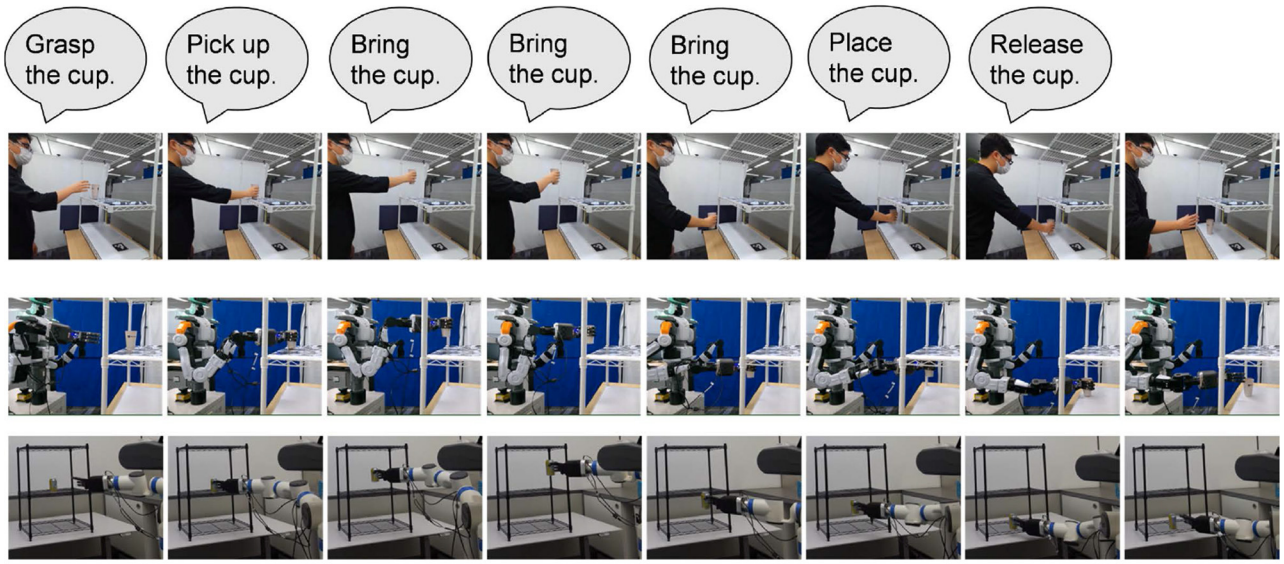

FIGURE 14
Shelf GMR operation.

6.3 Garbage-disposal GMR operation

The garbage-disposal GMR operation in Figure 15 consists of:

1. grasp the can (Active-force closure),
2. pick up the can (PTG11),
3. bring the can (STG12), and
4. release the can (Active-force closure).

The difference between the box-placement operation and this garbage-disposal operation is whether the the object is placed and then released or not placed and then released in the air. It is interesting to note that even if the order of tasks does not change that much, the purpose of the task sequence can change significantly by simply shaving off some of the tasks.

7 Summary and discussions

This paper introduces a system for automatically generating robot programs from human demonstrations, referred to as learning-from-observation (LfO), and describes three task common-sense requirements for applying this system to the household domain. Unlike the more common learning-from-demonstration or programming-by-demonstration approaches, LfO does not attempt to directly mimic human trajectories. Instead, LfO semantically maps human behavior to robot behavior. That is, human behavior is first transformed into machine-independent representations, referred to as task models, based on verbal and visual input. This representation consists of frameworks that specify *what-to-do*, referred to as task models, with associated skill parameters that specify *where-to-do*. Notably, the skill parameters

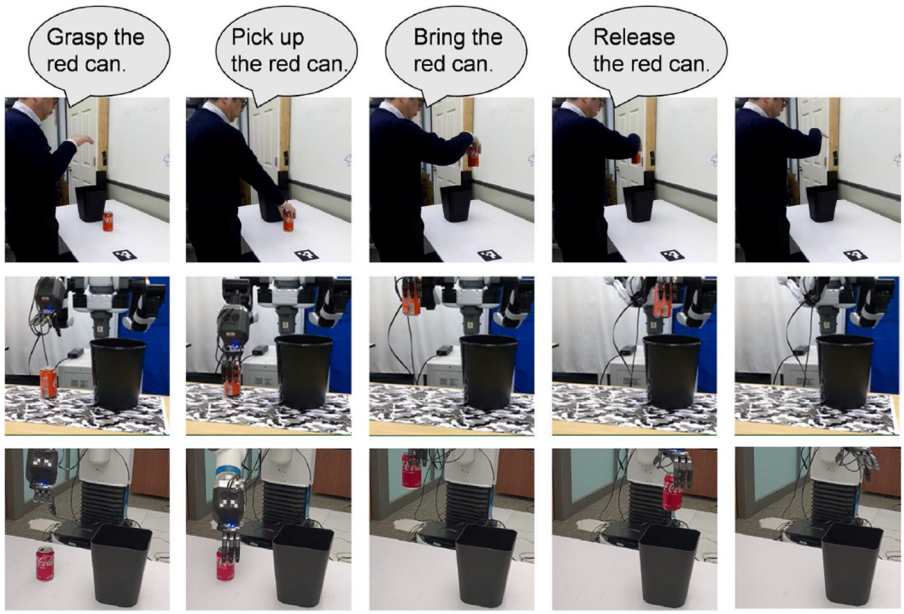

FIGURE 15
Garbage-disposal GMR operation.

are not the trajectories themselves, but rather a collection of local features with respect to the object and the task, obtained from the trajectories, a collection of important relative trajectories with respect to the object and the task, such as from which direction to approach for grasping the object in the grasp task or to which direction to approach for placing the object to the table in the place task. These task models are then mapped to the primitive actions of each robot using the skill-agent libraries, a collection of skill-agents that encompass *how-to-do*, designed and trained using reinforcement learning to each robot hardware. This indirect mapping aims to overcome the kinematic and dynamic differences between humans and robots.

The purpose of this indirect mimicking can also be rephrased as distinguishing between reusable and non-reusable information in the demonstration, focusing only on the reusable information. The other information for execution is generated during the robot's execution based on local force feedback and other factors. For representing this reusable information, LfO uses three types of representations: Labanotation, Contact-web, and face-contact relations.

For instance, let's reconsider Labanotation, which expresses whole-body postures. As reusable information, the sequence of postures (key poses) at specific sampling times (brief stop timings) should be used, while the trajectories between those postures do not need to be reused. This is because it is impossible to mimic everything due to differences in hardware between the demonstrator and the robot, i.e., differences in weight, height and arm lengths between them. Therefore, it is necessary to express only the key points that make the movements appear similar to a human spectator. For example, for human dance, in the robot dance uploaded on YouTube (see text footnote[1]), the robot's key poses mimic the keyposes of Master Yamada on the left. On the other hand, the movements of the robot's various part between key poses are generated to make stable postures each time for maintaining balance and preventing falling. Therefore, the detailed movements of the robot itself differ from Master Yamada's movements. However, after watching this video, there is little sense of discrepancy between the two, at least for the authors.

The tendency of humans to focus only on the sequence of key poses at key timings was also observed in Perera et al. (2009). This also aligns with the Gestalt view of movements, which suggests the humans perceive a dance as a sequence of key poses as a whole. Therefore, according to the Gestalt point of view, when generating robot movements from human demonstrations, it is not necessary to mimic the entire time-series trajectory; instead, this suggests the approach that only imitates key poses, which in fact we follow in the dancing robot and this household robot. Similar reasoning has been applied in the design of the contact-web and face-contact relation based on the distinction between re-usable and non-usable information.

As mentioned in the introduction, LfO is a teaching system based on Reddy (2007)'s 90% AI, where the system gets hints from human demonstrations to solve difficult problems such as determination of a grasp strategy, a collision avoidance path and postures toward the target object. For example, in collision avoidance, as explained in the shelf demonstration in Section 6.2, the human demonstrator consciously presents key way-points for collision avoidance, such as first moving the object out of the shelf, then adjusting its height only after it is far enough to avoid collisions with the shelf. Namely, the human demonstrator solves the collision avoidance and provides the global guidance of the collision-free path to the system. The system, then, automatically generates a trajectory passing though these way-points according to the global guidance, where three Bring-carefully (STG12) tasks, instead of one Bring (PTG12) task, are generated for the collision avoidance purpose from the

human demonstration. This 90%AI concept was also used in the Labanotation-based IK, where the system solves redundance-degree IK from the initial pose given from the Labanotation. These 90% methods, even in the automatic generation system, utilize human demonstrations as initial solutions, making it easier to reach a global solution in problems with multiple local solutions. Improving the interaction between humans and systems based on 90%AI approach is one of the key approaches for human-robot collaboration.

In this LfO project, we have chosen to focus on home service robots in care-giving sector, as a non-traditional robot application, i.e., non-industrial domain, which faces a shortage of labor despite the aged society. Unlike industrial environment, various objects are present in the proximity environment, requiring the system to pay attention to specific relationships. We proposed Labanotation, contact-web, and face-contact relation as representations of relationships for this focus-of-attention. Similarly, agriculture can be considered another non-industrial domain (Lauretti et al., 2023, 2024). This field presents additional challenges such as handling flexible objects and the need for advanced sensing technologies due to the outdoor environment. Nevertheless, considering that this field also involves cluttered environments, we believe that the representations proposed in this paper can essentially used to focus attention and extract only re-usable information in these environments. Applying such new service robot concepts to these fields is a direction worth pursuing in the future.

Data availability statement

The raw data supporting the conclusions of this article will be made available by the authors, without undue reservation.

Ethics statement

Written informed consent was obtained from the individual(s) for the publication of any potentially identifiable images or data included in this article.

Author contributions

Theory and the whole system architecture is designed by KI. The robot implementation is done by JT and KS. Vision module is designed by NW and AK. All authors contributed to the article and approved the submitted version.

Funding

The author(s) declare that no financial support was received for the research, authorship, and/or publication of this article.

Conflict of interest

KI, JT, KS, NW, and AK were employed by Microsoft.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

References

Akgun, B., Cakmak, M., Jiang, K., and Thomaz, A. L. (2012). Keyframe-based learning from demonstration: Method and evaluation. *Int. J. Soc. Robot.* 4, 343–355. doi: 10.1007/s12369-012-0160-0

Asfour, T., Gyarfas, F., Azad, P., and Dillmann, R. (2008). Imitation learning of dual-arm manipulation tasks in humanoid robots. *Int. J. Human. Robot.* 5, 289–308. doi: 10.1142/S0219843608001431

Billard, A., Calinon, S., Dillmann, R., and Schaal, S. (2008). "Robot programming by demonstration," in *Springer Handbook of Robotics* (Cham: Springer), 1371–1394. doi: 10.1007/978-3-540-30301-5_60

Cao, Z., Simon, T., Wei, S.-E., and Sheikh, Y. (2017). "Realtime multi-person 2d pose estimation using part affinity fields," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 7291–7299. doi: 10.1109/CVPR.2017.143

Cutkosky, M. R. (1989). On grasp choice, grasp models, and the design of hands for manufacturing tasks. *IEEE Trans. Robot. Autom.* 5, 269–279. doi: 10.1109/70.34763

De Mello, L. H., and Sanderson, A. C. (1990). And/or graph representation of assembly plans. *IEEE Trans. Robot. Autom.* 6, 188–199. doi: 10.1109/70.54734

Dillmann, R., Asfour, T., Do, M., Jäkel, R., Kasper, A., Azad, P., et al. (2010). Advances in robot programming by demonstration. *KI-Künstliche Intell.* 24, 295–303. doi: 10.1007/s13218-010-0060-0

Feix, T., Romero, J., Schmiedmayer, H.-B., Dollar, A. M., and Kragic, D. (2015). The grasp taxonomy of human grasp types. *IEEE Trans. Hum. Mach. Syst.* 46, 66–77. doi: 10.1109/THMS.2015.2470657

Fischler, M. A., and Bolles, R. C. (1981). Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM* 24, 381–395. doi: 10.1145/358669.358692

Hutchinson-Guest, A. (1970). *Labanotation: The System of Analyzing and Recording Movement*. New York: Theatre Arts Books.

Ikeuchi, K., Ma, Z., Yan, Z., Kudoh, S., and Nakamura, M. (2018). Describing upper-body motions based on Labanotation for learning-from-observation robots. *Int. J. Comput. Vis.* 126, 1415–1429. doi: 10.1007/s11263-018-1123-1

Ikeuchi, K., and Reddy, R. (1991). "Assembly plan from observation," in *Annual Research Review* (CMU-RI).

Ikeuchi, K., and Suehiro, T. (1994). Toward an assembly plan from observation I. task recognition with polyhedral objects. *IEEE Trans. Robot. Autom.* 10, 368–385. doi: 10.1109/70.294211

Ikeuchi, K., Wake, N., Sasabuchi, K., and Takamatsu, J. (2024). Semantic constraints to represent common sense required in household actions for multimodal learning-from-observation robot. *Int. J. Rob. Res.* 43, 134–170. doi: 10.1177/02783649231212929

Kang, S. B., and Ikeuchi, K. (1997). Toward automatic robot instruction from perception-mapping human grasps to manipulator grasps. *IEEE Trans. Robot. Autom.* 13, 81–95. doi: 10.1109/70.554349

Kuniyoshi, Y., Inaba, M., and Inoue, H. (1994). Learning by watching: extracting reusable task knowledge from visual observation of human performance. *IEEE Trans. Robot. Autom.* 10, 799–822. doi: 10.1109/70.338535

Lauretti, C., Tamantini, C., Tomé, H., and Zollo, L. (2023). Robot learning by demonstration with dynamic parameterization of the orientation: an application to agricultural activities. *Robotics* 12:166. doi: 10.3390/robotics12060166

Lauretti, C., Tamantini, C., and Zollo, L. (2024). A new dmp scaling method for robot learning by demonstration and application to the agricultural domain. *IEEE Access.* 12, 7661–7673. doi: 10.1109/ACCESS.2023.3349093

Lozano-Perez, T. (1983). Robot programming. *Proc. IEEE* 71, 821–841. doi: 10.1109/PROC.1983.12681

Lozano-Perez, T., Mason, M. T., and Taylor, R. H. (1984). Automatic synthesis of fine-motion strategies for robots. *Int. J. Rob. Res.* 3, 3–24. doi: 10.1177/027836498400300101

Marr, D. (2010). *Vision: A Computational Investigation Into the Human Representation and Processing of Visual Information*. Cambridge, MA: MIT press. doi: 10.7551/mitpress/9780262514620.001.0001

Miller, G. A. (1956). The magical number seven, plus or minus two: some limits on our capacity for processing information. *Psychol. Rev*. 63, 81. doi: 10.1037/h0043158

Minsky, M. (1988). *Society of Mind*. New York: Simon and Schuster. doi: 10.21236/ADA200313

Nakaoka, S., Nakazawa, A., Kanehiro, F., Kaneko, K., Morisawa, M., Hirukawa, H., et al. (2007). Learning from observation paradigm: leg task models for enabling a biped humanoid robot to imitate human dances. *Int. J. Rob. Res*. 26, 829–844. doi: 10.1177/0278364907079430

Okamoto, T., Shiratori, T., Kudoh, S., Nakaoka, S., and Ikeuchi, K. (2014). Toward a dancing robot with listening capability: keypose-based integration of lower-, middle-, and upper-body motions for varying music tempos. *IEEE Trans. Robot*. 30, 771–778. doi: 10.1109/TRO.2014.2300212

Perera, M., Kudoh, S., and Ikeuchi, K. (2009). Keypose and style analysis based on low-dimensional representation. *J. Inf. Proc. Soc. Japan* 50, 1234–1249.

Piaget, J. (2020). *La psychologie de l'intelligence*. Paris: Dunod.

Ramachandran, V. S., and Blakeslee, S. (1998). *Phantoms in the Brain*. New York, NY: William Morrow.

Rayat Imtiaz Hossain, M., and Little, J. J. (2017). Exploiting temporal information for 3D pose estimation. *arXiv e-prints arXiv-1711*.

Reddy, R. (2007). "To dream the possible dream," in *ACM Turing Award Lectures*, 1994. doi: 10.1145/1283920.1283952

Saito, D., Sasabuchi, K., Wake, N., Takamatsu, J., Koike, H., and Ikeuchi, K. (2022). "Task-grasping from a demonstrated human strategy," in *Proceedings of the International Conference on Humanoids*. doi: 10.1109/Humanoids53995.2022.10000167

Samejima, K., Katagiri, K., Doya, K., and Kawato, M. (2006). Multiple model-based reinforcement learning for nonlinear control. *Electr. Commun. Japan* 89, 54–69. doi: 10.1002/ecjc.20266

Sasabuchi, K., Saito, D., Kanehira, A., Wake, N., Takamatsu, J., and Ikeuchi, K. (2023). Task-sequencing simulator: Integrated machine learning to execution simulation for robot manipulation. *arXiv preprint arXiv:2301.01382*.

Sasabuchi, K., Wake, N., and Ikeuchi, K. (2021). Task-oriented motion mapping on robots of various configuration using body role division. *IEEE Robot. Autom. Lett*. 6, 413–420. doi: 10.1109/LRA.2020.3044029

Sato, Y., Bernardin, K., Kimura, H., and Ikeuchi, K. (2002). "Task analysis based on observing hands and objects by vision," in *IEEE/RSJ International Conference on Intelligent Robots and Systems* (IEEE), 1208–1213.

Schaal, S. (1996). "Learning from demonstration," in *Advances in Neural Information Processing Systems*, 9.

Schaal, S. (1999). Is imitation learning the route to humanoid robots? *Trends Cogn. Sci*. 3, 233–242. doi: 10.1016/S1364-6613(99)01327-3

Schaal, S., Ijspeert, A., and Billard, A. (2003). Computational approaches to motor learning by imitation. *Philos. Trans. R. Soc. London* 358, 537–547. doi: 10.1098/rstb.2002.1258

Takamatsu, J., Morita, T., Ogawara, K., Kimura, H., and Ikeuchi, K. (2006). Representation for knot-tying tasks. *IEEE Trans. Robot*. 22, 65–78. doi: 10.1109/TRO.2005.855988

Takamatsu, J., Ogawara, K., Kimura, H., and Ikeuchi, K. (2007). Recognizing assembly tasks through human demonstration. *Int. J. Rob. Res*. 26, 641–659. doi: 10.1177/0278364907080736

Takamatsu, J., Sasabuchi, K., Wake, N., Kanehira, A., and Ikeuchi, K. (2022). Learning-from-observation system considering hardware-level reusability. *arXiv preprint arXiv:2212.09242*.

Tsuda, M., Takahashi, T., and Ogata, H. (2000). Generation of an assembly-task model analyzing human demonstration. *J. Robot. Soc. Japan* 18, 535–544. doi: 10.7210/jrsj.18.535

Wake, N., Kanehira, A., Sasabuchi, K., Takamatsu, J., and Ikeuchi, K. (2022). Interactive learning-from-observation through multimodal human demonstration. *arXiv preprint arXiv:2212.10787*.

Wake, N., Sasabuchi, K., and Ikeuchi, K. (2020). "Grasp-type recognition leveraging object affordance," in *HOBI Workshop, IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*.

Yoshikawa, T. (1999). Passive and active closures by constraining mechanisms. *J. Dyn. Sys. Meas. Control*. 121, 418–424. doi: 10.1115/1.2802490