

# Interactive Task Encoding System for Learning-from-Observation

Naoki Wake<sup>1</sup>, Atsushi Kanehira<sup>1</sup>, Kazuhiro Sasabuchi<sup>1</sup>, Jun Takamatsu<sup>1</sup>, and Katsushi Ikeuchi<sup>1</sup>

**Abstract**—We present the Interactive Task Encoding System (ITES) for teaching robots to perform manipulative tasks. ITES is designed as an input system for the Learning-from-Observation (LfO) framework, which enables household robots to be programmed using few-shot human demonstrations without the need for coding. In contrast to previous LfO systems that rely solely on visual demonstrations, ITES leverages both verbal instructions and interaction to enhance recognition robustness, thus enabling *multimodal LfO*. ITES identifies tasks from verbal instructions and extracts parameters from visual demonstrations. Meanwhile, the recognition result was reviewed by the user for interactive correction. Evaluations conducted on a real robot demonstrate the successful teaching of multiple operations for several scenarios, suggesting the usefulness of ITES for multimodal LfO. The source code is available at <https://github.com/microsoft/symbolic-robot-teaching-interface>.

## I. INTRODUCTION

Household robots with manipulation capabilities are increasingly being considered as an alternative labor force in various settings including home environments [1]. While typical robotic systems assume that a robot performs specific operations in a fixed environment, household systems need to provide the ability to adjust operations to fit the user's needs and environment. Learning-from-Observation (LfO) is a framework that aims to teach manipulative operations through human demonstrations without coding, making it a promising solution for programming household robots [2].

In LfO, a human demonstration is encoded into an intermediate representation of object manipulation, so-called a *task model* (Fig. 1). The task model consists of a sequence of primitive robot actions, so-called *tasks*, and parameters to achieve the tasks, so-called *skill parameters*. Because the task model is an abstract representation of object operations, the encoded task model is theoretically applicable to multiple environments and arbitrary hardware.

Although studies have shown successful implementation of LfO systems in various settings, they have been primarily based on visual demonstration. For instance, vision-based LfO systems have been developed for part assembly [3]–[5], knot tying [6], and dancing [7]–[9]. On the other hand, studies have shown that human uses verbal instructions to make teaching more interactive and efficient [10]–[12]. Inspired by the nature of language, we have developed an LfO system that utilizes both visual and verbal information, namely, *multimodal LfO* [13], [14]. However, existing multimodal LfO systems have focused on robust visual recognition and information enrichment, and applications that take advantage of interactivity have not yet been proposed.

This paper aims to present a practical pipeline of the task-model encoder that employs interaction (Fig. 1). We

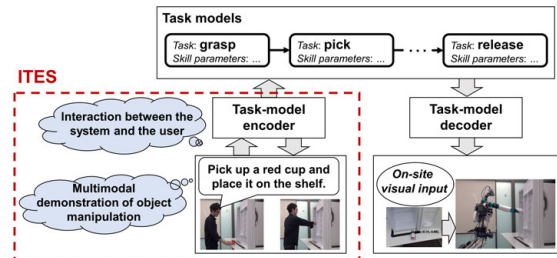


Fig. 1. Overview of the robot teaching framework, so-called Learning-from-Observation (LfO). The red box indicated the Interactive Task Encoding System (ITES) for LfO. ITES encodes multimodal human demonstrations into a sequence of primitive robot actions, referred to as *task models*.

refer to the encoder as the Interactive Task Encoding System (ITES). To enable users to teach at the task granularity level, we adopt a simple method of “stop-and-go demonstration.” In the stop-and-go demonstration, a user pauses the hand motions when tasks switch. At every pause, the demonstrator gives a verbal instruction for the next task before resuming the hand movements. Given the stop-and-go demonstration, ITES recognizes tasks and skill parameters and outputs a task model, with improved stability achieved through GUI-based interactive correction. The contributions of this paper include proposing a practical pipeline for multimodal LfO with interaction capability and demonstrating the system’s effectiveness on a real robot with a dexterous hand.

## II. RELATED WORKS

### A. Role of language in recognition

Natural language has been shown to be useful in guiding vision systems. For example, we have previously shown that the use of verbal input can help a robot-vision system determine the timings and the location of object grasping and thus make the recognition robust [13]. In addition, verbal instructions include semantic constraints in manipulations that are not explicitly taught through visual demonstration [15].

Language can be applied not only to vision-system guidance but also to interactive education. Recent research has proposed interactive systems that utilize human language for remembering users’ previous inputs [16], operations in new environments [17], and clarifying uncertain instructions [18]. However, to our knowledge, no multimodal LfO system has been proposed that actively utilizes user interaction. This study proposes a system that implements user interaction to effectively utilize verbal instructions for task recognition.

### B. Robot task planning from language

Starting with the work of the SHRDLU [19], the problem of task planning based on language instructions has been

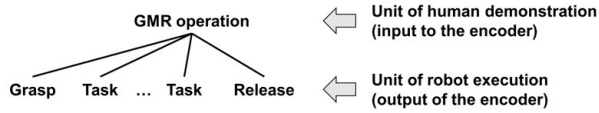


Fig. 2. Manipulative operations with different granularity.

studied for decades. Before the development of Natural language processing (NLP) technology, the dominant method of understanding sentences was by parsing verbs and syntax (e.g., [20], [21]). In recent years, the use of NLP technology has become more common, and systems have been proposed to analyze abstract instructions [22], [23] and complex instructions, including conditional branching [24].

Multimodal robot teaching has been studied in both end-to-end approaches based on large models [22], [25]–[29] and symbolic robot teaching approaches including multimodal LfO [14], [30]–[32]. However, the previous LfO systems used a simple knowledge base that maps tasks to verbs in a single sentence, which limits the flexibility of input texts and the length of task sequences to be taught at once. In this paper, we aim to teach tasks of arbitrary length through more flexible sentences by utilizing a step-by-step teaching method and a task recognition model based on NLP.

### III. TEACHING STRATEGY FOR MULTIMODAL LFO

In the context of robot teaching, the granularity gap between human demonstrations and robot execution can be problematic. This section proposes a teaching method to bridge this gap.

#### A. Unit of human demonstration and robot execution

In this paper, we define that a unit of human demonstration starts with grasping an object followed by several manipulative tasks, and ends with releasing the object. We call this unit a grasp-manipulation-release (GMR) operation. We believe that various manipulative household operations can be achieved as a result of multiple GMR operations. For example, cleaning the table after a meal can be broken down into the GMR of a plate for clearing dishes and the GMR of a sponge for wiping the table. Note that we focused on single-armed operations and assumed that only one object is being manipulated in a GMR operation.

On the other hand, the unit of robot execution, *task*, can be finer than GMR operations. In a typical LfO, a task is defined as a transition of a target object’s state. For example, we have proposed a task set based on the types of motion constraints imposed on the object [15]. Consequently, a GMR operation is divided when the motion constraints to the object change. As an example, a GMR operation of “picking up a cup and carrying it on the same table” is divided into the tasks of picking up the object (PTG11) bringing it (PTG12), and placing it (PTG13) (symbols in parentheses are from [15]). Table. I shows examples of GMR operations and their task components. In this paper, we also include grasp and release in the task set because these actions involve a transition in the contact state between the robot hand and the object.

TABLE I: Examples of GMR operations (modified from [32])

GMR operation	Explanation
Grasp-PTG11-PTG12-PTG13-Release	pick, bring, and place something
Grasp-PTG31-Release	slide something open
Grasp-PTG33-Release	slide something close
Grasp-PTG51-Release	rotate something open
Grasp-PTG53-Release	rotate something close

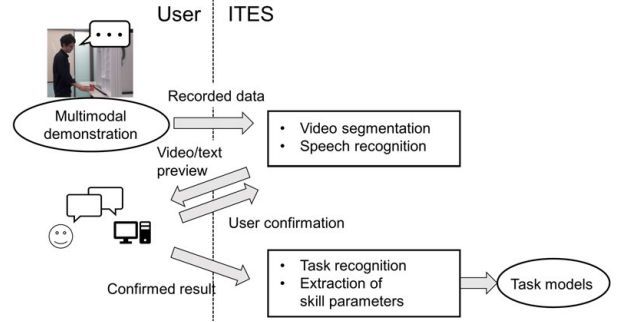


Fig. 3. The pipeline of ITES, which involves interaction with users.

#### B. Stop-and-go teaching to resolve granularity differences

We designed a teaching method that allows users to effectively teach at the granularity of tasks. First, to enable users to teach multiple tasks in a GMR operation at once, we adopted a stop-and-go demonstration technique. This method involves pausing the hand movements when tasks switch in order to inform the system of the task transition. For instance, in a pick-and-place operation, the demonstrator stops the hand motion when approaching the object, grasping it, lifting it off the table, carrying it above the table, placing it on the table, releasing it, and moving the hand to a home position.

Additionally, we adopted a teaching method that alternates between visual demonstration and verbal instruction, rather than simultaneously. This is because simultaneous teaching is considered a form of dual tasking that requires a higher cognitive load for the user. Such teaching methods may not be suitable for inexperienced users, especially for the elderly [33]. To clearly convey the relationship between verbal and visual demonstrations, the demonstrator provides step-by-step verbal instruction for the next task at each pause, rather than giving verbal instructions all at once before or after visual demonstrations.

### IV. IMPLEMENTATIONS OF ITES

This section describes the implementation of ITES (Fig. 3). ITES comprises two stages, one before and one after user interaction. In the initial stage, ITES detects times when a manipulating hand stops and segments the video and speech accordingly. The speech is transcribed using a third-party speech recognizer, and the result of segmentation and speech recognition are previewed for user modification if necessary. In the later stage, an NLP-based recognizer identifies the task sequence from the verbal input, and skill parameters are extracted for each task to compile a task model.

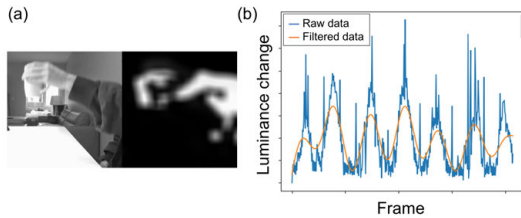


Fig. 4. (a) An example of the pixel-wise absolute difference between adjacent luminance images (right panel). In the video, the hand is picking up the cup. (b) Time series of the luminance change for a stop-and-go demonstration.

### A. System input

We use an Azure Kinect sensor [34] to record RGB-D images and a speech input during the demonstration. The sensor is placed in positions that could capture the entire demonstration. The image resolution is 1280x720, and the nominal sampling rate for the video and speech is 30 Hz and 48000 Hz, respectively. Noise in the speech was suppressed using noise filters to improve speech recognition [35].

### B. Video segmentation and speech recognition

Given a stop-and-go demonstration, ITES segments the video and audio at the times when a manipulating hand stopped. To detect the times, we characterized the intensity of hand motions based on the changes in luminance [36]. For this calculation, RGB images are converted to YUV images, and the Y channel is extracted as the luminance. The luminance images are spatially filtered using a moving average of 50x50 window, and the pixel-wise absolute difference is calculated between adjacent frames (Fig. 4(a)). The mean of the difference is taken as the luminance change at each time. After removing outliers and applying a low-pass filter of 0.5 Hz, the local minima of the time series are extracted as the stop timings (Fig. 4(b)). After the confirmation of the split timings (in Sec. IV-C), the RGB-D video and the audio are segmented based on the detected timings. The split speech is transcribed using a third-party cloud speech recognition service [37] and previewed to the user for additional confirmation.

### C. Previewing for the user modification

Since the speech content and the correspondence between video and audio are critical to task recognition and the extraction of skill parameters, we implemented a feature that allows users to modify the computation result of video segmentation and speech recognition. After the luminance-based video segmentation, the user is prompted for each split video. The user can ignore videos unrelated to the teaching (e.g., movements before and after a GMR operation) and merge over-split videos through a button-based GUI (Fig. 5 (a)). Note that the user gives the demonstration again when the under-segmentation occurred.

After confirming the video segmentation, the video and audio are split again, and the resulting audio segments are transcribed into text using a speech recognition service [37].

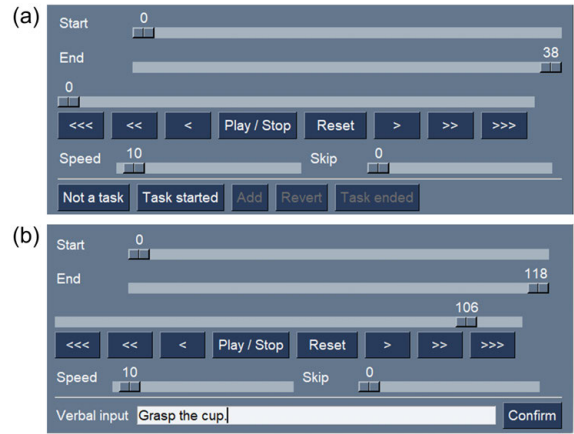


Fig. 5. User interfaces for checking the computation result of video segmentation and speech recognition. (a) An interface to ignore and merge videos that were segmented by ITES. (b) An interface to correct speech recognition results.

The transcribed text is then displayed to the user in order so that the user can modify through a GUI (Fig. 5 (b)).

### D. Task recognition

After the interactive correction, each task is recognized based on the transcribed texts using a language-based recognition model. To train the model, we manually annotated an existing video dataset of preparing breakfast [38]. We chose the cooking domain because cooking needs the use of a variety of foods and tools with manipulation. We labeled the video with task labels and prepared the video dataset of a single task using a third-party video annotation tool. The dataset contained 12 task classes with a total of 1340 videos. Table II shows the task classes and the number of data.

We collected motion instructions for the egocentric videos of each task using a crowdsourcing service, called Amazon Mechanical Turk. Specifically, 100 instruction sentences were collected for each task. Then, we trained a recognition model to associate each instruction with its corresponding task. To this end, we prepared a random forest model trained on top of a fixed BERT model [39]. We collected egocentric videos to match the perspective of the demonstrator during teaching, rather than from a third-person perspective. Table III shows examples of sentences collected by different cloud workers. We observed variations in the verbs and nouns that appear in the instructions. The confusion matrix of task recognition is shown in Fig. 6, where 10% of the sentences were used for testing. We conducted ten-fold cross-validation and obtained an average performance of 83%. This result suggests that the system can robustly recognize tasks from natural verbal instructions with variations.

### E. Skill parameter extraction

Every task requires skill parameters for robots to decode. [13], [14]. Once a task is determined, processes called daemon run to extract the parameters by analyzing the corresponding video. Here we briefly explain the examples of skill parameters and computation for extracting the parameters.

TABLE II: Statics of the labeled tasks. Task symbols in [15] are indicated in parentheses.

Task label	Count	Mean length (S.E.) (sec)
Picking (PTG11)	122	0.63 (0.03)
Bringing (PTG12)	108	1.46 (0.20)
Placing (PTG13)	113	0.73 (0.05)
Rotating_hinge_to_open (PTG51)	38	1.14 (0.14)
Rotating_hinge_to_close (PTG53)	32	0.75 (0.11)
Wiping (STG2)	30	5.59 (1.42)
Peeling (STG3)	24	3.55 (0.90)
Pouring (STG5)	42	2.26 (0.54)
Holding (STG6)	239	3.50 (0.32)
Cutting (MTG1)	22	4.52 (0.75)
Grasping	464	0.56 (0.02)
Releasing	392	0.37 (0.01)

TABLE III: Examples of sentences collected using a crowd-sourcing service.

A video of opening a refrigerator door
“Open the refrigerator door.”
“Pull to open the fridge door.”
“Grab the refrigerator handle and pull it to open the door.”
A video of washing a place with a sponge
“Wipe the dish with the sponge in an anti-clockwise direction.”
“Clean the plate with the sponge in your hand.”
“Wipe the plate with the sponge.”
A video of pouring water from a kettle
“Pour water from the kettle into the mug.”
“Pour water from the pitcher into the cup.”
“Empty the water from the kettle into a cup.”

#### 1) The name and the 3D positions of the target object:

The name of the target object is used to help recognize other skill parameters. We predefined a set of object name and assumed that users would specify object names verbally. In the current ITES, an object name is extracted using a third-party language parser [40]. The 3D positions of the grasped object is calculated from the RGB-D images.

2) *Hand laterality*: Hand laterality is crucial information that relates to other skill parameters, such as grasp type and approach direction to the target object. To extract hand laterality, the video of grasping the object is analyzed. Assuming a constant object location, the 2D location of the object is extracted from the first RGB image using an object detector. In the last frame, the 2D locations of both hands are detected using a hand detector, and the hand used for

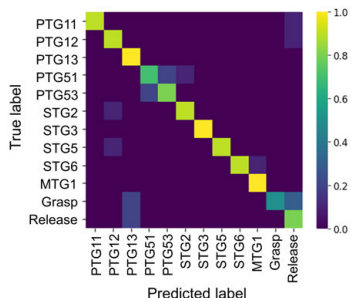


Fig. 6. Confusion matrix of task recognition from texts.

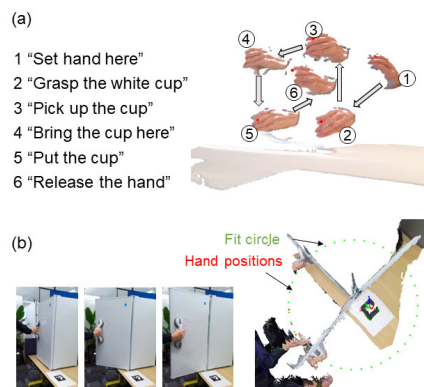


Fig. 7. (a) Extracted hand positions and (b) the estimation of the rotating\_hinge task.

manipulation is identified as the hand closer to the object.

3) *Grasp type*: Grasp type is critical for successful task execution of tasks followed by the grasp [41]. ITES detects the grasp type using an image classifier model. The image of the manipulating hand at the last frame of the grasping video is input into the model, and the grasp type is determined. The likelihood of grasp types associated with the object name is also considered to improve the accuracy of the classification [42], [43].

4) *Hand positions and hand trajectory*: The demonstrator’s hand motion is crucial for successful task completion and avoiding collisions with the environment, as well as for manipulating articulated objects such as doors or shelves. ITES extracts 3D hand positions during the demonstration using the 2D hand detector and depth images (Fig. 7(a)). For rotating\_hinge tasks, the hand trajectory is parameterized by applying circular fitting to the hand positions during the corresponding video (Fig. 7(b)).

5) *Human pose*: Human pose contains implicit knowledge for achieving tasks efficiently [44]. Following our previously proposed task model design [14], we encode the human arm postures at the start and end of each task video. The 3D poses of the demonstrator are estimated using a third-party 3D pose estimator [45], and each of the body parts (upper/lower arms of left and right arms) is encoded into spatially digitized 26-point directions on the unit sphere.

## V. EXPERIMENTS

We tested the proposed ITES based on the performance of multimodal LfO. To this end, we prepared a multimodal LfO system by integrating ITES into a task-model decoder that we had previously implemented. The control policies of robots were trained using reinforcement learning [41], [46] on top of our in-house simulator [47]. Since ITES is designed to be applicable to a wide range of home environments, the LfO system was qualitatively examined from two aspects: (1) if the system applies to a wide variety of GMR operations by combining tasks, (2) if the system provides flexibility for users to adjust a GMR operation in different scenes.

To check the (1) system applicability, we tested three GMR operations commonly observed in household situa-

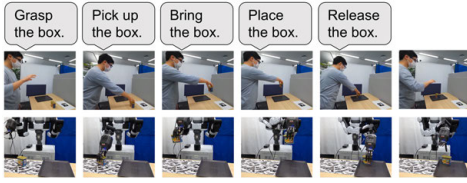


Fig. 8. Results of multimodal LfO for pick-carry-place a box.

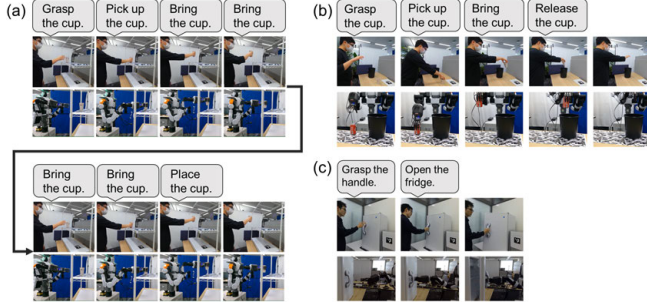


Fig. 9. Results of multimodal LfO for the GMR operation of (a) pick-carry-place a cup with multiple bringing tasks in-between, (b) throwing away a cup, and (c) opening a door.

tions: “pick-carry-place a box,” “throw away a cup,” and “open a fridge.” To check the (2) system flexibility, we considered a case of a pick-carry-place operation that accompanies multiple *bring* (PTG12) tasks to avoid obstacles. We tested the system with a humanoid robot [48] and a dexterous robot hand with four fingers [49].

#### A. GMR of pick-bring-place an object

Fig. 8 shows the overview of teaching a “pick, carry, and place a box” operation. This GMR operation consists of *grasp*, *pick* (PTG11), *bring from one location to another* (PTG12), *place* (PTG13), and *release* tasks. The verbal input and the visual demonstration are shown at the top of the figure. The result shows that the demonstration was successfully executed by the robot. To check the system flexibility, we tested the “pick-carry-place” operation with a box placed on a shelf (Fig. 9(a)). For this environment, several waypoints are required to avoid collision with the shelf, thus user instructed multiple *bring* (PTG12) tasks in-between the *pick* (PTG11) and *place* (PTG13) tasks. The result shows that the multiple bring tasks can be executed by the robot, suggesting that the system has the flexibility to allow users to adjust GMR operations to different scenes.

#### B. GMR of throwing away a cup and opening a door

We tested the applicability of the system to a variety of GMR operations by teaching additional tasks, including the operations of throwing a cup away (Fig. 9(b)) and opening a door (Fig. 9(c)). The throwing operation consists of *grasping*, *picking* (PTG11), *bringing from one location to another* (PTG12), and *releasing* tasks. The opening operation consists of *grasping*, *opening to a certain range* (PTG5), and *releasing* tasks. Those operations were completed by the robot, suggesting that the proposed LfO system can operate a variety of operations by composing the tasks.

#### C. Failure cases of task-model recognition

During the experiment, we observed several failure cases. One is the case where the recorded visual demonstration was not segmented correctly due to inefficient hand-stop time. Additionally, we observed several cases when skill-parameter extraction failed due to the incorrect recognition of hands, objects, grasp types, and human poses. Such misrecognition was typically caused by occlusion. For example, an occluded human hand caused the failure of extracting hand position and trajectory. In those cases, the user needed to discard the demonstration and start over.

### VI. DISCUSSION

In this paper, we proposed ITES, a pipeline of the task-model encoder for multimodal LfO. By assuming alternating stop-and-go visual demonstration and step-by-step verbal instruction, GMR operations can be taught at the granularity of tasks while taking correspondence between visual and verbal inputs. Additionally, ITES featured an interaction function in order to allow the user to modify the recognition results (Fig. 3). Experiments tested the applicability and flexibility of ITES for various GMR operations in different scenes. As a result, a robot successfully executed the taught GMR operations. Although not included in this paper, we have additionally confirmed that the encoded GMR operations were operated by another robot with different degrees of freedom (results are shown in [46]). These results suggest the usefulness of ITES.

The design philosophy of LfO is similar to low-code/no-code software development (LCSD), which enables non-programmers to participate in software development with minimal coding through a visual platform [50], [51]. Likewise, LfO allows non-programmers to program household robots without coding. However, robot manipulation requires many skill parameters that are difficult to represent in languages, such as arm postures and hand trajectories; some of them are based on common sense and may not be consciously apparent to the demonstrator [52]. Thus, specifying such parameters solely on a visual platform can be time-consuming. ITES addresses this challenge by extracting the parameters from visual demonstration. Our approach extends LCSD to handle multimodal input.

The method of providing multimodal input is not unique. As described in Sec. III-B, the present ITES employed a teaching method that alternates between visual demonstration and verbal instruction. This approach has the advantage of explicitly teaching the correspondence between verbal and visual inputs while limiting the cognitive load on the user. However, some users may prefer simultaneous teaching to reduce teaching time, while others may prefer to give all verbal instructions at the beginning or the end. A further user study is needed to determine which teaching method is appropriate from a usability perspective.

#### ACKNOWLEDGMENTS

We thank Dr. Sakiko Yamamoto (Ochanomizu University), Dr. Etsuko Saito (Ochanomizu University), and Dr. Midori

Otake (Tokyo Gakugei University) for their help and advice in creating the video dataset of tasks.

## REFERENCES

- [1] C. A. Smarr, T. L. Mitzner *et al.*, “Domestic robots for older adults: attitudes, preferences, and potential,” *International journal of social robotics*, vol. 6, no. 2, pp. 229–247, 2014.
- [2] K. Ikeuchi and T. Suehiro, “Toward an assembly plan from observation. i. task recognition with polyhedral objects,” *IEEE TRO*, vol. 10, no. 3, pp. 368–385, 1994.
- [3] J. Takamatsu, K. Ogawara *et al.*, “Recognizing assembly tasks through human demonstration,” *The International Journal of Robotics Research*, vol. 26, no. 7, pp. 641–659, 2007.
- [4] C. Pérez D’Arpino and J. A. Shah, “C-learn: Learning geometric constraints from demonstrations for multi-step manipulation in shared autonomy,” in *ICRA*, 2017, pp. 4058–4065.
- [5] Y. Zhu, C. Jiang *et al.*, “Inferring forces and learning human utilities from videos,” in *CVPR*, 2016, pp. 3823–3833.
- [6] J. Takamatsu, T. Morita *et al.*, “Representation for knot-tying tasks,” *IEEE TRO*, vol. 22, no. 1, pp. 65–78, 2006.
- [7] K. Ikeuchi, Z. Ma *et al.*, “Describing upper-body motions based on labanotation for learning-from-observation robots,” *IJCV*, vol. 126, no. 12, pp. 1415–1429, 2018.
- [8] S. Nakaoka, A. Nakazawa *et al.*, “Learning from observation paradigm: Leg task models for enabling a biped humanoid robot to imitate human dances,” *IJRR*, vol. 26, no. 8, pp. 829–844, 2007.
- [9] T. Okamoto, T. Shiratori *et al.*, “Extraction of person-specific motion style based on a task model and imitation by humanoid robot,” in *IROS*. IEEE, 2014, pp. 1347–1354.
- [10] D. Soto and G. W. Humphreys, “Automatic guidance of visual attention from verbal working memory,” *Journal of Experimental Psychology: Human Perception and Performance*, vol. 33, no. 3, p. 730, 2007.
- [11] L. M. Justice, P. C. Pullen, and K. Pence, “Influence of verbal and nonverbal references to print on preschoolers’ visual attention to print during storybook reading,” *Developmental Psychology*, vol. 44, no. 3, p. 855, 2008.
- [12] G. Lupyán and M. J. Spivey, “Making the invisible visible: Verbal but not visual cues enhance visual detection,” *PLoS one*, vol. 5, no. 7, p. e11452, 2010.
- [13] N. Wake, I. Yanokura *et al.*, “Verbal focus-of-attention system for learning-from-demonstration,” in *ICRA*. IEEE, 2021.
- [14] N. Wake, R. Arakawa *et al.*, “A learning-from-observation framework: One-shot robot teaching for grasp-manipulation-release household operations,” in *SII*. IEEE, 2021.
- [15] K. Ikeuchi, N. Wake *et al.*, “Semantic constraints to represent common sense required in household actions for multi-modal learning-from-observation robot,” *arXiv preprint arXiv:2103.02201*, 2021.
- [16] H. Brock and R. Gomez, “Personalization of human-robot gestural communication through voice interaction grounding,” in *IROS*. IEEE, 2021, pp. 846–853.
- [17] T. V. Ilyevsky, J. S. Johansen, and J. M. Siskind, “Talk the talk and walk the walk: Dialogue-driven navigation in unknown indoor environments,” in *IROS*. IEEE, 2021, pp. 4903–4910.
- [18] D. Lanza, R. Menicatti, and A. Sgorbissa, “Abductive recognition of context-dependent utterances in human-robot interaction,” in *IROS*. IEEE, 2020, pp. 10975–10981.
- [19] T. Winograd, “Understanding natural language,” *Cognitive psychology*, vol. 3, no. 1, pp. 1–191, 1972.
- [20] B. J. Thomas and O. C. Jenkins, “Roboframenet: Verb-centric semantics for actions in robot middleware,” in *ICRA*. IEEE, 2012, pp. 4750–4755.
- [21] S. Tellex, T. Kollar *et al.*, “Understanding natural language commands for robotic navigation and mobile manipulation,” in *AAAI*, vol. 25, no. 1, 2011, pp. 1507–1514.
- [22] M. Ahn, A. Brohan *et al.*, “Do as i can, not as i say: Grounding language in robotic affordances,” *arXiv preprint arXiv:2204.01691*, 2022.
- [23] N. Wake, A. Kanehira *et al.*, “Chatgpt empowered long-step robot control in various environments: A case application,” *arXiv preprint arXiv:2304.03893*, 2023.
- [24] P. Pramanick, H. B. Barua, and C. Sarkar, “Decomplex: Task planning from complex natural instructions by a collocating robot,” in *IROS*. IEEE, 2020, pp. 6894–6901.
- [25] H. Ito, H. Ichiwara *et al.*, “Integrated learning of robot motion and sentences: Real-time prediction of grasping motion and attention based on language instructions,” in *ICRA*. IEEE, 2022, pp. 5404–5410.
- [26] C. Lynch and P. Sermanet, “Language Conditioned Imitation Learning Over Unstructured Data,” in *Proceedings of Robotics: Science and Systems*, Virtual, July 2021.
- [27] E. Jang, A. Irpan *et al.*, “BC-z: Zero-shot task generalization with robotic imitation learning,” in *CoRL*, 2021.
- [28] S. Stepputtis, J. Campbell *et al.*, “Language-conditioned imitation learning for robot manipulation tasks,” *Advances in Neural Information Processing Systems*, vol. 33, pp. 13 139–13 150, 2020.
- [29] C. Lynch and P. Sermanet, “Language conditioned imitation learning over unstructured data,” *arXiv preprint arXiv:2005.07648*, 2020.
- [30] N. Gopalan, E. Rosen *et al.*, “Simultaneously Learning Transferable Symbols and Language Groundings from Perceptual Data for Instruction Following,” in *Proceedings of Robotics: Science and Systems*, 2020.
- [31] S. Venugopalan, M. Rohrbach *et al.*, “Sequence to sequence-video to text,” in *ICCV*, 2015, pp. 4534–4542.
- [32] I. Yanaokura, N. Wake *et al.*, “A multimodal learning-from-observation towards all-at-once robot teaching using task cohesion,” in *SII*. IEEE, 2022, pp. 367–374.
- [33] P. Verhaeghen, D. W. Steitz *et al.*, “Aging and dual-task performance: a meta-analysis,” *Psychology and aging*, vol. 18, no. 3, p. 443, 2003.
- [34] “Azure kinect dk — microsoft azure,” <https://azure.microsoft.com/en-us/products/kinect-dk/>, (accessed Jan. 5, 2023).
- [35] N. Wake, M. Fukumoto *et al.*, “Enhancing listening capability of humanoid robot by reduction of stationary ego-noise,” *TEEE*, vol. 14, no. 12, pp. 1815–1822, 2019.
- [36] K. Ikeuchi and T. Suehiro, “Towards an assembly plan from observation. i. assembly task recognition using face-contact relations (polyhedral objects),” in *ICRA*, 1992, pp. 2171–2172.
- [37] “Speech to text — microsoft azure,” <https://azure.microsoft.com/en-us/products/cognitive-services/speech-to-text/>, (accessed Jan. 5, 2023).
- [38] A. Saudabayev, Z. Rysbek *et al.*, “Human grasping database for activities of daily living with depth, color and kinematic data streams,” *Scientific data*, vol. 5, no. 1, pp. 1–13, 2018.
- [39] J. Devlin, M.-W. Chang *et al.*, “Bert: Pre-training of deep bidirectional transformers for language understanding,” *arXiv preprint arXiv:1810.04805*, 2018.
- [40] C. D. Manning, M. Surdeanu *et al.*, “The stanford corenlp natural language processing toolkit,” in *ACL*, 2014, pp. 55–60.
- [41] D. Saito, K. Sasabuchi *et al.*, “Task-grasping from human demonstration,” *IEEE Humanoids (accepted)*, 2022.
- [42] N. Wake, D. Saito *et al.*, “Object affordance as a guide for grasp-type recognition,” *arXiv preprint arXiv:2103.00268*, 2021.
- [43] N. Wake, K. Sasabuchi, and K. Ikeuchi, “Grasp-type recognition leveraging object affordance,” *HOB1-RO-MAN Workshop*, 2020.
- [44] K. Sasabuchi, N. Wake, and K. Ikeuchi, “Task-oriented motion mapping on robots of various configuration using body role division,” *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 413–420, 2020.
- [45] Google, “Google/mediapipe.” [Online]. Available: <https://github.com/google/mediapipe>
- [46] J. Takamatsu, K. Sasabuchi *et al.*, “Learning-from-observation system considering hardware-level reusability,” *arXiv preprint arXiv:2212.09242*, 2022.
- [47] K. Sasabuchi, D. Saito *et al.*, “Task-sequencing simulator: Integrated machine learning to execution simulation for robot manipulation,” *arXiv preprint arXiv:2301.01382*, 2023.
- [48] “A next-generation robot able to work together with people,” <http://nextage.kawada.jp/en/>, (accessed Jan. 5, 2023).
- [49] “Dexterous hand series,” <https://www.shadowrobot.com/dexterous-hand-series/>, (accessed Jan. 5, 2023).
- [50] K. Rokis and M. Kirikova, “Challenges of low-code/no-code software development: A literature review,” in *International Conference on Business Informatics Research*. Springer, 2022, pp. 3–17.
- [51] A. Sahay, A. Indamutsa *et al.*, “Supporting the understanding and comparison of low-code development platforms,” in *SEAA*. IEEE, 2020, pp. 171–178.
- [52] K. Ikeuchi, J. Takamatsu *et al.*, “Applying learning-from-observation to household service robots: three common-sense formulation,” *arXiv preprint arXiv:2304.09966*, 2023.