Available online at www.sciencedirect.com

ScienceDirect

Procedia CIRP 107 (2022) 914–919

www.elsevier.com/locate/procedia

55th CIRP Conference on Manufacturing Systems

System of Robot Learning from Multi-Modal Demonstration and Natural Language Instruction

Shuang Lu*[a], Julia Berger[a], Johannes Schilp[a,b]

[a]*Fraunhofer IGCV, Am Technologliezentrum 10, 86159 Augsburg, Germany*

[b]*Chair of Digital Manufacturing, Augsburg University, Am Technologiezentrum 8, 86159 Augsburg, Germany*

*Corresponding author. Tel.: +49-821-90678-322 ; fax: +49-821-90678-199. *E-mail address:* shuang.lu@igcv.fraunhofer.de

Abstract

Collaborative robots are set to play an important role in the future of the manufacturing industry. They need to be able to work outside of the fencing and perform new tasks to individual customer specifications. The necessity of frequent robot re-programming is a great challenge for small and medium sized companies alike. Learning from demonstration is a promising approach that aims to enable robots to acquire from their end users new task knowledge consisting of a sequence of actions, the associated skills, and the context in which the task is executed. Current systems have limited support for integrating semantics and environmental changes. This paper introduces a system combining several modalities as demonstration interfaces, including natural language instruction, visual observation and hand-guiding, which enables the robot to learn a task comprising a goal concept, a plan and basic actions, with consideration for the current environment state. The task thus learned can then be generalized to similar tasks involving different initial and goal states.

© 2022 The Authors. Published by Elsevier B.V.
This is an open access article under the CC BY-NC-ND license (https://creativecommons.org/licenses/by-nc-nd/4.0)
Peer-review under responsibility of the International Programme committee of the 55th CIRP Conference on Manufacturing Systems

Keywords: Human robot collaboration; Cognitive robotics; Sensor

1. Introduction

For several decades, a key focus of research has been on increasing production flexibility in manufacturing systems. The ability to deal with uncertainties has become increasingly relevant against the background of the global pandemic. A collaborative robot (cobot) is able to work hand-in-hand with human workers in performing a variety of tasks, particularly in assembly, logistics and maintenance. Cobots are very useful in managing worker shortages and minimizing potential business losses resulting from unmet orders [1]. In this context, robot tasks have to be re-programmed each time a new request is received from the factory. Robot programming consists of two hierarchical steps: the definition of a task plan by a sequence of robot actions, followed by the writing of a program for each action in the robot controller. The high cost of such complex programming procedures is a fundamental barrier to the deployment of cobots. Much research has been devoted to reducing the complexity of robot programming, e.g. an assembly task plan can be generated directly from CAD files [2]. However, a task plan is not normally available with ad-hoc assistive tasks, such as ”Put the tool on the table”. In the course of action programming, a pre-defined parameterized trajectory, also known as a skill, is defined by the user interface and then translated into robot specific syntax in the controller [3]. The parameters are primarily robot positions and orientations.

User friendly programming interfaces are becoming increasingly available on the market, examples including the touchscreen tablet by Universal Robots and the tablet in the Fanuc CRX series, in which motion sequences can be defined by drag & drop. Hand-guiding is another central function of cobots, which allows the user to rapidly and intuitively interact with the robot and program it [4]. To explore user experiences with modern robot programming interfaces, a user study was conducted by Ajaykumar and Huang [5]. Participants were asked to program a practice pick and place task using a hand-guiding function and a UR-5 teach pendant device. The result shows that participants had a poor mental model of their programs, especially those that involved way points.

Multi-modal perception plays a key role in enabling robots to learn and execute tasks in unstructured environments [6]. Various user interfaces have been developed in research for

2212-8271 © 2022 The Authors. Published by Elsevier B.V.
This is an open access article under the CC BY-NC-ND license (https://creativecommons.org/licenses/by-nc-nd/4.0)
Peer-review under responsibility of the International Programme committee of the 55th CIRP Conference on Manufacturing Systems
10.1016/j.procir.2022.05.084

This is a resupply of March 2023 as the template used in the publication of the original article contained errors. The content of the article has remained unaffected.

robot operating and programming [7], involving eye gaze, hand gesture and voice control. Effective analysis of the scene, for both path planning and observing the behavior of humans in the robot workspace can be performed using computer vision technologies based on a three dimensional (3D) camera [6]. Natural language can be used for manipulation task instructions [8]. Sensors and algorithms enable robots to understand and interact with an unstructured environment.

In the context of robotics and automation, learning from demonstration (LfD)) is a promising paradigm for enabling intuitive programming by non-expert users. Not only can LfD be applied to human motor skills [9], but it is also relevant for learning task plans. Ekvall and Kragic [10] developed a task-level planning framework from demonstration, in which a task plan is learned by identifying a primitive sequence of actions. Kinesthetic teaching via a hand-guiding interface has been widely investigated for the purpose of learning human motor skills such as striking motions in robot table tennis [11]. Video demonstration has recently been increasingly used for learning geometric relations between objects and environment [12].

The aim of this work is to develop an LfD system for cobots with cognitive skills, in which the robot is able to learn and reproduce a task in an unstructured and dynamic environment. Existing LfD systems are not so focused on how to select interfaces for LfD systems. The prudent integration of interfaces enables the robot to learn the concept of a task program and generalize it intelligently and autonomously to other similar tasks. Thus, the complexity of robot programming is able to be reduced. This system enables the user to program the robot through integrated demonstration and instruction of a task plan, goal concept and basic actions. The robot is then able to understand the semantics of the task goal and apply this knowledge to similar tasks.

This work considers the following questions:

- What types of interfaces are suitable for acquiring the task goal, plan and actions?
- What characterizes the system architecture?
- How can the learned tasks be represented?

2. Background and Related Work

Classical robot programming can be divided into offline (OFP) and online approaches. In OFP the task program is defined on devices separated from the robot. This work focuses on online approaches, as they integrate the environment model in the robot task program. LfD is an online approach in which the task program is created by non-expert users directly within the robot working environment. Task-oriented programming (TOP) is another intuitive programming method, in which the user defines the task program on an abstract level [13]. By employing TOP, the user can program the robot without defining linear or point-to-point motions for different types of industrial robots. A hybrid TOP system was devised by Berg and Reinhart [2], in which the assembly sequence is generated offline, skills are generated online with object poses, and an online module enables the actions of human workers to be recognized for adaptive collaborative assembly. LfD systems not only achieve the same level of abstraction as TOP but they also improve the degree of intuitiveness. Moreover, they are more applicable to ad-hoc assistive tasks, in which the primitive action sequences are generated online in accordance with the human workers' requirements.

There is plenty of literature available on trajectory and plan learning in LfD, focusing generally on developing models for representing trajectories or a task plan. Key contributions in this area are summarized in Sections 2.1 and 2.2. For goal-directed robot learning, the sensory interfaces for goal representation are an essential component of an LfD system, and Section 2.3 outlines a number of representative works. As for perception, state-of-the art algorithms in computer vision and natural language understanding relevant to LfD are covered in Sections 2.4 and 2.5. Finally, works presenting multi-modal approaches are discussed in Section 2.6.

2.1. Trajectory Learning

Kinesthetic teaching is an LfD method in which a human physically guides a robot to perform a certain skill [14]. Human motor skills, such as striking motions in table tennis, can be learned from 25 recorded motions [11]. The basic movements, involved in an industrial assembly task, like picking and placing, can also be learned by kinesthetic teaching [15]. One popular way of representing motor skills is to use dynamic movement primitives (DMPs), which enable a robot to learn a stable dynamic system from a single demonstration represented by a non-linear differential equation [16]. Once learned, DMPs are capable of avoiding obstacles attaining the goal position. From multiple demonstrations Gaussian mixture model and Gaussian mixture regression (GMM-GMR) are used to extract the optimal trajectory [17]. Hidden Markov Models (HMMs) can also be used to represent a skill [18]. However, they do not consider collision avoidance.

2.2. Integrated Robot Learning of Task Plan and Actions

Task planning is the problem of finding a sequence of actions for achieving a desired goal state [10]. Motion segmentation plays an essential role in task plan learning. Ding et al. developed a learning strategy for assembly tasks [19], in which a motion is segmented based on human hand centroid velocities. A markerless vision capture system based on Kinect is used to acquire continuous human hand movements. Kyrarini et al. developed an LfD system consisting of two learning modules: high-level task learning and low-level skill learning. In high-level task learning, the task is segmented into actions, i.e. it is first split into subtasks and each subtask then divided into actions; the objects involved are then assigned specific IDs. The actions are defined as the sequence: "start arm moving", "grasp object", "release object" and "stop arm moving". Kinect is used for object detection and recognition based on point cloud processing [15].

This is a resupply of March 2023 as the template used in the publication of the original article contained errors. The content of the article has remained unaffected.

2.3. Task Goal Learning

The goal of a task is to achieve a certain setting in the environment. Different goal representations result in different ways of deriving a task plan for achieving a goal [20]. Ekvall and Kragic represent the goal state as the pose of task relevant objects from visual input [10]. They assume that all objects are obsolete from each other and that there is no other object in the scene. The task goal learned cannot be generalized to new tasks in a cluttered scene. Furthermore, the representation is not applicable to real-world production scenarios, in which the objects to be assembled have more complex spatial relationships. Agkun and Thomaz used HMMs with a single object of attention to represent the goal [21]. Scene graph structure is used in [20] to represent the goal state. An R-CNN object detector is first applied to grocery objects. A scene graph structure is then generated, based on a depth estimation. The spatial relations between object pairs like *in* and *out* can then be estimated.

2.4. Visual Perception

Ding et al. [19] used a 3D camera for human movement tracking and object recognition. Thanks to the availability of datasets in research, they were able to train various neural network architectures with RGB images. The dataset from ImageNet Challenge contains more than 1 million images with 1000 object classes. The annotations fall into two categories: (1) image-level annotation of a binary label denoting the presence or absence of an object class in the image, e.g., "there are cars" in this image" but "there are no tigers," (2) object-level annotation of a tight bounding box and class label around an object instance in the image, e.g., "there is a screwdriver centered at position (20, 25) with a width of 50 pixels and a height of 30 pixels" [22]. Ren et al. proposed a Faster R-CNN algorithm for object detection with region proposal networks [23]. Unless they are everyday objects, the manufactured products tend to have less distinctive textural features. Understanding a visual scene is more than just object recognition. Spatial and temporal relations between objects are also essential to robot learning. Scene graphs can be used to model the relations from an image, in which an object hierarchy is defined with part-of relation. The Visual Genome dataset is widely used to connect structured image concepts to natural language. It comprises over 108k images where each image as an average of 35 objects, 26 attributes and 21 pairwise relationships between objects (e.g. tree near the water) [24]. Deep-learning approaches can then be applied to train a scene graph generator [25].

A hand model with keypoints is defined to enable the detection of hands from images. A large dataset with annotation is then used to train a detector using a deep-learning approach. Lugaresi et al. [26] from Google developed a framework that infers 21 3D landmarks of a hand from a single RGB frame.

2.5. Natural Language Instruction

People use verbal cues to communicate intent to other people. It is therefore intuitive and effective for human workers to instruct a robot task using simple sentences, such as "bring me a screwdriver". A dataset of natural language instructions for object reference in robot manipulation scenarios is introduced in [8]. A total of 1582 individual written instructions were collected by way of online crowdsourcing.

In natural language understanding (NLU), intent classification is the task of correctly labeling a natural language utterance from a pre-defined set of intents. There are several publicly available NLU toolkits for building conversational agents, namely Watson, Dialogflow, LUIS and Rasa [27], which enable the definition of intents by simple text input and the training of customized classifiers. Possible intents are weather, play or alarm in our daily lives. Semantic role labeling (SRL) has been studied for decades in natural language processing (NLP). As defined in [28], the aim of SRL is to answer the questions of "Who did What do Whom and How, When and Where?" in text. Most SRL research is based on an approach requiring training on role-annotated data [29] that can be used for information extraction.

2.6. Multi-Modal Approaches

"Multi-modal" means that more than one demonstration interface is applied to the LfD system. Kartmann et al. developed a representation which allows a robot to manipulate a scene based on verbal commands by specifying spatial object relations [30]. Existing works show that it is feasible and beneficial to build an LfD system based on demonstrations and verbal commands. However, the question of how to build a system of task learning from multi-modal demonstration and natural language instruction has not yet been systematically explored.

3. System Design and overview

The focus of this work is on a pick & place task. For the sake of clarity, the system design will be explained on the basis of an example task "Pick up all objects in the gray box and place them in the blue box" illustrated in Fig. 1. A human demonstrator firstly show the robot how to perform the required task. The system is capable of recording, interpreting and transferring the information to a robot. The learned task is then incorporated as robot knowledge. It consists of three parts: a goal concept for achieving the goal state from the initial environment state; a high-level symbolic action sequences based on the goal concept; and low-level trajectory for each basic action. The hierarchical structure of the task is shown in Fig. 2. Finally the task is segmented into basic actions, as proposed by Berg [31]: reach, grasp, move, position and release. Robot motions are then generated based on learning from basic actions. Details of the system elements and architecture are outlined in the following sections.

3.1. System architecture

To reproduce the task, the robot has to learn each basic action, but also the environment state during the demonstration.

This is a resupply of March 2023 as the template used in the publication of the original article contained errors. The content of the article has remained unaffected.

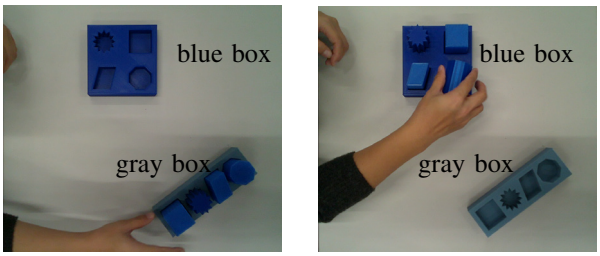

Fig. 1. (a) Initial Scene; (b) Goal Scene

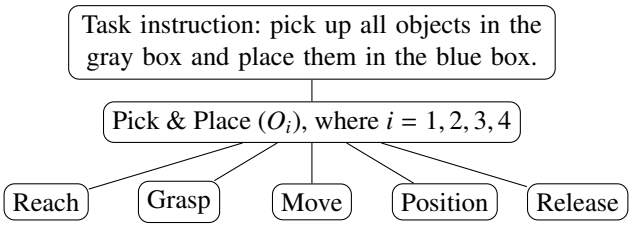

Fig. 2. Hierarchical task structure

A comprehensive system is thus developed as shown in Fig. 3. The system comprises three main elements: 1) the human robot interfaces: 2) the software modules for perception, reasoning and action generation; and 3) the robot. The system design process consists of the following steps:

- Defining the functions of the software modules by answering the question: What does the robot need to understand from the human demonstration in order to 1) reproduce the task and 2) execute the task in changed environment settings.
- Identifying suitable interfaces and inference methods to realize each functions in the first step.
- Developing the system architecture to enable the interaction of various elements and modules.

The software module consists of three sub-modules: the concept manager (CM), environment manager (EM) and robot manager (RM). The CM is responsible for acquiring semantics from the task goal. The result is given to the EM for semantic grounding with initial and goal scene understanding. Robot motion learning and execution is handled in the RM. The following types of human robot interface are considered: hand-guiding, a graphical user interface (GUI), speech and a 3D camera. The functions of each sub-module and the attendant interfaces are discussed in the following subsections.

3.2. Concept manager

The CM is responsible for processing the direct input from a human demonstrator. The information is acquired either from a speech or text input from a graphical user interface (GUI). Speech input necessitates an additional speech-to-text function. The CM has four further functions: intent classifier, action parser, object parser and location parser.

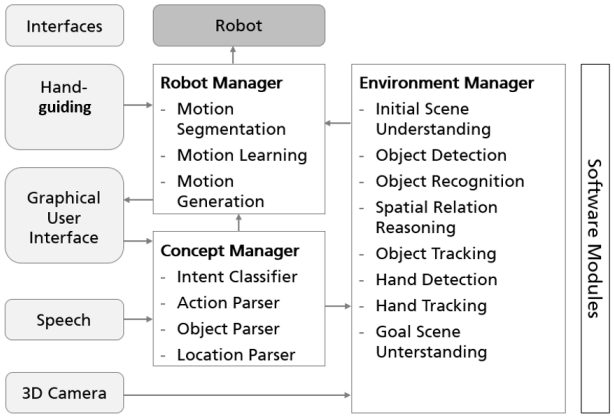

Fig. 3. System architecture

The intent classifier is predefined with three intents: instruction, tool control and status control. The status has four possible states: start, pause, continue and finish. Commands for robot tools are processed in the tool controller. Table 1 presents a summary of all possible intents. If the input intent is a task instruction, information will be further extracted by an action parser, object parser and location parser using the SRL method.

Table 1. List of intents.

Intent	Natural Language text	Interface
Task instruction	Pick up all the objects in the gray box	Speech or GUI
Tool control	Close/open gripper	Speech
Status control	Start, pause, continue, finish	Speech

As described in [8], a robot task instruction consists of three parts: an action verb, the object related to the action, and its location, e.g. ”Pick up the yellow cube on the table.” The task relevant information can be extracted using the SRL technique introduced in Section 2.5. The resulting labeling of the example sentence is shown in Fig. 4. It is generated using the web based tool devised by the University of Stuttgart[1]. Further labels relating to SRL can be found in [32].

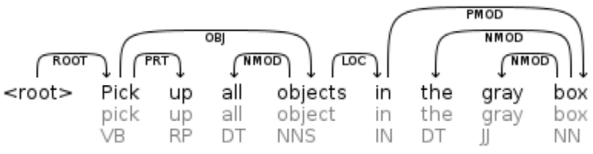

Fig. 4. Semantic Role Labeling

Descriptions of all labels and related parser functions are listed in Table 2. The information extracted consists of: 1) ”pick up” from the action parser; 2) ”all objects” from the object parser; 3) ”in the gray box” from the location parser. The information is transferred to the EM to improve the scene un-

[1] http://en.sempar.ims.uni-stuttgart.de/

This is a resupply of March 2023 as the template used in the publication of the original article contained errors. The content of the article has remained unaffected.

Table 2. List of semantic labels related to parser functions.

Label	Description	Parser Functions
ROOT	Root	
PRT	Between verb and particle	Action parser
OBJ	Object	Object parser
NMOD	Modifier of nominal	Object parser
PMOD	Modifier of preposition	Location parser
LOC	Locative adverbial or nominal modifier	Location parser

derstanding. In general, the CM handles all the discrete input events from humans during task demonstrations.

3.3. Environment Manager

As shown in Fig. 5, a 3D camera is installed on the robot. The EM receives the information from the object parser and location parser in the CM, which initiates object detection and object recognition. In initial and goal scene understanding, the scene graph structure of the task in Fig. 1 is generated as shown in Fig. 6.

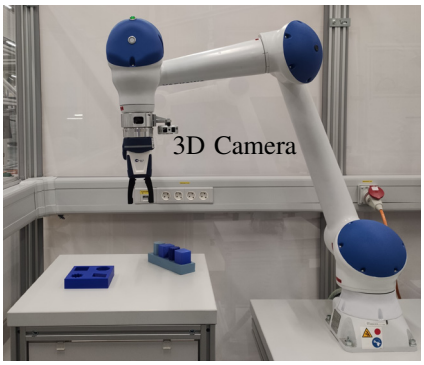

Fig. 5. Demonstration setup

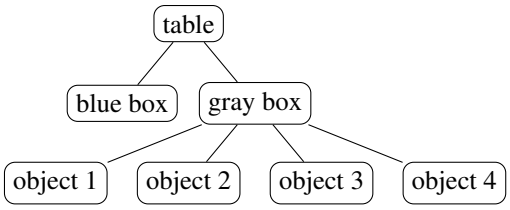

Fig. 6. Initial scene graph structure

Together with the initial scene graph, all poses of objects are identified in the robot base frame. Each object has its attributes, such as appearance and poses. The EM uses CAD model based object recognition [33]. Spatial relation reasoning is realized by calculating poses from different objects in a single coordinate representation [20]. Hand detection and tracking are triggered by a start signal from the CM. Object detection in the initial scene results in a unique ID being created for each object. Each object is tracked with its individual ID in a video. Similarly, human hands are detected and tracked in each frame. The hand keypoints are detected from RGB images by the MediaPipe framework [26] for each frame. In the next step, the depth value corresponding to each keypoint is estimated from the depth frame. Hand trajectories are then generated from each frame with respect to time. In the end, the goal scene graph structure are grounded with goal concept from the CM. The symbolic semantics learned are stored as robot knowledge for future use.

3.4. Robot Manager

As shown in Fig. 7, motion segmentation consists of two steps. First, the recorded hand and object trajectories are segmented into pick & place movements relating to different objects. The number of segments is the same as the number of objects. They are segmented by the hand and object trajectories from each other, i.e. to find out which object was moving with hands together by calculating the distances. The release command is also used here to improve the robustness. In the second step, each pick & place movement is segmented into five basic actions. If the learned sequence needs to be modified, the basic action can be updated with a symbol as an identifier. Renewed demonstration for the entire task is not necessary. Grasp and release are two movements with very short duration. It is very difficult to distinguish them from recorded hand trajectories. In order to improve the accuracy, the command input from the CM is integrated into the segmentation process. Further segmentation can be realized by the contact state of the hand and object. The reach movement has no contact to the object in the scene. Move and position can be segmented by the velocity of hand trajectories, where position is slower than move action.

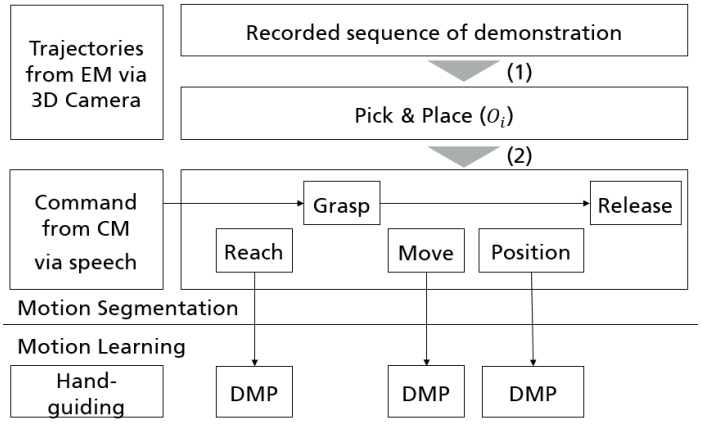

Fig. 7. Motion segmentation and learning. (1) Segmentation based on hand and object trajectories & release command. (2) Segmentation based on voice commands & hand moving velocities.

After learning the symbol sequences, the basic action is learned by kinesthetic teaching with hand-guiding. DMPs developed by Ijspeert et al. [34] are used to learn each basic action from a single reference sample.

In motion generation, the robot can generate the action by defining initial and goal positions for a learned DMP. It is able to avoid an obstacle while moving towards the goal position.

This is a resupply of March 2023 as the template used in the publication of the original article contained errors. The content of the article has remained unaffected.

4. Conclusion and future work

With the aim of reducing the complexity of robot programming, this work introduces an LfD system with which a robot is able to learn a task goal concept, a plan and basic actions. The robot possesses cognitive skills such as sensing, reasoning and learning. These are realized by integrating multi-modal demonstration interfaces such as hand-guiding, a 3D camera and a GUI. Meanwhile, the user can communicate with the robot using natural language instructions. As future work, the system will be implemented to validate the design concept.

Acknowledgements

This research was conducted within the project MeMoRob (AKZ: DIK0358/01) funded by the Bavarian Ministry of Economic Affairs, Regional Development and Energy (StMWi). The authors would like to thank the StMWi for their financial support.

References

[1] Agrawal, M., Eloot, K., Mancini, M. and Patel, A., 2020. Industry 4.0: Reimagining manufacturing operations after COVID-19. [online] McKinsey & Company. Available at: <https://www.mckinsey.com/business-functions/operations/our-insights/industry-40-reimagining-manufacturing-operations-after-covid-19> [Accessed July 29,2020].

[2] Berg, J., and Reinhart G., 2017. An integrated planning and programming system for human-robot-cooperation. Procedia CIRP 63 (2017): 95-100.

[3] Lambrecht, J., Kleinsorge, M., Rosenstrauch, M., & Krüger, J., 2013. Spatial programming for industrial robots through task demonstration. International Journal of Advanced Robotic Systems, 10(5), 254.

[4] Safeea, M., Bearee, R., & Neto, P., 2017. End-effector precise hand-guiding for collaborative robots. In Iberian Robotics conference (pp. 595-605). Springer, Cham.

[5] Ajaykumar, G., & Huang, C. M., 2020. User needs and design opportunities in end-user robot programming. In Companion of the 2020 ACM/IEEE International Conference on Human-Robot Interaction (pp. 93-95).

[6] Falco, P., Lu, S., Cirillo, A., Natale, C., Pirozzi, S., & Lee, D., 2017. Cross-modal visuo-tactile object recognition using robotic active exploration. In 2017 IEEE International Conference on Robotics and Automation (ICRA) (pp. 5273-5280). IEEE.

[7] Berg, J., Lu, S., 2020. Review of Interfaces for Industrial Human-Robot Interaction. Curr Robot Rep 1, 27–34.

[8] Scalise, R., Li, S., Admoni, H., Rosenthal, S., Srinivasa, S. S., 2018. Natural language instructions for human–robot collaborative manipulation. The International Journal of Robotics Research, 37(6), 558-565.

[9] Ravichandar, H., S. Athanasios, Polydoros, S. Chernova, Aude and Billard. "Robot Learning from Demonstration: A Review of Recent Advances." (2019).

[10] Ekvall, S., & Kragic, D., 2008. Robot Learning from Demonstration: A Task-level Planning Approach. International Journal of Advanced Robotic Systems.

[11] Muelling, K., Kober, J., Kroemer, O., & Peters, J. (2012). Learning to select and generalize striking movements in robot table tennis. The International Journal of Robotics Research, 32, 263 - 279.

[12] Jin, J., Petrich, L., Dehghan, M., & Jagersand, M. (2020). A geometric perspective on visual imitation learning. In 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (pp. 5194-5200). IEEE.

[13] Kugelmann, D., 1999. Aufgabenorientierte Offline-Programmierung von Industrierobotern

[14] B. Akgun, M. Cakmak, J. W. Yoo and A. L. Thomaz, "Trajectories and keyframes for kinesthetic teaching: A human-robot interaction perspective," 2012 7th ACM/IEEE International Conference on Human-Robot Interaction (HRI), 2012, pp. 391-398.

[15] Kyrarini, M., Haseeb, M.A., Ristić-Durrant, D. et al. Robot learning of industrial assembly task via human demonstrations. Autonomous Robots 43, 239–257 (2019).

[16] Schaal, S., Peters, J., Nakanishi, J., & Ijspeert, A. (2003, October). Control, planning, learning, and imitation with dynamic movement primitives. In Workshop on Bilateral Paradigms on Humans and Humanoids: IEEE International Conference on Intelligent Robots and Systems (IROS 2003) (pp. 1-21).

[17] S. Calinon, F. Guenter and A. Billard, 2007. On Learning, Representing, and Generalizing a Task in a Humanoid Robot in IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), vol. 37, no. 2, pp. 286-298, April 2007.

[18] S. Calinon, F. D'halluin, E. L. Sauser, D. G. Caldwell and A. G. Billard, 2010. Learning and Reproduction of Gestures by Imitation. in IEEE Robotics & Automation Magazine, vol. 17, no. 2, pp. 44-54, June 2010.

[19] Ding G, Liu Y, Zang X, Zhang X, Liu G, Zhao J.,2020. A Task-Learning Strategy for Robotic Assembly Tasks from Human Demonstrations. Sensors; 20(19):5505.

[20] Z. Zeng, Z. Zhou, Z. Sui and O. C. Jenkins, 2018. Semantic Robot Programming for Goal-Directed Manipulation in Cluttered Scenes. IEEE International Conference on Robotics and Automation (ICRA), 2018, pp. 7462-7469.

[21] Akgun, B., Thomaz, A., 2016. Simultaneously learning actions and goals from demonstration. Auton Robot 40, 211–227 (2016).

[22] Russakovsky, O., Deng, J., Su, H. et al., 2015. ImageNet Large Scale Visual Recognition Challenge. Int J Comput Vis 115, 211–252 (2015).

[23] Ren, S., He, K., Girshick, R., & Sun, J., 2016. Faster R-CNN: towards real-time object detection with region proposal networks. IEEE transactions on pattern analysis and machine intelligence, 39(6), 1137-1149.

[24] Krishna, R., Zhu, Y., Groth, O. et al., 2017. Visual Genome: Connecting Language and Vision Using Crowdsourced Dense Image Annotations. Int J Comput Vis 123, 32–73.

[25] Xu, D., Zhu, Y., Choy, C. B., & Fei-Fei, L., 2017. Scene graph generation by iterative message passing. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 5410-5419).

[26] Lugaresi, C., Tang, J., Nash, H., McClanahan, C., Uboweja, E., Hays, M., ... & Grundmann, M., 2019. Mediapipe: A framework for building perception pipelines. arXiv preprint arXiv:1906.08172.

[27] Liu, X., Eshghi, A., Swietojanski, P., & Rieser, V. ,2021. Benchmarking natural language understanding services for building conversational agents. In Increasing Naturalness and Flexibility in Spoken Dialogue Interaction (pp. 165-183). Springer, Singapore.

[28] Do, Q.N., Bethard, S., Moens, M., 2016. Facing the most difficult case of Semantic Role Labeling: A collaboration of word embeddings and co-training. COLING.

[29] Màrquez, L., Carreras, X., Litkowski, K. C., & Stevenson, S., 2008. Semantic role labeling: an introduction to the special issue.

[30] Kartmann, R., Zhou, Y., Liu, D., Paus, F., & Asfour, T., 2020. Representing Spatial Object Relations as Parametric Polar Distribution for Scene Manipulation Based on Verbal Commands. In 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (pp. 8373-8380). IEEE.

[31] Berg, J.,2020. System zur aufgabenorientierten Programmierung für die Mensch-Roboter-Kooperation

[32] Johansson, R., 2008. Dependency-based semantic analysis of natural-language text. Lund University.

[33] Ip, C. Y., & Gupta, S. K., 2007. Retrieving matching CAD models by using partial 3D point clouds. Computer-Aided Design and Applications, 4(5), 629-638.

[34] A. J. Ijspeert, J. Nakanishi, H. Hoffmann, P. Pastor and S. Schaal, 2013. Dynamical Movement Primitives: Learning Attractor Models for Motor Behaviors in Neural Computation, vol. 25, no. 2, pp. 328-373, Feb.

This is a resupply of March 2023 as the template used in the publication of the original article contained errors. The content of the article has remained unaffected.