

Article

Robot Cooking—Transferring Observations into a Planning Language: An Automated Approach in the Field of Cooking

Markus Schmitz ^{*}, Florian Menz , Ruben Grunau , Nils Mandischer , Mathias Hüsing 
and Burkhard Corves 

Institute of Mechanism Theory, Machine Dynamics and Robotics, RWTH Aachen University,
52062 Aachen, Germany; mandischer@igmr.rwth-aachen.de (N.M.); huesing@igmr.rwth-aachen.de (M.H.);
corves@igmr.rwth-aachen.de (B.C.)

* Correspondence: schmitzm@igmr.rwth-aachen.de

Abstract: The recognition of human activities from video sequences and their transformation into a machine-readable form is a challenging task, which is the subject of many studies. The goal of this project is to develop an automated method for analyzing, identifying and processing motion capture data into a planning language. This is performed in a cooking scenario by recording the pose of the acting hand. First, predefined side actions are detected in the dataset using classification. The remaining frames are then clustered into main actions. Using this information, the known initial positions and virtual object tracking, a machine-readable planning domain definition language (PDDL) is generated.

Keywords: human activity recognition; learning from demonstration; PDDL; robot cooking; task planning; motion tracking



Citation: Schmitz, M.; Menz, F.; Grunau, R.; Mandischer, N.; Hüsing, M.; Corves, B. Robot Cooking—Transferring Observations into a Planning Language: An Automated Approach in the Field of Cooking. *Eng* **2023**, *4*, 2514–2524. <https://doi.org/10.3390/eng4040143>

Academic Editor: Antonio Gil Bravo

Received: 3 August 2023

Revised: 21 September 2023

Accepted: 22 September 2023

Published: 7 October 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The automation of cooking processes is currently still subject to some limitations. A vision of extracting recipe and process data using only tracked motion data, making this information available to the robot in machine-readable form, and thus enabling automated cooking by a robot or even in a human-robot collaboration, is presented in this paper. In a cooperation between human and machine, it is essential that the machine can correctly interpret and categorize the actions of the human. This is especially relevant in a situation where the human is supposed to teach the machine specific work processes. In passive observation, a method of learning from demonstration (LfD), the machine learns only by observing the human performing a task [1]. One important aspect of this method is to enable non-experts to teach robots or automated systems specific work processes. Traditionally, robot programming has been a complex and technical task, requiring specialized knowledge. However, with LfD approaches, even individuals without programming expertise can instruct machines effectively [2].

The increasing demand for automation in cooking processes is evident both in the rising sales of multipurpose kitchen appliances (e.g., Vorwerk's Thermomix [3]) and in the expected growth of the global market for smart kitchens [4]. Although multipurpose kitchen appliances offer some degree of automation, there is still significant room for improvement. Integrating robot arms into the cooking process could enhance the level of automation and streamline various tasks. While there have been some concepts and prototypes exploring the use of robot arms in cooking tasks [5,6], none have yet made it to the mainstream commercial market. This indicates that there are still challenges and complexities, such as the high price and the necessity of preprocessing ingredients, to overcome before such systems become widely available and accessible to the general public [7].

Several studies show that, in principle, robotic hardware is already capable of providing all the necessary functions for household and cooking processes [8,9]. The more challenging part is designing and enabling the grippers to grasp tools and selecting ingredients of the cooking process of varying texture, shape and solidity [10,11]. Another important aspect is safety when using robots in open workspaces in the household environment. Dangers arise particularly from collisions between humans and robots [12] and especially when tools such as knives are used in the context of the cooking scenario [13]. In addition to safety, ethical considerations such as acceptance must also be taken into account. From today's point of view, people would rather let a robot serve food or do the cleaning than let it cook food for them [14]. Cognition has an important role in the field of cooking processes and includes the recognition of cooking states, ingredients, and the composition of dishes obtained from motion data [15]. However, the use of sensing technology is expensive and the development of robust algorithms is challenging to realize the recognition of cooking states, ingredients, and the overall composition of cooked dishes, as well as the learning of subjective taste preferences [16].

Motion planning is a challenging task, since collision-free motion has to be achieved in a complex 3D environment, while also taking into account required contact with objects, including controlled intrusion (e.g., when cutting) [13,17]. In addition to motion planning, there must be efficient task planning that can divide the given recipe into individual tasks and orchestrate their execution [18,19]. Ideally, the tasks can be divided between different actors, such as multiple robots and/or humans, to prepare a tasty dish together [8].

In this context, the primary objective of this paper is to observe and analyze human activities in the domain of cooking processes. To achieve this goal, an automated method is developed to enable LfD through passive observation of human hand posture. By combining this information with the known objects and their positions, a machine-readable planning domain definition language (PDDL) is created. PDDL is a formal language, commonly used in the field of artificial intelligence and robotics, for representing and describing planning problems and actions [20]. It provides a structured and unambiguous way to model the actions, consisting of preconditions and effects, involved in a task, making it an ideal choice for translating human actions into machine-readable instructions. The PDDL representation will facilitate the machine's understanding and interpretation of human actions, thereby allowing it to effectively learn and perform cooking tasks and even collaborate with a human actor.

2. Concept

The challenge to be solved in cooking scenarios is that the actions and their effects and preconditions are largely unknown and vary greatly depending on the recipe. First, a cooking process is recorded. This is carried out using passive markers and a motion capture system. The question arises of how the recorded motion data can be analyzed. Motion data contains the position and orientation of the hand for each frame. These can naturally be supplemented with descriptive statistics, describing, for instance, positions, orientations, velocities, and accelerations within a time window. The basic idea and challenge for this research is exemplified in Figure 1: The progression from the recording of the cooking process to the execution of the resulting plan by robots.

To create this plan, the executed actions have to be identified from the motion data. A distinction is made between two different types of actions: "processing" and "moving/changing position". While there can be an infinite number and variety of actions related to processing, there are only a few actions (e.g., pick, move, place) that change an object's position. It is also essential to closely examine the effects and preconditions of each action type. The processing actions can have very different effects and variations, whereas the change in position is the central effect of the "moving/changing position" action. Conversely, it is necessary to uniquely identify position-changing effects as they contribute to causality and enable process interpretation based on motion data. In this

way, the emerging plan can subsequently be decomposed into a PDDL (planning domain definition language).

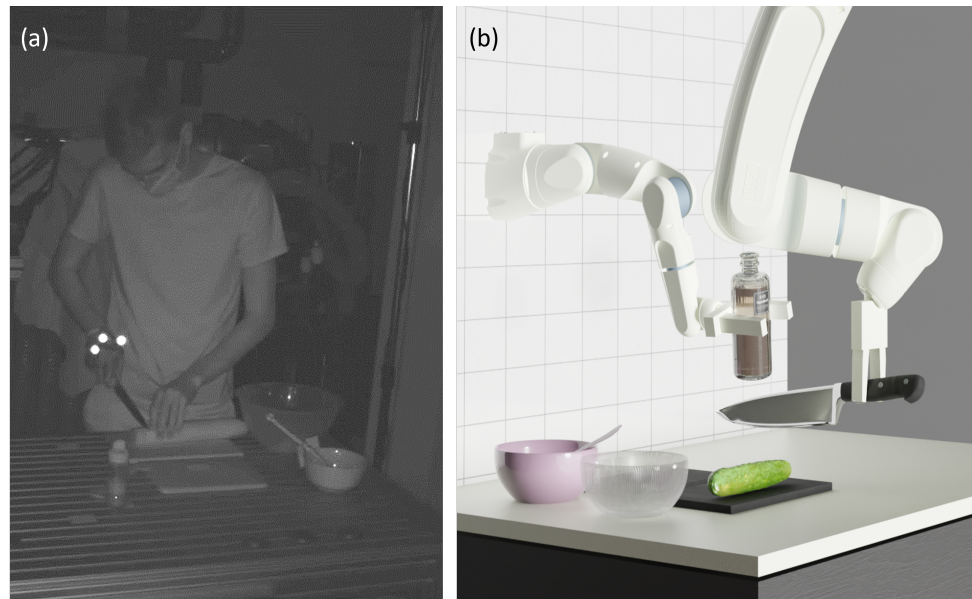


Figure 1. (a) The recording of the cooking process for this project; (b) a simplified conceptual rendering of the execution of the resulting plan.

These requirements align with the selection criteria of supervised and unsupervised learning. Classification is suitable when there is a finite number of classes. However, this requires training data that behave similarly for different individuals performing actions like pick, place, and move, for example. For the case of an unknown number of actions and a still unknown structure, clustering is a method of unsupervised learning to discover data patterns and correlations in the motion data.

After the motion data has been analyzed, it has to be transferred into a PDDL. In the context of our study, the machine-readable PDDL generated from the observed human activities and their combination with known positions of objects will serve as the bridge between the human demonstrator and the robot or automated system. The PDDL representation allows the machine to interpret and understand the sequence of actions performed by the human during the cooking process. By leveraging this formal representation, the machine can efficiently plan and execute the same task or collaborate with the human actor seamlessly. An overview of the complete process, including recording, classification, clustering, and transfer into PDDL, which are subject of the research in this paper, is provided by Figure 2. The goal of our approach is to provide a general solution, independent of the specific actors involved in the execution of the resulting plan. Therefore, the execution aspect is excluded from the scope of this paper.

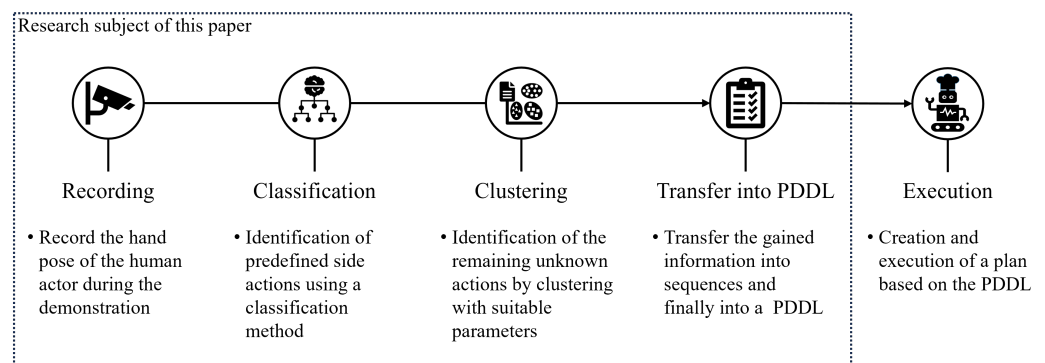


Figure 2. Schematic representation of the concept.

3. Recording

First, the cooking process is recorded by a motion capture system consisting of seven PrimeX13 cameras, sourced from OptiTrack (Corvallis, OR, USA). The schematic representation of the workspace is given in Figure 3. During the recording, the cook wears a glove with three passive markers placed on the back of the hand in a defined fixed position relative to each other. The position of each marker is determined by triangulation. These three points provide enough information to derive a pose. Specifically, the back of the hand is considered as a rigid body with 6 degrees of freedom. The recording was captured at 120 frames per second.

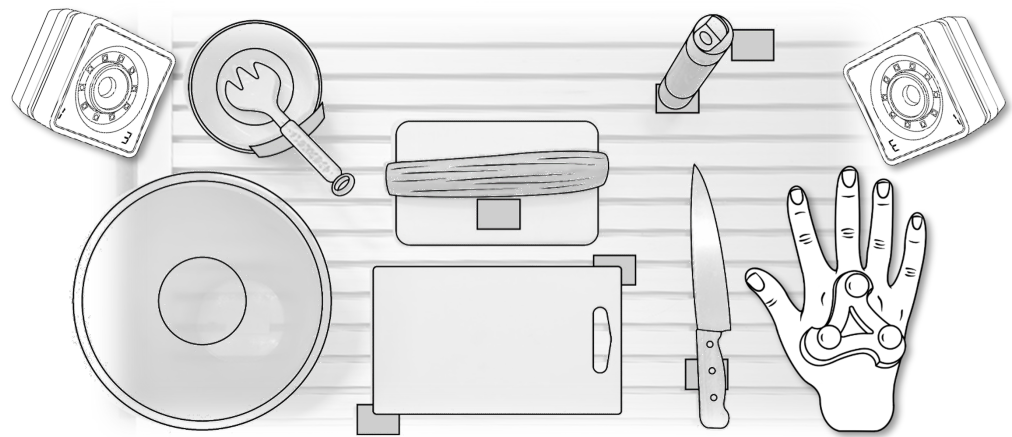


Figure 3. Schematic view of the workspace. Cameras (**top left and right**), hand with passive markers (**bottom right**) and used tools and ingredients.

In addition to the continuous recording of the trajectory, all objects in the workspace are identified once before the cooking process, as well as their initial positions. In the productive scenario, this is to be automated by an additional camera and image recognition. As previously mentioned, all recorded data are basically poses with time stamps. Other variables such as velocity, acceleration and angle in relation to the tabletop were derived from this. A look into this raw data already reveals some characteristics which are to be worked out as illustrated in Figure 4.

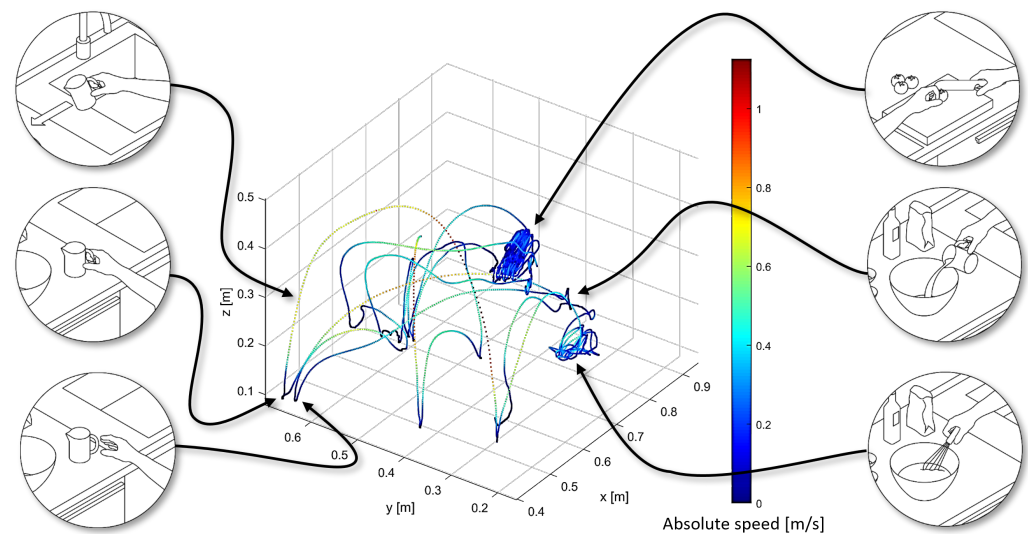


Figure 4. Illustration of the absolute speed within the space using Matlab R2021b.

4. Classification

When analyzing this cooking process, it is noticeable that some short actions occur repeatedly, which are called side actions in the following. These actions are part of the distinct actions “moving/changing position” that are important to track the positions of objects, as presented in the concept section. These are:

- Pick
- Move
- Place

Assuming that these actions are very consistent in their execution and occur in every cooking process, a classifier is trained to find these actions in the dataset. This ensures the generality of the procedure, since no main actions are anticipated, which are very individual. To train the classifier, a separate dataset is first recorded. In the training dataset, an action is assigned manually to each frame, with a numerical value assigned to it. There is a wide range of classification methods, from simple decision trees to weighted decision forests and neural networks. For our purposes, the RUS Boosted Trees method was chosen based on some preliminary research. RUS Boosted Trees is an algorithm that combines random under-sampling (RUS) with Boosted Trees to address the issue of imbalanced datasets. By applying RUS, the algorithm removes samples from majority classes randomly, thereby achieving a balanced class distribution. This approach improves the training process, mitigating runtime and storage problems [21].

A large number of parameters are available for the classification. Since these are partially dependent on each other and under the assumption that more parameters are not equivalent to a better classification, a parameter analysis is performed. With the classification method and parameters now selected, the classifier is trained and applied to the original dataset. Assuming that an action takes significantly longer than a single frame (recording at 120 frames per second), the classification results are smoothed. This is achieved by performing a check for each frame using a sliding window. Within this window, the number of times each action occurs is counted. Then the most frequent action within the window is selected for this frame. Classification provides a basic structure to the dataset, allowing the analysis of the remaining actions to continue with reduced complexity.

5. Clustering

The remaining frames are then to be categorized. This is the aim of the clustering. In this step, only those frames are considered which have not already been assigned by the classification, as shown in Figure 5.

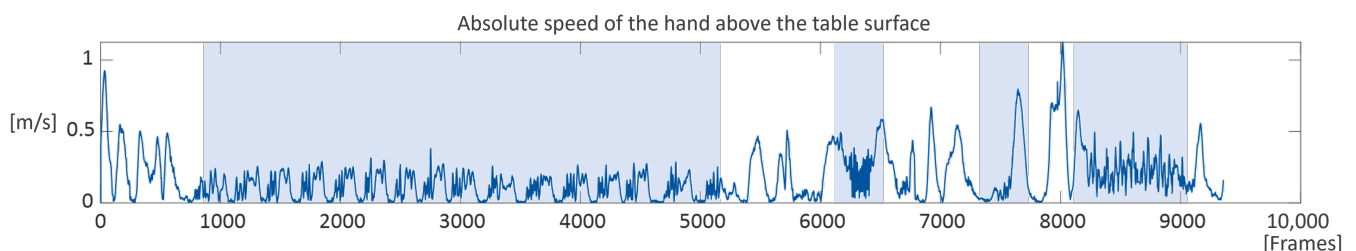


Figure 5. Graph of the absolute speed of the hand above the table surface within the cooking process. The frames with a white background have already been identified by the classification.

In contrast to classification, there is no search for a defined number of predefined classes. This approach makes it possible to analyze the cooking process despite a high degree of variation in the execution. Without this dynamic approach, there would always be limitations—a classification with its training dataset can only in a limited way represent the diversity of a cooking process in a non-standardized environment. The downside is that, in contrast to classification, the results are only clusters, without any description. A

stirring action, for example, is only recognized as “cluster2”. The approach to this problem will be considered later on.

The selection of suitable parameters for clustering includes the absolute speed of the hand above the table surface and the angular changes of the hand. In contrast to the absolute speed in space, the values of the absolute speed above the table surface reveal significant variations between actions, for example, low velocities during cutting, medium velocities during stirring, and high velocities during tilting. With respect to angle, the variation in hand rotation is considered, not the absolute values. Here, the angle changes show low values for cutting, larger values for tilting, and very high values for stirring. Before the actual clustering analysis, the dataset goes through pr-processing where the standard deviation of the angle is calculated for each frame within a defined window width and then smoothed. Using both parameters simultaneously allows for clearer discrimination of values and leads to improved clustering results (see Figure 6) compared to using a single parameter. The characteristics of other parameters are less significant and are not taken into account in the analysis.

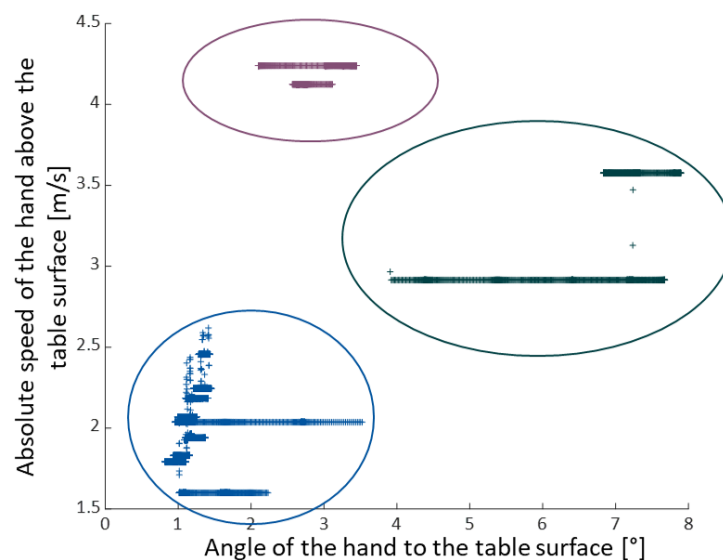


Figure 6. Three found clusters identified by analyzing the absolute speed of the hand above the table surface and the angle of the hand to the table surface. Tilt is marked as purple, stir is green and slice is blue.

In order to obtain a clustering result that is as accurate as possible, the different clustering approaches are evaluated in terms of their quality. For this purpose, the training dataset is used, where the actions have already been assigned to the frames manually. An intuitive approach could be to determine the percentage of frames correctly assigned by clustering. However, this would lead to an overvaluing of long-lasting actions. In addition, it should also be evaluated how accurately all existing individual actions were recognized. Clustering alone does not provide any information on which actions are covered by a specific cluster. Therefore, each individual cluster had to be evaluated with respect to each individual action. The combination of the two evaluation approaches presented below turned out to be the most suitable solution to establish these associations and to evaluate how accurately the clusters represent different actions:

- Determining the percentage of frames in the cluster that belong to the specific action. To do this, the number of frames associated with this action within the cluster is divided by the total number of frames in the cluster. For example: “90 percent of the cluster belongs to the slice action”.

$$\text{score}(\text{cluster}, \text{action}) = \frac{n\text{Frames}(\text{cluster}, \text{action})}{n\text{Frames}(\text{cluster})} \quad (1)$$

However, this alone is not meaningful because a high score does not necessarily mean that the majority of the action was detected. Therefore, a second approach was used.

- Determining which percentage of a specific action is covered by this cluster. To do this, the number of frames associated with that action within the cluster is divided by the total number of frames associated with that action within the entire recording. For example: “70 percent of the slice frames are present in this cluster”.

$$\text{score}(\text{cluster}, \text{action}) = \frac{n\text{Frames}(\text{cluster}, \text{action})}{n\text{Frames}(\text{recording}, \text{action})} \quad (2)$$

Combining these two approaches by multiplying Formulas (1) and (2), a result is obtained that answers the following questions: How consistent and complete is the cluster regarding a specific action?

$$\text{score}(\text{cluster}, \text{action}) = \frac{n\text{Frames}(\text{cluster}, \text{action})^2}{n\text{Frames}(\text{recording}, \text{action}) * n\text{Frames}(\text{cluster})} \quad (3)$$

An algorithm then uses these ratings to determine the most reasonable combinations of cluster and action. The corresponding scores are combined into an arithmetic mean, which is the overall score of the clustering. This automated evaluation makes it possible to quantitatively compare the approaches found. The parameters thus found are then used to apply clustering to the main dataset. In our study, a unique fingerprint is found for each action, consisting of the orientation of the back of the hand and its absolute speed of the hand above the table surface. The raw data of these two parameters are processed in such a way that the characteristics lead to the formation of the required clusters.

6. Transfer into PDDL

The motion data and the combined classification and clustering results, as well as the locations, are used to create sequences that summarize specific sections of the frames. The sequences, which can be assigned to actions known from the classification, are directly named accordingly. Location information is added for each sequence—both the location at the beginning and at the end of the action. Numeric values are transferred directly to the known locations, assuming the location with the least distance. Objects are divided into different subtypes. For example, a container like a bowl can contain ingredients. Objects such as knives and spoons are assigned to the tools subtype, while cuke and dressing are considered ingredients. The positions and states of the objects at the beginning of the scenario are known.

The schedule, which consists of the individual sequences, is now analyzed. For each action, it is recorded which object is at the active position. A “pick” action causes this object to be in the hand afterwards. Only one object can be in the hand, but, if this is a container, it can contain another object. For example, the dressing can be in the bottle. A “move” action with an object in hand changes the status of that object to be “on hand”. A “place” action sets the new position of the object. Thus, virtual object tracking is implemented. If an unknown action is performed, all available information is used to describe the action. The objects at the active position, the object in the hand, and the active position itself are used. For example, it could be:

Cuke is processed by knife on cuttingboard.

The analyzed schedule is transferred to PDDL, where the domain name and requirements are defined statically. Then the types are defined following the structure in Figure 7, representing the different objects, actors, locations and actions. The subtypes of objects mentioned before are defined as movables. Furthermore, under the type hand, the possible actors, robot and human, are defined. The specific locations are listed in the subtype location. All unknown actions detected in the analysis of the schedule are summarized under treatment.

Then, the predicates are statically defined to describe the relationships and states of the objects and actors. Five predicates are used:

- *on* describes the location of a moveable or hand
- *in* describes that an ingredient is in a container
- *is* describes the treatment of an ingredient
- *holding* describes which movable a hand is holding
- *hand-empty* describes that the hand of an actor is empty (or not)

These predicates are already used in object tracking. The information is prepared during object tracking in such a way that the predicates can be read out after each sequence. This allows the start and end situations to be read directly from the predicates in the PDDL. Furthermore, this allows the observation of changes in the predicates in the case of unknown actions. The known standard actions are also defined statically.

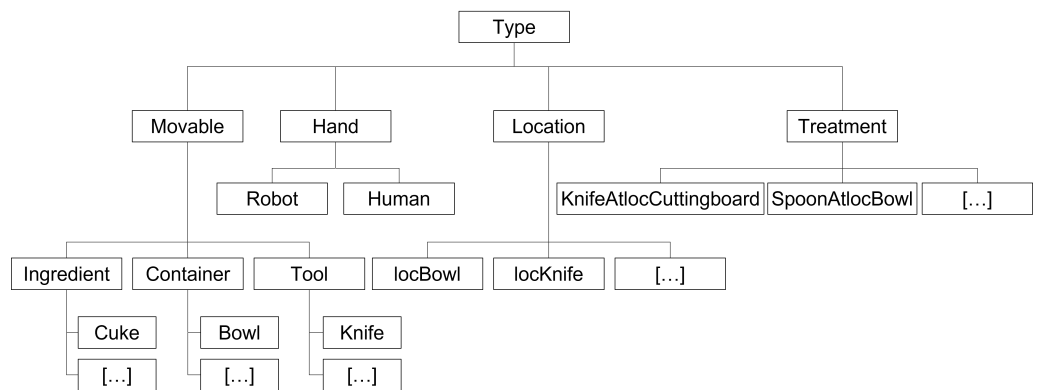


Figure 7. The structure of the types in the PDDL.

Unknown actions are now generated. Here a loop runs through all sequences of the analyzed schedule, whereby the distinction between *ToolOnHand*, *IngredientOnHand* and *ContainerOnHand* provides the general structure. This allows unknown actions to be captured and described. The problem and the associated domain are defined statically to define the specific task. The objects are named concretely and both the objects and the involved actors (hands) are specified.

The initial state is defined, with the hands empty and the objects at their initial locations. The target state is also defined to capture the intended result of the actions and, thus, describe the goal of the task. Some examples of the resulting PDDL and the plan are given in Figure 8.


```

(a) (:action pick
      :parameters (?hand - hand ?mov - moveable ?loc - location )
      :precondition (and
                    (on ?hand ?loc)
                    (on ?mov ?loc)
                    (hand-empty ?hand)
                    )
      :effect (and
              (not (on ?mov ?loc))
              (holding ?hand ?mov)
              (not (hand-empty ?hand))
              )
      )

(b) (:action KnifeAtlocCuttingboard
      :parameters (?hand - hand ?locCuttingboard - locCuttingboard
                  ?cuttingboard - cuttingboard ?knife - knife
                  ?ingredient1 - ingredient ?ingredient2 - ingredient
                  ?ingredient3 - ingredient ?KnifeAtlocCuttingboard -
                  KnifeAtlocCuttingboard )
      :precondition (and
                    (on ?hand ?locCuttingboard)
                    (on ?cuttingboard ?locCuttingboard)
                    (holding ?hand ?knife)
                    (on ?ingredient1 ?locCuttingboard)
                    (on ?ingredient2 ?locCuttingboard)
                    (on ?ingredient3 ?locCuttingboard)
                    )
      :effect (and
              (not(on ?ingredient1 ?locCuttingboard))
              (in ?ingredient1 ?cuttingboard)
              (not(on ?ingredient2 ?locCuttingboard))
              (in ?ingredient2 ?cuttingboard)
              (not(on ?ingredient3 ?locCuttingboard))
              (in ?ingredient3 ?cuttingboard)
              (is ?ingredient1 ?KnifeAtlocCuttingboard)
              (is ?ingredient2 ?KnifeAtlocCuttingboard)
              (is ?ingredient3 ?KnifeAtlocCuttingboard)
              )
      )

(c) (:goal
      (and
        (on cuttingboard1 locKnife1)
        (on bowl1 locBowl1)
        (on bottle1 locBottle1)
        (on knife1 locKnife1)
        (on spoon1 locSpoon1)
        (in cuke1 bowl1)
        (in dressing1 bowl1)
        (is cuke1 KnifeAtlocCuttingboard1)
        (is cuke1 CuttingboardAtlocBowl1)
        (is cuke1 SpoonAtlocBowl1)
        (is dressing1 SpoonAtlocBowl1)
        (is dressing1 BottleAtlocBowl1)
      )
      )

(d) 001: (move robot2 lochuman1 loccuttingboard1)
     002: (move robot1 locrobot1 locbottle1)
     003: (pick robot1 bottle1 locbottle1)
     004: (move robot1 locbottle1 locbowl1)
     005: (bottleatlocbowl robot1 locbowl1 bowl1 bottle1 dressing1
           dressing1 dressing1 bottleatlocbowl1)
     006: (move robot1 locbowl1 locbottle1)
     007: (drop robot1 bottle1 locbottle1)
     008: (move robot1 locbottle1 locbowl1)
     009: (move robot2 loccuttingboard1 loccuke1)
     010: (pick robot2 cuke1 loccuke1)
     011: (move robot2 loccuke1 loccuttingboard1)
     012: (drop robot2 cuke1 loccuttingboard1)
     013: (move robot2 loccuttingboard1 locknife1)
     014: (pick robot2 knife1 locknife1)
     015: (move robot2 locknife1 loccuttingboard1)
     016: (knifeatloccuttingboard robot2 loccuttingboard1
           cuttingboard1 knife1 cuke1 cuke1
           knifeatloccuttingboard1)
     017: (move robot2 loccuttingboard1 locknife1)
     018: (drop robot2 knife1 locknife1)

```

Figure 8. Snippets of the resulting PDDL: (a) The side task pick; (b) the unknown task cut; (c) The described goal state; (d) Extract from the final plan.

7. Conclusions

The presented approach stands out in the landscape of LfD methods due to its extensive flexibility in converting human-guided demonstrations into executable task plans using a PDDL, making it suitable as a versatile solution that can be used for various applications.

After the three side tasks are classified once, the presented approach enables one-shot robot teaching. In this way, our approach addresses a common limitation of many existing LfD methods, which is their struggle to adapt to new application areas. In addition, the transfer to a PDDL offers great agility in the creation of task schedules depending on the robotic or human actors involved.

While we highlight the potential of our approach, it is worth mentioning that this research is likely to face some challenges and limitations. Variability in human actions, changes in kitchen layouts, and handling unforeseen situations are some of the issues that require further research and development. These challenges are not unique to our approach but represent inherent difficulties in the automation of complex tasks.

As the focus of this paper is limited to the creation of task planning based on LfD, its execution requires overcoming corresponding challenges, such as mechanical implementation, robot programming, and trajectory planning. Solving these challenges requires future research.

Moreover, as household robotics advance, we must consider the ethical implications of increased automation and collaboration with machines in daily life. Ethical considerations should include privacy, safety, and the impact on human employment and society. Addressing these concerns, including transparency and accountability, should be integrated into the development and deployment of such technologies.

In conclusion, the proposed method of using passive observation to teach machines through LfD in the context of the cooking process shows great promise in enhancing automation and making it accessible to non-experts. As technology advances and more research is conducted in this area, we can expect significant strides towards achieving seamless human-machine cooperation in various real-world applications.

Author Contributions: Conceptualization, M.S., F.M., R.G. and N.M.; methodology, M.S., F.M. and R.G.; software, M.S., F.M. and R.G.; validation, M.S., F.M. and R.G.; formal analysis, M.S., F.M. and R.G.; investigation, M.S., F.M. and R.G.; resources, M.S., F.M. and R.G.; data curation, M.S., F.M. and

R.G.; writing—original draft preparation, M.S., F.M. and R.G.; writing—review and editing, M.H. and B.C.; visualization, M.S., F.M. and R.G.; supervision, M.H. and B.C.; project administration, M.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

PDDL	Planning domain definition language
LfD	Learning from demonstration
RUS	Random under-sampling

References

- Ravichandar, H.; Polydoros, A.; Chernova, S.; Billard, A., Recent Advances in Robot Learning from Demonstration. *Annu. Rev. Control Robot. Auton. Syst.* **2020**, *3*, 297–330. [CrossRef]
- Wake, N.; Arakawa, R.; Yanokura, I.; Kiyokawa, T.; Sasabuchi, K.; Takamatsu, J.; Ikeuchi, K. A Learning-from-Observation Framework: One-Shot Robot Teaching for Grasp-Manipulation-Release Household Operations. In Proceedings of the 2021 IEEE/SICE International Symposium on System Integration (SII), Iwaki, Fukushima, Japan, 11–14 January 2021; pp. 461–466.
- Infografik: Thermomix-Geschäft Brummt. Available online: <https://de.statista.com/infografik/9513/umsatz-von-vorwerk-ingeschaefsbereich-thermomix/> (accessed on 25 July 2023).
- Smart Kitchen Market Revenue Worldwide from 2017 to 2027. Available online: <https://www.statista.com/statistics/1015395/worldwide-smart-kitchen-market-revenue/> (accessed on 1 September 2023).
- Samsung Bot Chef. Available online: <https://news.samsung.com/global/tag/samsung-bot-chef> (accessed on 25 July 2023).
- Moley Robotics. Available online: <https://www.moley.com/> (accessed on 25 July 2023).
- Jiang, A.Z.; Zhou M. Design of Affordable Self-learning Home Cooking Robots. In Proceedings of the 2022 IEEE International Conference on Networking, Sensing and Control (ICNSC), Shanghai, China, 15–18 December 2022; pp. 1–6.
- Ilic, S.; Hughes, J. Complexity and Similarity Metrics for Unsupervised Identification of Programming Methods for Robot Cooking Tasks. In *Human-Friendly Robotics 2022, Proceedings of the HFR: 15th International Workshop on Human-Friendly Robotics, Delft, The Netherlands, 22–23 September 2022*; Springer Proceedings in Advanced Robotics; Springer: Berlin/Heidelberg, Germany, 2022; Volume 26.
- Bao, S.; Luo, B. Design of a fully automatic intelligent cooking robot. *J. Phys. Conf. Ser.* **2021**, *1986*, 012101.
- Yang, Y.; Zhu, H.; Liu, J.; Li, Y.; Zhou, J.; Ren, T.; Ren, Y. An Untethered Soft Robotic Gripper with Adjustable Grasping Modes and Force Feedback. In Proceedings of the IEEE International Conference on Robotics and Biomimetics (ROBIO), Jinghong, China, 5–9 December 2022; pp. 1–6.
- Lee, Y.C.; Lim, S.J.; Hwang, S.W.; Han, C.S. Development of the robot gripper for a Home Service Robot. In Proceedings of the ICCAS-SICE, Fukuoka, Japan, 18–21 August 2009; pp. 1551–1556.
- Paez-Granados, D.; Billard, A. Crash test-based assessment of injury risks for adults and children when colliding with personal mobility devices and service robots. *Sci. Rep.* **2022**, *12*, 5285. [CrossRef] [PubMed]
- Mitsioni, I.; Karayiannidis Y.; Kragic, D. Modelling and Learning Dynamics for Robotic Food-Cutting. In Proceedings of the IEEE 17th International Conference on Automation Science and Engineering (CASE), Lyon, France, 23–27 August 2021; pp. 1194–1200.
- Ivanov, S.; Webster C. Restaurants and robots: public preferences for robot food and beverage services. *J. Tour. Future* **2022**, *9*, 229–239. [CrossRef]
- Malmaud, J.; Huang, J.; Rathod, V.; Johnston, N.; Rabinovich, A.; Murphy, K. What’s Cookin’? Interpreting Cooking Videos using Text, Speech and Vision. In Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Denver, CO, USA, 31 May–5 June 2015; pp. 143–152.
- Fedorov, F.S.; Yaqin, A.; Krasnikov, D.V.; Kondrashov, V.A.; Ovchinnikov, G.; Kostyukevich, Y.; Osipenko, S.; Nasibulin, A.G. Detecting cooking state of grilled chicken by electronic nose and computer vision techniques. *Food Chem.* **2021**, *345*, 128747. [CrossRef] [PubMed]
- Miao, R.; Jia, Q.; Sun, F. Long-term robot manipulation task planning with scene graph and semantic knowledge. *Robot. Intell. Autom.* **2023**, *43*, 12–22. [CrossRef]
- Bollini, M.; Tellex, S.; Thompson, T.; Roy, N.; Rus, D. Interpreting and Executing Recipes with a Cooking Robot. In *Experimental Robotics*; Springer Tracts in Advanced Robotics; Springer: Berlin/Heidelberg, Germany, 2013; Volume 88, pp. 481–495.

19. Jeon, J.; Jung, H.R.; Yumbla, F.; Luong, T.A.; Moon, H. Primitive Action Based Combined Task and Motion Planning for the Service Robot. *Front. Robot. AI* **2022**, *9*, 713470. [[CrossRef](#)] [[PubMed](#)]
20. Fox, M.; Long, D. PDDL2.1: An Extension to PDDL for Expressing Temporal Planning Domains. *J. Artif. Intell. Res.* **2003**, *20*, 61–124. [[CrossRef](#)]
21. Tanha, J.; Abdi, Y.; Samadi, N.; Razzaghi, N.; Asadpour, M. Boosting methods for multi-class imbalanced data classification: an experimental review. *J. Big Data* **2020**, *7*, 70. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.