July 2023

Ontological Context for Gesture Interpretation[1]

Petr VANC [a], Karla STEPANOVA [a] and Daniel BESSLER [b]
[a] *Czech Institute of Informatics, Robotics and Cybernetics (CTU), Prague, Czechia*
[b] *University of Bremen, Institute for Artificial Intelligence, Bremen, Germany*
ORCiD ID: Petr Vanc https://orcid.org/0000-0002-2810-6961, Karla Stepanova https://orcid.org/0000-0003-4239-2092, Daniel Beßler https://orcid.org/0000-0002-9876-5743

Abstract. This study explores gesture interpretation by utilizing an ontological context. The aim is to store gestures and scene context data in an ontology and use its knowledge graph to actuate the robot arm to perform sets of manipulation tasks used in various environments. The knowledge graph captures the relationships between gestures, objects in the scene, and the desired actions. By putting the ontological context into use, the system can understand the meaning behind the gestures and execute the appropriate actions. The paper focuses on the development of the ontology, including the creation of class properties and the embedding of gestures within the ontology. Additionally, the paper explores how the integration of specifying context interpretation from the ontology may look to enhance the interpretation of gestures. The proposed approach aims to provide more intuitive and adaptive gesture-based supervisory control of robots in general. We tested the proposed ontological system in several tests so that it may be used in our future applications.

Keywords. ontologies, HRI, gesture interpretation, hand gestures

1. Introduction

Human-robot interaction (HRI) plays a crucial role in enabling robots to assist humans in various tasks. One important aspect of HRI is the ability of robots to understand and interpret human gestures accurately. Gesture recognition allows robots to perceive and respond to human commands and intentions, enhancing their usability and effectiveness in assisting humans. Similarly, the ability to understand and represent gestures enables robots or virtual agents to behave in a way that is easily interpretable by humans.

Robot control using hand gestures so far is mainly considered a direct mapping between gestures and actions without any context of the environment (e.g. [19]). However, context is crucial to properly interpreting meaning of the nonverbal communication. The human intent for robot control may be determined from a set of gestures in the given con-

[1] P.V. was supported by CTU Student Grant Agency (reg. no. SGS23/138/OHK3-027/23). K.S. was supported by the Czech Science Foundation (project no. GA21-31000S). D.B. was supported by the trilateral project #442588247 "AI4HRI – artificial intelligence for human-robot interaction" which is partly funded by the German Research Foundation (DFG).

text [8], [21] (i.e., interpreting the gestures with respect to the user, objects on the scene, or performed task), however a very simplified representation of the environment and context was considered, which makes it hard to reason correctly on the historical data and make a generalization to new environments. There are also a few efforts that propose a more robust knowledge representation of nonverbal communication [15], however, these do not consider the context of the environment, objects, and robot itself and are more focused on human-device interaction, i.e., using gestures for operating a game/tablet which is compared to human-robot interaction in a simplified environment that requires also different representation of affordances.

In this work, we link these two worlds by proposing a robust knowledge representation for the interpretation of gestures within the context of realistic robotic environments. To achieve accurate and context-aware gesture recognition, we built upon our previous work [21], and, in addition, represent the gestures and the relevant context in an ontology. The ontology enables us both to collect experiences from historical interactions and to reason on top of the acquired knowledge or current state. Our approach utilizes a pretrained set of gestures based on a semaphoric model which discretely classifies different types of gestures and employs the gesture toolbox [23] to prepare the target set of gestures specifically designed for the tested environment. The focus is on the application of gesture recognition in the context of kitchen environments, which can be further expanded to other workspaces.

The kitchen environment presents unique challenges for robots, as it is relatable and understandable for humans and contains various basic manipulation tasks such as food preparation, desk organization, and cleaning in a kitchen setting. In these scenarios, a subset of objects from the YCB dataset [3] is utilized consisting of kitchen equipment and food items.

The proposed ontology and related code is available at `https://github.com/petrvancjr/gesture-ontological-context-interpreter`.

2. Related Work

First, we review related work about the usage of human gestures for robot control. Second, we summarize the most significant works on modeling and representing gestures.

In previous works, hand movements were used to teleoperate the robot by directly mapping the user's hand to the robot end-effector [25]. In [4], hand motions recognized by a combination of depth cameras and inertial measurement units (IMUs) were used for robot teaching. Another way of control is commanding gestures using action signaling (e.g., [21]). In this case, recognized gestures are linked to specific robot actions.

Most of the current gesture-controlled devices (e.g., Hi5 VR Glove [7]) were used only experimentally. Only a few (e.g., Leap Motion [24], Oculus Quest's hand tracking) made it into mass production and even fewer people started using them on a daily basis. One example is the smartphone industry, where developers used hand gestures to control basic controls, e.g., music [14] or resizing pictures. Gestures are also used in virtual reality, including environment control [12]. In contrast to these approaches, all consider only a very simple notion of context. Our hypothesis is that using situation context from a rich knowledge graph is the key to making gestures a reliable and natural means of communication.

The modeling of gestures is in many cases rather informal (e.g., [1]), or restricted to geometrical characteristics (e.g., [16], [11]). For example, Ousmer et al. [16] decomposes gestures into different segments with associated hand poses to support the recognition of gestures. In contrast, our goal is rather to support their interpretation, and thus geometrical characteristics are not sufficient. A well-formalized account is provided by the SUMO ontology which focuses on modeling the communication underlying a gesture [17]. However, gestures are not characterized by the affordances of the environment in the SUMO ontology. Another related ontology is the HDGI ontology [18]. It is designed for human-device interaction and puts particular emphasis on the link between gestures and device context and affordances. However, here we consider human-robot interaction which requires a more detailed representation of affordances and context.

In most works the scene context is not taken into account, e.g. the system uses fixed mapping from gestures to robot task [19]. Other formal accounts rather employ probabilistic representation [9] for human intent recognition in shared-control robotics. It looks at the integration of human gestures and robotic actions. We build upon [21] method for mapping the context, which is using a Bayesian neural network to estimate the next user-intended action. The key is to construct a feature vector that properly describes the scene context. Contextual characteristics were extracted from the working data set based on their natural properties which were hand-picked based on common sense and stored as ontological properties.

3. Application Domain

In this section, we will describe the considered application domain including the utilized objects (Sec. 3.1), robot actions (Sec. 3.2), and gestures (Sec.3.3). For the gesture detector, we use the Leap Motion Controller [24] and *Franka Emika Panda* [5] as our robot which is popular to be used for human-robot interaction while being able to easily manipulate the objects.

3.1. Objects

The scenario in consideration comprises a table and several items that are on top of it. The objects are a subset of the popular YCB dataset [3] because they are objects which people interact with on a daily basis. We took mainly kitchen equipment and food items to accomplish specified tasks, which we will describe in the next section. The objects are randomly arranged on the scene within every new task. See Fig. 1 for the set of selected objects.

3.2. Set of actions

We defined the number of actions the robot can perform while still being able to accomplish given scenarios. The final list consists of 7 actions: *Pick*, *Pour*, *Put*, *Place*, *Move-up*, *Move-right*, and *Move-left*. Actions *Pick*, *Pour*, *Put*, and *Place* are tied to a specified object. Action *Place* places the object into specified storage, the storage defined within reach on the side of the table.

Action *Move-up* moves the robot end-effector into the home (upper) position. Actions *Move-left* or *Move-right* move the robot end-effector according to the common co-

(a) Bowl (b) Cheez It (c) Foam (d) Mug (e) Mus-tard (f) SPAM (g) Sugar (h) Toma-toes

Figure 1. Set of the selected objects. It is a subset of YCB objects [3].

ordinate axis by a certain amount. Additional object-focused features might be added for convenience. For example, if an object is near the end-effector position, the position is adapted to attain the position above the given detected object.

Additionally, the system supports a non-robotic action *Select-object* which computes probabilities of objects being selected based on the direction of the pointing finger (see Sec. 3.3 for more details).

3.3. Gesture description

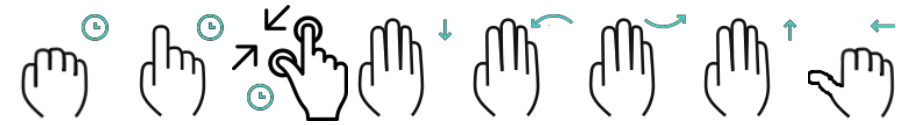
Our system utilizes two gesture types: action gestures and point (deictic) gestures.

The first action gesture type is defined based on gestures taxonomy [10], the Semaphoric Gestures Model. It is used to discretely classify a pre-trained set of gestures. We selected a set of 8 action gestures that are shown in Fig. 2. The detectors used in our system are returning the confidence of each gesture from this set in real-time. When any gesture has enough evidence, its data are written into the ontology. The gesture detection uses a combination of static and dynamic gesture detectors. The static detector uses a single time-frame hand structure (see Fig. 4), from which then the feature vector (of length 57) is extracted for gesture classification. On the other side dynamic gesture detector uses a moving time frame of hand movement but only hand pose as the feature is used. The detectors are combined to form the final *Compound gesture*, resp. specific hand configuration plus movement (see Fig. 2 description).

Point (deictic) gestures have the effect of triggering the procedure of object choice. The procedure chooses the closest object to the user's pointed line. This involves the calibration with a scene [22]. The poses of objects are retrieved from the knowledge graph which saves recent object positions. We use the CosyPose detection method [13] to get 6DoF poses of objects.

4. Ontological Characterization

The ontological nature of gestures appears to be somewhat diverse. Gestures are, on the one hand, acts of non-verbal communication where an agent attempts to convey some information to other agents in its surrounding. They are, on the other hand, also bodily expressions in the form of postures and motions. This observation is captured, for example, in the SUMO ontology where a gesture is seen as *any body motion which is also*



(a) Steady Grab | (b) Steady Point | (c) Steady Pinch | (d) Swipe Down | (e) Swipe Left | (f) Swipe Right | (g) Swipe Up | (h) Thumbs Left

Figure 2. Set of gestures. Gestures are a combination of Static and Dynamic detectors (see 3.3). For example *Steady-Grab* (a) is a combination of *no movement* with *Grab* hand configuration. Same way the *Thumbs-Left* (h) is hand movement left and *Thumb-left* hand pose.

an instance of communication [17]. Another foundational account for an ontological notion of gestures is included in the ontology for *Information Objects* where a gesture is seen as an instance of *bodily motion*, and each bodily motion is seen as the realization of an abstract piece of information[20]. However, such foundational definitions are rather vague about the actual meaning of gestures. In our implementation, we commit to the IO ontology definition simply for pragmatic reasons as the alignment with the adopted knowledge framework was less cumbersome.

The intended meaning of a gesture can often only be understood when taking context into account. A hand gesture indicating a stop signal, for instance, is vague and can only be understood if it is clear to which objects and actions it refers. Another aspect is that the detection of gestures could be wrong, but that the falsely detected gesture does not make sense in the current context in which case it could be discarded or re-classified. This suggests the importance of relationships between gestures and the context in which they occur. These relationships are of primary concern for us, but they are rarely considered in related literature about gesture ontologies or are not designed for robotics use cases.

Nevertheless, it is worth investigating to what extent existing ontologies that link gestures and context could be adopted for robotics use-cases. To this end, we adopt the HDGI ontology [18] as it defines a rather comprehensive model of gestures that also includes links to device affordances and context. The ontology does however not model interactions between agents and what is afforded to them. But this interaction is important for robotics use-cases as robots have different capabilities to execute an action. Thus, we rather employ a more fine-grained notion of affordance from an existing ontology that was designed with robotics use-cases in mind [2]. We consider an affordance as the description of a disposition and a disposition is an absolute property, it does not depend on a context. Furthermore, the ontology defines affordances as the descriptive context between dispositional pairs and thus can express a relation between two disposed objects. For example, the robot is disposed to handle small-sized objects while a small-sized object is disposed to be grasped, carried, thrown, and so on. The concepts and relations used or defined in the proposed ontology are shown in Fig. 3.

The central notion is the `Gesture` concept. It is characterized as a type of task, i.e. as a sub-concept of `CommunicationTask`. The notion of task is imported via the robotics affordance ontology [2] which in turn imports the foundational ontology *DOLCE+DNS Ultralite (DUL)* [6]. Following DUL, tasks describe how certain events are to be interpreted, executed, etc. In the case of `CommunicationTask`, this is done through the three roles linking sender, receiver, and message to the task. For gestures, the role of the message is actually taken by the event that executes the gesture, i.e. the act of performing the

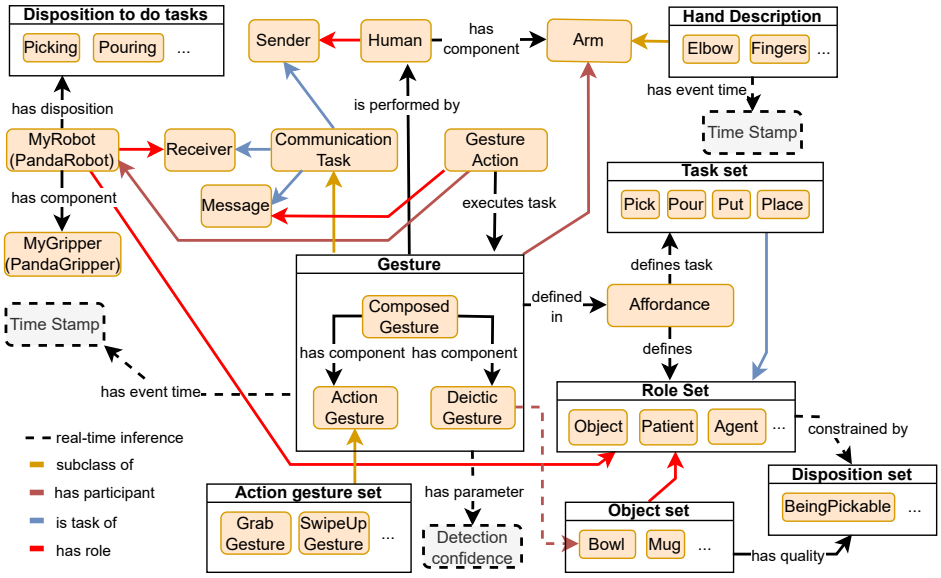


Figure 3. Ontology graph shows class definitions with *Composed Gesture* in the middle. *Gesture set* if defined in Fig. 2, *Object set* is defined in Fig. 1, *Hand Description* in detail is on Fig. 4, Scene objects have qualities of static properties (see Fig. 5. In our implementation, we define four easy robot tasks, see Sec. 3.2.

gesture is the message transported. Finally, we say that each gesture *is defined in* one or more affordances representing some action potential in the environment. For example, gesture *grab* has affordance representing *picking* the object.

Definition 1 *A* gesture *is a communication task that is defined in an affordance.*

An affordance further defines a task related to affordance and creates constraints for objects taking the roles of the task. Namely, those objects need to be the host of certain dispositions. For example, an affordance of picking up an object may refer to a task where the picked object must be a host of the *pickable* disposition, and where the agent must be the host of the *can-pick* capability (disposition).

The gesture concept is further decomposed into three cases: `DeicticGesture`, `ActionGesture`, and `ComposedGesture`. First, a *deictic* gesture is a pointing gesture used to draw the attention of the receiver to a particular object or region of interest. Second, an *action gesture* refers to a task request. Finally, a *composed gesture* is a combination of several gestures during an episode. An instance of `ComposedGesture` is created each time the gesture episode ends. In our implementation, this depends on our hand sensor [24], which has a limited field of detection. The end of the episode is defined as the hand disappearing from the detection area. Each gesture event gives us data about confidence, timestamp, name of gesture which has been triggered, and potentially the relevant selected object, on which the user wants to work.

Features of scene objects are defined as object properties. These properties are shown in Figure 5 with relation to object type (e.g., Bowl). Properties are hand-picked based on our needs: Sizes (*SmallSize* and *LargeSize*) based on the ability to fit inside the *PandaGripper*, *Color* as a visual property, *Sharpness* property defining rounded or sharp

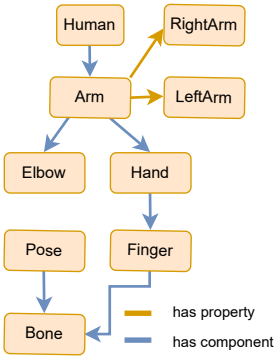


Figure 4. Hand structure definition. The human can have max. 2 instances of *Arm* with the property if it is *RightArm* or *LeftArm*. Each Arm has always single Hand. One hand has always 5 fingers and each finger has 4 finger bones (except the Thumb which got 3).

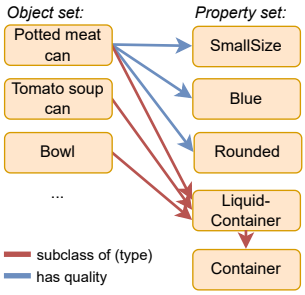


Figure 5. Small sample of object relations with its static properties.

objects (this property might be used when clarifying object choice), and finally Object types defining whether the object is a container and of which type (*Liquid-container*, or *Object-container*). Dynamic properties are supposed to be updated in real-time to keep the world representation up-to-date. They include current *Container capacity*, *Accessibility* (disposition to be interacted, e.g. object is on top of a stack or in reach by robot's gripper), or if an object has been already manipulated, which can help estimate the user's next choice based on their preference. Geometric parameters include instant *Pose* value. Perception methods might be used for some properties, e.g. get pose (in our case by CosyPose method [13]) of the object. If no such technique is available, the logic methods might be used, e.g. when action *Pour* is done, the capacity of a container changes.

5. Experience Acquisition

In this chapter, we discuss the potential approach for acquiring experience in the context of gesture interpretation using ontological context. While this work is currently in progress, we outline the steps and considerations that could be taken to acquire the necessary to test the usage of our knowledge graph and the possibility to improve the context-based gesture control system.

To acquire experience in gesture interpretation, a large dataset of human-robot interactions needs to be collected. This dataset should include our defined kitchen scenarios and tasks in which humans interact with the robot using described gestures. From the interactions, the data about the user's Task instance reference for the current context (Dynamic properties of objects and applied gesture). The next step is to take advantage of ontologies and scale the number of object properties.

Instantaneous object properties are written in real-time into ontology. As we discussed poses of objects [13] and other properties experimentally or by hand, e.g. current capacity of Liquid-container. Accessibility of an object by the property, that the given object is in a predefined boundary, and checker to estimate if no other object is detected on top of the given object.

The Arm instances also enable us to store raw hand movements. Based on this data structure, we may run gesture set classification that uses collected data training and improving discussed gesture recognizers.

6. Discussion and Conclusion

Representing world and gesture information as ontology enables us to define the knowledge of the world in the right format with the possibility to scale its properties by size without getting confusing. One of the ways is to generate a better embedding vector for context-dependent action generation.

The following steps involve the validation of the proposed setup and a comparison test of how useful is using the knowledge graph to other methods in terms of scaling, for example, the number of properties on context-based action estimation. Last but not least, it would be interesting to explore the assembly of context vector embeddings with automatic methods. This would involve some scraping method, evaluator, and discriminator, to evaluate if the context vector is chosen properly for a given environment.

References

[1] Arendttorp, E. M. N., Rodil, K., Winschiers-Theophilus, H., and Magoath, C. (2022). Overcoming legacy bias: Re-designing gesture interactions in virtual reality with a san community in namibia. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, CHI '22, page 1–18, New York, NY, USA. Association for Computing Machinery.

[2] Beßler, D., Porzel, R., Mihai, P., Beetz, M., Malaka, R., and Bateman, J. (2020). A formal model of affordances for flexible robotic task execution. In *Proc. of the 24th European Conference on Artificial Intelligence (ECAI)*.

[3] Calli, B., Singh, A., Walsman, A., Srinivasa, S., Abbeel, P., and Dollar, A. M. (2015). The YCB object and model set: Towards common benchmarks for manipulation research. In *2015 International Conference on Advanced Robotics (ICAR)*, page 510–517.

[4] Du, G., Chen, M., Liu, C., Zhang, B., and Zhang, P. (2018). Online robot teaching with natural human–robot interaction. *IEEE Transactions on Industrial Electronics*, 65(12):9571–9581.

[5] Franka Emika GmbH (2018). Panda: The versatile industrial robot arm.

[6] Gangemi, A. (2021). *The DOLCE+DnS Ultralite ontology (DUL) Version 4.0*.

[7] Glowacki, B. R. and Freire, R. (2019). An open source etextile vr glove for real-time manipulation of molecular simulations.

[8] Islam, J., Ghosh, A., Iqbal, M. I., Meem, S., and Ahmad, N. (2020). Integration of home assistance with a gesture controlled robotic arm. In *2020 IEEE Region 10 Symposium (TENSYMP)*, pages 266–270.

[9] Jain, S. and Argall, B. (2018). Recursive bayesian human intent recognition in shared-control robotics. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, page 3905–3912.

[10] Karam, M. and m. c. schraefel (2005). A taxonomy of gestures in human computer interactions. Project report, University of Southampton.

[11] Khairunizam, W., Ikram, K., Bakar, S. A., Razlan, Z. M., and Zunaidi, I. (2018). Ontological framework of arm gesture information for the human upper body. In Hassan, M. H. A., editor, *Intelligent Manufacturing & Mechatronics*, pages 507–515, Singapore. Springer Singapore.

[12] Khundam, C. (2015). First person movement control with palm normal and hand gesture interaction in virtual reality. In *2015 12th International Joint Conference on Computer Science and Software Engineering (JCSSE)*, pages 325–330.

[13] Labbé, Y., Carpentier, J., Aubry, M., and Sivic, J. (2020). Cosypose: Consistent multi-view multi-object 6d pose estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*.

[14] Mccarthy, S. (2019). LG touch-less interface smartphone will unveil at MWC 2019 in Barcelona. *UWIRE Text*, page 1–1.

[15] Neto, P., Simão, M., Mendes, N., and Safeea, M. (2019). Gesture-based human-robot interaction for human assistance in manufacturing. *The International Journal of Advanced Manufacturing Technology*, 101(1):119–135.

[16] Ousmer, M., Vanderdonckt, J., and Buraga, S. (2019). An ontology for reasoning on body-based gestures. In *Proceedings of the ACM SIGCHI Symposium on Engineering Interactive Computing Systems*, EICS '19, New York, NY, USA. Association for Computing Machinery.

[17] Pease, A., Niles, I., and Li, J. (2002). The suggested upper merged ontology: A large ontology for the semantic web and its applications. In *Working notes of the AAAI-2002 workshop on ontologies and the semantic web*, volume 28, pages 7–10.

[18] Perera, M., Haller, A., Rodríguez Méndez, S. J., and Adcock, M. (2020). HDGI: A human device gesture interaction ontology for the internet of things. In *The Semantic Web – ISWC 2020: 19th International Semantic Web Conference, Athens, Greece, November 2–6, 2020, Proceedings, Part II*, page 111–126, Berlin, Heidelberg. Springer-Verlag.

[19] Raheja, J. L., Shyam, R., Kumar, U., and Prasad, P. B. (2010). Real-time robotic hand control using hand gestures. In *2010 Second International Conference on Machine Learning and Computing*, pages 12–16.

[20] Sanfilippo, E. M., Jeanson, L., and Laroche, F. (2018). Towards an ontology for information objects. 1st workshop on semantic web technologies for human and social sciences, SWTHS, Catania, Italie.

[21] Vanc, P., Behrens, J. K., and Stepanova, K. (2023a). Context-aware robot control using gesture episodes. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, page 9530–9536, London, United Kingdom. IEEE.

[22] Vanc, P., Behrens, J. K., Stepanova, K., and Hlavac, V. (2023b). Communicating human intent to a robotic companion by multi-type gesture sentences. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), to appear in*.

[23] Vanc, P., Stepanova, K., and Behrens, J. K. (2022). Controlling robotic manipulations via bimanual gesture sequences. In *2022 IEEE International Conference on Robotics and Automation (ICRA) Workshop*, page 2.

[24] Weichert, F., Bachmann, D., Rudak, B., and Fisseler, D. (2013). Analysis of the accuracy and robustness of the leap motion controller. *Sensors*, 13(55):6380–6393.

[25] Zhang, W., Cheng, H., Zhao, L., Hao, L., Tao, M., and Xiang, C. (2019). A gesture-based teleoperation system for compliant robot motion. *Applied Sciences*, 9(24):5290.