IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 29 March 2024, accepted 10 April 2024, date of publication 15 April 2024, date of current version 22 April 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3389091

RESEARCH ARTICLE

Smart Perception for Situation Awareness in Robotic Manipulation Tasks

ORIOL RUIZ-CELADA, ALBERT DALMASES, ISIAH ZAPLANA, AND JAN ROSELL
Institute of Industrial and Control Engineering, Universitat Politècnica de Catalunya, 08034 Barcelona, Spain

Corresponding author: Isiah Zaplana (isiah.zaplana@upc.edu)

This work was supported by European Commission's Horizon Europe Framework Program with the Project IntelliMan (AI-Powered Manipulation System for Advanced Robotic Service, Manufacturing and Prosthetics) under Grant 101070136.

ABSTRACT Robotic manipulation in semi-structured environments require perception, planning and execution capabilities to be robust to deviations and adaptive to changes, and knowledge representation and reasoning may play a role in this direction in order to make robots aware of the situations, of the planning domains and of their own execution structures. This paper proposes an approach aimed at enhancing the perception capabilities of robotic systems through the integration of various technologies. In particular, the novelties of the proposed smart perception module include the combination of visual sensor data, object detection, and pose estimation techniques, leveraging a fiducial markers and deep learning-based methods, being able to integrate multiple sensors and perception pipelines. In addition, reasoning capabilities are introduced through the utilization of ontologies. The result is a robust and smart perception system capable of handling both simulated and real-world scenarios which in turn provides the required functionalities to allow the robot to understand its surroundings, with a primary focus on robotic manipulation tasks. The discussion on the tools used and the key implementation details are included, as well as the results in some simulated and real scenarios that validate the proposal as a module that provides situation awareness to allow a manipulation framework to adapt the robot actions to uncertain and changing scenarios.

INDEX TERMS Perception, situation awareness, robotic manipulation, reasoning, ontologies.

I. INTRODUCTION

As robotic systems evolve, they are increasingly being deployed in dynamic, unstructured environments that require advanced capabilities beyond simple observation and reaction. Today robots are expected to understand, plan, and make decisions while adapting to changes in their environment. This shift has produced a significant growth in the field of cognitive robotics, which seeks to equip robotic systems with a higher degree of autonomy and self-configuration capabilities [1].

Cognitive robotics is about creating systems that can reason, learn from experience, accumulate knowledge, understand, and even exhibit social behavior to some degree. In this context, perception is the link connecting the physical world to the robot's functionalities and forms the foundation of cognitive robotics: providing a smart perception systems that transcend basic functionalities, will give a deeper understanding of the world that may enable more intelligent responses.

The current work proposes a smart and adaptable perception module that manages data from multiple vision sensors in order to represent the robotic environment in a manner that supports making reasoned deductions. This module serves as a crucial subsystem within a broader autonomous robotic framework, providing the necessary perceptual capabilities for higher-level cognitive functions such as planning and execution. Two distinct pipelines for interpreting environmental data are implemented. The first one is a fiducial-based pipeline using ArUco markers, that provides a simple yet efficient mechanism for detecting and estimating the pose of predefined marker tags. The second estimates 6D poses from objects using RGB images as input. This is achieved by combining a Mask R-CNN instance segmentation algorithm

The associate editor coordinating the review of this manuscript and approving it for publication was Yangmin Li.

implemented by leveraging the Detectron2 platform[1] and an Augmented Autoencoder (AE) architecture [2]. In order to control the robotic hardware, implement the aforementioned pipelines and efficiently manage data and provide perception-related functionalities, the module is developed using the Robot Operating System (ROS) middleware. ROS scalability complements the modular design of the proposed system, enabling potential future expansion and enhancements.

Furthermore, a Knowledge Representation and Reasoning (KR & R) structure is employed to make robots understand their environment and infer new information from the existing one. Such new information can be retrieved via specific queries, enhancing the system decision-making capabilities. The integration of ontologies into the system is focused on reasoning over spatial relationships between objects in the system and their location within symbolic regions of the perceived environment as a proof of concept of the module functionality and the reasoning mechanism.

The contributions of the proposal are the following:

- Implementation of a perception module that can flexibly handle data from various camera sensors, using two distinct but complementary perception pipelines, process the data and provide visualization and data retrieval capabilities.
- Incorporation of a Knowledge Representation and Reasoning capabilities to the perception module, by utilizing ontologies (built upon the Autonomous Robotics IEEE standard ontology) to structure the data and provide a reasoning layer over the perceived data, allowing the robot to be aware and understand the current situation of the environment.
- Implementation of a ROS interface to facilitate the integration of the module into larger autonomous robotic frameworks, showcasing the practical applicability and scalability of the current work.
- Discussion on the available tools for the implementation of smart perception projects.

The aim of this work is not to improve the accuracy performance of existing perception pipelines, rather to demonstrate how they can be integrated with KR & R capabilities to endow the system of situation awareness.

This paper is organized into six main sections. Following this introduction, section II presents the background information and related works, covering key concepts such as perception for manipulation, knowledge representation and reasoning, and a review of relevant literature. Section III provides an overview of the general schema of the framework, focusing on the situation awareness block of the framework and its proposed architecture. The specifics of the situation awareness block as a perception module are then detailed in section IV. Section V presents the results of the work, including the experimental setup, validation, and a discussion on the findings and their implications. Finally, section VI summarizes the main conclusions, the potential impact of the project, and the directions for future work.

[1]https://github.com/facebookresearch/detectron2

II. BACKGROUND

This project intersects multiple domains including object detection, pose estimation, and the integration of ontologies within robotic systems, each with extensive literature and related works. This section exposes the theoretical foundations and essential concepts for each relevant area in the project, as well as implementation choices and particularizations.

A. PERCEPTION FOR MANIPULATION

Perception for manipulation is primarily focused on object detection and pose estimation. The perception mechanisms are responsible of transforming raw sensor data into a structured and comprehensible representation of the world. The output typically comprises a list of identified objects with attributes such as the object type, position and orientation. Detection and pose estimation techniques can be divided into those based on fiducial markers and the learning-based methods.

Fiducial markers, while simple, efficient and reliable, are restricted to known tagged objects. Marker-based approaches, including popular fiducial marker libraries like ArUco[2] and AprilTag,[3] have minimal computational demands and work well in real-time on devices with limited computational power.

On the other hand, learning-based methods offer greater adaptability, allowing the identification of a wide range of objects without prior tagging. However, their deployment requires higher computational resources and large annotated training datasets. Notable examples include Mask R-CNN [3] or YOLO[4] for object detection, and EfficientPose [4], CosyPose [5], Augmented Autoencoders [2], PoseCNN [6], or Deep Object Pose Estimation (DOPE) [7] for pose estimation.

In the following subsections, each of the two object detection and pose estimation pipelines choices are further developed. Note that the module implementation allows users to choose the detection methods (marker-based, learning-based or a combination of both) and its respective configuration for each active camera in the system. This way, the hardware management and the perception capabilities can be customized according to the perceptive needs of the task.

1) ArUco MARKER-BASED OBJECT DETECTION AND POSE ESTIMATION

ArUco tags are square markers with a binary matrix encoding an identifier, allowing each tag to be uniquely recognized. This makes them a popular choice in computer vision and robotics for their ease of detection and pose estimation, even

[2]https://www.uco.es/investiga/grupos/ava/portfolio/aruco/
[3]https://april.eecs.umich.edu/software/apriltag
[4]https://github.com/ultralytics/ultralytics

under variable lighting conditions and with low-resolution cameras. For the object detection and pose estimation task, a unique ArUco ID is assigned to each object, enabling the system to detect, recognize, and determine the position and orientation of these objects relative to the camera. Note that this process requires human intervention to physically tag the objects and associate the corresponding ArUco IDs with each one, as well as the transformation between the tag reference frame and the object reference frame for those objects whose model is known. The proposed pipeline leverages the aruco_ros[5] package for enhanced flexibility to manage any arbitrary number of different marker sizes. The implemented module allows to flexibly configure the different tag sizes to be used, the ArUco ID ranges per size, and also permits to associate different tags to the same object if required.

2) MASK R-CNN AND AUGMENTED-AUTOENCODER-BASED OBJECT DETECTION AND POSE ESTIMATION

The chosen approach to learning-based object detection and pose estimation treats the 2D object detection as a sub-problem of the 6D object pose estimation task. This involves initially obtaining a 2D representation of all objects in an RGB image, while simultaneously determining their classes. Mask R-CNN is the model selected for this task, providing the object label, bounding box, and binary mask for each detected object. This is implemented using Detectron2,[6] a library that eases the implementation of state-of-the art object detection algorithms.

After the 2D detection process, the pipeline determines the translations and 3D orientations of the objects relative to the camera sensors. The adopted Augmented Autoencoder method for this task is trained on synthetic views of the 3D models to estimate object orientation. This approach, in contrast to other methods, does not require large volumes of manually pose-labeled object data for training but relies on synthetic 3D models instead.

Both the Mask R-CNN and Autoencoder architecture have been adapted so that they can be integrated into the ROS environment.

B. KNOWLEDGE REPRESENTATION AND REASONING

The knowledge representation and reasoning block is built taking ontologies as its core. An ontology can be defined as an explicit, formal description of terms within a specific domain, along with their relations [8]. Ontologies are an essential part of the smart section of the perception module, not only as a mechanism to interpret and structure the data generated by the object detection and pose estimation components, but also as a tool to reason and infer new knowledge.

In practical terms, an ontology is composed of multiple components like concepts (or classes), which represent ideas or entities within a domain; relations, acting as the explicit connections or associations that are declared between these entities; and properties, which describe the unique characteristics, attributes, or features of these entities. Alongside these components, instances (or individuals) are added to represent specific manifestations of the concepts within the domain. When these individuals are integrated into the ontology, the entire set of entities, their relations, and properties collectively form what is referred to as a knowledge base (KB). In essence, a KB is a repository that contains instances of classes described within the ontology structure, transforming the abstract constructs of an ontology into usable information, and can be used to build a World Model [9].

In the context of robotics, ontologies allow enhancing data communication, integration, and interoperability across diverse systems or elements within the same system, providing a shared and accessible understanding of a domain. In addition, ontologies facilitate the reuse or extension of already existing domain knowledge, such as standardized ontologies or specific ontologies developed by research groups. This leads to more efficient development processes, reducing potential errors and avoiding the redefinition of already defined concepts. Bearing this factors in mind, the ontology designed for the perception module is built upon the Autonomous Robotics (AuR) [10] IEEE standard ontology to avoid redefining terms, ease of future expansion and compatibility.

A great challenge of ontology usage is found in extracting, inferring, and manipulating the knowledge encoded within an ontology. This is where query languages such as SPARQL[7] play a key role. SPARQL is a semantic query language for databases able to retrieve and manipulate data stored in Resource Description Framework (RDF) format. This querying mechanism is particularly useful in the context of ontologies represented using the Web Ontology Language (OWL), which is often serialized in RDF, since it can navigate and search in large and complex sets of data.

In addition to SPARQL, rule languages such as Semantic Web Rule Language (SWRL[8]) play a key role in the context of ontology-based robotic systems. This tool essentially adds an extra layer of expressiveness to ontologies, which are unable to express rule-based knowledge on their own. In this context, SWRL allows users to write if-else type of rules expressed in terms of OWL concepts to provide more powerful deductive reasoning capabilities. For instance, Figure 3 illustrates how SWRL rules are used to express conditions like: ''If an object has a specific property, like a specific color, then it is graspable by a robot''. These rules can encapsulate more complex implicit knowledge about the world and can be automatically applied by the inference engine every time the knowledge base is updated. SWRL rules can reference named classes, properties, individuals, and data values from an ontology but also have built-ins for providing basic arithmetic calculations, string manipulations,

[5]https://github.com/pal-robotics/aruco_ros
[6]https://github.com/facebookresearch/detectron2
[7]https://www.w3.org/TR/sparql11-query/
[8]https://www.w3.org/Submission/SWRL/

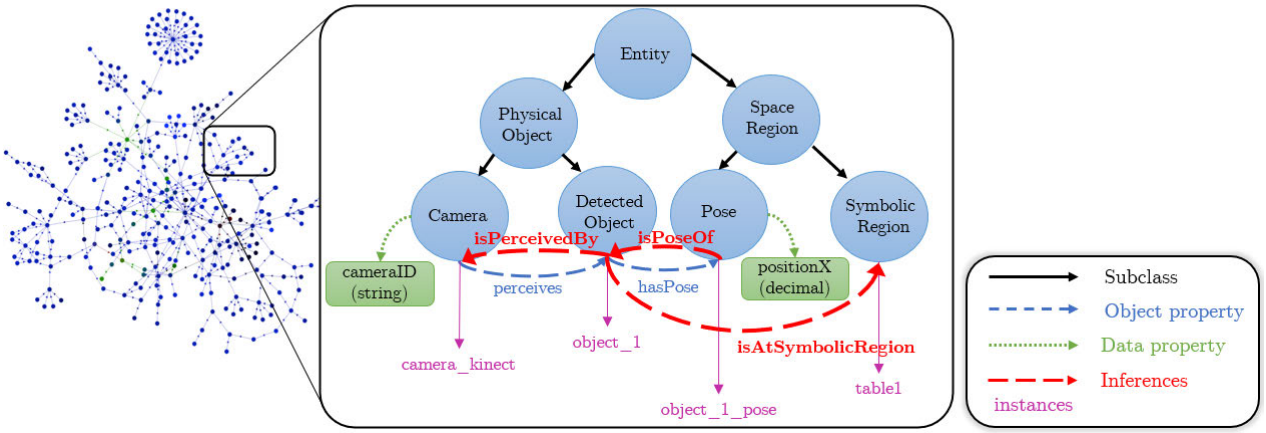


FIGURE 1. Visual representation of an example of perception ontology. The concepts of the domain are represented as classes (`PhysicalObject`, `SpaceRegion`, etc.), their known relationships as object properties (`perceives`, `hasPose`, etc.) and specific characteristics of instances as data properties (`cameraID`, `positionX`, etc.). In this example, "`object_1`" is an instance of the "`DetectedObject`" class with an associated pose "`object_1_pose`" that has a certain property "`positionX`" value. Thanks to this kind of constructions and definitions, the ontological structure is able to infer new data (shown in red), for instance it can reason if a certain object is at a symbolic region using the "`isAtSymbolicRegion`" property, given its associated position properties like "`positionX`", "`positionY`" and "`positionZ`".

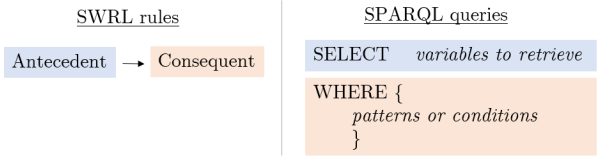


FIGURE 2. On the left, the structure of a SWRL rule is composed by an antecedent (the "if" part) and the consequent (the "then" part) linked with a "->" symbol). On the right, the structure of a typical "SELECT" SPARQL query. SELECT clause specifies the variables we are interested to retrieve and WHERE specifies the patterns that the data need to match.

and other commonly needed functions. SWRL rules can be used to infer spatial relationships and combine it with task-related properties such as if the robot can reach a location [11], [12]. Finally, as a reasoning engine, the usage of rule reasoners like Pellet[9] are required to reason within the ontological framework.

Regarding the construction and manipulation of ontologies, OWL (Web Ontology Language) ontology language is a good choice since it is built on the foundation of Description Logic (DL), compatible with globally accepted web protocols and ensures interoperability with a variety of data formats like Extensible Markup Language (XML) or Resource Description Framework (RDF). Furthermore, OWL adopts an *open-world* assumption that allows assuming that not only specifically defined predicates can be true. For example, if a robot is operating in an unknown environment, and it does not detect any obstacles, a *closed-world* assumption would assume that there are none, whereas an *open-world* assumption is open to the fact that there might be some, being open to new information.

Developing an ontology requires careful decision-making about the proper class structure and the use of classes or data properties to establish relationships. To develop such task, Protégé,[10] an open-source ontology editing platform with a user-friendly interface, is used. Protégé simplifies the process of creating complex classes, properties, and relationships, and includes in-built reasoners and plug-ins supporting SWRL rules and SPARQL queries. Furthermore, Owlready2[11] is chosen as a Python API to manage the data in the ontology, setting of SWRL rules and the execution of SPARQL queries on the ontology directly from the ROS environment.

C. RELATED WORKS

Knowledge representation and perception-based semantic understanding are relevant research areas in intelligent robotics, with a particular focus on ontology-based knowledge representation. This approach has emerged as an important foundation of autonomous robots, allowing these systems to interact with a multitude of environments, specially those that are dynamic, unstructured, and partially observable. This allows robots to perform tasks, make decisions, and adapt to their surroundings, irrespective of their field of application. Key contributions in this field are found across various robotic disciplines, including manipulation [13], [14], [15], motion planning [11], [12], navigation [16], [17], [18], [19], [20], human-robot interaction [21], service robotics [22], social robotics [23], search and rescue operations [24], and more broad-based industrial applications [25], [26], [27].

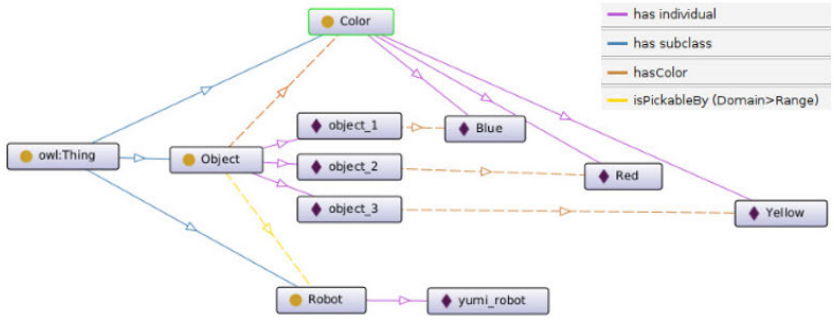
More focused on the subject of the current proposal, it is relevant to further review the significant role that ontology-based perception plays in enhancing the capabilities of robotic systems. For example, KnowRob 2.0 [13] offers a comprehensive knowledge processing system for complex manipulation tasks. It leverages ontology to bridge the gap between high-level instructions and the planning and

[9] https://github.com/stardog-union/pellet

[10] https://protege.stanford.edu/

[11] https://github.com/pwin/owlready2

(a)



(b)
```
Object(?obj), Robot(?robot), hasColor(?obj, Red) -> isPickableBy(?obj, ?robot)
```

(c)
```
SELECT ?object ?property ?robot
WHERE { ?object onto:isPickableBy ?robot.
        BIND("isPickableBy" AS ?property) }
```

(d)
```
* Owlready2 * Pellet Reasoner Synched!
[object2 isPickableBy yumi_robot]
```

FIGURE 3. Example of the steps to reason "If an object is red, then it is graspable by a robot". (a) Definition of the ontological structure in Protégé and visualized in a graph form. (b) Definition of the SWRL rule to formalize the reasoning. (c) Definition of SPARQL query used to retrieve the inferred data. (d) Result shown in the terminal after running the Pellet Reasoner engine in owlready2.

execution stages. The CRAM cognitive architecture [28] uses KnowRob 2.0 as its Knowledge Representation and Reasoning framework for everyday manipulation tasks. As its Perception Executive, it employs RoboSherlock [29], which offers a symbolic language to specify perception tasks. It uses logical atoms as annotations to perform structured queries, using SWI-Prolog as its main reasoning engine. RoboEarth [18], which is also built upon KnowRob capabilities, enables robots to rapidly learn and adapt to complex navigation tasks by using ontology to encode concepts and relationships in navigation maps. However, KnowRob-based approaches are not built to be compatible with the AuR standard [10], limiting their shareability across different autonomous robotic applications.

The Robot control for Skilled Execution of Tasks (ROSETTA) ontology [25] focuses on robotic devices and skills for manufacturing tasks. It introduces an object recognition module that merges perception data with geometric features in a Knowledge Base. The Smart and Networked Underwater Robots in Cooperation Meshes (SWARMs) ontology [17] enables shared understanding among robots in maritime missions. It also includes a probabilistic ontology, PR-OWL, for managing uncertainty in sensor information interpretation. The Perception and Manipulation Knowledge (PMK) [30] ontology and the Robot Task Planning Ontology (RTPO) [20], facilitate complex task planning for autonomous robots. PMK extends its knowledge base with sensor-related knowledge and is based on AuR, while RTPO focuses on an efficient knowledge representation in robot task planning, incorporating both continuous and discrete perceived information. These works focus on representing Knowledge relevant to their application, but do not focus on the integration of perception pipelines to incorporate Knowledge from the world, neither include data-driven methodologies.

III. APPROACH OVERVIEW

This section provides a high-level overview of the general robotic manipulation framework, called BE-AWARE [31]. The approach presented here proposes the smart perception component to achieve Situation Awareness in the framework, which will be integrated with the rest of the modules. The various layers of abstraction involved are explained, focusing on the situation awareness block and its architecture implemented by the smart perception module proposed here.

A. GENERAL SCHEMA

BE-AWARE is a manipulation framework conceived to enhance the basic functions of perception, planning and execution by an ontology-based knowledge representation and a reasoning core. The framework schema, shown in Figure 4, comprises a threefold structure. The Primary Functions triplet (in white) correspond to the basic perception, planning and execution modules. The Awareness triplet (in red) uses the KR & R core to allow the robot to be: a) aware of the situation (i.e. of the objects in the environment, their features and relative locations), aware of the domain (i.e. of the predicates describing the state and of the actions and their preconditions and effects), and aware of the execution (i.e. of the execution structures with associated monitoring and recovery strategies). Finally, the Adaptation triplet (in green) use these awareness capabilities to allow the framework to achieve a robust and reliable adaptive behavior to successfully perform manipulation tasks in semi-structured and changing scenarios by being able to: a) automatically set the planning problem by reasoning on the initial and goal

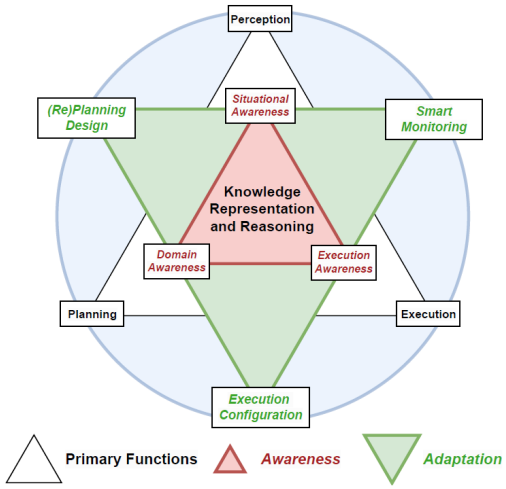


FIGURE 4. General schema of the BE-AWARE framework (taken from [31]).

situations and on the domain, b) automatically configure the execution by reasoning on the execution structures and on the domain actions, and, c) automatically tune the monitoring procedures by reasoning on the task execution structure and on the current and desired situations.

B. SITUATION AWARENESS

The proposed perception module serves as the foundation of the situation awareness block, integrating the principles of robust software infrastructure and intelligent perception into the system. The situation awareness will be supported by an ontology to further understand the environment, objects within it, the robots themselves, their features and spatial relations. Incorporating the idea of situational awareness, the system can go beyond mere data collection and interpret their meanings and relationships to have an updated knowledge state of the world at every instant of interest.

The core of the system infrastructure is built based on the Robot Operating System (ROS). The use of ROS significantly accelerates the implementation process and eliminates the need to develop each functionality from scratch. To complement ROS, the RViz 3D visualization tool and the Gazebo 3D robotics simulator are used. RViz serves as a powerful tool for visualizing sensor data and achieving a more intuitive understanding of the system's operations. On the other hand, the Gazebo simulator can emulate a testing environment including sensors, objects, and other robotic hardware.

The smart aspect of the current module relies on a custom-developed "smart perception ontology". This ontology is designed with its domain centered on the possible requirements of robotic manipulation tasks from the perception point of view. It includes concepts regarding objects in the workspace, their sensed physical attributes, spatial relationships between them, and other entities of interest. The ontology serves as the structure for a dynamic knowledge base, which is updated based on the processed perceived data of the module. Finally, the knowledge base is used to reason about the environment using a set of defined SWRL rules, and the information can be retrieved through SPARQL queries.

C. SYSTEM ARCHITECTURE

The proposed Perception Module architecture, shown in Figure 5, consists of multiple functional blocks integrated as a ROS node structure along with the KR & R part. The main aspects and ideas are explained next, from the initialization of sensors, the processing of the raw data into useful information and its assertion on the ontology, to the final retrieval of reasoned information.

- **Sensor Initialization block:** The perception module is initialized by starting the desired set of sensors. So far, only camera sensors are considered. Its activation is done using their respective hardware drivers along with ROS launch files. The block extracts raw camera image data and publishes it on ROS topics. The implementation is well structured to be extendable to other type of sensors; RFID sensors will be considered in the future.
- **Perception Nodes block:** Once the cameras are active, this block forms the foundation for object detection and pose estimation operations. It automatically reads the user-defined specifications from a configuration file to launch and set up, for each active camera, the desired detection pipelines characteristics. The block outputs the perceived data in a separate ROS topic for each camera and active detection method in the system.
- **Perception Manager block:** This block is responsible for synchronizing all the data contained in the output ROS topics of the previous block and processes it into a unified stream of information also broadcasted as a ROS topic. The treatment process mainly implies fusion of data from different sensors, merging poses when necessary, and assigning unique identifiers to every object. Furthermore, the block also aims to provide functionalities such as visualization of the perceived environment and the retrieval of perception data from the system in the form of ROS services. Up to this point, the perception module might resemble other traditional perception systems that do not incorporate any reasoning capabilities.
- **Knowledge Base Manager block:** This block bridges the ROS modules with the ontology-based knowledge management system. This allows the incorporation of the perceived data into the ontology, setting up the SWRL rules, performing reasoning and infer new information and also retrieve relevant information via SPARQL queries.

IV. THE SMART PERCEPTION MODULE

After introducing the general system architecture, this section aims to provide more detail on the working aspects of the perception nodes block, the perception manager block, the

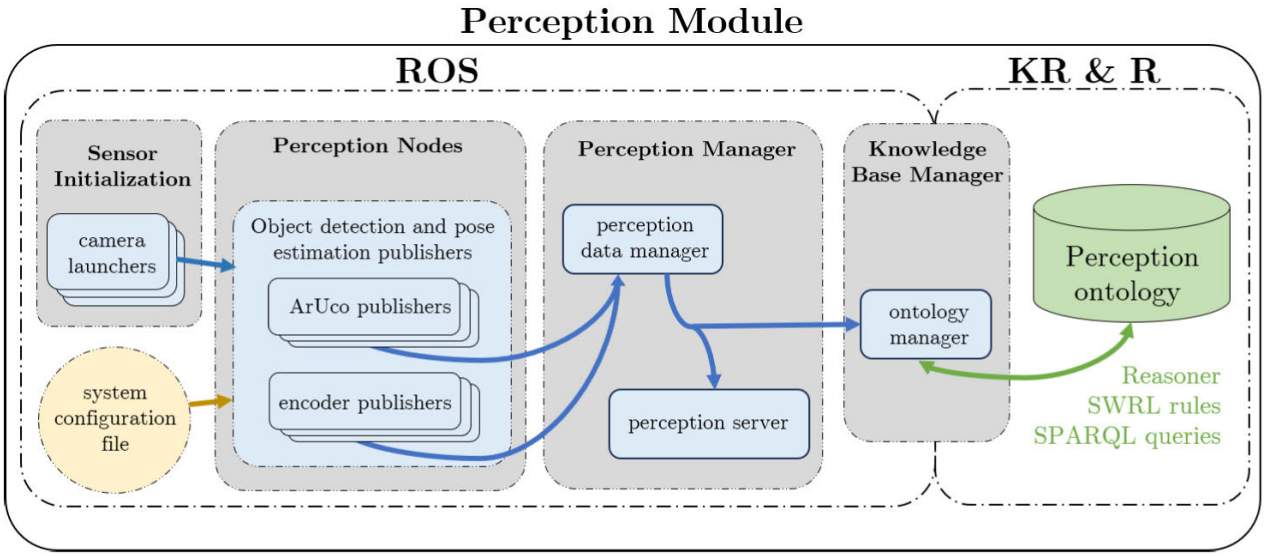


FIGURE 5. Proposed architecture of the perception module. Each gray box is a separated subsystem on the module, where the blue boxes correspond to different ROS nodes in charge of implementing the desired functionalities within the block. The blue arrows, connecting the blocks correspond to the interchange of information between the nodes via ROS topics. Note that the "knowledge base manager" block lies between all the ROS infrastructure and the KR & R section as a link between the two.

knowlege-base manager block, as well as an overview of the proposed perception ontology.

A. PERCEPTION NODES

The Perception Nodes are responsible for object detection and pose estimation. The configuration and initiation of the various publishing nodes are set by a configuration file in consideration of the available sensors in the system. The configuration file is responsible for setting various initialization aspects, for example, the launching of the ArUco-based publisher node, the encoder-based publisher node or both, for each camera; and it is also in charge of configuring each of the perception publishing node characteristics like the sizes of ArUco markers to be recognized by the system or the learning models to be loaded, among others. This, altogether, provides a flexible system for managing various cameras and detection mechanisms.

When the system activates ArUco detection for a certain camera, it recognizes ArUco markers within the environment and discerns their spatial location and orientation in relation to both the camera and the global coordinate frame. If a detected marker is registered in a user-defined ArUco object database, additional data regarding the detected object can be published, like the object's name, category, or the pose of the reference frame of the object with respect to both the camera and the global reference frames.

On the other hand, the encoder-based method consists of two main components: the unit for object detection, and the unit for object pose estimation. The object detection unit uses the Mask R-CNN model to return detected classes, confidence scores, bounding boxes, and segmentation masks for all detected objects. The Detectron2 library is leveraged to implement the object detection. It is responsible for automatically configuring the detection model, its weights, and the detection thresholds set by the user.

The pose estimation unit uses an Augmented Autoencoder architecture [2] to estimate the 6D pose of each detected object within the bounding boxes produced from the detection stage. The orientation is estimated by feeding the bounding box crop into a trained encoder to convert it into a lower-dimensional representation known as latent space. Then, by finding the closest match to a previously generated codebook of latent spaces, the orientation can be estimated. Note that the codebook is generated offline using the 3D model of an object and associates each latent space to a known orientation. Finally, the translation is obtained using the pinhole camera model. When the system activates an encoder based detection for a certain camera, its node is also equipped with mechanisms to estimate the color and possible occlusions between objects. The TensorFlow[12] library is used to implement the encoder model.

B. THE PERCEPTION MANAGER

The perception module is powered by a central processing unit that synchronizes and processes data from all active sensors in the robotic system. This central node collects, filters, and integrates sensor data into a unified information stream that is published as a custom ROS message type on a single ROS topic. This is done by automatically subscribing to sensor perception data topics (either ArUco or encoder, or both) for each active camera. Then data consistency and synchronization is ensured using a flagging system. This mechanism allows the publication of full perception data only when all expected sensor data has been successfully received, and it contains the most recent data from all sensors.

[12]https://www.tensorflow.org/

The data treatment process includes tasks such as renaming and assigning a unique identifier to every object in the system, updating the number of cameras that detect a specific object, and updating the IDs of occluded objects. It particularly focuses on fusing pose data from different sensors, especially in those cases when multiple cameras detect the same object. The system iterates over all cameras and their respective object detections. For each new processed object, it is checked if its position and orientation match a previously processed object within set thresholds. If a match is identified within these tolerances, the position and orientation data are merged with existing data in the dictionary, averaging the positions and using spherical linear interpolation (SLERP) for orientations. If no match is found within the thresholds, the object is considered a new detection.

Finally, a separate node from this block has access to the latest processed perception data and is responsible for broadcasting all object transformations for visualization purposes and provide data retrieving functionalities in the form of ROS services.

The rate at which the perception messages are published is controlled to ensure the system has enough time to process the data. Currently, a conservative estimation of one second has been chosen.

C. THE KNOWLEDGE BASE MANAGER

The system features a unit that acts as a bridge between the ROS modules and the ontology-based knowledge management system. This node also retrieves the most recent perception information from the system and asserts the data onto the ontology. This involves creating necessary individuals and associating them with appropriate data and object properties.

The ontology management unit safeguards the original ontology's integrity by operating on a temporary copy during initialization. When shutting down, the temporary ontology is erased. However, the system provides an option to record the active ontology's state in a separate file for future reference or troubleshooting.

This unit is also responsible for establishing user-defined SWRL rules to extend the system's inferencing abilities or query the information using predefined SPARQL queries. The SWRL rules include spatial relationships and region properties formalization, which provide the system with ways to derive objects relative positions to each other or to predefined symbolic regions in the environment. For example, the SWRL rule depicted in Figure 6 determines if one object is on top of another in a 3D space illustrating how SWRL rules can infer spatial relationships. On the other hand, the SPARQL query, shown in Figure 7, is another powerful example on how to extract data about the inferred relationships in the system. It uses the ''isOnTopOf'' relationship to retrieve data about all the objects involved in this spatial relationship.

```
DetectedObject(?o1), DetectedObject(?o2),
hasPose(?o1,?p1), hasPose(?o2,?p2),
positionX(?p1,?x1), positionY(?p1,?y1),
positionZ(?p1,?z1), positionX(?p2,?x2),
positionY(?p2,?y2), positionZ(?p2,?z2),
subtract(?dx,?x1, ?x2), pow(?dx_sqr,?dx,2),
subtract(?dy,?y1,?y2), pow(?dy_sqr,?dy,2),
add(?dist_sqr_xy,?dx_sqr,?dy_sqr),
divide(?sqroot,1,2),
pow(?dist_xy,?dist_sqr_xy,?sqroot),
greaterThan(?z1,?z2),
lessThan(?dist_xy,0.1)->isOnTopOf(?o1,?o2)
```

FIGURE 6. SWRL rule to determine the "isOnTopOf" spatial relationship between two objects in a 3D environment.

```
SELECT ?object1 ?relationship ?object2
WHERE {
    {
      ?object1 onto:isOnTopOf ?object2 .
      BIND("isOnTopOf" AS ?relationship)
    }
}
```

FIGURE 7. SPARQL query to retrieve all "isOnTopOf" spatial relationships in the system.

D. THE PERCEPTION ONTOLOGY

As previously introduced, the ''smart perception'' ontology is built upon the Autonomous Robotics Ontology (AuR), incorporating the required aspects in the current perception domain, such as pose management and object qualities. The graphical structure of classes and properties is shown in Figure 8. The various classes added to organize and categorize entities within the domain branch from the AuR `PhysicalObject`, `PhysicalAttribute` and `SpaceRegion` classes.

Regarding the object properties defined, they are categorized into perception-related properties like, for instance, `perceives`/ `isPerceivedBy`, `hasPose`/ `isPoseOf`, `isOccludedBy`/ `isOccluding`, etc.; spatial relationship properties like `isOnTopOf`/ `isBelowOf`, `isInFrontOf`/`isBehindOf`, etc.; and properties related to symbolic regions like `isAtSymbolicRegion`/ `isSymbolicRegionOf`.

On the other hand, data properties assign specific data values or attributes to the individuals of a class. Some examples include attributes like `arucoID`, `cameraType`, `detectionConfidence`, `encoderObjectClass`, `encoderObjectID`, `positionX`, `positionY`, and `positionZ`, `orientationW`, `orientationX`, `orientationY`, and `orientationZ`, `timeStamp`, `regionID`, etc.

V. RESULTS

In this section, the experimental results are presented, including details on the experimental setup like the workspace, hardware and software used, as well as the steps for setting up the whole system.

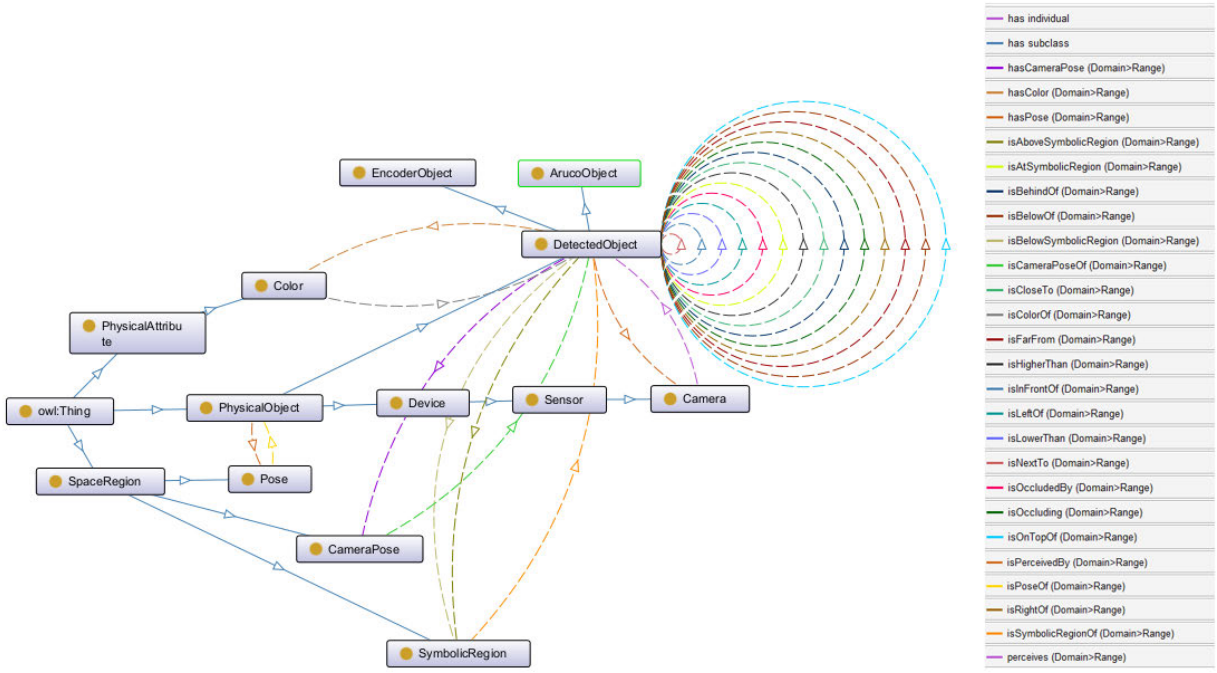


FIGURE 8. Full diagram of the smart perception ontology classes and object properties visualized in Protégé editor.

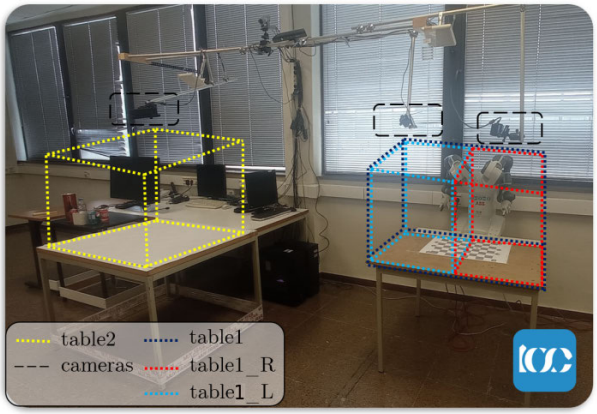


FIGURE 9. IOC manipulation workspace, including the camera infrastructure as well as the symbolic regions considered.

A. EXPERIMENTAL SETUP

The experiments were conducted at the Institute of Industrial and Control Engineering (IOC) robotics lab. As shown in Figure 9 the lab hosts two separate table-stations for manipulation tasks, each equipped with a camera infrastructure for perception. The perception system currently consists of three cameras: two Microsoft Kinect v2 and one Intel RealSense SR300, and up to four differentiated symbolic regions are considered. Additionally, all deep learning and inference operations are executed on a GeForce RTX 3080 Ti GPU, which delivers the computational power necessary to process complex learning models, thus ensuring the efficiency and effectiveness of the module.

To effectively deploy the module in real-world scenarios, a setup stage is required. For marker-based perception, specific objects are chosen and tagged with markers. Concurrently, a database is created, allowing the association of each distinctive marker ID to a known object. This can also be used to manually establish a transformation between the ArUco marker's placement on an object and another point of interest, such as where the object should optimally be grasped. On the other hand, for learning-based perception, both the 2D detector and autoencoder are trained using the desired image data and 3D models respectively, to generate the desired deployment models. For instance, simulation scenarios are tested with the YCB[13] benchmarking dataset, but due to the lack of exact object matches on the laboratory, real-world applications are tested using a can. Finally, the desired perception module configuration is meticulously set in the configuration file aforementioned. Moreover, the various symbolic regions of interest are measured and cataloged in a system-usable file. To enhance the system's reasoning skills, SWRL rules and SPARQL queries are also set in dedicated files, ready to be used and interpreted by the system when required.

B. EXPERIMENTAL VALIDATION

Upon activating the perception module, the object and pose detection systems are launched and processed to convert the raw camera sensor data into a unique source of processed data accessible via a single ROS topic. The bash terminal can be used to visualize the results of all the data packed in the topic.

[13]https://www.ycbbenchmarks.com/

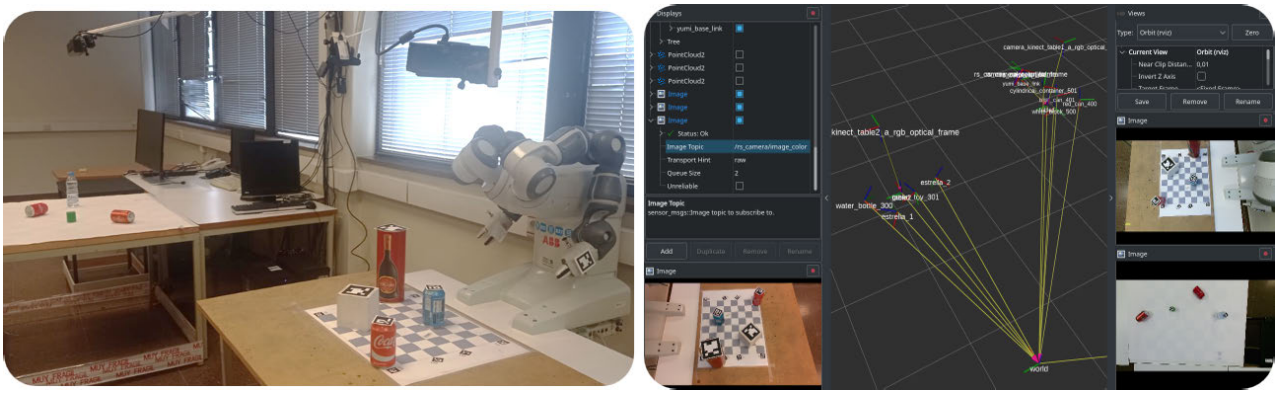


FIGURE 10. On the left, a sample test set up the IOC robotics lab manipulation section to detect objects using both ArUco-based and encoder-based pipelines. On the right, a sample of the processed environment visualized on RViz.

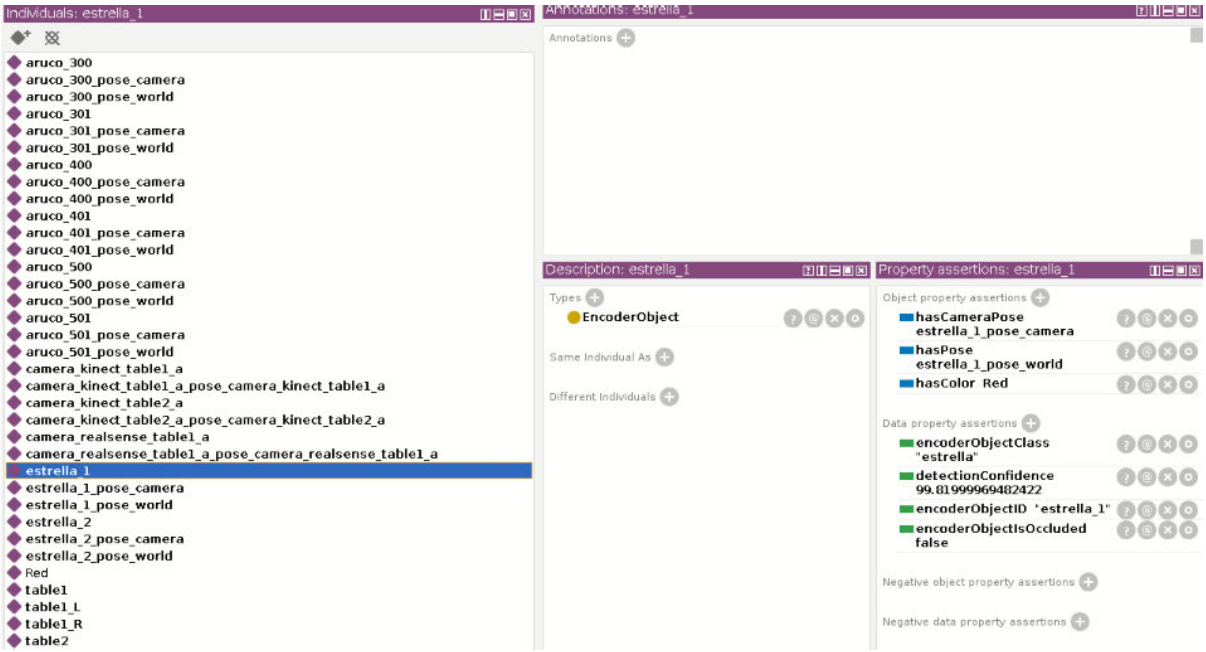


FIGURE 11. Instances asserted on the ontology, creating a knowledge base, visualized in Protégé.

Within the ROS framework, the perception module offers various functionalities as ROS services, specially to retrieve potential information of interest like the objects detected by a specific camera or specific object properties. Figure 10 show a possible object setup on the environment as well as the graphical visualization of pose data in RViz.

The ability to reason is a key component of ontology integration, as it allows the system to deduce information that goes beyond the raw perceptual data. For this reason, the module fuses the information acquired through the perception module into an ontology by collecting the processed data and asserting it, either automatically or on demand. This creates a knowledge base about objects in the environment and their characteristics. Figure 11 shows a sample of the assertion of instances on the ontology. Finally, the perception module can be used to generate new information about the environment, such as existing spatial relationship or symbolic regions of objects, as well as retrieve the reasoned information through queries.

This process illustrates the effective functioning and interaction of the perception module with its components and the environment, from reading raw sensor data to retrieve new information, confirming its potential to enhance robotic manipulation tasks in real-world scenarios.

C. DISCUSSION

The implementation of a smart perception module has shown its potential in advancing situation awareness in robotic manipulation tasks. The integration of the perception using techniques for object detection and pose estimation, and the inclusion of an ontology-based approach, has successfully enhanced the understanding of the environment of the robot. The inclusion of the ontology-based layer allows having awareness of the environment from the geometric knowledge

of each particular object. The smart perception module structures this raw perception data into an organized and interconnected form, by following the ontology. Awareness is thus achieved by allowing the system to easily navigate and infer new relations between the knowledge of each individual object.

For example, the system revealed insights about the environment that were not immediately apparent even for a human, specially those regarding relative poses between objects or their location in predefined symbolic regions. Everything by employing the combination of SWRL rules, SPARQL queries, and the Pellet reasoner.

Through the use of SWRL rules, the module was able to derive specific relationships between entities, enabling the system to better understand varied scenarios. For instance, it could determine if one object was above another, or if they were adjacent, based on the data provided by the perception system. This level of spatial awareness is essential for tasks requiring delicate manipulations or spatial reasoning. Furthermore, with SPARQL queries, the system could retrieve, filter, and present data from the ontology, allowing the robotic system to get a more refined and targeted view of its environment on demand. The Pellet reasoner was indispensable in this process. As a robust ontology reasoner, it ensured the consistency of the knowledge base, validated inferred relations, and proved instrumental in deriving new relationships from the provided data.

It is worth mentioning that the results achieved were consistent and accurate, which underscores the module's robustness and readiness for deployment in various real-world scenarios. The collaboration between the proposed perception techniques and advanced reasoning tools has set a new path for what is possible in the domain of robotic cognition. Situation awareness, as reflected through the perception module ability to identify objects, estimate their pose, and make smart reasoning about the environment, forms the foundation of effective and autonomous robotic manipulation enabling robots to reason about its environment, and potentially react to changes or plan its actions accordingly.

VI. CONCLUSION

This paper highlights the advancements made in robotic situation awareness through the design and deployment of a smart perception module. Its key success is the integration of object detection, pose estimation, and ontology-based reasoning, to enrich perception and cognition in robotic systems.

A notable strength of this module lies in its ability to adeptly merge learning-based and marker-based methods for both object detection and pose estimation. This duality ensures its versatility and adaptability across varied contexts. Benefiting immensely from the modularity and interoperability of ROS, the presented perception module can be easily integrated in the proposed general robotic framework BE-AWARE [31]. The union of the perception module with an ontology-based approach has opened the capabilities of using ontologies not only in terms of perception but wider robot reasoning aptitudes such as planning and execution. Future work will test the integration of the module within the framework working on complex manipulation scenarios.

Technological improvements on the object detection accuracy of the existing perception methods could be implemented into the smart perception module, enhancing the performance. While the present focus was on spatial relationships and symbolic regions for situation awareness, the true potential extends beyond these parameters. An evolved situation awareness could include dynamic environmental factors, understanding patterns of change over time, and recognizing common system conditions or constraints. This would allow robots to not only recognize their immediate surroundings but also to anticipate potential changes, hazards, or opportunities in their environment. Also, social and even self-awareness skills would equip robots to interpret human behaviors, and self-evaluate their own operational states for a better task execution.

REFERENCES

[1] I. Kotseruba and J. K. Tsotsos, "40 years of cognitive architectures: Core cognitive abilities and practical applications," *Artif. Intell. Rev.*, vol. 53, no. 1, pp. 17–94, Jan. 2020, doi: 10.1007/s10462-018-9646-y.

[2] M. Sundermeyer, Z.-C. Marton, M. Durner, M. Brucker, and R. Triebel, "Implicit 3D orientation learning for 6D object detection from RGB images," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2018, pp. 1–17.

[3] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," 2017, *arXiv:1703.06870*.

[4] Y. Bukschat and M. Vetter, "EfficientPose: An efficient, accurate and scalable end-to-end 6D multi object pose estimation approach," 2020, *arXiv:2011.04307*.

[5] Y. Labbe, J. Carpentier, M. Aubry, and J. Sivic, "CosyPose: Consistent multi-view multi-object 6D pose estimation," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2020, pp. 574–591.

[6] Y. Xiang, T. Schmidt, V. Narayanan, and D. Fox, "PoseCNN: A convolutional neural network for 6D object pose estimation in cluttered scenes," in *Proc. Robot., Sci. Syst.*, Pittsburgh, PA, USA, Jun. 2018. [Online]. Available: https://www.roboticsproceedings.org/rss14/index.html, doi: 10.15607/RSS.2018.XIV.019.

[7] J. Tremblay, T. To, B. Sundaralingam, Y. Xiang, D. Fox, and S. Birchfield, "Deep object pose estimation for semantic robotic grasping of household objects," in *Proc. Conf. Robot Learn.*, A. Billard, A. Dragan, J. Peters, and J. Morimoto, Eds., 2018, pp. 306–316. [Online]. Available: https://proceedings.mlr.press/v87/tremblay18a.html

[8] T. R. Gruber, "A translation approach to portable ontology specifications," *Knowl. Acquisition*, vol. 5, no. 2, pp. 199–220, Jun. 1993. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1042814383710083

[9] R. Sakagami, F. S. Lay, A. Dömel, M. J. Schuster, A. Albu-Schäffer, and F. Stulp, "Robotic world models—Conceptualization, review, and engineering best practices," *Frontiers Robot. AI*, vol. 10, Nov. 2023, Art. no. 1253049, doi: 10.3389/frobt.2023.1253049.

[10] *IEEE Standard for Autonomous Robotics (AuR) Ontology*, Standard 1872.2-2021, 2022, pp. 1–49.

[11] R. Bernardo, J. M. C. Sousa, and P. J. S. Gonçalves, "A novel framework to improve motion planning of robotic systems through semantic knowledge-based reasoning," *Comput. Ind. Eng.*, vol. 182, Aug. 2023, Art. no. 109345. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0360835223003698

[12] R. Bernardo, J. Sousa, and P. Gonçalves, "Ontological framework to improve motion planning of manipulative agents through semantic knowledge-based reasoning," in *Proc. RobOntics Workshop Ontologies Auto. Robot.*, Seoul, South Korea, Aug. 2023, pp. 1–10.

[13] M. Beetz, D. Beßler, A. Haidu, M. Pomarlan, A. K. Bozcuoglu, and G. Bartels, "Know rob 2.0—A 2nd generation knowledge processing framework for cognition-enabled robotic agents," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2018, pp. 512–519.

[14] M. Diab, M. Pomarlan, D. Beßler, A. Akbari, J. Rosell, J. Bateman, and M. Beetz, "SkillMaN—A skill-based robotic manipulation framework based on perception and reasoning," *Robot. Auto. Syst.*, vol. 134, Dec. 2020, Art. no. 103653.
[15] Y. Zhao, A. Sarkar, L. Elmhadhbi, M. H. Karray, P. Fillatreau, and B. Archimède, "An ontology of 3D environment where a simulated manipulation task takes place (ENVON)," *Semantic Web*, pp. 1–28, Dec. 2023. [Online]. Available: https://content.iospress.com/articles/semantic-web/sw233460
[16] C. Hernández, J. Bermejo-Alonso, and R. Sanz, "A self-adaptation framework based on functional knowledge for augmented autonomy in robots," *Integr. Comput.-Aided Eng.*, vol. 25, no. 2, pp. 157–172, Mar. 2018.
[17] X. Li, S. Bilbao, T. Martín-Wanton, J. Bastos, and J. Rodriguez, "SWARMs ontology: A common information model for the cooperation of underwater robots," *Sensors*, vol. 17, no. 3, p. 569, Mar. 2017. [Online]. Available: https://www.mdpi.com/1424-8220/17/3/569
[18] M. Waibel, M. Beetz, J. Civera, R. D'Andrea, J. Elfring, D. Gálvez-López, K. Häussermann, R. Janssen, J. M. M. Montiel, A. Perzylo, B. Schießle, M. Tenorth, O. Zweigle, and R. V. De Molengraft, "RoboEarth," *IEEE Robot. Autom. Mag.*, vol. 18, no. 2, pp. 69–82, Jun. 2011.
[19] J. Crespo, R. Barber, O. M. Mozos, D. BeBler, and M. Beetz, "Reasoning systems for semantic navigation in mobile robots," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2018, pp. 5654–5659.
[20] X. Sun, Y. Zhang, and J. Chen, "RTPO: A domain knowledge base for robot task planning," *Electronics*, vol. 8, no. 10, p. 1105, Oct. 2019. [Online]. Available: https://www.mdpi.com/2079-9292/8/10/1105
[21] A. Olivares-Alarcos, S. Foix, S. Borgo, and G. Alenyà, "OCRA—an ontology for collaborative robotics and adaptation," *Comput. Ind.*, vol. 138, Jun. 2022, Art. no. 103627. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0166361522000227
[22] G. H. Lim, I. H. Suh, and H. Suh, "Ontology-based unified robot knowledge for service robots in indoor environments," *IEEE Trans. Syst., Man, Cybern. A, Syst. Humans*, vol. 41, no. 3, pp. 492–509, May 2011.
[23] S. Lemaignan, R. Ros, L. Mösenlechner, R. Alami, and M. Beetz, "ORO, a knowledge management platform for cognitive architectures in robotics," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2010, pp. 3548–3553.
[24] X. Sun, Y. Zhang, and J. Chen, "High-level smart decision making of a robot based on ontology in a search and rescue scenario," *Future Internet*, vol. 11, no. 11, p. 230, Oct. 2019. [Online]. Available: https://www.mdpi.com/1999-5903/11/11/230
[25] M. Stenmark and J. Malec, "Knowledge-based instruction of manipulation tasks for industrial robotics," *Robot. Comput.-Integr. Manuf.*, vol. 33, pp. 56–67, Jun. 2015. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S073658451400060X
[26] T. Hoebert, W. Lepuschitz, M. Vincze, and M. Merdan, "Knowledge-driven framework for industrial robotic systems," *J. Intell. Manuf.*, vol. 34, no. 2, pp. 771–788, Feb. 2023.
[27] M. Merdan, T. Hoebert, E. List, and W. Lepuschitz, "Knowledge-based cyber-physical systems for assembly automation," *Prod. Manuf. Res.*, vol. 7, no. 1, pp. 223–254, Jan. 2019, doi: 10.1080/21693277.2019.1618746.
[28] M. Beetz, G. Kazhoyan, and D. Vernon, "The CRAM cognitive architecture for robot manipulation in everyday activities," 2023, *arXiv:2304.14119*.
[29] M. Beetz, F. Bálint-Benczédi, N. Blodow, D. Nyga, T. Wiedemeyer, and Z.-C. Márton, "RoboSherlock: Unstructured information processing for robot perception," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2015, pp. 1549–1556.
[30] M. Diab, A. Akbari, M. Ud Din, and J. Rosell, "PMK—A knowledge processing framework for autonomous robotics perception and manipulation," *Sensors*, vol. 19, no. 5, p. 1166, Mar. 2019. [Online]. Available: https://www.mdpi.com/1424-8220/19/5/1166
[31] O. Ruiz-Celada, A. Dalmases, R. Suárez, and J. Rosell, "BE-AWARE: An ontology-based adaptive robotic manipulation framework," in *Proc. IEEE 28th Int. Conf. Emerg. Technol. Factory Autom. (ETFA)*, Sep. 2023, pp. 1–4.

ORIOL RUIZ-CELADA received the double master's degree in industrial engineering and automatic control and robotics from Barcelona School of Industrial Engineering, Universitat Politècnica de Catalunya (UPC), in 2022. He is currently pursuing the Ph.D. degree in automatic control, robotics, and computer vision with UPC, working on the development of an ontology-based adaptive manipulation execution framework for multiple robots. His final thesis explored the inclusion of behavior trees in autonomous robot manipulator systems, with a focus on reacting to changes in the environment and performing tasks and motion replanning.

ALBERT DALMASES received the master's degree in automatic control and robotics from Barcelona School of Industrial Engineering, Universitat Politècnica de Catalunya (UPC), in 2023. His final thesis explored the inclusion of ontologies for the development of smart perception for robotic manipulation. He has worked on several robotic projects related to deep learning.

ISIAH ZAPLANA received the bachelor's and master's degrees in pure and applied mathematics in 2013 and the Ph.D. degree in applied mathematics and robotics from Universitat Politècnica de Catalunya (UPC), in 2018. He was a Postdoctoral Researcher with the Advanced Robotics Department, Italian Institute of Technology, from 2019 to 2020, and the Department of Mechanical Engineering, University of Leuven, from 2021 to 2022, where he was involved in the development of strategies combining vision, artificial intelligence, and robotics for the optimal performance of industrial robotic systems in the execution of different manipulation tasks. He rejoined UPC as a Maria Zambrano Postdoctoral Fellow, in 2022. From 2023, he is an Assistant Professor of robotics and computer vision.

JAN ROSELL received the B.S. degree in telecommunication engineering and the Ph.D. degree in advanced automation and robotics from Universitat Politècnica de Catalunya (UPC), Barcelona, Spain, in 1989 and 1998, respectively. He joined the Institute of Industrial and Control Engineering (IOC), in 1992, where he developed research activities in robotics. He has been involved in teaching activities in automatic control and robotics as an Assistant Professor, since 1996, and an Associate Professor, since 2001. His current technical areas include task and motion planning, mobile manipulation, and robot co-workers.