Primitive Action Recognition based on Semantic Facts

Adrien Vigné, Guillaume Sarthou, Aurélie Clodic

Primitive Action Recognition based on Semantic Facts

Adrien Vigné[1 [0009−0003−0958−2596]], Guillaume Sarthou[2 [0000−0002−4438−2763]], and Aurélie Clodic[1 [0009−0009−6484−8143]]

[1] LAAS-CNRS, Université de Toulouse, CNRS, INSA, Toulouse, France
firstname.surname@laas.fr
[2] IRIT, Université de Toulouse, CNRS, Toulouse, France
firstname.surname@irit.fr

Abstract. To interact with humans, a robot has to know actions done by each agent presents in the environment, robotic or not. Robots are not omniscient and can't perceive every actions made but, as humans do, we can equip the robot with the ability to infer what happens from the perceived effects of these actions on the environment.
In this paper, we present a lightweight and open-source framework to recognise primitive actions and their parameters. Based on a semantic abstraction of changes in the environment, it allows to recognise unperceived actions. In addition, thanks to its integration into a cognitive robotic architecture implementing perspective-taking and theory of mind, the presented framework is able to estimate the actions recognised by the agent interacting with the robot. These recognition processes are refined on the fly based on the current observations. Tests on real robots demonstrate the framework's usability in interactive contexts.

Keywords: Action Recognition · Human-Robot Interaction · Cognitive Robotics.

1 Introduction

Where robots have been restricted for a while at performing complex tasks on their own in an autonomous way, or in coordination with other robotic agents, the field of Human-Robot Interaction brings the new challenge of robots performing shared tasks with humans. In light of the definition of joint action, this means that robots should be able to interact with humans and coordinate their actions in space and time to bring about a change in the environment [18]. Cooperation and collaboration tend to be key features to make robots more adaptative and thus flexible with respect to humans' actions.

As a prerequisite to joint action, Tomasello in [21] emphasized intentional action understanding, meaning that an agent should be able to read its partner's intentions. In this way, when observing a partner's action or course of actions, the agent should be able to infer its partner's intention in terms of goal and plan to achieve the goal. Where in a shared task one can assume a shared goal to

Fig. 1. A shared task example where the robot, performing its own part of the task, cannot monitor the human activity.

exist, a shared plan can only be estimated by both partners. As a consequence, during the entire realisation of a shared task, agents should continue to monitor others' actions to be able to adapt and coordinate their own actions.

Considering a shared task like cooking, when performing its own actions, a robot has to perceive the elements it has to interact with. Wanting to grasp a knife, the robot needs to look at the knife to estimate its position. However, when focused on such elements, monitoring others' actions can become unfeasible. Even having multiple visual sensors, we cannot assume that the human will perform its part of the plan in front of the robot, as illustrated in Fig. 1. In the same way, we cannot assume to act in a fully instrumented area allowing the robot to be omniscient. In such realistic applications, the need to detect human actions with as little visual information as possible is mandatory.

In the following, we will make the distinction between action and task considering a hierarchic task decomposition point of view. This means that a task can be decomposed into a set of sub-tasks and actions, where each sub-task can also be decomposed in such a way. We consider as actions the leaves of the resulting decomposition, meaning actions that can be directly executed by a robotic agent (i.e. pick, place, release, etc). Reversing this assumption, a human task can be monitored through the detection of the underlying human actions. Task recognition is out of our current scope as requiring as a first step the recognition of actions.

In this paper, we present a lightweight method for action recognition based on a semantic knowledge flow. This knowledge is obtained through the use of the DACOBOT robotic architecture [16]. The main contribution of this work is the possibility to detect actions through the changes they brought to the environment. Such a contribution allows to pass over the general assumption of constant monitoring of the humans using visual sensors. The side contribution of this work, more related to the context of Human-Robot Interaction, is the ability to estimate the actions perceived by each agent it interacts with thanks to perspective-taking.

In Sec. 2, we discuss related work and how action recognition is generally performed. A detailed explanation of the approach is then provided in Sec. 3 before

providing an overview of the knowledge flow in which it has been integrated in Sec. 4 and its application in Human-Robot Interaction in Sec. 5. Finally, Sec. 6 presents results on a dataset and Sec. 7 concludes the paper.

2 Related Work

Action recognition takes its application in various fields [2] such as health care, sports analysis, and robotics. It is used, for example, to monitor patients in healthcare in order to detect falls [19], or to anticipate human action for autonomous driving vehicles [7]. In robotics applications, action recognition is intensively used to learn tasks from video demonstrations [22]. In the field of Human-Robot Interaction, action recognition has become an important topic as detailed in [6], with applications such as gestures learning [25] or risk evaluation for decision making [26].

To date, two approaches coexist to recognise human actions: data-driven and knowledge-based. While data-driven approaches aim to directly deal with sensor data such as images, knowledge-based approaches rather focus on the analysis of semantic data either stated or extracted beforehand.

Data-driven approaches were initially based on 2D images with the use of pattern matching [1] or support vector machine [17]. The use of deep neural networks has then allowed the generation of more robust recognitions [24] but with the initial assumption of no occlusion. This concern has been later addressed in [23] to deal with real-world scenarios and thus environments like offices with desks and chairs. For finer estimations of the humans poses and thus more precise recognition, similar approaches but using 3D point clouds have been proposed [9].

The data-driven approaches also provide solutions to the problem of recognising human actions when the robot cannot perceive directly the human activity. A combination of RF-based (Radio Frequency) and vision-based detection has been used in [8] where the RF part can provide information when it is impossible for the vision. Other solutions aim at equipping the environment itself instead of the robot with multiple sensors like cameras [5] to provide the greatest vision and thus always keep track of the human's body. The main inconvenience of such solutions is the use of dedicated environments or specific robot hardware.

With regard to all the presented data-driven approaches, a general concern is that they mostly recognise humans' activities (i.e. drinking, sleeping, eating or humans' gestures) rather than primitive actions. In addition, as these approaches focus on the human body, the track of objects is not considered. Nevertheless, for human monitoring in a joint task, one would rather need low-level actions recognition (to maybe recognise higher level tasks on top of it) such as picking or placing and a track of the objects involved in the task.

On the other hand, knowledge-based approaches rely on data already processed by the robot in order to abstract its environment. All these data are thus centrally stored and formalised. One such formalism is ontology which can be formalised thanks to the Ontology Web Language (OWL). Riboni et al. in [11], explain that the human action recognition can be handled by an ontology-based

approach with a result at the same level as the better data-driven algorithm. Nevertheless, they also specify that the ontology-based approach needs a way to have a time representation to reach this level of result. Thanks to [10], this time representation can be solved. In this work, they define a temporal Web Ontology Language (tOWL) as an extension to OWL with which it is possible to have a time representation of actions or events in the ontology. This enables to recognise actions thanks to ontology reasoning. However, it does not manage knowledge uncertainty or noise in the perception. To solve this, Rodriguez et al. in [12] propose to use a fuzzy ontology described and formalized in [20]. Thanks to their model and the use of a fuzzy ontology, Rodriguez et al. solve the problem of uncertainty and time representation, but their system does not detect low-level actions.

Finally, at the intersection of data-driven approaches and knowledge-based approaches, some hybrid approaches have been proposed [4, 3]. In such works, the data-driven part is used to recognise the low-level activities while the knowledge-base part is used the recognise higher-level activities, based on the low-level actions. While still demonstrating the usability of knowledge-based methods, the need to continuously observe humans still exists.

3 Approach and recognition

Let's consider an example of a robot and a human working together to prepare a meal in a kitchen. If we observe someone holding a fork, it must have been picked up somewhere. Similarly, if utensils appear on the workplace, someone must have placed them there. A human can infer which actions have caused these changes in the environment without seeing them, even if some parameters can remain unknown (e.g. who acted?).

This cognitive process allows the recognition of actions thanks to the observation of changes in the environment and also allows an estimation of the possible set of actions in a given situation [21]. For example, if we see Bob's hand approaching an apple on the workplace, we can estimate that Bob's next action will probably be related to the apple, but we cannot predict whether he will pick it up or push it. If we observe Bob grasping the apple, we can refine our estimation because the set of possible actions in this state is limited.

Taking inspiration from this human ability, we choose to represent actions as sequences of geometric changes in the environment. In this section, we thus present our method to recognise on-the-fly actions, based on symbolic facts.

3.1 A dynamic state machine to handle the recognition

To represent the recognition process introduced earlier, we have chosen State Machines (SM) where transition conditions represent the steps of the recognition process, i.e. the changes to be perceived. Thus, a pick action can be recognised by the following transitions of a SM: (1) the agent's hand approaches the object (2) the object is in the agent's hand (3) the object is no longer on its support.

These changes in the geometric situation of the environment can be abstracted using semantic facts, resulting in the following sequence:

1. $?A\ hasHandMovingToward\ ?O$
2. $?A\ isHolding\ ?O$
3. $NOT\ ?O\ isOnTopOf\ ?S$

To represent the unspecified entities involved in the sequence (i.e. the agent, the object, and the support), we use variables here represented by question marks followed by a literal. During a recognition process, these variables will be instantiated and will thus constrain the following facts of the sequence. For example, perceiving first Bob's hand approaching the object `o_1` meaning the fact `(bob hasHandMovingToward o_1)`, the variables A and O become instantiated and constrain the rest of the sequence. The next expected fact would thus be `(bob isHolding o_1)`.

Even if sequence representation is convenient, some facts could be unperceived by the robot. We propose a way to specify the minimal set of facts to be perceived to recognise an action with the use of the tag $REQUIRED$. The resulting description of an action is provided in List. 1.1.

Listing 1.1. Extract of the models file for a pick_over action

```
Pick_over:
  sequence:
    - ?A hasHandMovingToward ?O
    - ?A isHolding ?O
    - NOT ?O isOnTopOf ?S REQUIRED
```

As a consequence, our actions are no longer some purely linear sequences and could rather be transposed to state machines as illustrated in Fig. 2. We can see that the transition carrying the fact $?A\ isHolding\ ?O$ connects both states $s0$ and $s1$ with state $s2$. These links mean that the transition between states $s0$ and $s1$ is not necessary to recognise the action. Not perceiving that the agent's hand approaches the object but perceiving that the agent is holding the object is sufficient to reach state $s2$ and to start the recognition. Nevertheless, due to such a bypass, one could notice that triggering the transition from $s0$ to $s3$, variable A will never be instantiated resulting in missing parameters.

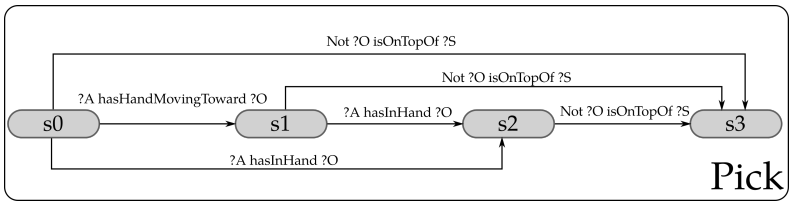

Fig. 2. State Machine for the detection of the pick action with only transition fact

3.2 Dynamically created state machines

In real-world situations in general and in human-robot interaction scenarios in particular, several agents can act simultaneously and even a single agent can do several actions at the same time, like picking two objects. An action recognition system must be able to handle the recognition of multiple actions in parallel that's why we designed **state machine factory**. When a semantic fact is submitted to a factory, if it allows to activate one of the transitions of the initial state, the factory will create an instance of the SM it is responsible for. Such a SM will be called an **active state machine**. The newly created SM will thus be in a different state than the initial one and some of its variables will already be instantiated.

When a new fact arrives in the recognition system, meaning a change in the environment has been perceived by the robot, this fact is first used to try to trigger a transition of all the active SMs. In the case the fact does not allow any of them to trigger any transition, then it is submitted to each factory to try to generate new SMs. Indeed, without this rule, multiple SMs recognizing the same action (in terms of instance) could exist at the same time. Nevertheless, several SMs coming from the same factory can exist simultaneously, that is to recognise the same action type performed by different agents simultaneously, or by the same agent on different objects.

When an active SM is finished, if all the variables used in the conditions of its transitions have been set, the SM is stated to be **complete**, otherwise, the SM is **incomplete**. Indeed, as not all transitions are required to recognise an action, some variables can stay unbounded.

Once a SM is finished, an action has been recognised. The finished SM is thus removed from the set of active SMs. In addition, as several SMs could have been created from the same semantic fact (based on the principle of progressive refinement when new facts arrive), all active SM involving facts used by the finished SM are also removed from the set of active SMs. The implicit hypothesis made here is that a fact can only be part of a single action performed by an agent.

4 Integration and Knowledge flow

In order to be fed with meaningful semantic facts representing the changes in the environment, our Action Recognition System has been integrated into the DACOBOT [16] robotic architecture. In this section, we present the knowledge flow illustrated in Fig. 3.

4.1 Geometrical Situation Assessment

In this architecture, the geometrical Situation Assessment is handled by the software Overworld [14]. This software can be connected to any perception system to perceive objects, humans, or areas. As the same entity can be perceived through several systems, Overworld is first able to aggregate the data from all the used

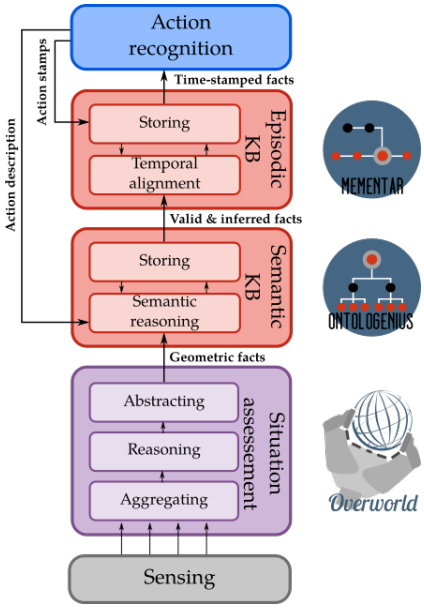

Fig. 3. Knowledge flow for Action Recognition System in the DACOBOT architecture

perception systems to create a unified 3D representation of the robot's environment. Thanks to geometrical reasoning based on the sensors' field of view, the entities' visual occlusions, and physics simulation, Overworld provides a coherent representation of the entire environment.

On the basis of the 3D representation, Overworld can then compute semantic facts. These facts can link objects together like with isOnTopOf or isInContainer. They can link objects or agents to areas with isInArea. They can also link agents to objects with facts such as hasInHand, isLookingAt, or hasHandMovingToward. These facts are computed at every update of the system and are output on a ROS topic. A fact is generated when it starts to be perceived (ADD) and when it stops (DEL).

An important feature of Overworld, essential for HRI, is its ability to estimate the perspective of other agents and their representation of the world. From there, in the same way it is done from the robot's perspective, Overworld computes and generates semantic facts from the others' perspective allowing the use of the theory of mind.

4.2 Semantic Knowledge Base

The architecture used considers as a central component a semantic knowledge base. This latter contains both common sense knowledge (general concepts like object types, colors, ...) and anchored knowledge related to the current situation. This knowledge can be accessed by every component of the architecture

allowing a unified and coherent representation among the entire architecture. This semantic knowledge base is managed by Ontologenius [15]. This software has been specially developed for robotic applications with good performances both on queries and dynamic updates. It is thus adapted to maintain the current state of the situation at a semantic level with online inferences resolutions. Regarding the knowledge stream, Ontologenius is directly connected to the output of Overworld. When new facts arrive in it, they are first analysed to verify their consistency regarding common sense knowledge, then once added to the knowledge, Ontologenius will reason on this knowledge in order to extract new facts. For example, from the fact `ADD (cup_1 isInTopOf table_3)`, we can infer thanks to inverse `ADD (table_3 isUnder cup_1)`.

As an output, Ontologenius sends on a ROS topic the validated facts as well as the inferred ones. However, as it does not deal with temporal aspects, the inferred facts cannot be stamped on the base of the used facts for the inference nor at the time of the inference. They are rather sent with an explanation about the facts involved in their inference.

Like Overworld, Ontologenius can maintain several knowledge bases in parallel, allowing theory of mind. Each output of Overworld (one per human agent in addition to the robot) is thus connected to a specific knowledge base.

4.3 Episodic Knowledge Base

As explained by Riboni et al. in [11], ontology-based action recognition is possible when linked to time representation. Regarding this temporal representation, the DACOBOT architecture proposes the software Mementar [13] as an episodic knowledge base. It is responsible for the organization of the semantic facts, provided by the ontology, on a temporal axis. While only the validated facts are already stamped, the inferred ones have to be aligned. To this end, Mementar finds the more recent fact among the ones used in the inference and aligns the inferred fact on this later. All the facts once correctly stamped are then republished on a ROS topic for the components (as the action recognition) needing continuous monitoring.

On the basis of this timeline, Mementar proposes a set of queries to retrieve past facts based on their timestamp, their order, or their semantics thanks to a link with the semantic knowledge base. In addition, Mementar allows to represent actions/tasks in the timeline with a start stamp and an end stamp. These actions can also be queried to retrieve the facts appearing during an action, the actions holding during an action, their stamps, or their type.

Finally, in the same way it has been done for the two previously presented components, Mementar can manage a timeline per agent allowing to manage theory of mind at a temporal level.

4.4 Action Recognition

The action recognition component described in this paper is connected to the output of Mementar where no distinction is made between the inferred facts and

the others. As illustrated in Fig. 3, as an output, the action recognition sends the description of the recognised actions to the semantic knowledge base and temporally marks them in the episodic knowledge base.

This description of the recognised actions at the semantic level allows us to link the actions to their parameters as a relation reification. An example of such a description is presented in List. 1.2. This description is stored in a description file and can reuse all the different variables used in the facts sequence linked to the action models. Here we reuse the variables A to provide the knowledge of who has acted. We also provide a way to symbolise the action itself with the specific variable ?.

Actions are thus described both at the semantic and episodic levels, each providing a different view of them and thus different ways to retrieve them. For example, to know the agent having performed a given action, one can query the semantic knowledge base. On the contrary, to know the facts that took place during a given action or to know when has started an action, one would rather query the episodic knowledge base.

5 Multi-human estimation and HRI

As described previously, all software used in the knowledge flow can manage in parallel multiple instances. This specificity provides multiple independent knowledge flows, one for each agent interacting with the robot in addition to the flow for the robot itself. Taking advantage of that, we can recognise actions from the knowledge flow of any available agent in order to estimate the actions they are aware of. In this way, the knowledge base of each agent can be updated independently which can lead to the generation of belief divergences.

To illustrate this divergence in beliefs, let's consider a robot and a human interacting together. The human temporarily leaves the room to pick up a tool. Meanwhile, the robot picks an object and places it in a drawer. When the human comes back, thanks to the actions recognition system, the robot can estimate that the human knows that it picked the object but can also estimate that he does not know that it placed the object in the drawer. Here a divergence in beliefs is raised between the knowledge bases of both agents.

Such piece of information could later be used by a decisional process, like a supervision component, to prevent future errors in the execution of a plan.

Listing 1.2. Description part of our model for the pick action

```
pick:
    description:
    - ?? isA PickAction
    - ?? isPerformedBy ?A
    - ?? isPerformedOn ?O
    - ?? isPerformedFrom ?S
```

In a similar way, actions with no visual effects on the environment, like scanning a bar code, can be estimated as unknown by the human partner and thus communication could be required to prevent a blockage in the execution of a plan.

6 Experimentations

To illustrate the possibilities offered by our action recognition system, we present here two scenarios tested on two different robots[3].

6.1 Scenario 1

In the first scenario, we use a Pr2 robot to pick a cube and to drop it in a box previously flipped by the human partner. Here we want to illustrate the recognition of the actions of the robot itself but also actions made by a human agent not perceived by the robot[4]. This case study thus demonstrates among others the recognition of incomplete action as the robot does not have access to all data needed to recognise all parameters of the action like who has performed the action.

In this scenario, our system has been able to recognise a pick and a place action of the robot but also a pick and a place action of an unknown agent. This illustrates the multiple recognition of actions even if some are incomplete and the capability to create and manage multiple SMs.

6.2 Scenario 2

In this second scenario, we use a Pepper as the robotic agent that is perceiving two human agents (a_1 and a_2) making some tabletop manipulation on cubes over boxes. Each human agent is equipped with a motion capture system to be perceived by the robot. The configuration of the scenario is represented in Fig. 4[5]. This scenario is decomposed into three parts.

In the first part of the scenario, each agent looks at the table, to initialise their knowledge base with the current state of the environment. After this initial step, one agent (a_2) turns around (Fig. 4a) and the other human agent (a_1) moves one cube. This later action is perceived by the robot and is also added to the estimated knowledge base of a_1 who has done the action. The pick is recognised between t0 and t2 for these two agents as it is presented in Fig. 5. Based on the estimation of the perspective of a_2, the action made by a_1 is not added to the knowledge base of a_2 as it could not be perceived by a_2. This part allows us to demonstrate the recognition of the actions from the point of view of different agents making a shared task.

[3] ROSbags: `https://gitlab.laas.fr/avigne/action_recognition_dataset`
[4] The agent is not perceived because it has not been equipped to do this.
[5] Video: `https://youtu.be/cwLLEAA_mCY`

Fig. 4. Representation of the situations used in scenario 2. At the left the situation where a_2 can't see the cubes. At the right the situation when a_2 turns around again to continue the task.

In the second part of the scenario, a_2 turns around again to see what has been done (Fig. 4b). With the estimation of his perspective, the robot now estimates that the agent has perceived that the cube has moved. This allows our system to recognise that a pick and a place action have been performed, from a_2 perspective, but with no additional information. Indeed, with the facts linked to this action (around t6), it's impossible from the point of view of a_2 to know who has done the action.

The last part of this scenario is a shared task between the two humans. They have to take at the same time one cube each and make a tower. In this part, we demonstrate the recognition of actions performed at the same moment on different objects and made by different agents. This simultaneous recognition is illustrated between the timestamp t11 and t12 in Fig. 5.

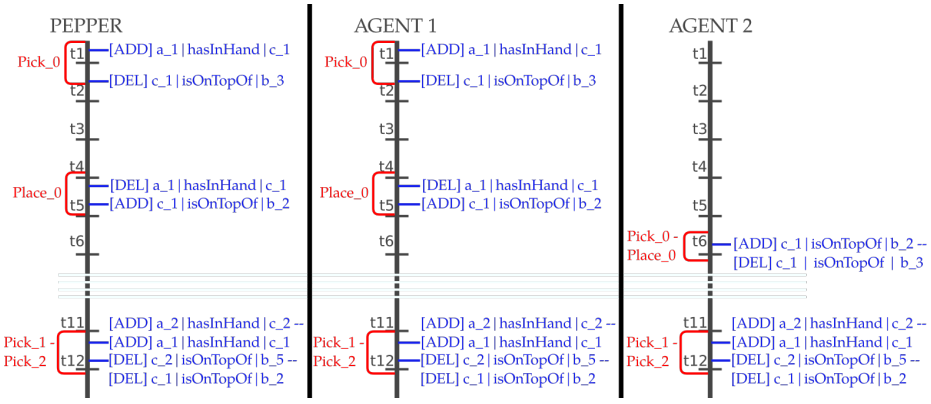

Fig. 5. Simplified view of the timelines maintained by Mementar for each agent of the scenario 2. Facts are represented at the right of the timeline and actions are at the left.

7 Conclusion and Future Work

In this paper, we present our Action Recognition system[6]. The recognition process uses state machines dynamically created and instantiated thanks to the semantic facts produced by the knowledge flow of the robotic architecture it has been integrated into. These state machines are easily configurable to be adapted to new actions. In addition, our system is also adapted to the HRI context thanks to the management of multiple knowledge flows in parallel relying on perspective-taking.

This system is a first step toward a larger system for task recognition based on Hierarchical Task Network (HTN), allowing us to validate and test all the requirements before handling this new challenge. Nevertheless, our action recognition system has some limitations that will have to be handled. The main limitation is due to the limited set of facts currently computed by the situation assessment. Indeed, we are aware that with the current set of facts, only pick and place can be detected. However, with the presented system, we can easily handle new sets of facts and thus describe and recognise new actions.

Another aspect that we want to develop would be a post-processing of detected actions to try to fulfil the incomplete actions and to remove false detection due to natural changes in the environment. Indeed, currently, an object falling on the ground would generate the recognition of a pick action and the presence of a single action at the place of action is not used to estimate who performed the action.

Acknowledgements

This work has been supported by the Artificial Intelligence for Human-Robot Interaction (AI4HRI) project ANR-20-IADJ-0006 and DISCUTER project ANR-21-ASIA-0005.

References

1. Aggarwal, J.K., Cai, Q., Liao, W., Sabata, B.: Nonrigid motion analysis: Articulated and elastic motion. Computer Vision and Image Understanding (1998)
2. Al-Faris, M., Chiverton, J., Ndzi, D., Ahmed, A.I.: A review on computer vision-based methods for human action recognition. Journal of imaging (2020)
3. Díaz-Rodríguez, N., Cadahía, O.L., Cuéllar, M.P., Lilius, J., Calvo-Flores, M.D.: Handling real-world context awareness, uncertainty and vagueness in real-time human activity tracking and recognition with a fuzzy ontology-based hybrid method. Sensors (2014)
4. Helaoui, R., Riboni, D., Stuckenschmidt, H.: A probabilistic ontological framework for the recognition of multilevel human activities. In: ACM international joint conference on Pervasive and ubiquitous computing (2013)
5. Iosifidis, A., Tefas, A., Pitas, I.: Multi-view human action recognition under occlusion based on fuzzy distances and neural networks. In: EUSIPCO. IEEE (2012)

[6] `https://github.com/vigne-laas/Procedural`

6. Ji, Y., Yang, Y., Shen, F., Shen, H.T., Li, X.: A survey of human action analysis in hri applications. Transactions on Circuits and Systems for Video Technology (2019)
7. Koppula, H.S., Saxena, A.: Anticipating human activities using object affordances for reactive robotic response. Transactions on pattern analysis and machine intelligence (2015)
8. Li, T., Fan, L., Zhao, M., Liu, Y., Katabi, D.: Making the invisible visible: Action recognition through walls and occlusions. In: ICCV (2019)
9. Li, W., Zhang, Z., Liu, Z.: Action recognition based on a bag of 3d points. In: computer society conference on computer vision and pattern recognition-workshops. IEEE (2010)
10. Milea, V., Frasincar, F., Kaymak, U.: towl: a temporal web ontology language. Transactions on Systems, Man, and Cybernetics (2011)
11. Riboni, D., Pareschi, L., Radaelli, L., Bettini, C.: Is ontology-based activity recognition really effective? In: PERCOM workshops. IEEE (2011)
12. Rodríguez, N.D., Cuéllar, M.P., Lilius, J., Calvo-Flores, M.D.: A fuzzy ontology for semantic modelling and recognition of human behaviour. Knowledge-Based Systems (2014)
13. Sarthou, G.: Mementar, `https://github.com/sarthou/mementar`
14. Sarthou, G.: Overworld: Assessing the geometry of the world for human-robot interaction. Robotics and Automation Letters (2023)
15. Sarthou, G., Clodic, A., Alami, R.: Ontologenius: A long-term semantic memory for robotic agents. In: RO-MAN. IEEE (2019)
16. Sarthou, G., Mayima, A., Buisan, G., Belhassein, K., Clodic, A.: The director task: a psychology-inspired task to assess cognitive and interactive robot architectures. In: RO-MAN. IEEE (2021)
17. Schuldt, C., Laptev, I., Caputo, B.: Recognizing human actions: a local svm approach. In: ICPR. IEEE (2004)
18. Sebanz, N., Bekkering, H., Knoblich, G.: Joint action: bodies and minds moving together. Trends in cognitive sciences (2006)
19. Sree, K.V., Jeyakumar, G.: A computer vision based fall detection technique for home surveillance. In: ICCVBIC. Springer (2020)
20. Tho, Q.T., Hui, S.C., Fong, A.C.M., Cao, T.H.: Automatic fuzzy ontology generation for semantic web. transactions on knowledge and data engineering (2006)
21. Tomasello, M., Carpenter, M., Call, J., Behne, T., Moll, H.: Understanding and sharing intentions: The origins of cultural cognition. Behavioral and brain sciences (2005)
22. Ullah, A., Ahmad, J., Muhammad, K., Sajjad, M., Baik, S.W.: Action recognition in video sequences using deep bi-directional lstm with cnn features. IEEE access (2017)
23. Weinland, D., Özuysal, M., Fua, P.: Making action recognition robust to occlusions and viewpoint changes. In: European Conference on Computer Vision (2010)
24. Weinland, D., Ronfard, R., Boyer, E.: A survey of vision-based methods for action representation, segmentation and recognition. Computer vision and image understanding (2011)
25. Yavşan, E., Uçar, A.: Gesture imitation and recognition using kinect sensor and extreme learning machines. Measurement (2016)
26. Zhang, H., Reardon, C., Han, F., Parker, L.E.: Srac: Self-reflective risk-aware artificial cognitive models for robot response to human activities. In: ICRA. IEEE (2016)