Initial Results with a Simulation Capable Robotics Cognitive Architecture

G. Trinidad Barnech[1*], G. Tejera[1], J. Valle-Lisboa[1], P. Núñez[2], P. Bachiller[2], and P. Bustos[2]

[1] Universidad de la República, Uruguay
∗ email: gtrinidad@fing.edu.uy
[2] Universidad de Extremadura, Spain

Abstract. In this paper, we present some conceptual and experimental results obtained from the integration of a Robotics Cognitive Architecture (RCA) with an embedded Physics simulator. The RCA used, CORTEX, is based on a highly efficient, distributed working memory (WM) called Deep State Representation (DSR). This WM already provides a basic ontology, state persistency, geometric and logical relationships among elements and tools to read, update and reason about its contents. The hypothesis that we want to explore here is that integrating a physics simulator into the architecture facilitates the enacting of a series of additional functionalities that, otherwise, would require extensive coding and debugging. Also, we characterize these functionalities in broad types according to the kind of problem they tackle, including occlusion, model-based perception, self-calibration, scene's structural stability and human activity interpretation. To show the results of these experiments, we use CoppeliaSim as the embedded simulator, and a Kinova Gen3 robotic arm as the real scenario. The simulator is kept synchronized with the stream of real events and, depending on the current task, several queries are computed, and the results projected to the working memory, where the participating agents can take advantage of them to improve the overall performance.

1 Introduction

According to G. Hesslow [16], the *simulation hypothesis* states that a simulated action can elicit perceptual activity that resembles the activity that would have occurred if the action had actually been performed. Closely related to this line of thought, the field of *Intuitive Physics* has gained relevance in recent years. Following Kubricht[2], "...humans are able to understand their physical environment and interact with objects and substances that undergo dynamic state changes, making at least approximate predictions about how observed events will unfold". Moreover, recent experiments show evidence that humans might have some sort of embedded mental game engine to help them reason about their environment [1][5]. This research is tightly connected to Robotics, and has early precedents in perception, with experiments showing internal visualization for

navigation [20][19], and in architectures, with the adding to SOAR of an internal simulator [18]. Although some authors have raised objections to the theory, arguing that it is only fruitful when both time and space scales are small [3], continuous advances in simulation technology and computer power are pushing the limits further with new demonstrations [12] [4].

In this paper, we present preliminary results on the integration of a PS (Physics Simulator) in a Robotics Cognitive Architecture (RCA) and how it can be used to solve known problems in robot perception. In particular, we aim to embed a simulation-based geometric reasoning pipeline into CORTEX. We have selected three use cases for this preliminary work that show the use of simulation to: (I) correct object poses perceived by the robot according to a physically plausible model, (II) detect pose perception errors and provide an automatic recalibration procedure and (III) reason about the persistence of out of sight objects. The following sections describe in more detail the CORTEX architecture, the use of simulation as an internal tool for complex scene understanding, and a description of the performed experiments to validate our application of this concept.

2 The CORTEX Architecture

The CORTEX cognitive architecture is a very dynamic proposal that has been evolving since its conception in 2016 [9][10]. In its current version, CORTEX defines a distributed architecture organized around a working memory (WM) and a set of agents that have access to it. The working memory is called Deep State Representation (DSR) and is formally a directed graph with vertices holding metric or symbolic data, and whose edges represent geometric or logic predicates. Vertices or nodes are concepts of the ontology, and edges are relationships between them. Being a WM, DSR is intended to represent the current situation involving the robot's body and intentions, and the space and objects proximal to it and relevant to the current task. Agents are responsible for creating and maintaining this representation, implementing the functionality of perception and motor modules, as well as the procedural, declarative and episodic memories of the Standard Model [23]

CORTEX establishes a way of operating within the working memory. The node representing the robot is always connected through an RT (Rotation-Translation geometric transformation) edge to one of the existing nodes representing a zone of space, typically a room in indoor scenarios. This edge is modified when the robot changes its location. From the robot, two distinct nodes, *body* and *mind* are connected downwards. The rest of the robot's parts are connected to *body* according to their kinematic relationships. These parts include rigid segments and pieces, joints, sensors and actuators. Raw data from sensors is stored in the node's attributes and made available to all agents. This is possible thanks to an efficient software design and implementation [6][7]. The other branch, *mind*, holds the current intention (or goal) of the robot. This intention is transformed into plans by deliberative agents, which create additional nodes

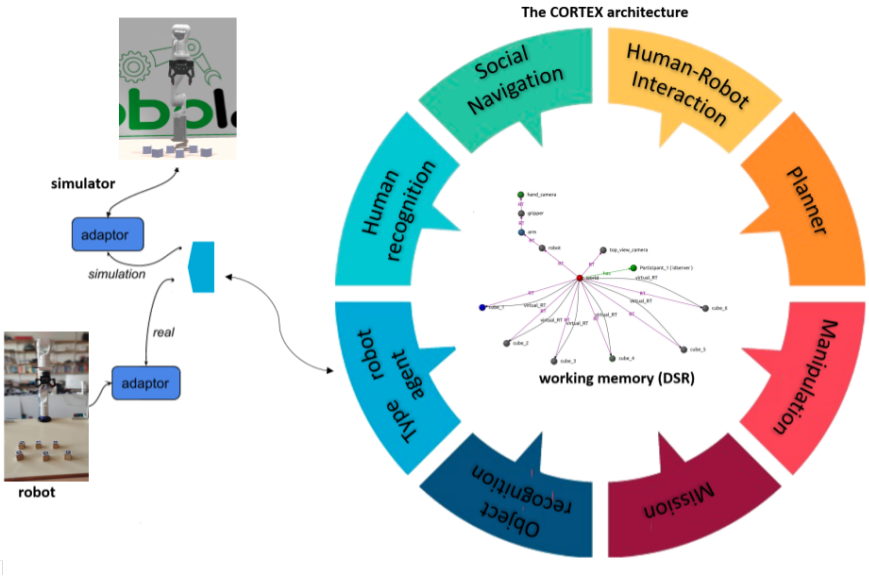


Fig. 1. The figure shows an instance of the CORTEX architecture with a double connection to the robot's body and a simulator.

hanging from *intention* to store and advertise the current list of subgoals or actions. All agents are made aware of the current intention and plans and take the actions of their choice to achieve the goal. Once this happens, the intention node is deleted and the agents go back to their local activities. Intentions are managed through a special agent called *mission-manager* that usually offers a GUI to the roboticist, accepts new missions from an interacting human or uses a scheduler to activate periodic missions. Figure 1 shows an abstract representation of a CORTEX deployment and Fig 2 a real case scenario for social navigation. More details on the implementation of CORTEX can be found in [11].

3 What can be obtained from an embedded simulator?

The working memory in CORTEX provides a persistent, structured state representation that can be used to implement complex cognitive mediated behaviours. However, there are many situations where direct perception and persistence are not enough to solve simple problems. What might be needed is some sort of pre-existing knowledge that could be injected into the WM, and provide a basic kind of common-sense. This knowledge can come from different sources and have various formal representations. A usual source in Cognitive Robotics is an ontology with rich relationships among concepts and an inference engine to answer queries, as in [17]. In this work, we also want to obtain predicates relating objects in the current scene, but we will focus on predicates that can be obtained from a physics simulator synchronized with the working memory. To explore this path

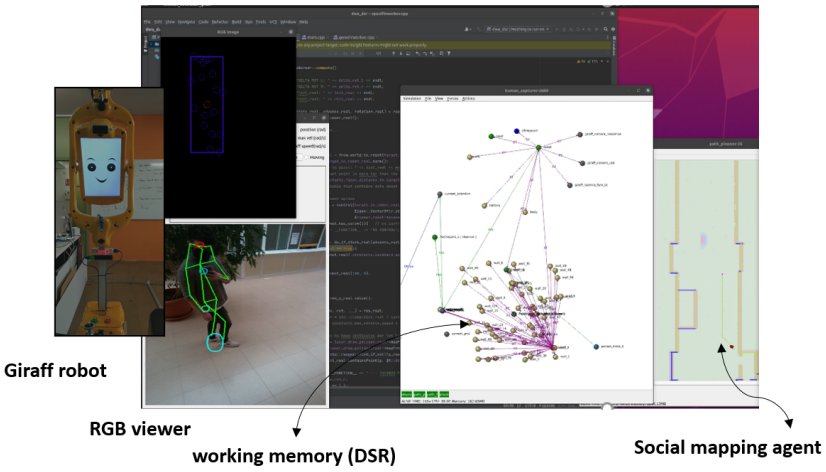


Fig. 2. Indoor example: Giraff robot navigation in a real scenario with people.

in a more systematic way, we have studied some examples and scenarios in which this common-sense knowledge would clearly improve the resolution of the task. The examples have been grouped into a set of informal categories that highlight common patterns. Three of them will be discussed here, as aforementioned in the Introduction, and the rest are briefly commented in the Conclusions section, as current research lines.

4 Experimental Setup

The experimental scenario includes of a Kinova Gen3 robotic arm placed on a table, and a set of 4cmx4cm cubes marked with a distinctive AprilTag [15] on one face. The arm has a RGBD camera (Realsense D415) attached to the wrist. Simulation is performed with the CoppeliaSim [14] using a choice of Physics engines[1]. The working memory in CORTEX is initialized from a file with a subgraph representing the robot and its sensors, and some additional nodes representing the scenario. An example configuration is presented in Fig. 4, where (a) shows the real world state and (c) how it is reproduced inside CoppeliaSim. All simulations run at 20 hertz on an Intel i9 10th generation and a RTX3090 GPU.

Three CORTEX agents have been created and deployed for these experiments.

1. **arm-controller**, this agent reads and inserts the gripper pose, read from the Kinova arm, into the working memory in real-time as a $SE(3)$ spatial transformation from the arm base. It also injects the raw stream of RGBD

[1] Bullet, ODE, Newton or Vortex

data obtained from the RealSense camera as an attribute of the *hand_camera* node.
2. **scene-estimator**, detects AprilTags and inserts model cubes into the working memory. The cubes hang from the node *camera* through an RT edge with the estimated relative pose.
3. **simulation-handler**, is in charge of synchronizing the working memory with the simulation bidirectionally. It reads cubes poses from the graph to update the simulation, and publishes them back as *Virtual_RT* edges. *Virtual_RT* edges are not part of the RT tree since they would induce loops. Instead, they are treated as symbolic edges representing an *opinion* from the simulator.

In CORTEX, agents run autonomously and can only communicate indirectly through the working memory. An example state of the DSR is given in Fig. 3 with all the information provided by the presented agents.

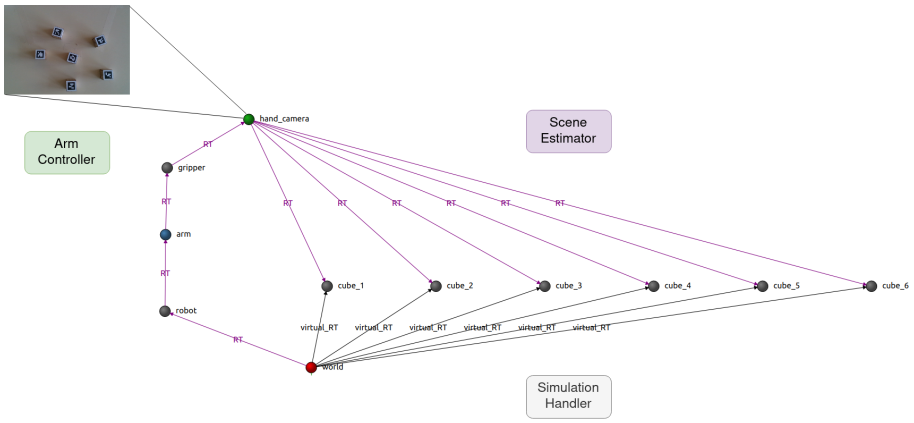


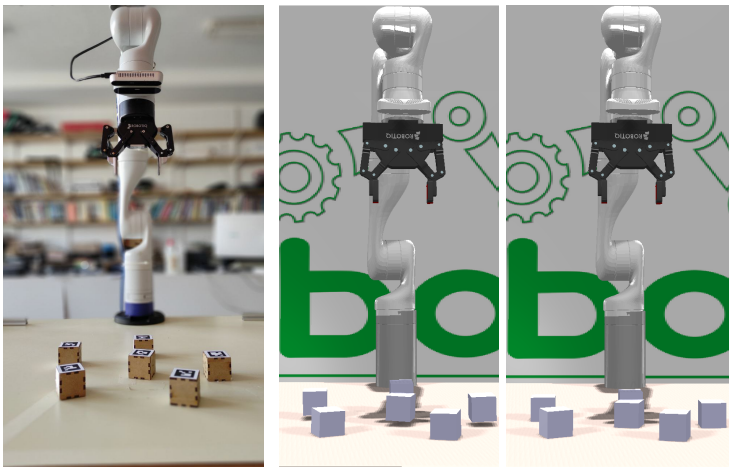
Fig. 3. Example DSR state. Arm Controller writes camera readings as an attribute of the hand_camera node. Position estimations of the cubes are presented as RT edges from *hand_camera*, inserted by the Scene estimator agent. Virtual_RT edges represent geometric transformations from the origin to every cube, and are given by the Simulation Handler agent.

We now describe three experiments, in which the use of the same integrated simulator, facilitates the approach to three well-known problems in Robotics.

4.1 Model-based perception

The perception of objects in the world by the robot is always a noisy process, subject to positioning errors. When the working scenario includes objects placed on top of, or leaning over other objects, the perception errors will yield physically unfeasible configurations, with floating and intruding objects. The simulator can apply the internal physics laws to quickly find a feasible configuration,

and project it back to the WM. The result being that the robot perceives a configuration of objects that has been corrected by a complex internal model following the laws of Physics (Fig. 4). By embedding this pipeline within the working memory, all agents gain access to this kind of physical reasoning, and a better understanding of the observed scene. Examples of this functionality have been shown in [12] [13]. In the following sections, we describe some ways that this can improve the robot's performance and precision.



(a) (b) (c)

Fig. 4. When the robot encounters state (a) using the RGBD information results in estimation (b) which suggest poses where cubes appear to be floating above the table. After applying the simulation physics, scene (c) is generated, correcting the initial erroneous perceptions.

4.2 Self calibration

As a result of model-based perception, there is a residual error computed between the estimated pose and the model corrected pose. This error can be systematic and derived from an incomplete calibration of the robot's sensors. Most frequently, the cause can be a miss alignment of the sensing device in the kinematic chain, i.e. RGBD camera or LIDAR, since this are usually hard coded by a human. In this case, the computed error can be used to re-calibrate the sensor by computing the corresponding derivatives.

Let c_i^r be the real world pose of cube i, p^r the camera's pose relative to its parent frame and c_i^e the pose for cube i estimated by the system. Using C^e (the set of all c_i^e) the simulation is synchronized, and after its physics are applied, c_i^s is obtained, as the corrected pose for cube i. In the following experiment, we aim to find a value for p^r that minimizes the distance between C^e and C^s, using

the simulator's corrections as a way to approximate C^r. The error function used is the average across all detected cubes of the distance described in equation 1, and Scipy's implementation of the Powell minimization method is executed.

$$dist(c_i, c_j) = \langle c_i^{rot}, c_j^{rot} \rangle^2 + ||c_i^{trans} - c_j^{trans}|| \quad (1)$$

Recalibration can come as a consequence of two distinct scenarios: (i) A well calibrated system which suffered some change (i.e. the sensor moved unexpectedly) or (ii) The initial sensor position is erroneous, and the correct one has to be found. We will explore the first situation here, but the same process can be applied for (ii).

Every time a new cube is detected, it is placed inside the simulation. In this scenario, the robot re encounters previously placed cubes, but the discrepancy between C^e and C^s (calculated with the error function) is too big. Many events could explain this difference, but a recalibration is triggered nonetheless to evaluate if a change in the camera pose can eliminate the error. If a value for p^r that aligns C^e to C^s is found, it is taken as the new camera position from this point on, since it is more likely to be the case than a coordinated reposition of all cubes in the scene.

To evaluate this functionality, a simple scene is created, similar to the one presented in Fig 4. After the cubes are detected and placed inside the simulation, the kinematic chain to the camera (p^r) is corrupted, in particular, the spatial transformation from the arm's tip to the camera is substituted by a random one. This generates the desired effect, where all cubes remain visible but the difference between detected (C^e) and remembered (C^s) positions is bigger than expected. For a setup of 4 cubes and a top view, 50 trials were conducted. Figure 5 presents how the error evolved during the optimization process. This graph shows that the method is capable of finding low error values for p^r even when starting from highly erroneous guesses. This convergence does not assure that p^r is in fact the real pose, but results show that the variance across trials is low and close to the measured position of the camera (Fig. 6).

4.3 Occlusion

When the robot ceases to see an object that is being tracked, its existence and continuity in the WM has to be updated by internal reasoning processes. As a representative example of occlusion, a recent experiment [12] shows a human holding a ball over two boxes and dropping it inside one of them. Then, the boxes are interchanged, and the system is asked for the location of the ball. For an RCA with an embedded simulator, the answer to the query is rather simple if we let gravity and collision detection algorithms operate on the free ball.

Once the system inserts any object in the simulator, its position is always reported in the WM as a Virtual RT. Unseen interactions are automatically computed, and the position information is available at any time to any agent that needs it.

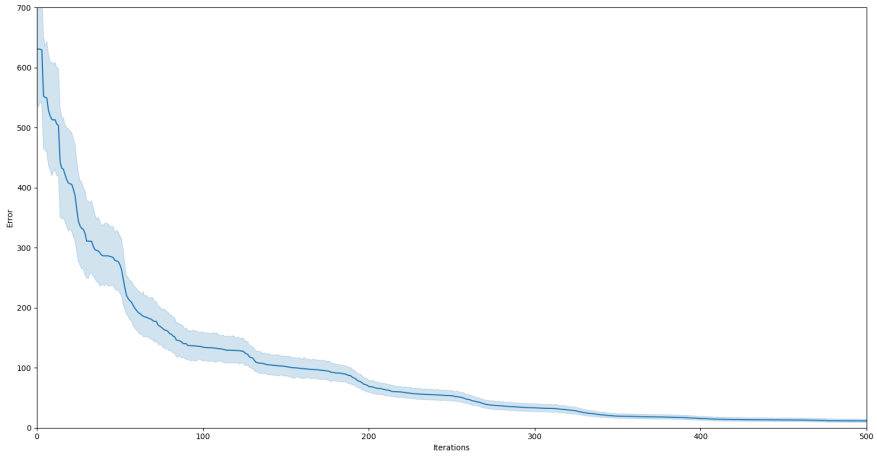


Fig. 5. Results of the recalibration after an undesired change in camera position. The blue line presents the mean across trials with the 95% confidence interval, showing how the system can find a low error solution despite initiating from highly erroneous guesses.

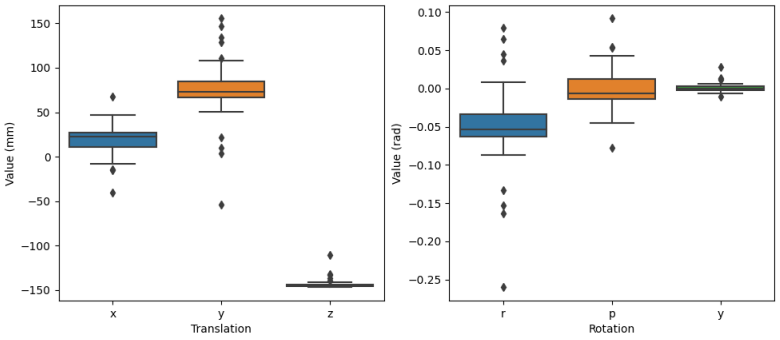


Fig. 6. Values for translation (x, y, z) and rotation (raw, pitch, yaw) from the arm tip to the camera found across trials.

In their experiments, Sallami et al. [12] detect human intervention with a state machine. When an object is presented in a configuration that violates strongly the effect of gravity, they conclude that it is being held by someone. Instead, we take advantage of the geometric reasoning provided by our setup, and use it for grasp detection. A new agent has been developed which detects human hands using the MediaPipe Hands algorithm [8] in combination with depth information, and places finger positions in the working memory. The *simulation-handler* agent then inserts the fingertips into the simulation as simple spheres. While one or more of these spheres is colliding with an object, gravity is not applied to it. Grasping and touching are not differentiated by the algorithm. This is one example of how having a physical duplicate of reality can endow the system with simple ways of detecting complex interactions, such as contact and gravity, that would be hard to describe with the use of rules or ad hoc algorithms.

Using these elements, we performed a similar experiment replicating the famous game of cups and balls (Fig. 7). The robot is presented with a cube and two boxes (Fig. 7(a)). One box is placed over the cube (Fig. 7(b)), and then the boxes are interchanged (Fig. 7(d)). At one point, the box containing the cube is moved over the edge of the table, dropping it to the floor (Fig. 7(e)). Thanks to the synchronized simulator, the system can detect this situation and continues to correctly track the cube position even though it is out of sight almost all the time. Being able to determine where the cube is without using extensive sets of rules or some case specific heuristics is a simple example of the advantages of our approach. Using internal simulation, this information is readily accessible to the robot.

5 Conclusions and future work

In this paper, we have presented some experiments performed with a Robotics Cognitive Architecture embedded with a Physics simulator. The goal has been to explore and test different uses of the new simulation capability, each of one enhancing some desirable aspects of the architecture. In the three examples, the underlying mechanism is the same, namely injecting back virtual RT edges with poses corrected through the filter of the Physics engine. This preliminary work is the first step in a long-term research focused on expanding the physical reasoning capabilities of autonomous robots by adding geometrical common-sense (GCS).

According to the distributed and reactive nature of CORTEX, the virtual RT edges are injected asynchronously by the simulator into the working memory. Instead of requiring explicit cues to access this geometric knowledge, our system is driven by the activity in the WM, and reacts to it by providing alternative RT edges. This line of though will be exploited in the future, when other common-sense agents handling declarative knowledge or episodic memories are developed.

In future work, we will expand the capabilities of the simulation by adopting several simplifying shortcuts that make the simulation tractable in real time for many objects [21]. This is needed to deal with more realistic situations. Besides, simplification allows for the deployment of multiple simulations and the use of

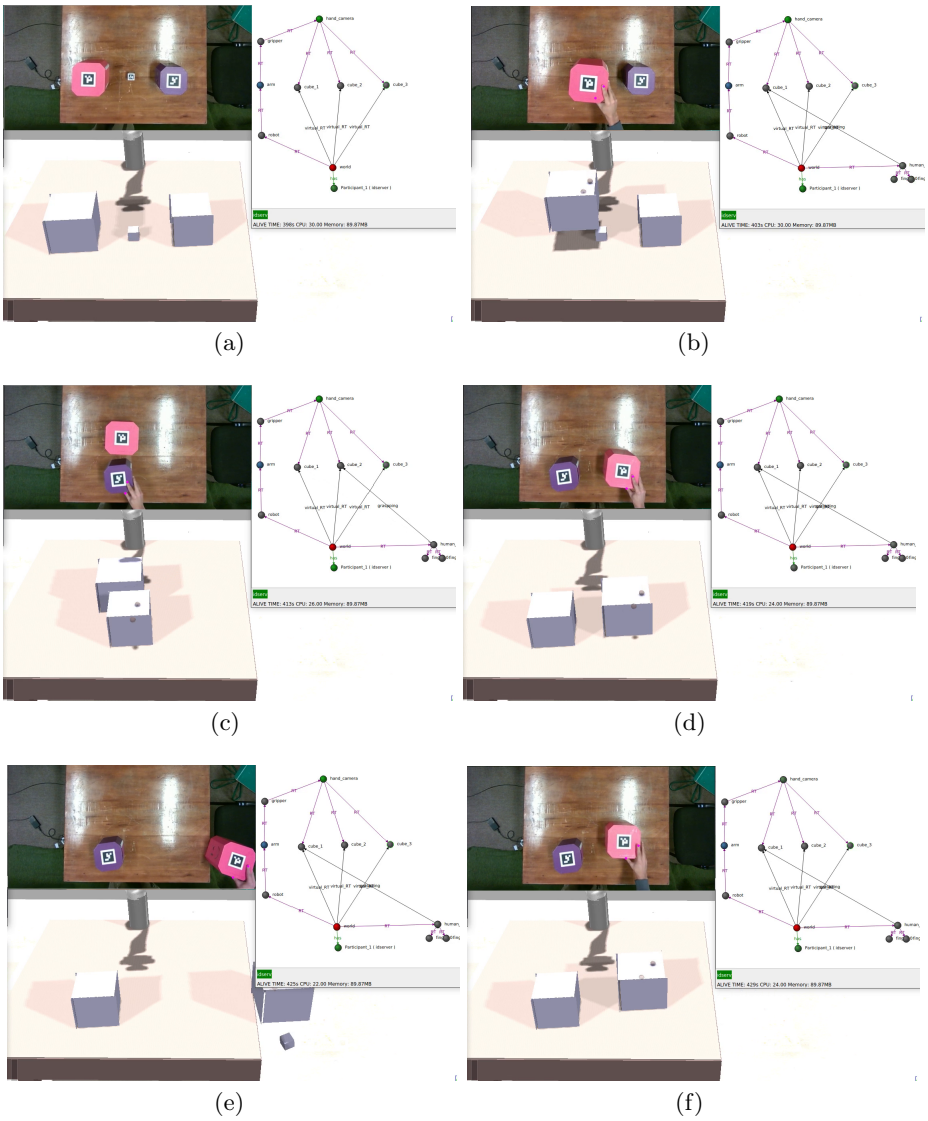


Fig. 7. Cups and balls experiment. (a) Shows the initial configuration. When a human hand is detected for the first time, a new node is inserted and two spheres representing fingertips placed in the simulation (b). Grasp detection can be seen in (c) and (d) where the WM has a *grasping* edge from the hand to the box being manipulated. In (e) the box containing the cube is placed over the edge and the simulator shows the cube falling to the ground.

a probabilistic approach to learning that can lead to unsupervised training of the robotic arm in cluttered situations. This in turn will ease the combination of probabilistic and neural network learning, an approach that has been successfully applied to other domains, such as program induction [22].

Another promising line of research targets the interpretation of human activities based on a combination of retargeting the perceived body into the robot's geometry, and the recovery of previous sensorimotor episodes in similar situations, i.e. grasping a box. When using a synchronized simulation where the human hand is retargeted into the robot's gripper, the physics engine can reproduce the most similar gripper configuration in which the object is held, and the gripper's force sensors will generate the corresponding response to the suspending effort. On one side, the predicate *hold(gripper, object)* can be evaluated now as a simple function of relative distances and velocities; on the other, we may say that the robot has a grounded interpretation of the current situation, that would enable it to make a description, predict future outcomes or react according to a purpose.

Acknowledgments. This work has been partially supported by the Feder funds and by the Extremaduran Goverment (projects GR21018 and IB18056), the MICINN RTI2018-099522-B-C42, by the Feder project 0770_EuroAGE2_4_E (Interreg V-A Portugal-Spain - POCTEP), and CSIC and CAP from Universidad de la República.

References

1. Tomer D. Ullman and Elizabeth Spelke and Peter Battaglia and Joshua B. Tenenbaum, Mind games: Game engines as an architecture for intuitive physics. Trends in cognitive sciences 21.9 (2017): 649-665.
2. James R. Kubricht, Keith J. Holyoak, Hongjing Lu, Intuitive Physics: Current Research and Controversies, Trends in Cognitive Sciences, Volume 21, Issue 10, 2017, Pages 749-759, ISSN 1364-6613, https://doi.org/10.1016/j.tics.2017.06.002.
3. Davis, Ernest, and Gary Marcus. "The scope and limits of simulation in automated reasoning." Artificial Intelligence 233 (2016): 60-72.
4. Mania, Patrick, et al. "Imagination-enabled robot perception." 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2020.
5. Battaglia, Peter W., Jessica B. Hamrick, and Joshua B. Tenenbaum. "Simulation as an engine of physical scene understanding." Proceedings of the National Academy of Sciences 110.45 (2013): 18327-18332.
6. Garcia, J.C., "G: a low-latency, shared-graph for robotics cognitive architectures" in Master Thesis, University of Extremadura, 2021
7. P. Núñez, J.C García, P. Bustos, P. Bsahciller Towards the design of efficient and versatile cognitive robotic architecture based on distributed, low-latency working memory. International Conference in Advanced Robotics and Competitions, Santa Maria da Feira, Portugal, 2022
8. Zhang, Fan, et al. "Mediapipe hands: On-device real-time hand tracking." arXiv preprint arXiv:2006.10214 (2020).

9. Bustos P, Manso-Argüelles Luis, Bandera A, Bandera J.P, García-Varea I, and Martínez-Gómez J., EUCognition Meeting - Cognitive Robot Architectures, CORTEX: a new Cognitive Architecture for Social Robots, Viena 2016
10. Bustos, P., Manso, L., Bandera, A., Bandera, J., García-Varea, Martín-Gomez, J. The cortex cognitive robotics architecture: Use cases. Cognitive Systems Research 55, 107–123, 2019
11. Bustos, P. García, J. C. Cintas, R. Martinena, E Bachiller, P. Núñez P. Bandera A. DSRd: A Proposal for a Low-Latency, Distributed Working Memory for CORTEX, Eds. Bergasa, L M. Ocaña, M. Barea, R. López-Guillén, E. Revenga, P., Advances in Physical Agents II, 2021, Springer International Publishing, pages 109-122, ISBN 978-3-030-62579-5
12. Sallami, Y. Lemaignan, S. Clodic, A. Alami, R. Simulation-based physics reasoning for consistent scene estimation in an HRI context IEEE International Conference on Intelligent Robots and Systems, pages 7834-7841, 2019.
13. Mosenlechner, L. Beetz, M., Fast temporal projection using accurate physics-based geometric reasoning IEEE International Conference on Robotics and Automation, pages 1821–1827, 2013.
14. E. Rohmer, S. P. N. Singh, M. Freese, "CoppeliaSim (formerly V-REP): a Versatile and Scalable Robot Simulation Framework", IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, 2013. www.coppeliarobotics.com
15. E. Olson, "AprilTag: A robust and flexible visual fiducial system", IEEE International Conference on Robotics and Automation, 2011, https://doi.org/110.1109/ICRA.2011.5979561
16. Hesslow G. Conscious thought as simulation of behaviour and perception. Trends in cognitive sciences 6, 242-247, 2002.
17. Beetz, Michael, et al. "Know rob 2.0—a 2nd generation knowledge processing framework for cognition-enabled robotic agents." 2018 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2018.
18. Samuel Wintermute, Integrating Action and Reasoning through Simulation, Proceedings of the 2nd Conference on Artificial General Intelligence (2009), 102-107, https://doi.org/10.2991/agi.2009.24, Atlantis Press
19. Ziemke, T., D.A. Jirenhed, and G. Hesslow, Internal simulation of perception: a minimal neuro-robotic model. Neurocomputing, 2005. 68: p. 85-104.
20. D.-A. Jirenhed, G. Hesslow, T. Ziemke, Exploring internal simulation of perception in mobile robots, in: K. Arras, A.-J. Baerveldt, C. Balkenius, W. Burgard, R. Siegwart (Eds.), 2001 Fourth European Workshop on Advanced Mobile Robotics—Proceedings, Lund University Cognitive Studies, vol. 86, Lund, Sweden, 2001, pp. 107–113.
21. I. Bass, K. A. Smith, E. Bonawitz and T. D. Ullman (2022) Partial mental simulation explains fallacies in physical reasoning, Cognitive Neuropsychology, DOI: 10.1080/02643294.2022.2083950
22. K. Ellis, C. Wong, M. Nye, M. Sable-Meyer, L. Cary, L. Morales, L. Hewitt, A. Solar-Lezama, J.B. Tenenbaum, DreamCoder: Growing generalizable, interpretable knowledge with wake-sleep Bayesian program learning, 2020, arXiv.2006.08381, doi:10.48550/ARXIV.2006.08381
23. Laird, J.E., Lebiere, C., Rosenbloom, P.S. A standard model of the mind: Toward a common computational framework across artificial intelligence, cognitive science, neuroscience, and robotics. Ai Magazine 38(4), 13–26 (2017)