See, feel, act: Hierarchical learning for complex manipulation skills with multisensory fusion

Ruprecht-Karls-Universität Heidelberg

Fakultät für Mathematik und Informatik

Institut für Angewandte Informatik

Seminar: AI in Games

Docent: Prof. Dr. U. Köthe

Speaker: Kerstin Küsters

Contents

Abstract 3
Introduction 3
Challenges 4
How it learns to play 4
Simulation 6
Experiments 6
How it plays Jenga 7
Materials and Methods 8
Conclusion 9
Table of figures 10
References 10

Abstract

Currently most robotic learning methodologies focus on visual information alone. This is often a fruitful endeavor but robotic learning can be tremendously improved by adding more information. Scientists at the MIT did this for a Jenga playing robot by adding tactile information. While for humans it comes naturally to combine various information inputs, e.g. visual, tactile, and hearing, etc. and infer actions based on these information, it proved more challenging for the robot. The proposed approach, hierarchical model abstraction (HMA), showed promising results with two sensory inputs.

Introduction

Human brains are capable of integrating multiple sensory channels, plan complex actions and learn abstract latent structures. This helps in the decision-making process since more information is available and therefore missing or having inconclusive information doesn't make the decision-making impossible. Ergo using multiple sensory channels for robots can provide them with enough information to still come to a viable course of action. Considering most robotic learning systems focus on visual data alone, the learning process is rather limited. Due to the fact that most learning algorithms rely on generic statistical models. With the additional data e.g. tactile input, the learning process could require less training and generalizations could be broader and more robust (paper). Combining inputs is the crucial part to learning. Main questions regarding the learning process are how do we use temporal tactile and visual information and how do we effectively infer and learn multimodal behavior to control touch? These questions define the two challenges of active perception and hybrid behavior. Active perception deals with how we learn by probing our environment with touch and vision. Hybrid behavior is concerned how we learn and infer to control touch.

The group of scientists at the MIT chose Jenga to train the robot. This is the perfect example, the game combines visual and tactile information since it requires the player to carefully push and pull blocks while keeping the tower intact. To achieve this the scientists at the MIT proposed a "Hierarchical Model Abstractions" (HMA) which emulates the hierarchical reasoning and multisensory fusion. The first step is to build abstractions in the joint space of touch and vision, the second step uses them to learn rich physics models.

They developed a simulation environment to compare the performance of their hierarchical learning approach to three other paradigms. There are some specifications regarding the tasks the robot is supposed to accomplish in the simulation and the experimental environment (Fazeli et al. 2019, 1):

1.) Sensing: The robot knows its pose, the pose of the blocks and the forces applied to it at every time step.
2.) Action primitives: push and extract/place. The latter is not learned but parametric and computed per call. The push primitive requires the robot to select a block, move to a collision-free position, selecting a contact location and heading. It needs to decide whether to keep pushing the block or choose another one.
3.) Base exploration policy: explore the tower for data collection it randomizes the push primitive, e.g. block, contact location, heading.
4.) Termination criteria:
   a) All blocks have been explored
   b) A block is dropped outside the tower or

c) The tower collapsed.

5.) Tower and robot specifications: simulated and real tower are composed of the same number and a similar distribution of movable and immovable blocks.

Challenges

To play Jenga the robot needs to learn a few things. What block is movable? How much force is needed to move it? How to not let the tower fall? Movable and immovable blocks are indistinguishable by mere look at them and need to be explored. If a suitable block is found the necessary force needs to be applied, the needed force is not a constant and varies with piece and tower level. Also the angle plays a role and needs to be adjusted while pushing the block. Pushing a block too far and have it fall out ends the game. The game is lost if the tower falls and some additional rules apply due to the visual systems limitations. So any action leading to damage needs to be avoided (Fazeli et al. 2019, 3).

There are more challenges regarding the learning policies:

1.) Many contact-rich manipulation skills are difficult to automate for large-scale data collection.
2.) Sim-to-real transfer of policies remains challenging because most simulators use computationally efficient but inaccurate models of frictional interaction.
3.) Tactile information is often intermittent i.e., making and breaking contact over a short duration. An effective integration of tactile information and the persistent visual stream is challenging.

How it learns to play

To learn the physics of the Jenga tower and about the abstractions the researchers went for a top-down bottom-up approach. The top-down learning builds the abstractions from the tower's physics, they are acquired before learning a detailed motion model and encode temporal macrobehaviors of the blocks. The variable identifies the type of macrobehavior. This way the robot learned about the blocks movability, meaning their resistance to pushing. For the tower itself the distribution of force is seen in Fig. 1 with a minimum, maximum and median. The next step is the state-transition via bottom-up learning in this case sensory data was used and factored by the abstraction. The Bayesian neural network (BNN), trained by the data collected during the exploration phase, is a probabilistic model for modelling the conditional distribution of future states in regard to current states and actions. With this the robot learned about the friction and force to push a block. To find the needed amount of force an analytical friction cone was computed. They assumed the applied force needs to be within these cones or on the boundary (Fazeli et al. 2019, 4 - 5).

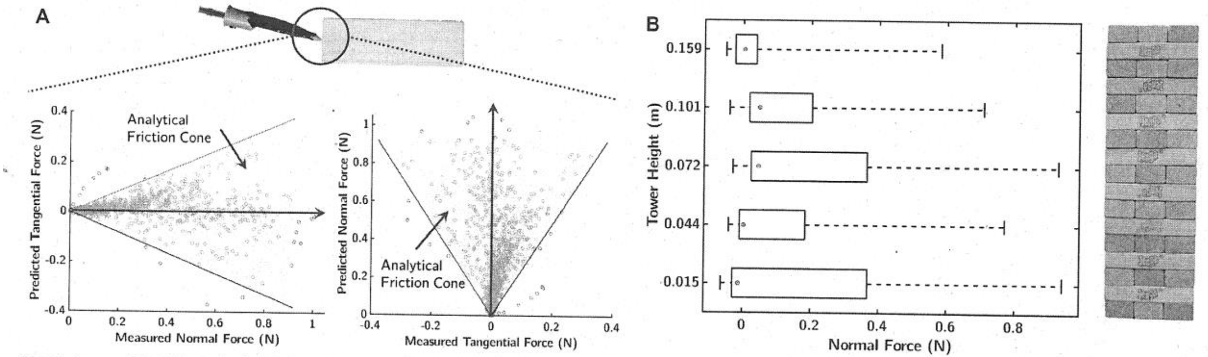

Fig. 4. Learned intuitive physics. (**A**) Overlay of the analytical friction cone and predicted forces given the current measurements. The friction coefficient between the finger material (PLA) and wood is between 0.35 and 0.5; here, we use 0.42 as an approximation. (**B**) Normal force applied to the tower as a function of the height of the tower. Each box plot depicts the minimum, maximum, median, and standard deviation of the force measures.

Fig. 1. Friction Cone and force measures.

Now the robot needs to learn to combine the information and make the right decision. Therefore the robot needs to infer the abstraction from the noisy sensor measurements, this provides a generative probabilistic model of physics. They used the Markov chain Monte Carlo (MCMC) sampling with Hamiltonian dynamics. A direct link between abstraction probabilities and how well the observations were explained could be identified. The robot assigns a probability to a block like ‘no block’ for the most probability mass and ‘easy move’ for some probability. At the beginning if the block offers only a small resistance, it is assigned the ‘no block’ state, pushing the block further transitions the state to ‘easy move’ due its increased displacement while the force doesn’t increase. The critical aspect is the perception algorithm with its high uncertainty at the beginning of the push. This uncertainty is due to the computed mask for the block, which doesn’t give much information and other factors can explain the current pose. Other than that the force measurement has the highest impact on the inference at the beginning. The configurations are hardly changing at the beginning but become more important along the push (Fig. 2). To gain some control an association between desirable abstraction and action has to be made, they added a penal to actions that resulted in increased probability of ‘no block’. This way the robot would know there is a block and pull it out eventually instead of dropping a block and lose the game. Another factor was to keep the tower damage minimal so large costs were assigned to tower perturbations (Fazeli et al. 2019, 5).

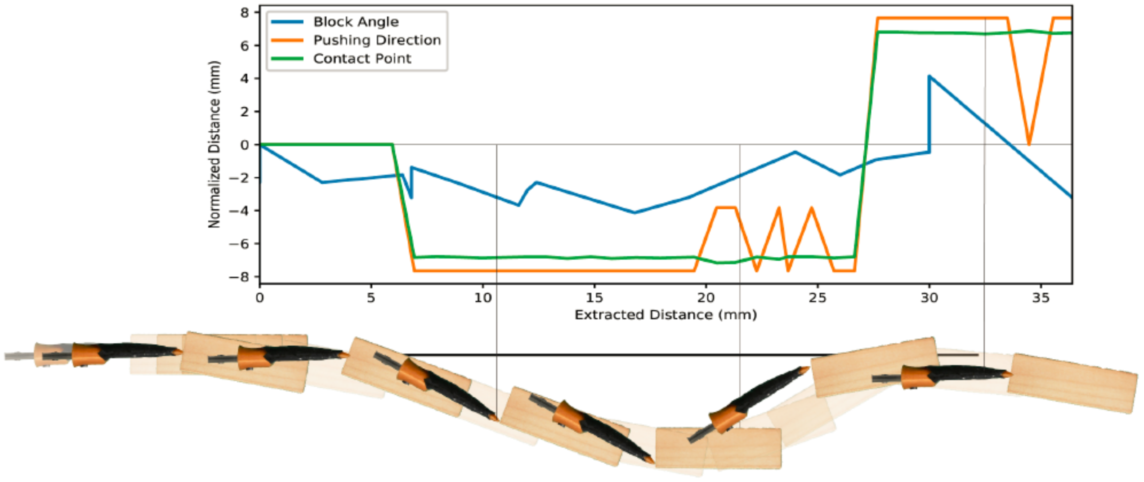

Fig. 6. Controlled block pushing. The robot selects the point on the block and the appropriate angle to push with such that it realigns the block with the goal configuration. Here, the block is beginning to rotate counterclockwise and is starting to move out of the tower. The robot selects a point close to the edge of the block and pushes it back in toward the tower center. We convert angles to normalized distances by scaling with the radius of gyration of the block (0.023 m). We have exaggerated the block translation to illustrate the fine-grained details of motion.

Fig. 2. Adjustment of the angle.

Simulation

The HMA was compared to the standard baselines of a feed-forward neural network (NN, nonhierarchical), a mixture of regressions (MOR, generic hierarchical) and a proximal policy optimization (PPO) implementation of reinforcement learning (RL, model-free). All have the same access to the data and the model predictive controller (MPC). During the exploration phase states and actions were collected and a model was trained. It was then evaluated on uniform test towers. A sample took about 2 s experimentally and 0.8 s in simulation, while a complete push is 45 steps and a sample is taken every step. A robot can extract 21 blocks on average, this also set the goal.

The HMA was successful within 100 samples, MOR took far longer but was successful, RL was slowest and the feed-forward NN saturated in performance and didn't reach the goal (Fig. 3). It either behaved too reckless or too conservative (Fazeli et al. 2019, 3).

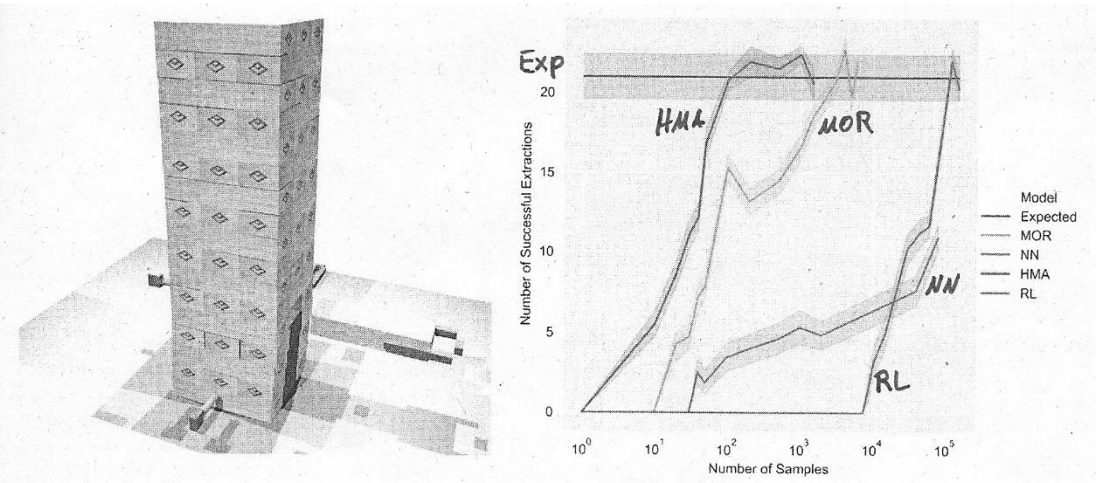

Fig. 3. Comparison of the four approaches.

Experiments

A camera and a sensor provided the modalities for the noisy approximations of the current pose of the blocks and the forces applied to the robot. The robot's forward kinematics were used to estimate the pose of the gripper, for the fingertips they computed the deflection. For this they used the measured force that was applied to the finger and known compliance parameters. This setup was then used for the HMA. An additional criteria had to be met was that the tower rotation or displacements

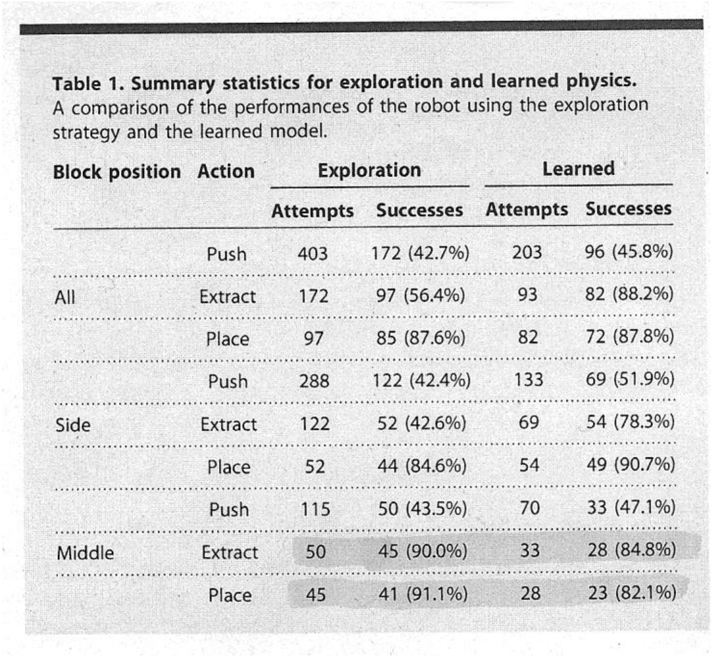

Table 1. Summary statistics for exploration and learned physics.
A comparison of the performances of the robot using the exploration strategy and the learned model.

Block position	Action	Exploration		Learned	
		Attempts	Successes	Attempts	Successes
	Push	403	172 (42.7%)	203	96 (45.8%)
All	Extract	172	97 (56.4%)	93	82 (88.2%)
	Place	97	85 (87.6%)	82	72 (87.8%)
	Push	288	122 (42.4%)	133	69 (51.9%)
Side	Extract	122	52 (42.6%)	69	54 (78.3%)
	Place	52	44 (84.6%)	54	49 (90.7%)
	Push	115	50 (43.5%)	70	33 (47.1%)
Middle	Extract	50	45 (90.0%)	33	28 (84.8%)
	Place	45	41 (91.1%)	28	23 (82.1%)

shouldn’t exceed 15° and 10 mm. This was due to poor predictions by the vision system. Another change was a hand-coded supervisory algorithm for the exploration strategy in order to mitigate damage and also allow for mistakes (Fazeli et al. 2019, 3-4).

The robot reached a success rate of 42.7 % the empirical average is at 47 % (Tab. 1). There are immense improvements between the exploration and the learning phase. Interesting are the middle blocks, the results show a negative development. This may be due to the fact of middle blocks being of constrained motion and a better weight distribution therefore being generally easier to remove. The main sources for failure were too much force applied to the tower or poorly controlled extraction since both led to issues with the vision system (Fazeli et al. 2019, 4).

How it plays Jenga

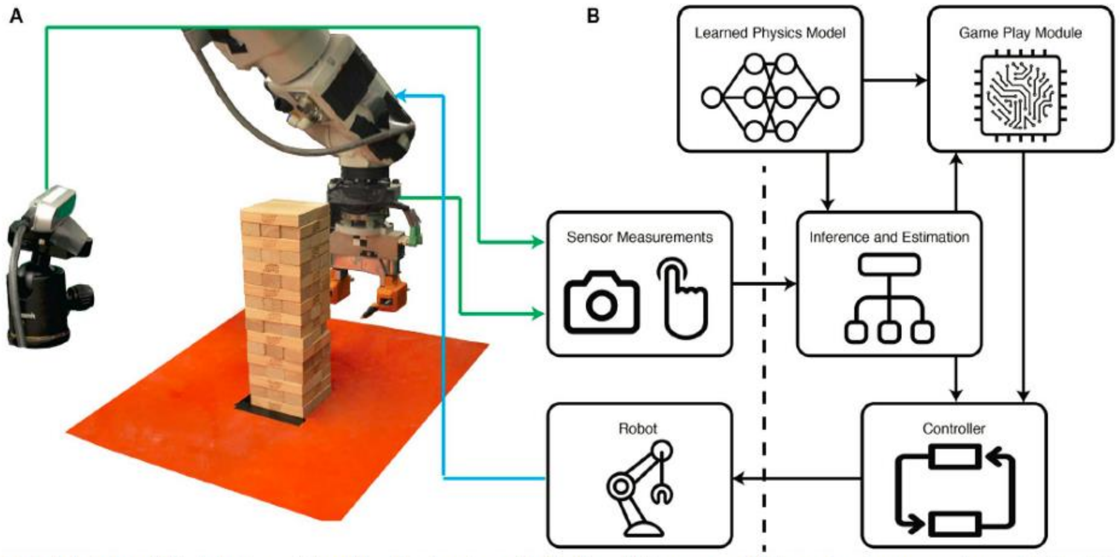

Fig. 1. Robot setup. (**A**) Physical setup consisting of the robot, Jenga tower, Intel RealSense D415 camera, and ATI Gamma force/torque sensor (mounted at the wrist). (**B**) Machine intelligence architecture with the learned physics model.

Fig. 4. Robot setup and its architecture behind.

The robot uses a camera for the visual input that is positioned to see two sides’ surfaces of the tower and a force/torque sensor for the tactile input. The information is passed to an inference and estimation module which makes the decision according to a learned physics model. The learned physics model is also part of the game play module. The conclusion of the inference and estimation module is sent to the game play module and the controller (Fig. 4).

Robot selects a block and moves to a collision-free configuration in the plane. Selects a contact location and heading and pushes for a distance of 1 mm and repeats. The action is completed if the robot chooses to retreat or a maximum distance of 45 mm is reached. Is the latter condition met the extract/place primitive becomes active and searches for collision-free grasp of the block and places it on top of the tower at a random unoccupied slot. This is not learned but parametric and computed per call.

The vision system records a low-pass filtered measurement of the force-torqued measurement and an RGB image which is processed by the perception algorithm to recover the 6-Degree-of-freedom (DOF) pose of block. The observation combined with the learned physics model are used to estimate the latent states and abstractions. These are send to the model predictive controller (MPC). Adding the forward rollouts of the learned physics model and computes the optimal action, which is then passed on to the robot for execution. Then repeat. A game play unit is responsible for sequencing the action primitives and monitoring the state of the tower with the sensor measurements and the inference block. This unit is hand-coded (Fazeli et al. 2019, 8).

Until the termination criterion is met the following execution loop is running:

1.) Select a random block and attempt ‘push’
2.) Choose between push poses and heading or retreat.
3.) If the block is pushed beyond ¾ of its lengths → extract/place routine

Materials and Methods

Clustering was used for the concept learning applying the Gaussian mixture model with a Dirichlet process (DP). This was done before to learn the abstractions. They used the mean and covariances of four clusters for it.

To model the state transitions in the physics model they used a BNN with the same input features as the baseline models except for an additional one-hot encoding of c_t, a latent variable.

For the probabilistic inference they let an MCMC perform over the learned physical representation. Four parallel MCMC chains ran on four CPU’s at run time with 3000 samples and a burn-in of 500 each. This caused one call to take 0.5 seconds. They decided that for longer periods the inference would be too costly regarding the computational expense.

The MPC gives a quadratic cost to the distance and the goal, change in action and, perturbation of the tower. A sampling-based, greedy MPC was applied to control the block motions. The depth is five steps and based on a learned model to select the cheapest action sequence.

The visual system first obtains the segment of each block via convolutional networks subsequently applying a template matching to recover the 6-DOF pose of the block. To teach the segmenting network a combination of 3000 synthetic images of the Jenga tower were rendered in Blender and 150 images in the experimental setup. The visual module then maps each segment to a 3-D block with its position and pose using a hidden Markov model (HMM) (Fig.7). Finding a sequence that explains visual observations well and maintaining temporal smoothness is the ideal. They used the classic Viterbi algorithm for inference, computing the probability of observations with three criteria. First, the intersection over union between the segment and the segment of the block from the template. Second, the intersection between the bounding box and the template segment. And third, the chamfer distances for the two sets of pixels within two segments (Fazeli et al. 2019, 8 - 9).

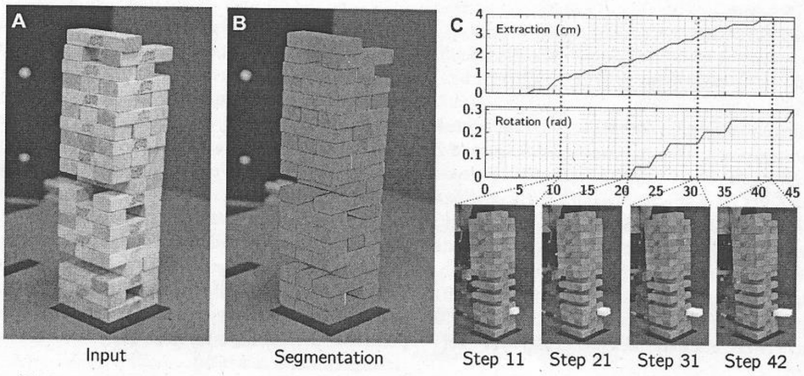

Fig. 7. The vision system. (**A**) We use a 280-by-280 patch with the tower at the center. For clarity, in this figure, we crop the irrelevant regions on both sides. (**B**) Segmented blocks. (**C**) For a single push, we identify the block being pushed and, based on its associated segment, infer the underlying pose and position of the block using an HMM.

Fig. 5. The information from the visual system.

Conclusion

The researchers were inspired by the 'concept learning' as it is studied in cognitive science, a concept meaning a type of abstraction composed of a typical feature set. This kind of understanding 'concept' is used for categorizing blocks on their general behavior, like 'no block' or 'easy block'. For humans it's a rather coarse abstraction, a stuck block isn't going to help reaching the goal, therefore a different block should be chosen. For a robot learning the general concepts brings two benefits, increasing the sample efficiency of learning and captured modalities can be used in controls and planning. Another aspect is the visual information this is covered by algorithms that produce autonomously acquired manipulation policies. This approach works ideally with a sufficient data stream and data can be automated effectively.

Two sources of information, like tactile and visual, complement the AI by providing information in the absence of the other. So basically, if one doesn't have the information needed the other may have it and the next step can still be inferred correctly. This makes the robot more stable and reliable, which are main concerns regarding the use of AI especially in the industry. A good deployment site would be where precise control is essential such as in electronic assemblage, logistics or disaster response (https://www.youtube.com/watch?v=ErdRlQbCviw). The robot is still far from human capability and it will take more research and time to achieve this level, but the results so far look very promising.

In the near future it is more likely for humans and robots to interact and work together. In this case the AI played alone and not against a human opponent. So what could be different? What if a human is working with the robot? These questions need to be considered as well in the future development of the robot.

There is also the action primitive extract/place which is not learned and the placement of a block is chosen at random. The action primitive doesn't sound to have a huge impact but for a flexible deployment it's probably essential, e.g. in disaster response. In the case of Jenga it could decide on whether to place a block in the middle or on the sides and stabilize the tower with this move.

Indeed there is still more to explore and further additions need to be made, but nonetheless the Jenga playing robot is an amazing step and its approach very promising.

Table of figures

- Figure 1, 3, 5: Fazeli, N., Oller, M., Wu, J., Wu, Z., Tenenbaum, J. B., Rodriguez, A. (2019): „See, feel, act: Hierarchical learning for complex manipulation skills with multisensory fusion“, in Science Robotics, Vol. 4, Issue 26.
- Figure 2 & 4: https://medium.com/syncedreview/ai-gaming-gets-physical-mit-robot-plays-jenga-e911fbaebdff
- Table 1: Fazeli, N., Oller, M., Wu, J., Wu, Z., Tenenbaum, J. B., Rodriguez, A. (2019): „See, feel, act: Hierarchical learning for complex manipulation skills with multisensory fusion“, in Science Robotics, Vol. 4, Issue 26.

References

- Fazeli, N., Oller, M., Wu, J., Wu, Z., Tenenbaum, J. B., Rodriguez, A. (2019): „See, feel, act: Hierarchical learning for complex manipulation skills with multisensory fusion“, in Science Robotics, Vol. 4, Issue 26. [Note: Access to the article was via interlibrary loan and page numbers are based on that version and do not correspond with the magazines version.]
- http://www.jenga.com/about.php [27.06.2019]
- https://www.youtube.com/watch?v=o1j_amoldMs [06.11.2019]
- https://www.youtube.com/watch?v=ErdRlQbCviw [03.07.2019]
- https://www.youtube.com/watch?v=bvOMhM_uDkI&t=218s [03.07.2019]
- http://news.mit.edu/2019/robot-jenga-0130 [03.07.2019]
- https://newatlas.com/mit-jenga-playing-robot/58276/ [04.07.2019]