Autonomous robots: potential, advances and future direction

S. Hangl, E. Ugur, J. Piater

Recent advances in machine learning, such as deep neural networks, have caused a huge boost in many different areas of artificial intelligence and robotics. These methods typically require a large corpus of well-prepared and labelled training data, which limits the applicability to robotics. In our opinion, a fundamental challenge in autonomous robotics is to design systems that are simple enough to solve simple tasks. These systems should grow in complexity step by step and more complex models like neural networks should be trained by re-using the information acquired over the robot's lifetime. Ultimately, high-level abstractions should be generated from these models, bridging the gap from low-level sensor data to high-level AI systems. We present first steps into this direction and analyse their limitations and future extensions in order to achieve the goal of designing autonomous agents.

Keywords: autonomous robots; cognitive robotics; developmental robotics; lifelong learning; robot creativity; robot playing

Autonome Roboter: Potenzial, Fortschritte und künftige Richtungen.

Jüngste Fortschritte im maschinellen Lernen, wie tiefe Neuronale Netze, haben einen großen Schub in vielen verschiedenen Bereichen der Künstlichen Intelligenz und Robotik bewirkt. Diese Methoden erfordern in der Regel einen großen Stamm an gut aufbereiteten Trainingsdaten, welche die Anwendbarkeit der Robotik begrenzen. Unserer Meinung nach ist es eine grundlegende Herausforderung in der autonomen Robotik, Systeme zu entwerfen, die einfach genug sind, um einfache Aufgaben zu lösen. Diese Systeme sollten dann Schritt für Schritt an Komplexität zunehmen. Komplexere Modelle, wie Neuronale Netze, sollten schließlich durch das Auswerten der über die Zeit gewonnenen Informationen laufend weiter trainiert werden. Letztendlich sollten aus diesen Modellen hochrangige Abstraktionen generiert werden, die gewisse fehlende Informationen von Low-Level-Sensordaten hin zu High-Level-Systemen der Künstlichen Intelligenz überbrücken können. Die Autoren stellen erste Schritte in diese Richtung vor und analysieren die Grenzen bzw. künftigen Erweiterungen mit dem Ziel, autonome Agenten zu entwerfen.

Schlüsselwörter: autonome Roboter; kognitive Robotik; lebenslanges Lernen; Roboter-Kreativität; Roboter-Spiele

Received August 17, 2017, accepted August 29, 2017

1. Introduction

The question about what constitutes human intelligence and what enables them to act so effortlessly in our highly unstructured world puzzled many of the greatest minds of humankind [1, 14, 22]. For a long time this topic was mainly reserved for philosophers. However, during the last century, groundbreaking discoveries in medicine, biology and developmental psychology disclosed principles involved in the human brain [4, 8, 9, 15]. Even though many basic principles of the human brain are understood in a rudimentary way, until recent years it was not possible to reproduce human-like performance in most scenarios. This strongly changed with the increase of computational power and the rise of machine learning approaches such as techniques summarised under the term *deep neural networks* [17]. These methods caused a breakthrough in several disciplines that were assumed to be unsolvable for several decades to come [11, 18]. Despite all these exciting developments, it is still not possible to buy robots in a shop in order to let them work in households or other unstructured environments.

1.1 Burning questions in robotics

There is a large variety of reasons for the lack of functional household robots. The sensorimotor space is huge and it is impractical or even impossible to learn complex skills from scratch by simple, unguided exploration of the whole space. We believe that a learning agent needs to be equipped with biases that allow it to reduce the problem complexity while still maintaining generality. An example of such a bias is the definition of meaningful dimensionality reductions like feature extractors on sensor data, limiting the controllable joints or ignoring certain dimensions of the sensor data. Even human infants exhibit such biases, for example in the form of the grasp reflex or the change of importance and strength of certain senses. Abstract concepts can be trained from experience which in turn can change the nature of the used biases, e.g. by unlocking consideration of additional dimensions of the sensor data. This enables the robot to learn how to solve more complex problems and to use its complete sensorimotor capabilities based on the learned abstractions.

Modern machine learning approaches typically require a huge amount of training data. Currently this often requires massive human intervention, which is equivalent to providing the described biases manually, e.g. by tagging images of objects or designing simulated environments. This limits the applicability of the powerful learning machinery to isolated scenarios in which experts design appropriate solutions by providing controlled learning environments [12], by designing simulated environments for learning [16, 21, 25]

Hangl, Simon, Intelligent and Interactive Systems, Universität Innsbruck, Technikerstraße 21A, 6020 Innsbruck, Austria (E-mail: simon.hangl@uibk.ac.at); **Ugur, Emre,** Bogazici University, Istanbul, Turkey (E-mail: emre.ugur@boun.edu.tr); **Piater, Justus,** Intelligent and Interactive Systems, Universität Innsbruck, Technikerstraße 21A, 6020 Innsbruck, Austria (E-mail: justus.piater@uibk.ac.at)

Fig. 1. The goal of *developmental robotics* is to design robots that learn like human infants. This can be done by life-long learning and interaction with the environment [7]

or by adding reasonable domain-specific assumptions [10]. When training is done, these methods generalise well to a large variety of environments for the specific task. However, there is an infinite number of tasks which makes it hard to follow this paradigm until all relevant tasks are covered.

1.2 Developmental robotics

In recent years a sub-discipline of robotics tackling this kind of problems attracted increased attention, that is, *developmental robotics* [2, 13, 24]. This branch treats the design of self-learning and autonomous agents, often inspired by human or primate infants; cf. Fig. 1. Stoytchev identified 5 key principles that most work in developmental robotics has in common in order to create autonomous robots [19]:

1. *Verification principle*: Novel skills can only be learned autonomously to the extent that robots can verify the success of performed action autonomously [20].
2. *Principle of embodiment*: In order to be able to gain new knowledge about the world the robot requires a body with which it can interact with the environment, i.e. perceive (e.g. with sensors) and act (e.g. by using motors).
3. *Principle of subjectivity*: During the lifetime of a robot, it makes many observations and gains knowledge about the world. It learns how to act in this world and how to achieve certain goals. How the robot does this strongly depends on its subjective history. The robot is subject to sensorimotor limitations, i.e. there is nothing to learn about the world if it cannot be accessed by sensors or motors, and to experimental limitations, i.e. the knowledge the robot is able to acquire is limited by the experiments it performs.
4. *Principle of grounding*: Even though the verification principle states that everything has to be verified autonomously, there must be an atomic information entity in which the robot has to trust. An example would be the low-level tactile feedback of an artificial skin if the robot has no other possibility to check for the correctness of the observed data.
5. *Incremental development*: Not everything can be learned at the same time, because some problems are simply too complex for a certain stage of development, e.g. humans do not learn integrals before addition.

These principles enable robots to gather information over a lifetime with which powerful learning techniques can be used. Robots can interact with and improve their knowledge about the world. The robot first gains simple knowledge and skills and simplifies whenever possible, e.g. by reducing the dimensionality of sensor data. The more experience the robot gains, the more complex the models get and the harder the task to solve can be. This represents different stages of development in a robot's life. In this work we summarise the authors' previous work that spans the transition from simple random combination of learned behaviours to elementary goal-based planning and finally to high-level cognitive capabilities. We check this work for agreement with the principles of developmental robotics and analyse the limitations and future challenges.

2. Robotic playing

Hangl et al. have presented work on robotic playing based on insights in developmental psychology [7]. Piaget studied human infants and identified different developmental stages [15]. An early stage in infant development is the so-called *coordination of secondary schemata*, which is dominant during an age between 8 and 12 months. In this phase, infants try to reach certain goals by sequencing behaviours they learned earlier. A key point is that they do not yet create sophisticated plans but rather learn that a certain combination works well to achieve a desired goal without actually understanding yet why.

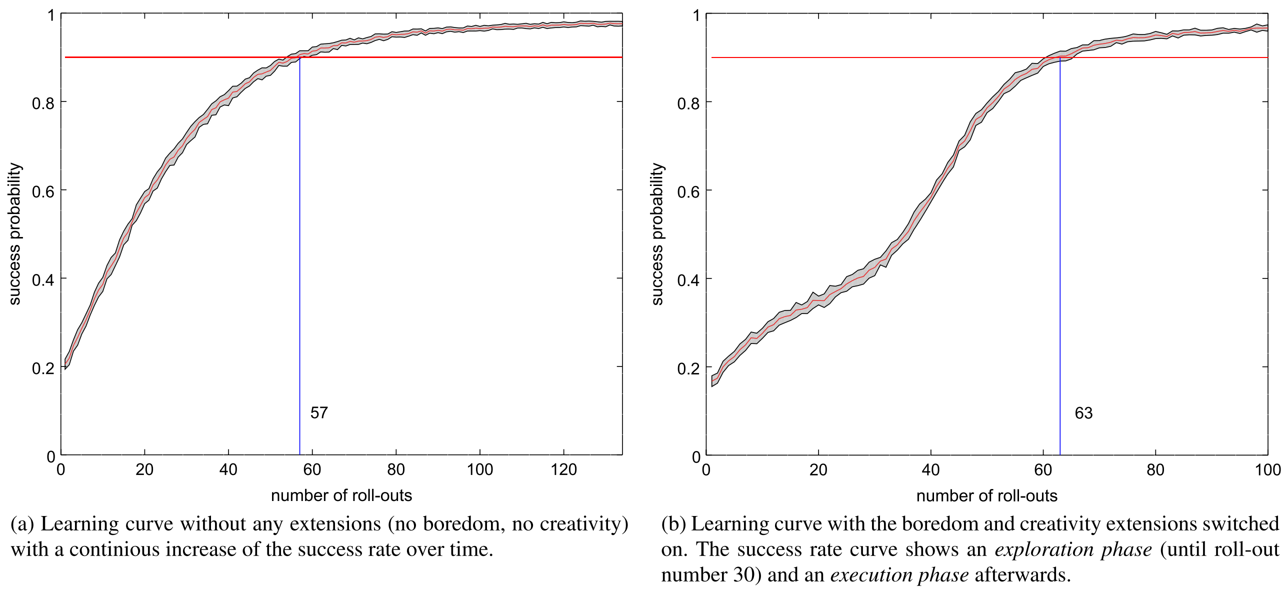
2.1 Learning behaviour sequences

We followed this principle by learning combinations of so-called *behaviours* which are functions $b : S \rightarrow S,\ s \mapsto s'$ that map (partially observable) environment states $s \in S$ to other states $s' \in S$. A behaviour $b^\sigma \in B$ is part of a skill $\sigma = (b^\sigma, \mathrm{Success}^\sigma)$ if the robot additionally holds a success predicate $\mathrm{Success}^\sigma$ which enables it to *verify* whether or not a goal was achieved. In this case the behaviour b^σ is called the *basic behaviour* of the skill σ. The *domain of applicability* (DoA) $D^\sigma = \{s \in S \mid \mathrm{Success}^\sigma(b^\sigma(s)) = \mathrm{true}\}$ is the set of all states in which the goal can be achieved.

The idea is to train new skills by teaching a basic behaviour, e.g. by kinesthetic teaching or simple programming, for a specific, restricted DoA. The robot then increases the DoA by using other trained behaviours to prepare the environment such that the basic behaviour can be executed successfully again. This is equivalent to learning behaviour sequences $b \circ b^\sigma(\hat{s})$ with $\mathrm{Success}^\sigma(b \circ b^\sigma(\hat{s})) = \mathrm{true}$ for states $\hat{s}$ that were not in the DoA of the skill before, i.e. $\hat{s} \notin D^\sigma$.

We illustrate this on the task of placing an object in a drawer. The basic behaviour b^σ is to place an object inside an open drawer with a simple trajectory. If the drawer is open, i.e. the environment is in state s_{open}, no preparation is required. If the drawer is closed, preparation is done by opening it with an opening behaviour b_{open} and therefore the skill is executed by $b_{\mathrm{open}} \circ b^\sigma(s_{\mathrm{closed}})$. Whether or not a drawer is open can be determined either by using vision or, as done in our framework, by analysing haptic feedback acquired by poking the drawer. The robot identifies the haptic feedback corresponding to relevant states, so-called perceptual states, by playing with the involved objects, and tries out different combinations of behaviours in order to achieve the task.

A *perceptual state* is characterised by task-relevant aspects of the sensor data. This generally requires dimensionality reduction and is done by training a classifier to decide which perceptual state is present, e.g. s_{open}, s_{closed}. The necessary information is gathered by playing with the object and exploring it with different sensing actions in the respective perceptual states. In the most basic version this requires a human playing partner which prepares the environment such that the corresponding perceptual states are present. The sensing action that is identified to be most effective to discriminate between different perceptual states is used in the future to estimate the state. Before executing a skill, the perceptual state is estimated, and, depending on the state, an appropriate preparatory behaviour is identified by autonomous playing.



(a) Learning curve without any extensions (no boredom, no creativity) with a continious increase of the success rate over time.

(b) Learning curve with the boredom and creativity extensions switched on. The success rate curve shows an *exploration phase* (until roll-out number 30) and an *execution phase* afterwards.

Fig. 2. Learning behaviour for a book grasping skill with different modes of the developmental robotic system described in Sects. 2 and 3. The graphs are generated from experiments published in previous work [5]

When a skill is well trained, it can be used as preparatory behaviour for other skills by constructing skill hierarchies, in order to solve more and more complex tasks. An example might be the task of filling the drawer with objects and closing it again. Here, the basic behaviour is to close the drawer, and the preparatory behaviour is the placement skill described before.

2.2 Analysis

A typical skill that can be learned with this kind of learning approach is how to grasp a book from a solid surface. The basic behaviour is to grasp a book at the spine. The book is squeezed between both hands (cf. Fig. 2). It is slightly lifted in order to get the fingers below it and is grasped afterwards. If the book is placed on the table in a different orientation, this does not work anymore. However, the robot can still increase the DoA by learning that rotating the book to the correct position is sufficient to achieve the goal. In this case, the useful behaviours are *rotate 90°, rotate 180°, rotate 270°* and the *void* behaviour (if the book is already rotated correctly). Each of them solves the problem for one of the four possible rotations of a book. The rotation of a book can be determined by sliding along its edges. The learning rate in this typical manipulation scenario is displayed in Fig. 2a and shows a continuous increase of the success rate. The longer the robot plays with the object, the more situations can be solved.

This kind of approach has some advantages that make it very easy to fulfil the principles of developmental robotics and are therefore well suited for autonomous robots.

A typical problem in fulfilling the *verification principle* is to gather the data required for making a valid decision about success. This is typically due to the large domain of applicabilities and strongly varying sensor data for different states. For example, the joint or Cartesian arm positions vary strongly for different object positions in a grasping task. We strongly reduce this problem by transforming the environment to the same state in which the basic behaviour was trained. This drastically reduces the variety of sensor data encountered. For example, the book is always pushed into the same orientation from which it is grasped, and therefore it is much simpler to train the success predicate Success$^{\sigma}$. In previous work we demonstrated that such a predicate can be trained from successful training data by using recurring neural networks [6].

Another important property is that the robot learns to ground relevant perceptual states in the sensor data, i.e. haptic data in our case. Here we make a pragmatic compromise by either having a human supervisor prepare the perceptual states or by telling the robot how to prepare them autonomously. This can be viewed as supervised playing, just as it is done with babies as well.

We emphasise the support for *incremental development* in which simple skills are trained first, which can then be used as preparatory behaviours for other, more complex skills.

3. Transition to goal-driven planning

Even though the approach described in Sect. 2 supports autonomous behaviour, cognitive capabilities and the expressive power are limited. For example the robot does not *understand* the environment it is acting on in a sense that it knows the effects of its actions. The autonomous playing is essentially based on trying out different combinations of behaviours in order to achieve a task without requiring a model of the environment, i.e. a function that predicts the effect of a behaviour. In the spirit of the principles of *incremental development* and *subjectivity* we extended this approach by learning such an environment model in order to reach the next developmental stage [5]. The idea arises from the insight that during the autonomous playing the effects of executed behaviours can be observed by estimating the perceptual state again after executing a behaviour. This way, a probability distribution $p(s' \mid s, b)$ over the effect $s' \in S$ of the behaviour $b \in B$ on the state $s \in S$ can be learned over time. Initially, without any knowledge, the distribution is uniform because the outcome of certain actions is completely unknown. The distribution is refined by updating the transition probabilities given observations made. We do so by using an approach to autonomous agents called *projective simulation* [3]. The environment model becomes more and more mature over time and can be used to unlock higher-level cognitive capabilities instead of the undirected exploration described in Sect. 2. As an example we enabled the robot to *feel bored* and implemented *creative behaviour*.

3.1 Bored robots

Experience showed that the robot wastes time if a perceptual state is present in which the solution is already known very well. If the goal is to learn how to achieve a task in general, there is no sense in learning how to solve the specific situation even better; this is *boring*. However, just rejecting a given situation is not enough in an autonomous setting, as further learning requires a different and more interesting situation. Here, we use the environment model to compute the likelihood of being able to reach a certain interesting and desired state $s_{\text{int}} \in S$. An *interesting state* is a state in which the solution is not known yet. The robot tries to reach the interesting state $s_{\text{int}} \in S$ from the current, boring state $s_{\text{bor}} \in S$. The likelihood of a transition $s_{\text{bor}} \xrightarrow{b_1} s' \xrightarrow{b_2} s'' \xrightarrow{\dots} s_{\text{int}}$ can be computed by using the environment model, and the robot can change the state if it is bored. We use the transition entropy $H\left(s' \mid s, b\right)$ to measure how predictable the outcome of executing the behaviour b is in a state s. If the entropy is high, the outcome is either unknown or highly unpredictable. If it is low, the outcome is highly predictable respectively. We normalise the transition entropies in order to obtain an entropy value $\hat{H}\left(s' \mid s, b\right)$ between 0 and 1. This can be used to compute the transition likelihood for a complete path $s_{\text{bor}} \xrightarrow{b_1} s' \xrightarrow{b_2} s'' \xrightarrow{\dots} s_{\text{int}}$ by multiplying the normalised entropies on the path. The robot then selects goal states s_{int} that it cannot solve yet but that have a high transition likelihood.

3.2 Creative robots

Another application of such an environment model is to enable the robot to show creative behaviour. In Sect. 2 the goal is to extend the domain of applicability by selecting *one* preparatory behaviour with $\text{Success}^{\sigma}(b \circ b^{\sigma}(\hat{s})) = \text{true}$. In many cases just one preparatory behaviour might not be enough. For example for reading a book that is placed upside down, it has to be *rotated* and *opened* and therefore $\text{Success}^{\text{read}}(b^{\text{rot}} \circ b^{\text{open}} \circ b^{\text{read}}(s_{\text{up}})) = \text{true}$. In the basic version described in Sect. 2, sequences of arbitrary length were forbidden for good reason, namely to prevent an explosion of the space to explore. For example, if 5 behaviours are available, allowing at most 3 behaviours in a sequence would require the robot to try 125 different combinations. However, if a model of the environment is available, not all combinations have to be tried out, but only those that seem to be promising according to the model. As soon as the robot has identified at least one perceptual state s_{goal} it can solve with the original basic approach, i.e. $\text{Success}^{\sigma}(b \circ b^{\sigma}(s_{\text{goal}})) = \text{true}$, this state is marked as a *goal state*. If another perceptual state $s_{\text{curr}} \in S$ is observed and there exists a strong transition $s_{\text{curr}} \xrightarrow{b_1} s' \xrightarrow{b_2} s'' \xrightarrow{\dots} s_{\text{goal}}$, it is promising to try whether the sequence $(b_1, b_2, \dots, b_L, b)$ yields success. This way the robot can creatively generate novel behaviours composed of more elementary behaviours in order to achieve increasingly complex tasks.

3.3 Analysis

In the work described above we demonstrated the importance of the principle of *incremental development*. The robot starts with the very basic version described in section 2 and collects all the available data for later use. The only additional overhead is introduced by requiring re-estimation of the perceptual state after executing a preparatory behaviour in order to train the environment model. An important property of our method is the incremental development. The brute-force exploration of behaviour combinations is replaced by higher-level planning in a natural way over time. There is no sharp boundary between the different phases which can be observed in Fig. 2b. Figure 2b shows the learning rate in the book grasping scenario described in Sect. 2.2. However, in this case the behaviours *rotate 180°* and *rotate 270°* are missing, which requires creative creation of these behaviours. This can be done by composing them out of *rotate 90°* behaviours. In the learning curve a first exploration phase with a continuous increase of the success rate can be seen. This is due to the fact that two situations can be solved with the available *rotate 90°* and *void* behaviours. At some point the robot reaches a plateau after around 30 attempts but shows a strong increase of the learning rate again as soon as the environment model is mature enough to creatively generate the behaviours *rotate 180°* and *rotate 270°*. High-level planning becomes more dominant as the environment model matures.

We demonstrated that our system mimics certain natural human behaviours such as *boredom* or *creativity*. Humans refuse to perform certain activities if they are monotonic and if there is nothing new to learn anymore; they avoid *boredom*. In this case humans search for different and more interesting tasks instead of simply being idle.

4. Bottom-up learning of high-level symbols

In Sects. 2 and 3 the goal is to equip the robot with the capabilities to slowly perform a transition from very low-level skill learning to more and more complex concepts in a smooth way. An approach slightly different in nature tackles the problem of mapping low-level data to high-level symbolic planning with logic planners [23]. The goal is not to solve a task directly but to learn a symbolic representation of action effects and object relations. If a task is to be achieved, a logic planner is run on the symbolic representation.

4.1 Dimensionality reduction by abstract symbol learning

Each object o in the scene has a *continuous object state* $\mathbf{f}_o$. The continuous object state is *grounded* in actual experience via the robot's sensors, and can consist of any sensor information available to the robot; in our work we use RGB-D sensors. Several objects can be present in the scene, and the *continuous world state* $(\mathbf{f}_{o_1}, \mathbf{f}_{o_2}, \dots, \mathbf{f}_{o_N})$ is a tuple combining all object states in the scene. Similarly to the notion of perceptual states in Sects. 2 and 3, a dimensionality reduction of the high-dimensional state vectors is required. The robot holds a set B of behaviours. Each behaviour $b \in B$ gives rise to *effect categories* ϵ^b by observing the effect of executing it. For example, if a ball o_{ball} is poked from the side with the behaviour b_{poke}, the ball rolls off the table and disappears, and the effect category is $\epsilon^{\text{poke}} = \text{disappear}$. Given a set B of behaviours, the tuple $S_o = (\epsilon^{b_1}, \epsilon^{b_2}, \dots, \epsilon^{b_M})$ of effect categories of all behaviours applied to the object o defines its *object category*.

The first goal is to learn the dimensionality reduction from the continuous world state to the object category. This is done by executing each behaviour many times for all available objects (using a real robot or in simulation) and measuring the change of the continuous object state $(\Delta\mathbf{f}_o)^b$. Given the set of state changes $\{(\Delta\mathbf{f}_o)_j^b\}$, the robot can perform unsupervised clustering in order to identify relevant dimensionality reductions ϵ^b of the state information. In order to be able to identify future effect categories, a classifier $\text{Predict}^b : \mathbf{f}_o \mapsto \epsilon^b$ is trained for each behaviour. In consequence, the estimated object category is given by $S_o = (\text{Predict}^{b_1}(\mathbf{f}_o), \text{Predict}^{b_2}(\mathbf{f}_o), \dots, \text{Predict}^{b_M}(\mathbf{f}_o))$. Moreover, the same procedure is done for behaviours involving multiple objects, e.g. *stacking*, and multi-object effect categories are learned as well.

4.2 Bootstrapping complex skills by previous experience

In a final step, the idea of learning dimensionality reductions from interaction is done for more complex actions like multi-object actions. The basic idea is to relate pairs of single-object categories (S_{o_1}, S_{o_2})

to multi-object effect categories, e.g. ϵ^{stack}. For example the single-object categories S_{o_1}, S_{o_2} might contain the effect categories ϵ_{inside} and $\epsilon_{\text{on\_top}}$ denoting the finger being inside a hollow object and the finger stopping on top of a rigid object respectively. In such a case stacking an object o_1 on top of an object o_2 with effect category ϵ_{inside} is likely to cause o_1 being inserted into o_2. The effect category ϵ_{inside} is again created by clustering visual features with data acquired from interacting with different pairs of objects. Again, a classifier is trained in order to predict the effect category given two objects; however, in this case it is not trained on raw image features but on the single-object categories S_{o_1}, S_{o_2} and their relations R_{o_1,o_2}, R_{o_2,o_1}. This keeps the learning problem tractable and prevents a state space explosion. Following this paradigm, logical rules modelling object relations of the form $\{(S_{o_1}, S_{o_2}, R_{o_1,o_2}, R_{o_2,o_1}) \rightarrow \epsilon^{\text{stack}}\}$ can be represented in symbolic notation like STRIPS. The symbol R_{o_1,o_2} denotes relations between objects like $o_1 \xrightarrow{\text{smaller}} o_2$ and are hand-crafted by the designer and later automatically detected by the system. This way, a logic planner can be asked to build towers of objects or to insert objects into each other.

4.3 Analysis

The method described in the previous section is quite different in nature to the approaches given in Sects. 2 and 3. Instead of slowly performing a transition from rudimentary brute-force skill learning to higher-level planning, high-level symbolic planning is learned directly. The level of abstraction is much higher compared to the autonomous playing. This enables the robot to generate complex plans for tasks that were never done before, e.g. building a tower of minimal/maximal height. Moreover, this also works for objects that were never used together for multi-object behaviours like stacking. This approach even works objects that were never seen before if their object categories can successfully be detected. This is possible because reasoning is based on single-object categories abstracted from the sensor data, instead of using the raw sensor data directly. Therefore complex object manipulations can be learned for unseen objects much faster by reusing the learned abstract object categories compared to performing the clustering of effect categories on visual features directly.

However, the direct bottom-up learning of high-level symbols comes at a cost, as it requires a carefully-designed setup for the acquisition of the symbolic world model. For example, identifying single-object effect categories requires a large amount of data with a well-prepared environment for each execution of a behaviour. This is the reason why the learning of effect categories was done mostly in simulation, which again requires some effort to design a simulated model of the environment.

5. Future challenges

In this work we investigate the problem of applying modern machine learning technologies to autonomous robotics. We pointed out that even though powerful supervised learning techniques are available, they cannot directly be applied to robotics. The required large amount of training data needs to be gathered autonomously in order to design robots that can act in unstructured environments.

Developmental robotics is one way to design robotic systems that addresses the problem of autonomous acquisition of the required data. The expressive power of the involved approaches, and therefore the complexity of the solvable tasks, shifts from simple to complex in a developmental paradigm. At each stage, this keeps the search space small enough to keep the problem solvable, and increases the complexity after certain basics have been acquired.

We presented two tracks of work, (i) robotic playing (Sects. 2 and 3) and (ii) bottom-up learning of high-level symbols (Sect. 4). Approach (i) starts from low-level, simple sequencing of behaviours and slowly develops goal-driven planning capabilities. In this setting, the level of autonomy is high and the required interaction with a human supervisor is very limited. Approach (ii) on the other hand reaches a much higher level of abstraction by using simulated scenes and well-founded priors at the cost of a reduced autonomy during learning.

We believe that one of the future challenges in autonomous robotics will be to combine such methods to provide an integrated life cycle of autonomous robots. This life cycle should start from low-level paradigms such as approach (i), and should lead to high levels of capability such as approach (ii). This might require additional, intermediate developmental stages like learning more sophisticated environment models than those presented in Sect. 3. From these detailed models powerful abstractions such as the high-level symbols introduced in Sect. 4 can be learned. This could involve mimicking typical human-like concepts such as *intuition* in order to create abstractions from strong environment models.

References

1. Aristotle (−369 BC): Theaetetus.
2. Asada, M., Hosoda, K., Kuniyoshi, Y., Ishiguro, H., Inui, T., Yoshikawa, Y., Ogino, M., Yoshida, C. (2009): Cognitive developmental robotics: a survey. IEEE Trans. Auton. Ment. Dev., 1(1), 12–34. doi:10.1109/TAMD.2009.2021702.
3. Briegel, H. J., De las Cuevas, G. (2012): Projective simulation for artificial intelligence. Sci. Rep., 2, 400, EP
4. Fukushima, K. (1988): Neocognitron: a hierarchical neural network capable of visual pattern recognition. Neural Netw., 1(2), 119–130.
5. Hangl, S., Dunjko, V., Briegel, H., Piater, J. H. (2017): Skill learning by autonomous robotic playing using active learning and creativity. CoRR. 1706.08560.
6. Hangl, S., Stabinger, S., Piater, J. (2017): Autonomous skill-centric testing using deep learning. In IEEE/RSJ international conference on intelligent robots and systems.
7. Hangl, S., Ugur, E., Szedmak, S., Piater, J. (2016): Robotic playing for hierarchical complex skill learning. In IEEE/RSJ international conference on intelligent robots and systems.
8. Hubel, D. H., Wiesel, T. N. (1968): Receptive fields and functional architecture of monkey striate cortex. J. Physiol., 195(1), 215–243.
9. Izhikevich, E. M. (2003): Simple model of spiking neurons. IEEE Trans. Neural Netw., 14(6), 1569–1572.
10. Kahn, G., Villaflor, A., Pong, V., Abbeel, P., Levine, S. (2017): Uncertainty-aware reinforcement learning for collision avoidance. CoRR. 1702.01182.
11. LeCun, Y., Bengio, Y., Hinton, G. (2015): Deep learning. Nature, 521(7553), 436–444.
12. Levine, S., Pastor, P., Krizhevsky, A., Ibarz, J., Quillen, D. (2017): Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection. Int. J. Robot. Res., doi:10.1177/0278364917710318
13. Lungarella, M., Metta, G., Pfeifer, R., Sandini, G. (2003): Developmental robotics: a survey. Connect. Sci., 15(4), 151–190.
14. McCarthy, J., Minsky, M. L., Rochester, N., Shannon, C. E. (1955): A proposal for the Dartmouth summer research project on artificial intelligence. AI Mag., 31(4), 12. 27, 2006.
15. Piaget, J. (1952): The origins of intelligence in children. New York: Norton.
16. Sadeghi, F., Levine, S. (2017): Cad2rl: real single-image flight without a single real image. In Robotics: science and systems.
17. Schmidhuber, J. (2015): Deep learning in neural networks: an overview. Neural Netw., 61, 85–117.
18. Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., et al. (2016): Mastering the game of go with deep neural networks and tree search. Nature, 529(7587), 484–489.
19. Stoytchev, A. (2009): Some basic principles of developmental robotics. IEEE Trans. Auton. Ment. Dev., 1(2), 122–130.
20. Sutton, R. S. (2001): Verification, the key to AI. Online essay. http://www.cs.ualberta.ca/sutton/IncIdeas/KeytoAI.html.
21. Tobin, J., Fong, R., Ray, A., Schneider, J., Zaremba, W., Abbeel, P. (2017): Domain randomization for transferring deep neural networks from simulation to the real world. CoRR. 1703.06907.
22. Turing, A. M. (1950): Computing machinery and intelligence. Mind, 59(236), 433–460.

23. Ugur, E., Piater, J. (2015): Bottom-up learning of object categories, action effects and logical rules: from continuous manipulative exploration to symbolic planning. In IEEE international conference on robotics and automation, ICRA (S. 2627–2633). New York: IEEE Press.
24. Weng, J. (2004): Developmental robotics: theory and experiments. Int. J. Humanoid Robot., 1(02), 199–236.
25. Yu, W., Liu, C. K., Turk, G. (2017): Preparing for the unknown: Learning a universal policy with online system identification. CoRR. 1702.02453.

Authors

Simon Hangl
is a Ph.D. student at the Intelligent and Interactive Systems (IIS) Group of the University of Innsbruck, Austria, where he also received his M.Sc. degree in Computer Science. Before starting his position as University Assistant at IIS, he worked as researcher at the Semantic Technology Institute (STI) Innsbruck. He was involved in 4 EU-FP7 projects in the areas of semantic technology and robotics. He is interested in developmental and cognitive robotics and complex robotic object manipulation.

Emre Ugur
is Assistant Professor in the Computer Engineering Department of Bogazici University, Turkey. After receiving his Ph.D. in Computer Engineering from Middle East, he worked at ATR Japan as a researcher (2009–2013), at the University of Innsbruck, Austria, as a senior researcher (2013–2016), and at Osaka University as specially appointed Assistant Professor (2015, 2016). He is interested in developmental and cognitive robotics, affordances and intelligent and adaptive manipulation.

Justus Piater
is Professor for computer science at the University of Innsbruck, Austria, where he leads the Intelligent and Interactive Systems Group. He holds a M.Sc. degree from the University of Magdeburg, Germany, and M.Sc. and Ph.D. degrees from the University of Massachusetts Amherst, USA, all in computer science. Before joining the University of Innsbruck in 2010, he was a visiting researcher at the Max Planck Institute for Biological Cybernetics in Tübingen, Germany, Professor of computer science at the University of Liège, Belgium, and a Marie-Curie research fellow at GRAVIR-IMAG, INRIA, Rhône-Alpes, France. His research interests focus on visual perception, learning and inference in sensorimotor systems. He has published more than 160 papers in international journals and conferences, several of which have received best-paper awards, and currently serves as Associate Editor of the IEEE Transactions on Robotics.