

Learning Object Relationships which determine the Outcome of Actions

Severin Fichtl¹, John Alexander¹,
Dirk Kraft²,
Jimmy Alison Jørgensen²,
Norbert Krüger^{2*}, Frank Guerin^{1†}

¹ University of Aberdeen,
King's College, AB24 3UE Aberdeen,
Scotland

² University of Southern Denmark,
Niels Bohrs Allé 1, DK-5230 Odense M,
Denmark

Received 15-12-2012

Accepted 27-03-2013

Abstract

Infants extend their repertoire of behaviours from initially simple behaviours with single objects to complex behaviours dealing with spatial relationships among objects. We are interested in the mechanisms underlying this development in order to achieve similar development in artificial systems. One mechanism is sensorimotor differentiation, which allows one behaviour to become altered in order to achieve a different result; the old behaviour is not forgotten, so differentiation increases the number of available behaviours. Differentiation requires the learning of both sensory abstractions and motor programs for the new behaviour; here we focus only on the sensory aspect: learning to recognise situations in which the new behaviour succeeds. We experimented with learning these situations in a realistic physical simulation of a robotic manipulator interacting with various objects, where the sensor space includes the robot arm position data and a Kinect-based vision system. The mechanism for learning sensory abstractions for a new behaviour is a component in the larger enterprise of building systems which emulate the mechanisms of infant development.

Keywords

Developmental Artificial Intelligence · Vision · Infant Development · Means-end Behaviour · Learning Preconditions

1. Introduction

In the period from six months of age through to two years, human infants undergo significant development in their skills and understanding relating to physical world objects and their manipulation. At six months they mostly deal with only one object at a time, performing simple actions such as sucking or banging; by two years they are capable of solving relatively complex problems which require them to put multiple objects in a spatial relationship, for example using simple tools. We are interested in building artificial systems which could mimic the mechanisms underlying infants' development of these skills and thereby achieve some understanding of the physical world. In a survey of this development, we have outlined six of these mechanisms [1], and the present paper (focussing on one mechanism) is part of the endeavour to create a complete working implementation of these mechanisms in an agent which could exhibit autonomous development through embodiment in a robot.

Observations of infants show that, at any particular age, they possess a repertoire of behaviours or manual skills which they apply to various objects or surfaces they encounter [2, 3]. Each such behaviour could be seen as roughly analogous to a planning operator in Artificial Intelligence, because there are situations which make them likely to be executed (like the precondition of a planning operator), and expected effects (postcondition), as well as some motor control program describing the behaviour executed. Piagetian theory calls such units schemas,

and we use this terminology here; other psychologists have similar units called "sensorimotor processes" [4], "skills" [5], or "perception-action routines" [3]. The repertoire of schemas which infants possess by two years is much larger and more sophisticated than the repertoire they have at six months. The focus of our work is on how new schemas are acquired. Within this problem there are the problems of identifying when a new schema should be created, and then learning new precondition, postcondition, and motor program for the new schema. In this paper, we focus on learning the precondition for a new schema. This is a particularly interesting problem in the case of "means-end behaviours"; these are problem solving situations where the infant cannot immediately achieve its goal, and so must sequence two actions, where the first facilitates the next [6], for example a toy may be obstructed by a box, and the infant may need to push the box out of the way before being able to take possession of the toy. Figure 1 illustrates such a situation.

Piaget believed that it is through learning means-end behaviours that infants begin to learn about important spatial relationships between objects [7]. The precondition of a schema (for a means-end behaviour) must capture the spatial relationship between objects which determines where the behaviour works or does not work. In learning preconditions the infant is learning new important abstractions over its sensor space. This can change how an infant understands a scene because the infant can begin to see things at a higher level of abstraction, noticing precisely those spatial relationships which are important in determining what object manipulations are possible (by itself or other agents). This is an important part of the development of an understanding of the world.

Both, sensory and motor aspects are crucial aspects of infancy and ongoing development in general. These aspects are very much intertwined in humans. In computational systems, sensory and motor

*E-mail: norbert@mimi.sdu.dk

†E-mail: f.guerin@abdn.ac.uk



Figure 1. This figure on the left illustrates a situation where an infant desires a toy which is obstructed by a box. The figure on the right shows the same situation after the infant removed the box and is now able to retrieve the desired toy.

aspects can also influence and bootstrap each other; however, unlike biological systems, artificial systems can choose to focus on either one independently. In this work we focus on the sensory part, but we also investigate the motor side in parallel in our research group.

In this work we experimented with the following two means-end behaviours: (i) pushing aside an obstacle to see if it would convert an obstructed scenario to an unobstructed one; (ii) pulling a supporting object to bring the supported object into reach (i.e. to bring the object on top into reach as in Fig. 2). In each case, we investigate how well we can learn the preconditions describing the spatial relationship among a pair of objects which determines whether or not the means-end behaviour will work. We did this using an agent which controls a simulated robot arm with 6 Degrees of Freedom in a physically realistic 3D World, and a Kinect-based vision system. For this vision system we also simulate the Kinect, including the noise of real Kinect devices. Of course it would be relatively easy for a programmer to simply code in the required spatial relationship so that an agent would not have to learn it, however our aim is to construct an agent which can learn world knowledge for itself. Such an agent should hopefully be able to extend its own knowledge, and learn things the designer might not have foreseen the need for. In particular we would like to endow robots with the capability to learn about important spatial relationships (determining the success of manipulation actions) which the designer might not have foreseen the need for. Our analysis of infant development (focussing on interaction with the physical world) [1] suggests that infants do possess a differentiation mechanism for spawning new schemas, and we see this capability as an essential component of any agent which would be able to display ongoing autonomous development.

To the best of our knowledge, this is the first attempt to build an artificial system to tackle the problem of the acquisition of preconditions for the two means-end behaviours above (removing the obstacle, and pulling the support). We believe that getting robots to do the kinds of tasks which infants do is an important area in developmental robotics because these are tasks which we know are part of a developing trajectory, leading to more sophisticated tasks which build on them. Our results suggest that (1) it is possible for a robotic system to autonomously learn its own sensor abstractions for new behaviours, and (2) that rapid learning of these abstractions can be facilitated if the agent adopts an active learning approach to selecting new examples.

There is a closely related and recent work by Rosman and Ramamoorthy [8] which learns spatial relationships between objects, such as “on” and “adjacent”. However this is learning the spatial relationships in a human-supervised fashion, using humans to pre-label a set of scenes

with the object relations seen in the scene. We differ in the philosophy of our approach because we believe that the robot should only learn relationships which are practically meaningful for it (e.g. those relationships that determine when a behaviour will be successful or not). In contrast to our work, [8] imposes a concept (e.g. on top) which a human believes to be useful for a robot, and the particular instantiation of that concept is decided by the human, rather than with reference to the robot’s own action in the world. We instead believe that the concepts which will be useful for a robot are likely to be those that emerge from its own interactions with the world. This is also related to Sutton’s “verification principle” [9, 10]; the relationship our system learns is grounded in its own experience, and hence it can always revisit that and relearn it if necessary (e.g. if its manipulation ability changes, such that a different variant of the relationship is needed now). In contrast, knowledge given by a human is not verifiable by the agent, and is just given as is with no opportunity for adjusting the idea. In our system training data can be gathered in an online fashion as the robot executes actions and sees the effects, furthermore the relationships which the robot learns are not limited by those decided by a human; whenever the robot needs to find situations in which a means-end behaviour works or does not, it can begin learning a classifier that will appropriately discriminate the situations.

In Section II, we review the background literature on infants’ acquisition of means-end behaviours, to motivate our computational work. Section III gives an overview of our computational work and the experiments carried out. Section IV presents the results of our experiments. Section V discusses the implications and significance of these results, and compares with related works. Section VI concludes and outlines directions for future work.

2. Motivation

Young infants start life with a limited set of repetitive behaviours, but as they progress through the first two years their repertoire of behaviours grows rapidly [1, 2]. The initial behaviours include “rhythmical stereotypical behaviours” which are either present at birth or seem to emerge as by-products of the normal maturation of motor control circuits [11]. These behaviours include “arm waving” (flapping of the arm vertically from the shoulder), bending and extending the wrist, flexion and extension of the fingers, etc. It is surmised that these behaviours may be opportunistically used by infants for the purpose of bootstrapping further development, for example by encouraging actions which will at some point lead to interesting results. In addition to the rhythmical stereotypical behaviours, there are the basic object behaviours such as looking at an object, and grasping it, and performing basic actions such as mouthing it or banging it on a surface. These behaviours are typically well established by six months [12, p. 174],[13].

During the period from 6 to 24 months, there is a rapid growth in the infant’s repertoire of behaviours. Following the terminology of Piaget [14], we can describe these infant behaviours using the term *schemas* which have roughly the same role as planning operators in typical AI systems. A schema has a precondition (describing situations where it is expected to be applicable), an action (or motor program), and a prediction (describing the expected result). The growth in the infant’s behavioural repertoire can then be described as the addition of new schemas to its repertoire. The addition of schemas may be driven by some underlying developmental mechanisms [1], one of which is *Sensorimotor differentiation*: When an existing schema is executed and it produces an unexpected result, this can start a process which attempts to discover how to reproduce this new result; the process must change both the old motor program as well as the precondition and pre-

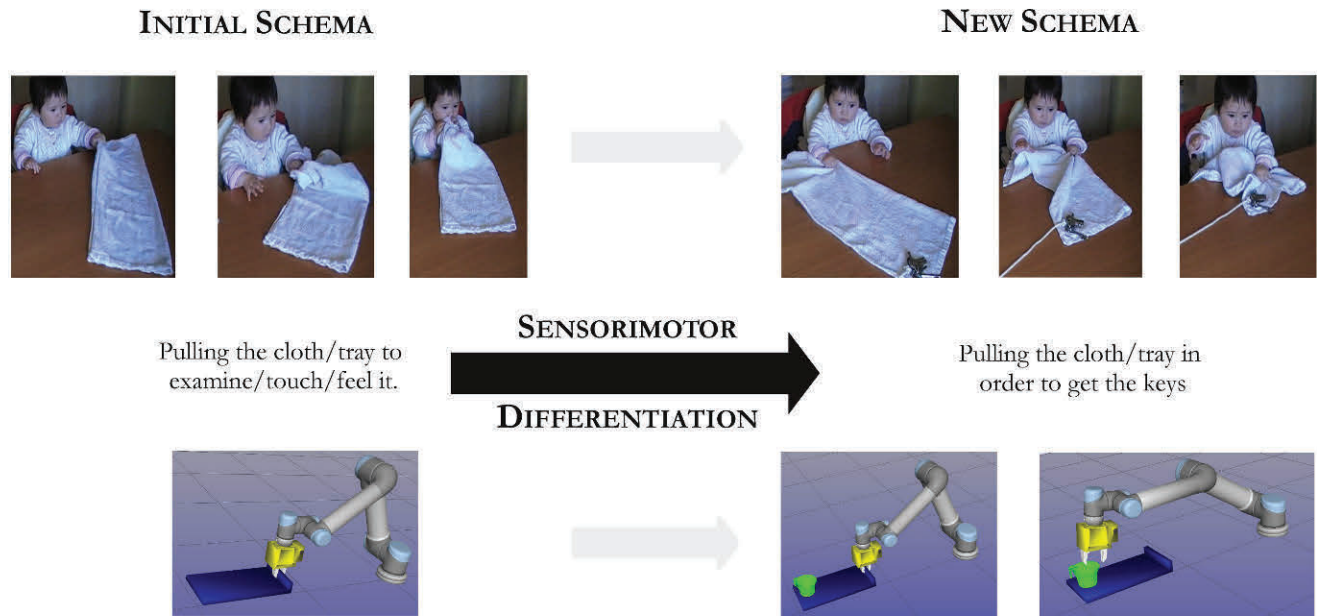


Figure 2. This figure illustrates how a new behaviour can be acquired by the differentiation mechanism. The original behaviour (initial schema) simply pulls a cloth/tray in order to bring it close. The new behaviour pulls the cloth/tray in order to bring the item supported by it closer. The new schema acquired will need to adjust its motor behaviour, and also to learn the situations in which this new behaviour can be expected to work.

diction of the old schema. Sensorimotor differentiation thus describes how one schema can spawn a new schema.

An example of this process resulting in the emergence of a new schema has been studied by Willatts [6], who studied the acquisition of the schema for pulling a cloth in order to bring an object resting on it within reach (where grabbing this object is the goal). Figure 2 illustrates this process. Willatts showed that during the 6 to 8 months period there is a gradual transition: initially the infant sees the goal (object out of reach) and means (cloth), and does not know the possibility for retrieval, so the infant plays with the means object (cloth) for its own sake, but in grabbing the cloth the object is brought closer. This accidental retrieval gradually becomes intentional. Furthermore, by 9 months it was shown that infants can adjust the means action (cloth pull) as appropriate to the goal, in situations where the goal may be far or near. The new pulling behaviour becomes quite different to the original retrieval behaviour; the cloth will be pulled further, even behind the infant (this is the learning of a new motor program); furthermore, the infant learns to discriminate the situations in which this cloth pulling is likely to work (this is the learning of a new sensor abstraction).

A further example which we have investigated concerns removing an obstacle (see also Piaget [2]): A young infant has a behaviour for waving an object back and forth on a table surface. At some later point, this behaviour becomes differentiated to produce a behaviour for deliberately displacing an object to one side in order to retrieve a visible toy behind it (see Figure 1). Again, the new behaviour is different as it becomes tailored to the new goal, and the situations where it is likely to work are learnt.

Piaget believed that it is through learning means-end behaviours such as these that infants begin to learn about important spatial relationships between objects [7]. This is an important part of the development of an understanding of the world because it helps an infant to understand

what it observes at a higher level of abstraction, noticing precisely those spatial relationships which are important in determining what object manipulations are possible (by itself or other agents). The importance of these acquisitions for cognitive development motivates our interest in attempting to build artificial systems which could autonomously make similar acquisitions.

3. System Overview

This section first briefly describes how the system detailed in this paper fits into the larger enterprise of building artificial systems which acquire new behaviours following a similar developmental trajectory to infants (Sec. 3.1). Following this we detail our simulated robotic arm, vision system, and experimental setup (Sec. 3.2, 3.3), followed by the experiments run and the learning approach used (Sec. 3.4).

3.1. Overview of the Developing System

The work reported in this paper is just one component in our developing system, which includes mechanisms of development allowing new sensorimotor schemas to be added to the infant's repertoire. We will give a brief overview in order to illustrate where the precondition learning fits into the full system. We first explain some terms: an "action" in our system is a skilled motor program which typically achieves some goal, for example, reaching for and picking up any of various objects, or pulling an object. A "relationship" between objects in this paper refers to a spatial arrangement of the objects; in particular we focus on the spatial relationship of one being on-top of the other such that pulling the lower one will make the upper one also move. Note that we are not

interested in the concept “on top” as a human adult might think of it, rather we are interested in the spatial relationship which exists if the two objects are to move together, so it is a very practical relationship which is grounded in a particular manipulation action. In our system a schema $\langle prec, M, pred, G \rangle$ has a precondition $prec$, a motor program M , a prediction $pred$, and also a postcondition G (which can be viewed as the goal this individual schema is trying to achieve). A postcondition (or goal) is necessary so that some schemas know when to terminate, and also because new schemas typically start life with a new goal and have to tailor their motor and sensor aspects to appropriately serve the goal. On a higher level in the framework, these schemas can be used and put into a sequence by a planner in order to achieve complex high-level goals. When considering such a sequence of planned actions, we would refer to the “goal” of an individual schema as a “subgoal”. The motor program controls the robot arm to achieve this subgoal. The simulated infant maintains a library of schemas some of which are provided initially, and some of which may subsequently be added to or modified. The initial schemas include behaviours such as look-at, reach, grasp, drop, etc. This is not to suggest that these behaviours are innate in infants, they are just starting behaviours for our system; we do not want our system to have to learn everything which an infant learns, instead we want it to start at a state with some competences/behaviours and then learn from there.

The agent analyses the current state of the world and produces a planning tree of future possibilities for its own actions; this tree is currently limited to a depth of three, but in the future this could be changed to a more dynamic depth based on the accuracies of the schema predictions. The planning tree shows possible chains of actions. To find these, when analysing the current state, the agent checks for each available schema whether its precondition is met, i.e. whether it is expected to achieve its goal when executed in the current state, and adds it to the tree if it is expected to be successful. The agent then uses a schema’s predictor to predict the next state of each added schema and adds a second layer of actions by analysing the predicted states. This is repeated until the depth limit is reached. The first level of the tree now contains every schema whose precondition is satisfied currently; the second level contains every schema which is predicted to be executable thereafter, and so on.

The agent, hence, understands the scene in terms of possibilities with regard to existing schemas. Figure 3 illustrates the routine the agent continuously loops through:

1. Computing the tree of executable schemas, where executable means they are expected to have a chance to achieve their goal.
2. Selecting one of the first level schemas for execution, where this selection can be random or guided by intrinsic motivation or a planning algorithm trying to achieve a not directly achievable goal.
3. Updating the agent’s knowledge based on the results of that execution (which may include the *harvesting* of a new schema).

If the result of the execution is a slight variation on the predicted result (*slight* here means that the subsequent possibilities in the tree are not affected by this change), then the prediction of that schema is updated. However, if the result of the execution causes an unexpected change in the tree, i.e., changes the future schema execution possibilities, then a *harvesting* process is initiated which creates a new schema (the term ‘harvesting’ is borrowed from Chaput [15]). This can be illustrated with the example of the support: before the system knows about the means-end behaviour of the support it may accidentally pull a supporting object and thereby cause a formerly unreachable object to come into reach. This will cause an unexpected change in the tree, because there is

now a new object in reach on which various actions (such as grasping) could be performed. hence, the criterion for harvesting has been met. A preliminary version of this model has been made in a masters thesis [16].

Harvesting is the first step in differentiation: it creates a copy of the schema which had just been executed, but gives it a different goal (to achieve the unexpected result). A goal can be to bring about a certain (set of) feature(s) in the state space, e.g. to touch or grasp an object. A more difficult goal would be to enable another schema, e.g. a goal could be to make an object reachable (which is the case in the harvesting of a schema for the support means-end behaviour). The old schema gets assigned the goal to achieve the previous standard outcome. Based on the new goal, the new schema then tries to modify its precondition so as to capture the situations in which the new goals can be reliably achieved; this is the differentiation of the precondition which is the focus of this paper. The motor program and prediction also need to be differentiated (but this is not tackled in this paper). For learning the motor program, the concept of *Goal Babbling* [17] may be useful as it allows for a limited type of exploration of motor programs, which is directed towards goals.

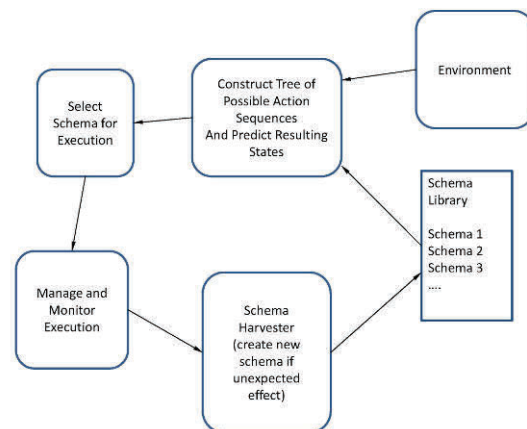


Figure 3. Illustration of the routine the agent continuously loops through: I) Based on the current state of the environment, the agent constructs a tree of possible actions II) The agent selects a schema for execution III) The agent executes the selected schema IV) The harvester module decides whether or not to create a new schema, based on the result of the last execution V) If a new schema was created it is added to the agent’s schema library

In the current work, we focus on improving and evaluating this process of adjusting the precondition of the new schema based on ongoing experiences generated from its execution. The problem of deciding when to create the new schema was found to be the easier part in our previous work, and is not addressed here.

3.2. The Simulated Environment

As our robot we use a simulated 6 degrees of freedom arm mounted on a table with a two finger gripper as its hand in the simulator RobWork-Sim [18, 19]. In our experiment, we use 3 different household objects (see Fig. 4):

1. A simple coffee cup
2. A cereal Box
3. A tray (with a protrusion to facilitate pulling)

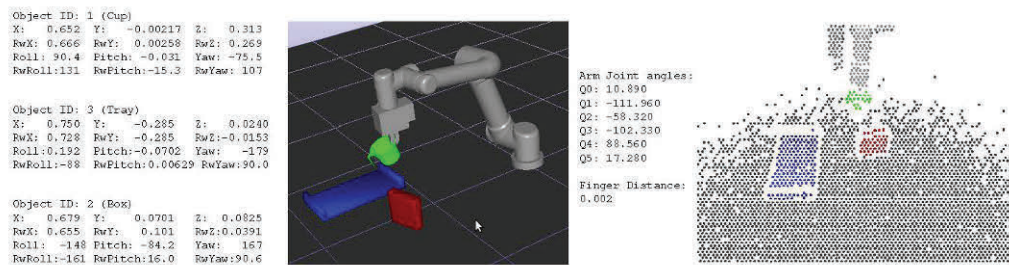


Figure 4. Illustration of the simulated world (screenshot on the left) containing the robot arm and the 3 household items used in experiments (Cup, Cereal Box and Tray); the camera is positioned to the left of this image, producing the textlets which can be seen in the rightmost screenshot above. The adjoined text shows the complete state space of the robotic system. Values prefixed with "Rw", such as "RwX" come direct from RobWorkSim, whereas values such as "X" are the result of the vision system operating on the textlets.

Our agent senses its internal state such as robot arm joint angles and calculates the position and orientation of its hand based on the joint angles. The full state space consists of this internal state space and the external statespace describing the world the robot is situated in. The internal state space has 13 Dimensions. The first 6 variables describe the joint angles of the robot arm. The second 6 variables describe the position and orientation of the hand of the robot arm. These 6 variables consist of 3 variables for the Cartesian position and 3 variables for the orientation of the hand in Roll, Pitch and Yaw values. The second 6 variables are calculated from the first 6 values. The last variable in the internal state space equates to the openness of the hand's fingers. Figure 4 shows this state space.

In our experiments, we first learnt preconditions using object position data directly from the RobWork simulator (i.e. perfectly accurate data), then we later used the less perfect data coming from our vision system, to see how well the learning system would cope with a more realistic noisy input.

3.3. The (Simulated) Vision System

Our robotic system uses a Kinect-based vision system [20] developed at SDU, to extract information about objects in the scene.

A Kinect is a 3D scanner camera system developed by Microsoft as motion sensing input device for the Microsoft game console Xbox 360. It is a popular alternative to expensive stereo camera systems and provides good results in close range applications with up to 3 meters distance from the Kinect device [21].

The Kinect system projects an infrared image (invisible for human eyes) over the scene and then takes a picture of the scene. Using its knowledge of the projected infrared image it is able to calculate an accurate depth map based on the distortion of the infrared image. This depth map describes the distance from the camera of each point of the surfaces visible to the camera system. Using the picture of the scene and the depth map, our vision system calculates a 3D point cloud as it is common amongst state of the art vision systems[22].

Based on this 3D point cloud and the colour information of the scene, our vision system creates surface patches as shown in Figure 4 on the right hand side. There are different layers of surface patches. We only use basic layer with surface patches which we call textlets. These textlets describe the surface of the scene with additional information, e.g. not only position in the space, but also the orientation and colour of the surface [23].

Because we worked with a simulator, we also simulate the Kinect sys-

tem, and included the noise of real Kinect devices. This gives us data about the depth to the objects in our 3D scene just as we would have obtained from a real Kinect looking at a real scene with 3D objects. The data from the simulated vision system is hence more noisy and less accurate than the perfectly accurate data provided by taking the locations of objects directly from the simulator.

Using different coloured objects for segmentation purposes, the vision system is able to recognise up to five objects and extract their centre of gravity positions in relation to the robot arm, and their orientation. This is achieved by running PCA over each object's textlet-based representation. Both, position and orientation of an object are described by 3 variables each. These variables are X, Y and Z for the position and Roll, Pitch and Yaw for the orientation in space. Together with the 13 internal state variables described above in Section 3.2, this gives a 43 Dimensional state space, where the 13 internal state variables describe the robot itself and the $5 * 6 = 30$ external state variables describe the configuration of the objects in the robots view.

Note that none of the values returned by the vision system is perfectly accurate. Centre of gravity is approximate, and orientation works best for a long object, but does not give much useful information for a small roundish object like a cup. The difference between the state space based on perfect RobWork data and the imperfect Vision System data can be clearly seen in Fig. 4 (by comparing a value with its corresponding "Rw" value written directly below it). In the remainder of this paper, when talking about "RobWork data", we mean the state space based on the perfect information provided by the simulator and similarly, with "vision based data" we mean the state space which is based on the data provided by the noisy Vision system. Part of the challenge of learning preconditions is to work with this imperfect data.

3.4. Experiments

3.4.1. Example 1: The Support

Our first example is where the robot pulls the tray (means action) in order to bring an object supported by it (in this case the cup) into reach (goal). In some cases this will not work, because the cup (or whatever supported object is desired) might not be fully on the tray. The learner's task is to recognise the situations where the means action is effective. Figure 5 illustrates a scene where the desired cup is initially out of reach, but the robot successfully brings it into reach by pulling closer the support on which the cup is standing. The robot then can successfully grasp the cup.

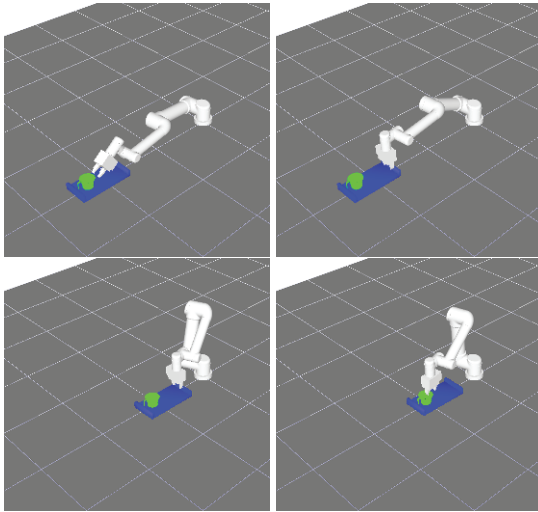


Figure 5. This figure illustrates how the robot brings into reach the initially out of reach cup by pulling closer the tray on which the cup is standing.

3.4.2. Example 2: The Obstruction

In “the obstruction” scenario there are two objects in the scene; the cereal box and the cup. The cup is the object that the agent wants to grasp, but the cereal box is obstructing the reach (but not the view). The desired object is therefore not reachable. In specific situations, e.g. as illustrated in Figure 6, pushing one object sideways (the box) may render the other object (the cup) graspable. The learner’s task here is to recognise situations where pushing the obstruction sideways will allow the desired object to be grasped.

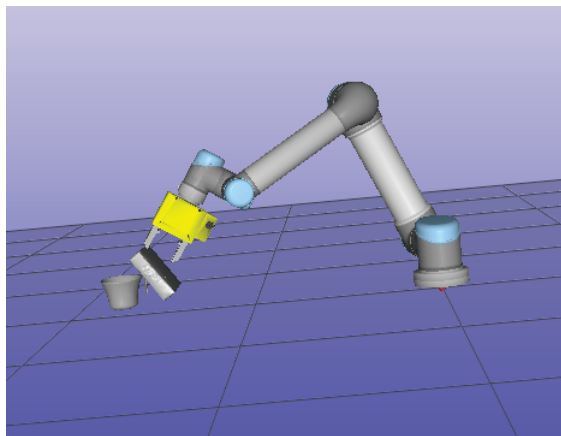


Figure 6. Illustration of one object (box) obstructing another (cup)

3.4.3. Pretraining the preconditions of schemas

Before learning the above preconditions, we need to endow the system with some initial schemas. These preliminary schemas are essential

because, for example, if one considers the task of gathering training data for recognising the support situation, the system needs to pull the support, and the resulting situation needs to be labelled; i.e. we need to know if the desired object was *reachable* after pulling the support; hence the *reachable* precondition needs to be learnt first. The motor programs of these initial schemas were hand-coded, while the pre- and postcondition were trained. These initial schemas are purely for single actions and are not trained with any knowledge of their applicability in means-end combinations.

We did this for three schemas:

- (1) Grasp(o): reaching and grasping an object o when unobstructed (e.g. the cup),
- (2) Push(o): reaching and pushing an object o (e.g. the box),
- (3) Pull(o): reaching and pulling an object o (e.g. the tray),

where o is a parameter for the object.

We also trained one handmade abstraction to recognise “the obstruction”; this recognises if one object is obstructed by another (e.g. the cup obstructed by the cereal box). The three schemas were trained by trying out each schema on a set of randomised environments.

The Grasp(Cup) schema precondition was trained on 3000 training examples out of which 1500 were successful reaches for the cup and 1500 were unsuccessful reaches. The trained schema precondition was tested on a validation set with 7862 examples (3931 successful/3931 unsuccessful). On this validation set the schema precondition achieved a classification success of 99.16% correct classifications using the RobWork data for training. Using Vision based data for learning the classification success was 99.13% correct classifications.

The Push(Box) schema precondition was trained on 2200 training examples (1642 successful/558 unsuccessful). The validation set contained 376 examples (284/92). On this validation set the schema precondition achieved a classification success of 89.63%. Push(Box) was only trained with RobWork data. The relatively low performance is due to fewer training examples.

The Pull(Tray) schema precondition was trained on 3000 training examples (1500/1500). The validation set contained 4380 examples (2190/2190). On this validation set the schema precondition achieved a classification success of 92.51% correct classifications using the RobWork data for training. Using Vision based data for learning the classification success was 91.69% correct classifications. Note that this is merely recognising situations where the tray can be pulled, and says nothing about the means-end behaviour of “the support”.

The obstruction recognising abstraction succeeded in about 93% of its classification trials. It was trained on 5879 examples (3329/2550) and tested against 997 examples (898/99). Again, this was only done with RobWork data, not Vision based data.

All Precondition classifiers and abstraction recognisers were trained using simple feed forward Neural Networks. They all had 6 input neurons and one hidden layer with 10 Neurons. The output layer was a single Neuron which was trained to output either 0 or 1 for predicted fail/success accordingly. The neurons used Sigmoidal activation functions and the training algorithm used was RPROP using cross entropy error instead of the usual root mean squared error.

3.4.4. Generating Data and Training

Experiment 1 is with the support: We generated 6,453 positions for support and cup, and then pulled the support to see if the cup would come into reach. The outcome was that 83% of examples were negative (the cup did not come into reach), and 17% positive.

Experiment 2 is with obstructed scenarios: We generated 16,361 scenarios with random positions for the cup and box, and then pushed the

box (obstacle) to see if it would convert an obstructed scenario to an unobstructed one. The outcome was that 94% of examples were negative (the cup was still unattainable after executing the push action), and 6% positive. This is probably qualitatively in line with typical real-world experiences, in that positive examples of configurations where the pushing of one object makes another accessible are relatively rare, for everyday toys and household objects in everyday spatial relationships.

Using this labelled data, we trained classifiers to predict situations where a "remove obstruction" or "pull support" means-end behaviour would work. The support classifier was separately trained twice (to compare results): once with the direct RobWork data for object position and orientation, and once using the Vision System's data. In the experiments with RobWork data, the preconditions (for such things as "reachable", see Sec. 3.4.3) learnt from RobWork data were used. For the experiments with vision data, the preconditions learnt from vision data were used. The obstruction classifier was only trained on direct RobWork data. We initially experimented with both logistic regression and neural networks for classification. The neural networks proved to be vastly superior and so we did not produce graphs for the logistic regression with all different training schedules. (The advantage of logistic regression is speed of training.)

We experimented with a range of networks and determined that a single hidden layer of 9 nodes was optimal. The training algorithm was RPROP [24, 25]. A validation set of 4380 examples (50% positive, 50% negative) was randomly selected. Existing work [26] shows that learning can be problematic when negative examples greatly outnumber positives; for this reason we also randomly selected a balanced set (50% positive, 50% negative) of training examples from the entire data.

3.5. Certainty Based Curiosity

The problem of learning of preconditions grows in difficulty with the size of the state space; hence it is of considerable difficulty in our 3D world with a 6 DOF arm. Given a very large set of training data, the determination of the precondition might be facilitated, but infants seem to learn from relatively few examples. For this reason we have looked at techniques which the program can use to select the training data which might be most useful. It does seem likely that infants use a similar technique because they do not select actions at random, but rather have some intrinsic motivations to prefer certain actions in certain situations. *Certainty Based Curiosity* [27] is a strategy to bootstrap the learning process so that schema precondition and prediction performances converge faster to their maxima. This bootstrapping is achieved by trying to perform the schema which is most likely to benefit from the experience. In every state each schema's precondition returns a value between 0 and 1. A 0 corresponds to the class "failure" where the schema is expected to be unable to achieve its goal when executed in the current state. A 1 corresponds to the class "success" where the schema is expected to achieve its goal when executed in the current state. Thus, the precondition implements a standard classifier, in this case with Sigmoidal output between 0 and 1. If the output of the precondition classifier lies close to 0 or 1, then the classifier seems to be certain about the outcome. E.g. 0.01 means there is no point in even trying and 0.99 means it will almost certainly work. If, however, the classifier output lies close to 0.5, then the classifier is not sure about the outcome. In fact, an output of exactly 0.5 means there is a 50% chance for either outcome, failure or success. This means the classifier is unable to predict the schema's failure or success with any confidence in the current state. In this case executing the schema is likely to generate experience which the classifier will benefit from.

Training examples were processed in batches of ten; i.e. each schedule trained with ten random examples initially, and thereafter made its

selection for the next ten based on its current classifier filtering the remaining examples. The network was trained on the examples according to the following schedules:

- Random: random selection of next samples.
- CBC (Certainty Based Curiosity): select the training examples whose current predicted classification lies in the range 0.3 to 0.7 (i.e. uncertain classification). If no training examples remain in this range a random selection is made.
- Ranked CBC: Sort all remaining samples according to how close their classification is to 0.5, and pick the next samples from the top (closest to 0.5) of the list.
- 0.2 epsilon greedy CBC: similar to ranked, but with 0.2 percent chance to pick a random sample. (0.2 was picked because in previous tests [27] this had given the best results.)

In each case an average was run over 50 complete trials (with different randomly initialised neural networks) to smooth results.

The idea behind the *Certainty Based Curiosity* approach to action selection would mean that we should place the agent in random situations and let it choose its actions guided by *Certainty Based Curiosity* (CBC). However in our work we have simply gathered all the training in advance from a random distribution of positions, and the CBC's role is to select the most "interesting" examples at each iteration of training. We expect to see that in the CBC guided runs the schemas converge to their best performance faster than in random walk runs. This was our hypothesis in advance of running the experiments; the actual results are described in the next subsection.

4. Results

Firstly we were able to learn the precondition of schemas (see Fig. 8). For the simpler schemas the precondition's accuracy approached 99%. For the more complex newly acquired means-end schema of bringing the cup into reach, the precondition's accuracy approached 81%. The support is a very difficult spatial relationship to learn (which is also true for infants [7]). The left image in Figure 7 shows an example of a scene which the precondition successfully learned to classify as a state in which pulling the support will bring the cup into reach. The right image in Figure 7 shows an example of a scene which the precondition successfully classified as a state in which this will not work.

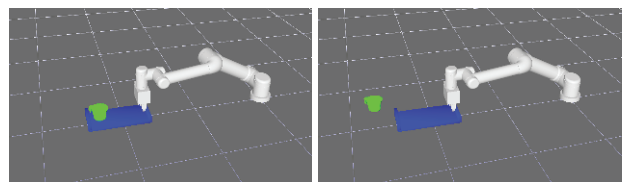


Figure 7. This figure illustrates two different cases where the precondition was successfully trained to predict whether pulling the support will bring the cup into reach. For the scene in the left image it learned that it is possible, for the scene in the right image it learned that it is not possible.

Using the realistic Vision based data to learn the precondition we were able to achieve similar accuracy as with the perfect noise-free RobWork Simulator data. Fig. 8 compares the performance of RobWork

based learning with Vision based learning for three schemas: Grasp Cup, Pull Support and Pull Support to bring the cup into reach. However it is worth noting that the direct RobWork based learning achieves high accuracy much faster than the Vision based learning, as can be seen by comparing Fig. 10 and Fig. 12, which are showing the same learning problem with the only difference being that Fig. 10 uses the vision system.

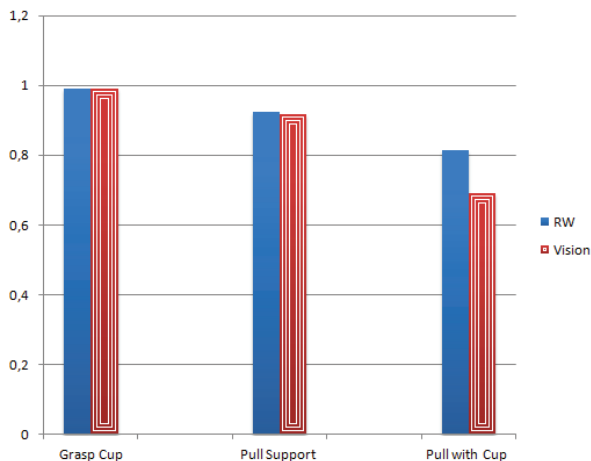


Figure 8. The Graph shows the accuracies of the preconditions for the three schemas Grasp Cup, Pull Support and bringing the cup into reach. It can be seen that Vision based learning performs almost as well as the RobWork based learning. 3000 training samples were used.

Secondly we looked at the influence of the different schedules by which training examples are selected on the learning rates of learning the preconditions. Figure 9 shows pulling the support and Figure 10 shows pulling the support to bring the cup into reach. The preconditions are first learnt with 10 random samples (which gives a performance little better than chance (50%)). The set of training samples is then step by step increased by 10 samples, using one of the different selection

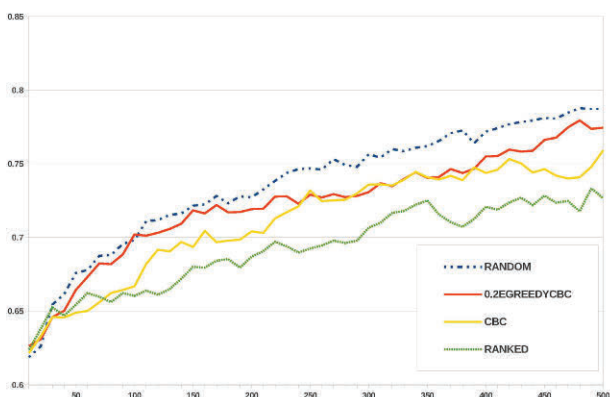


Figure 9. Learning Rates of different sample picking schedules for pulling the tray, using input data from the computer vision system. The different schedules for the four graphs are described in Section 3.5. Precondition accuracy is on the y-axis and number of samples on the x-axis.

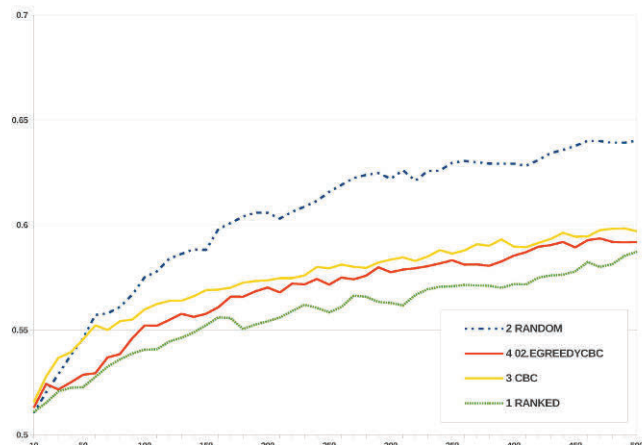


Figure 10. Learning Rates of different sample picking schedules for pulling the tray with a cup on top, in order to bring the cup into reach, using input data from the computer vision system. The different schedules for the four graphs are described in Section 3.5. Precondition accuracy is on the y-axis and number of samples on the x-axis.

The graphs show that learning preconditions with reasonable accuracy is possible from a moderate amount of training data (e.g. 500 samples). The graphs also show that learning from random samples performs better than using more directed schedules. These results are unexpected as we showed that directed schedules can perform better than random in a previous publication [27]. Figures 11 and 12 show how a directed schedule outperformed "random" in our previous experiments for two means-end schemas: Pull support to bring cup into reach and push a cereal box away to unobstruct the cup. The discrepancy is probably due to the higher noise in the data of the experiments using the vision system. Kääriäinen [28] showed that any active learner has a lower bound of $\Omega(\frac{\eta^2}{\epsilon^2})$ on the sample complexity. The Ω -notation [29] is related to the more commonly used O-notation, but where the O-notation describes an upper bound, the Ω -notation describes a lower bound (i.e. the best case when it comes to the sample complexity in active learning). The noise in the data is represented by η and ϵ describes the classification error. This means when η is large active learning can not outperform random sampling. In fact, the opposite can be the case. This is, because the most informative samples active learning is trying to find, also tend to be the most noise-prone [30]. That means that the learning algorithm has to use very noisy samples for learning and cannot even rely on the less informative but also less noisy samples from further away from the decision boundary which a random sampling would provide. This explains why the random sampling outperforms the directed sampling when using the noisy vision based data. Selecting more noisy samples would lead to worsening performance for the classifier.

5. Discussion

We have shown that learning preconditions for schemas is possible with reasonable accuracy using a few hundred training examples. We also show that with many thousands of experiences, it is possible to learn preconditions with very high accuracy. In order to try to reduce the number of training samples required we experimented with differ-

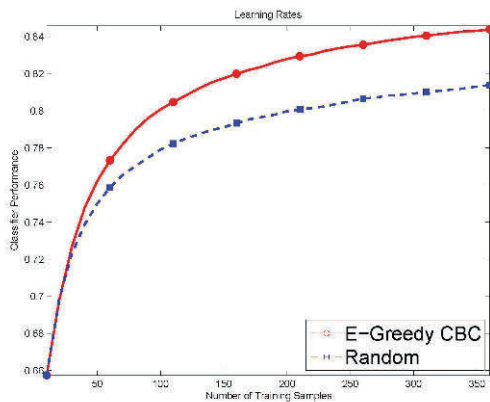


Figure 11. Learning Rates of E-Greedy and Random sample picking in the Unobstruct Cup Case using perfect input data direct from the simulator (i.e. no computer vision system).

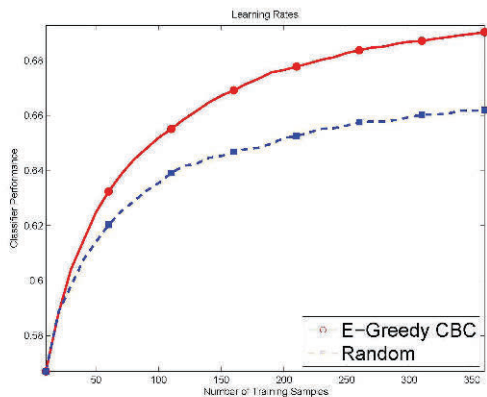


Figure 12. Learning Rates of E-Greedy and Random sample picking in the Pull Support Case using perfect input data direct from the simulator (i.e. not using computer vision system); note that this is the same graph as Fig. 10 except this one does not use the vision system.

ent active learning strategies for choosing the next training samples. Previous results using data directly from the simulator suggested that using a directed schedule produced faster learning. However, our results here using the vision system within the simulator suggest that schedule where the learner chooses randomly from the set of available experiences produces marginally better results than a directed schedule. This disparity in results is probably due to the high degree of noise within the vision data.

We are interested to discuss here the similarities and differences between what our system learns and what infants learn in the 6 to 11 months period. However, we first must point out that we are not attempting to create a model of an infant with our work. Our computational system is significantly simpler than an infant, and even if some correspondence might be achieved in what they learn, this would in no way imply that there was a correspondence in the implementation. We do not believe it would be fruitful to “replicate” the detail of the results of an infant study by building a computational system which could pro-

duce similar data points to that produced by the infants, because the implementations would be so different that the computational model would shed no light on how infants operate. However we are interested in copying the developmental trajectory of infants, and tackling similar problems in a similar ordering to infants, because we believe that infant developmental trajectories can give a useful ordering of tasks and competences from easy to hard, where subsequent ones build on previous acquisitions.

Now we compare what our system learns and what infants learn in the 6 to 11 months period. One point worth noting is that there have been only a handful of systematic studies analysing how infants acquire the means-end behaviours for the support or obstruction scenarios. While we have some studies which probe the competence at a particular age [31–33], Willatts’s [6] goes further and tracks the change during the period of transition; he studied the acquisition of the support over the 6 to 8 months period, and analysed eye-gaze (to determine if the infant intentionally pulled the support in order to retrieve the supported object, or just to play with the support). This showed that accidental retrieval gradually becomes intentional. However, there are still a lot of unknowns about the infant’s progression; e.g. it would be beneficial for a roboticist to see a systematic study of the situations where the infant classifies correctly or incorrectly, right up to 11 months. Such studies are not easy to carry out as one needs a lot of access to the infants to test them very frequently. We are left then with studies which are more anecdotal than systematic, such as Piaget’s [2, 7]. In general these show that infants learn fairly rapidly from a few examples, although the studies have not monitored every waking hour of the infants, so we do not know how many examples they try while nobody is looking. Notwithstanding this we can point out some definite major differences between our learning and infants: infants have some complex background knowledge which they apply to this problem, largely present as early perceptual competences [34]. There are many potentially useful abstractions in perception which the infant may be using, such as paying particular attention to edges of objects, or the area between two objects. Infants may also have biases to give priority to certain information when learning about causality for example. In addition there are numerous potential sources of knowledge which the infant could be bringing to bear on the problem of understanding when the support works or not; e.g. the infant has prior experience of pushing objects on surfaces and feeling the frictional resistance [13], the infant has prior knowledge of how inanimate objects are not supposed to move unless caused to move by some contact [35], the infant has experience of gravity and how one object presses on the object underneath. We are not sure if any of these sources are in fact used by the infant learning the support, but they are available. Our artificial system by comparison is very ignorant about the world, and is learning with very little background knowledge, apart from its ability to segment objects and analyse relationships between the information extracted about distinct objects. Our system is learning purely from this fairly low level perceptual data; this perceptual data is more limited than what the infant has, and the infant additionally has more sources of other information that it might potentially use. This would explain why infants could learn with fewer examples.

Theoretically we can also discuss the relationship between our “differentiation” and Piagetian assimilation and accommodation. In Piaget’s accounts [2, 7] most phenomena in infancy involve elements of assimilation and elements of accommodation (rarely is something pure assimilation or pure accommodation), and differentiation is no different. These two processes will apply to the motor part as well as the sensory part (precondition) of the schema, although the present paper has only focussed on the sensory part. When a schema is about to be differentiated, such as the schema for pulling being differentiated to “pull support”, there is initially assimilation because the new phenomenon is

seen as similar to the behaviour of pulling, and so the pulling schema is used as a basis for the new schema. There is also an element of accommodation because it is recognised that the schema needs to be adjusted (accommodation); this adjustment (accommodation) happens over a longer timescale than the original assimilation, because extensive training data needs to be gathered about support relationships before an accurate classifier can be trained. Overall accommodation is the dominant process in differentiation, because it is essentially a change to a cognitive structure, however the initial assimilation to some schema which will act as a basis is also crucial. This is a little different to some other descriptions of assimilation and accommodation in computational works. Some neural network models interpret assimilation as the changing of weights in the network whereas accommodation is used for changes to the architecture (e.g. new connections) [36]. We find this interpretation a little too narrow, as Piaget's notion assimilation would seem to apply even if weights are not changed and a network can be used "as is" to classify; also Piaget's notion of accommodation would seem to apply where any change is made, and changing the weights in a network can lead to radical changes in what it would recognise. An example closer to ours is the reinforcement learning system of Tommasino, et al. [37]. They also clearly separate the two processes and use assimilation to apply to the case where an existing expert can be used "as is", whereas they use the term accommodation where an expert with similar sensorimotor mappings is used as a basis and then modified. Under our interpretation we would say that the latter case has an element of assimilation because of the choice of an expert with similar sensorimotor mappings; ultimately this comes down to how people interpret Piaget's writings and what he meant with these two terms; this issue of varying interpretations is also mentioned in the same paper by Tommasino, et al. [37]. The idea of building a library of schemas (or re-using existing "experts") also entails a number of interesting problems related to assimilation and accommodation which we have not addressed here, for example how should the choice be made between adjusting an existing skill or creating a new one? How can the context be used to decide which new situations can be tackled by existing skills? (See also [38]).

5.1. Comparison with related work

Apart from the work of Rosman and Ramamoorthy [8] which we discussed in the introduction, we have not found a related work with which we can make a direct comparison, however several works are less directly related. Our work could be described as autonomously learning planning operators, for which there is some related work in AI [15, 39–41]; Chaput [15] learns new operators via a self-organising map which looks at vectors of all sensor values before and after an action; this was effective in the scenario they used with a small number of binary sensors, but it would not easily scale to larger state spaces. The work of Mourao et al. [41] learns action effects for a robot manipulation scenario. The sensor abstractions are provided (e.g. predicates such as "object x is in object y") and the system learns which predicates become true after the action is performed. Their predicates are predefined whereas in our work we want the state space to develop over time, so that the agent effectively invents new predicates (such as the spatial relationship defining the support relation). Muga and Kuipers' work [40] is somewhat close in that they do not need to predefine predicates and can autonomously find regions within variables; however compared to this we need the ability to learn a wider class of possible preconditions, for example those that involve relationships among variables. Work on learning "affordances" is quite close to ours; Ugur et al. [42] learns affordance predictors for behaviours by learning the mapping from the object features to discovered object effect categories. These predictors can then be used by an agent to make plans to achieve de-

sired goals. This work is quite similar to ours in that essentially it boils down to classification; i.e. once effect categories have been clustered Ugur et al. use a classifier to learn the mapping from the initial object features to these effects. They use SVMs where we use neural networks. At a conceptual level a difference in the approaches concerns what drives the learning of a classifier. In Ugur et al.'s work the decision about what effects a classifier should be learned for is dictated by the choice of features which the agent has been given as its perceptual world. In our work the decision about which action effects to learn a precondition for is dictated by whether those action effects facilitate other subsequent actions which were previously impossible. We would speculate that it is likely that infants combine both approaches.

One major claim we can make for our work relative to others reviewed is that it seems to be one of the few works which tackles these specific actual early infant means-end behaviours; i.e. the first means-end behaviours that infants acquire in the second half of the first year. We are not aware of any other developmental robotics work which tackles the acquisition of the support for example. Some of the works reviewed above tackle domains far removed from infancy (e.g. Chaput's forager [15]) or tasks more advanced than what infants engage in the first year [41]. As stated in the introduction, we believe that the acquisition of early means-end behaviours is of major importance to cognitive development because it is through these that infants begin to gain a higher level understanding of the world, in this case by beginning to take notice of important spatial relationships between objects, which determine how they might behave under manipulation. More generally we believe that the tasks, and ordering, present in early infancy could be quite advantageous to follow because it leads through a developmental trajectory from easy to difficult, with later acquisitions building on earlier ones.

6. Conclusion and Future Work

In this work we have looked at the problem of learning the precondition (and making it more accurate) for a newly discovered means-end behaviour. We have shown that learning these preconditions accurately takes rather a lot of training data, especially when realistic visual input is used. When using the data from our (noisy) vision system we have shown a slight advantage to selecting training examples randomly rather than by an active learning strategy (which contradicted our previous results using noiseless data direct from the simulator). It could be expected that using more detailed and accurate vision information (not just the position and orientation) may well reduce the noise and make an active strategy more appropriate.

Our approach is quite naive in that it looks at rather crude parameters extracted from the visual data, such as position and orientation, which lose most of the detail of the surfaces of the objects involved and their spatial relationships. In future work we intend to make use of more detailed visual data so as to enable the system to more accurately learn the precise spatial relationships between objects which determine the success of various means-end actions which require objects to be in special relationships (which is very relevant to tool use for example).

In future work we would also like to collaborate with psychologists studying infants to learn more about the order in which infants acquire various stages of competence in their learning of the support and other means-end behaviours. For example, we know from Uzgiris and Hunt's studies that although an 8 month-old can successfully pull a support (as a means to retrieve an object), the necessary spatial relationship (on top of) is not understood, and up until 10 months or later the infant will still pull a support even if the desired object is held above it and not touching it [31, p.111], or resting on an object close to the support.

We would like to see a rigorous testing of a wide variety of borderline support cases to determine where infants misclassify the support relationship, and how this changes from 8 to 11 months.

Another aspect to consider is the different sources of knowledge and the order in which infants develop skills for successfully using a tool such as the support, and develop their perceptual skills for recognising the causal relationship when they see it used by someone else. Schlesinger and Langer's results show that causal perception develops later than the (action) skill for using the support [33]. If developing robots are to follow the trajectory of infant development this suggests that the emphasis should be firstly on their own exploratory actions with the tool, rather than learning purely from observation. However Schlesinger and Langer think it unlikely that causal activity is the only determinant in the development of infants causal perception, and further research could help to elucidate the potential contributions of other sources of knowledge. For example linguistic sources of knowledge may also be important. It is claimed that language input may influence category formation as early as 3 months [43], however it is not clear at what age language influences the formation of concepts of spatial relationships such as those dealt with in this paper. Research has shown that 5-month-old infants' categorisation of spatial relationships was not influenced by the language they were exposed to, but by adulthood the extent to which their native language marked these spatial relationships altered their sensitivity to these concepts [44]. It remains unclear at what age the language began to influence the developing spatial relationship concepts. Casasola showed the influence of language on 18-month-olds learning to categorise the support relationship [45]; interestingly her study showed that 18-month-olds failed to categorise support situations, which is surprising given that they would by this time have spent more than half their lives knowing how to use a support in a means-end activity; it suggests that when the context of the task is different infants do not necessarily draw on the knowledge they might have available in another context (i.e. the context of acting to achieve a goal). Other studies examining how linguistic terms for spatial relationships are learned have also shown the difficulty of transferring what is learned in one context to another; they have shown that experience of concrete interactions with the objects facilitate the learning, and also that abstract artificial objects provide situations in infants learn poorly, instead infants require objects with functionally relevant properties [46]. In summary significant further research would be required to elucidate the contributions of all the potential sources of information that an infant might use to learn a spatial relationship, and the additional difficulties provided by varying contextual aspects.

If we step back from the details and look at how the problem tackled here fits into the big picture of the mechanisms of development, we see that finding preconditions is one way to find new abstractions over states. The preconditions of means-end behaviours are particularly interesting because they tend to capture spatial relationships among objects (e.g. the means object and the goal object). It is conceivable that sensory abstractions discovered in this way could subsequently be used to dynamically extend the state space of a cognitive system, so that it adds new higher level state variables. Each time the state space is looked at, the cognitive system can take the base variables from the simulator and extend them step by step with all abstractions found so far. This mechanism may lay a foundation for future work on ongoing emergence by emulating Mandler's mechanism of *Perceptual Analysis* [47]. Newly discovered abstractions could be thought of as emergent symbolic elements in the system. Our long term goal is to create a developmental AI system that evolves from subsymbolic space to symbolic reasoning by finding/creating symbols on its own.

Acknowledgements

This work was supported by the EU Cognitive Systems project XPERIENCE (FP7-ICT-270273) and Leverhulme Grant F/00 152/AL. We also like to thank Lauriane Rat-Fischer for useful references.

References

- [1] Frank Guerin, Dirk Kraft, and Norbert Krüger. A survey of the ontogeny of tool use: from sensorimotor experience to planning. *IEEE Transactions on Autonomous Mental Development*, 5(1):18–45, 2013.
- [2] J. Piaget. *The Origins of Intelligence in Children*. London: Routledge & Kegan Paul, 1936. (French version 1936, translation 1952).
- [3] Jeffrey J. Lockman. A perception-action perspective on tool use development. *Child Development*, 71(1):137–144, 2000.
- [4] L. B. Smith. Dynamic systems, sensori-motor processes and the origins of stability and flexibility. In J. Spencer, M. Thomas, and J. McClelland, editors, *Toward a unified theory of development: Connectionism and dynamic systems theories reconsidered*. Ox. Uni. Press, 2009.
- [5] Kurt W. Fischer. A theory of cognitive development: The control and construction of hierarchies of skills. *Psychological Review*, 87(6):477–531, 1980.
- [6] P. Willatts. Development of problem-solving strategies in infancy. In D.F. Bjorklund, editor, *Children's Strategies: Contemporary Views of Cognitive Development*, pages 23–66. Lawrence Erlbaum, 1990.
- [7] J. Piaget. *The Construction of Reality in the Child*. London: Routledge & Kegan Paul, 1937. (French version 1937, translation 1955).
- [8] Benjamin Rosman and Subramanian Ramamoorthy. Learning spatial relationships between objects. *The International Journal of Robotics Research*, 30(11):1328–1342, 2011.
- [9] Richard S. Sutton. Verification, the key to ai, 2006. Unpublished document, available on author's webpage <http://www.cs.ualberta.ca/~sutton/Incldeas/KeytoAI.html>.
- [10] A. Stoytchev. Some basic principles of developmental robotics. *IEEE Trans. Auton. Mental Development*, 1(2):122–130, 2009.
- [11] Esther Thelen. Rhythmical stereotypies in normal human infants. *Animal Behaviour*, 27(3):699–715, Dec 1979.
- [12] T. G. R. Bower. *Development in Infancy*. San Francisco : W.H. Freeman, 1982.
- [13] Kristine S. Bourgeois, Alexa W. Khawar, S. Ashley Neal, and Jeffrey J. Lockman. Infant manual exploration of objects, surfaces, and their interrelations. *Infancy*, 8(3):233–252, 2005.
- [14] Jean Piaget. *The Child's Conception of the World*. Harcourt, Brace and Co., New York, 1929.
- [15] Harold Chaput. The Constructivist Learning Architecture: A Model of Cognitive Development for Robust Autonomous Robots. *PhD Thesis*, (The University of Texas at Austin, Artificial Intelligence Laboratory), 2004.
- [16] Severin Fichtl. Making Artificial Intelligence which Copies the Way Babies Learn. *Master Thesis*, (Aberdeen University, Department of Computing Science), 2011.
- [17] M Rolf, J J Steil, and M Gienger. Online Goal Babbling for rapid bootstrapping of inverse models in high dimensions. In *Develop-*

- ment and Learning (ICDL), 2011 IEEE International Conference on, volume 2, pages 1–8, 2011.
- [18] Gert Kootstra, Mila Popović, Jimmy Alison Jøstate spacergensen, Kamil Kuklinski, Konstantin Miatliuk, Danica Kragic, and Norbert Krüger. Enabling grasping of unknown objects through a synergistic use of edge and surface information. *Int. J. Rob. Res.*, 31(10):1190–1213, September 2012.
- [19] J. A. Jørgensen, L.-P. Ellekilde, and H. G. Petersen. Rob-WorkSim - an Open Simulator for Sensor based Grasping. In *ISR/ROBOTIK 2010 (41st International Symposium)*. VDE-Verlag, June 2010.
- [20] SørenMaagaard Olesen, Simon Lyder, Dirk Kraft, Norbert Krüger, and JeppeBarsøe Jessen. Real-time extraction of surface patches with associated uncertainties by means of Kinect cameras. *Journal of Real-Time Image Processing*, pages 1–14, 2012.
- [21] Kourosh Khoshelham and Sander Oude Elberink. Accuracy and Resolution of Kinect Depth Data for Indoor Mapping Applications. *Sensors*, 12(2):1437–1454, 2012.
- [22] R B Rusu and S Cousins. 3D is here: Point Cloud Library (PCL). In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 1–4, May 2011.
- [23] N. Pugeault, F. Wörgötter, and N. Krüger. Visual primitives: Local, condensed, and semantically rich visual descriptors and their applications in robotics. *International Journal of Humanoid Robotics (Special Issue on Cognitive Humanoid Vision)*, 7(3):379–405, 2010.
- [24] Martin Riedmiller and Heinrich Braun. RPROP - A Fast Adaptive Learning Algorithm. In *Proceedings of the International Symposium on Computer and Information Science VII*, Institut für Logik, Komplexität und Deduktionssysteme, Universität Karlsruhe, 1992.
- [25] M Riedmiller and H Braun. A direct adaptive method for faster backpropagation learning: the RPROP algorithm. *IEEE International Conference on Neural Networks*, 1993.
- [26] Miroslav Kubat, Robert Holte, and Stan Matwin. Learning when negative examples abound. In *Proceedings of the 9th European Conference on Machine Learning*, ECML '97, pages 146–153, London, UK, UK, 1997. Springer-Verlag.
- [27] Severin Fichtl, John Alexander, Dirk Kraft, Jimmy Alison Jørgensen, Norbert Krüger, and Frank Guerin. Rapidly learning preconditions for means-ends behaviour using active learning. In *2nd IEEE International Conference on Development and Learning and Epigenetic Robotics*, 2012.
- [28] Matti Kääriäinen. Active Learning in the Non-realizable Case. In José L. Balcázar, Philip M. Long, and Frank Stephan, editors, *Algorithmic Learning Theory*, volume 4264 of *Lecture Notes in Computer Science*, pages 63–77. Springer Berlin Heidelberg, 2006.
- [29] Donald E Knuth. Big Omicron and big Omega and big Theta. *SIGACT News*, 8(2):18–24, April 1976.
- [30] Sara Stolbach. Active Learning Models and Noise. *Course Paper, Columbia University*, (Advanced Topics in Computational Learning Theory, COMS 6253), 2007.
- [31] Ina C. Uzgiris and J. McV. Hunt. *Assessment in infancy : ordinal scales of psychological development*. University of Illinois Press, 1975.
- [32] E. Bates, V. Carlson-Luden, and I. Bretherton. Perceptual aspects of tool-using in infancy. *Infant Behavior and Development*, 3(2):181–190, 1980.
- [33] Matthew Schlesinger and Jonas Langer. Infants' developing expectations of possible and impossible tool-use events between ages 8 and 12 months. *Developmental Science*, 2(2):195–205, 1999.
- [34] P.J. Kellman and M.E. Arterberry. *The Cradle of Knowledge*. MIT-Press, 1998.
- [35] J. M. Mandler. How to build a baby: II. conceptual primitives. *Psychological Review*, 99(4):587–604, 1992.
- [36] Denis Mareschal. Computational perspectives on cognitive development. *Wiley Interdisciplinary Reviews: Cognitive Science*, 1(5):696–708, 2010.
- [37] Paolo Tommasino, Daniele Caligiore, Marco Mirolli, and Gianluca Baldassarre. Reinforcement learning algorithms that assimilate and accommodate behaviours: an experimental comparison. In *2nd IEEE International Conference on Development and Learning and Epigenetic Robotics*, 2012.
- [38] A.G.E. Collins and M.J. Frank. Cognitive control over learning: Creating, clustering and generalizing task-set structure. *Psychological Review*, 120:190–229, 2013.
- [39] Terry Zimmerman and Subbarao Kambhampati. Learning-assisted automated planning: Looking back, taking stock, going forward. *AI MAGAZINE*, 24(2):73–96, 2003.
- [40] J. Mugan and B. Kuipers. Autonomous learning of high-level states and actions in continuous environments. *IEEE Trans. Autonomous Mental Development*, 4(1):70–86, 2012.
- [41] Kira Mourão, Ronald Petrick, and Mark Steedman. Using kernel perceptrons to learn action effects for planning. *International Conference on Cognitive Systems (CogSys 2008)*, pages 45–50, 2008.
- [42] E. Ugur, E. Oztop, and E. Sahin. Goal emulation and planning in perceptual space using learned affordances, 2011.
- [43] A. Ferry, S.J. Hespos, and S. Waxman. Language facilitates category formation in 3-month-old infants. *Child Development*, 81(2):472–479, 2010.
- [44] S. J. Hespos and T. Piccin. To generalize or not to generalize: Spatial categories are influenced by physical attributes and language. *Developmental Science*, 12(1):88–95, 2009.
- [45] Marianella Casasola. Can language do the driving? the effect of linguistic input on infants' categorization of support spatial relations. *Developmental Psychology*, 41(1):183–192, 2005.
- [46] Katharina J Rohlfing. Facilitating the acquisition of under by means of in and on—a training study in polish. *J Child Lang*, 33(1):51–69, 2006.
- [47] Jean Matter Mandler. The foundations of conceptual thought in infancy. *Cognitive Development*, 7(3):273–285, 1992.