A Demonstration of the Efficiency of Developmental Learning

Marek W. Doniec, Ganghua Sun, Brian Scassellati
Department of Computer Science
Yale University
New Haven, Connecticut 06511, U.S.A.
{marek.doniec, ganghua.sun, brian.scassellati}@yale.edu

Abstract— **Previous research has suggested that developmental learning can make the learning of advanced sensorimotor and cognitive skills possible. In this paper, we demonstrate that developmental learning based on skill progression is also more efficient than traditional divide-and-conquer methods. Using a model based on the skills of reaching and pointing to visual targets, we demonstrate an implementation for a humanoid robot that is more efficient at learning joint attention skills than other published methods. This efficiency results from (1) a structured set of learning tasks that progresses from low-dimensional to high-dimensional problems and (2) a greater exploitation of the learning environment that does not follow from the completely task-based decomposition that divide-and-conquer provides.**

I. Introduction

The word "development" has been used in computer science with many different meanings. It can mean maturation of sensory and motor capacities such as improvement of visual acuity and increase of muscle strength. It is often used as a substitute for learning of specific skills such as reaching or walking. In this paper, we reserve the word "development" to mean the acquisition of a progression of skills. Although there is a great deal of variability among individual infants, typically skills in different domains are mastered in an orderly fashion. Infants usually learn to sit and crawl before they start to walk. Single words are uttered before syntax and grammar are mastered.

It has been suggested that following a developmental progression of skills might enable a robot to achieve the intelligence or capabilities of an infant [1][2]. Most applications of developmental learning in robotics focus on the learning of individual skills. Metta et al. proposed a method to learn visually-guided reaching by assuming that each arm configuration can be decomposed into a few motor primitives such that the total number of degrees of freedom to be controlled are drastically reduced [3]. Schaal et al. introduced a procedure to use imitation to jumpstart the learning of complex movements such as a tennis swing [4][5]. Rosenstein and Barto demonstrated that learning to use tools can be facilitated with a structured form of reinforcement learning [6]. (Other learning examples include [7],[8] and [9].) All of these studies have shown that diverse, sophisticated skills can be learned by exploiting recent findings from neurophysiology, psychology and machine learning. However, the benefits of *incremental* skill learning have not been sufficiently studied.

Incremental development of skills allows for a structured decomposition of a complex system. Constraints based on limited perceptual or cognitive capabilities of an infant aid in the acquisition of complex skills simply by allowing learning to occur first in lower-dimensional spaces. As infants master basic skills, the already acquired skills become useful tools to reduce the complexity of learning more complex skills. While the efficiency of such a learning process is never questioned within developmental psychology, computational models of development have yet to truly demonstrate this efficacy.

While developmental learning at first glance seems to closely resemble traditional divide-and-conquer approaches that are well known in engineering, the subtleties of a developmental process provide benefits that exceed those of divide-and-conquer strategies. In divide-and-conquer, a complex problem is broken into a set of simpler sub-problems. Once solutions to the sub-problems have been constructed, these complete components are connected together (typically in a sequential chain) to produce a solution to the larger problem. Developmental learning differs from this process in that (1) skills need not be learned or applied sequentially and (2) the decomposition of subproblems can be modified at each stage through interaction with the environment, resulting in a set of components that on face value will not produce the desired complex behavior but in fact will achieve that result through interaction with the appropriate environment. (We will return to this point in Section III.)

Empirical studies on how efficiency can be achieved through a developmental structure are lacking. Current work focuses primarily on uncovering useful mechanisms, such as motor synergies and imitation, for skill learning. Breazeal has studied the use of appropriate facial expressions to facilitate the learning of social skills [10]. In this work, appropriate facial expressions are considered to be innate skills and do not require learning. In one rare example, Metta and Fitzpatrick show that by exploiting existing motor skills, the concept of object affordance can be learned [11]. They, however, also have not provided data on how the developmental method improves efficiency.

In this paper, we provide an extended example of how developmental learning can be more powerful and efficient than traditional divide-and-conquer methods by exploiting already acquired skills. The following section describes a developmental model for learning joint attention behaviors, which allow an individual to attend to the same object as another individual. We demonstrate how this developmental approach allows for a faster, more efficient, and more accurate behavior than those produced by the best divide-and-conquer methods available. We then conclude with a discussion of where the advantages seen in developmental learning originate.

II. An Extended Example: Learning Joint Attention

In this section, we demonstrate the implementation of a system for joint attention which gives a humanoid robot called Nico the ability to attend to the same object of interest as a human caregiver. The system is constructed from a set of basic skill behaviors that include reaching to a visual target and pointing to a visual target. We describe first the existing systems that have achieved reasonable results on joint attention tasks.

A. Related Work

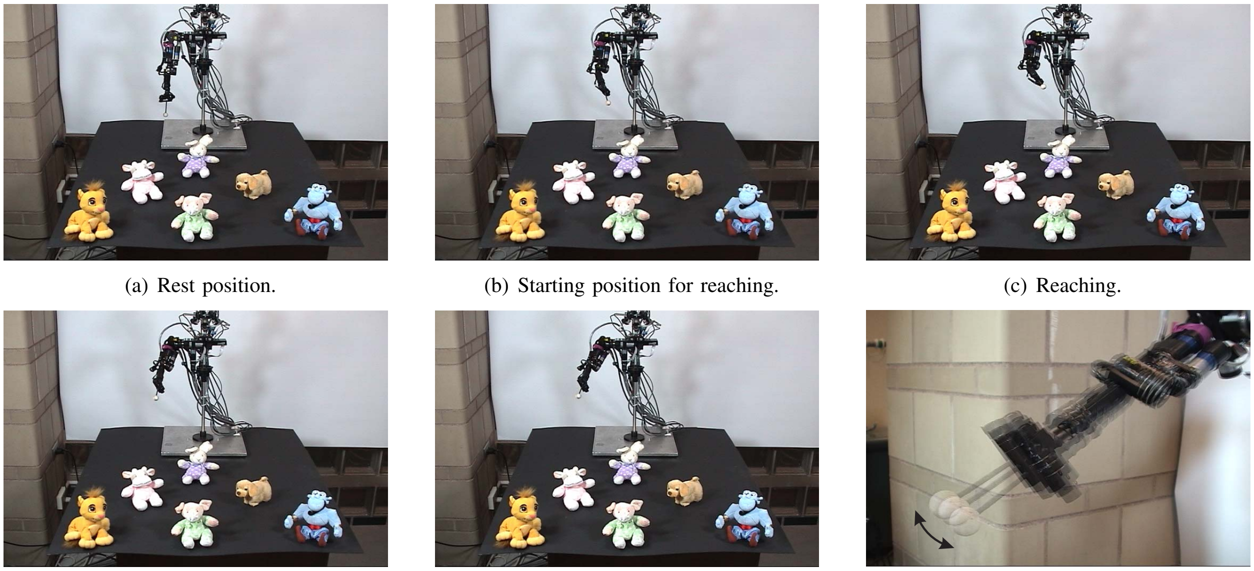
Reaching can be defined as the arm movement that enables the hand or the end-effector to touch a desirable object. Pointing is the gesture signaling ones interest in an object. Joint attention is the process of recognizing and attending to the object another person is looking at. While reaching can be seen as a sensorimotor skill, pointing and joint attention are considered to be important social skills. A typical infant starts to reach for objects at the age of about four to five months [12]. Only at around nine months is imperative pointing exhibited, which is not very different from a simple reach and is often seen as an infant's attempt to reach for an object that is too far away from it [13]. At about six months, some infants can identify which side of the room a caregiver is looking at. However, they in general are not able to localize the correct object of attention until at least six months later [14]. The temporal order of the appearance of reaching, pointing and joint attention means that an infant has already practiced reaching for several months before it exhibits imperative pointing and another several months have passed before reliable joint attention is achieved. This suggests that it is possible that the acquisition of a reaching skill may benefit the generation of a pointing gesture, which may further facilitate the learning of joint attention.

[15] describes a procedure by which the humanoid robot Cog autonomously learns pointing gestures. The procedure consists of two steps. In the first step, a mapping from the image coordinates to the pan/tilt encoder coordinates of the eye motors is learned. In the second step, the mapping from the motor coordinates to the arm gestures is learned. In order to simplify the second mapping, four specific arm positions are selected as motor primitives and other arm positions are represented as linear combinations of these four primitives. The major drawback of this approach is that it requires an artificial definition of pointing. For the sake of simplicity, pointing is defined as the arm position that makes the arm end-effector cover the center of the camera image when the object of interest is foveated. A more intuitive definition of pointing is the arm position which aligns the whole arm or at least the lower arm to the desired object. However, this definition requires the visual detection of the position and orientation of the whole or at least part of the arm if pointing is to be learned from scratch. This computer vision problem is by no means easy to solve.

Nagai et al. have demonstrated a system for joint attention learning by watching for the changes of the caregiver's head pose [16]. Whenever the caregiver moves her head, the robot moves its head in response to look at one of the salient objects within its field of view. The robot then assumes that the caregiver is looking at the same object and records the position of this object and the current head pose of the caregiver as one training sample. Initially, object selection is random and if there are multiple objects in the robot's field of view, the robot may select a different object from the one the caregiver is looking at. Over time, object selection is gradually taken over by a neural network which uses head pose change of the caregiver as input. This model succeeds in learning joint attention because the negative samples in the training set tend to cancel each other out. Unfortunately, the convergence of the model requires such a large number of training samples that the effectiveness of the model can only be demonstrated by simulation. The neural network needs more than $2{\cdot}10^5$ learning steps to accomplish an acceptable success rate of about 80% with three objects in the robot's field of view. The success rate drops substantially when the number of objects increases.

Triesch et al. have proposed a theoretical model for the learning of the joint attention skill [17]. The model assumes that both the infant and the caregiver are located in an idealized grid world, where interesting objects can only exist at a limited number of positions. In their model, the infant acquires the gaze following skill through reinforcement learning. The performance of this model heavily depends on the probability of the caregiver looking at the right positions, i.e. the positions occupied by interesting objects. Similar to Nagai's model, the performance of Triesch's model also deteriorates quickly with the total number of objects in the grid world and a large number of training samples is required for convergence.

We believe that the disadvantages of the two aforementioned models arise from (1) treating the robot/infant as a purely observational agent and (2) removing the learning of joint attention from the rich context of development. During the time infants acquire joint attention, skills such as reaching and pointing are already available. Instead of being passive, infants actively explore their environments with whatever skills available to them. In addition, infants seem to be able detect and sometimes expect contingent responses of their caregivers from an early age[18], [19]. We propose that active pointing and the concept of contingency can vastly accelerate the learning of joint attention skill. To validate this hypothesis, we simulate an active infant with a robot that is programmed to point to objects within its field of view and then capture the response of its caregiver after an appropriate time delay. In this way, the number of false samples in the training set



(a) Rest position.

(b) Starting position for reaching.

(c) Reaching.

(d) Elbow reaches singularity.

(e) Pointing.

(f) Oscillations.

Fig. 1. Pointing to a visual target. **a-c:** The robot first moves its arm to the starting position and then attempts to reach a specific object (the lion on the far left). **d:** The elbow joints have reached their limits during reaching and are no longer used for trajectory generation afterwards. **e:** The robot's arm forms a straight line from the shoulder toward the object. **f:** The iterative nature of our reaching model results in some small oscillations of the end effector at the end of the reaching movement. They are interpreted by the experiment subjects as an attempt of the robot to direct their attention to one of the objects on the table.

can be substantially reduced.

B. Learning to Reach

We previously implemented a fast method for learning iterative reaching through motor babbling [20][21]. In this method, the robot randomly moves its arm and records the position of the end-effector and the corresponding joint angles each time. These data are used to train a neural network to represent a forward kinematic model of the arm. The forward model takes a joint configuration as input and produces the corresponding end effector position. When the robot reaches for an object it computes a local Jacobian matrix J at the current joint configuration $\Delta\theta$ using the learned forward model. We then derive the pseudo-inverse $J^{\#}$ of J and calculate the joint displacement $\Delta\theta$ to move the end effector closer to the target by multiplying $J^{\#}$ with the normalized distance vector $\Delta x = x_{obj} - x_{ee}$, where x_{obj} is the object position and x_{ee} the end-effector position. This method can be easily extended to handle the joints in the neck and generates very natural looking curved reaching trajectories [22].

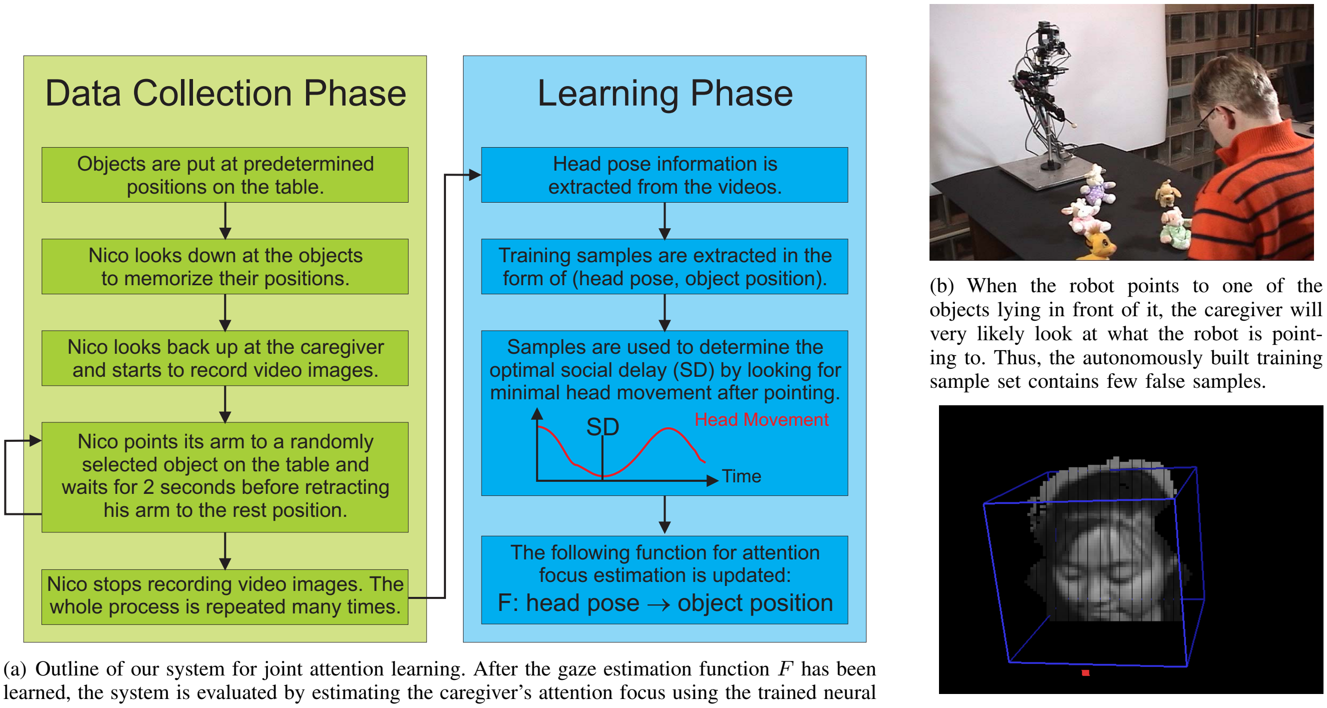
C. From Reaching to Pointing

The aforementioned reaching model can be extended to produce imperative pointing. It has been proposed that early imperative pointing results from the infant being unaware of how far it can reach; by trying to reach for an object but ultimately failing to achieve this goal because the object is too far away, the infant unconsciously produces a pointing gesture [23]. In this case, the end effector has been moved as close to the object as possible.

When an object is out of reach, it means that in order to get as close as possible the elbow has to be fully stretched. The reaching model described above allows this to happen, but in the reaching mode the arm movement will be stopped when the elbow is fully stretched. We extended the reaching model such that after the elbow is stretched, the columns corresponding to the two elbow joints in the local Jacobian matrix J are deleted before $J^{\#}$ is calculated. The iterative arm movement is continued by simply using only the shoulder joints.

Since the iterative method used in our approach based on learning forward model tries to minimize the distance between the end-effector and the object, the arm will gradually move into a straight line pointing directly at the object. (The robot is in fact pointing from the shoulder on towards the object although it has never been provided with the vector pointing from the eye cameras to the shoulder. In a naive solution, this information would be necessary in order to achieve an accurate pointing gesture.) Without any stopping mechanism, the discrete steps used in arm control cause some oscillations around this perfect line of pointing. This behavior looks very similar to the imperative pointing of an infant. Figure 1 shows several snapshots of a pointing gesture that is used in our experiments for joint attention learning.

It is important to note that this pointing behavior is obtained purely by cutting out the elbow joints after it reaches singularity. The development of imperative pointing happens along the way as the robot learns how to reach, but the robot is initially unaware of the fact that by executing a reaching movement to an object that is too far away, the arm configuration eventually looks like a pointing gesture.


Data Collection Phase
Objects are put at predetermined positions on the table.
Nico looks down at the objects to memorize their positions.
Nico looks back up at the caregiver and starts to record video images.
Nico points its arm to a randomly selected object on the table and waits for 2 seconds before retracting his arm to the rest position.
Nico stops recording video images. The whole process is repeated many times.
Learning Phase
Head pose information is extracted from the videos.
Training samples are extracted in the form of (head pose, object position).
Samples are used to determine the optimal social delay (SD) by looking for minimal head movement after pointing.
SD
Head Movement
Time
The following function for attention focus estimation is updated:
F: head pose → object position

(a) Outline of our system for joint attention learning. After the gaze estimation function F has been learned, the system is evaluated by estimating the caregiver's attention focus using the trained neural network and issuing motor commands to make the robot point to the closest object.

(b) When the robot points to one of the objects lying in front of it, the caregiver will very likely look at what the robot is pointing to. Thus, the autonomously built training sample set contains few false samples.

(c) Output of the Watson head pose tracker [24]. The subject is looking at an object to her right.

Fig. 2. Outline of our system for joint attention learning, joint attention scenario, and output of the head pose tracker used.

D. From Pointing to Gaze Estimation

Once the skill of pointing has been acquired it is used to actively learn joint attention. The robot points to draw the caregiver's attention toward an object and records the caregiver's head pose in the process. Contrary to the approach in [16], we actively select which object to attend to and thus have far fewer negative samples. We still might pick up false samples when two objects are so close to each other that the caregiver cannot distinguish which one the robot is pointing to. But in this case the error is small and slows the learning process only marginally.

An outline of the experiment can be seen in Figure 2(a). The robot is presented with multiple objects lying in front of it (six small stuffed animals are used in this experiment). The robot first engages the caregiver by looking at him/her. The robot then looks down at one of the objects on the table and records its position. It then looks back at the caregiver and starts to move its arm to point toward the object. Since we wish to record the caregiver's face at this point and cannot use visual feedback for pointing, the pointing gestures have been prerecorded for all possible object positions used in our experiment. The robot waits for five seconds and then retracts its arm while still recording the user. Each of these events (arm starts moving, arm movement stops, arm is retracted) is marked in the video. These marks are used to help determine when the caregiver is looking at the object.

Videos of the caregiver are recorded in stereo at ten frames per second. The video is then processed by a head pose tracking library [24]. Output of this tracker for one of our test subjects can be seen in Figure 2(c). Through our experiences with Watson we have discovered that due to the lighting condition in our lab and the image quality of the cameras we use, the output of Watson does not always precisely describe the head pose of the experiment subjects. But it is consistent enough to make our learning module that associates head pose information to object position work.

For each pointing trial, it is necessary to extract a single head orientation that best characterizes the user looking at the object. In order to collapse the stream of head poses into a single value we use the social delay approach described in [25]. It has hypothesized that in many social interactions, the response time of an individual can be modelled by a Gaussian distribution. To estimate the parameters of this Gaussian distribution, we have analyzed our data to determine the dynamics of joint attention. Figure 3 shows the amount of movement of the caregiver's head pose during one training example. There are two peaks, namely one where the caregiver moves his head to look at the object and one where the caregiver moves his head back to look at the robot. To gather a heuristic for the social delay we measure the local minimum of the motion signal after the robot has stopped moving its arm. Because the noise of the movement signal is high, smoothing the signal with a box filter lowers the standard deviation. The result is shown in Figure 3. In our experiments, the mean value of the social delay (the difference of the time the robot's arm stops moving and the time corresponding to the local minimum of

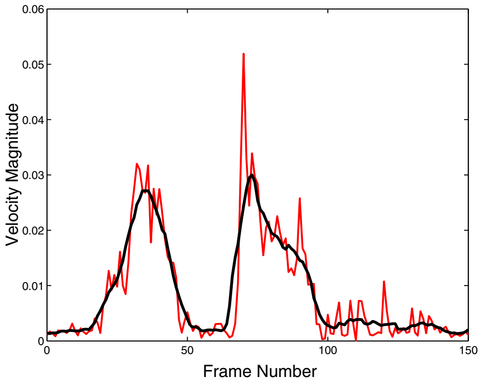


Fig. 3. Example of observed head movement during pointing. The lighter (red) curve is the observed head movement. The darker curve has been smoothed out with a box filter. The first peak occurs when the user shifts his gaze to look at the object the robot is pointing to. The second peak occurs when the user looks back at the robot.

the observed head movement) is about $2.25s$ with a standard deviation of $0.7s$.

Once the correct head pose corresponding to a certain object position has been extracted the data can be used as a training sample for the joint attention learning.

E. Evaluating the Developmental Method

In the last step of the experiment, we train and evaluate a joint attention neural network which converts a measured head pose of the caregiver into a motor command to fixate the distal object of attention. The robot begins a trial by looking at the caregiver. The arrangement of the objects in front of the robot stays unchanged. When the caregiver looks down at any object on the table, the perceptual system computes the head pose and provides that value as the input to the trained joint attention model. The model produces an estimated motor command to fixate the object of attention. The actual motor command is determined by selecting the object that is closest to the position estimated by the joint attention model. The robot fixates that object and then points toward the object to emphasize its attention towards it. The robot succeeds in establishing joint attention if the caregiver (when asked) indicates that the robot is attending to the same object.

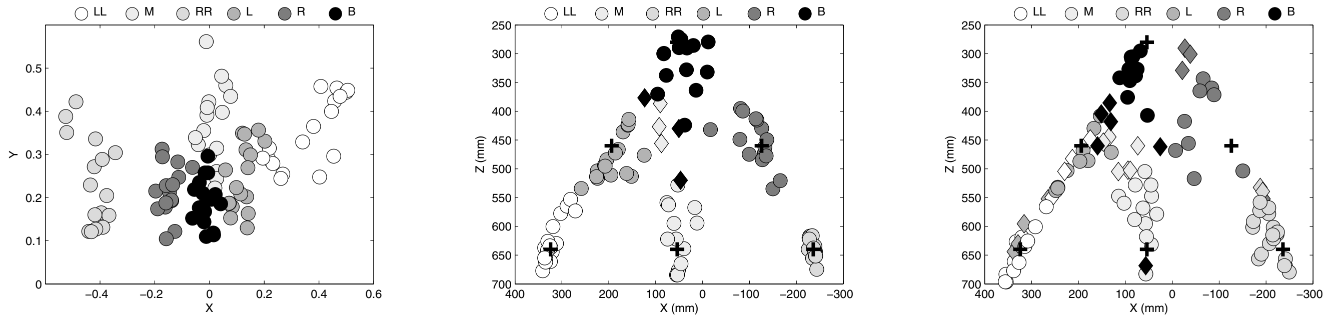
Two groups of test subjects were used. The first group included two of the authors, who were obviously aware of the details of the implementation. The second group consisted of six individuals who were unfamiliar with this work. Each subject sat in front of the robot and were told that they were to observe the robot as if it were a small child and that the robot would be pointing to objects in front of them. The subjects were asked to keep their body still and to initially look straight at the robot. (This step was necessary for the head pose recognition software in our system to work.) They were instructed to look at an object whenever they think the robot is pointing to it. Since we were using head pose estimation, we asked our subjects to move their head rather than their eyes whenever possible, although this instruction seemed to be unnecessary.

The same coordinate system was used to measure the object positions on the table and the head pose of the test subjects. It is based on the position of the eye cameras when the robot has moved its head into the posture for video acquisition. The origin is at the focal point of the left camera. The X-axis points to the focal point of the right camera. The Y-axis and Z-axis point straight down and toward the experiment subjects respectively. The approximate X and Z coordinates (in mm) of the objects on table in the coordinate system described above are as follows - LL: $[324, 640]^T$, M: $[-236, 640]^T$, RR: $[-236, 640]^T$, L: $[194, 460]^T$, R: $[-126, 460]^T$, B: $[54, 280]^T$. The Y components of the object positions are all of the same value because the table is parallel to the X-Z plane of the coordinate system we use. The average distance of two neighboring objects is about 250mm.

The head pose a subject maintains at the beginning of an experiment is used by the head pose recognition system as the reference pose. It is described as $[0, 0, -1]^T$, which is a vector parallel to the table and pointing toward the robot. For the second author of this paper, we have collected 100 head poses $P_{i,i=1,2,...,100}$ from ten video sequences and paired them with the associated object positions $O_{i,i=1,2,...,100}$. Figure 4(a) plots these head poses by using the first two components of P_i. (The third components are redundant since P_i is a normalized unit vector.) The shading of each marker indicates which object position it is associated with. Additionally we collected 100 head poses from five different test subjects from group two. Each of them provides 20 data points.

We use a simple Radial Basis Function Network (RBFN) to learn the association between P_i and O_i. The two free parameters for training a RBFN - the spread of the Gaussians in the hidden layer and the error threshold as stopping criterion - are determined by a simultaneous optimization procedure. The learning performance of the RBFN is first tested on the data collected on the same test subject shown in 4(a). The complete data set is repeatedly split into a training set (80 samples) and a test set (20 samples). Each training set produces a RBFN whose performance is then checked with the samples in the test set. The RBFN projects each head pose vector into a position on the table. The closest object to this position determines the class label of the sample. By using the optimal learning parameters, a RBFN achieves on average a 90.24% recognition rate with a standard deviation of 4.99%.

We also investigated whether the model learned with the data of one person is useful for determining the gaze direction of other people. A RBFN was trained with the complete 100 sample set shown in 4(a). When the head poses in the same sample set are projected into object positions on the table with this RBFN, a better recognition rate of 93.94% is achieved. Figure 4(b) shows these projected positions. The same coding system is used to show the correct object association of each projected position. If a head pose is incorrectly classified by the RBFN, its projected position is plotted with a diamond marker instead of a round one. The positions of toys on the table are plotted with crosshairs. The axes of 4(a) are arranged in such a way that toy positions in it are topologically



(a) Head pose data of the second author (100 samples). Each marker represents a head pose vector projected on the X and Y axis.

(b) Projected object position data of the second author. The RBFN used for this projection is trained with data gathered on the same subject. Out of one hundred samples, only six are misclassified.

(c) Projected object position data of the other experiment subjects. The RBFN trained on the main test subject is tested on the hundred data samples gathered on five other test subjects. Although the number of misclassifications is larger, the result is still impressive considering the variations among the five different test subjects.

Fig. 4. Head pose data for the second author (a) and head poses projected by a trained RBFN into object positions on the table (b & c). The object position each head pose is associated with is indicated with the gray level of the marker. For b & c diamond markers represent misclassifications and the positions of the original six toys on the table are presented with crosshairs.

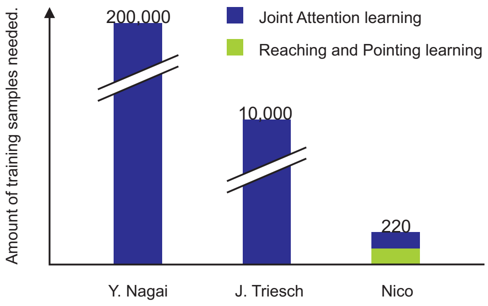


Fig. 5. Quantitative comparison between different joint attention models. The system presented in the paper is designated "Nico" after the name of our humanoid robot.

consistent with those perceived by test subjects. When the same network is used on the head poses on the second data set, which consists of mixed data from different test subjects, a recognition rate of 73.74% is achieved. 4(c) shows these projected positions. This lower recognition rate is partly caused by test subjects maintaining different distances to the robot and variations in their initial head poses towards the robot. Also, although the test subjects are instructed to use head pose change to indicate where they are looking at, some did use more eye movements than the others. (Note that a mapping that assigns a random class label to a head pose will only lead to a recognition rate of only 16.66%.)

Both the individual components of this system and the overall success of the joint attention behavior outperform other published work in this area. With similar accuracy, our developmental method for learning to point succeeded using only 120 learning samples compared to 2000 samples used in [15]. The developmental joint attention method required only 220 samples total (including samples for learning pointing) compared to 10,000 samples in Triesch's method [17] and 200,000 samples in Nagai's approach [16] (see Figure 5).

III. Discussion

Why did the developmental method work so well? Similar to a pure engineering divide-and-conquer approach, we have split the main problem into simpler modules and assembled those together. However, rather than following a purely task-driven decomposition, the developmental model allows us to exploit the nature of the environment and the capabilities afforded by the more basic skills to modify the learning problem. In this example, we first developed a pointing skill which allowed us to designate objects of interest in the environment, effectively changing the problem from one of recovering the underlying statistical distribution of gaze locations that Triesh and Nagai's methods both exploit into a method for self-generating appropriately labeled training data. While this change does require the construction of the basic skills of reaching and pointing (which neither alternate model requires), this change in the nature of the learning problem results in two orders of magnitude fewer training examples. While the amount of training is only one measurement of the efficiency of the system, if you consider that the methods of Nagai and Triesch both require hand-labeled training examples, it is obvious that this difference results in an overall decrease in the amount of human intervention and effort required.

It is important to see that this advantage does not simply derive from splitting the problem into two or a simple assumption about the environment. Instead we use the fact, that by first learning one skill we can use this skill to influence our environment and thus facilitate the development of another skill. In fact we can even sometimes directly develop one skill from another as in the case of developing a pointing skill from the reaching skill, which is the third key issue exploited in developmental learning.

The solutions presented here outperform monolithic approaches to the same problems both in quality (our pointing is view independent) and quantity (both our examples needed far fewer samples then monolithic solutions). Both our solution and that of Nagai are real world models whereas Triesch's model assumes a discrete world. While both Nagai's and Triesch's solutions require a certain amount of off-line work, e.g. manual labeling of the samples, our system works on-line. In addition, our model can be applied across subjects (learned on one subject and tested on the other) with a recognition rate of 73.4%. Nagai's solution has not been cross-subject tested and Triesch's solution does not distinguish between subjects (agents).

It is unlikely that this magnitude of a change in efficiency can be achieved on any arbitrary task. It is also unclear how to properly identify which tasks can benefit from a developmental methodology. However, we have demonstrated that the overall increase in efficiency may be sufficient to warrant some exploration.

IV. Acknowledgement

Support for this work was provided by a National Science Foundation CAREER award (#0238334). Some parts of the architecture used in this work was constructed under NSF grants #0205542 (ITR: A Framework for Rapid Development of Reliable Robotics Software) and #0209122 (ITR: Dance, a Programming Language for the Control of Humanoid Robots). This research was supported in part by a grant of computer software from QNX Software Systems Ltd. We would like to thank Louis-Philippe Morency for providing us with the Watson library and promptly answering whatever question we have regarding to this library. Our special thanks to Charles Baldner, Pu Chen, Christopher Crick, Viksit Gaur, David Glenn, Kevin Gold, Elizabeth Kim, Fred Shic, Yung-Ruey Yen for their participation in the experiments.

References

[1] R. Brooks, C. Breazeal, R. Irie, C. Kemp, M. Marjanovic, B. Scassellati and M. Williamson, Alternative essences of intelligence, In *Proceedings of the 15th National Conference on Artificial Intelligence (AAAI-98)*, pp. 961-976.

[2] M. Asada, K. MacDorman, H. Ishiguro, Y. Kuniyoshi, Cognitive developmental robotics as a new paradigm for the design of humanoid robots, *Robotics and Autonomous Systems*, 37:185-193, 2001.

[3] G. Metta, G. Sandini and J. Konczak, A developmental approach to visually-guided reaching in artificial systems, *Neural Networks*, 12(10):1413-1427, 1999.

[4] S. Schaal, A. Ijspeert and A. Billard, Computational approaches to motor learning by imitation, *Philosophical Transaction of the Royal Society of London: Series B, Biological Sciences*, 358:537-547, 2004.

[5] A. Billard, Y. Epars, S. Calinon, G. Cheng, S. Schaal, Discovering optimal imitation strategies, *Robotics and Autonomous Systems*, 47(2-3):68-77, 2004.

[6] M. Rosenstein and A. Barto, Supervised actor-critic reinforcement learning, In *Learning and Approximate Dynamic Programming: Scaling Up to the Real World*, Ed. J. Si, A. Barto, W. Powell and D. Wunsch, pp. 359-380, John Wiley & Sons, New York, 2004.

[7] L. Berthouze and M. Lungarella, Motor skill acquisition under environmental perturbations: On the necessity of alternate freezing and freeing of degrees of freedom, *Adaptive Behavior*, 12(1):47-63, 2004.

[8] K. Dautenhahn and A. Billard, Studying robot social cognition within a developmental psychology framework, In *Proceedings of the 3rd International Workshop on Advanced Mobile Robots*, 1999.

[9] P. Varshavskaya, Behavior-based early language development on a humanoid robot, In *Proceedings of the 2nd International Conference on Epigenetics Robotics*, pp. 149-158, 2002.

[10] C. Breazeal, *Designing Sociable Robots*, MIT Press, 2002.

[11] G. Metta and P. Fitzpatrick, Early integration of vision and manipulation, *Adaptive Behavior*, 11(2):109-128, 2003.

[12] C. von Hofsten, Structuring of early reaching movements: a longitudianal study, *Journal of Motor Behavior*, 23(4):280-292, 1991.

[13] S. Baron-Cohen, *Mindblindness: An essay on autism and theory of mind*, MIT Press, 1995.

[14] G. Butterworth and N. Jarrett, What minds have in common is space: Spatial mechanisms serving joint visual attention in infancy, *British Journal of Developmental Psychology*, 9:55-72, 1991.

[15] M. Marjanović, B. Scassellati and M. Williamson, Self-Taught visually-guided pointing for a humanoid robot, In *Proceedings of the 4th International Conference on Simulation of Adaptive Behavior (SAB-96)*, Cape Cod, Massachusetts, 1996.

[16] Y. Nagai, K. Hosoda and M. Asada, How does an infant acquire the ability of joint attention?: A Constructive Approach, In *Proceedings of the Third International Workshop on Epigenetic Robotics*, pp. 91-98, 2003.

[17] J. Triesch, C. Teuscher, G. Deák and E. Carlson, Gaze Following: why (not) learn it?, *Developmental Science*, in press.

[18] A. Leslie and S. Keeble, Do six-month-old infants perceive causality?, *Cognition*, 25:265-288, 1987.

[19] J. Nadel, I. Carchon, C. Kervella, D. Marcelli and D. Réserbat-Plantey, Expectancies for social contingency in 2-month-olds, *Developmental Science*, 2(2):164-173, 1999.

[20] G. Sun and B. Scassellati, Reaching through Learned Forward Model, In *Proceedings of IEEE-RAS/RSJ International Conference on Humanoid Robots (Humanoids 2004)*, 2004.

[21] G. Sun and B. Scassellati, A Fast and Efficient Model for Learning to Reach, *International Journal of Humanoid Robotics*, 2(4):391-414, 2005.

[22] G. Sun and B. Scassellati, Exploiting vestibular output during learning results in naturally curved reaching trajectories, In *Proceedings of Fifth International Workshop on Epigenetic Robotics*, 2005.

[23] B. Scassellati, Imitation and mechanisms of joint attention: A developmental structure for building social skills on a humanoid robot, In *Computation for Metaphors, Analogy and Agents*, Ed. C. Nehaniv, Springer-Verlag, pp. 176-195, 1998.

[24] L. Morency, A. Rahimi and T. Darell, Adaptive view-based appearance model, In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition*, 2003.

[25] K. Gold and B. Scassellati, Learning about the self and others through contingency, *AAAI Spring Symposium on Developmental Robotics*, Palo Alto, California, 2005.