Active Learning of Joint Attention

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

***Abstract*— Joint attention is the skill of attending to the same object another person is looking at. The acquisition of this skill is crucial in children for the development of many social and communicative abilities, and has been proposed as a critical social capability for interactive robots. Although recent attempts to model the acquisition of this skill on a robot have been moderately successful [5], [8], they all assume that the robot remains passive during the learning process. Infants, on the other hand, have already acquired some rudimentary sensorimotor skills by the time they start to learn joint attention. We believe that these sensorimotor skills can jumpstart and considerably accelerate the learning of joint attention. In this paper we demonstrate on a humanoid robot how to use pointing and reaching to accelerate the learning of joint attention. We show that a robot can acquire this skill with a 95% accuracy after a total of only 220 training samples compared to 85% accuracy after totals of 10,000+ samples in other approaches [5], [8].**

I. Introduction

Joint attention is the skill of attending to the same object another person is looking at. It is a skill that is fully acquired by children around the age of 18 months [1] and that is crucial for the development of further social and language abilities [2].

Recently researchers have investigated how an agent might be learning joint attention by trying to simulate the development of this skill on humanoid robots [3], [4], [5]. All of these approaches assume that the child robot is passive and learns joint attention purely by observing the caregiver and environment. In contrast, children acquire a rich set of sensorimotor skills before they begin to show joint attention behavior. Children start learning how to reach and point at an age of four to five months [6]. Initial pointing gestures (known as imperative pointing) consist of an extended arm as if reaching for an object without the extended index finger characteristic of adult pointing gestures (known as declarative pointing). Infants are typically skilled at imperative pointing by the time they start to learn joint attention, around 9 months of age [4], [7].

It is our hypothesis that pointing gestures can be exploited to rapidly and efficiently learn joint attention behavior. We present in this paper a robot that first learns how to reach and point to objects using traditional techniques. The robot then actively directs the attention of the caregiver by pointing to specific objects. The object's location (obtained from visual depth estimation) and the head pose of the caregiver (also obtained from visual information) are used as positive training examples for a neural network which produces the joint attention function. That is, given a sensed head pose estimate, the network predicts of the location of the attended object. We demonstrate that this layered learning technique can obtain higher performance than other reported techniques and obtains this accuracy in two orders of magnitude fewer training examples.

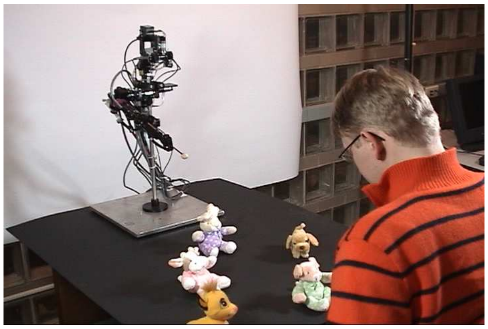
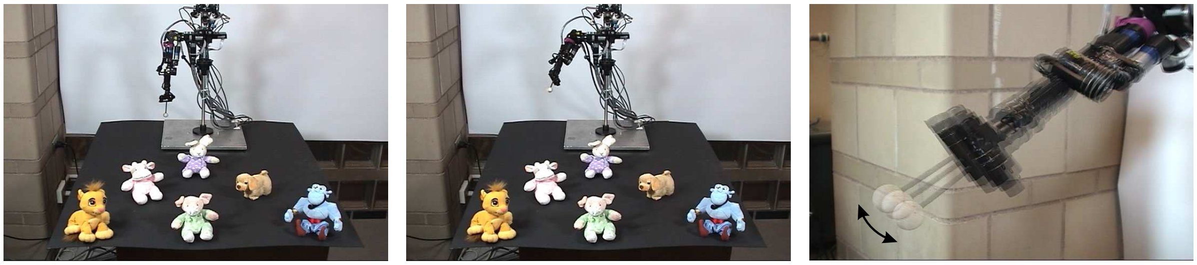


Fig. 1. Typical joint attention scenario.

The rest of the paper is organized as follows. Section 2 gives an overview of related work. In Section 3 we present our methodology and explain how learning joint attention takes place in our robot. Results are presented in section 4 and a discussion is given in section 5.

II. Related Work

Nagai, Hosoda & Asada [5] demonstrated a system which learned joint attention by watching for shifts in the caregiver's head pose. Whenever the caregiver's head moved, the robot would move its head to look at one of the salient objects within its field of view. Initially, this selection was random; each salient object within the field of view was equally likely to be selected. The robot then assumed that the caregiver was looking at the same object that it was attending to. The caregiver's head pose and the object location were used as a training sample. If there were multiple objects in the robot's view, this example had a chance of being false; the robot might randomly select a different object. Over time, the selection of a visual target is accomplished by a neural network which produces a gaze shift motor command given a head pose change of the caregiver. Although the training samples contain many false associations (from an incorrect initial random object selection), this model succeeds in learning joint attention because the random spatial distribution of negative examples tend to cancel each other out.


(a) Start position.

(b) Pointing.

(c) Strong indication.

Fig. 2. Pointing to a visual target. **a,b:** The robot is presented with multiple objects on the table and points to one of them with its arm. **c:** Small oscillations emerge naturally from our trajectory generation algorithm. **Object labels:** LL: lion front left, M: pig front middle, RR: genie front right, L: cow middle left, R: dog middle right, B: rabbit in the back.

One advantage of this approach is that the robot's field of view grows as the neural network becomes more and more accurate. Whenever the network selects a location that does not contain a salient visual target, a second gaze shift is performed to a randomly chosen salient object within the field of view. If the second gaze shift is in the same direction as the initial shift, the robot can attain an object position that is initially outside of its range. In this way, the effective field of view of the network can increase over time.

The disadvantage of Nagai, Hosoda & Asada's method is that it requires a large amount of training data. In fact, the amount of training samples needed is so large that the authors had to gather initial data and then simulate the learning environment and the caregiver's gaze. The neural network needed more then $2 * 10^5$ learning steps to accomplish an acceptable success rate of about 0.8 with three objects in the FOV. The success rate drops substantially when the number of object increases.

Triesch et al [8] have proposed a theoretical model for the study of joint attention development. 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 both these models arise from the assumption that the robot/infant remains passive throughout learning. We propose that simple sensorimotor skills can vastly accelerate the learning process. Our approach is for the robot to draw the caregiver's attention toward an object by pointing to it and thus actively acquiring a positive training example for a joint attention network (see figure 1).

III. Methodology

Our experiments used an upper torso humanoid robot, called Nico, constructed to match the kinematics of a one-year old infant. Nico's head has a total of seven degrees of freedom (DOFs) including separate yaw and simultaneous pitch for both eyes. The arms have six DOFs each, two at the shoulder, elbow and wrist respectively. All arm and head joint angles are constrained to represent the abilities of a human infant. The eyes are equipped with two miniature CCD cameras each, one for foveal and one for peripheral vision. In the experiments described in this paper we used only the peripheral vision cameras.

Our experiment is conducted in three phases. First the robot learns to reach by acquiring a forward model of its arm. Second the robot extends this model to pointing. Finally, it uses the gained skills to learn how to recognize the caregiver's gaze towards an object. Due to the low resolution of our current cameras we were not able to estimate eye gaze and used only head pose estimation. The following subsections describe the algorithms used and the experimental setup.

A. From Reaching to Pointing

We previously implemented a fast method for learning iterative reaching through motor babbling [9], [10]. In this method, the robot randomly moves its arm and records the position of the end-effector each time. This data is used to train a neural network that represents a forward model of the arm. When the robot reaches for an object it will create the Jacobian matrix for the current end effectors position and use it to move its end-effector closer to the target. In our model this Jacobian matrix J is a 3×6 matrix (6 DOFs in the arm and 3 dimensions to describe the end-effectors position). We then derive it's local inverse $J^{\#}$ with a simple algebraic calculation:

$$J^{\#} = J^T \cdot (J \cdot J^T)^{-1} \tag{1}$$

The desired arm displacement in joint angles can be calculated by multiplying $J^{\#}$ with the distance vector $D = O_{pos} - EE_{pos}$ where O_{pos} is the object position and EE_{pos} the end-effector position:

$$\dot{\theta} = J^{\#} \cdot D \tag{2}$$

This behavior generates very natural looking curved reaching trajectories [11].

This reaching model can be extended to produce imperative pointing. Children use imperative pointing as a result of not being able to reach objects that are too far away to grasp [4]. It

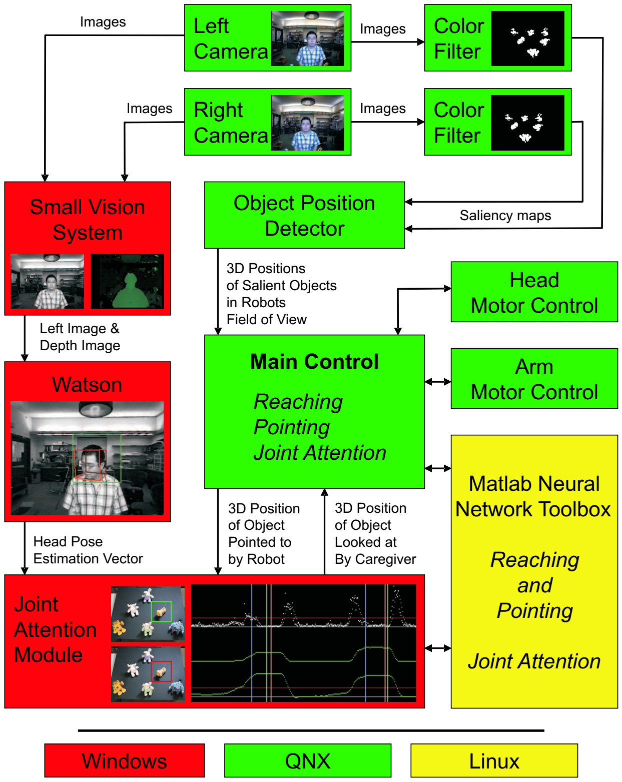


Fig. 3. Outline and data flow of our system used during the joint attention experiment.

has been proposed that early imperative pointing results from the child being unaware of the distance they can reach; the imperative point results from trying to reach for an object but ultimately failing because it is too far away. In this case, the arm has been extended to come as close to the object as possible. In many cases, a caregiver will eventually recognize the gesture and move the pointed-to object into the child's grasp. The reward of obtaining the desired object might be used to learn that pointing can also achieve the goal of getting an object. We have thus altered out reaching model to follow this conjecture.

When an object is out of reach it means that to get as close as possible we usually have to stretch our elbow. The iterative reaching model alone will allow for this to happen, but the arm movement will stop here since the elbow joint reaches singularity. We have extended the model to remove the elbow joints from the Jacobian matrix calculation at this point and continue iterative movement by simply using the shoulder joints. The details are as follows:

$$J_{3\times 6} \Rightarrow J'_{3\times 4},\ \theta_{6\times 1} \Rightarrow \theta'_{4\times 1} \qquad (3)$$

$$\dot{\theta}' = J'^{\#} \cdot D \qquad (4)$$

Since the iterative method used in our approach tries to minimize the distance between the end-effector and the object, out arm will move into a straight line pointing directly at the object. It means that the robot is in fact pointing from the shoulder on towards the object although it was never explicitly told the vector between the head cameras and the shoulder. In a straight forward model, this information would be necessary in order to calculate an accurate pointing gesture. Without any stop mechanism, the discrete steps used in arm control may cause oscillation around this perfect line of pointing. With minimal smoothing, the behavior of this system looks very much like the imperative pointing of an infant. Figure 2 shows a sample scenario of the robot pointing to one of the objects on the table.

It is important to note that this pointing behavior is obtained purely by cutting away the two elbow joints when they reach singularity during object reaching. Learning how to point imperatively happens along the way as our robot learns how to reach. The robot is not currently aware of the fact that it is pointing and that it cannot reach the object.

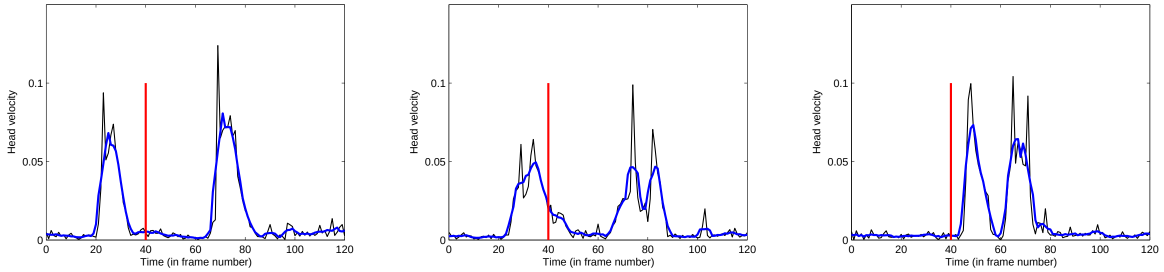
B. Learning Joint Attention

Pointing is used to actively learn joint attention. The robot points to draw the caregiver's attention towards an object and records the caregiver's head pose in the process. Contrary to the approach in [5], we actively select which object to attend to and thus have far fewer negative samples. We still might encounter negative samples when two objects are so near each other that the caregiver cannot distinguish the pointing gesture or when we encounter a glitch in some component of the system. But in this case the error is very small and slows the learning process only slightly.

Fig. 3 gives an outline of the software modules that control Nico during the joint attention learning phase of the experiment. The neural network for reaching and pointing is already trained and the control module uses output from the object position detector to find objects it can point to. The output from the stereo cameras is also fed into the SVS (small vision system) for stereo processing and from there into the Watson headpose detector [12]. Based on the theory of contingency learning the joint attention module analyzes the headpose estimation and extracts learning samples for the joint attention neural networks. The complete system runs at ten frames / samples per second.

The experiment is conducted as follows. The robot is presented with multiple objects on a table in front of it (six small stuffed animals were used in this experiment as shown in fig. 2). 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 towards the object. The neural network for pointing is precise enough by now to allow for pointing without visual feedback. The robot waits for two seconds and then retracts its arm. Each of these events is signaled to the joint attention module (start pointing, arm movement stops, retract). These markers are used by the joint attention module to determine when to extract an appropriate headpose training sample. Information regarding to how this is done is presented in section IV-A.

If enough training samples have been gathered the control module trains a neural network for attention estimation. Cur-



(a) (b) (c)

Fig. 4. Examples of how three different subjects respond to the robot's pointing gestures. The black curve represents the velocity of the head movement recorded during the experiments. The blue curve is its smoothed-out version. The red line marks the time when the robot completes a pointing gesture.

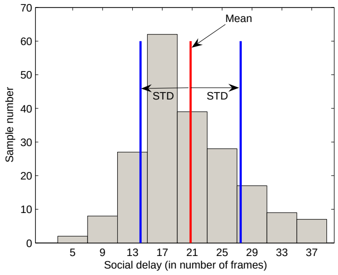


Fig. 5. Histogram of the time delays of the experiment subjects' responses to the robot's point gestures. Despite the individual differences, the distribution shows a single peak and resembles a Gaussian. The mean and the standard deviation of the distribution are $2.08s$ and $0.67s$ respectively.

rently this happens when the system has 100 training samples, a number found by trial and error. In the future we would like to include an evaluation function that allows the robot to take this step autonomously.

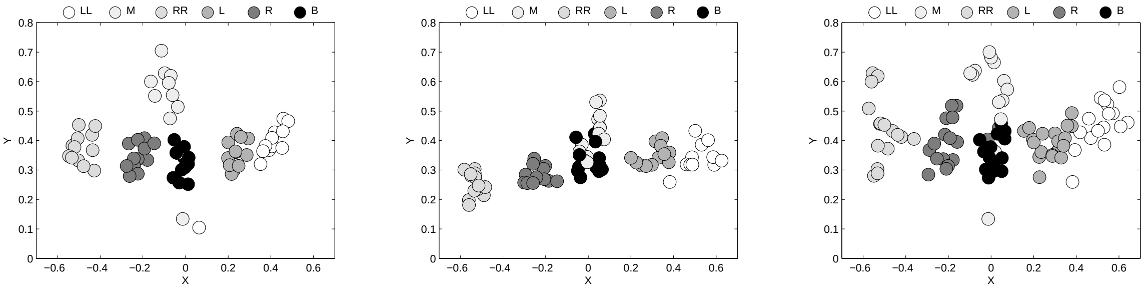
Once a joint attention neural network has been trained the last step of the experiment starts. The robot begins a trial by looking at the caregiver. When the caregiver looks down at any object on the table, the perceptual system computes the head pose angle and provides that value as the input to the 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 a visually salient object closest to the position estimated by the joint attention model. The robot foveates 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.

IV. Results

A. Social Delay

Fig. 4 shows examples of how three different experiment subjects respond to the robot's pointing gestures. The black curves represent the velocity of the subjects' head movement within a time window of 12 seconds (the video grabbers operate at 10 frames/second). The blue curves are simply the smoothed-out versions of the actual velocity curves using a box filter. The red line in each plot reflects the time point when the robot completes the current pointing movement. It is obvious that different subjects respond to the robot in different manners. Subject M (Fig. 4(a)) follows the robot's arm movement closely and finds the object the robot is pointing to even before the robot fully completes the movement. The velocity curves in Fig. 4(a) reflects this behavior and shows that the velocity of the head movement falls back to zero before the red line. Fig. 4(b) shows the velocity curves of Subject P who also starts to move their head soon after the robot starts its arm movement. However, P stops the head movement after the robot completes the pointing gesture. The velocity curves in Fig. 4(c) characterizes the behavior of Subject L who apparent uses eye movements to follow the robot's arm movement and only starts to move the head after the robot completes the pointing gesture. In addition to the difference in moving the head to look at the object the robot points to, the length of the time spent on looking at the object varies from person to person. While subject M often looks at the object longer than two seconds, subject L usually moves the head back quickly to look at the robot's eyes.

During the experiments, each time the robot pointed to an object, the joint attention module extracted from Watson's continuous output a single head pose that best characterizes the caregiver's response. It has been hypothesized in [13] that the response delay of an individual in a social interaction can be modelled by a Gaussian distribution. A careful analysis of the data collected during our experiments has partially confirmed this hypothesis. For each of the subjects, the original velocity curves recorded during experiments are smoothed with a box filter that is twenty frames wide. These curves are then segmented into a number of episodes, each of them containing the subject's response to a particular pointing gesture produced by the robot. For each episode, the difference between the time the robot completes the arm movement and the time the subject exhibits a minimum amount of head movement is calculated. This difference characterizes the delay of the subject's response to the robot's social gesture. These delays are then aggregated and shown in Fig. 5. Despite the differences in how subjects react to the robot's pointing (as illustrated in Fig. 4), the distribution of the social delays shows a single peak and some resemblance to a Gaussian with a



(a) Head pose data of Subject L.

(b) Head pose data of Subject P.

(c) Mixed data from five different subjects (20 samples each).

Fig. 6. Headpose samples collected autonomously by the robot. Each marker represents a head pose vector projected to the X and the Y axis. The shading of each marker indicates the object position it is associated with.

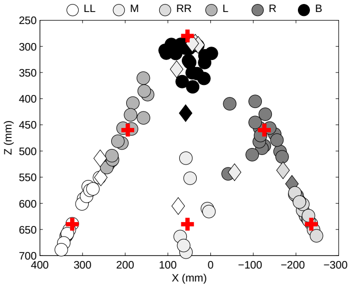


Fig. 7. A RBFN is trained on a mixed data set portrayed in Fig. 6c. The performance of this RBFN is tested on another mixed data set consisting of 100 samples. This plot shows the projection of the head pose data in the test set on the X-Z plane. The original positions of the objects on the table (illustrated in Fig. 2) are marked with red crosshairs. The shading of each marker is determined by the closest-neighbor rule. Only 15 samples are misclassified. These samples are plotted with diamond markers.

mean of $2.08s$ and a standard deviation of $0.67s$. During the automatic sample extraction mode, the main control module records the head pose data obtained from Watson after a $2s$ delay and associates this head pose with the position of the object the robot is currently pointing to. All training sets described in the following section are built in this manner.

B. System Performance

The same coordinate system is used to measure the object positions on the table and the head poses of the test subjects. The origin is at the focal point of the left camera, and 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 Y components of positions of the six objects on the table 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 of a subject at any particular moment is described with a three dimensional vector output by Watson. Only two of the components are independent. All training samples are collected using the social delay approach described in the last section. Each training sample is in the form of (p_i, o_i), where p_i and o_i describe the head pose and the associated object position respectively. We use a simple Radial Basis Function Network (RBFN) to learn the association between p_i (input) and o_i (output). 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 performance of the RBFN is first evaluated by testing on the same individual that the network was trained upon. The training set and the test set consist of 80 and 20 samples respectively. The performance of the trained network is very good in this case; an average recognition rate of 95% is achieved. However, if we use a network trained on one subject directly to test on the head poses of another subject, the average recognition rate is only 62%. The reason for this severe degradation is the variance of head pose data among different subjects. Fig. 6(a) and Fig. 6(b) visualize the head pose data of subject L and subject P by using only the first two components of $p_i s$. (The third component is redundant since p_i is a normalized vector.) The shading of each marker indicates which object position it is associated with. These two plots show that although both L's and P's head pose data are well clustered, they have significant differences. This performance issue can be resolved by training a neural network on a mixed data set collected on different subjects. Fig. 6(c) shows the head pose data of a training set (100 samples) created by mixing data collected from five different subjects. When a network trained on this training set is applied to the head poses contained in another mixed data set (100 samples), a recognition rate of 85% is achieved. Fig. 7 shows the projection of the head poses in the test set. The same color 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 the objects on the table are plotted with crosshairs. The axes of 7 are arranged in such a way that object positions in it are topologically consistent with Fig. 2.

V. Discussion

In this paper we have presented a complete robotic system that learns joint attention much faster then previously proposed systems. Our system runs in real time and all the learning is

TABLE I

COMPARISON OF EXISTING JOINT ATTENTION MODELS.

	Y. Nagai, K. Hosoda and M. Asada [5]	J. Triesch, C. Teuscher, G. Deák and E. Carlson [8]	Nico
Real world model:	+ Yes.	− No. Model assumes discrete world.	+ Yes.
Real time online training:	− Off line learning.	− Only simulated.	+ Complete online algorithms.
Number of samples / time steps required:	− $200,000$ samples	− $10,000$ to $100,000$ timesteps (reinforcement learning)	+ 220 samples in real time
Recognition rate with multiple objects in view:	− Drops fast with number of objects ($\sim 80\%$ for 3 objects).	− Drops fast with number of objects.	+ $\sim 95\%$ for 6 objects
Cross subject training / evaluation:	− Model tested on only one subject. Neural network takes face image as input and is likely highly personalized.	− Agents are characterized by probability of looking at an object but there is no behavioral personalization.	+ Training on multiple subjects yields $\sim$ 85% recognition rate.

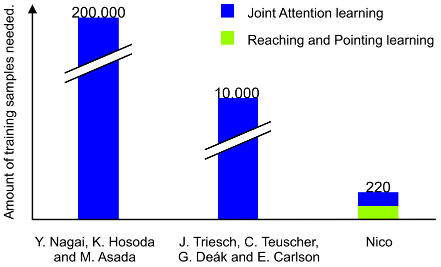


Fig. 8. Quantitative comparison between different joint attention models and implementations.

done online without any human intervention. This is made possible by breaking the learning task apart into finer and easier tasks and learning them one after the other. First the robot learns how to reach objects using as few as 120 examples. Once the robot has learned how to reach objects it extend the reaching model to pointing. This is done without the need for further data samples. Using these two basic skills our robot can learn joint attention with only 100 additional data samples. In addition to being computationally efficient our system works on multiple subjects.

When compared to other existing joint attention learning approaches our system has multiple advantages that are summarized in table I. The two most important points are that our system reduces the number of training samples by a factor of at least 50 and it is the first complete online system that works autonomously. Triesch's approach is purely theoretic, whereas Nagai's system performs training off line. In regard to speed other systems need up to 200,000 samples for a comparable accuracy to that of our system after 220 samples. This can be seen in figure 8. Additionally our system has been trained and tested on multiple subjects whereas other joint attention learning approaches did not attempt to generalize to different subjects. When trained on multiple subjects and tested on another data set from multiple subjects we had a 85% recognition rate with our system. It is worth mentioning that the modular design presented here is easier to identify and isolate weaknesses and could more easily be extended for further improvements.

Having a set of basic skills allows the robot to decompose the learning of more advanced skills into smaller learning units. This can dramatically increase learning speed as in our joint attention model. Thus we believe that this model can be extended to let the robot continuously learn more and more complicated skills.

For future work we would like to add eye gaze tracking support to make the model more accurate and allow the users to move freely. In order to do so foveal vision cameras could be used. Afterwards we would like to expand our developmental model to learn to recognize the caregiver's pointing gestures. We would like to use the newly learned gaze estimation to classify pointing directions correctly and thus allow for a fast learning.

REFERENCES

[1] G. E. Butterworth and N. L. M. 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.

[2] D. C. Dennett, *Consciousness Explained*, Little, Brown, & Company, 1991.

[3] R. A. Brooks, C. Breazeal, R. Irie, C. C. Kemp, M. Marjanovic, B. Scassellati, M. M. Williamson, *Alternative Essences of Intelligence*, In Proceedings of the American Association of Artificial Intelligences, p. 961-968.

[4] B. Scassellati, *Imitation and mechanisms of joint attention: A developmental structure for building social skills on a humanoid robot*, In C. Nehaniv, ed., Computation for Metaphors, Analogy and Agents, Vol. 1562 of Springer Lecture Notes in Artificial Intelligence, Springer-Verlag, p. 176-195, 1998.

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

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

[7] S. Baron-Cohen, *Mindblindness: An essay on autism and theory of mind*, MIT Press, 1996.

[8] J. Triesch, C. Teuscher, G. Deák and E. Carlson, *Gaze Following: why (not) learn it?*, Developmental Science, in press (2006).

[9] C. G. Sun and B. Scassellati, *A Fast and Efficient Model for Learning to Reach*, International Journal of Humanoid Robotics (in press).

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

[11] C. G. Sun and B. Scassellati, *Exploiting Vestibular Output during Learning Results in Naturally Curved Reaching Trajectories*, Proceedings of Fifth International Workshop on Epigenetic Robotics, 2005.

[12] L. P. Morency, A. Rahimi and T. Darell *Adaptive View-based Appearance Model*, Proceedings IEEE Conf. on Computer Vision and Pattern Recognition, 2003.

[13] K. Gold and B. Scassellati, *Learning About the Self and Others Through Contingency*, AAAI Spring Symposium on Developmental Robotics, Palo Alto, 2005.