Building Behaviors Developmentally: A New Formalism

Brian Scassellati
MIT Artificial Intelligence Lab
545 Technology Square
Cambridge, MA, 02139, USA
scaz@ai.mit.edu

Abstract

This paper advocates a developmental approach to building complex interactive behaviors for robotic systems. A developmental methodology is advantageous because it provides a structured decomposition of complex tasks, because it facilitates learning, and because it allows for a gradual increase in task complexity. The developmental approach provides a structured means both of dividing a task among research units, as well as a metric for evaluating the progress of the task. Initial work with developmental modeling has also hinted that these skill decompositions may make the overall task easier to accomplish through the re-use of knowledge gained from developmental precursors.

We report here on two projects of building behaviors developmentally for a humanoid robot. In the first project, the robot learned to reach to a visual target by following a developmental progression similar to those observed in infants. The second project outlines a proposal for building social skills developmentally.

Introduction

One of the main tasks for our field is to develop methods to explain, construct, and understand complex interactive behavior. If we are to build complete, integrated systems that can utilize a multitude of behaviors to survive in the real world, we must find methodologies that allow us to scale our behavioral systems easily. Earlier methods, based on models of adult biological systems or on ad-hoc computational formalisms, have failed to provide a structure that facilitates scaling simple behaviors into complex, adaptive behaviors.

An alternative methodology is to use a developmental approach to designing robotic systems. A developmental approach divides high-level tasks into computationally simpler, and developmentally earlier, behaviors. A developmental approach is deals directly with the scaling problem, that is, how can one exploit existing structures to facilitate learning new, more sophisticated behavior?

Our group has been constructing an upper-torso humanoid robot called Cog (see Figure 1), in part to explore how building behaviors developmentally can help us overcome the scaling problem. The robot has been built to roughly approximate the motion of a human, with twenty-one degrees of freedom, and to roughly approximate the sensory capabilities of a human, with visual, auditory, proprioceptive, and vestibular senses. In this paper, we will examine the potential benefits of a developmental methodology and provide two examples of how this methodology can be applied to robotic systems.

Advantages to the Developmental Methodology

A developmental methodology is advantageous because it provides a structured decomposition of complex tasks, because it facilitates learning, and because it allows for a gradual increase in task complexity. In this section, we explore these three themes, drawing on examples of biological development and the implications that these insights have to the development of robotic systems. We focus primarily on human development, but developmental phenomena can be observed in virtually all animals.

Development gives a structured decomposition

Often, the most difficult aspect of building a complex, integrated system is simply dividing the problem into a set of manageable behaviors. The ways in which we decompose problems can greatly influence the difficulty of implementing those pieces. A useful decomposition breaks the problem into manageable pieces that overlap enough to allow integration to be successful, but do not unnecessarily duplicate effort. For the decomposition to be effective, we must be certain that the pieces will combine into the desired behavior.

Developmental studies give insights into decompositions of complex skills that are both useful and effective. Development is an incremental process in which

Figure 1: Cog, an upper-torso humanoid robot. Cog has 21 degrees of freedom to approximate human motion, and visual, vestibular, proprioceptive, and auditory senses to approximate human perception.

later forms of behavior arise from the situated context that earlier behaviors provide. These precursors provide a task decomposition that is not necessarily a minimal set of required behaviors, but at the least a sufficient set. This decomposition also allows for larger behavioral tasks to be divided between a set of research teams, or between members of a research group.

This decomposition is also useful because it provides a set of stable behavioral precursors which can be used as an evaluation metric. While it is often difficult to evaluate the progress of a robotic project, the presence of a set of precursor behaviors gives a demonstrable means to evaluate the progress of a system. The evaluation of the system is the set of behaviors that it is capable of performing, and this can be compared with other results based on standard observation techniques from ethology, psychology, and cognitive science.

Development facilitates learning

Some of the most significant insights development provides us as robot designers are identifying the constraints, interactions, and strategies infants exploit to promote learning in complex environments. Human development is fundamentally about scaling. The range of behaviors that a human infant acquires within the first year of life easily dwarfs the complete range of behaviors that current robotic systems offer. Developmental approaches make explicit the flexibility that our robotic systems require, and offer insights on how simpler behaviors may bootstrap more complex behaviors.

Perhaps the most obvious insight is that development is an *incremental* process. At each stage in the process, prior structures and their behavioral manifestations place important constraints on the later structures and proficiencies. Knowledge that is gained from learning a precursor behavior can be recruited by a more complex behavior to simplify the learning process, or simpler behaviors may serve as construction blocks for later behaviors. The earlier forms bootstrap the later structures by providing subskills and knowledge that can be re-used. By following the developmental progression, the learning difficulties at each stage are minimized.

Development provides a gradual increase in complexity

Another key insight is that it is not a disadvantage to start with an immature and uncalibrated system. A system that begins simple and becomes gradually more complex allows for more efficient learning. For example, infants are born with low acuity vision which simplifies the visual input they must process. The infant's visual performance develops in step with their ability to process the influx of stimulation (Johnson 1993). The same is true for the motor system. Newborn infants do not have independent control over each degree of freedom of their limbs, but through a gradual increase in the granularity of their motor control they learn to coordinate the full complexity of their bodies. A process where the acuity of both sensory and motor systems are gradually increased significantly reduces the difficulty of the learning problem (Thelen 1993).

Development does not only consist of a gradual increase in internal complexity, but also a gradual increase in the complexity of the external world. The complexity of the infant's external world is carefully structured by the caregiver. By carefully controlling the environment, the caregiver can bias how the learning proceeds. This approach is in stark contrast to most machine learning methods, where the robot learns in a usually hostile environment, and the bias, instead of coming from the robot's interaction with the world, is included by the designer. By gradually increasing the complexity of the environment, learning will be made easier, and the robustness of learning will also be improved.

By exploiting a gradual increase in complexity both internal and external, while reusing structures and information gained from previously learned behaviors, we hope to be able to learn increasingly sophisticated behaviors. We believe that these methods will allow us to construct systems which scale autonomously.

Examples of Developmental Systems

Our group has been exploring building complex behaviors by studying human development. The following sections detail two examples of how we have been using studies of human development to build complex, interactive behavior.

The common hardware platform that we have used for these studies is the upper-torso humanoid robot called Cog (see Figure 1). The robot approximates a human being from the waist up with twenty-one degrees-of-freedom (DOF) and a variety of sensory systems. Mechanically, Cog has a torso with a two degree-of-freedom (DOF) waist, a one DOF torso twist, and a three DOF neck. Two six DOF arms, each joint powered by a motor through a series torsional spring, exhibit a natural compliant behavior similar to human arms (Williamson 1995). The head assembly consists of a three DOF active vision system with two active "eyes," each eye composed of two (a wide peripheral and a narrow foveal) CCD cameras. The head also contains a vestibular system, with three rate gyroscopes and two inclinometers, and an auditory system with two microphones and crude pinnae.

The robot's "brain" is a heterogeneous parallel processor network that contains dedicated sensory processors, for visual and auditory computation, as well as general purpose processors for motor control and higher level computation. In addition, much of the local motor control is handled on dedicated on-board controllers for faster response.

In addition to the full humanoid robot, we have also developed active head platforms, similar in mechanical design to Cog's head, with identical computational systems; the same code can be run on all platforms. Sections of the following work were developed and tested using these systems.

Example #1: Visually-Guided Pointing

Taking a strictly engineering approach to the problem of visually-guided reaching, our task can be formulated as follows: Given the location of an interesting stimulus within the image plane, we desire to move the arm to an extended posture that intersects that position. If we were to follow a classical robotics approach to this problem, we would take the two-dimensional position of the object within the image plane as our input, and convert that to a trajectory through eleven degrees of freedom (two from the eye, three from the neck, and six from the arm). We could determine the kinematics and dynamics of this system and compute the requisite function. However, this solution is very complex, incapable of adapting to new configurations of the robot, and gives no insight of how to generalize to other hand-eye coordination tasks. Instead, we look to infant development for examples of how to build these behaviors.

Diamond has shown that children between the ages of 5 and 12 months undergo a number of distinct developmental stages (Diamond 1990) . By the age of 5 months, the infant has already gained a significant amount of motor control over her eyes and neck, but is just beginning to use her arms to reach deliberately for objects. The first step in this progression is a very stereotyped reach along the visual axis directly toward the object of interest. During this stage, the infant's reach is ballistic; once the reach has begun, the trajectory is modified only once the infant comes into contact with an object. Children at this stage always begin a reach from a rest position in front of their bodies. If they miss the target, they return to the rest position and try again.

To achieve visually-guided pointing with our humanoid robot, we construct a system that first learns to foveate the visual target, and then to orient to that target, and finally to reach for that target. The learning is done on-line and completely self-supervised as the robot attempts to reach for targets. The original work was conducted with the neck in a fixed position, eliminating 3 degrees of freedom from the problem. With this assumption, the training takes place in the following two phases:

1. The system learns to saccade to a visual target. An image correlation algorithm constructs a saccade map $\vec{S} : \vec{x} \rightarrow \vec{e}$, which relates positions in the camera image $\vec{x} = (x, y)$ with the motor commands necessary to foveate the eye at that location $\vec{e} = (pan, tilt)$.
2. The system learns to ballistically reach along the angle of gaze. Using motion feedback from the moving arm, the system learns a ballistic mapping $\vec{B} : \vec{e} \rightarrow \vec{\alpha}$ between the position of the eyes in the head $\vec{e} = (pan, tilt)$ and the arm motor commands necessary to move the arm to that location $\vec{\alpha} = (\alpha_0 ... \alpha_5)$.

Following the developmental progression simplifies the second step of this task, as we will see below. More information on this project can be found in (Marjanović, Scassallati & Williamson 1996).

To learn the saccade map, the system simply attempts to foveate objects and learns from its mistakes. This map is implemented as a 17×17 interpolated lookup table, which is trained by the following algorithm:

1. Initialize with a linear map obtained from self-calibration.
2. Randomly select a visual target.
3. Saccade using the current map.
4. Find the target in the post-saccade image using correlation.
5. Update the saccade map based on L_2 error.
6. Go to step 2.

The system converges to an average of < 1 pixel of error per saccade after 2000 trials (1.5 hours).

To learn the ballistic map incrementally, we face three fundamental problems. First, we require an unsupervised error signal. We accomplish this by using a motion detection routine to find the end of the hand as it reaches out. Since the eyes are already pointing toward the target, the error is simply the vector from the center of the image to the projection of the hand on the image plane. The second problem that we face is that our error signal is in the coordinates of the image plane. We solve this by re-using the saccade map that we learned in the previous step to convert the error vector from image coordinates to eye motor coordinates. That is, we use the saccade map to determine the position that we would need to look to in order to foveate the hand. The saccade map transforms our error metric into one of the coordinate systems used by the ballistic map. However, our third problem is that the error signal is in the coordinate frame of the input to our mapping, not the coordinates of the output. To train a system in this way, we use a learning technique similar to the distal supervised learning of (Jordan & Rumelhart 1992); we train a forward map of the system $\vec{F}$ which is the inverse of the ballistic map $\vec{B}$. The forward kinematic model $\vec{F}$ is useful in that it gives an expectation of where to look to find the arm. This can be used to generate a window of attention to filter out distractions in the motion detection.

The maps $\vec{B}$ and $\vec{F}$ are both implemented using a radial basis function consisting of 64 Gaussian nodes distributed evenly over the input space. The nodes have identical variance, but are associated with different output vectors. The learning routine follows the following steps:

1. Initialize the maps $\vec{B}$ and $\vec{F}$.
2. Randomly select a visual target.
3. Saccade to that target using the learned saccade map.
4. Attempt to reach for the target using the current ballistic mapping.
5. As the arm moves, track the end of the arm using visual motion detection.
6. Once the arm has finished reaching, compute the visual error signal $\vec{x}$ between the center of the visual field and the arm position.
7. Transform the visual error from image coordinates $\vec{x}$ to eye motor coordinates $\vec{e}$ using the saccade map.
8. Backpropagate $\vec{e}$ through the forward map $\vec{F}$ to generate an error signal in terms of the arm coordinates $\vec{a}$.
9. Backpropagate the error in arm coordinates $\vec{a}$ through the Ballistic map.
10. Return the arm to the resting position.
11. Go to step 2.

The system trains to reach reliably within approximately 700 reaches, which requires 90 minutes of training. This system is not yet capable of generic hand-eye coordination, but the developmental progression that we have modeled here provides insights on how to generalize this skill (see (Diamond 1990)).

Although the developmental framework in this task greatly simplifies the computation necessary, the ballistic reaching task could be accomplished with a brute-force approach using current technology without resorting to a developmental approach. In the next section, we look at a task that has not been adequately addressed in current technology: enabling a machine to interact with a naive human in a natural way.

Example #2: Building Social Skills Developmentally

In our work, we have also examined social interaction from this developmental perspective. One research program focuses on a developmental implementation of an early precursor of social communication, shared attention mechanisms. Shared attention, the ability to selectively attend to an object of mutual interest, is part of the large repertoire of social cues, such as gaze direction, pointing gestures, and postural cues, that humans so easily master. The primary focus of this research is to investigate how individuals develop the skills to recognize and produce these social cues by implementing models of this developmental progression on our humanoid robot. A more detailed account of this project can be found in (Scassellati 1996).

We are interested in shared attention as a precursor to social communication for two reasons. First, we believe that by using a developmental program to build social capabilities we will be able to achieve a wide range of natural interactions with untrained observers (Brooks, Ferrell, Irie, Kemp, Marjanovic, Scassellati & Williamson 1998). Constructing a machine that can recognize the social cues from a human observer allows for more natural human-machine interaction and creates possibilities for machines to learn by directly observing untrained human instructors. Second, by building models from developmental psychology, we further these models by providing a test bed for manipulating the behavioral progression. With an implemented developmental model, we can test alternative learning conditions, environmental conditions, and abnormal developmental progressions. This investigation of shared attention asks questions about the development and origins of the complex non-verbal communication skills that humans so easily master: What is the progression of skills that humans must acquire to engage in shared attention? When something goes wrong in this development, as it seems to do in Autism, what problems can occur, and what hope do we have for correcting these problems? What parts of this complex interplay can be seen in other primates, and what can we learn about the basis of communication from these comparisons?

A Developmental Model of Shared Attention The mechanisms for shared attention in humans are not a single monolithic system. Evidence from childhood development shows that not all mechanisms for shared attention are present from birth (Hobson 1993). There are also developmental disorders, such as Autism,[1] that appear to affect shared attention (Frith 1990).

We begin with a developmental model from (Baron-Cohen 1995). Baron-Cohen provides a model that gives a coherent account of the observed developmental stages of shared attention behaviors in both normal and blind children, the observed deficiencies in shared attention of Autistic children, and a partial explanation of the observed abilities of primates on shared attention tasks. What Baron-Cohen's model does not provide is a task-level decomposition of necessary skills and the developmental mechanisms that provide for transition between his stages. Our current work is on identifying and implementing a developmental account of one possible skill decomposition (Scassellati 1996). The skill decomposition that we are pursuing can be broken down into four stages: gaze monitoring, interpolation of gaze, identifying pointing, and utilizing pointing.

The first step in producing mechanisms of shared attention is gaze monitoring, the tendency to monitor what the caregiver is looking at. Just as human infants have an innate preference for looking at human faces and eyes, our robot has a hard-wired preference to look at human faces and eyes. In its simplest form, this skill is the ability to tell when someone is looking at you. This requires a great deal of perceptual ability, but relatively little motor control.

The second step will be to engage in shared attention by interpolation of gaze. Povinelli and Preuss report that in all of the great apes, when a caretaker moves its gaze to a new location, the child will move its gaze to that same location (Povinelli & Preuss 1995). This basic form of imitation serves to focus the child's attention on the same object that the caregiver is attending to. This functional imitation appears simple, but involves many separate proficiencies, as we will see in the following section.

The third step in our account will be to engage in shared attention through pointing. Just as gaze direction can indicate a request for shared attention, a pointing gesture is a request for the observer to attend to a distal object. To understand pointing as a mechanism of shared attention requires both the ability to recognize postural and gestural cues but also the ability to extrapolate from that posture to a distal object. As a singular skill, this is as complex as gaze following. However, including this skill as part of a developmental program makes the task more attainable; the infrastructure for extrapolation from a body cue is already present from the first two stages, it need only be applied to a new domain.

The final step is to enable our robot to *request* shared attention by pointing at an object. This step adds more complex computational and motor control requirements to our system, but the skills from the first three stages can be combined with imitation to achieve requests for shared attention. By imitating the successful pointing gestures of other individuals, the robot can learn to make use of similar gestures. Our robot has already learned the sensori-motor coordination necessary to point to a visual target (see the previous example). Adding the ability to imitate pointing requests will require the ability to perceive relevant gestures, to map the perceived gestures to gestures that the robot is capable of performing, and to produce new gestures at an appropriate time.

Current Results The hardware platform that we use for vision is a binocular, foveated, active vision system (Scassellati 1998*a*) shown in Figure 1.[2] There are two cameras per eye, one which captures a wide-angle view of the periphery (approximately 110° field of view) and one which captures a narrow-angle view of the central (foveal) area (approximately 20° field of view with the same resolution).

Implementing the first developmental stage requires finding faces and eyes. The strategy that we use can be decomposed into the following five steps:

1. Use a motion-based pre-filter to identify potential face locations in the peripheral image.
2. Use a ratio-template based face detector to identify target faces.
3. Saccade to the target using a learned sensori-motor mapping.
4. Convert the location in the peripheral image to a foveal location using a learned mapping.
5. Extract the image of the eye from the foveal image.

Further details about this method can be found in (Scassellati 1998*b*).

To identify face locations, the peripheral image is converted to grayscale and passed through a pre-filter stage. The pre-filter allows us to search only locations

[1] While the deficits of Autism certainly cover many other areas of perceptual and cognitive capabilities, some researchers believe that the missing mechanisms of shared attention may be critical to the other deficiencies (Baron-Cohen 1995). In comparison to other mental retardation and developmental disorders (like Williams and Downs Syndromes), the deficiencies of Autism in this area are quite specific (Karmiloff-Smith, Klima, Bellugi, Grant & Baron-Cohen 1995).

[2] Two additional copies of this platform exist as desktop development platforms. While there are minor differences between the platforms, these are not important to the work reported here. Some of the results in this section were obtained from those platforms.

that are likely to contain a face, greatly improving the speed of the detection step. The pre-filter selects a location as a potential target if it has had motion in the last 4 frames, was a detected face in the last 5 frames, or has not been evaluated in 3 seconds. A combination of the pre-filter and some early-rejection optimizations allows us to run face detection at 20 Hz with little accuracy loss.

Face detection is done with a method called "ratio templates" designed to recognize frontal views of faces under varying lighting conditions (Sinha 1996). A ratio template is composed of a number of regions and a number of relations, as shown in Figure 2. Overlaying the template with a grayscale image, each region is convolved with the grayscale image to give the average grayscale value for that region. Relations are comparisons between region values, such as "the left forehead is brighter than the left temple." In Figure 2, each arrow indicates a relation, with the head of the arrow denoting the lesser value. The match metric is the number of satisfied relations. The more matches, the higher the probability of a face.

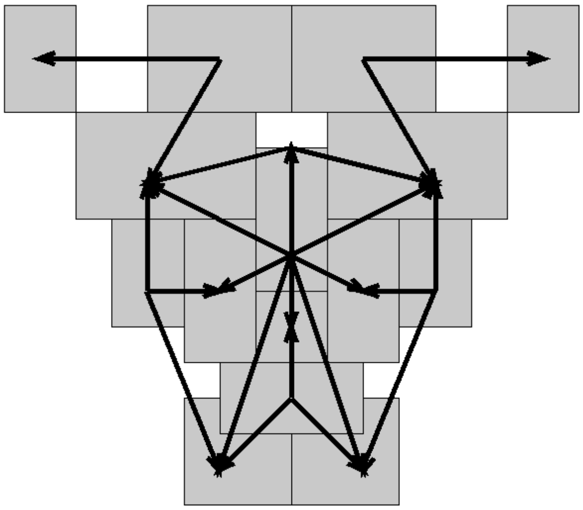

Figure 2: A ratio template for face detection. The template is composed of 16 regions (the gray boxes) and 23 relations (shown by arrows).

Once a face has been detected, the face location is converted into a motor command using the saccade map that was learned as part of the ballistic reaching task detailed in the previous section. After the active vision system has saccaded to the face, the face and eye locations from the template in the peripheral camera are mapped into the foveal camera using a second learned mapping. The mapping from foveal to peripheral pixel locations can be seen as an attempt to find both the difference in scales between the images and the difference in pixel offset. In other words, we need to estimate four parameters: the row and column scale factor that we must apply to the foveal image to match the scale of the peripheral image, and the row and column offset that must be applied to the foveal image within the peripheral image. This mapping can be learned in two steps. First, the scale factors are estimated using active vision techniques. While moving the motor at a constant speed, we measure the optic flow of both cameras. The ratio of the flow rates is the ratio of the image sizes. Second, we use correlation to find the offsets. The foveal image is scaled down by the discovered scale factors, and then correlated with the peripheral image to find the best match location.

Once this mapping has been learned, whenever a face is foveated we can extract the image of the eye from the foveal image. This extracted image is then ready for further processing. Figure 3 shows the result of the face detection routines on a typical grayscale image before the saccade. Figure 4 shows the extracted image of the eye that was obtained after saccading to the target face.

Figure 3: An example of the face detector. The 128x128 grayscale image was captured by the active vision system, and then processed by the pre-filtering and ratio template detection routines. One face was found within the image, and is shown outlined.

This work is still in progress. To accomplish gaze monitoring, we are in the process of adding the capabilities to detect eye contact and to extrapolate the angle of gaze from the orientation of the head and neck and the location of the pupil within the eye. However, this developmental structure has already given us interesting additional imitative capabilities. By adding a tracking mechanism to the output of the face detector and then classifying these outputs, we have been able to have the system mimic yes/no head nods of the caretaker. As the caretaker shakes his head yes, the robot will also shake its head yes. While this is a very simple form of imitation, it is highly selective. Merely producing horizontal or vertical movement is not sufficient for the head to mimic the action – the movement must come from a face-like ob-

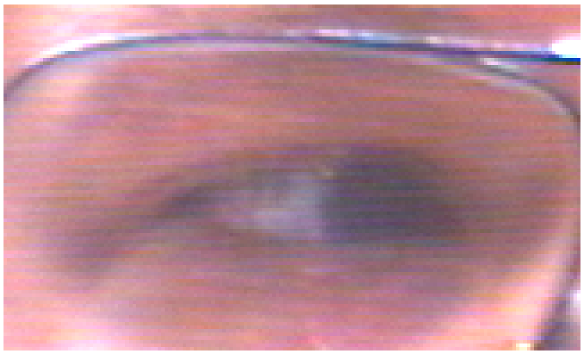

Figure 4: Extracted image of the eye from the foveal image.

ject. Video clips of this imitation are available from http://www.ai.mit.edu/projects/cog/Text/video-index.html.

Conclusion

This paper has advocated a developmental approach to building complex interactive behaviors for robotic systems. We focused on three reasons that the developmental methodology is advantageous: because it provides a structured decomposition of complex tasks, because it facilitates learning, and because it allows for a gradual increase in task complexity.

We have motivated a deeper exploration of the developmental methodology with two examples from our humanoid robot project. In learning to reach to a visual target, the developmental methodology provided a structured system for re-applying useful knowledge and in limiting the dimensional complexity of our task. In building mechanisms of shared attention, we can begin to see a means of decomposing complex social skills into a set of manageable tasks, and a way in which these tasks can facilitate the bootstrapping of more advanced skills. In both these examples, a developmental perspective has enabled our group to tackle complex task domains with a structured, formal methodology.

References

Baron-Cohen, S. (1995), *Mindblindness*, MIT Press.

Brooks, R. A., Ferrell, C., Irie, R., Kemp, C. C., Marjanovic, M., Scassellati, B. & Williamson, M. (1998), Alternative Essences of Intelligence, *in* 'Proceedings of the American Association of Artificial Intelligence'. In submission.

Diamond, A. (1990), *Development and Neural Bases of Higher Cognitive Functions*, Vol. 608, New York Academy of Sciences, chapter Developmental Time Course in Human Infants and Infant Monkeys, and the Neural Bases, of Inhibitory Control in Reaching, pp. 637–676.

Frith, U. (1990), *Autism : Explaining the Enigma*, Basil Blackwell.

Hobson, R. P. (1993), *Autism and the Development of Mind*, Erlbaum.

Johnson, M. H. (1993), Constraints on Cortical Plasticity, *in* M. H. Johnson, ed., 'Brain Development and Cognition: A Reader', Blackwell, Oxford, pp. 703–721.

Jordan, M. I. & Rumelhart, D. E. (1992), 'Forward Models: supervised learning with a distal teacher', *Cognitive Science* **16**, 307–354.

Karmiloff-Smith, A., Klima, E., Bellugi, U., Grant, J. & Baron-Cohen, S. (1995), 'Is there a social module? Language, face processing, and theory of mind in individuals with Williams Syndrome', *Journal of Cognitive Neuroscience* **7:2**, 196–208.

Marjanović, M. J., Scassallati, B. & Williamson, M. M. (1996), Self-Taught Visually-Guided Pointing for a Humanoid Robot, *in* 'From Animals to Animats: Proc 1996 Society of Adaptive Behaviour', Society of Adaptive Behavior.

Povinelli, D. J. & Preuss, T. M. (1995), 'Theory of mind: evolutionary history of a cognitive specialization', *Trends in Neuroscience*.

Scassellati, B. (1996), Mechanisms of Shared Attention for a Humanoid Robot, *in* 'Embodied Cognition and Action: Papers from the 1996 AAAI Fall Symposium', AAAI Press.

Scassellati, B. (1998*a*), A Binocular, Foveated Active Vision System, Technical report, MIT Artificial Intelligence Lab. In submission.

Scassellati, B. (1998*b*), Finding Eyes and Faces with a Foveated Vision System, *in* 'Proceedings of the American Association of Artificial Intelligence'. In submission.

Sinha, P. (1996), Perceiving and recognizing three-dimensional forms, PhD thesis, Massachusetts Institute of Technology.

Thelen, E. (1993), Self-Organization in Developmental Processes: Can Systems Approaches Work?, *in* M. H. Johnson, ed., 'Brain Development and Cognition: A Reader', Blackwell, Oxford, pp. 555–591.

Williamson, M. (1995), Series Elastic Actuators, Master's thesis, MIT Department of Electrical Engineering and Computer Science.