Eye Finding via Face Detection for a Foveated, Active Vision System

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

Abstract

Eye finding is the first step toward building a machine that can recognize social cues, like eye contact and gaze direction, in a natural context. In this paper, we present a real-time implementation of an eye finding algorithm for a foveated active vision system. The system uses a motion-based prefilter to identify potential face locations. These locations are analyzed for faces with a template-based algorithm developed by Sinha (1996). Detected faces are tracked in real time, and the active vision system saccades to the face using a learned sensorimotor mapping. Once gaze has been centered on the face, a high-resolution image of the eye can be captured from the foveal camera using a self-calibrated peripheral-to-foveal mapping.

We also present a performance analysis of Sinha's ratio template algorithm on a standard set of static face images. Although this algorithm performs relatively poorly on static images, this result is a poor indicator of real-time performance of the behaving system. We find that our system finds eyes in 94% of a set of behavioral trials. We suggest that alternate means of evaluating behavioral systems are necessary.

Introduction

The ability to detect another creature looking at you is critical for many species. Many vertebrates, from snakes (Burghardt 1990), to chickens (Ristau 1991), to primates (Povinelli & Preuss 1995), have been observed to change their behavior based on whether or not eyes are gazing at them. In humans, eye contact serves a variety of social functions, from indicating interest to displaying aggression (Nummenmaa 1964).

Eye direction can also be a critical element of social learning. Eye direction, like a pointing gesture, serves to indicate what object an individual is currently considering. While infants initially lack many social conventions (understanding pointing gestures may not occur until the end of the first year), recognition of eye contact is present from as early as the first month (Frith 1990; Thayer 1977). Detection of eye direction is believed to be a critical precursor of linguistic development (Scaife & Bruner 1975), theory of mind (Baron-Cohen 1995), and social learning and scaffolding (Wood, Bruner, & Ross 1976).

This paper presents the first steps on a developmental progression for building robotic systems that can utilize eye direction as a social signal (Scassellati 1996). The initial goal of our system is to obtain a high resolution image that contains an eye for further processing. We present an algorithm for finding faces and eyes in a cluttered environment. The algorithm that we present has been implemented on a binocular, foveated active vision system (Scassellati 1998), which is part of a humanoid robot project (Brooks & Stein 1994; Brooks *et al.* 1998).

Overview

The nature of the active vision system constrains our implementation (Ballard 1989). Three copies of this hardware system are currently in use, one on a humanoid robot (see Figure 1) and two as desktop development platforms (see Figure 2). Each has an identical computational environment and very similar mechanical and optical properties (Scassellati 1998). Similar to other active vision systems (Sharkey *et al.* 1993; Coombs 1992), there are three degrees of freedom; each eye has an independent vertical axis of rotation (pan) and the eyes share a joint horizontal axis of rotation (tilt). We use a foveated vision system to gain both a wide field of view while retaining a high acuity central area, which is a rough approximation of the unequal distribution of photoreceptors on the human retina (Tsotsos 1988). Unlike other foveated systems (Kuniyoshi *et al.* 1995; van der Spiegel *et al.* 1989), there are two cameras per eye, one which captures a wide-angle view of the periphery and one which captures a narrow-angle view of the central (foveal) area.[1]

The active vision platform is attached to a parallel network of digital signal processors (Texas Instruments TMS320C40). Each node in the network contains one processor with the option for more specialized hardware

[1] The peripheral camera has an approximate field of view of 120°, while the foveal camera has an approximate field of view of 20°.

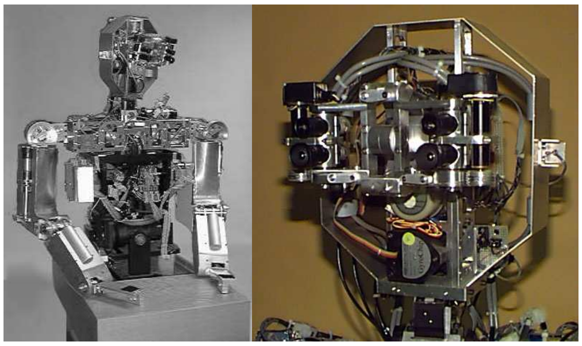

Figure 1: At left, Cog, an upper-torso humanoid robot. At right, a close-up of Cog's active vision system.

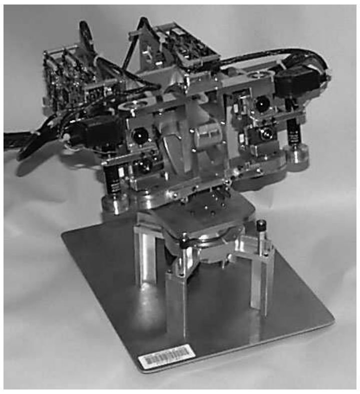

Figure 2: One of the desktop active vision development platforms used in this work.

for capturing images, performing convolution quickly, or displaying images to a VGA display. Nodes may be connected with arbitrary bi-directional hardware connections, and distant nodes may communicate through virtual connections. Each camera is attached to its own frame grabber, which can transmit captured images to connected nodes.

The initial goal of our system is to obtain high resolution images of the eyes of a person anywhere in the robot's environment. Because the peripheral camera has a very wide field of view (see Figure 5), we cannot extract eye features from this image. Just as humans must foveate an object to discriminate fine detail, our foveal cameras must be pointed in the direction of a given object in order to provide sufficient resolution. Other research has focused on the tracking of eyes and facial features for video conferencing (Graf *et al.* 1996; Maurer & von der Malsburg 1996), as a user interface (Baluja & Pomerleau 1994; Heinzmann & Zelinsky 1997), or in animation (Terzopoulous & Waters 1991), however, these techniques generally begin with calibrated high resolution images where the face dominates the visual field. Our behavioral goal also provides a constraint on the speed and accuracy of the processing; we are not as concerned with missing a face in a single image since we will have another opportunity to detect in the next frame. Highly accurate (and computationally expensive) techniques for face and eye detection (Rowley, Baluja, & Kanade 1995; Turk & Pentland 1991; Sung & Poggio 1994) may not be necessary. Finally, to better fit with the goals of building social skills developmentally (Scassellati 1996), we prefer an implementation that is biologically plausible.

Our strategy for finding eyes decomposes into the following five steps:

1. The incoming wide-field image is filtered using motion and past history to find potential face locations.
2. For each potential face location, a face detection algorithm based on ratio templates (Sinha 1996) is used to verify the presence of a face.
3. The face location with the highest score is selected and the active vision system saccades to that face using a learned sensorimotor mapping.
4. With the template estimate of the eye locations, and a self-calibrated peripheral-to-foveal mapping, the location of the eye in the foveal image is computed.
5. A high-resolution foveal image of the eye is captured for further processing.

We begin with a detailed discussion of the ratio template face detection algorithm, and then discuss the prefiltering technique that we use to enable the algorithm to run in real time.

Finding Faces

Our choice of a face detection algorithm was based on three criteria. First, it must be a relatively simple computation that can be performed in real time. Second, the technique must perform well under social conditions, that is, in an unstructured environment where people are most likely to be looking directly at the robot. Third, it should be a biologically plausible technique. Based on these criteria, we selected the ratio template approach described by Sinha (1994).

The ratio template algorithm was designed to detect frontal views of faces under varying lighting conditions, and is an extension of classical template approaches (Sinha 1996). While other techniques handle rotational invariants more accurately (Sung & Poggio 1994), the simplicity of the ratio template algorithm allows us to operate in real time while detecting faces that are most likely to be engaged in social interactions. Ratio templates also offer multiple levels of biological plausibility; templates can be either hand-coded or learned adaptively from qualitative image invariants (Sinha 1994).

A ratio template is composed of a number of regions and a number of relations, as shown in Figure 3. For each target location in the grayscale peripheral image, a template comparison is performed using a special set of comparison rules. Overlaying the template with a 14 pixel by 16 pixel grayscale image patch at a potential face location, each region is convolved with the grayscale image to give the average grayscale value for that region. Relations are comparisons between region values, for example, between the "left forehead" region and the "left temple" region. The relation is satisfied if the ratio of the first region to the second region exceeds a constant value (in our case, 1.1). This ratio allows us to compare the intensities of regions without relying on the absolute intensity of an area. In Figure 3, each arrow indicates a relation, with the head of the arrow denoting the second region (the denominator of the ratio). This template capitalizes on illumination-invariant observations. For example, the eyes tend to be darker than the surrounding face, and the nose is generally brighter than its surround. We have adapted the ratio template algorithm to process video streams. In doing so, we require the absolute difference between the regions to exceed a noise threshold, in order to eliminate false positive responses for small, noisy grayscale values.

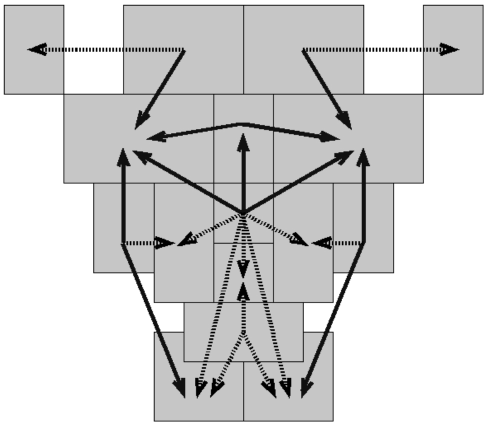

Figure 3: A 14 pixel by 16 pixel ratio template for face detection. The template is composed of 16 regions (the gray boxes) and 23 relations (shown by arrows). Adapted from Sinha (1996).

The ratio template algorithm can easily be modified to detect faces at multiple scales. Multiple nodes of the parallel network run the same algorithm on different sized input images, but without changing the size of the template. This allows the system to respond more quickly to faces that are closer to the robot, since closer faces are detected in smaller images which require less computation. With this hardware platform, a 64×64 image and a 14×16 template can be used to detect faces within approximately three to four feet of the robot. The same size template can be used on a 128×128 image to find faces within approximately ten feet of the robot.

Improving the Speed of Ratio Templates

To improve the speed of the ratio template algorithm, we have implemented two optimizations: an early-abort scheme and a motion-based prefilter.

At the suggestion of Sinha (1997), we further classified the relations of our ratio-template into two categories: eleven essential relations, shown as solid arrows in Figure 3, and twelve confirming relations, shown as dashed arrows. We performed a post-hoc analysis of this division upon approximately ten minutes of video feed in which one of three subjects was always in view. For this post-hoc analysis, an arbitrary threshold of eighteen of the twenty-three relations was required to be classified as a face. This threshold eliminated virtually all false positive detections while retaining at least one detected face in each image. An analysis of the detected faces indicated that at least ten of the eleven essential relations were always satisfied. None of the confirming relations achieved that level of specificity. Based on this analysis, we established a new set of thresholds for face detection: ten of the eleven essential relations and eight of the twelve confirming relations must be satisfied. As soon as two or more of the essential relations have failed, we can reject the location as a face. This increases the speed of our computation by a factor of 4, as shown in Table 1, without any observable decrease in performance.

To further increase the speed of our computation, we use a pre-filtering technique based on motion. The prefilter allows us to search only locations that are likely to contain a face. Consecutive images are differenced, thresholded, and then convolved with a 14×16 kernel of unitary value (the same size as the ratio template) in order to generate the average amount of super-threshold movement for each potential face location. If that average motion value for a location exceeds threshold, then that location is a candidate for face detection. For each incoming frame, a location is a potential target for the face detection routine if it has had motion within the last five frames, if the ratio template routine verified a face in that location within the last five frames, or if that location had not been checked for faces within the last three seconds. In this way, we capture faces that have just entered the field of view (through the motion clause) and faces that have stopped moving (through the past history clause). The prefilter also resets every three seconds, allowing the system to re-acquire faces that have dropped below the noise threshold. The prefilter automatically resets any time the active system moves, since this generates induced motion of the visual field. This filtering technique increased the speed of the face detection routines by a factor of five for 64×64 images and a factor of eight for 128×128 images (see Table 1). The smaller image size appeared to saturate at 20 Hz due to constant computational loads in the rest of the system, primarily from drawing display images to a VGA display. The filtering technique greatly reduced the number of background locations to be searched without any observable loss of accuracy.

Static Evaluation of Ratio Templates

To evaluate the static performance of the ratio template algorithm, we ran the algorithm on a test set of static

Image Size	Detection Method		
	Template	+ Early-Reject	+ Prefilter
64×64	1 Hz	4 Hz	20 Hz
128×128	.25 Hz	1 Hz	8 Hz

Table 1: Processing speed for two image sizes with various optimizations. The original ratio template method is enhanced by a factor of four with the addition of the early-reject optimization, and by an additional factor of five to eight by the prefilter optimization. The system saturated near 20 Hz due to constant computational loads in other parts of the network. All statistics are for a single TMS320C40 node with no other processes.

face images first used by Turk and Pentland (1991). The database contains images for 16 subjects, each photographed under three different lighting conditions and three different head rotations.

To test lighting invariance, we considered only the images with an upright head position at a single scale, giving a test set of 48 images under lighting conditions with the primary light source at 90 degrees, 45 degrees, and head-on. Figure 4 shows the images from two of the subjects under each lighting condition. The ratio template algorithm detected 34 of the 48 test faces. Of the 14 faces that were missed, nine were the result of three subjects that failed to be detected under any lighting conditions. One of these subjects had a full beard, while another had very dark rimmed glasses, both of which seem to be handled poorly by the static detection algorithm. Of the remaining five misses, two were from the 90 degree lighting condition, two from the 45 degree lighting condition, and one from the head-on condition. While this detection rate (71%) is considerably lower than other face detection schemes (Rowley, Baluja, & Kanade 1995; Turk & Pentland 1991; Sung & Poggio 1994), this result is a poor indicator of the performance of the algorithm in a complete, behaving system, as we will see below.

Using the real-time system, we determined approximate rotational ranges of the ratio template algorithm. Subjects began looking directly at the camera and then rotated their head until the system failed to detect a face. Across three subjects, the average ranges were ± 30 degrees pitch, ± 30 degrees yaw, and ± 20 degrees roll.

Finding Eyes

Once a face has been detected, the active vision system must accurately saccade to that location, bringing the foveal camera into position to capture a high-resolution image of the eye. To solve this sensorimotor problem, we could build an accurate kinematic and dynamic model of the robotic system and use that model to compute saccade motions. However, the kinematic solution is difficult to characterize accurately, taking into account the misalignments of the cameras, the optical imperfections, and the imperfect construction of any real system (Ballard 1989). An accurate kinematic solution is also extremely hardware dependent; the kinematic solution for one of the development platforms would differ significantly from the solution for the active vision system of the humanoid robot.

Figure 4: Six of the static test images from Turk and Pentland (1991) used to evaluate the ratio template face detector. Each face appears in the test set with three lighting conditions, head-on (top), from 45 degrees (middle), and from 90 degrees (bottom). The ratio template correctly detected 71% of the faces in the database, including each of these faces except for the middle image from the first column.

An alternative is to learn the functional mapping between the position of the target on the image plane and the motor commands necessary to foveate that object. A learned solution can be adapted not only for each instance of a hardware platform, but also each time a specific hardware platform undergoes some kind of change. We desire the system to be completely self-supervised and on-line so that learning can proceed continuously and without any human intervention.

Because the ratio template algorithm gives us an implicit location for the eyes within a verified face, we need to solve only two problems: how to learn to saccade to the target face, and how to map the locations in the peripheral image to locations in the foveal image. We choose to saccade to the target face, and then to relocate the position of the face in the peripheral image, not only to resist noise in the saccade mapping but also to allow for slight movements of the subject during the saccade.

Figure 6: Six detected faces and eyes. The lower image of each pair shows the post-saccade location of the detected face. The upper image of each pair shows the section of the foveal image obtained from mapping the peripheral template location to the foveal coordinates. Only faces of a single scale (roughly within four feet of the robot) are shown here.

Figure 5: An example face in a cluttered environment. The 128x128 grayscale image was captured by the active vision system, and then processed by the prefiltering and ratio template detection routines. One face was found within the image, and is shown outlined.

Saccading to a Face

The problem of saccading to a visual target can be viewed as a function approximation problem, where the equation

$$\vec{S}(\vec{e}, \vec{x}) \mapsto \Delta\vec{e} \qquad (1)$$

defines the saccade function $\vec{S}$ which transforms the current motor positions $\vec{e}$ and the location of a target stimulus in the image plane $\vec{x}$ to the change in motor position necessary to move that target to the center of the visual field.

Marjanović, Scassellati, and Williamson (1996) learned a saccade function for this hardware platform using a 17×17 interpolated lookup table. The map was initialized with a linear set of values obtained from self-calibration. For each learning trial, a visual target was randomly selected. The robot attempted to saccade to that location using the current map estimates. The target was located in the post-saccade image using correlation, and the L_2 offset of the target was used as an error signal to train the map. The system learned to center pixel patches in the peripheral field of view. The system converged to an average of < 1 pixel of error in a 128×128 image per saccade after 2000 trials (1.5 hours). With this map implementation, a face could be centered in the peripheral field of view. However, this does not necessarily place the eye in a known location in the foveal field of view. We must still convert an image location in the peripheral image to a location in the foveal image.

Mapping Peripheral to Foveal Images

After the active vision system has saccaded to a face, the face and eye locations from the template in the peripheral camera are mapped into the foveal camera using a second mapping. This mapping matches a rectangular region of pixels in the peripheral image with a (larger) rectangular region of pixels in the foveal image. To estimate this function, four parameters must be determined: the scale factor in each dimension between the peripheral and foveal images (to determine the size of the foveal rectangle), and the offset between the peripheral rectangle and the foveal rectangle in each dimension.

To estimate the difference in scale between the two cameras, we exploit the active nature of the system. We can estimate the difference in scale factors by moving the eyes at a constant velocity and observing the rates of optical flow of a background patch. The ratio of the flow rates is the ratio of the scale between the cameras. To accomplish this, the system first verifies that there is no motion within the field of view, using the motion detection routines established for the prefilter. The system then selects a patch of background pixels from the center of the field of view. Once the eye begins moving, a simple correlation-based tracking algorithm monitors the rate that the peripheral and foveal patches move. While it is possible to estimate both the horizontal and vertical scale factors in one movement, the initial implementation of this system used separate horizontal and vertical motions to estimate the scale parameters. For both the development platform and the active vision head of the humanoid robot, the scale factors in both the horizontal and vertical dimensions was 4.0. The scale factor is a function of the differences in the cameras and lenses, not in the alignment of the cameras. This value is thus stable between the eyes and throughout the lifetime of the system.

Once we have an estimation of the scale factors, we can find the differences in offset position using image correlation. The foveal image is scaled down by the computed ratios. The scaled-down image can then be correlated with the peripheral image to find the best match location. Because the foveal and peripheral cameras are aligned vertically, the depth of the patch being considered can affect the row position of the best match. For our initial estimates of this function, we record the correlation when the ratio template algorithm, running at a specific scale, has detected a face. This gives us an estimate of where the face occurs in the foveal image for faces of this size. This method is imprecise, but obtains results that are sufficient for the overall behavior of the complete system. For one ratio template scale running on the development platform, the best match for the left eye was located at an offset of 168 rows and 248 columns, and at an offset of 184 rows and 250 columns for the right eye (based on 512×512 images). We expect slightly different values for each eye, since the offset values are affected by misalignments in the individual camera mountings.

Once this mapping has been obtained, 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 5 shows the result of the face detection routines on a typical grayscale image before the saccade. After the face was detected, the system saccaded to that position, converted the template-specified eye location in the peripheral image to a foveal bounding box, and extracted the portion of the foveal image. Examples of this process are shown in Figure 6. While the peripheral image has barely enough information to detect the face, the foveal image contains a great deal of detail.

Evaluation

The evaluation of this system must be based on the behavior that it produces, which can often be difficult to quantify. The system succeeds when it eventually finds a face and is able to extract a high resolution image of an eye. However, to compare the performance of the entire system with the performance of the ratio template algorithm on static images, a strawman quantitative analysis of a single behavior was studied. To simplify the analysis, we considered only a single scale of face detection, requiring the subjects to sit within 4 feet of one of the active vision platforms. The subject was to remain stationary during each trial, but was encouraged to move to different locations between trials. These tests were conducted in the complex, cluttered background of our laboratory workspace (identical to Figure 5).

For each behavioral trial, the system began with the eyes in a fixed position, roughly centered in the visual field. The system was allowed one saccade to foveate the subject's right eye (an arbitrary choice). The system used the prefiltering and ratio template face detection routines to generate a stream of potential face locations. Once a face was detected, and remained stable (within an error threshold) for six cycles (indicating the person had remained stationary), the system attempted to saccade to that location. In a total of 140 trials distributed between 7 subjects, the system extracted a foveal image that contained an eye on 131 trials (94% accuracy). Of the missed trials, two resulted from an incorrect face identification (a face was falsely detected in the background clutter), and seven resulted from either an inaccurate saccade or motion of the subject.

This quantitative analysis of the system is extremely promising. However, the true test of the behavioral system is in eventually obtaining the goal. Even in this simple analysis, we can begin to see that the total behavior of the system may be able to correct for errors in individual components of the system. For example, one incorrect face identification was a temporary effect between part of the subject's clothing and the background. Once the system had shifted its gaze to the (false) face location, the location no longer appeared face-like. Without the arbitrary imposition of behavioral trials, the natural behavior of the system would then have been to saccade to what it did consider a face, achieving the original goal. Also, in some failures that resulted from incorrect saccades, the system correctly identified a face location but failed to center that location due to an incompletely trained saccade map. The system extracted only a portion of the eye in these cases. The saccade map is extremely non-linear in the far reaches of the visual field due to the extremely wide field of view of the peripheral cameras (as described in Scassellati (1998)). If the system had been allowed to continue behaving, rather than being forced into the next trial, it would again have detected a face, this time very close to the center of the field of view and within the well-trained region of the saccade map, and corrected for its slight error.

If our behavioral test had allowed for a second chance to obtain the goal, the failure rate can be estimated as the product of the failure rates for each individual trial. If we assume that these are independent saccades, the probability of failure for a two-attempt behavior becomes $0.06 \times 0.06 = .0036$. As we allow for more and more corrective behavior, the stability of the system increases. While individual trials are probably not completely statistically independent, we can see from this example how the behavior of the system can be self-stabilizing without requiring extremely accurate perceptual tools.

Issues like these make quantitative analysis of behaving systems difficult, and often misleading (Brooks 1991). Our system does not require a completely general-purpose face recognition engine. In a real-world environment, the humans to whom the robot must attend in order to gain the benefits of social interaction are generally cooperative. They are attempting to be seen by the robot, keeping their own attention focused on the robot, facing toward it, and often unconsciously moving to try to attract its attention. Further, the system need not be completely accurate on every timestep; its behavior need only converge to the correct solution. If the system can adequately recognize these situations, then it has fulfilled its purpose.

Future Work

The eye detection system presented here performs the localization task reasonably well. However, there are still many open research questions, and many implementation details that require additional support, including:

- A better arbitration scheme for multiple faces at multiple scales.
- Additional tolerance to face rotation.
- Depth-based prefiltering.
- A richer characterization of the peripheral to foveal map to include depth discrepancies.
- Continuous smooth-pursuit tracking for acquired faces.

These additions will enhance the richness of the overall behavior of the system.

This paper has presented the first step toward building social skills for a humanoid robot. With the ability to obtain high resolution images of eyes, it is possible for the system to detect eye contact and to identify where a person is looking. Building a general system that can recognize direction of gaze requires the following additional competencies:

- The ability to detect head and eye orientation.
- The ability to integrate head and eye orientation to produce a visual gaze angle.
- The ability to extrapolate a gaze angle toward an object in the world (perhaps utilizing a rough depth map).

These research areas in turn lead to more interesting social behaviors, allowing a human-like robot to interact with humans in a natural, social context and providing a mechanism for social learning.

Conclusions

This paper has presented an eye and face finding system for a foveated active vision system. The basic face detection algorithm was based on the ratio template design developed by Sinha (1996) and adapted for the recognition of frontal views of faces under varying lighting conditions. We have further developed this algorithm to increase the processing speed by a factor of twenty by using a combination of early-abort detection and a motion-based prefilter. While this algorithm performed relatively poorly on a standard test set of static face images, this measurement was a poor indicator of how the algorithm would perform on live video streams. By utilizing a pair of learned sensorimotor mappings, our system was capable of saccading to faces and extracting high resolution images of the eye on 94% of trials. However, even this statistic was misleading, since the behavior of the overall system eventually corrected for trials where the first saccade missed the target. To further evaluate behaving systems in complex environments, more refined observation techniques are necessary.

Acknowledgments

The author receives support from a National Defense Science and Engineering Graduate Fellowship. Support for this project is provided in part by an ONR/ARPA Vision MURI Grant (No. N00014-95-1-0600). The author wishes to thank Rod Brooks, Pawan Sinha, and the members of the Cog group for their continued support.

References

Ballard, D. 1989. Behavioral constraints on animate vision. *Image and Vision Computing* 7:1:3–9.

Baluja, S., and Pomerleau, D. 1994. Non-intrusive gaze tracking using artificial neural networks. Technical Report CMU-CS-94-102, Carnegie Mellon University.

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

Brooks, R., and Stein, L. A. 1994. Building brains for bodies. *Autonomous Robots* 1:1:7–25.

Brooks, R. A.; Ferrell, C.; Irie, R.; Kemp, C. C.; Marjanovic, M.; Scassellati, B.; and Williamson, M. 1998. Alternative essences of intelligence. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence (AAAI-98)*. AAAI Press.

Brooks, R. A. 1991. Intelligence without reason. In *Proceedings of the 1991 International Joint Conference on Artificial Intelligence*, 569–595.

Burghardt, G. 1990. Cognitive ethology and critical anthropomorphism: A snake with two heads and hog-nosed snakes that play dead. In Ristau, C., ed., *Cognitive Ethology: The Minds of Other Animals*. Erlbaum.

Coombs, D. J. 1992. Real-time gaze holding in binocular robot vision. Technical Report TR415, U. Rochester.

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

Graf, H. P.; Chen, T.; Petajan, E.; and Cosatto, E. 1996. Locating faces and facial parts. Technical Report TR-96.4.1, AT&T Bell Laboratories.

Heinzmann, J., and Zelinsky, A. 1997. Robust real-time face tracking and gesture recognition. In *Proceedings of IJCAI-97*, volume 2, 1525–1530.

Kuniyoshi, Y.; Kita, N.; Sugimoto, K.; Nakamura, S.; and Suehiro, T. 1995. A foveated wide angle lens for active vision. In *Proc. IEEE Int. Conf. Robotics and Automation*.

Marjanović, M.; Scassellati, B.; and Williamson, M. 1996. Self-taught visually-guided pointing for a humanoid robot. In *From Animals to Animats 4: Proceedings of the Fourth International Conference on Simulation of Adaptive Behavior (SAB-96)*, 35–44. Bradford Books.

Maurer, T., and von der Malsburg, C. 1996. Tracking and learning graphs and pose on image sequences of faces. In *Proc. 2nd Int. Conf. on Automatic Face- and Gesture-Recognition*, 176–181. IEEE Press.

Nummenmaa, T. 1964. *The Language of the Face*, volume 9 of *University of Jyvaskyla Studies in Education, Psychology and Social Research*. Reported in Baron-Cohen (1995).

Povinelli, D. J., and Preuss, T. M. 1995. Theory of mind: evolutionary history of a cognitive specialization. *Trends in Neuroscience* 18(9):418–424.

Ristau, C. 1991. Attention, purposes, and deception in birds. In Whiten, A., ed., *Natural Theories of Mind*. Blackwell.

Rowley, H.; Baluja, S.; and Kanade, T. 1995. Human face detection in visual scenes. Technical Report CMU-CS-95-158, Carnegie Mellon University.

Scaife, M., and Bruner, J. 1975. The capacity for joint visual attention in the infant. *Nature* 253:265–266.

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 binocular, foveated active vision system. Technical Report 1628, MIT Artificial Intelligence Lab Memo.

Sharkey, P. M.; Murray, D. W.; Vandevelde, S.; Reid, I. D.; and McLauchlan, P. F. 1993. A modular head/eye platform for real-time reactive vision. *Mechatronics Journal* 3(4):517–535.

Sinha, P. 1994. Object recognition via image invariants: A case study. *Investigative Ophthalmology and Visual Science* 35:1735–1740.

Sinha, P. 1996. *Perceiving and recognizing three-dimensional forms*. Ph.D. Dissertation, Massachusetts Institute of Technology.

Sinha, P. 1997. Personal communication. August, 1997.

Sung, K.-K., and Poggio, T. 1994. Example-based learning for view-based human face detection. Technical Report 1521, MIT Artificial Intelligence Lab Memo.

Terzopoulous, D., and Waters, K. 1991. Techniques for realistic facial modeling and animation. In Magnenat-Thalmann, M., and Thalmann, D., eds., *Computer Animation '91*. Springer-Verlag.

Thayer, S. 1977. Children's detection of on-face and off-face gazes. *Developmental Psychology* 13:673–674.

Tsotsos, J. K. 1988. A "complexity level" analysis of vision. *International Journal of Computer Vision* 1(4).

Turk, M., and Pentland, A. 1991. Eigenfaces for recognition. *Journal of Cognitive Neuroscience* 3(1).

van der Spiegel, J.; Kreider, G.; Claeys, C.; Debusschere, I.; Sandini, G.; Dario, P.; Fantini, F.; Belluti, P.; and Soncini, G. 1989. A foveated retina-like sensor using ccd technology. In Mead, C., and Ismail, M., eds., *Analog VLSI implementation of neural systems*. Kluwer Academic Publishers. pp. 189–212.

Wood, D.; Bruner, J. S.; and Ross, G. 1976. The role of tutoring in problem-solving. *Journal of Child Psychology and Psychiatry* 17:89–100.