Curiously Adaptive Growing Intelligent Neural Gas:
A Hybrid Approach to Category-Based Intrinsic Motivation

Maria D. Kelly, Rachel M. Lee, & Ashley M. Oudenne
Department of Computer Science
Swarthmore College, PA

May 11, 2009

Abstract

Intrinsic motivation, that is, internal rewards that reinforce certain behaviors in organisms, is an integral part of human development. This idea can be translated to the field of developmental robotics as a way to implement autonomous learning. This paper introduces the Curiously Adaptive Growing Intelligent Neural Gas (CAGING) algorithm, a hybrid system that builds on previous implementations to create a more robust, adaptable model. The CAGING algorithm categorizes the environment and utilizes these categories to make predictions about the robot's sensory data. The progress that can be made by exploring a particular category acts as intrinsic motivation for the robot. The model drives the robot to maximize its learning progress by attempting to learn about novel yet predictable situations. The experiment presented in this paper, "AIBO Television," demonstrates the effectiveness of the CAGING algorithm. In this experiment, a physical robot is placed in front of a computer monitor that displays a graphics program with three circles, each moving in a distinct way. Experimental results show that the robot focuses on the circles whose trajectories can be predicted and ignores circles with unlearnable movements. The system, while more autonomous and robust than other implementations, still suffers from some of the same problems with real world sensory data that previous systems exhibit. We conclude that the CAGING algorithm is an effective combination of existing intrinsic motivation systems.

1 Introduction

Intrinsic motivation is a biological term for the observation that "some activities provide their own inherent reward"[4, p. 627]. For these activities, external rewards such as food are not necessary for motivation. Rather, something internal sparks organisms to pursue these activities. For example, biologists believe that infants are intrinsically motivated to learn to communicate because it interests them somehow. This interest—a kind of curiosity—is an internal drive. If curiosity can motivate a child, why not a robot?

Implementing curiosity in a robot, however, poses its own challenges. Determining the exact mechanism behind curiosity remains an open problem. There are a variety of ways to represent curiosity, but most roboticists consider it to be a measure of the novelty of a situation. If a situation is new or unpredictable, the robot is internally driven to focus its attention on it. As the novelty of the situation wanes, the robot loses interest in it and seeks out something new. The benefit of developing a curious robot is autonomy: the robot, without the guidance of a programmer, strives to learn about its environment. This autonomy enables the robot to adapt to new situations and to learn new skills.

Our current work is motivated by a need for a more robust and adaptive curiosity-based learning system than those currently in existence. A current implementation of intrinsic motivation, Intelligent Adaptive Curiosity [7], detailed in the next section, lays the groundwork for further progress in using an intrinsic motivation system to model human development. However, limitations inherent in the system prevent its application to problems that involve richer sensory data. The system detailed in this paper combines current topological learning methods with an intrinsic motivation system. The resulting algorithm, Curiously

Adaptive Growing Intelligent Neural Gas (CAGING), provides a more sophisticated method for curiosity-based learning.

In the next section, we describe related work. In Section 3, we outline the CAGING algorithm in detail. The current experiment, "Aibo Television," is described in Section 4, including how CAGING was implemented on a physical robot. In Sections 5 and 6 we discuss the results of the AIBO Television experiment. In Section 7 we explore future research directions and extensions of the current experiment. Finally, we summarize our findings and conclude in Section 8.

2 Related Work

2.1 Intelligent Adaptive Curiosity

Intelligent Adaptive Curiosity, or IAC, is a system that uses intrinsic motivation to push a robot toward novel situations in which it can maximize its learning progress [7]. IAC is an unsupervised and incremental system able to adapt to new environments. The system begins considering all situations as belonging to one group, or region. As the robot develops, it splits this region into progressively smaller, more specialized regions. The robot explores the regions that will allow it to gain the most learning progress.

On each time step, the system takes in a vector of the current sensory information and generates a list of potential actions (or a subset of potential actions if the list is infinite). For each potential motor actions, the system creates a sensorimotor vector, a combination of the current sensory data and the potential motor action. Each sensorimotor vector is matched to one of the regions, and the region which gives the highest potential learning progress is selected (except with some probability of a random action). The sensorimotor vector corresponding to the chosen action is added to the winning region as an exemplar of the region [7].

The learning progress in a region is determined by computing the robot's error in predicting its sensorimotor flow [6]. At each time step, the robot feeds its current sensor and next motor values into the region's 'expert,' which can be a neural network, that is trained on all of the exemplars from the region. The expert predicts the sensor values for the next time step. The difference between the actual sensor values and the predicted sensor values is stored in a list of past errors. The list is smoothed and a final error is computed, and the inverse of this error is the learning progress for that time step.

IAC has been used successfully in highly constrained environments. In the Playground Experiment [7], in which a Sony AIBO say on a baby playmat, visual bit tags were placed on objects that the robot could bite, bash, and observe. The robot used IAC to learn how to interact with the tagged objects (i.e. bash at them, bite at them) and to predict what it would sense when it interacted with them. In another experiment, the same researchers added a communicating Sony AIBO to the playmat so that the developing robot could learn to communicate with it [6]. The communicating robot mimicked the sounds of the developing robot only when the developing robot was looking at it. The developing robot used IAC to learn to predict what it would sense when it made a noise while looking at the communicating robot.

While both of these experiments do show evidence of curiosity-based learning, they are also highly controlled. They rely on bit tags that allow the robot to identify objects of interest, which is not very applicable to most real-world robotic tasks. Additionally, the IAC regions are split after the region accumulates a certain number of exemplars determined by the programmer. This limits the autonomy of the system.

2.2 Growing Neural Gas

A Growing Neural Gas (GNG) is an incremental network model that learns topological relations in a set of data [5]. Like IAC, a GNG learns without supervision and is incremental. A GNG node contains a model vector of data from a sample space. The GNG begins with two nodes randomly positioned in the sample space connected by an edge. On each time step, a new input signal is generated and a new node is created. The squared distance between the nearest unit to the new node and the new node itself is the error, and this error gets added to a variable storing the error over all the nodes in the graph. The age of the edges emanating from the nearest node are incremented, and the nodes connected to the nearest node all move closer to the new node. Old edges are removed, along with nodes with no edges. In this way, the graph of nodes grows to fill the sample space.

One variation of a traditional GNG is an Equilibrium-GNG, which only adds a new node if the error over all the nodes is above a certain threshold [8]. This ensures that the Equilibrium-GNG will grow until it reaches an equilibrium between the decay and accumulation of error. If the error increases past the threshold, new nodes are once again added until equilibrium is re-established.

GNGs are often used in maze-navigation tasks. Jefferson Provost et al. used an Equilibrium-GNG to create a map of distinctive states (vectors) inside a simulated maze [8]. This system, called SODA (Self-Organizing Distinctive-State Abstraction), combined an Equilibrium-GNG with a hill-climbing algorithm to navigate a maze. A simulated robot explored the T-shaped maze environment and used its laser range-finder sensors to gather data about the environment. This data was put into a vector and added to the Equilibrium-GNG. After the robot explored the environment and the GNG was created, SODA mapped these distinctive states onto higher-level actions that took the robot from one distinctive state to the next, allowing it to successfully complete the maze. However, the maze environment was tailored specifically so that the two sides of the top of the T-shaped maze were visually unique, helping the GNG to characterize them as different distinctive states. Additionally, the environment was very stable, with little real-world noise. This suggests that GNG, by itself, may not be robust enough to handle real-world mobile robotic navigation and categorization.

2.3 Hybrid GNG Systems

The value of a GNG is that it can be combined with other systems, as in the SODA algorithm, to create a more robust and autonomous architecture. These hybrid systems make use of the GNG's incremental classification abilities to increase the autonomy of mobile robots. George von Wichert used a GNG to teach a robot to navigate in a normal office environment [9]. The robot, named ALEF, used ultra-sonic sensors and a camera to record 760 images of the office environment taken during its training. These images were fed into a GNG so that optically similar images were grouped together, giving the robot an internal representation of the office environment. ALEF used this map to navigate the environment, recognizing the difference between open areas and a shelf, a desk, or a narrow hallway.

In a related experiment, researchers Baldassarri, Puliti, Montesanto, and Tascini used a GNG to enable a simulated mobile agent to classify a basic indoor environment [1]. This environment consisted of three corridors, and the world was represented with few chromatic characteristics and dull colors. This helped simulate real-world indoor conditions. The robot was tasked with identifying the correct zone of the specific corridor, a relatively general task that can be easily adapted to different situations. As in the von Wichert experiment, camera images of the environment were fed into the GNG after being filtered. Two GNGs were used: one grouped images by their RGB content, and then the second, layered on top of the first, determined into which class of images a new image should be placed. An important difference between this system and the regular GNG system is that the *user* decides into how many regions the system divides the corridors. This makes the system less autonomous since the user dictates how the system categorizes the world.

Another example of a hybrid system is the GNG-Soft network developed by Carlevarino et al. [3]. The system combined the GNG's effective node distribution method for representing the input space of a problem with the SoftMax function's approximation properties. Instead of adding new nodes after a certain number of time steps, as in the original GNG implementation, or after an error threshold is crossed, as in the Equilibrium-GNG implementation, the GNG-Soft network only added a node if the global error was above a certain threshold *and* the derivative of the error was approximately zero. This ensured that the error itself was not decreasing, which would indicate that the graph was still adjusting to the addition of the last node. The GNG-Soft network was tested on a robotic head that featured two independently-moving cameras. The robot's goal was to map image-based targets onto motor commands. With this node-addition criteria, the robot successfully created the mapping in spite of moderately noisy data. The combination of the GNG and SoftMax systems led to a more robust and adaptable hybrid system.

3 The CAGING Algorithm

With this previous work in mind, we introduce a new algorithm for implementing category-based intrinsic motivation that capitalizes on the strengths of the two algorithms detailed in the previous section—Intelligent Adaptive Curiosity and Growing Neural Gas. The system presented in this paper, the Curiously Adaptive Growing Intelligent Neural Gas (CAGING) algorithm, is a hybrid of the two systems. This algorithm consists of two phases: the first phase utilizes a GNG as a front-end mechanism which allows the robot to construct a categorical representation of the environment, and the second executes a modified version of IAC, using the resulting categories of the GNG mapped onto modified IAC regions.

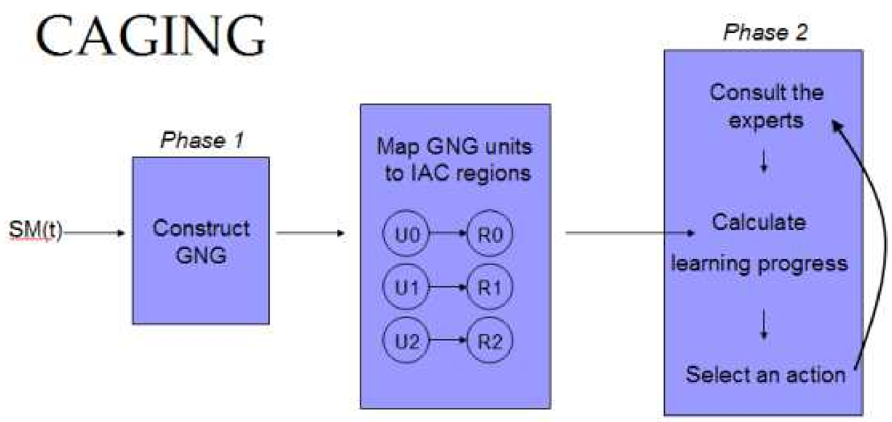

Figure 1: Flow chart depicting the two phases of the CAGING algorithm. In Phase 1, sensorimotor data is incorporated into a GNG. Then, in Phase 2, IAC is executed using regions that correspond to GNG units.

3.1 Phase 1: Develop a Growing Neural Gas

The first phase of the CAGING algorithm implements an Equilibrium-GNG to categorize the environment. In this phase, the robot randomly selects which action to execute, exploring the environment and accumulating sensorimotor information into a Growing Neural Gas. The GNG, given an error threshold ϵ and a stability factor K, adds a new unit when the average error across all units in the GNG exceeds ϵ. Typically a number of nodes are added toward the beginning; if the environment is somewhat stable, the number of units should level off. Once the GNG has gone K steps without inserting a new unit, the GNG is considered stable and complete. The algorithm proceeds to Phase 2, and the GNG does not get modified after this point.

3.2 Phase 2: Execute IAC Using Modified Regions

In Phase 2 of the algorithm, each of the units from the GNG is matched with an IAC region that initially has an empty error list E and an expert, realized as a neural network. Each region also has an associated learning progress, which is defined to be the additive inverse of the difference between the average error rate over the last several time steps and the average error rate of the same number of errors but from a time window several steps before the current step t. On a given step of Phase 2, the robot must select an action to execute on the next time step. To decide which action to take, the robot first makes a list of the possible actions $\boldsymbol{M}$ (if the list is infinite, the list is composed of a sample of the candidate actions).

For each candidate action M(t+1) from $\boldsymbol{M}$, the sensory information from the current step S(t) is combined with M(t+1) to form the sensorimotor vector SM. This vector is then matched to a unit from the GNG developed in Phase 1 by computing the Euclidian distances from SM to each model vector in the GNG and selecting the minimal distance. The candidate region R for a particular SM is the region associated with this matching GNG unit. The learning progress of each candidate region is computed and the action for which the robot expects the maximal learning progress is chosen (except when a random action is chosen with some probability ρ). Once the action is chosen, a prediction is made for the sensors on the next time step by consulting the expert of the winning region. Then, the action is executed. The error of the prediction is computed and added to the errors list and the expert of the winning region is trained only on the most recent exemplars.

4 The AIBO Television Experiment

The current experimental setup is called “AIBO Television” and involves a physical robot, the Sony AIBO, which sits on a charging station in front of a computer monitor (Figure 2). The monitor runs a simple graphics program consisting of red circles that are either stationary or moving. The robot sits under a makeshift tent to prevent glare from interfering with its camera image. We expected the robot to focus on circles with more predictable trajectories and to ignore those circles with unpredictable, random motion. Specifically, we expected that the robot would first focus and learn to predict the static circle because it never moves and is easily learnable, that it would then shift its attention to the horizontal circle because it moves in a predictable pattern but is more difficult to learn, and that it would generally ignore the randomly moving circle because its trajectory is unlearnable. The experiments were run using the PyRobot simulator [2].

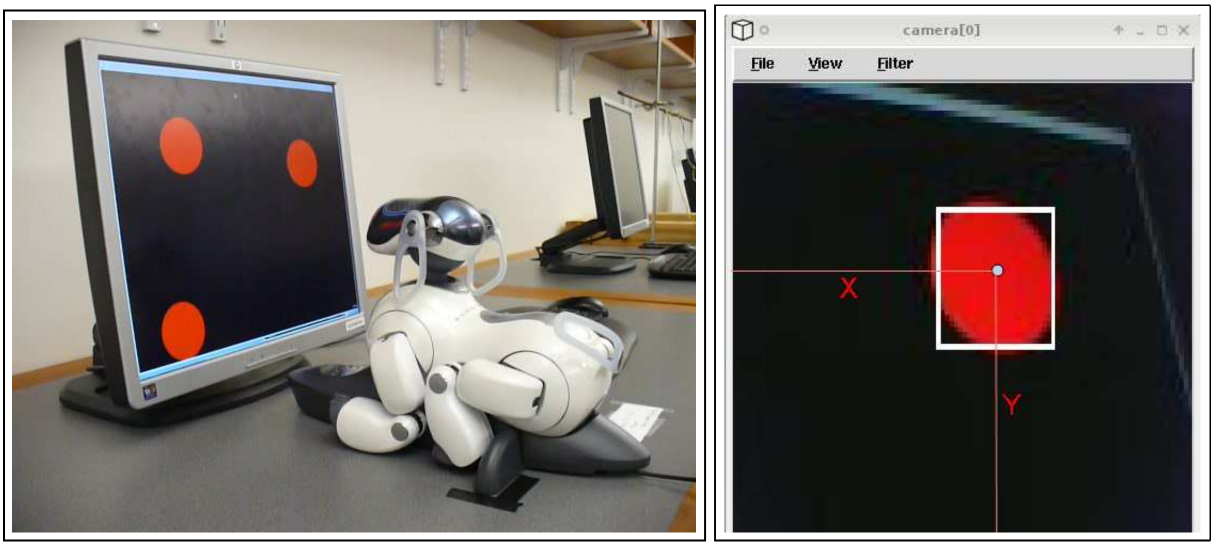

Figure 2: (Left) The “AIBO Television” experimental setup. The AIBO sits on a charger in front of a monitor that displays a graphics program. (Right) A sample AIBO camera image. The camera filter places a bounding box around red areas in the image. The sensory input for the robot is the x- and y- coordinates of the center of this bounding box.

4.1 Experimental Setup

The AIBO is equipped with a camera that takes in images of the monitor. These images are preprocessed using the PyRobot software and areas that match a certain “red” RGB range of values are determined and a bounding box is placed around sections of red [2]. The monitor displays three distinct red circles, one in each of three quadrants or focus zones of the screen: the first is a stationary circle, the second is a circle which moves horizontally across the zone in a predictable way, and the third is a circle which moves randomly to a new location in the zone at each time step (see Figure 2). The robot's sensory information S(t) consists of two inputs, the x- and y-coordinates of the center of the bounding box in the camera image (see Figure 2). The AIBO's potential motor actions M(t+1) are limited to three basic actions, each corresponding to a movement of the head (through setting the values of the pan (p), tilt (t) and roll (r) of the AIBO's head) that results in the AIBO looking at a single one of the three focus zones (the other two focus zones and their respective circles are out of view).

4.2 CAGING Implemented

The robot is equipped with the CAGING system and is initially allowed to look randomly at any of the three quadrants in order to develop a GNG representation of the camera images. The model vectors of the GNG are the sensorimotor vectors SM(t) = (x, y, p, t, r). For this experiment, the error threshold of the GNG $\epsilon = 0.25$ and the stability factor K = 300 steps. Once the GNG stabilizes, the IAC phase of the CAGING algorithm begins. The mapping that the AIBO has to learn in this stage of the experiment is:

$$\mathrm{S}M(\mathrm{t}) = (\mathrm{x},\ \mathrm{y},\ p,\ t,\ r) \mapsto \mathrm{S}(\mathrm{t}+1) = (\mathrm{x},\ \mathrm{y})$$

Using the CAGING algorithm described in the previous section, the robot chooses its actions according to which of the three possible actions corresponds to the region with the maximal potential learning progress or, with probability $\rho = 0.10$, chooses a random action to execute. The learning progress is computed as the inverse of the difference between the average of the last 15 prediction errors and the value of this average 10 steps ago. Once an action has been chosen, a prediction is made for S(t+1) by consulting the expert of the winning region. Then, after the action has been executed, the error of the prediction is computed and added to the errors list. Finally, the expert of the winning region is trained on the five most recent exemplars.

5 Results

After running three trials of the AIBO Television experiment on the AIBO and three trials in simulation, we found that the robot was generally able to predict the location of the red circles with learnable movements. Our results indicate that the CAGING algorithm constitutes a working implementation of curiosity-based learning. However, the results differed somewhat from our original expectations for the experiment due to real-world conditions.

5.1 Real-World AIBO Television Experiment

Figure 3 illustrates a typical Equilibrium-GNG network. In less than 400 time steps, 10 units representing the paths of the three circles emerge. One unit represents the location of the static circle. Three units arranged in a row represent the horizontal movement in the focus zone of the upper right quadrant. A smattering of five nodes located toward the center of the quadrant represents the focus zone containing the random circle. In atypical runs of the Equilibrium-GNG creation phase, the system creates units that are ambiguous and can be attributed to more than one of the three focus zones. Such units, however, do not affect the performance of the GNG, because they are never matched with a sensorimotor vector during Phase 2.

As stated in previous sections, the system selects a focus zone containing either a static circle, a horizontally moving circle or a randomly moving circle. Figure 4 shows the average percentage of time that the system selects one of the three focus zones. The system selects the focus zone that contains the static circle for over half of the total running time of the trial. This behavior is not consistent with the results presented in [7].

In order to examine the extent to which noise affects the performance of CAGING, another three trials were run in simulation.

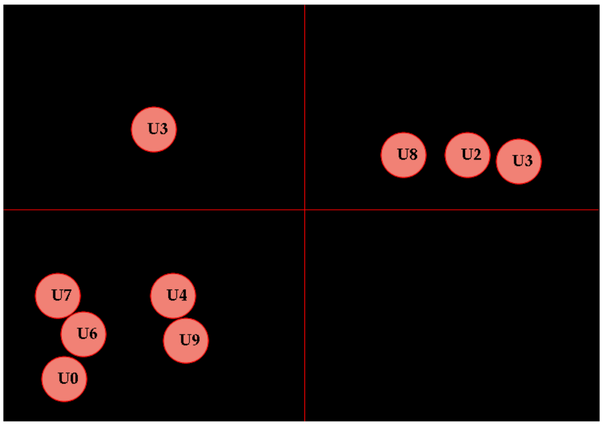

Figure 3: A characteristic Equilibrium-GNG developed by CAGING in the AIBO Television experiment. The top-left quadrant corresponds to the static focus zone, the top-right to the horizontal focus zone, and the bottom-left to the random focus zone.

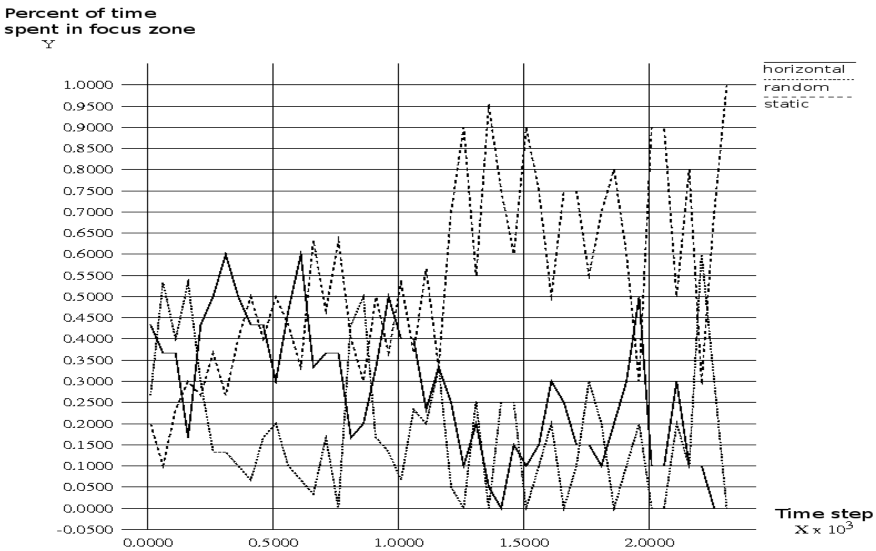

Figure 4: Average percentage of time steps spent in focus zones over three trials.

5.2 Simulated AIBO Television Experiment

Figure 5 shows the results of a typical simulated trial, with sensory inputs that included 15% simulated noise. These results are consistent with results from simulations conducted in [7]. The system rapidly differentiated between learnable and unpredictable situations. Within 300 time steps, all learning progresses have reached zero and the behavior of the system becomes random (see Figure 6).

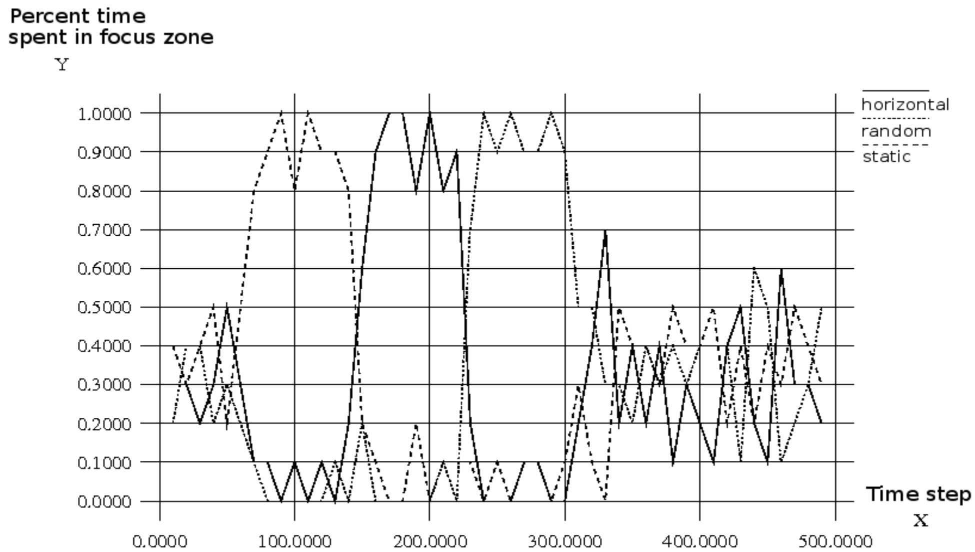

Figure 5: Typical run of a simulated trial

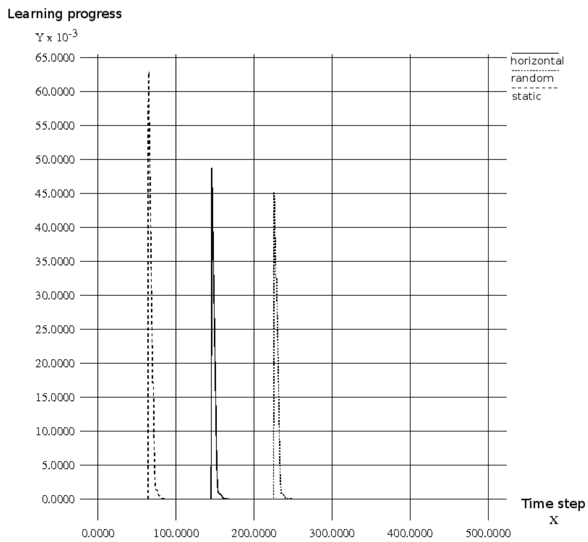

Figure 6: Typical plot of learning progress of a simulated trial

5.3 Two-Choice Trial

Figure 7 shows the results of a trial in which the option of viewing the static circle has been removed. Here, CAGING, for a majority of the time, chooses to view the horizontally moving circle. This result is consistent with our previously established expectations that the system will select the circle that moves in the most predictable pattern.

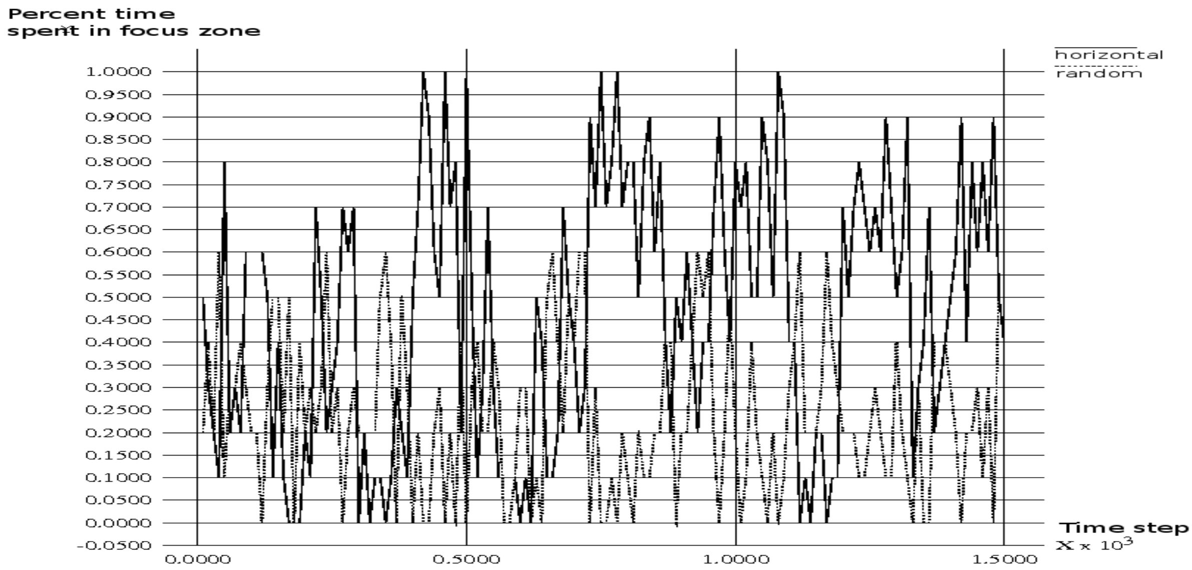

Figure 7: Percentage of time steps spent focusing on the horizontally and the randomly moving circles in the Two-Choice trial

6 Discussion

6.1 Equilibrium-GNG

As expected, the Equilibrium-GNG distributes units sensibly. The Equilibrium-GNG, as seen in Figure 3, represents the sensorimotor space in a way that enables more intuitive interpretation. The resulting categorization allows the system to concentrate its learning efforts on more relevant regions of the sensorimotor space. This behavior contrasts the somewhat arbitrary and increasingly fine-tuned splitting of the entire sensorimotor space used by IAC.

6.2 IAC Phase

We predicted that the system would select the focus zone that contains the horizontally moving circle—the circle that shows behavior that is challenging to predict, but not impossible. Our results suggest that CAGING *does* fixate on the most predictable yet challenging focus zone, but that this zone is in reality the focus zone containing the static circle.

An examination of Figure 8, a plot of the learning progress calculated for the three focus zones, clarifies the details of CAGING's selection process. The learning progress calculated from observing the focus zone that contains the horizontally moving circle and the focus zone that contains the randomly moving circle are much lower than that of the static circle, even though at times the focus zone with the static circle is also negative. These negative learning progresses indicate that the amount of error the system makes in predicting the position of the circle increased over time. While such data is to be expected from the prediction of the randomly moving circle's location, the extent to which the learning progress values of the static and horizontally moving circles are negative may explained by two possible occurrences.

The first possible cause of these results is that the pattern of the circles' movement cannot be predicted by the expert neural networks' current structure. Currently, the experts are implemented with a neural network that contains no hidden nodes. However, the simpleness of the circles' movement patterns renders

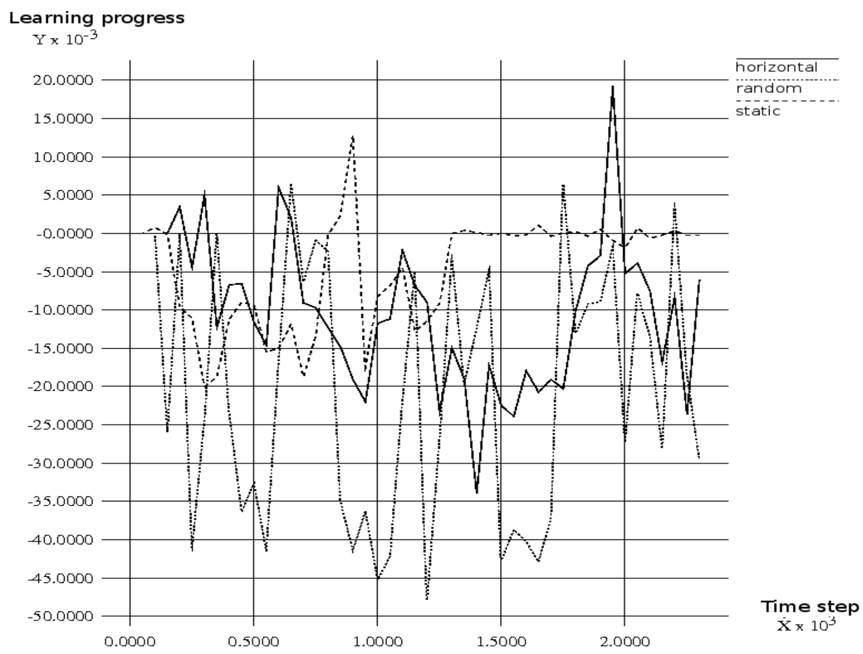

Figure 8: Average learning progress over three trials

this option unlikely. The second possible explanation is that noise interferes significantly with the system's ability to make accurate predictions. In this experiment, several factors may have hindered the system's prediction ability. In spite of efforts to minimize glare on the screen, it continued to impact the camera's ability to place a bounding box around the entire circle that was in view. The glare on the screen often changed the perceived RGB value such that it no longer fit into the range that the bounding box considered to be "red." As a result, the size of the bounding box around the circle changed, altering the perceived center of the circle. Another real-world problem is the robot's inability to place its head in exactly the same position each time. Even slight movements of the head resulted in significantly different x- and y-coordinates of the bounding box. We believe that this contributed greatly to the difficulty in predicting the horizontally moving circle during the real-world trials.

The results of trials run in simulation (Figure 5) reinforce the idea that noise plays a significant role in CAGING's overall performance. The spikes in average learning progress over these simulated trials (Figure 6) demonstrates that CAGING is able to make rapid progress as long as noisy fluctuations in sensor values are not too unpredictable.

In contrast to these simulated trials, the system which takes in input from the AIBO's camera selects the focus zone containing the static circle for over half of the system's total running time. Such behavior suggests that, although CAGING addresses IAC's issues with region splitting and memory efficiency, it continues to suffer from some of IAC's limitations. Because noise distorts the appearance of the circle in the camera image, the fluctuations in the area and location of the bounding box with respect to the circle cause it to appear to move. Thus, the system concentrates its prediction efforts on the static circle, that is, the circle that seems to move the least out of the three.

7 Future Work

There are many possible future directions for research with CAGING. The biggest limitation of the system as it stands is an inability to cope with large amounts of noise that often occur in a real-world trial. In the AIBO TV experiment, the robot encounters difficulties in predicting the movements of the red horizontally moving circle due to image distortion and motor noise. One possible way to address this is to use a different filtering method so that the bounding box around the red circle does not fluctuate as much. Using a different robot with higher-quality camera equipment might also improve the robot's prediction ability.

To further expand upon the current experiment, the graphics program could be modified to include more circles (or different shapes) moving in different ways. For example, shapes could expand and contract, move in non-linear patterns, or change color. It would be interesting to discover which of the movements are easily predictable and which are more difficult. After more shapes are added, the robot's degrees of freedom could be increased so that it would learn where to place its head on its own, instead of using pre-defined pan, tilt, and roll values like those in the current implementation.

Additionally, the CAGING system could be tested using a variety of robots in other situations. The experiment detailed in this paper discusses CAGING in visual prediction situations, but the system may be used in robotic tasks in general. CAGING could be applied to navigation or communication-oriented tasks to test its applicability to different situations.

8 Summary and Conclusions

In this paper we have presented a hybrid approach for implementing intrinsic motivation using learning progress as the basis for modeling curiosity. CAGING first categorizes the environment using an Equilibrium-GNG, and then implements a modified version of IAC, which uses these categories to focus on learning novel and predictable situations. In the AIBO Television experiment, an AIBO employing the CAGING algorithm learned to predict the location of static and horizontally moving circles seen on a monitor. Due to noise from the real world environment, the static circle became the most learnable of the three; as a result, it was selected most often. When the robot's actions were limited to looking either at the horizontal or the randomly moving circle in the Two-Choice trial, the robot focused on the horizontally moving circle for a significant portion of the time. Although CAGING did not select the horizontally moving circle while it had the option of viewing the static circle, the results from the Two-Choice trial indicate that the CAGING algorithm is in fact able to distinguish between learnable and unlearnable situations.

In conclusion, we believe that the CAGING algorithm serves as a more robust, adaptive system for modeling intrinsic motivation. The system allows the robot to be more autonomous, as it is able to categorize its environment and employ a kind of curiosity to motivate its learning. Intrinsic motivation is important to the field of developmental robotics because it closely models mental development in biological organisms. A robot that develops autonomously does not require the direction of a user; instead, it can take in information about its environment, develop its own categorization of the world, and select appropriate actions to execute. CAGING helps pave the way for further progress in the field.

8.1 Acknowledgments

We would like to thank Professor Lisa Meeden for her guidance and inspiration. Also, we would like to thank Honey (aka 10.0.0.4) for being such a good, patient dog.

References

[1] P. Baldassarri, P. Puliti, A. Montesanto, and G. Tascini. Visual self-localisation using automatic topology construction. In *ICIAP '03: Proceedings of the 12th International Conference on Image Analysis and Processing*, page 368. IEEE Computer Society, 2003.

[2] Douglas S. Blank, Deepak Kumar, Lisa Meeden, and Holly A. Yanco. Pyro: A python-based versatile programming environment for teaching robotics. *ACM Journal of Educational Resources in Computing*, 36(2):1–15, 2004.

[3] A. Carlevarino, R. Martinotti, G. Metta, and G. Sandini. An incremental growing neural network and its application to robot control. In *IJCNN '00: Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks (IJCNN'00)-Volume 5*, page 5323. IEEE Computer Society, 2000.

[4] Edward L. Deci, Richard Koestner, and Richard M. Ryan. A meta-analytic review of experiments examining the effects of extrinsic rewards on intrinsic motivation. *Psychological Bulletin*, 125(6):627–668, 1999.

[5] Bernd Fritzke. A growing neural gas network learns topologies. *Advances in Neural Information Processing Systems 7*, 1995.

[6] Pierre-Yves Oudeyer and Frederic Kaplan. Discovering communication. *Connection Science*, 18(2):189–206, 2006.

[7] Pierre-Yves Oudeyer, Frederic Kaplan, and Verena V. Hafner. Intrinsic motivation systems for autonomous mental development. *IEEE Transactions on Evolutionary Computation*, 11:265–286, 2007.

[8] Jefferson Provost, Benjamin Kuipers, and Risto Miikkulainen. Developing navigation behavior through self-organizing distinctive state abstraction. *Connection Science*, 18(2), 2006.

[9] Georg von Wichert. Selforganizing visual perception for mobile robotic navigation. In *Proceedings of the 1st Euromicro Workshop on Advanced Mobile Robots*, pages 194–200. Society Press, 1996.