A Biologically Inspired Spatial Computer that Learns to *See* and *Act*

Paul Robertson, Robert Laddaga

BBN Technologies, Inc., Cambridge, MA 02138, USA

Vision and motor control are usually studied as separate phenomenon. They perform very different functions, they are performed by different regions of the brain, and one is perception while the other is actuation. The two structures did, however, co-evolve. While they are different structures they work together in reasoning about and manipulating the outside world. Both structures have some similar attributes. For example both the motor cortex and the visual cortex are laid out in a manner that preserves topological adjacency and the hippocampus, where positional awareness is represented also represents places in the world through a topological map. In all case the layout of the areas suggests algorithms that depend upon propagation through a kind of spatial computer in order to solve navigational tasks that combine perception, actuation, and spatial awareness. In this paper we take the position that it makes sense to study the computational aspects of learning to perform such tasks together rather than as separate disciplines and that by observing the similarities of the layouts of the associated areas we can gain some insight into a general learning engine that utilizes spatial computing principles in order to achieve complex behaviors in a complex world that can only be modeled imprecisely. This paper describes such an approach embedded within simple robotic devices.

Index Terms— **Biomorphic Computing, Computer Vision, Memory-Based Learning, Spatial Computing.**

I. Introduction

Much is known about the biological neuronal structures that constitute the human vision system [1] as well as that of other animals such as frogs[2], cats, and macaque monkey[3]. Schematic diagrams of brain function are well documented for studied species [3, 4] and experimentation has yielded a level of understanding of what is computed by the blocks in these diagrams. Some researchers are in the process of building simulations of these building blocks of visual processing. These schematics are often presented as if they were circuit diagrams resulting from millions of years of evolution—as rigid in their structure and as common among individuals as are kidneys, hands and feet. Computer vision researchers strive to find the right operators and the right representations necessary to reproduce the capabilities of the human visual system. In this paper we present a different perspective on the human visual system that suggests placing a greater emphasis on structures that can learn representations, rather than on designing the representations.

We are developing an architecture [5, 6] for learning to see and act that is based on engineered emergence. In our approach the solution is engineered by connecting learning components into a spatial computer. Perception and actuation procedures are learned within the spatial computer based upon the location where information converges within the spatial computer and the perceptual history of the local components within the computer. While this paper covers early implementations of these ideas encouraging results have already been obtained.

The memory architecture described implements a layered spatial computer where the key operations on the computer involve (1) looking up near matches that are reported in order of their Euclidean distance in spatial compute space; and (2) searching memorized sequences in a manner in which the results are found in order of the Euclidean distance within the spatial computer spanned by the strings. Early processing stages lay out memory elements in space that mirrors the real world. Abstract layers are formed on top of these layers and the positions of learned abstractions are chosen so as to minimize the distances to all of the lower level contributors. Search distance through the spatial layout of the memory elements crucially drives the performance of the system.

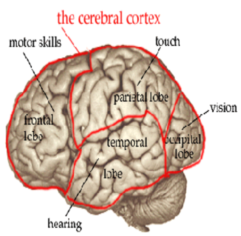
The paper is divided into three sections: First we provide a very brief overview of what is known about biological vision systems to the extent that it is relevant to this discussion; Next we outline a view of the human (and other animal) vision systems as spatial computers; Finally we discuss early work with an artificial spatial computer's attempt to learn to see and act.

II. Biological Vision systems

The major components of the human vision system consist of the eyes, the lateral geniculate nucleus (LGN), and the visual cortex. The newest part of the system resides in the cerebral cortex—the visual cortex—and occupies approximately one third of our brain mass depending upon where one draws the line between visual processing and higher level reasoning about vision.

The visual cortex is at the back of the brain and is divided into left and right parts. Projections from both eyes reach both left and right parts of the visual cortex and communication between the left and right sides can occur through the corpus callosum. Over the years experiments with animals and observation of humans, particularly those suffering brain damage resulting from stroke or otherwise induced lesions, have resulted in a significant understanding of the structure and functional nature of the visual cortex.

Digital Object Identifier inserted by IEEE



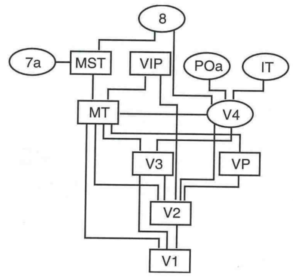


Figure 1 (left) shows the position of the visual cortex in relation to other functional areas in the human brain. Figure 2 (right) shows the, now famous, functional schematic of visual cortex due to Van Essen et. al. [3, 4].

It is beyond the scope of this paper to describe in detail the structure of the functional blocks of the visual cortex. What is important for the current discussion is that the structure of the visual cortex consists of functional blocks that receive projections from other functional blocks within the visual cortex and from outside the visual cortex (such as the LGN). These projections tend to be topographical. Projections from the retina map onto areas in V1 so as to retain topological adjacency. The mapping from the retina onto V1 has been demonstrated and the map is readily understood. The further up the chain one goes the less obvious the mapping becomes as adjacency is less closely tied to the retinal image and more related to semantic features.

III. Biological Spatial Computer for Vision

What is computed within the visual cortex is determined by its physical location within the structure. There is a right place to compute low-level attributes of the visual field based upon where the inputs that pertain to that field project onto the cortex. The basic wiring diagram of the visual cortex as well as the nature of projections between functional units and the density of neurons in those areas is likely determined by genetic encoding—resulting from a long period of evolution. The process implemented by the functional blocks, however, is quite likely more plastic and driven by experience.

There is some reason to believe that the functional blocks, rather than implementing hardwired and specific functions, instead provide a spatial learning engine that learns specific functions as a result of (1) the spatial position of the blocks within the biological spatial computer, (2) the inputs that are projected into that region, and (3) attributes of the physical world that the cortex is exposed to. Two results in particular support this view, one involves the deliberate limitation of visual experience during development and the other involves experiments that rewire the optic pathways. Since these findings are important to the premise of this paper we provide a very brief overview of the results below.

In 1970 Blakemore and Cooper[7] reported experimental results that showed that kittens that had lived their entire lives in a controlled environment in which only edges of a certain orientation were presented had visual cortices that did not have cells sensitive to edged orientations that they did not observe. These experiments support the notion that structure in the primary visual cortex is, at least to some extent, learned.

In other experiments, rewiring the optic pathways so that visual information is redirected to another brain area, not associated with vision, have shown that visual capabilities develop in the new area. While there is clearly some value to the large scale structure of visual cortex and the projections between the functional units, the function of the units themselves is largely driven by the information that they are exposed to by virtue of afferent projections.

The visual cortex can be thought of as a spatial computer where **what** is learned depends upon the environment to which it is exposed, and **where** it is learned depends upon the connectivity of the functional units.

IV. An Artificial Spatial Computer for Vision

To experiment with a spatial learning computer for vision we implemented a memory-based learning [8] component that could be replicated in a topological map such as found in projections into the primary visual cortex's V1. These components receive projections from a small circular region from a specific place in an image. When an image fragment falls upon the region the distribution of light intensities is recorded without preprocessing except for intensity and distribution normalization. When an image fragment closely matches one that has already been stored the responding memories are recalled via nearest neighbor lookup using k-d trees. After exposing the components to a collection of image fragments clusters of memories in the k-d trees develop that respond to the common features found in the images – namely no edges present and edges at different orientations. Without deletion the size of memory rapidly grows and the k-d trees develop dense populations of memories of frequently reoccurring edge orientations. When memories become excessively huge localized regions within the k-d trees are taken and replaced by a single entry at the center (mean) of the cluster and with a count indicating the number of memories that were originally there and a histogram representing the distribution of the replaced memories. Subsequent images that fall on the component use the count and histogram to simulate the response that would have resulted from a population of memories.

The system self-organizes its memory so that it resembles receptive cells in V1. This organization depends upon (a) very simple rules for distributing, spatially, parts of an image, (b) large memory for recording instances of patterns that fall upon the components, and (c) a strategy for collapsing multiple representations of similar memories.

By connecting up projective pathways in ways similar to Figure 2, we believe that a structure of components sensitive to color (V4), motion (MT), etc. could be *learned* from exposure to colored images containing moving objects. This latter experiment has not been performed yet partly because of the computation cost and partly because we were more interested in extending the learning paradigm to support intrinsic motivation [9, 10] by extending the system to include action as well as perception.

For the most part animals don't exist as passive creatures that observe the world. They exist as creatures that interact with their world and simultaneously learn to act in and perceive their environment. By allowing action as well as perception, the learning process can be enhanced by the ability to manipulate the world.

Experimental Desktop Robotic Platform

In a system that can "act and see" the results of actions impacts what is perceived. This correlation between action and perception must be learned by the spatial computer in which the action spaces as well as the perceptual spaces are laid out as topological maps within the spatial computer. We have implemented a very simple desktop robotic platform upon which to develop our early experiments with the spatial learning system.

Figure 3: Robotic bug with synthesized vision from an overhead camera and four actuation spaces corresponding to turn left, turn right, move forward, and move backward.

The platform consists of two robotic bugs pictured in Figure 3 that have a small number of control actions. They can move forward and backward and can turn left and right for some duration of time. The legs slip on the surface injecting a noise component into the actuation procedures forcing a tight perception/action control loop.

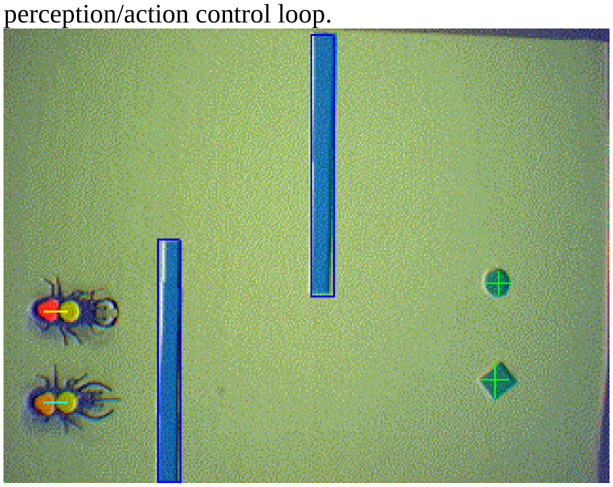


Figure 4: Experimental setup. Blue objects are barriers and green objects are targets.

An overhead camera maps the experimental setup by interpreting blue objects are barriers, green objects as targets, and identifying the position and orientation of the two bugs based upon the two colored dots on their backs. Figure 4 shows a simple experimental setup with two bugs in a starting position. The computer display of the images from the overhead camera is shown with interpretation overlaid on the images. The blue rectangles around the blue barriers indicates interpretation of the blue strips as barriers, the green '+' marks on the green patches indicate targets and the lines on the backs of the bugs indicates the position and orientation of the bugs. From the camera images and the interpretation of the objects in the scene the system synthesizes a bug's view of the world from the perspective of each bug that the bugs must learn to interpret. The bugs must learn to play actuation procedures that get them from their current position to the position where the targets are located. All computation is performed on a desktop computer that simulates the spatial computer and the bugs are controlled by a control interface pictured in Figure 5.

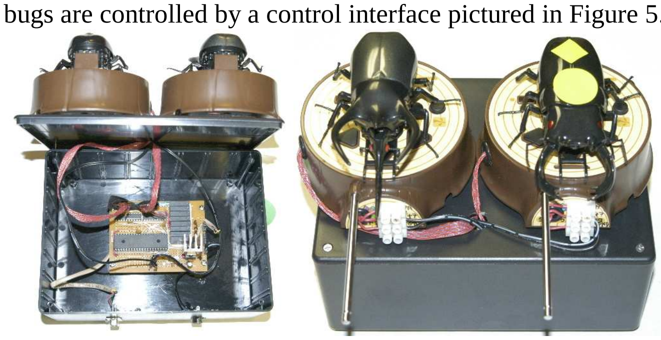


Figure 5: Control interface for the robotic bugs. The interface controls the handheld remote control devices that double as chargers.

The example of learning that we described above in which simple edge detectors were learned passively from observations in an unsupervised manner required no motivation.

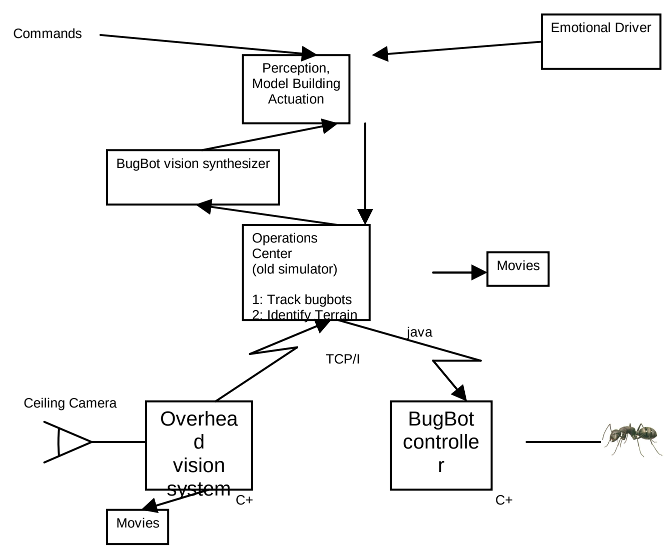


Figure 6: Schematic of the desktop robotic platform

Our extended experiment requires that we introduce motivation. Two forms of motivation are observed by the robotic bugs: extrinsic motivation – where we provide an explicit goal – and intrinsic motivation which is a built-in desire to experiment in poorly understood areas of the event space. From the beginning these two forms of motivation are in conflict. The intrinsic motivation pushes the robots to experiment while the extrinsic motivation pushes the robots to complete their task. In our initial experiments we used a crude hardwired resolution to this conflict. In upcoming experiments we are introducing a learning system of somatic markers [19] to mediate this conflict.

Extrinsic motivation takes the form of goal states that are given to the robotics bugs (that are implemented as separate entities). A goal state is given as the synthesized view that the robot will see when it has reached its target, in this case the view that the bug will see when it is on top of the target.

Cached Event-String Automata

In Sciences of the Artificial [17], Herb Simon introduced us to the idea that an ant walking through a complex environment may exhibit complex behavior but that the complex behavior is largely a result of the environment and that the ant may implement rather simple control laws. Rod Brooks [13] described a hierarchical architecture that would allow the implementation of simple control laws that interact with the environment to produce complex and robust behavior. Brooks argued against models by claiming that the best model of the world was the world itself. Richard Feynman [14] gave us an anecdote about an early, informal, experiment that he performed with ants in order to understand how they learned efficient routes between food and colony even though the process of discovering the food resulted from a random exploration process. The key involved the use of pheromones that would be left in the environment that the ants could follow to and from the food. Errors made by ants in following the correct path would sometimes lead to an ant taking a shortcut and accidentally getting back on the path. Such shorter paths would be reinforced by pheromone trails and so the path would gradually be optimized over time and sooner or later would look fairly optimal. Feynman's ants used the world as their model in the Brooks sense and learned paths by annotating the world using chemicals. It should be noted that much more is now known about the mechanisms used by ants in navigating their world [18].

By remembering the perceived world as sequences of perceptions and actions we can simulate the process of learning described by Feynman and by doing so learn to perform skills. By remembering everything, we can find what we need when we need it. While this kind of learning may seem inefficient it has many advantages. David Marr [15] noted:

> "... the problem is not totally intractable despite the huge sizes of all the relevant event spaces. The reason is that only a very small proportion of the possible events can ever actually occur, simply because of the length of time for which a brain lives. This means, first, that the memory can be quite coarse; and secondly, that if anything much happens twice, it is almost certain to be significant."

Our approach uses a memory based learning approach [8] in which memories of sensory inputs and actions are recorded for later use (in a cache). These memories are hierarchically organized and generalized. Skills are applied to new situations by finding relevant memories and replaying them.

The cache is populated both from action/event sequences resulting from goal directed behavior and self-motivation [9, 10] driven by a meta-strategy of exploration in poorly represented parts of the event space as a low priority process.

In the following discussion we introduce 'Cached Event String Automata' (CESA), describe algorithms that permit CESA to learn behavioral skills in real world environments, and discuss performance implications of the approach.

Sequences of sensor inputs such as a visual image, leg position feedback and haptics; and actions such as actuation commands are recorded. A goal is specified by presenting a solution in perceptual terms with relevant details of the goal state present. We refer to these sequences of perceptual and actuation events as event strings – or strings for short. Our approach stores event strings in a manner that permits rapid retrieval of close matches and new complex activities are formed by stitching together fragments of generalizations of prior strings.

String Stitching

String stitching looks up previously learned sequences that when joined together get from the current state to the final state. Any *action* taken by the robot represents a choice on the part of the robot to perform that action. Strings are therefore hinged at action points. We consider sequences to be Markovian in that it doesn't matter how a robot arrived at a given state.

Our learning approach depends upon reusing fragments of old skills in order to build new ones. The approach therefore depends upon there being pre-existing skills. An important question therefore is 'where do the initial skills come from?' There are a number of potential solutions to the problem of prior skills:

1. The initial sequences could be hand generated;
2. The robot could be tele-operated through a certain skill a number of times – each time recording the sequences of observations and actions; and
3. The robot could explore its world by itself.

(1) is probably not feasible; (2) is potentially very useful and is similar to the way we teach humans; and (3) provides a way for the robot to automatically adapt when the environment changes.

Our experiments to date have focused mostly on (3). Current experiments with sensory-motor learning with a humanoid robot are exploring (2).

Intrinsic Motivation driven by Surprise

Surprise-based learning is implemented as a low-priority process that is used to soak up free time resources such as when the robot is already in a goal state and is thus free to play. As mentioned earlier, a somatic marker based mechanism will be introduced later to allow experimentation to occur at more opportune moments while being extrinsically motivated if such experimentation doesn't interfere with the completion of it main goal. This approach will permit the robot to stop and smell the roses along the way, as it were, and to benefit from learning whenever a learning opportunity arises if time permits. The currently implemented system however operated as described below.

When in a goal state (i.e.: nothing to do) the robot randomly (weighted by interest) chooses to:

1. Do nothing.
2. Pick one of the available actions.

Existing strings predict an outcome (next element on the string). If the string base accurately predicts the outcome the interest level in the action is decreased. Otherwise the interest level (Surprise) is increased. When the robot encounters the opportunity to make an action for which it has a high interest it may decide to experiment with the action with increased probability proportional to the level of interest that has accumulated from prior encounters. As the robot experiments with actions of interest the cache fills up with strings that predict the outcomes of the actions and as they become more accurate the level of interest is decreased and the system gradually settles down to no exploration because of lack of surprise.

Whenever an action yields an unpredicted response the surprise is noted and later when the context is appropriate exploration will take place.

String Hinge Spreading Activation

A string is an instantiation of a path through a (Markov Decision Process) MDP. Actions represent decisions to follow a path through an MDP. A string can therefore be cut at any action (hinge) along the string. Take a string, select its hinges and find follow on strings from the hinges and recursively repeat this process until the goal is reached. This simple algorithm finds solutions in order of least number of splices. Solutions are collected until an *adequate* solution is found then the best solution found up to that point is invoked resulting in the robot performing an action. An adequate solution is one that meets certain constraints whose nature we will not elaborate in this paper. An import point, however, is that the solutions are found in order of string splicing complexity (least number of splices) and are not necessarily optimal in any other sense. By allowing a few solutions to be collected before choosing one there is the possibility of selecting a more optimal solution. It is not necessary to find the optimal solution because over time incrementally better solutions will be collected which will tend to improve to a local optima.

A CESA is a 5-tuple < ω_1, ω_g, A, Ω, $S_{A\omega}$ > where: $\omega_1 \in \Omega$ is the initial observation $\omega_g \in \Omega$ is the goal observation A is the set of actions, Ω is the set of observations, $S_{A\omega}$ is a set of cached strings whose elements $e_i \in A \cup \Omega$, Learning takes a CESA and produces a new CESA' that changes the $S_{A\omega}$ by adding, removing and or otherwise modifying the entries.

We have already mentioned that the strings stored in the CESA memory are Markovian in nature and that actions in a string represent decision points. Each recorded string represents a path taken through a Hidden Markov Model (HMM). If there were a sufficient number of them the probabilities of transitions at the decision points could be calculated. A CESA then evolves into a non-compact representation of an HMM. If the environment changes the structure of the HMM can change as new sequences are experienced. By not compactly representing the HMM we are able to adapt to a changing environment without additional mechanism.

A single string allows the activity represented by that string to be learned. Generalization is performed by performing nearest neighbor matching. When there is only one string the nearest neighbor will be that string but as more strings are cached the nearest neighbor match becomes more accurate and string splicing allows for learned fragments to be assembled into complex activities. As more strings are learned the probabilities at decision points are gradually estimated.

The approach described above resembles spreading activation [11] and motor procedures encoded in human Basal Ganglia [12]. Experiments to date have successfully demonstrated the ability for a simple insect-like robot to simultaneously learn motor procedures, image interpretation procedures, and a cognitive map so that the robots can avoid obstacles as seen through their vision inputs and get from a starting position to a target position through a maze after sufficient trials in the maze.

V. CONCLUSIONS AND FUTURE WORK

In a spatial computer the layout of the computing elements determines in some way the performance of the system. The system described in this paper draws heavily on this principle. Nearest neighbor search finds entry points into the learned structure, new memories are laid out in locations that will allow them to be found by nearest neighbor search, and paths through strings of recorded events and actions are searched for in order of Euclidean distance through the spatial computer in a manner that returns the solutions in order of increasing total distance. While the current implementation utilizes a simulation on a sequential computer that is decidedly not spatial, the architecture lends itself to straightforward and efficient implementation using simple hardware blocks, such as using an FPGA, or on other highly parallel platforms. Distance can be computed by propagating broadcasts that naturally return matches in order of distance.

The overwhelming majority of research into computer vision has focused on implementing visual operators and designing representations that are well suited to the image processing task. Sometimes these operators and representations are closely related to neurological findings and sometimes they are more closely related to understandings of the physics of the image formation process. While there is a lot of evidence that some of these operators and representations are similar to those found experimentally in animal visual systems it seems likely to the authors that these operators and representations are extremely complex; and more importantly highly dependent, perhaps inevitable, consequences of the spatial computer and specific projections. The authors propose that the endeavor of reverse engineering the visual cortex may be an unmanageably hard task and that focusing on a learning spatial computer may lead to faster progress towards usable artificial vision systems and may also lead to massively parallel implementations that will support the computing requirements of visual systems for complex scenes.

While the computational cost of simulating the structures described in this paper on a single sequential computer are prohibitive the local distributed nature of the spatial computing formulation of the solution suggests that a massively distributed implementation using existing technology, such as FPGAs is feasible.

The spatial computer described above in support of the experiments in learning to see and act was laid out by hand. While this was appropriate for an initial experiment it is clear that achieving interesting intelligent learning systems requires that more intricate spatial computers be specified. To this end, we propose to implement a graphical language for specifying functional areas, the projections between functional areas along with distributions.

We have described an architecture that learns complex collections of visual operators that can adjust as the environment changes. For some applications it may be appropriate and even preferable to freeze the state of learning and to prevent further adaptation. Such freezing of the learned state could lead to significant performance improvements – or the ability to run on a less powerful platform – and the ability to test the performance of the system with the knowledge that it is not going to change. To accomplish this, a more aggressive clustering approach would be appropriate.

REFERENCES

[1] M. J. Farrah “The Cognitive Neuroscience of Vision” John Wiley. 2000.

[2] J. Y. Lettvin, H. R. Maturana, W. S. McCulloch and W. H. Pitts “What the Frog’s Eye Tells the Frog’s Brain”. “The Mind: Biological Approaches t its Functions” Eds. W. C. Corning, M. Balaban 1968 pp233-258

[3] J. H. R Maunsell and D. C. Van Essen “The connections of the middle temporal visual area (MT) and their relationship to a cortical hierarchy in the macaque monkey” J. Neuroscience Vol 3 #12 pp2563-2586. Dec 1993.

[4] D. C. Van Essen and J. H. R. Maunsell “Hierarchical Organization and the Functional Streams in the Visual Cortex. TINS September 1983 pp370-375.

[5] P. Robertson, S. Reece, and J. M. Brady “Algorithms for Adaptation in Aerial Surveillance.”, AFRL-IF-RS-TR-2002-52. 2002

[6] P. Robertson and R. Laddaga. “GRAVA: An Architecture Supporting Automatic Context Transitions and its Application to Robust Computer Vision”, Proceedings of the 4th International and Interdisciplinary Conference CONTEXT 2003, Stanford, CA

[7] C. Blakemore and G. F. Cooper “Development of the brain depends on the visual environment” Nature Oct 31; 228(5270) pp477-478, 1970.

[8] A. Moore. A tutorial on kd-trees. Extract from PhD Thesis, 1991. Available from http://www.cs.cmu.edu/simawm/papers.html.

[9] F. Kaplan P. Oudeyer and V.V. Hafner. Intrinsic motivation systems for autonomous mental development. IEEE Transactions on Evolutionary Computation, 11, April 2007.

[10] S. Sing A. G. Barto and N. Chentanez. Intrinsically motivated learning of hierarchical collections of skills. In Proceedings of the 3rd International Conference on Developmental Learning (ICDL 04), 2004.

[11] J. R. Anderson. A spreading activation theory of memory. Journal of Verbal Learning and Verbal Behavior, 22:261–295, 1983.

[12] G. S. Berns and T. J. Sejnowski. A computational model of how the basal ganglia produce sequences. Journal of Cognitive Neuroscience, 10.

[13] R.A. Brooks. A robust layered control system for a mobile robot. IEEE Journal of Robotics and Automation, RA-2:14–23, 1986.

[14] R.P. Feynman. ’Surely You’re Joking, Mr. Feynman!’ (Adventures of a Curious Character). W. W. Norton and Company, 1985.

[15] D. Marr. A theory for cerebral neocortex. Proceedings of the Royal Society of London B, 176:161–234, 1970.

[16] D. Precup R. S. Sutton and S. Singh. Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning. Artificial Intelligence, 112:181–211, 1999.

[17] H.A. Simon. Sciences of the Artificial. MIT Press, 1996.

[18] M. V. Srinivasan “Animal Behavior: Homing in on Ant Navigation.”, nature 411, pp752-753 June 2001

[19] A.R Damasio, D. Tranel, and H. Damasio, "Somatic markers and the guidance of behaviour: theory and preliminary testing" pp. 217-229. In H.S. Levin, H.M. Eisenberg & A.L. Benton (Eds.). *Frontal lobe function and dysfunction*. New York: Oxford University Press. (1991).