A COMPUTATIONAL NEUROSCIENCE MODEL OF WORKING MEMORY WITH APPLICATION TO ROBOT PERCEPTUAL LEARNING

M. Tugcu[1], X. Wang[1]J. E. Hunter[1], J. Phillips[2], D. Noelle[2] and D. M. Wilkes[1], *Member, IEEE*
[1]Center for Intelligent Systems
Vanderbilt University
Nashville TN 37235-0131 USA
{mert.tugcu;xiaochun.wang;jonathan.e.hunter;mitch.wilkes}@vanderbilt.edu

[2]Dept. of Engineering
Merced University
Merced, CA USA 95344
{jphillips7;dnoelle}@ucmerced.edu

ABSTRACT
An issue of critical importance to both robots and biological creatures is the efficient use of the limited resources available for survival. One strategy employed by higher animals is the focusing of attention onto the few items, e.g., percepts, salient to reaching the current goals, and ignoring distracting input. In these animals, the pre-frontal cortex working memory plays a significant role in the focus of attention. A recently developed Working Memory Toolkit (WMtk) is based on a computational neuroscience model of working memory. We apply this model/toolkit to two perceptual learning problems from robot vision related to navigation and landmark detection. Our system is described along with two perceptual learning experiments. The results of these experiments are given and show impressive performance both in terms of accuracy and speed of learning. To our knowledge, this is the first such application of a computational neuroscience model of working memory to a robot.

KEY WORDS
Learning, working memory, perception, and computer vision

1. Introduction

Robotics is a fascinating research topic. For centuries people have been interested in building machines that mimic living beings. Robots have reached a high level of performance, as can be seen in the literature, yet there is much left to do to achieve the level of lifelike sophistication that many desire. This level of intelligence requires the ability to self-organize, learn and adapt to complex, dynamic, and uncertain environments. Fukuda *et al.* Have observed that sensing, perception, cognition, planning, control and actuation and fundamentally important to obtaining an intelligent robot [1]. In this paper we employ a novel implementation of a computational neuroscience model of working memory to the problem of perceptual learning.

Perception is an awareness of things through the physical senses. Perceptual learning is the ability to construct compact representations of sensory events based on their statistical properties in the perceptual level as opposed to the behavior or cognitive level [2][3]. However, not having a human's sophisticated vision system, the robot perceives the world differently. Therefore, there is no guarantee that the robot vision system can recognize or detect objects defined by human designers as easily as well. As a result, to operate in and to recognize objects in realistic environments, a robot should have the ability to form percepts on its own, either learned through interaction with a supervisor or deduced autonomously by natual association among features of sensory information and, based on that, to segment the image into meaningful regions [4][5].

Perceptual systems support the performance of motor actions [6]. However, at any given moment there may be a very large number of sensory stimuli impinging on the robot, and many of these may be irrelevant to successful completion of the task at hand. It is critical that the robot be able to focus its attention on the few relevant percepts, and thus ignore the many other distracting ones [7], since focusing on the distracters consumes valuable resources. One way of achieving this is the use of some form of reinforcement learning (RL) in which focusing attention onto percepts that facilitate achieving the current goal results in reward [8]. In fact, recent models of pre-frontal cortex working memory (WM) have been developed by computational neuroscientists, and these models employ a modified form of RL to learn to focus attention onto the most salient information [11][12].

The behavior of working memory goes beyond simple stimulus-response reflexes and is typified by delayed-response tasks [9]. In neuroscience, a popular neuro-computational account of working memory function has been proposed based on the evidence that the midbrain dopaminergic system is associated with reward learning [11]. Motivated by this biological inspiration, an open source software library, called the Working Memory Toolkit, has recently been developed by J.L. Phillips and D.C. Noelle at Vanderbilt University, to provide a computational neuroscience model of prefrontal cortex [12].

We have been integrating such an adaptive working memory structure into a robotic system involving perceptual learning, actuators, reasoning, and short- and long- term memory structures by embedding this working memory mechanism into robotic control system [13]. In this paper, we present the results of experiments which demonstrate that robots equipped with such perceptual learning and behavior learning systems can learn to perform tasks successfully in realistic environments. In the following, Section II talks about working memory and the Working Memory Toolkit. Section III describes visual features and similarity measures. In Section IV, a brief review of our vision-based autonomous mobile robot system is given. In Sections V and VI, experiments are designed and results presented. Finally, conclusions are made in Section VII.

2. Working Memory

Cognitive psychologists use the term, working memory, to refer to the type of memory that is active and relevant only for a short period of time, usually on the scale of seconds. Considerable evidence is available that the brain obeys the distinction between working and other forms of memory, that the prefrontal cortex has a specialized role in working memory processes, and that there may be multiple working memory domains in the prefrontal cortex, all organized according to a common functional principle but each dedicated to a different information-processing system [14]. One classical functional model of working memory was Baddeley and Hitch's three-component model in which a Central Executive controls two working memory subsystems: the phonological loop, which is responsible for the storage and processing of linguistic, and the visuo-spatial sketch pad, which is responsible for the storage and processing of the visio-spatial information [10].

The neuro-computational model for working memory is based on reinforcement learning developed in the machine learning community, in which learning about stimuli or actions is solely on the basis of rewards and punishments associated with them [8]. Unlike supervised learning, reinforcement learning is minimally supervised because what actions to take in particular situations is not told explicitly but must be worked out on the basis of the reinforcement given. There are two broad classes of reinforcement learning. In the first case, called the static action choice, the reinforcement is delivered immediately after the action is taken. This makes learning relatively easy. In the second case, called the sequential action choice, the reward or punishment is dependent on an entire sequence of actions and, therefore, is partially or wholly delayed until the sequence is completed. This makes learning more difficult since learning the appropriate action at each step in the sequence has to be based on future expectation of reward.

Though the ability of animals to learn appropriate actions in response to particular stimuli on the basis of associated rewards or punishments is a focus of behavioral psychology, there exists evidence that the midbrain dopaminergic system is associated with reward learning. The model for it is the Rescorla-Wagner rule and temporal difference learning. The Rescorla-Wagner rule [15], as a concise account of certain aspects of classical conditioning, is based on a simple linear prediction of the reward, v, associated with a stimulus vector, $\vec{u}$, by way of weight vector, $\vec{w}$,

$$v = \vec{w}^T \vec{u} \qquad \vec{w} \rightarrow \vec{w} + \varepsilon\delta\vec{u} \tag{1}$$

In order to establish the value of the weight, a learning rule is designed to minimize the expected squared error between the actual reward and the predicted total future reward, which can be approximated as in the following,

$$\delta(t) = \sum_{\tau=0}^{T-t} r(t+1+\tau) - v(t) \approx r(t) + v(t+1) - v(t) \tag{2}$$

where $v(t)$ is the actual reward, and $v(t+1)$ is an approximation of total future reward. This approximation gives rise to the name TD learning, the *temporal difference* learning rule. The detailed description of TD learning is beyond the scope of this paper. Please see [12] for details.

The prediction error δ plays an essential role in both the Rescorla-Wagner rule and the temporal difference learning rule. It has been suggested that the activity of dopaminergic neurons in the ventral tegmental area (VTA) in the midbrain can represent this quantity. Substantial evidence has been available that dopamine is involved in reward learning. The similarity between the responses of the dopaminergic neurons and the prediction error has been observed. Therefore, the activity of the dopaminergic neurons provides a prediction error for the reward. In other words, the prediction error is an ongoing difference between the amount of reward that is delivered and the amount that is expected.

Now the mechanism by which one of the possible responses is chosen has been solved, the important question is how to model the mechanism and implement it for the benefit of robots. David Noelle and Joshua Phillips, at Vanderbilt University, have created a set of software tools, called the Working Memory Toolkit (WMtk), for developing working memory systems that can be easily and tightly integrated into robotic control mechanisms [12]. Being general and flexible enough to be used on a variety of robotic platforms, the WMtk provides an extensive API that facilitates the fabrication of systems that utilize biologically-inspired working memory components. As a software library written in ANSI C++, the WMtk consists of a set of classes and methods that are designed to aid in the construction of a working memory system that uses TD learning to select working memory contents. Since it is biologically inspired, the working memory system utilizes a neural network to make decisions about memory management. On each "time step" of the system simulation, it makes intelligent decisions such as when its contents should be updated and/or protected (erased and/or retained). The WMtk is a configurable system, having parameters that can be adjusted depending on the environmental requirements. Furthermore, it requires several user-defined functions to determine how the reward of the current time step is going to be produced, or how the contents of the working memory will be released when no longer needed by the system. Before describing the use of the WMtk in our experiments, the visual features and the mobile robot system we use are first introduced.

3. Visual Features

Color is an identifying feature that is local and largely independent of view and resolution. As a representation of how much of each color component occurs in a small image region, a color histogram vector, invariant to translation, rotation and scale [16], is used. Further, due to their similarity to the receptive fields of V1 simple cells, Gabor filters (or wavelets) have been frequently used as texture-based similarity measures in image processing. As a result, the combination of the color histogram and Gabor texture measures is a useful representation for segmenting images.

3.1 HSV Color Histogram

In order to construct color features, a histogram of color measurements in HSV space for each object is computed as follows: The hue is broken evenly into 100 bins, [0.00 … 1.00]. The saturations and values are evenly distributed into 10 bins each ranging from 0.00 to 1.00. Each color can be represented by combination of the three bins, resulting in 10,000 different color features or the bins for the histogram. The histogram then can be constructed by looking at each color feature in a selected region and finding the number of pixels that corresponds to the associated bin in the histogram. Thus, there are 10,000 discrete color bins, each representing the percentage of pixels of a certain color that are in a small region of the. The reasons why we use such a high dimensional space are: (1) to differentiate colors as much as possible, and (2) to mimic sparse coding as observed in the brain [18]. There have been strong theoretical reasons and experimental evidence suggesting that the brain uses a relatively small subset of neurons to represent each information item (e.g., specific sensory stimuli from an object), rather than being represented either by the activity of a single, individually meaningful cell or by the global activity pattern across a whole cell population [17]. This is known as sparse coding.

3.2 Gabor Texture Measures

Orientation stimuli are first sensed in the region of the visual cortex known as V1. The optimal stimulus in V1 is a sine-wave grating described by its spatial frequency. Most neurons in the striate cortex respond best when a sine-wave grating of a particular spatial frequency is placed in the appropriate part of the visual field. For orientation-selective neurons, the grating must be aligned at the appropriate angle of orientation.

The complex Gabor filters, first introduced by Gabor, are complex exponentials with a Gaussian envelope, or Gaussians modulated by complex harmonics. From experimental data fitting, a mathematical approximation of the spatial receptive field of a simple cell may be provided by a Gabor function, which is a product of a Gaussian function and a sinusoidal function. When placing the origin of the coordinates at the center of the receptive field, the observed receptive field structures using a Gabor function is approximated to be [18],

$$D_s(x,y) = \frac{1}{2\pi\sigma_x\sigma_y}\exp\left(-\frac{x^2}{2\sigma_x^2} - \frac{y^2}{2\sigma_y^2}\right)e^{j(kx+\phi)}$$

(3)

where σ_x and σ_y determine the extent of the receptive field in the x and y directions, respectively; k is the preferred spatial frequency and determines the spacing of light and dark bars that produce the maximum response; and ϕ is the preferred spatial phase.

Each of the 40 Gabor wavelets are applied to the image, and the average energies of the filtered outputs are computed for each small image region (described in more detail in Section V). These average energies are used as orientation and texture features.

3.3 Similarity Measures

The patterns belonging to the same perceptual cluster should be close together in the feature space, while patterns in different clusters should be further apart from one another. A major issue, in order to discover similarities and dissimilarities and to reveal the organization of patterns into "sensible" clusters for both supervised learning and unsupervised learning, is to define the "similarity" criteria between two feature vectors. After setting up an appropriate similarity measure, the next step is to design an algorithm to search for similarities and dissimilarities among the input patterns and then cluster the vectors on the basis of the adopted similarity measure.

In a K-dimensional space, two commonly used special cases of this metric family are the L_1-norm and L_2-norm in mathematics, which are obtained by setting $r = 1$ or 2, respectively, in the Minkowski power metric formula,

$$d_{ij} = \left(\sum_{k=1}^{K} \left| x_{ik} - x_{jk} \right|^{r} \right)^{1/r} \tag{4}$$

In multidimensional scaling literature, these two metrics are also called the city-block distance and the Euclidean distance, and believed to represent two types of processing. As indicated by Shepard, Garner and Shepard [19][20][21], for unitary or holistic stimuli, such as hue, saturation and brightness of colors, the closest approximation to an invariant relation between data and distances has uniformly been achieved in a space endowed with the familiar Euclidean metric; on the other hand, for analyzable or separable stimuli, such as size and brightness difference (orientation), the closest approach to invariance has generally been achieved with the city-block metric. To simulate this rule, the Euclidean distance is used for the HSV color histogram component of the feature vector and the city-block distance is used for the texture measure component of the feature vector. That is,

$$d_{ij} = \sqrt{\sum_{k=1}^{10000} \left(x_{ik} - x_{jk} \right)^2} + \sum_{k=10001}^{10041} \left| x_{ik} - x_{jk} \right| \tag{5}$$

3.4 Clustering and the Search Tree

Based on this mixed metric, the minimum spanning tree method is used first to find the representative points for each cluster. Then, a nearest neighbor labeling algorithm is used to assign labels to the remaining points according to the label of their nearest neighbor, thus building up the database.

Performing a true nearest neighbor search over a large database is time consuming, giving rise to the need for efficient search structures and algorithms. Common structures used to organize data for efficient nearest neighbor search include the Kd-tree (K-dimensional tree) [22], the R-tree [23], Quad-trees [24], and Binary space partitioning (BSP) trees [25].

In order to achieve efficient search, a tree-structured vector quantization method is used to generate an approximate 3-way nearest neighbor searching tree. Given the obtained training set, at the first level, an initial set of 3 representative patterns are selected as the cluster centers. Then the whole set of items is clustered into three subsets by assigning each feature vector to its closest representative cluster center according to the mixed distance measure. At the second level, for each of the 3 clusters obtained, three feature vectors are selected as the cluster centers for the subset of items at the node. This procedure continues until either all the leaf nodes belong to the same object class (a pure node) or the number of leaf nodes is below some limit, e.g., one hundred. Every feature vector in the leaf nodes has a landmark associated with it.

In order to search through the tree, given a new feature vector, its distances to the cluster centers at the top tree node are calculated and the winner is the cluster center that the feature vector is nearest to. This process is continued as the tree is descended until a leaf node is reached. When coming to a leaf node, if it is a pure node, then stop. Otherwise, do a nearest-neighbor search, and the winner is the training feature vector with minimum distance according to the chosen metric.

4. Our System

A functional characterization of our mobile robot system is shown in Figure 1. As mentioned previously, the features used in this system are the 10,000 element HSV color histogram and 41 Gabor texture measures. For new images obtained by the robot, the segmentation should yield a set of connected regions in the image, as shown in Figure 2. A connected-component labeling algorithm is then applied to find these regions, called blobs, and are used to produce the input data for working memory, called "chunks." Each blob not only represents one specific cluster in the feature space, but also represents a clustering of pixels in the image space.

A fundamental property of a mobile robot is the ability to navigate. Therefore, the next step for the development of this system is for the robot to acquire sensory-motor associations to use to guide movement. Suppose that the response system of the robot has a repertoire of innately programmed activities, one of which is to approach and investigate stimuli that are either obstacles or open space, the first experiment was designed for this purpose using the WMtk.

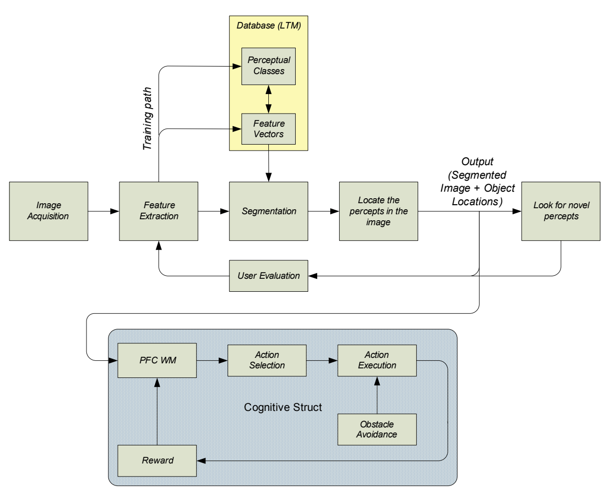


Fig. 1: Diagram of our robot system.

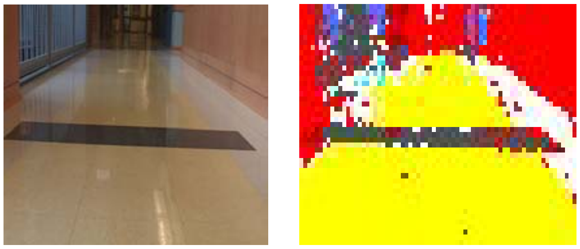


Fig 2: A typical segmentation result for our system.

5. Method

In the following, each experiment is conducted using both a supervised and an autonomous method for classification. The supervised version uses labels set by the user to classify the feature vectors for the training set. The autonomous version takes the vectors in the training set and classifies them by groups using a minimum spanning tree method.

For the autonomous method, the training database is created from 20 pictures taken from the environment, feature vectors are extracted resulting in 66,740 feature vectors. Based on the nearest neighbor distance, a minimum spanning tree is constructed. Next the edges in the tree are sorted and cut in decreasing order until the number of feature vectors in the largest cluster is below an empirically chosen threshold. Then any cluster which has more than 100 feature vectors is believed to represent a percept. Finally, a nearest neighbor labeling algorithm is used to assign labels to the remaining points. Each class is assigned a different color for visual display of the segmentation. These classified vectors are used to create an approximate nearest neighbor search tree by which new feature vectors are evaluated efficiently.

The database for the supervised method is obtained from on-site classification of percepts by the user. The database was collected over several days at different times to account for objects in the environment under differing lighting conditions. The GUI of our system is designed to facilitate the rapid collection of supervised feature vectors. The current set of 58,198 classified vectors collected are then used to create the approximate nearest neighbor search tree.

The images that are collected by the robot are broken into overlapping 15x15 pixel windows with centers that increment 10 pixels vertically and horizontally. These numbers were empirically chosen and found to yield acceptable results for our experiments. Each of the vectors evaluated from the image are classified using the search tree. The vision system allows the robot to identify percepts with good reliability. The resulting blobs are analyzed to extract size and position information and are sorted by size from largest to smallest.

6. Experiments and Results

6.1 Experiment 1: Learn Percepts for Motion

The first experiment is to learn properties of percepts through the robot's own interaction with the world. The task the robot is expected to perform in this experiment is to learn the meaning of the percepts with regard to motion.

The experiment takes place in the hall outside of the Intelligent Robotics Laboratory in Featheringill Hall. The hallway has yellow floor tiles in the center with white tiles along the edges near the wall. The hall also has periodically located black tiles across the center of the hall forming a decorative stripe. The walls consist of wood panels, light blue painted wall sections, and a light blue railing. There are also bricks columns and blue floor tiles at the ends of the hall.

Suitable percepts are formed and several meaningful blobs are obtained from images acquired by the robot's camera. Nevertheless, these are still just sensory inputs to the robot. Since the robot does not have any prior information about these percepts and their association with the task of motion, the task in this experiment is to find whether the percepts represent an obstacle or open space. To do so, the following steps were carried out by the robot equipped with the WMtk:

- The WM size parameter is set to 1 to evaluate the percepts individually. This means that the WM can select at most one of the available chunks per episode.
- The robot obtains a picture and segments the picture into blobs. Only the bottom half of each picture is used for segmenting. Each blob is used as an input chunk to the working memory toolkit. The WM chooses one of these chunk percepts.
- The robot directs itself toward the percept and proceeds forward for 5 meters or until the sonar sensor indicates an obstacle, forcing the robot to stop.
- If the robot moved 5 meters without detecting an object with its sonar, it will receive the maximum reward of 5. If the robot moved less than 5 meters, it will receive a partial reward of the distance it did travel.
- The robot is trained through multiple episodes in the environment and the expected reward for each percept is monitored.

Since working memory is associated with a goal-directed task, the goal in this experiment is to see whether the robot will choose the best percept for distance travel. Two sets of experiments were conducted to show the effectiveness of the WMtk. In the first set, the robot was put in the middle of a hallway and let to go freely without repositioning the robot after each trial. In the second set, the robot was put in the middle of the hallway to begin. Instead of letting it go freely, after each trial, it was put back to the center of the hallway to begin the next trial. Further, in each set, after one percept is determined to be the dominant precept associated with open space, that percept will be blocked to see how the remaining percepts will be affected in working memory.

With a WM size of 1, the weight parameters inside the WMtk are directly associated with the percepts, and are related to the expected reward associated with these percepts. The plots of these weights through the episodes are shown in Figures 3 and 4. For all the trials for experiment 1, the general trend is about the same, the percepts that are detected decrease in value until around episodes 50-70 where the dominant percept begins to grow in a linear fashion. All percepts that are not encountered often do not change much or at all from the initial value. In the autonomous case, the percept labeled Object 10 becomes the dominant percept for motion, which is expected, since that percept represents yellow floor. For the supervised case, the percept labeled YellowFloor also becomes the dominant percept for motion. The conditions of free travel versus forced centered travel seem to make no significant difference for the speed with which the robot learned this task.

Figure 4 shows the percept YellowFloor being blocked at about episode 90 to observe the influences to the other percepts. The result shows that the percept labeled BlackFloor begins to grow and the other percepts that would represent less motion such as WoodPanel are further decreased. The blocking of the dominant percept allows for further development of the other percepts clearly showing a second best option.

In experiment 1, the robot has learned which percept (YellowFloor / Object 10) is the best for motion, i.e., which percept represents open space.

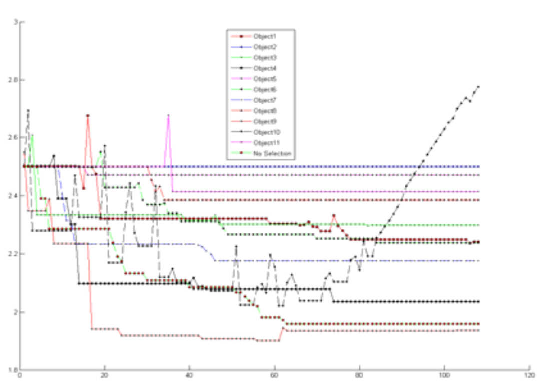


Fig. 3. Experiment 1 – Autonomous - Free

6.2 Experiment 2: Learning a Landmark

The second experiment is for the robot to use working memory to learn to recognize landmark locations in the environment through the percepts. The identification of landmarks in the environment can be advantageous for navigation. In the first experiment, the robot learned what percept to move toward to move freely in the hall environment. Now it must learn about landmark locations for tracking its location in the environment. To train the robot on landmark locations the following steps are taken:

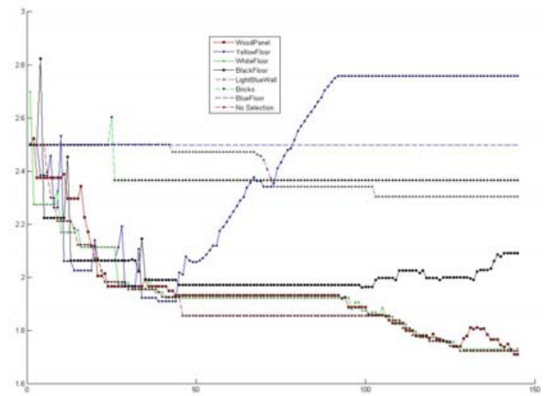
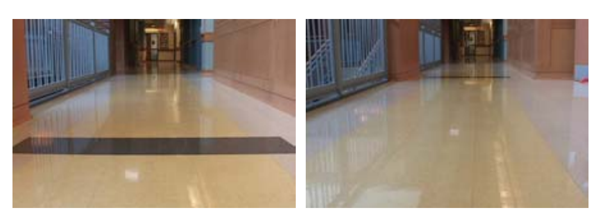


Fig. 4. Experiment 1 – Supervised – Free

- The WM size is set to values of 1, 2, and 3 to determine which percept or combinations of percepts are most useful for recognizing a landmark location.
- A set of training images are taken from the environment to represent landmark and non-landmark locations as seen in Figure 5. The presence of the black floor tiles in the foreground is the visual indication of the landmark. The set contains an equal number of images containing landmarks and not containing landmarks.
- The training images are selected in random order and processed using both the supervised and autonomous search trees from experiment 1.
- Each percept that is available in the image is presented twice to the WM, once as a chunk that indicates the presence of a landmark and once as a once indicating no landmark present. Once the WM has selected a set of chunks, a vote is taken to determine whether the majority vote is for landmark or for no landmark. A tie vote is interpreted as a choice of no landmark. If the choice is correct the reward is 1, otherwise it is 0.
- Each of the WM sizes is trained for both 1000 and 5000 episodes.



(a) (b)

Fig. 5: (a) Landmark (b) Non-Landmark

The task of learning the landmark location converges quickly in all cases. Figure 6 is a plot of the average reward values over a sliding 25 point window over 1000 episodes. As can be seen, the reward value converges to 1 in about 50 to 60 episodes.

Within the WMtk there is a parameter for exploration. A higher value for this parameter increases the chance that the working memory system will explore other options in the choices rather than just pursuing the maximum expected reward. The percentages noted in Table A result from the trials with the exploration percentage set at 0.05. Table B is the result using only the weights within the WM to determine landmark choices without exploration, i.e., the exploration parameter is set to 0. The percentage values for both tables are including the early misclassifications that occur in the WM training. The percentages are higher in Table B than in A and more accurately represent the performance of WM after training.

Examination of the log files of the choices made during training show a consistent tendency towards percept BlackFloor_L (BlackFloor indicating landmark) for the supervised case, and Object1_L (Black Floor indicating landmark) for the autonomous case as can be seen in Table C. Note that Object1 = black floor and Object 10/11 = yellow floor. This may be interpreted as the system learning that the percepts that correspond to the black floor tiles are indicative of the landmark.

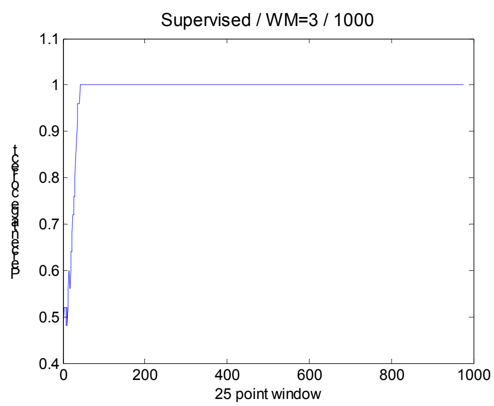


Fig. 6: Average reward over a sliding 25 point window.

Table A. Percentage correct for Landmark classification with WM exploration set at 0.05

WM size	Autonomous 1000 episodes	Autonomous 5000 episodes	Supervised 1000 episodes	Supervised 5000 episodes
1	92.30%	94.04%	93.10%	85.86%
2	85.10%	95.68%	90.00%	93.20%
3	95.60%	97.00%	95.30%	96.60%

Table B. Percentage correct for Landmark classification with WM exploration set at 0.00

WM size	Autonomous 1000 episodes	Autonomous 5000 episodes	Supervised 1000 episodes	Supervised 5000 episodes
1	96.70%	99.52%	98.10%	99.52%
2	95.50%	99.72%	95.50%	97.42%
3	95.60%	99.62%	98.00%	99.38%

The landmark percepts for the autonomous WM size 3 in Table C does not use Object1_L, but continues to correctly classify the images. Nonetheless, it does continue to observe that the presence of the black floor tile percepts is important to correctly identifying the landmark images. The WM has shown that it can learn which percepts identify landmark locations quickly and with high accuracy.

Table C. Landmark Percepts for 1000 episodes

WM size	Autonomous Landmark Percept/s	Supervised Landmark Percept/s
1	Object1_L	BlackFloor_L
2	Object1_L Object10_L	BlackFloor_L YellowFloor_L
3	Object10_L Object1_NL Object9_L	BlackFloor_L BlackFloor_NL YellowFloor_L

7. Conclusion

This paper explored an issue of critical importance to both robots and biological creatures regarding the efficient use of the limited resources available for survival. The strategy of learning to focus attention onto only a few salient items/percepts, was used to address this problem. A recently developed Working Memory Toolkit (WMtk) is based on a computational neuroscience model of pre-frontal cortex working memory, and was found to be valuable to this strategy. We applied this model/toolkit to two perceptual learning problems from robot vision related to navigation and landmark detection. The results of these experiments were given and show impressive performance both in terms of accuracy and speed of learning.

Acknowledgement

We would like to express our gratitude to the National Science Foundation for its very valuable support of this work under award 0325641.

References

[1] T. Fukuda, R. Michelini, V. Potkonjak, S. Tzafestas, K. Valavanis., & M. Vukobratovic How far away is “artificial man”?, *IEEE Robotics & Automation Magazine*, 8(1), 2001, 66-73.

[2] B.A. Olshausen & D. J. Field, Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381, 1996, 607-609.

[3] R. Goldstone, Perceptual learning. *Annu. Rev. Psyhchol*, 49, 1996, 585-612.

[4] J.M. Buhmann, T. Lange & U. Ramacher, Image segmentation by networks of spiking neurons, *Neural Computation*, 17, 2005, 1010-1031.

[5] J. Hunter, Human Motion Segmentation and Object Recognition using Fuzzy Rules, *Proceedings of 14th Annual IEEE International Workshop on Robot and Human Interactive Communication*, Nashville, TN, 2005, 210-216.

[6] R.W. Sperry, Neurology and the mind-brain problem, *American Scientist*, 40, 1952, 291-312.

[7] P. Milner, *The Autonomous Brain: A Neural Theory of Attention and Learning*, (Mahwah, N.J.:L. Erlbaum Associates, 1999).

[8] R.S. Sutton, *Reinforcement learning: an introduction*, (Cambridge, Mass. : MIT Press, 1998).

[9] W. S. Hunter, The delayed reactions in animals and children, *Behav Monogr*, 2, 1913, 1-86.

[10] A. Baddeley, Working memory: looking back and looking forward, *Nature Review Neuroscience*, 4(10), 2003, 829-839

[11] R.C. O'Reilly, D.C. Noelle, T.S. Braver, & J.D. Cohen, Prefrontal cortex and dynamic categorization tasks: Representational organization and neuromodulatory control, *Cerebral Cortex*, 12, 2002, 246-257.

[12] J.L. Phillips & D.C. Noelle, A biologically inspired working memory framework for robots, *Proc. of the 27th Annual Meeting of the Cognitive Science Society*, Stresa, Italy, 2005.

[13] M. Wilkes, M. Tugcu, J. Hunter, & D.C. Noelle, Working memory and Perception, *Proc. of the IEEE Intl. Workshop on Robot and Human Interactive Communication (RO-MAN)*, Nashville, TN, 2005, 686-691.

[14] P.S. Goldman-Rakic, Working memory dysfunction in schizophrenia, *Journal of Neuropsychiatry Clin Neurosci*, 6, 1994, 348-357

[15] R.A. Rescorla, and A.R. Wagner, A theory of Pavlovian conditioning: The effectiveness of reinforcement and non-reinforcement, *In A.H. Black and W.F. Prokasy, eds., Classical Conditioning II: Current Research and Theory, 64-69*, (New York, NY: Appleton-Century-Crofts, 1972).

[16] M.J. Swain, Color Indexing, *International Journal of Computer Vision*, 7(1), 1991, 11-32.

[17] Foldiak, P. Sparse coding in the primate cortex. *In M.A. Arbib (Ed.), The handbook of brain theory and neural networks (2nd ed)*, (Cambridge, MA: MIT Press, 2002)

[18] P. Dayan, & L.F. Abbott, *Theoretical Neuroscience*, (Cambridge, MA: MIT Press, 2001).

[19] R.N. Shepard, Attention and the metric structure of the stimulus space. *Journal of Mathematical Psychology*, 1, 1964, 54-87.

[20] W.R. Garner, *The processing of information and structure*, (New York, NY: Wiley, 1974).

[21] R.N. Shepard, Toward a universal law of generalization for psychological science, *Science*, 237, 1987, 1317-1323.

[22] J.L. Bentley, Multidimensional binary search tree used for associative searching, *Communications of the ACM*, 18, 1975, 509-517.

[23] A. Guttman, R-tree: A dynamic index structure for spatial searching, *Proceedings of the ACM SIGMOD Conference*, Boston, MA, 1984, 47-57.

[24] R. Finkel & J. Bentley, Quad trees: A data structure for retrieval on composite keys, *Acta Informatica*, 4, 1974, 1-9.

[25] M. De Berg, M. De Groot & M. Overmars, New results on binary space partitions in the plane, *Computational Geometry Theory and Applications*, 8, 1997, 317-333.