A VISION-BASED PERCEPTUAL LEARNING SYSTEM FOR AUTONOMOUS MOBILE ROBOT

By

Xiaochun Wang

Dissertation
Submitted to the Faculty of the
Graduate School of Vanderbilt University
in partial fulfillment of the requirements
for the degree of
DOCTOR OF PHILOSOPHY
in
Electrical Engineering

August, 2007

Nashville, Tennessee

Approved:
Professor Mitch Wilkes
Professor Kazuhiko Kawamura
Professor Douglas Hardin
Professor Nilanjan Sarkar
Professor David Noelle

ACKNOWLEDGEMENTS

I would like to take this opportunity to thank Dr. Mitch Wilkes, the chairman of my Ph.D committee, for being so supportive of my academic goals and for being so kind to provide me with the protected academic time and financial aid to pursue these goals. As my advisor and mentor, he has shown me what a good professor should be.

I am also grateful to all of those whom I have had the pleasure to know during my study at Vanderbilt, particularly, the Electrical Engineering department, which supported me financially to fulfill my course work, Dr. James A Cadzow and Dr. Francis Wells, who helped me enter the Ph.D program, Dr. Kawamura, without his strict requirements on my research proposals I could not be clear about what I was supposed to do for my Ph.D, Dr. David Noelle, without his kind suggestions and efforts about my proposed experiments I could not be able to finish my study so soon, Dr. Douglas Hardin and Dr. Nilanjan Sarkar, who are so nice to be willing to serve on my committee, Dr. Bharati Mehrotra, who is so kind to be willing to share with me her experiences as a scientific researcher, Flo Fottrell and her husband, for always being there to cheer me up and to teach me how to make appetizers, Jonathan Hunter, Mert Tucgu, Soradech Krootjohn, Faisal.Al-hammadi for their friendships and technical supports. Without all these helps, this work would not have been possible.

Finally, I would like to thank my parents, my brother and my sister-in-law, and my lovely nephew for their unconditional love, financial support and guidance which are with me in whatever I pursue.

ABSTRACT

Autonomous robots are intelligent machines capable of performing tasks in the real world without explicit human control for extended periods of time. A high degree of autonomy is particularly desirable in fields where robots can replace human workers, such as state-of-the-practice video surveillance system and space exploration. However, not having human's sophisticated sensing and control system, two broad open problems in autonomous robot systems are the perceptual discrepancy problem, that is, there is no guarantee that the robot sensing system can recognize or detect objects defined by a human designer, and the autonomous control problem, that is, how the robots can operate in unstructured environments without continuous human guidance. As a result, autonomous robot systems should have their own ways to acquire percepts and control by learning.

In this work, a computer vision system is used for visual percept acquisition and a working memory toolkit is used for robot autonomous control. Natural images contain statistical regularities which can set objects apart from each other and from random noise. For an object to be recognized in a given image, it is often necessary to segment the image into nonoverlapping but meaningful regions whose union is the entire image. Therefore, a biologically based percept acquisition system is developed to build an efficient low-level abstraction of real-world data into percepts. Perception in animals is strongly related to the type of behavior they perform. Learning plays a major part in this process. To solve how the robots can learn to autonomously control their behavior based on percepts they've acquired, the computer vision system is integrated with a software package called the Working Memory Toolkit (WMtk) for decision making and learning. The WMtk was developed by Joshua L. Phillips & David C. Noelle based on a neuron computational model of primate working memory system. The success of the whole system is demonstrated by its application to a navigation task.

TABLE OF CONTENTS

Appendix

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

Robotics is a fascinating research topic, and for centuries people have been interested in building machines that mimic living beings. After four robot design generations, from the early days of the application of control theory to complex systems, followed by the sense-think-act cycle of the functional approach, and then to the Brooks' behavior-based robotics, and finally to the recent cognitive robotics, different branches, such as industrial robots, mobile robots, and humanoid robots, have come to a remarkable level of maturity as evidenced in the literature. However, to build and realize such artificial intelligent creatures, capable of moving and operating autonomously so as to replace trained individuals in a variety of applications, still remains as the ever-existing and continuous desire of mankind. As an absolute prerequisite for the design of such robots, human-like intelligence refers to the ability to self-organize, to learn and adapt to varying, complex, dynamic, and uncertain environments, and to interact with the environment and perform tasks efficiently and safely through using their on-board reasoning/problem solving capabilities. To imitate humans, any intelligent robot should be made up of elements such as sensing, perception, cognition, planning, control and actuation [1].

Perception is an awareness of things through the physical senses, especially, vision. 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 from the human, giving rise to the problem of perceptual discrepancy. 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 realistic environments and to recognize objects in a given image, a robot must have the ability to form percepts on its own, either learned from interacting with a supervisor or figured out by natual association among features of sensory information and, based on that, to segment the image into nonoverlapping but meaningful regions whose union is the entire image [4]. In general, by meaningful, the regions in the segmented image should be homogeneous and should have well-defined boundaries, denoting reliable objects. With high spatial frequencies being filtered out, image segmentation reduces the amount of storage and makes behavior resistant to the loss of information. Our motivation is to develop an object recognition vision system based on a set of features associated with each object in the image to assist in the analysis of data that may be applied successfully across different appplication domains [5].

Perceptual systems evolved to improve the performance of the motor system, however, not the other way around [6]. For example, given several sources of sensory stimuli present trying to attract attention in different directions, the agent may have to choose between possible courses of action and, when one has been decided on, must prevent any irrelevant input from interfering with the behavior. Therefore, perception and action are tightly coupled and any basic behavior defines a category of action in response to a specific sensory input. In this process, learning plays a major part, "enabling animals to pass the motivational properties of innately reinforcing stimuli to accompanying or perceptibly related stimuli, making attention originally directed only to an innate reinforcer direct to percepts associated with the reinforcer, and allowing initially neutral objects to arouse a motivational state and links the motivation system to perceptual systems" [7]. For instance, to survive, animals must identify sources of nourishment and dangerous situations, and respond accordingly by approaching the food and fleeing the danger. It is in this sense that behavior learning is the process that forms association between perception and action, established by reinforcement in the form of reward and punishment. For higher animals such as humans, evidence suggests that the occurrence of a response depends on its predicted outcome, so that behaviors are planned on the basis of future benefits rather than present contingencies alone [8]. If the outcome of an action satisfies this expectation, an innate reward is issued, facilitating the intended behavior. This kind of learning is referred to as reinforcement learning in the machine learning community and has been applied to many robotics tasks [9].

Then how the perceptual learning and behavior learning can be organized to fulfill some task toward a goal? To answer this difficult question, it would be useful to look at the biological solution examined by human and higher animals. As will be explained in the next chapter in more details, in human (primate) brains, the posterior cortex, including the occipital, parietal, and temporal lobes, largely supports perception and perceptual memory, while the frontal areas of cortex, consisting of prefrontal, premotor, and motor areas, supports action and motor memory. The whole cerebral cortex projects to the basal ganglia, and the outputs then funnel back to the frontal cortical areas, to some non-cortical motor systems, and to dopaminergic systems in the midbrain. Neurons in the dopaminergic systems project to both the basal ganglia (thus forming a feedback loop) and the frontal areas, and play an important role in the learning of behavior [10]. Results from lesions and some neurophysiological studies suggest that the prefrontal cortex, and, to a certain extent, also the premotor cortex, may serve to define the goals of future actions and order sequential actions toward such goal, and provide neural support to three cognitive functions that are indispensable for conducting temporal sequences: working memory, motor attention, and inhibitory control. Working memory is memory in the active state that the organism uses for the performance of acts toward a goal. Motor attention is the preparation for action or attention directed to prospective action. The inhibitory control of interference is integrative and prevent external or internal interfering influences from conflicting with and leading astray the structure of behavior, speech, or thought [11][12].

Introduced ninety years ago by W.S. Hunter, the behavior of working memory goes beyond simple stimulus-response reflexes and is typified by goal-directed delayed-response tasks [13]. In psychology, the most famous model for the working memory is the Baddeley & Hitch model, in which a central executive acts as the supervisory system and controls the flow of information from and to two short-term storage systems: the phonological loop and the visuo-spatial sketchpad [14]. In computational neuroscience, a popular neurocomputational account of working memory function has been proposed based on evidence that the midbrain dopaminergic system is associated with reward learning [15]. Motivated by this biological inspiration, an open source software library, called the Working Memory Toolkit, was developed by J.L. Phillips and D.C. Noelle at Vanderbilt University to provide a computational neuroscience model of the prefrontal cortex [16].

The process of embedding such an adaptive working memory structure into a realistic robotic system involving perceptual learning, actuators, reasoning, and short- and long- term memory structures has already been addressed [17]. In this thesis, it will be demonstrated that robots equipped with autonomous perceptual learning and behavior learning systems can perform tasks successfully in a real world environment.

In this the following, Chapter II talks about visual perceptual acquisition. Perceptual learning is described in Chapter III together with an introduction to working memory and the Working Memory Toolkit. Autonomous navigation is given in Chapter IV. Chapter V talks about a vision-based autonomous perceptual learning system built upon a mobile robot platform. In Chapter VI, experiments are designed as tasks to test this vision-based autonomous perceptual learning system. Finally, experimental results are presented and conclusions are given in Chapter VII.

CHAPTER II

VISUAL PERCEPT ACQUISITION

Eighty percent of our perceived information about the external world reaches us by way of the eyes. As a result, vision is the primary sensory modality. Disscussions of vision often consist of three distinct stages: sensation, perception, and recognition. As the initial stage of sensory transduction, sensation begins with photon receptors in the retina. Perception is the active process of selecting, organizing, interpreting the information detected by sensors, and transforming these raw sensory signals into distinct percepts. As certain categories in the mind, percepts are the brain's internal representation of specific external visual stimuli and are the results of complicated interactions between multiple visual areas. The goal of perception is to allow animals to recognize things through the process of assembling sensory information into a useful and reliable representation of the world. Natural images contain statistical regularities which distinguish objects from each other and from random noise. For successful visual recognition, each object must have attributes that can be used to differentiate it from others and, based on that, to segment the whole image into meaningful objects. How does the brain represent a visual stimulus internally? Let's briefly look at the human visual system.

Visual Stimulus Representation

Biological background

The human brain consists of complex-structured cells called neurons. The neuron receives inputs from other neurons through the dendrites. If it is sufficiently activated by the inputs, it produces voltage spikes called action potentials, which travel along the axons. At the end of the axons, these voltage spikes can be passed to the dendrites of other neurons through connections called synapses. The elaborate branching structure of the dendrite tree allows a neuron to receive inputs from many other neurons through synaptic connections.

The brain contains around 10^{11} neurons, which have more than 10^{15} synapses and at least 2000 miles of axonal connections [18]. Fortunately, there are several rules of organization that can simplify the explanation of how the brain works. First, neurons with similar patterns of connections and response properties are clustered together to form areas which seem specialized to perform different tasks. Second, many of these different areas themselves are subdivided into smaller processing units such as columns, a common feature within the visual system. Finally, there is a duplication of visual areas on either side of brain, a feature called lateralization which allows the brain to carry out a greater variety of tasks with a limited amount of brain tissue.

The human visual system includes the eye, particularly the retina, the lateral geniculate nucleus (LGN), and the cortical visual areas at the back of the brain [19]. The light from target stimuli is caught and focused onto the retina at the back of the eye. This pattern of light is then transformed by the light-sensitive photoreceptors into a pattern of neural activity that represents the image. The photoreceptors form synapses with bipolar cells which in turn synapse onto ganglion cells.

Each ganglion cell is connected to a number of photoreceptors via bipolar cells and its activity is altered by the stimulation of the retinal area corresponding to these photoreceptors. This retinal area is called the ganglion cell’s receptive field and is arranged in center-surround organization, either ON-center, OFF-surround cells or OFF-center, ON-surround cells. The mathematic model for the receptive field is convolution.

There are two types of photoreceptors: the cones and the rods. The cones are concentrated in the fovea, where the highest resolution is achieved. They mediate diurnal visual function and provide high-acuity color vision. The rods are concentrated in the periphery. They mediate nocturnal vision and provide only low-acuity monochrome vision. For color vision to be possible in humans, there are three types of cones: the blue, red and green cones. The responses from the three different cones are compared to allow color discrimination.

The axons of the ganglion cells form the optic nerves that pass out of eye to the brain at the optic disk, join together and cross over each other at the optic chiasm, and then project to the dorsal *LGN* on both sides of the brain. The LGN neurons project mainly to the first cortical visual area, called the primary visual cortex, striate cortex or V1, which lies posteriorly in the occipital lobe. Though the topography of the visual field is preserved in the projection from the LGN to V1, it is a much larger and more complex structure than the LGN. Having around 200 million neurons, V1 is made up neurons of two broad categories: simple neurons and complex neurons. In addition, there is also a class of cells that has a concentric center-surround receptive field similar to that of the LGN and the retinal ganglion cells.

Simple cells receive direct inputs from LGN. Their receptive fields, no longer concentrically arranged, are an integration of those of ganglion cells used to detect the orientation of the edges. One type of simple cell has a receptive field that contains an extended narrow central portion flanked by two antagonistic areas. The center can be either excitatory or inhibitory. Such cells have an optimal activation by a bar of light that is no more than a certain width, but entirely fills the center area, and is oriented at a certain angle. The symmetry of the receptive field as well as the preferred orientation of light and dark bar varies with different cells. Another type of simple cell has a receptive field that may just consist of two longitudinal regions facing each other, one excitatory and one inhibitory. There is also a situation where the cell is sensitive to the length of the stimulating bar. For such a case, there seems to be an additional antagonistic area at the top or bottom of the receptive field, and the optimal stimulus is an appropriately oriented bar or edge that stops in a particular place. These cells are named end-inhibited or end-stopped cells. Mathematically, the receptive fields of the simple cells have the kinds of configurations similar to Gabor Wavelets. Complex cells are like simple cells and require a specific orientation of a dark-light boundary. But there are no longer distinct excitatory and inhibitory areas. Being an idealized version of cortical organization, the cells in V1 are organized into repeating columns and processing modules. Each module contains neurons sensitive to wavelength, movement, and line or edge of particular orientation within a specific portion of the visual field.

Regions of cortex adjacent to V1 are also concerned with vision. The main target of V1 is Visual Area 2 (V2), which immediately surrounds V1. It is from V2 that two different visual pathways depart from each other and project to different cortical areas. One visual pathway, which, called the “what” pathway (also called the ventral system), is concerned with the identification of an object such as color and form, projects to visual area 4 (V4), then to the temporal lobes, and finally to the ventrolateral prefrontal cortex. The other pathway, which, called the “where” pathway (also called the dorsal system), is concerned with the relative spatial position of an object, projects to visual area 3 (V3) and visual area 5 (V5), then to the parietal lobes, and finally to the dorsolateral prefrontal region (DL). The cortex of the occipital, parietal, and temporal lobes largely supports perception and perceptual memory, while the cortex of the frontal lobe supports action and motor memory.

As we move up the object-processing pathways, the receptive field of a cell gets significantly larger and the cells also respond to more and more complex stimuli. Most of these cells also have receptive fields close to, or including, the fovea (75% of anterior inferior temporal (IT) cells include the fovea). The increase in receptive field size makes possible the development of a visual response that is unaffected by the size and position of a stimulus within the visual field. Believed to be essential for visual object recognition, the final purely visual stage of the ‘what’ pathway is identified by area TE in the middle and inferior temporal gyri. For their maximal activation, most cells in TE require moderately complex features such as moderately complex shapes or combinations of such shapes together with orientation, color, or simple texture to represent the image of a natural object. To represent different features contained in the image, the activation of multiple cells should be combined. Though not complex enough on their own to represent the image of a natural object, simple features such as orientation, color, or simple texture, which are extracted and represented by cells in V1, can be regarded as a set of immutable, universal primitives that are rich enough to represent all visual objects. When a novel stimulus is encountered for the first time, a particular combination of active cells will be reorganized to denote this new object and will be activated for subsequent encounters of the same object. Though larger than those of cells in the earlier stages along the ventral visual pathway, the receptive fields of the TE cells are smaller than the largest receptive fields found in the dorsal visual pathway which is mainly for the process of spatial information [3]. Apparently, the large receptive fields of the TE cells indicate a possible segmentation of the neural image into non-overlapping objects. In the following, some related work on more advanced segmentation techniques in computer vision is presented.

Related work

There are many methods available for feature binding and sensory segmentation in practice. One possible approach is to combine an edge detection approach and an edge-linking approach. Currently, the most competitive approaches for image segmentation are formulated as statistical neural network models for probabilistic grouping of distributional and histogram data. By identifying groups of similar image primitives, the inference essentially amounts to finding local minima of some objective energy function, corresponding to stable states. Here the image primitives can be local features that are spatially smoothed in image patches and can be mathematically represented as vectors in a metric space, giving rise to the feature space. Then, the observed data can be explained through prototypical distributions on the feature space.

In SEEMORE [20], a pure feedforward, receptive-field-based recognition model was designed to examine more directly the plausibility of the feature-space account. In this biological model for rapid, general purpose object recognition, an internal representation of a view of an object was encoded by a set of 102 feature channels, including 23 color channels, 11 coarse-scale intensity corner channels, 12 intensity “blob” features (parameterized by the shape (round and elongated) and size (small, medium, and large) of bright and dark intensity blobs), 24 contour-shape features (including straight angles, curve segments of varying radius, and parallel and oblique line combinations, and 16 shape- and texture-related features based on the outputs of Gabor functions at five scales and eight orientations). This feature space was implied to be represented by the activity of neurons at the top of the feedforward feature-extraction hierarchy. To measure the similarity of one object view to another, a simple (e.g., Euclidean) distance was computed. To recognize an input, a search was conducted for the most similar training view previously stored in memory.

The color histograms of multicolored objects were used in a technique called Histogram Intersection as stable object representations to provide a robust, efficient cue for indexing into a large database [21]. Later, the histogram clustering model (HCM) was proposed in [22]. To segment an image using HCM, the image is first decomposed into *N* not necessarily disjoint image patches, each having *M* features (or bins). For each image patch $i \in N$, a histogram $h_i = (h_{i,j})_{1 \le j \le M}$ is defined as a tuple $h_i \in N^M$. If n_i denotes the number of observations (e.g., pixels) belonging to image patch *i*, then, $n_i = \sum_j h_{i,j}$. The probability to observe features *j* in a given image patch *i* can be estimated by $p_i = (p(j \mid i))_{1 \le j \le M}$, where the empirical conditional probabilities are,

$$p(j \mid i) = \frac{h_{i,j}}{n_i} \tag{2.1}$$

Based on the feature vectors obtained from such a histogram, image segmentation can be realized by partitioning the set of image patches into a predefined number of disjoint cluster or segments. The model was successfully applied to segmentation based on texture analysis of images and its competitive performance was demonstrated in [23]. In [23], a filter bank of twelve Gabor filters with four orientation and three scales were used on each image patch. Each filter output was discretized into 16 equally sized bins. Based on the local distributions of Gabor coefficients, texture images were segmented.

Further, the HCM has also been integrated into neural network for image segmentation. For example, motivated by HCM, a network of leaky integrate-and-fire (IAF) neurons that encode segment assignments of image blocks was proposed to segment gray-scale images [24]. Alternative neural network architectures that follow a similar line of thought are locally excitatory, globally inhibitory oscillator networks (LEGION) consisting of locally coupled IAF neurons. Combined with a Markov-random-fields-based preprocessing, LEGION has also been applied for the purpose of texture segmentation [25].

As a different approach to feature binding, spin models have each feature represented by a spin variable which attains one of a discrete set of spin states. Two features can be bound when both share the same spin states. With regard to applications, the great advantage of spin models is that the derived energy or cost functions characterize the stable output states as their minima [26]. However, spin models lack biological plausibility.

Another model for perceptual grouping, the competitive layer model (CLM) was implemented in a standard additive recurrent neural network for feature binding and sensory segmentation [27]. The CLM is made of a set of layers of feature-selective neurons. The neurons at the same position on each layer form a column. Each column is associated with a particular feature which is described by a parameter vector. For example, for texture segmentation, a vector of local Gabor filter responses at different spatial frequencies and orientations can be employed as the feature vector. A binding between two features can be achieved by having each column assign its feature to one of the layers according to some energy-based rules. Within each layer, the neurons are coupled by lateral interaction. Salient segments are carried by those active layers.

Learning

Once a set of features is obtained, the recognition task reduces to partitioning the feature space. Each feature appears as a point in the feature space and patterns pertaining to different classes will fall into different regions in the feature space. Learning is this process of classifying a pattern into the right category. The pattern classification (or pattern recognition) system is expected to have the ability to improve the performance through iterative adjustment of the system parameters in the hope that after an iteration of the learning process the system will become a more knowledgeable and effective system and produce a higher recognition rate. Usually, the pattern recognition system will first undergo a training process, during which the system is repeatedly presented with a set of input patterns, the so-called training set (or prototypes). If each input pattern is given along with the category to which this particular pattern belongs, the learning is called supervised learning. Then the task left is to find separating boundaries based on these input-output pairs. The training process is determined to be completed when all the patterns in the training set have been correctly classified. Next the system is presented with new patterns that have not been seen before but belong to the same population of patterns used to train the system [28].

Supervised learning

For supervised learning, decision surfaces can formally be defined to be separating boundaries between different regions in the feature space. Patterns on the decision surface are of equal similarity to both regions separated by this decision surface. For n-dimensional pattern points, decision surfaces are $(n-1)$-dimensional. The decision surface between two classes, i, j, can be defined as

$$d_i(\vec{x}) - d_j(\vec{x}) = f_i(\vec{w},\vec{x}) - f_j(\vec{w},\vec{x}) = 0 \qquad (2.2)$$

where $d(\overset{\omega}{x}) = f(\overset{\omega}{w}, \overset{\omega}{x})$ is called the discriminant function, $\overset{\varpi}{x} = [x_1, x_2, \Lambda\ , x_n, 1]^T$ and $\overset{\varpi}{w} = [w_1, w_2, \Lambda\ , w_n, w_{n+1}]^T$ represent the augmented feature vector in pattern space and the weight vector in weight space, respectively. The weight space is an $(n+1)$-dimensional Euclidean space in which the coordinate variables are $w_1, w_2, \Lambda\ , w_n, w_{n+1}$. For example, if there are *M* classes and $\overset{\omega}{x}$ is in class *i*, then

$$d_i(\overset{\varpi}{x}) > d_j(\overset{\varpi}{x}) \quad \forall j \neq i, \quad j = 1, 2, \Lambda\ , M \tag{2.3}$$

The decision surface that separates classes *i* and *j* is then given by

$$d(\overset{\varpi}{x}) = d_i(\overset{\varpi}{x}) - d_j(\overset{\varpi}{x}) = 0 \tag{2.4}$$

A linear discriminant function for a class *k* will be of the following form

$$d_k(\overset{\varpi}{x}) = w_{k1}x_1 + w_{k2}x_2 + \Lambda\ + w_{kn}x_n + w_{k,n+1} = \overset{\varpi}{w}{}_k^T \cdot \overset{\varpi}{x} \tag{2.5}$$

A nonlinear discriminant function for a class *k* can be described as

$$d(\overset{\varpi}{x}) = w_1 f_1(\overset{\varpi}{x}) + w_2 f_2(\overset{\varpi}{x}) + \Lambda\ + w_M f_M(\overset{\varpi}{x}) + w_{M+1} \tag{2.6}$$

where $f_i(\overset{\varpi}{x}), i = 1, 2, \Lambda\ , M$ are linearly independent, real and single-valued functions which are independent of the w_i (weight).

The objective of training a system is actually to find the weight vectors with the a priori information obtained from the training samples. Initially, the weight vector is chosen as random small numbers. During the training period, patterns are presented one at a time through all prototypes (training patterns) and the weights are updated to minimize the errors between the target value and the computed value. Each complete pass through all the patterns is called an iteration. After one iteration, all patterns are presented again in the same sequence to carry on another iteration. This is repeated until no correction is made through one complete iteration. If the prototypes are linearly separable, the algorithms may converge to a correct weight vector $\overset{\omega}{w}$. Two commonly used supervised learning methods are statistical discriminant functions and neural network approaches. Detailed descriptions of these supervised learning methods are beyond the scope of this thesis but can be found in many pattern recognition books such as [28], and therefore will not be repeated here. In the following, only a very simple neural network, a perceptron, is presented.

Neural networks consist of a large number of simple processing units called artificial neurons as shown in Fig 2.1. The goal of neural networks is to process information in a way so as to map multiple inputs to single outputs. The basic operation of an artificial neuron is to apply an activation function to a weighted sum of input vector plus a bias. The most commonly used activation functions are the step function, the identity function, and the sigmoid function. Often, the activation values are restricted to the range [0,1].

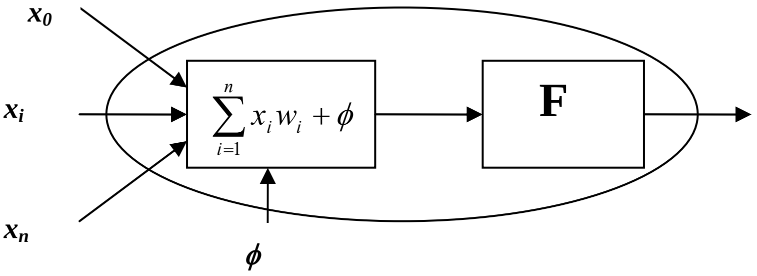


Fig 2.1. Artificial neuron

Perceptron neural network models [29] use a step activation function and are designed as a classifier for linearly separable input samples. The simplest perceptron network is one with a single artificial neuron and is useful for a two-class problem or a binary decision. The weights and bias of the artificial neuron define a linear surface that separates the two classes. At the beginning of the training session, small random values are used for the weights and the bias. Then an input vector is presented, and the output is calculated. If the actual output is the same as the desired output, stop and go to the next input vector. Otherwise, update the weights and bias as,

$$w_i(k+1) = w_i(k) + \alpha\varepsilon x_i \quad b(k+1) = b(k) + \alpha\varepsilon \tag{2.7}$$

where *k* represents the iteration number and $\varepsilon = \overline{o} - o$ represents the error between the desired output and the actual output. This update process is repeated for each input vector until there is no change in the weights or the maximum number of iterations is reached. For problems with more than two classes, the extension from a single neuron perceptron to a multi-neuron perceptron is obvious and weight vectors exist for each artificial neuron, forming a weight matrix for all the artificial neurons.

Unsupervised learning

For many other cases where there exists no a priori knowledge of categories into which the patterns are to be classified, the input patterns group themselves by natural association based on some properties in common. It is expected that the degree of natural association is high among members belonging to the same category and low among members of a different category according to some similarity measures. As a result, patterns belonging to the same cluster should be very close together in the pattern space, while patterns in different clusters should be further apart from one another. This learning process is called unsupervised learning. In unsupervised learning, there is no class labeling available, nor do we know how many classes there are within the input patterns.

To discover similarities and dissimilarities and to reveal the organization of patterns into "sensible" clusters for both supervised learning and unsupervised learning, a major issue is to define the "similarity" between two feature vectors and, after setting up an appropriate measure for it, to design an algorithm to search for similarities and dissimilarities among these patterns and then cluster the vectors on the basis of the adopted similarity measure.

It is desired that the similarity measure is given in numerical form to indicate the degree of resemblance between patterns in a group, between a pattern and a group of patterns, or between pattern groups. Many different functions have been suggested for this purpose with the most popular ones being the family of Minkowski-metrics. 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,

$$d_{ij} = \left(\sum_{k=1}^{K} \left| x_{ik} - x_{jk} \right|^{r} \right)^{1/r} \tag{2.8}$$

where $\overset{\varpi}{x}_i$ and $\overset{\varpi}{x}_j$ are two vectors in n-dimensional space.

Based on the similarity measure, it is expected that the degree of natural association is high among members belonging to the same category and low among members belonging to different categories. As a result, patterns belonging to the same cluster should be very close together in the pattern space, while patterns in different clusters should be further apart from one another. Therefore, unsupervised learning is also called clustering.

Clustering algorithms are usually classified into several broad methods: hierarchical methods, partitioning methods, density-based methods, grid-based methods, model-based methods, and graph theoretical methods. The hierarchical methods can be classified as being either agglomerative (bottom-up), which starts with each feature vector being a single group followed by merging, or divisive (top-down), which starts with all the objects in the same cluster followed by splitting. Typical examples include BIRCH (Balanced Iterative Reducing and Clustering using Hierarchies) [30], CURE (Clustering Using Representatives) [31], ROCK [32]. Partitioning methods attempt to directly decompose data into disjoint clusters and work well for finding spherical-shaped clusters. Typical examples include the famous *K*-means algorithm [33], PAM (Partitioning Around Medoids) and CLARA (Clustering LARge Applications) [34], CLARANS (Clustering Large Application based on RANdomised Search) [35], and *K*-mode algorithm [33]. Density-based methods group neighboring objects into clusters based on neighborhood density conditions. Typical examples include DENCLUE (DENsity-based CLUstering) [36], DBSCAN (Density-Based Spatial Clustering of Application with Noise) [37], and OPTICS (Ordering Points To Identify the Clustering Structure) [38]. Grid-based methods quantize the space into a finite number of the cells that form a grid structure, and then perform clustering on the grid structure. Typical examples include STING (Statistical Information Grid-based method) [39], and WaveCluster [40]. Model-based methods hypothesize a model for each of the clusters and then find the best fit of the data to that model. Typical examples include statistical approaches such as COBWEB, CLASSIT, AutoClass and neural network approaches such as Competitive learning and Self-organizing maps [29]. Finally, graph theoretic approaches are suggested for all the samples to be considered simultaneously so as to overcome the shortcoming that the clustering results are dependent on the presentation ordering of the pattern samples, with the possibility of an increase in computational time and a substantial cost in rapid-access storage [28]. Typical examples include similarity matrix, spanning tree methods, shared nearest neighbor maximal spanning tree for clustering, and graph theoretic clustering based on limited neighborhood sets, and CHAMELEON [41]. Similar to the supervised case, network approaches exist for unsupervised learning, including Kohonen self-organizing, competitive learning, the adaptive resonance theory (ART), neocognitron models, and modified competitive learning, and the popular recurrent Hopfield network [42].

Partitioning methods try to find representative points (median or medoids) so as to minimize the sum of distances of points from their closest medoid. CURE was proposed to remedy the drawback of using a single centroid/medoid to represent a cluster by using a constant number of such representative points and shrinking them towards the centroid/mediod, which helps CURE in avoiding the problem of noise and outliers. Based on these ideas, in this paper, the minimum spanning tree method is used first to find the substantial 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 to build up the database.

Minimum Spanning Tree

The minimum spanning tree method is a graph analysis of arbitrary point sets of data. In a graph, two points can be connected by either a direct edge or a sequence of edges called a path. A loop in a graph is a closed path. A connected graph has one or more paths between any pair of points. A tree is a connected graph without closed loops. A spanning tree is a tree that contains every point in the data set. If a value is assigned to each edge in the tree, the tree is called a weighted tree. For example, the weights for each edge can be the distance between the two points. The weight of a tree is the total sum of edge weights in the tree. The minimum spanning tree (MST) is the spanning tree that has the minimal total weight among all possible spanning trees for the data set. The minimum spanning tree has the following property that can be used for clustering if the weight associated with each edge denotes the distance between the two points. That is, the weight associated with every edge in the minimum spanning tree will be the shortest distance between two subtrees that are connected by that edge. Therefore, removal of the longest edge will theoretically result in a two-cluster grouping. Removal of the next longest edge will result in a three-cluster grouping, and so on. These correspond to choosing breaks where maximum weights occur in the sorted edges. Two greedy algorithms are usually used to build the minimum spanning tree: Kruskal's algorithm and Prim's algorithm [33]. In this research, Prim's algorithm is used. The construction of the minimum spanning tree starts with some root node s and the tree T greedily grows from s outward. At each step, among all the edges between the nodes in the tree T and those not in the tree yet, the node and the edge associated with the smallest weight to the tree is added to the tree. This is an $O(N^2)$ problem. When the tree is built, after sorting the edge in decreasing order, the edges can be cut to form clusters.

Continue Learning & Novelty Detection

Novelty is defined as changes that occur in the environment. Due to our brain's persistent interest in novelty, the human species can succeed in this world. Therefore, the ability to continue learning new or novel objects is important in any signal classification scheme and novelty detection is considered as a key component of a machine learning system which can only be trained on known data. Novelty detection has a variety of applications, including fault detection, radar target detection, detection of masses in mammograms, hand written digit recognition, internet and e-commerce, statistical process control, etc., and has been regarded as an extremely challenging task since there is no single best model for every situation. Typical solutions to this problem aim to model normal data and to determine the abnormality by using a distance measure and a threshold. No matter whether a global novelty threshold or multiple thresholds (one for each class) are used, there is always a trade-off between the error rate and the reject rate, i.e., the proportion of data rejected. As a result, the performance level of a pattern recognition system on novelty detection tasks is commonly evaluated by both. Two broad categories of techniques used for novelty detection are statistical approaches and neural network approaches. In statistical approaches, the statistical properties of the data are employed to generate the model, which can then be used to estimate whether a test sample belongs to the same distribution or not. Of the statistical approaches, there are two main modeling techniques: parametric models, which assume that the data come from a family of known distributions, and nonparametric models, in which no assumption on the form of the data distribution is made. The former are simple but may have limited practical importance in many real world situations, while the latter are more flexible but more computationally expensive. Using robust statistics for data density estimation, examples of the advanced parametric statistical novelty detection methods include probabilistic Gaussian mixture modeling (GMM) approaches, hidden Markov modeling (HMM) approaches, and hypothesis testing. In a GMM, the *k*-means algorithm is employed to estimate the parameters, while in an HMM, the expectation-maximization (EM) algorithm is used to estimate the parameters. Typical nonparametric novelty detection methods are *k*-nearest neighbor based density estimation, Parzen density estimation and string matching approaches. In *k*-nearest neighbor approaches, the categories of the *k*-nearest neighbors in the training data of a test pattern are used to weight the category candidates together with some predefined threshold distance. In the Parzen approach, both the density of the training data and the rejecting pattern are estimated to achieve novelty detection. In string matching methods, the training data are treated as templates represented by strings and then the novelty detection is implemented by computing some measure of dissimilarity between the training and testing data. Detailed description of these algorithms is beyond the scope of this thesis but can be found in [43] and will not be repeated here.

As an alternative, neural networks have been widely used for novelty detection and have the advantages of demanding a small number of parameters for optimization and making no a priori assumptions on the properties of the data. However, neural networks have increased computational complexity and can't be as easily retrained as statistical models. The major issue is how to learn the new units with computational efficiency in both time and space. There are two main techniques for tackling this problem: pruning algorithms, which begin with a large network size and delete inactive units during training, and constructive algorithms, which begin with a small set of units and add new ones to satisfy the needs. The second strategy is more useful since the computational efficiency can be improved by freezing the units already existing and updating only the new units in the iterative training procedure. As the best known and most widely used class of neural networks, multi-layer perceptrons (MLP) usually detect novelty by thresholding either the Euclidean distance between the output pattern and the target pattern or the highest output value. Support vector machines (SVM) based approaches and Adaptive Resonance Theory (ART) approaches use hyper-spheres to surround the training classes and try to produce closed decision boundaries. After training, a test pattern is deemed to be novel if it is outside of the hyper-spheres. Other neural network approaches include Radial basis function (RBF) networks, auto-associator approaches, Hopfield networks, oscillatory neural networks, self-organizing map (SOM) based approaches and habituation based approaches. A good survey of these neural network approaches can be found in [44] and will not be described here. In summary, after finishing identifying novel samples, they should then be accumulated and clustered to see whether they are just outliers of known classes or whether they represent truly novel patterns.

CHAPTER III

PERCEPTUAL LEARNING

As Sperry pointed out long ago, the sensory systems evolved to improve the performance of the motor system, but not the other way around [45]. That is to say, behavior is not always determined by sensory stimulation. For example, when two sources of sensory input try to attract the subject in different directions, in order to function effectively, the subject must be able to ignore all stimuli except the one it regards as the most important. This sort of dilemma happens all the time. In other words, if several stimuli are present, the subject may have to choose between possible courses of action and, when one has been decided on, must prevent any irrelevant input from interfering with the behavior. It is in this way that attention comes into play.

Attention is a process that highlights a particular stimulus. It selects the items relevant to the current task from the vast array of sensory input and determines which associations of those items should follow. For an explanation of how attention identifies and targets the relevant stimuli, the sensitization of the appropriate sensory pathways is an essential component of any response. At the beginning, a goal to perform some action comes first, which exerts a major influence on the type of stimulus to which the sensory system is attuned. Next, after the goal has been established, attention is focused on stimuli needed to proceed with the task. For example, when a two-year-old baby finds some new toy which is out of his reach, he may have the desire in the first place to approach it. Subsequently, the location of the toy guides the direction of his movement. At the center of the model, a goal must be established, often felt as a need, usually in the form of internal stimuli, which will be satisfied by external sources, other stimuli often called reinforcers. During learning, the goal will evolve to activate responses that are most likely to reduce the need. Therefore, in this model, it is obvious that the stimuli play a dual role, serving to motivate sometimes while to inform and guide responses at other times.

Since any basic behavior defines a category of action in response to sensory input – a stimulus-response category, perception and action are tightly coupled. This coupling was given a name, sensory-motor coordination, by American philosopher, John Dewey. In the process of explaining human behavior, as a relatively recent evolutionary development from the accumulated experiences of sensory-motor coordination, learning plays a major part and allows animals to transfer the motivational properties of innately reinforcing stimuli to accompanying or perceptibly related stimuli. To survive, animals must recognize sources of nourishment and dangerous situations, and respond accordingly by approaching the former and staying away from the latter. In other words, learning makes attention initially directed only to an innate reinforcer directed to objects associated with the reinforcer. For example, when you are hungry and out of your house, you will look for a restaurant. When you finish eating in the restaurant and want to get out, you will look for the doors. It is learning that allows initially neutral objects to arouse a motivational state and links the motivation system to perceptual systems. As another example, when you see wall, you will stop going toward it and turn to open space. For higher animals such as humans, evidence suggests that the occurrence of a response depends on its predicted outcome and behaviors are planned on the basis of future possibilities rather than present contingencies alone [46][47]. If the predicted outcome is a reward, the intended response in the form of a behavior is facilitated. On the other hand, if an inverse outcome is predicted, the response is strongly inhibited.

Based on the same principle, the robot learning problem addresses the question of making a robot perform certain tasks in the world successfully. As a result, every learning process has some "goal", which can be most precisely defined in terms of perceived states of the world, i.e., the sensor states, obtained by transforming the true state of the world. Suppose G is an identifiable goal state, X and Y are two different perceived states of the world, the learning problem can be formally described by,

$$G : X \to Y : R \tag{3.1}$$

which can be interpreted as, suppose the robot finds itself in a state satisfying condition *X*, if, in reaching the goal *G*, a state satisfying condition *Y* becomes active, a reward *R* is received [48]. If a task can be defined in term of a set of such reward receiving goals, a qualitative measure of robot performance can be the sum of the rewards it receives over time. Given this definition of robot performance, the robot learning problem is to improve robot performance through experience. Since this type of situation is common in robotics, reinforcement learning may be applied to many robotics tasks.

Therefore, for accurate assessment of progress toward reaching a goal, a wide range of different types of knowledge in terms of sensor states and internal states are required. Then the question becomes where these types of knowledge are retained and for what purpose are they stored. In cognitive psychology, according to their functionality, different kinds of memory, such as working memory (short-term memory), episodic memory, and long-term memory, have been proposed for temporal or permanent information storage on the process of personal goals. In the following, an introduction to these types of memory is given.

Working Memory

Psychological model

Developed from an earlier concept of considering short-term memory to be a unitary temporary storage system typified by the model of Atkinson and Shiffrin [49], Baddeley and Hitch proposed the famous three-component WM model. This model accounts for a limited capacity system for temporarily storing and manipulating of information used to control an ongoing behavior. In the original Baddeley and Hitch model, there are three components, the ‘central executive’ aided by the ‘phonological loop’ and the ‘visuospatial sketchpad’. The central executive, as an attentional control system, processes the information supplied by the other two slave systems to attain a goal. The phonological loop is assumed to hold verbal and acoustic information while the sketchpad is to hold visuospatial information [50]. Recently, an ‘episodic buffer' has been proposed and added as a mechanism for combining information from various subsystems into a form of temporary representation. As a limited-capacity temporary storage system controlled by the central executive, it also offers a possible solution to the binding problem and the role of consciousness, and serves as an interface between the working memory and Tulving's episodic memory [51]. Where is the working memory located in the brain? Let's continue with some biological background next.

Biological background

Anatomically, the prefrontal cortex is an important region for short-term working memory. As mentioned in Chapter II, starting from the retinal, there are two visual pathways: the 'what' pathway and the 'where' pathway. The "what" pathway (also called the ventral system) takes care of the identification of an object while the "where" pathway (also called the dorsal system) focuses on the relative spatial position of an object. The 'what' system projects to the inferior convexity (IC) ventrolateral to the principal sulus, and the 'where' system projects to the dorsolateral prefrontal region (DL). Experimental results suggest that IC mediates working memory for objects and DL mediates spatial working memory. Although the visual system is composed of a vast number of neurons, its ability to process fully and store distinct and independent objects is strictly limited. It has been suggested that objects must compete for attention and processing 'space' in the visual system, and that this competition is influenced by both automatic and cognitive factors [52].

Now we move from the posterior part, which is mainly sensory, to the anterior part, which is mainly motory. In contrast to the posterior cortex being 'sensor' cortex, the entirety of the frontal cortex, in general, is 'motor' cortex devoted to organismic action, such as skeletal movement of various body parts, ocular movement, the expression of emotion, speech, or visceral control and so on, together with considerable functional cooperation between them. The cortex of the occipital, parietal, and temporal lobes largely supports perception and perceptual memory, while the cortex of the frontal lobe supports action and motor memory. Substantial evidence is available that the memory storage and perception processing are functions of large networks of interconnected neurons of the posterior cortex. Particularly, though by mechanisms still poorly understood, the hippocampus seems to play an essential role in the processing of memory formation. Being made of neurons representing the various inputs that have become associated with that memory, a network represents a memory or an item of knowledge. Since perception and memory largely share the same network and perceptions are processed through pre-established memory networks, they form the neural foundations for the intimate relationship between perception and memory. We see the world the way we know it, the way we have learned to see it [53][54][55]. At the root of it, the sensory information is processed through hierarchically ordered areas of the posterior cortex and the neuronal networks (memories) they harbor.

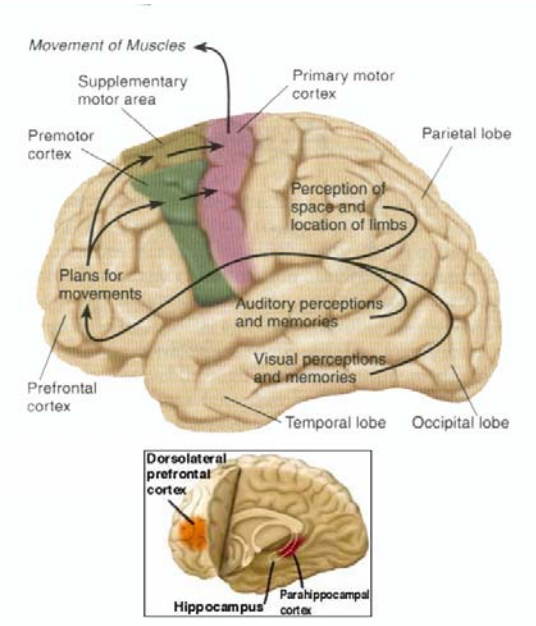


Fig 3.1 Prefrontal Cortex and Hippocampus

Similar to the posterior cortical areas, the frontal areas are hierarchically organized as well for representation and processing of action. At the lowest stage of this frontal hierarchy is the primary motor cortex. Above it are the premotor areas, and above these, in the frontal motor hierarchy, are the dorsolateral areas of the prefrontal cortex. Like perceptual memory, motor memory is also hierarchical but is separated into different action domains, each providing the basic neural substrates for the representation and processing of a different category of movement.

In all instances, prefrontal cortex serves the purpose of ordering sequential actions toward a goal, whatever that goal may be, and plays a fundamental role in the temporal organization of behavior. In close cooperation with other brain structures, it provides neural support to three cognitive functions that are indispensable for conducting temporal sequential actions: (1) short-term memory (working memory), (2) motor attention or set, and (3) inhibitory control of interference [56].

Working memory is memory in the active state that the organism needs and uses for the performance of acts in short term. The behavior of working memory goes beyond simple stimulus-response reflexes and is typified by delayed-response tasks, which was introduced ninety years ago by W.S. Hunter [57]. In the various forms of delayed-response tasks, the instruction cue disappears before the trigger appears in a given trial. The sensory stimulus and motor response are separated by a brief delay period, during which the information contained in the instruction needs to be kept in this short-term memory until it can be used for the reaction. Thus, the prefrontal cortex will sustain activation of neuronal populations during the active retention of sensory information, as during the delay of a delayed response task. In the process of reaching a goal, the old content of this short-term memory can be given renewed timelines and terms by the action at hand, which is realized by reinforcement in the form of either reward or punishment. When the goal is attained, the content of this short-term memory has outlived its usefulness and must be discarded since its term has expired.

The whole cerebral cortex projects to the basal ganglia, which have classically been considered as primarily motor structures. The outputs of the basal ganglia then project back to the frontal cortical areas, to the primary motor cortex, to some non-cortical motor systems, and to dopaminergic systems in the midbrain. This last pathway is important since the nucleus there contains dopaminergic neurons that project to both the basal ganglia itself and the frontal areas, and play an important role in the learning of behavior [58]. A biological correlate for reward learning may be found in the activity of dopamine neurons. They respond rather uniformly to unexpected rewards which are particularly effective for learning. The selected action is maintained by the reinforcer. Being much more focused on goal-directed learning from interaction than are other approaches to machine learning, reinforcement learning is the basis of neurocomputational models of working memory.

The second function of the prefrontal cortex is motor attention, which is conventionally understood as the preparation for action or attention directed to prospective action. While the content of working memory is mainly sensory, the content of motor attention is mainly motory. Being complementary to each other, they help mediate cross-temporal contingencies. The third function of the prefrontal cortex, the inhibitory control of interference, is integrative and protects the structure of behavior, speech, or thought from interfering influences, external or internal, that may conflict with it and lead it astray.

Episodic Memory and Conjunctive Coding

Though working memory is important, it is beyond its ability to fuse co-occurring stimuli into configurations of these stimuli as temporary representations, upon which individuals can remember their personal past, and to consciously recollect experienced events and other happenings in subjectively apprehended space and time. As proposed some thirty years ago by Tulving, episodic memory refers to a neurocognitive system that enables humans to experience past events in their minds [59]. It is about the distributed representation of contextual information such as stimulus features or internal states in particular places at a particular time, i.e., about "what" in "where" at "when" [60]. The duration of episodic memory, typically a few minutes or a few hours, makes it separate from the conceptions of working memory [51]. But it is closely related to the 'episodic buffer' mentioned above, where inputs from different modalities are bound with information retrieved from different areas of long-term memory into a neural representation that remains in the working memory for a period of time (no more than a few minutes). The outputs of the episodic buffer are thought to retain event specific knowledge. Therefore, functionally, episodic memories can be regarded as short-duration neural representations which selectively retain representations of states of the episodic buffer, and thus summarize the previous states of the working memory so as to keep track of the marked changes in the contents of conscious progress towards the active goals. The purpose of retaining such records of recent states of consciousness in episodic memory over a period of minutes and hours is to supply highly specific information on recent plan execution and to provide a constant evaluation of progress on current goals. This is needed because, given the heavy storage cost of the detailed content of working memory generated in a short period, long-term retention is infeasible. Episodic memory retains only sensory-perceptual summaries coming out of working memory and abstract knowledge that relates to very specific goals. In other words, episodic memory contains both abstract-conceptual and sensory-perceptual knowledge. It is possible that the former functions to organize the latter into coherent chunks of experience. It is believed to provide a link from working memory to long-term memory in such a way that momentary experience may be retained and later incorporated into consciously formed memory. Though Tulving and colleagues proposed that the right PFC was specialized for retrieval and the left for encoding, the hippocampus and associated structures are long known to be involved in the recall of recent memories, i.e., episodic memory. In humans, the structures of the hippocampal region are located in the medial temporal lobes.

A scene in which events occur can provide a place, space, or context for experience. The stimulus features that make up this context can be represented in two ways, i.e., the feature representation where the scene is represented as a set of independent individual features associated with the scene, and the conjunctive representation where the scene is represented as a set of features combined into a unitary representation that encodes their conjunction. In other words, the feature representation makes a discrete cue associated with a response while the conjunctive representation makes a single representation formed from the combination of multiple cues associated with a response and different configurations of stimuli have different responses. It has been suggested (and there is sufficient evidence) that neocortical systems can represent the independent individual features of an environment whereas the elaboration of features into a conjunctive representation of a place or a location involves both cortex and the hippocampus [61][62]. More specifically, the hippocampus uses compound stimuli, constructed through conjunctive representation, to control performance [63].

Also referred to as unitary representation and configural association, the conjunctive representation can have advantages over the feature representation by solving nonlinear discrimination problems that require the construction of conjunctive representations. For example, in a bi-conditional discrimination task of form AB+, CD+, AC-, BD-, a reward is issued for responding in the presence of the AB and CD compounds and a punishment is issued for responding in the presence of the AC and BD compounds. However, since each element (A, B, C, D) is equally associated with rewarded and punished trial outcomes, their linear combinations will end up with equal associative strengths across the compounds. To solve this problem, conjunctive representations of the compounds should uniquely be associated with their respective trial outcomes. For another example, when two events have overlapping features, such as ABCDE and ABCEF, if only the single feature representation scheme is used, there will be potential interference. However, if each combination has a unique representation, there will be reduced interference. Therefore, conjunctive representation can produce pattern-separation for similar experiences or input patterns [64].

Now that the mechanism by which working memory in the frontal cortex uses to control behavior is made clear, the important question is how to model the mechanism and implement it for the benefit of robots.

Working Memory Toolkit (WMtk)

Dr. David Noelle and his PhD student, Joshua Phillips, at Vanderbilt University, created a set of software tools, called the NSF ITR Robot-PFC Working Memory Toolkit (WMtk), for developing working memory systems that can be easily and tightly integrated into robotic control system to perform goal-directed delayed-response tasks [65]. This working memory system learns to identify informational chunks worthy of retention using a model of the interactions between the brain's dopamine (DA) system and the PFC. The neuro-computational model for working memory is based on reinforcement learning, in which learning about stimuli or actions is solely based on the basis of rewards and punishments associated with them [66]. 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. 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. There are two broad classes of reinforcement learning. In the first case, the reinforcement is delivered immediately after the action is taken. This makes learning relatively easy. In the second case, 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. However, this latter case is what we are most interested in. In the following, the temporal difference (TD) learning algorithm, which is actually used in the WMtk, is given. A more complete description of other types of reinforcement learning can be found in [9].

TD learning

As a concise account of certain aspects of classical conditioning, the Rescorla-Wagner rule [67] is based on a simple linear prediction of the reward associated with a stimulus. When a stimulus, denoted by a binary variable *u*, is presented, the expected reward, denoted by *v*, is expressed as this stimulus variable multiplied by a weight *w*,

$$v = wu \tag{3.2}$$

To establish the value of the weight, *w*, a learning rule is designed to minimize the expected squared error, $(r - v)^2$ between the actual reward *r* and the predicted value *v*. The Rescorla-Wagner rule, as a trial-by-trial learning rule, is used to minimize this error,

$$w \rightarrow w + \varepsilon \delta u \tag{3.3}$$

where $\delta = r - v$ is the crucial prediction error, and ε is the learning rate, which can be interpreted in psychological terms as the associability of the stimulus with the reward. The activity of dopaminergic cells in the ventral tegmental area (VTA) is believed to encode a form of this prediction error. For generalization of this rule to cases in which more than one stimulus is used to associate with a reward and the expected reward is the sum of each stimulus parameter multiplied by its corresponding weight, the Rescorla-Wagner rule can be extended to have a vector format,

$$v = \vec{w}^{T} \vec{u} \tag{3.4}$$

$$\vec{w} \rightarrow \vec{w} + \varepsilon \delta \vec{u} \tag{3.5}$$

Not only can animals associate stimuli with rewards and punishments, they also can predict the future time within a trial at which reinforcement will be delivered. Such predictions can then be used to optimize behavior when rewards are delayed. It is in this sense that the discrete time variable $v(t)$ can be interpreted as a prediction of the total future reward expected from time t onward to the end of trial, that is,

$$\sum_{\tau=0}^{T-t} r(t+\tau) \tag{3.6}$$

Since this quantity summarizes the total expected worth of the current state, it can be used for optimization. Additionally, $v(t)$ can be approximated by a discrete time linear filter. For the case of a single time-dependent stimulus $u(t)$, that is,

$$v(t) = \sum_{\tau=0}^{t} w(\tau) u(t-\tau) \tag{3.7}$$

A simple modification for Eq 3.3 would appear to be,

$$w(\tau) \to w(\tau) + \varepsilon \delta(t) u(t-\tau) \tag{3.8}$$

where $\delta(t) = \sum_{\tau} r(t+\tau) - v(t)$ is the difference between the actual and predicted total future reward. Since the total future rewards on a given trial, $r(t+1), r(t+2), \Lambda, r(T)$, are unknown, a possible solution can be obtained by the recursive formula

$$\sum_{\tau=0}^{T-t} r(t+\tau) = r(t) + \sum_{\tau=0}^{T-t-1} r(t+1+\tau) \approx r(t) + v(t+1) \tag{3.9}$$

where $v(t+1)$ provides an approximation of the average value of the $\sum_{\tau=0}^{T-t} r(t+1+\tau)$. Approximating $\delta(t) = \sum_{\tau} r(t+\tau) - v(t)$ by $\delta(t) = r(t) + v(t+1) - v(t)$ results in the temporal difference (TD) learning rule [65],

$$w(\tau) \to w(\tau) + \varepsilon \delta(t) u(t-\tau) \text{ with } \delta(t) = r(t) + v(t+1) - v(t) \tag{3.10}$$

The name of the rule comes from the fact that $v(t+1) - v(t)$ is the difference between two successive estimates. $\delta(t)$ is usually called the temporal difference. A neural system called the adaptive critic learns to predict the expected future reward $v(t+1)$ given features of the current situation [68].

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.

WMtk interface

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. More specifically, as a wrapper for the input information, the Chunk class is basically a simple data structure that takes a pointer to some memory location and a string as arguments and stores some arbitrary piece of information that is believed by the user to be useful for processing by the working memory. The pointer points to any data structure the user may define. The string allows the user to store a basic amount of type information along with the pointer so that the pointer can be cast to its appropriate type later when needed. The input chunk could be a coordinate, an angle, a plan, an action, or any other “chunk” of information. Being biologically inspired, the working memory system utilizes a neural network to make decisions about memory management. As a result, like most neural networks, when presented to the working memory, the chunks are organized in the form of a feature vector. This vector is wrapped in a FeatureVector class, which also provides some extra functionality for filling the vector with values. It is an important class and provides the interface between the user and the working memory system. On the one side, the user provides Chunk and State information to the neural network. On the other side, the working memory system makes intelligent decisions about memory management, i.e., what to keep in working memory. Finally, as the most complex class in the toolkit, the WorkingMemory class is the workhorse of the WMtk. It is a limited store and supplies all of the functionality thought to pertain to working memory in the system. 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). In order for it to operate correctly, the required inputs to the WorkingMemory class include the size of the working memory, the size of the chunk vector (i.e., the size of the feature vector), the size of a state vector which defines the set of possible states the system will be in, the current state the system is actually in, the user-defined reward function, the user-defined state function, the user-defined chunk assignment function, and the user-defined chunk delete function (so as to release the computer memory when no longer needed).

Conjunctive coding

When the working memory size is larger than 1, the toolkit creates a conjunctive code of all combinations of the chunk vectors in working memory to mimic the conjunctive representation. The size of the network weight file will be:

$$\text{State\_Vector\_Size} + ((\text{Chunk\_Vector\_Size} + 1)\text{\^{}WM\_size}) + \text{Bias\_Unit} \qquad (3.11)$$

where the first State_Vector_Size entries in the weights file will be associated with the state vector only, the next ((Chunk_Vector_Size + 1)^WM_size) entries will correspond to a combination of chunks in working memory, and, finally, there is one more entry for the Bias_Unit weight, Bias_Unit = 1. The extra element that gets added to each Chunk_Vector is set to 1 if the WM slot is empty and 0 if a chunk is loaded in the slot. Each weight in the ((Chunk_Vector_Size + 1)^WM_size) portion of the file will contain the value of a particular combination of WM contents. With only a single WM slot, this equation simplifies to a single entry for each element associated with the chunk vector encoding. With a larger WM_size, the network weights don't read out so clearly. For example, if we have an experiment with three different kinds of chunks, the Chunk_Vector_Size will be 3. The chunk type #1 is then encoded (in binary) as 100, chunk type #2 as 010, and chunk type #3 as 001. Further, if the working memory of size 2, this makes for a conjunctive coding of size 16. Therefore, there will be totally 16 combinations of chunks as

$$0123 \times 0123 = \begin{bmatrix} 00 & 01 & 02 & 03 \\ 10 & 11 & 12 & 13 \\ 20 & 21 & 22 & 23 \\ 30 & 31 & 32 & 33 \end{bmatrix}$$

If there is one chunk of type #2 (in slot 1) and one chunk of type #3 (in slot 2), this vector will be filled like so:

$$0100 \times 0010 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

If there is one chunk of type #1 (in slot 1) and the second slot is empty (the empty slot is automatically given an encoding of 0001):

$$1000 \times 0001 = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

A completely empty memory looks like:

$$0001 \times 0001 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Two chunks of type #2 in WM looks like so:

$$0100 \times 0100 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

CHAPTER IV

AUTONOMOUS NAVIGATION

The key word, autonomous, used in the title reflects upon the development of robotics systems that are more and more based on somewhat human-like attributes such as perception and cognition under the requirement that those systems operate in natural environments and perform goal-directed tasks successfully with minimal supervisory communication from a human. As we move deeper into the area of autonomous robotic systems, major areas of robotic research include the understanding of how humans robustly perform tasks such as vision, perception, cognition, thinking, reasoning, speech understanding, pattern recognition, decision-making, reasoning and control. In this chapter, some basic cognitive aspects of path planning and navigation are introduced, followed by some approaches adopted in mobile robotic navigation research, which guide our design of intelligent sensory and control mechanisms. Finally, some landmark based scene recognition methods are explored.

Path Planning and Navigation

Except industrial robots, most robots would be of no advantage without the ability of goal-directed motion, i.e., navigation. Modern animals such as insects (bees, rats, pigeons) as well as humans exhibit an astonishing degree of spatial competence. Understanding how navigation is achieved biologically may be useful to human (engineers) robotics researchers.

Cognitive Aspects

The mechanisms by which animals find their way around have proved fascinating. Three systems are fundamental to human and animal navigation: a path integration system for computing and updating the relationship between one’s current position and other significant environmental locations, a scene recognition system for guiding navigation through familiar terrain, and a reorientation system for determining one's position and heading direction when one has become disoriented [69].

Dead reckoning, also known as path integration [70] and vector navigation [71] in animal navigation, refers to navigation using motion information to keep a running calculation of the distance moved and in which direction(s) since the last known location without the use of discrete external cues (landmarks). The purpose of decd reckoning is to continuously keep track of one's own position relative to the point of departure by continuously recording the rotational and translational components of their movement and integrating these angular and linear data into a format called the home vector. This home vector is obtained from various signals derived from the animal's own locomotor activities and can be used to determine its current location by comparing it with that at the start. Therefore the summation of the home vectors can be used to continuously update the relationship between the animal's current location and the locations of significant places. In the animal literature, the terms allocentric and egocentric have been offered as two theoretical accounts of path integration. In egocentric processes, the animal determines the spatial relationships of features in the environment with respect to its own body. In allocentric processes, the animal stores and uses information about the relationship of these features to each other. However, allocentric directions play little role to guide perception and action without being transformed into a set of egocentric coordinates centered on the perceiver. For travel over short distances and memory over short-time durations, egocentric representations may be sufficient. However, if the final destination is not in the immediate view, spatial memory is necessary. For travels over time durations perhaps as long as years and distances of kilometers, long-term spatial memory is needed in the form of an allocentric representation that is independent of the subject's current location and history of movement. Allocentric representation is also referred to as a "cognitive map" [72] in which locations are defined in terms of their relationship to each other or to the environment as a whole. Both egocentric and allocentric representations contribute to tasks demanding spatial memory.

However, path integration, i.e., dead reckoning, is sensitive to the accumulation of small errors in the process of updating the representation of its location on the basis of a record of the subject's movement, and therefore, when used alone, has a characteristically limited capability. Once it fails, either partially or completely, the presence of stimuli, such as landmarks, in the environment is required to guide the subject's behavior. As a result, the path integration system should be supplemented by a system for recognizing significant and familiar places in the landscape to provide accurate guidance for navigation. Then, a series of such discrete external cues can be chained together in order to navigate a more complex path through the environment, which is called route-following and route-learning. The use of landmarks for goal location identification is fundamental to path integration [73], which makes the ability to recall the appropriate memories of contexts a significant part of navigating through familiar terrain.

Animals use a strategy of piloting or cognitive mapping in which external landmarks are used to reach a goal location. Usually an isolated natural landmark on its own may easily be mistaken for another similar isolated natural landmark in a different location. Ambiguity can be reduced by not restricting attention just to the landmark cue in question but broadening it to include the larger contextual cues in which the landmark is set. For example, along a route, a sequence of similar looking landmarks may be encountered and the different scenes in which these landmarks are embedded may also be confusable by sharing elements in common. To lessen any ambiguity, the landmark elements that comprise a scene should also be bound together into the so-called configural association, not just random picks from some arbitrary stimuli. A configural association refers to the unique representation of a scene by the joint occurrence of the landmarks and the contextual cues. Only after the landmarks and the contextual scene have been bound together into an association can they be more resistant to interference from conflicting information acquired in another context. The formation of configural associations between a particular set of landmark cues and a particular set of contextual cues is called compound learning, configural learning, or contextual learning, which must take place in every new situation [74].

For contextual cues to play a role in landmark recognition, they are assumed to be somewhat invariant to distance to the observer. It has also been noted that landmarks are likely to be learned from the same vantage points. With increasing distance from that vantage point, the landmark views tend to change gradually and the expectation of encountering the associated local cue (landmark) may fall off with distance [75]. However, landmarks have different effects on navigation with regard to their distances from the observer. Nearby landmarks specify the positions of significant objects, whereas distant landmarks specify the subject's direction (similar to a compass-based system).

Finally, when path integration is fully disrupted and people are disoriented, a reorientation system is needed to restore the representation of the spatial relationship between the subject and its environment by analyzing the shape of the current surrounding layout and relating that shape to the remembered shape of the layout before disorientation.

Biological neural basis

Though a number of important neuropsychological and neuroanatomical issues still remain unresolved, considerable evidence from a large number of studies across many animal species has suggested that the hippocampal region, particularly the hippocampal gyrus, plays a critical role in spatial behavior in terms of the acquisition and retention of object-place association or object-location memory. It has been claimed that the hippocampus region is especially engaged in spatial learning and takes part in forming maps of the external world. These maps, called cognitive maps, were believed to be located in a specialized brain region called the hippocampus [76]. In this strategy, the hippocampus, as a network of cells and pathways that receive information from all of the sensory systems, learns about the spatial configural represenations of the sensory world and then acts on the motor systems to produce appropriate spatial behavior. This theory is supported by the discovery of place cells in the hippocampus of the rat that fire only when the rat is in a particular restricted region of its environment regardless of the animal's orientation. This region is referred to as the place field. It was from observing this phenomenon that O'Keefe and Nadel hypothesized that the hippocampus might act as a cognitive map. The hippocampus is also well known to be important for the formation (and possible storage) of episodic memories [74]. More recently, head-direction cells have been discovered in the presubiculum, mammillary bodies and anterior thalamic nuclei. It is observed that the head-direction cells fire when the rat is heading in a particular direction, without regard to its location [77]. Found in both rats [78] and monkeys [79], head direction (HD) cells are neurons that fire when an animal points its head in a specific direction. Different neurons have different 'preferred directions' and appear to provide a compass-like representation of direction for the animal.

Place and head-direction representations, connected by a fibre bundle, are to some extent interdependent, and overall are strongly coupled [80]. In principle, egocentric sensory inputs can be transformed into an allocentric reference frame through the head direction signal, which may result in place fields being established [81][82].

Transverse patterning problem

As to scene recognition, it has been suggested that one of the main functions of the hippocampus is to help overcome working memory limitations [83]. An important source of evidence supporting the view that the hippocampus contributes to memory by binding together some features of a scene into conjunctive representations comes from the studies of the transverse patterning problem. In the transverse patterning problem, two of three stimuli, A, B, and C, are presented on each trial and reward is issued according to the following rule:A^+B^-, B^+C^-, and C^+A^-. The first rule means that A will be rewarded when presented with B. The second rule means that B will be rewarded when presented with C. The third rule means that C will be rewarded when presented with A. Since each pattern is presented equally often, each individual stimulus, A, B, and C, will receive the same total reward. As a result, this problem cannot be solved the single feature representations individually. Instead, each of the conjunctive representations, AB, BC, and CA, must be uniquely associated with the appropriate response. The model should be able to be generalized to include novel combinations.

Mobile Robot Navigation

Three fundamental questions in mobile robotics are: (1) Where am I? (2) Where am I going? (3) How do I get there?, which translate into, (1) self-localization (2) path planning (3) map-building and map-interpretation [84]. Localization is the agent's ability to establish its own position within a frame of reference. As an extension of localization, path planning requires the determination of the agent's current position and the position of a goal location within the same frame of reference. A map, in this context, denotes a mapping of the world onto an internal representation, which is not necessarily a metric map of the environment but can be any means of describing locations within the frame of reference.

Given a Cartesian frame of reference, a robot's position is (under ideal, i.e., unrealizable, conditions) always precisely definable, and navigation is simple and perfect. Unfortunately, the frame itself is not anchored in the real world but moves with respect to locations within the world. As one way to overcome this problem, mobile robots use internal geometrical representations of the robot's environment to perform the navigational task. Though such geometrical representations can either be obtained from sensor data or supplied by the designer, they are subject to alteration in the course of operation. These "classical" approaches are the core of proprioception-based systems. Another way to overcome this problem is to anchor the navigation system within the world itself, rather than within an internal frame of reference, and detect unique features, landmarks, which enable the robot to navigate with respect to these external features. By identifying and following landmarks or sequences of landmarks in a specific order, navigation is achieved with respect to the world. This approach is referred to as piloting. In this dissertation, the contribution to mobile robot navigation focuses on how to identify a goal location by selecting consistent landmarks for scene recognition to support navigation. In the following, approaches developed for landmark-based scene recognition are surveyed.

Related work in landmark learning

One technique suggested in the literature is to ask the user to define the landmarks before the robot explores. In this approach, camera images are taken at regular intervals as the robot travels, and objects that human designers believe to be easy to recognize are extracted as landmarks, such as doors or line segments [85]. However, this approach suffers from the problem of perceptual discrepancy, that is, the humans perceive the world differently from the robot. For example, the corners, junctions, or corridors defined by human designers may be good landmarks along the hallway from a human point of view, but may not be necessarily well suited to the perceptions of the robot. Therefore, there is no guarantee that the mobile robot can recognize or detect these objects easily. Alternatively, instead of having a human user specify the landmarks, learning methods have been used to select an optimal set of landmarks for performing self-localization in one specific environment. This is a preferred method, not only because it does not require direct programming and adapts well to environmental changes, but also because it is often viewed as an essential part of an intelligent system.

In Thrun’s implementation [86], Bayesian learning was used and the robot’s raw sensory perceptions (camera images) were projected onto vectors in a low-dimensional space, which were then used to make up the landmarks. The projection was then followed by an optimization performed by minimizing directly the robot’s error in self-localization. Thrun’s results yielded superior performance over localization using human-determined landmarks. Vlassis et al. developed a similar but computationally cheaper technique and showed that their optimization approach produced better results than principal component analysis [87]. Fleischer and Marsland used a similar method with the difference being that they didn't carry out any analysis of the utility of the landmarks selected, but instead used a self-acquired model of 'typical' sequences of perceptions, which was independent of any particular task or environment [88]. These approaches rely on an acquired model of a previously explored environment and therefore are suitable for off-line selection of landmarks.

The next set of approaches is dedicated to on-line landmark selection purposes. In Zimmer's implementation, a topological map was created through a process of 'life-long learning' and can be continuously adapted on line by the robot during exploration. Based on comparison of accumulated error statistics at each of the nodes, global statistical information was used to decide where to add and delete nodes in the map [89]. In a related idea, Bourque and Dudek addressed the 'vacation snapshot' problem of deciding in which locations to take camera images in order to obtain a set of images that can best represent an entire environment, followed by keeping running statistics on what is a 'typical' perception, together with backtracking to previously visited locations that were subsequently found to be 'atypical' [90]. Fleischer and Marsland's method differed from these approaches in that they only used local sensory information to decide when to add landmarks to the map [88]. Similarly, Simhon and Dudek used only local information to address the problem of deciding the best locations upon which to create local metric maps and to localize itself within such a map using a distinctiveness function based on the robot's ability. The drawback of this method is that it relied heavily on the user's intuition in selecting an appropriate distinctiveness function. Related work on the problem of learning symbols to describe a robot's route through the environment was done by segmenting a robot's sensory flow into categories [91][92].

Memories of landmark-dependent target locations serve as a place-recognition system, to which specific motor responses can be linked to guide the animal to its goal [93]. Toward this objective, research on spatial systems focuses more and more on spatial behavior as a learned behavior. Then how can the WMtk be used for such behavior learning? The system described in the next chapter is our approach to solving the perceptual discrepancy problem and realizing scene recognition that integrates the autonomous perceptual learning system and the working memory system.

CHAPTER V

A VISION-BASED AUTONOMOUS MOBILE ROBOT SYSTEM

The primary focus of our research is on computer vision-based perception and its interaction with an adaptive working memory system on a robot to facilitate developmental robotics. This research is conducted through implementing the methodologies outlined in Chapter II and Chapter III on a mobile robot platform and applying the system to mobile robot navigation described in Chapter IV. Since experienced-based learning is widely adopted by human beings, primates and many other animals, a major goal in robotics has been the development of robots that employ a natural process of learning through past experiences. There is biological evidence that adaptive working memory (WM) structures, existing in primate brains, are important to the learning and performing of tasks by focusing on the features most relevant to the current task and by providing the embodiment necessary for exploring the issues of task learning through accumulating rewards with regard to these features. However, an adaptive working memory cannot function in isolation. It must be part of a realistic system involving perception, actuators, reasoning, and short- and long-term memory structures. Toward this goal, a system is proposed in Fig 5.1 that is composed of working memory (cognitive structure), long term memory (database), and perceptual modules (computer vision). The center of this system is PFC WM. It communicates directly with other modules, receiving data from the perceptual system. The attentional focus of the robot is realized by the limited storage capacity of WM which has some means of deciding which information should enter the working memory and which should be removed, and holds the objects currently within the attention of the robot. However, without a human's sophisticated visual system, the robot perceives the world differently from a human. To solve this perceptual discrepancy problem, the first stage in the development of our vision-based autonomous mobile robot system is to let it acquire percepts on its own by way of employing local color and texture features.

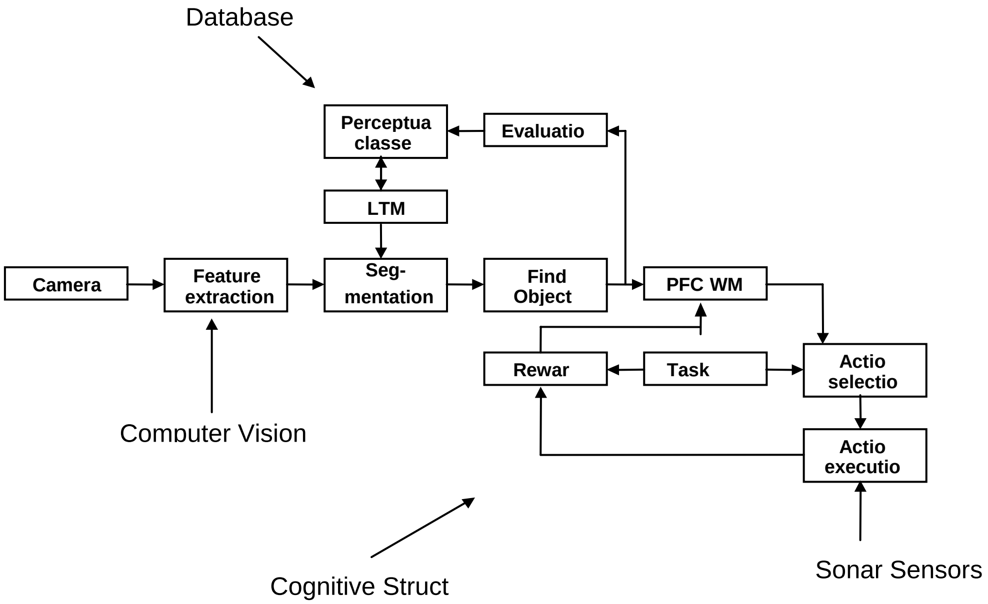


Fig 5.1 Our current system diagram

Visual Percept Acquisition

HSV color histogram

It was Isaac Newton who presented two crucial concepts for color vision [94]. First, white light can be broken down by refraction through a lens into colors of decreasing wavelength from red to violet. Secondly, the long and short wavelength ends of the color ‘spectrum’ can be combined to create a color circle, with the non-spectral color purple at the interface between red and violet. In 1802, Thomas Young correctly proposed the trichromatic (three color) theory, which suggested that, for a human observer, any color could be reproduced by various quantities of three colors selected from various points in the spectrum. Biologically, there exist three cones in the human retina: red cones, green cones and blue cones. The responses from the three different cones are compared to allow color discrimination. One of the most important functions of the visual system is its ability to recognize an object under a variety of different viewing conditions. For example, during a single day, the spectral content of daylight changes significantly. However, surfaces and objects retain their perceived color despite the changes in the wavelength and energy composition of the light reflected from them. This phenomenon is called color constancy and is displayed in a wide range of species, including humans. Wavelength-selective cells whose responses seem to correlate with the human perception of colors have been reported in LGN, V1, V2 [95]. In V4 and beyond, it has been reported that there is no pre-specified wavelength composition that leads to a color and to that color alone and that V4 is concerned with automatic color constancy operations [96].

To follow this process, the images and videos taken from a camcorder are stored for each frame as a RGB (red, green, blue) color image. Next, the RGB color images are converted to the HSV (Hue Saturation Value) color space for use [97]. The Hue describes each color by a normalized number in the range from 0 to 1 starting at red and cycling through yellow, green, cyan, blue, magenta, and back to red. The Saturation describes the vibrancy of the color and represents the purity of a color such as the "redness" of red. The less saturation in a color, the more pale it looks (washed out). The Value describes the brightness of the color. For normalized RGB values in the ranges from 0 to 1, the conversion to HSV is done in the following manner:

$$H = \begin{cases} (0 + \frac{G-B}{MAX-MIN}) \times 60 & if \quad R = MAX, \\ (2 + \frac{B-R}{MAX-MIN}) \times 60 & if \quad G = MAX, \\ (4 + \frac{R-G}{MAX-MIN}) \times 60 & if \quad B = MAX, \end{cases} \quad S = \frac{MAX-MIN}{MAX}, \quad V = MAX \qquad 5.1)$$

where *MAX* is the maximum value of (*R*, *G*, *B*), and *MIN* is the minimum. From the above formulas, it can be seen that, if $MAX = MIN$, *H* is undefined and *S* = 0, there is no hue and the color lies along the central line of grays; if $MAX = 0$, $V = 0$ and S is undefined, the color is pure black and there is no hue, saturation and value.

As the outputs of the above formulas, the Hue values range from 0 to 360, and the Saturation and Value range from 0 to 1. The Hue values are next normalized from 0 to 1. Orange with a bit of red is 0.00 to 0.05, yellow is 0.05 to 0.14, yellow-green is 0.14 to 0.22, green is 0.22 to 0.28, blue-green is 0.28 to 0.45, blue is 0.45 to 0.54, blue-violet is 0.54 to 0.75, purple is 0.75 to 0.81, red-violet is 0.81 to 0.92, and red is 0.92 to 1.00.

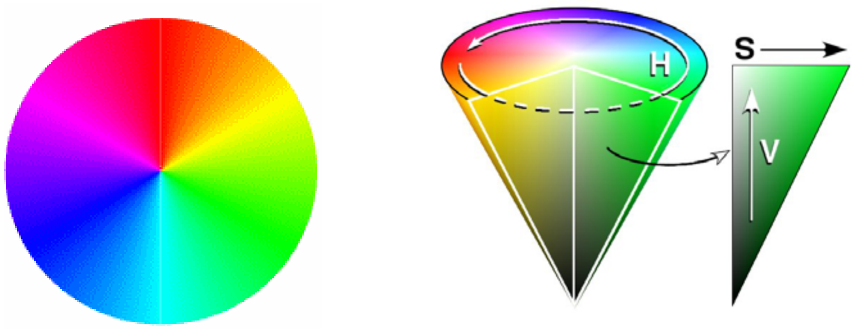


Fig 5.2 RGB and HSV color spaces (Copyright, 2002) [98]

To construct color features, a histogram (or equivalently a probability density function, pdf) of color measurements in the HSV space for each object is computed as follows. The hue is broken into 100 bins of equal width, [0.00 … 1.00]. The saturations and values are evenly distributed into 10 bins each ranging from 0.00 to 1.00. The bin values are [0.00 0.10 0.20 0.30 0.40 0.50 0.60 0.70 0.80 0.90 1.00]. Each color can be represented by combining the three bins, one from the hue bins, one from the saturation bins and one from the value bins. All possibilities of the combinations equal 10,000 different color features, or bins, for the histogram. The histogram can then be constructed for a region of the image by looking at each color feature and finding the number of pixels in the region that correspond to that feature. After doing this for all the color features in the object, there are 10,000 numbers, each representing the number of pixels of a certain color in the selection region. The total number of pixels in the selection region divides these 10,000 numbers, resulting in a highly sparse feature vector of high dimension. The reasons why we use a sparse high dimension space are: (1) to differentiate colors from each other as much as possible (2) to mimic sparse coding scenario. 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 using either the activity of a single, individually meaningful cell or by the global activity pattern across a whole cell population. This is often referred to as sparse coding [99].

Gabor texture measure

Orientation stimuli are sensed in V1, and the optimal stimulus in V1 is a sine-wave grating which is classified 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 [100], 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 can 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 can be approximated as,

$$G(x,y,\theta,\lambda,\varphi,\sigma,\gamma) = e^{-\frac{\tilde{x}^2+\gamma^2\tilde{y}^2}{2\sigma^2}} e^{j(2\pi\frac{\tilde{x}}{\lambda}+\varphi)} \tag{5.2}$$

where $\tilde{x} = x\cos\theta + y\sin\theta$, $\tilde{y} = y\cos\theta - x\sin\theta$, θ specifies the orientation of the wavelet, λ specifies the wavelength of the sine and cosine waves and determines the spacing of light and dark bars that produce the maximum response, φ specifies the phase of the sine and cosine waves and determines where the ON-OFF boundaries fall within the receptive field, σ specifies the radius of the Gaussian, and finally, γ specifies the aspect ratio of the Gaussian.

As with pure sine and cosine waves, the Gabor functions (localized frequency filters with Gaussian envelopes) can provide a complete description of any complex waveform. By optimally specifying the combination of spatial frequency and spatial location information, these functions may be utilized to characterize any complex stimulus. Therefore a visual scene may be broken down into a large number of patches and the total information in the scene could be represented by the outputs of an array of Gabor filters with different frequency, orientation, and phase tuning for each patch.

Given each image patch as used in color histogram generation, the convolutions of the 15×15 image blocks with each of these Gabor wavelets are calculated to generate the orientation feature. In this work, The following set of parameters are used, $\theta \in \{0, \pi/8, 2\pi/8, 3\pi/8, 4\pi/8, 5\pi/8, 6\pi/8, 7\pi/8\}$; $\lambda \in \{\sqrt{2}, 2\sqrt{2}, 3\sqrt{2}, 4\sqrt{2}, 5\sqrt{2}\}$, $\varphi \in \{0, \pi/2\}$, $\sigma = \lambda$, and, finally, $\gamma = 1$. This results in eight orientation, five frequencies, and two phases for a total of 80 different wavelets. The magnitude of each corresponding sine and cosine is calculated and averaged to obtain 40 texture measures. These Gabor wavelet masks are shown in Appendix B.

For each image patch, 40 Gabor texture measures are appended to the 10000-dimensional HSV color histogram to form a local feature vector of a total dimension of 10040, which, together with a similarity measure and a clustering algorithm, is subsequently processed to form percepts.

Metrics

To organize patterns into “sensible” clusters for both supervised learning and unsupervised learning, a major issue is to define the “similarity” between two feature vectors and, after setting up an appropriate similarity measure, 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. Mathematical models for similarities are metrics, particularly the family of the Minkowski-metrics. 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,

$$d_{ij} = \left(\sum_{k=1}^{K} \left| x_{ik} - x_{jk} \right|^{r} \right)^{1/r} \tag{5.3}$$

In the multidimensional scaling literature, these two metrics are also called the city-block distance and the Euclidean distance, and are believed to represent two types of processing. As indicated by [101][102][103], 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 portion of the feature vector and the city-block distance is used for the texture measure portion of the feature vector. That is,

$$d_{ij} = \sqrt{\sum_{k=1}^{10000} (x_{ik} - x_{jk})^2} + \sum_{k=10001}^{10040} | x_{ik} - x_{jk} | \tag{5.4}$$

Clustering algorithm

To cluster the obtained feature vectors, the minimum spanning tree method is used to find the representative points for each percept. First, a minimum spanning tree is constructed. Next the edges in the tree are sorted and cut in decreasing order to form separated groups until the number of feature vectors in the largest cluster is below a threshold. At this point, any cluster which has more than 100 feature vectors is believed to represent a percept. Finally, a nearest neighbor search algorithm is used to assign labels to the remaining points according to the label of their nearest neighbor. At this step, we have finished building up the long term perceptual memory.

Database tree

Though computer memory and speed have improved tremendously nowadays, performing a pure nearest neighbor search over a large database is still time consuming. A method for efficient search structures is desired. Common structures used to organize data for efficient nearest neighbor search include Kd-tree (K-dimensional tree) [104], which is a generalization of the simple one-dimensional binary tree, R-tree [105], which is a natural extension of the B-tree for *k*-dimensions, Quad-trees [106], which represent a hierarchical decomposition of a space of data points, Binary space partitioning (BSP) trees [107], which are similar to Kd-trees but use an arbitrary hyperplane rather than one that is parallel to an attribute axis.

To build a memory network, in our approach, a tree-structured vector quantization method is used to generate a 3-way approximate nearest neighbor search tree. Given the obtained training set, a tree root node exists at the first level which includes all the indices to the database feature elements. Then an initial set of 3 representative patterns, called centers, are randomly selected from the elements of the root as its children cluster centers (or tree nodes) at the second level, and the whole set of the elements is clustered into three subsets by assigning each feature vector to its closest representative cluster center according to the mixed distance measure. At the third level, for each of the 3 clusters obtained at the second level, three feature vectors are selected randomly as its children cluster centers and elements are clustered correspondingly, resulting in 9 cluster tree nodes at the level. This procedure continues until either all the elements in a node belong to the same object class (a pure node) or the number of the elements in a node is below some limit, say, one hundred. Every feature vector in the leaf nodes has a percept landmark associated with it.

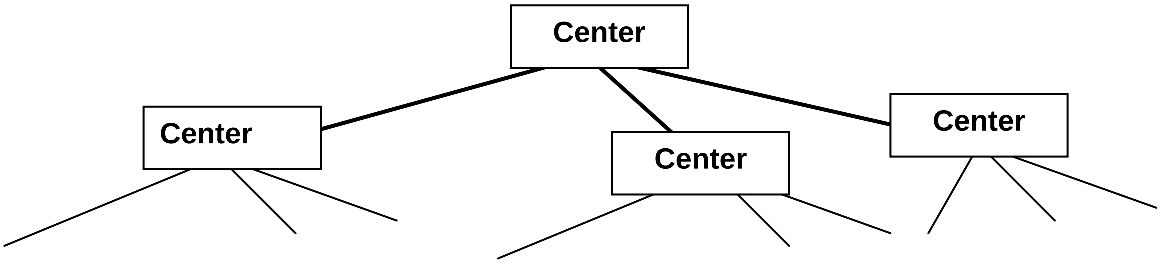


Fig 5.3 Illustration of database tree structure

Next, to search through the tree, given a new feature vector, its distances to the randomly chosen cluster centers at the second level are calculated and the winner is the center that the feature vector is nearest to. At the third level, the feature vector's distances with the three children centers of the winner at the second level are computed and the corresponding new winner is selected. This procedure continues until coming to a leaf node. If it is pure, then stop. Otherwise, do a nearest-neighbor search, and the winner is the training feature vector which gives minimum distance according to the chosen metric.

For any new images grabbed by the robot, the segmentation should yield a set of connected regions in the image. A connected-component labeling algorithm is then applied to find these regions, or blobs, and can be used to produce working memory chunks. Each blob not only represents one specific cluster in the feature space, but also represents a clustering of pixels in the image space (image space clustering). The parameters associated with each blob are the object label of the blob, the location of the blob center, the width and height of the blob. These parameters can be organized into a data structure used later as an input chunk to the WMtk.

With the perceptual discrepancy problem being addressed and the perceptual module being established, the next step is to integrate the perception acquisition system with the WMtk to facilitate the design of intelligent sensory and control mechanisms on the mobile robot platform

Integration of the WMtk into Robot Control

For each new image grabbed and segmented, a number of blob chunks are obtained as recognized objects in the image. When presented to the working memory, according to our experience, not all chunks but those most relevant to the current goal task will be retained in the working memory. Similarly, the capacity of the working memory is restricted so as to provide to the robot focus for various task solving efforts as well as guidance for appropriate action search. That is to say, those blob chunks that are critical for current task success should be selected for retention in the working memory system. Therefore, which blob chunks in the current context should be actively maintained in working memory and which can be safely discarded are the primary problems the working memory system has to solve. The Working Memory Toolkit (WMtk), adopted in this research uses a model of the working memory system, grounded in computational neuroscience models of the working memory circuits of the prefrontal cortex [108], to provide a mechanism for learning which chunks should actively remain in memory based on the robot's past experience. It is believed that the working memory circuit of the prefrontal cortex with connection to and from midbrain dopamine neurons may guide the updating of working memory contents and the learning of motor sequences.

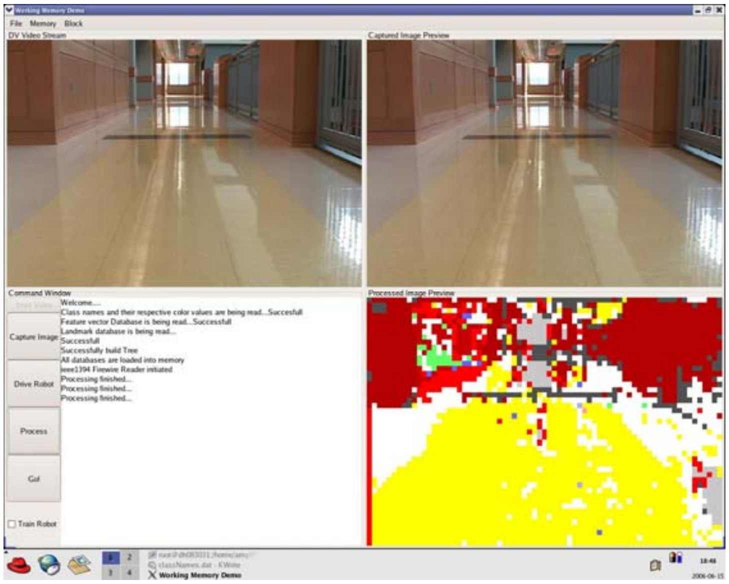
In use of the WMtk to build a robotic working memory system, a WorkingMemory object is created and configured to hold a list of candidate blob chunks that will be delivered by the perceptual modules. In order to evaluate the retention of a chunk or a combination of chunks, after an action is taken by the robot toward a goal, a prediction of task success in the form of a reward is passed to the WorkingMemory together with the candidate blob chunk list. Then the WorkingMemory object uses the TD learning algorithm to learn which chunk or chunk combination to retain and which to discard based on the reward. Next, the robot proceeds with another sensing, acting and reward delivering cycle.

In the toolkit, the structure of chunks is nothing more than a pointer to a memory location and a string that will be defined by the user to denote the chunk type. As a result, there is no way that the WMtk can automatically extract meaningful features from the candidate chunks. To solve real world problems, the WMtk allows the user to define four functions, one to map any chunk into a real-valued feature vector that the adaptive critic uses to assess the value of the chunk, one to map the robot's current situation into a real-valued state vector, one to return a scalar reward value associated with the current situation for the WMtk to compute the TD error so as to drive learning in the adaptive critic, and finally one to release the memory occupied by the chunks when no longer needed.

Having been equipped with this collection of user-specified functions, on each time step of any task, a new list of candidate chunks is first presented to the WorkingMemory object generated by the robot control system. To solve the problem of deciding which chunks to retain, every possible subset of the collection of candidate chunks that can fit into the limited capacity of the WorkingMemory object is examined and translated into a real-valued feature vector, which, combined with the vector encoding of the robot's current situation, produces an input vector to the adaptive critic. The chunk or the combination of chunks that are selected to remain in the working memory is the one(s) that yields the highest estimate of future reward. All other chunks are deleted from the working memory. The temporal difference error is then computed to adjust weights in the adaptive critic. In order for the adaptive critic to explore various new memory combinations from time to time so that it won't get stuck in a local minimum, an exploration rate is used to specify the probability with which a random combination of chunks can be retained in the working memory instead of the optimal subset. This allows a balance of "exploration versus "exploitation" of previously learned knowledge in the reinforcement learning algorithms to enable the discovery of better choices. After a certain period of the learning cycle, the weights in favor of the chunk or chunk combinations most relevant to the current task will be strengthened while those irrelevant to the current task deprecated. As a result, when presented with new patterns after training, the working memory system will choose the ones associated with the highest weights to facilitate the action, somewhat similar to sensory-motor-coordination. With the software part of the mobile robot system being ready, in the following, the mobile robot platform employed in this research is introduced.

Human-Robot Interaction Interface

As shown in Fig 5.4, a GUI interface was developed for human-robot interaction using GTKmm under Fedora Core 4 and 5 (Fedora Core is an RPM-based Linux distribution, developed by the community-supported Fedora Project and sponsored by Red Hat). GTKmm is a computer graphics library and allows us to create the interface which consists of a number of widgets such as those displaying windows, buttons, menus, etc. Derived from the GTK window class, the interface has four windows, one for the video stream which is received from a video camera, one for selecting a single frame from the stream, one for showing the processed image from that single frame, and one text based window to give human user the status of system execution. We also have a variety of buttons on the interface to control the system, such as loading a database, starting the video, capturing a specific image, processing the captured image, and moving the robot.



```
Welcome....
Class names and their respective color values are being read...Succesfull
Feature vector Database is being read...Successfull
Landmark database is being read...
Successfull
Successfully build Tree
All databases are loaded into memory
ieee1394 Firewire Reader initiated
Processing finished...
Processing finished...
Processing finished...
```

Fig 5.4 Vision-based human-robot interaction interface

To display video, the video camera (SONY Digital Handycam DCR VX2000 digital video camera recorder) sends the data to the computer through an IEEE 1394 interface. Our display class receives the signal, collects a frame in a buffer and draws the frame in the window using the X11 windowing system (X11 dynamic graphics program for data visualization). If you want to capture a specific frame, the callback function of the select button will send the frame to the second window on the interface. If you want to process that frame, the event associated with the process button will segment the grabbed image into a group of percepts.

The system is implemented using multiple threads, one for video display, one for video frame processing, and one for moving the robot. Mutual exclusion (most often referred to as mutex) allows the programmer to "attach" locks to resources. If a thread wishes to modify or read a value from a shared resource, the thread must first gain the lock. Once it has the lock it may do what it wants with the shared resource because no other thread should have access to that variable. Once the thread finishes using the shared resource, it unlocks the mutex, allowing other threads to access the resource. This is referred to as serializing access to the shared resource.

From this point on, we can use the system to perform different experiments for different research topics.

Mobile Robot Platform

The mobile robot used in this work is a Pioneer 2 AT four-wheel-drive mobile robot [108], called Skeeter, in the Intelligent Robotics Lab, Vanderbilt University. Pioneer is a family of small and intelligent mobile robots, both two-wheel and four-wheel-drive, whose architecture was originally developed by Dr. Kurt Konolige of SRI International, Inc. and Standford University. These mobile robots contain all of the basic components for sensing and navigation in a real-world environment, including battery power, drive motors and wheels, position/speed encoders, and integrated sensors and accessories. All these components are managed via an onboard microcontroller, with independent motor/power and sonar-controller boards for a versatile operating environment. To support accessories available for Pioneer as well as the user's own custom attachments, the controller has two RS232-standard communication ports and an expansion bus. In this project, a laptop and a workstation computer that both run the Fodera core 4 operating systems are used.

Fig 5.5 Mobile robot Skeeter with camera and laptop

Microcontroller

Pioneer 2's microcontroller uses a high-performance 20 MHz Siemens 88C166-based microprocessor with integrated 32K flash-ROM. The microcontroller also has 32K of dynamic RAM, two RS232-compatible serial ports, several digital, analog-to-digital, and PSU I/O user-accessible ports, and an eight-bit expansion bus. All the I/O ports, except those used for motors, encoders, and sonar, are available to the user for Pioneer 2 accessory hardware. The microcontroller runs a special embedded operating system called P2OS. The host computer talks to the P2OS microcontroller over the standard RS232 serial line. This driver offers access to the various P2OS-mediated devices, logically splitting up the devices' functionality.

Motors and Position Encoders

Pioneer 2's drive system uses high-speed, high-torque, reversible-DC motors. Each front drive motor includes a high-resolution optical quadrature shaft encoder that provides 9,850 ticks per wheel resolution (19 ticks per millimeter) for precise position and speed sensing and advanced dead-reckoning. Pioneer 2's motors are software controlled and disabled when not connected to a client.

Sonar Sensors

In addition to high-precision wheel-motor encoders for finer odometry, and translational and rotational speed controls, Pioneer 2 supports up to two sonar range-finding arrays. One array, affixed under the front of the Deck and atop the Nose, provides forward- and side-range sensing. The other optional sonar array is attached just beneath the rear Deck and provides rearward, as well as side sensing. All arrays contain eight sonars, for a total of 16 sonars around the robot. The sonar positions are fixed in both arrays: one on each side, and six facing outward at 20-degree intervals, together resulting in 360 degrees of nearly seamless sensing. The sonar firing rate is 25 Hz (40 milliseconds per sonar per array) and sensitivity ranges from ten cm (six inches) to more than five meters (16 feet). In other words, objects closer than ten cm are not detected. The sonar's firing pattern may also be accessed and controlled through software. The default is left-to-right in sequence for the forward array and right-to-left on the rear. One sonar from each array "pings" at a time. On the whole, these sonars provide object detection and range information for navigation around obstacles.

With the autonomous mobile system being developed, the next step is to apply the system to some real world task. To achieve the goals, experiments are designed in the next chapter to test the autonomous abilities of the system.

CHAPTER VI

PROPOSED EXPERIMENTS

Scientific research is not just about matter, but also about method. The scientific method comprises the analysis of existing knowledge, identification of open questions, the design of an appropriate experimental procedure to investigate the question, and the analysis of the results. With all the information being explored in the above, the proof of a robot control program still rests on physical experiments. The next step towards our understanding of the complex interaction between robots, the world they operate in, and the tasks they are trying to achieve is therefore to design physical experiments.

Since a truly intelligent robot is expected to be able to deal with uncertain, ambiguous, contradictory and noisy data, to learn through its own interaction with the world, to be able to assess events with respect to the goal it is trying to achieve, and to alter its behavior when necessary, four experiments are proposed in the following. For each experiment, the following steps are needed: (1) Hypothesis (2) Experiment description (3) Evaluation of experimental results to determine if the hypothesis is contradicted or supported.

Experiment 1: Learn Percepts Autonomously

Hypothesis: simultaneous clustering of visual features in feature space and image space can be used for autonomously discovering and learning percepts.

To deal with uncertain, ambiguous, contradictory and noisy data, the first proposed experiment in this research is to evaluate the performance of clustering with as little a priori knowledge input from a human teacher as possible. To find reasonable percepts, the fundamental idea is to cluster the obtained feature vectors in an unsupervised manner.

1. First the robot will use a video camera to take one picture at one location and extract feature vectors, then move 1 meters ahead and repeat the process until it moves a total of 20 meters, resulting in 66,740 feature vectors.
2. Upon the obtained 66,740, the robot will perform an unsupervised clustering algorithm to form clusters in the feature space. The number of clusters can be chosen reasonably larger than the “true” number of clusters in the images so that later a possible “merge” operation could be applied. When this grouping is done, the robot will assign a label to each cluster to form the long-term database.
3. Next, test images are grabbed at the same locations where the training images are obtained and new feature vectors are extracted. To the test images, do a simple nearest neighbor search with respect to the database to segment the test images, and then do the approximate nearest neighbor search with respect to the database tree implementation.
4. The segmentation of test images based on the clusters obtained should yield a set of connected regions in the image. A connected-component labeling algorithm is then applied to find these regions, which are called blobs in our research. These blobs can be used to produce working memory chunks.
5. Find all the image blobs (candidate working memory chunks) in the test images. Each blob not only represents one of the clusters in the feature space, but also represents a clustering of pixels in the image space (image space clustering).

To summarize, the intention of this first experiment is to look for the presence of discovered features in the set of images collected during the testing run.

Since the robot is equipped with a set of basic actions, such as move forward, move backward, turn left, turn right, stop etc., the next question is how to associate each acquired percept with each basic action? This question will be answered by Experiment 2.

Experiment 2: Learn the Meaning of Each Percept with Regard to Movement

Hypothesis: the robot can learn to associate each percept with regard to motion, the so-called sensory motor coordination.

With percepts being identified in the previous experiment, the purpose of this experiment is to show the acquired models of objects could identify target objects with good reliability and to learn some properties of objects through the robot's own interaction with the world using the working memory toolkit. At the end of the first experiment, several meaningful blobs are obtained. Since the robot doesn't know anything about each blob, the task in this experiment is to find whether it represents an obstacle or it is open space.

1. Initially set all weights associated with each percept to be 0.5 and 2.5, respectively.
2. In the same place where training and test images are grabbed, ask the robot to take pictures, partition the pictures into blobs, and remember each blob together with their location. Each blob will be used as an input chunk to the working memory toolkit.
3. Ask the robot to move toward each object for 5 meters long.
4. If the robot moves 5 meters long without bumping onto that object, it will receive the maximum reward, 5, and gather evidence that the object is associated with an open space.
5. If the robot moves $x < 5$ meters long and then bumps onto that object, it will receive a less reward, x, and gather evidence that the object is associated with an obstacle.

Since working memory is always associated with a goal-directed task, the goal in this experiment is to see whether the robot will be stopped consistently by some blobs but not by others. This can be summarized in Fig 6.1. The robot segments the image to form percepts. Percepts are input as chunks to the working memory, which selects one and passes it to the Robot for it to move toward. The response from the robot is used as a feedback reward to the working memory.

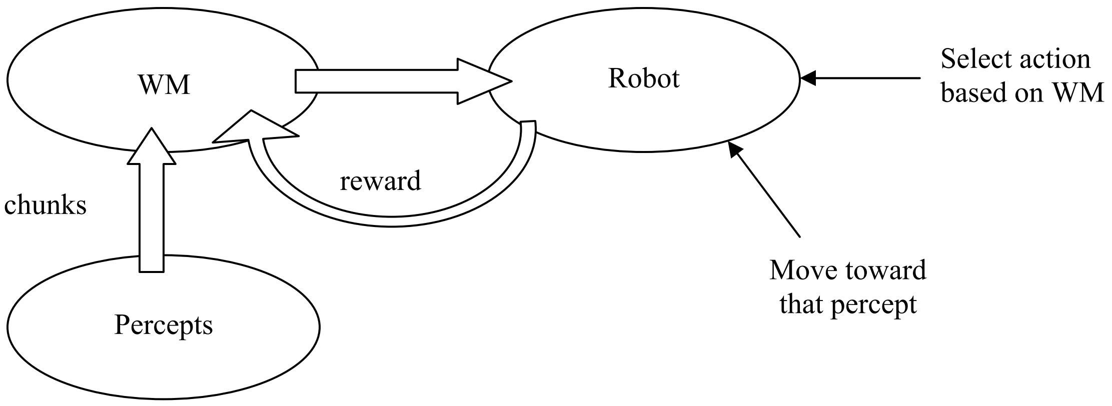


Fig 6.1 Percept learning

Experiment 3: Learn Perceptual Landmarks of a Location

Hypothesis: the robot can learn which percept or percepts are most useful for identifying a specific location even in the presence of distracting percepts.

This experiment is to demonstrate that a robot can learn one or more percepts to locate objects and, based on that, build internal representations of objects in its environment without a priori knowledge of these representations from a human teacher. This basically is a classification problem that learns the correlation between the percept and the target location. For geometrical reasons, the landmarks are supposed to be ones that are close to the goal and large in size so as to best aid the pinpointing of a goal. Therefore, pictures are taken at the current and the goal locations.

1 A specific location in a hallway is chosen as a target location to be taught to the robot. This location will have at least one clearly distinguishing characteristic, such as a stripe of black floor tiles.

2 A set of images, with the robot in different orientations, are collected, in which the black stripe is clearly visible in the foreground of the image. Also, in a majority (e.g., 80%) of these images, an irrelevant distractor, e.g., a green ball, will also be present. These images are all labeled as “target”.

3 A second set of images is obtained having no view of the black stripe. The irrelevant distractor will as appear in a minority percentage (e.g., 20%) of these images. The percentage may possibly be varied in multiple tests. Each of these images is labeled "non-target".

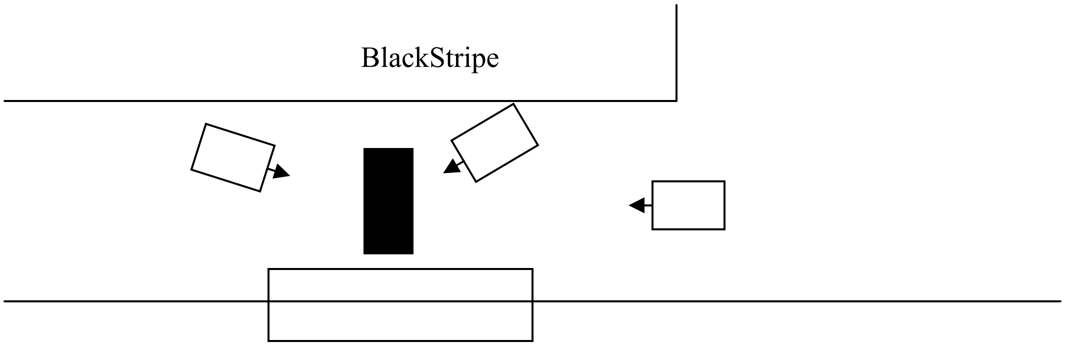


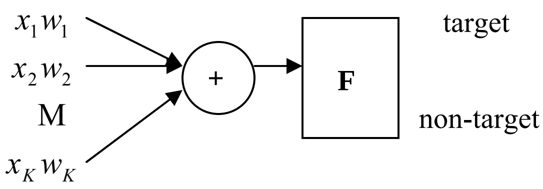
Fig 6.2 Example environment

4 These two sets of images are combined to form a labeled training set. Each image will be segmented into blobs from which candidate working memory chunks are obtained. This collection of blobs/chunks is duplicated so that we have two identical sets. All of the blobs in the first set are given the additional label of "target". All those in the second set are labeled "non-target". The resulting collection of blobs from these two sets is presented to the working memory which will select up to *N* of them. The robot will use the labels in the selected chunks as votes for a target or non-target decision for this image. In the case of a tie (including 0 vote), the decision defaults to non-target. If the decision is correct, the resulting reward is 1; otherwise, it is 0.

5 After training, a set of test images (target and non-target images, with distractors as before) is collected. Each is processed as during training and the target/non-target decision is recorded for each. We measure the total accuracy of the system on these test images. We also measure the percentage of false alarms that occur in non-target images that contain the distractor.

A competing system, based on a single perceptron, will be trained on the training image set. The feature vectors used as inputs to the perceptron are binary vectors denoting whether one or more blobs of a particular cluster category exist in the image and, therefore, having *K*-dimension.

$$\vec{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_K \end{bmatrix} \qquad \vec{w} = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_K \end{bmatrix}$$



where $x_i = 1$, if there is one or more blobs of category *i* in the image, and $x_i = 0$, otherwise. The training perceptron will be applied to the test image set and its target/non-target decisions will be recorded. The total accuracy and false alarm percentages (as measured for the proposed system) will be measured for the perceptron system. The results of the proposed system and those of the perceptron will be compared. The output of an artificial neuron when the sigmoid activation function is used is then given by

$$o(t) = \frac{1}{1 + \exp(-\vec{x}^T \vec{w})} \tag{6.1}$$

Experiment 4: Learn Novel Percepts

Hypothesis: after training, the robot may still learn a new object that it has not previously encountered.

So far an autonomous mobile robot has been equipped with the ability to learn and to navigate. However, the question still remains: which task should the robot actually perform? In addition to typical industrial applications such as factory transportation or repetitive cleaning tasks, more demanding tasks include inspection and surveillance tasks in which decisions must be made based on its sensory perceptions and important information should be distinguished from unimportant ones. Previous research has often used the concept of habituation to differentiate between “normal” and “novel” features. The habituation is a reduction in behavioral response when a stimulus is perceived repeatedly. Because any agent in the real world has to be able to make distinctions between different types of objects in order to act appropriately, the goal of novel object detection is not only to find but also to gather information about such objects. In the following, an experiment on novelty detection is proposed. The goal is for the working memory to know, or learn to pay attention to, novel stimuli.

For each image patch (i.e., the 15×15 block used to generate a feature vector) in a new image, we can measure the novelty of that patch as the distance to the nearest training database vector. The larger the distance, the higher the measured novelty. Thus we can produce a grayscale "novelty image" from the new image. The novelty image can be thresholded (using an empirically chosen threshold) to reveal novel blobs. A connected component labeling algorithm is applied to these blobs. The robot selects the largest blob as a novel object if it is sufficiently large (also an empirically chosen threshold). The feature vectors associated with this object are assigned to a new category, category K+1. New images are collected, and, if novel blobs are found in them, the robot assigns their feature vectors to category K+1. In this research, the practical and simplifying assumption is made that the robot only encounters one new object at a time. Larger numbers of new objects will be left for future research.

If the robot is returned to the same environment with the "new" object being present, it should now not regard the object as novel and needing to be learned. A green ball is used as the novel object in this project.

In summary, the goal of these experiments is to examine the ability of an autonomous mobile robot system to form percepts from visual input, to learn some properties of these percepts, to find its way around based on learned knowledge about these percepts, and finally, to perform some useful tasks, all on its own.

CHAPTER VII

EXPERIMENT RESULTS AND CONCLUSION

Experiment 1 Results

To obtain a percept database, 20 pictures along the hallway were taken, using a SONY Digital Handycam DCR VX2000 digital video camera recorder, and 66,740 unlabelled feature vectors were extracted from these images. Based on the nearest neighbor distance, a minimum spanning tree is constructed from this data. 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 a user-specified threshold obtained through error-and-trial. 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 to build up the database. Each class was assigned a different color for displaying purposes. Eleven clusters were found using this method and corresponding percepts are denoted in the following table 7.1. It can be seen that each percept identified by a human has multiple percepts perceived by our robot vision system.

Table 7.1 Autonomously learned percepts

Denoting Symbol	Percept Name Identified by Human
Object1	BlackStripe
Object2	WoodPanel
Object3	LightReflection
Object4	WoodPanel
Object5	WhiteFloor
Object6	WoodPanel
Object7	WoodPanel
Object8	WoodPanel
Object9	WhiteFloor
Object10	YellowFloor
Object11	YellowFloor

Simple nearest neighbor search

Nine more images along the hallway were taken as test images. A simple nearest neighbor search with respect to the original database was done to segment these test images. The segmentation results based on the Euclidean distance, the proposed mixed metric and the city-block distance for the first image are shown below in Fig 7.1. The one in the upper left corner is the original image. The one in the upper right corner is the segmented image using mixed metric. The one in the lower left corner is the segmented image using Euclidean metric. Finally, the one in the lower right corner is the segmented image using city-block metric. The results for 8 other images are shown in Appendix B. From these results, in our qualitative visual evaluation, it seems that the mixed metric is superior to the Euclidean distance and city block distance. The segmentations using the Euclidean distance consistently misclassified the black stripe, which is believed to be a very important percept. The segmentations using the city block distance found some of the black stripe, but missed part of it from time to time. The city block distance also tended to misclassify some wood panel as black. For the rest of the percepts, they gave different misclassifications. The mixed metric and the Euclidean distance misclassified the railing as the wood panel, while the city block distance saw it as the white floor, which is not too surprising

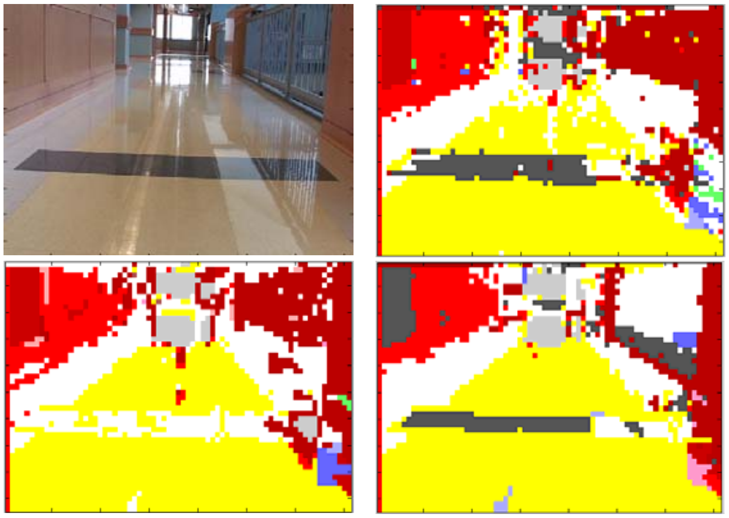
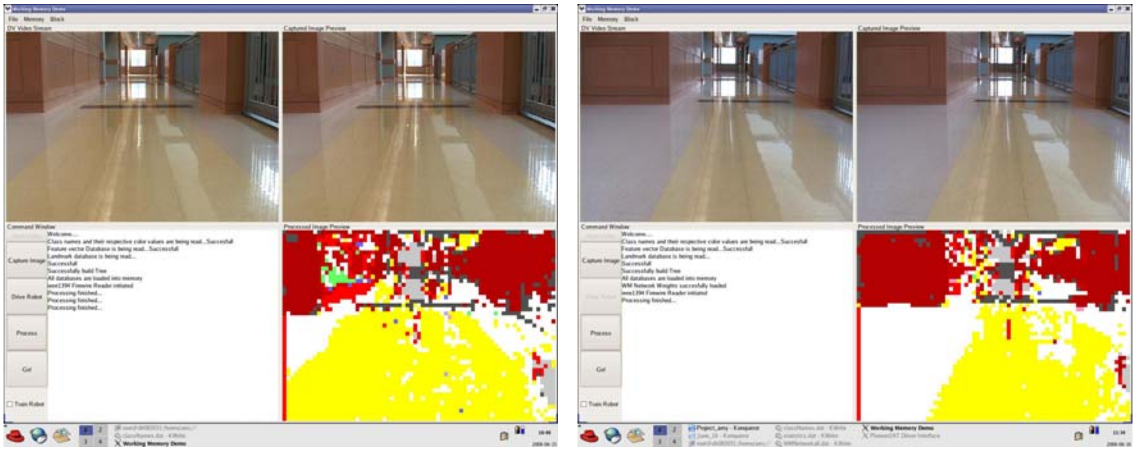


Fig 7.1 Simple nearest neighbor search results

since the railing is a hard percept to model. On the whole, it can be seen that all of metrics generally did a reasonable job and clearly identified the yellow floor, the white floor, and the wood panel.

Approximate nearest neighbor fast search

To apply the computer vision system to a real environment and speed up the processing time, an approximate nearest neighbor search with respect to the database tree was done to segment the test images. The segmentation results based on the mixed metric through the tree are shown in Fig 7.2.



```
Welcome...
Class names and their respective color values are being read...Successfull
Feature vector Database is being read...Successfull
Landmark database is being read...
Successfull
Successfully build Tree
All databases are loaded into memory
ieee1394 Firewire Reader initiated
Processing finished...
Processing finished...
Processing finished...
```

```
Welcome...
Class names and their respective color values are being read...Successfull
Feature vector Database is being read...Successfull
Landmark database is being read...
Successfull
Successfully build Tree
All databases are loaded into memory
WM Network Weights successfully loaded
ieee1394 Firewire Reader initiated
Processing finished...
```

Fig 7.2 Our perceptual system and fast segmentation results with respect to database tree

It can be seen that it worked pretty well in the foreground of the images, which is actually the part the robot is most interested in. To summarize, the intention of this experiment was to examine the autonomously discovered features in the set of images collected during the test run, with a particular emphasis on seeing whether they tend to correspond well to objects in the environment.

Experiment 2 Results

To show the performance of the second experiment, the weights associated with each percept were recorded at each trial for two sets of initial weights, 0.5 (pessimistic) and 2.5 (moderately optimistic). The results are shown in Fig 7.3 and Fig 7.4, respectively. The plot on the left of each figure was the weights obtained until one of the percepts eventually dominated. The percept that dominated corresponds to the yellow floor (denoted by Object10 in the experiment since the robot doesn't know about the name we humans use). The plots on the right of each figure were the weights obtained after the yellow floor has been blocked from being presented to the working memory. The percept that became the second dominant one corresponded to the black stripe running across the yellow floor as shown in Fig 7.1. In general, of all the numerous stimuli surrounding an animal, only one, prescribed by the goal selector, is admitted to the motor system to guide its activity. That is, for the most part, only one response plan controls the motor system at any time. Therefore the number of chunks in WM is chosen to be one, i.e., the working memory size is one. However, it happens that sometimes the working memory will choose no percepts. As a result, the feature vector used in the WMtk has a position reserved when no chunk is chosen by the working memory.

From Fig7.3, though at the beginning, the model generates random responses from sensory input, it learns the right percept very quickly, in no more than 10 trials. However, from Fig 7.4, it seems that the weights may not always appear rational, but they are always based on the data available at the time and will be modified by any relevant information the individual has absorbed along the way. At the beginning, the model generates responses in a seemingly random fashion. This is believed to be the case in early stages of learning, when the animal makes random responses or a response acquired previously in a somewhat similar environment. It plans randomly, but fails to execute plans that have no or less rewarding associations. Subsequently, it uses trial-and-error, and the weights drop until the most useful percept can be spotted. Finally, the speed with which a well-practiced response is made suggests that it is no longer chosen randomly.

A frequently successful response, which is actually the yellow floor percept, acquires associations from the context and gradually (after approximately 50 trials) becomes the most likely one to be activated. When this association grows strong enough, though responses still compete with each other to become active, the associated sensory input will break the balance in favor of the most frequently and vigorously active response occurring in that context. As soon as the corresponding

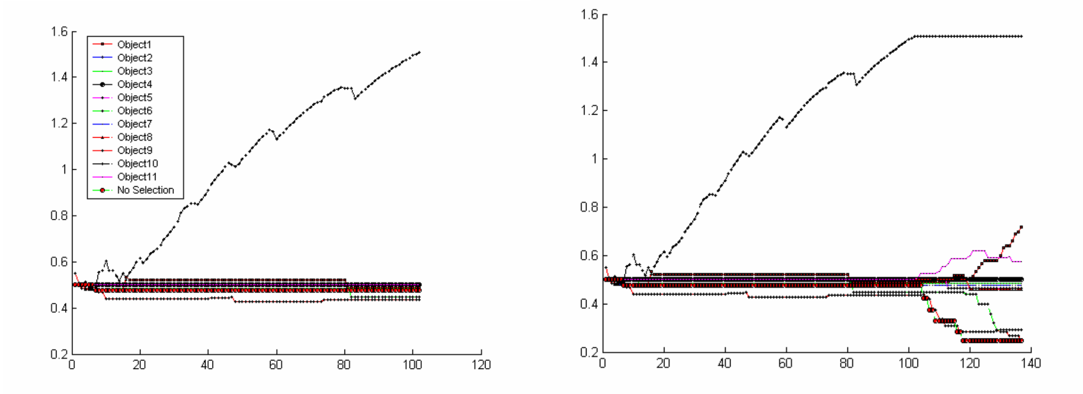


Fig 7.3 Weight logs for each percept when the robot was let to go freely with initial average weight at 0.5

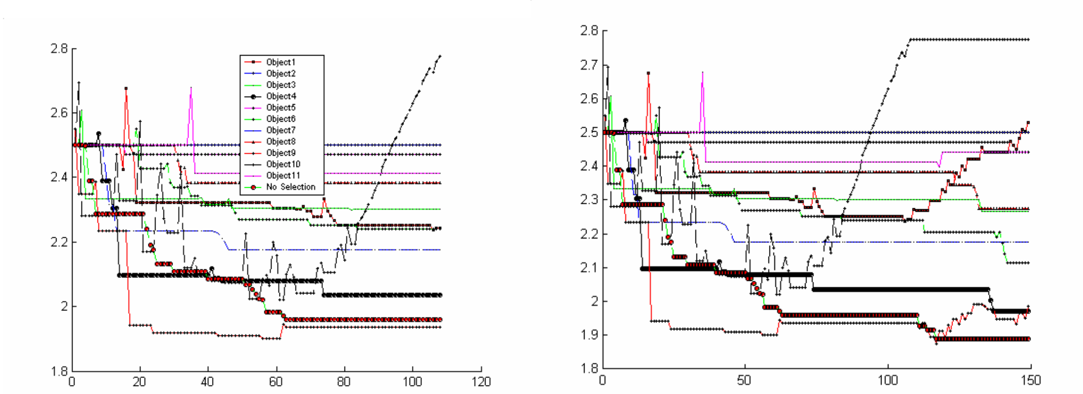


Fig 7.4 Weight logs for each percept when the robot was let to go freely with initial average weight at 2.5

yellow floor was recognized as being associated with open space, it was always the chosen percept and its weight took off quickly. The same performance was observed for the second dominant percept, black stripe (denoted by Object1). The reason why there were more or less large fluctuations at the beginning is that random noise was introduced by the toolkit so that the algorithm would not be locked at the local minimum, an example of the so-called exploration vs. exploitation trade off. It can be seen that the weights of some percepts didn't change much. This is because those percepts were rarely observed in the lower half of the image, even though they were considered when the database was built.

In terms of behavior, at the beginning, the robot moves about at random, improving its chances of encountering anything it needs. In other words, this kind of exploration may occur in the absence of any reinforcing stimulus, but it need no longer be random when the robot can learn. The ability to learn enables the robot to remember what it has seen and where it has been. After a percept (stimulus) has been approached and examined several times during the learning, the robot will evolve to activate responses that are most likely to reduce the need, that is, to find open space, by more and more switching attention to the sensory input necessary for effective

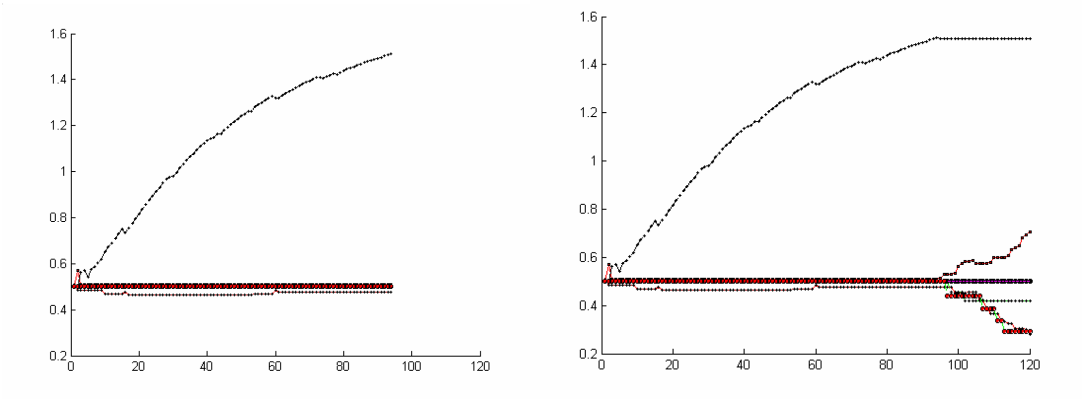


Fig 7.5 Weight logs for each percept when the robot was forced back to the center after each trial with initial weights set to 0.5

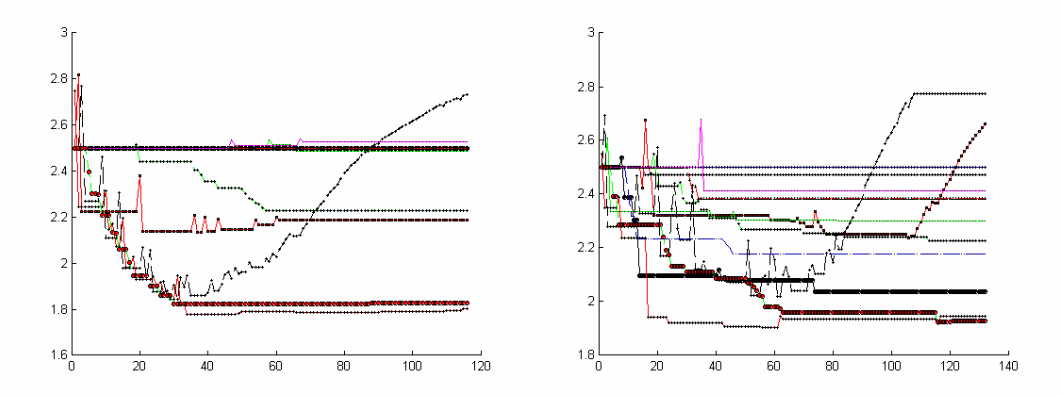


Fig 7.6 Weight logs for each percept when the robot was forced back to the center after each trial with initial weights set to 2.5

response, that is, always being able to move 5 meters without being stopped. Unattended signals or percepts that are not immediately necessary for the performance of a response may also be presented by the sensory systems and delivered, but will eventually be excluded from the parts of the motor system engaged in performing the current task.

As shown in Fig 7.5 and Fig 7.6, similar performance was observed for the second set of experiment, where the robot was forced back to the center of the hallway after each trial. During the trials of the second experiment with initial average weight of 2.5, after the robot had started learning, a percept displayed by the color blue, as shown in Fig 7.7, showed up after 4pm in the afternoon and was always chosen more favorably than even the yellow floor. Its weight was denoted with magenta line in the figure on the right. The phenomenon shows the system adapting to a percept that only appeared at a particular time of the day, i.e., in a particular lighting condition.

Experiment 3 Results

This experiment is for the robot to use the WMtk to learn to understand target/non-target locations denoted by the existence/non-existence of certain landmarks in the environment through the learning of the configural representation of the percepts. The identification of landmarks in the environment is important for navigation. In the last experiment, the robot has learned what percept can be moved toward to move freely in the hall environment. Now it must learn about target locations for tracking its state in the environment. To train the robot on target/non-target locations, two sets of images are taken from the environment that represent landmark and non-landmark locations as seen in Fig 7.8. Each set contains an equal amount of images taken from both types of locations. For good network training results, the images are selected at random from the two sets and segmented based on the search tree created from experiment 1. Each

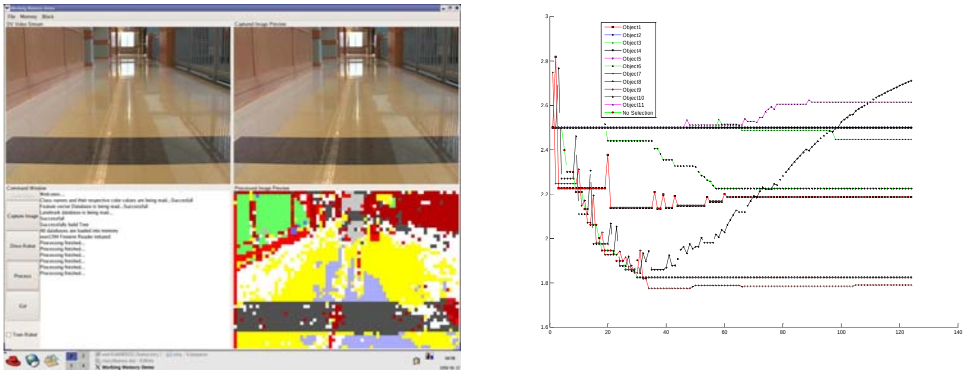


Fig 7.7 Log of the weights for each percept when the robot was forced back to the center after each trial

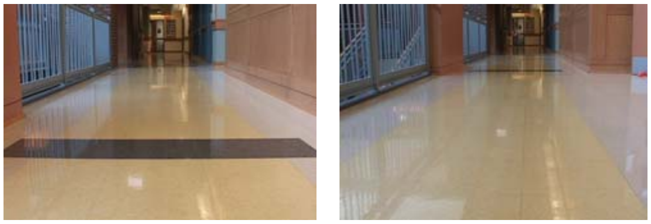


Fig 7.8 (left) Target image (right) Non-target image

percept that is available in the image will be presented to the WM and duplicated as two chunks, one representing the percept as a landmark percept (e.g. Object1_L) and the other as a non-landmark percept (e.g. Object1_NL).

In the simulation, the target images are denoted by 1 while the non-target images are denoted by 0. Each image is a separate trial episode for the WM. The WM size is set to values of 1, 2, and 3 to determine which percept or conjunctions of percepts are most useful for identifying a target location. For the cases of WM size being greater than 1, based on the chunk labels Object_L or Object_NL, a voting mechanism is applied to determine the final decision of the WM with regard to whether the image is a target location or not. For example, if there are three chunks, Objecti_NL, Objectj_L and Objectk_L, the vote will be 1, guessing this to be a target location. In other words, the WM chooses the majority of L (landmark) or NL (non-landmark) as the final vote for this location. If there is a tie happening, such as there are equal L (landmark) and NL (non-landmark) chunks, the decision defaults to 0. Being binary, the reward is 1 if there is an agreement between the vote and the true label of the image and 0 otherwise. If the WM chooses nothing, it will get a reward of 0. The exploration percentage is set to 0.05 and 0.00. Each of the WM sizes is trained for both 1000 and 5000 episodes.

In Table 7.2 and Table 7.3, the total accuracy of the system for two different lengths of trials are shown for two cases, i.e., with and without a distractor, using 3 different working memory sizes. The exploration parameter reflects the percentage of the noise introduced intentionally into the neural network so that the solution won't be stuck with a local minimum instead of a global minimum. For working memory size of one, the neural network implementation inside the WMtk degenerates to the linear case, i.e., single percept coding. For working memory sizes of more than one, conjunctive coding becomes possible and it can be seen that the performances increase dramatically.

In order to see if the performance of the system improves after learning, a plot of the average correct classifications over a sliding window of 25 points over 1000 episodes is shown in Fig 7.9

below. It can be seen that, at the beginning, the working memory system starts with random guesses and chooses randomly. With training, it learns the correct choices more and more and converges to 1 in about 50 to 60 episodes when the working memory size is larger than one.

Table 7.2 Percentage of correct location classification with two WM exploration rates

WM size	WM exploration set at 0.05		WM exploration set at 0.00	
	1000 trials	5000 trials	1000 trials	5000 trials
1	49.40%	49.80%	47.90%	50.06%
2	95.40%	95.90%	96.50%	99.62%
3	88.40%	96.44%	98.80%	99.40%

Table 7.3 Percentage of correct location classification with two WM exploration rates with a distractor

WM size	WM exploration set at 0.05		WM exploration set at 0.00	
	1000 trials	5000 trials	1000 trials	5000 trials
1	51.30%	49.22%	52.30%	48.88%
2	89.50%	95.56%	98.90%	99.54%
3	94.40%	96.28%	92.80%	96.82%

Thus, the system is capable of learning configural representations. The same procedures were followed for the case where training images contain a distractor. Similar performance was observed and is shown in Fig 7.10. The presence of a distractor seems to have little impact on the WM toolkit.

In order to examine whether the performance of the system improves with learning, its behaviors in the weight space are analyzed in the following. With working memory sizes larger than 1, there are totally ((Chunk_Vector_Size + 1)^WM_size) weights, each containing a value associated with the performance of a particular combination of WM contents. With a working memory size of one, the case simplifies to a single entry for each element associated with the chunk vector encoding. Though all the weights were initialized to be 0.5, they change after learning. First in Fig 7.11, the weight distributions at episode 1000 for three working memory sizes with two WM exploration rates without a distractor are shown. When the working memory size is one, Object1 has the highest weight, and Object9 and Object10 have the lowest weights. This is intuitive because, when an image of the target location is presented after training, if Object1 is chosen by the working memory, its associated largest weight makes it dominate the

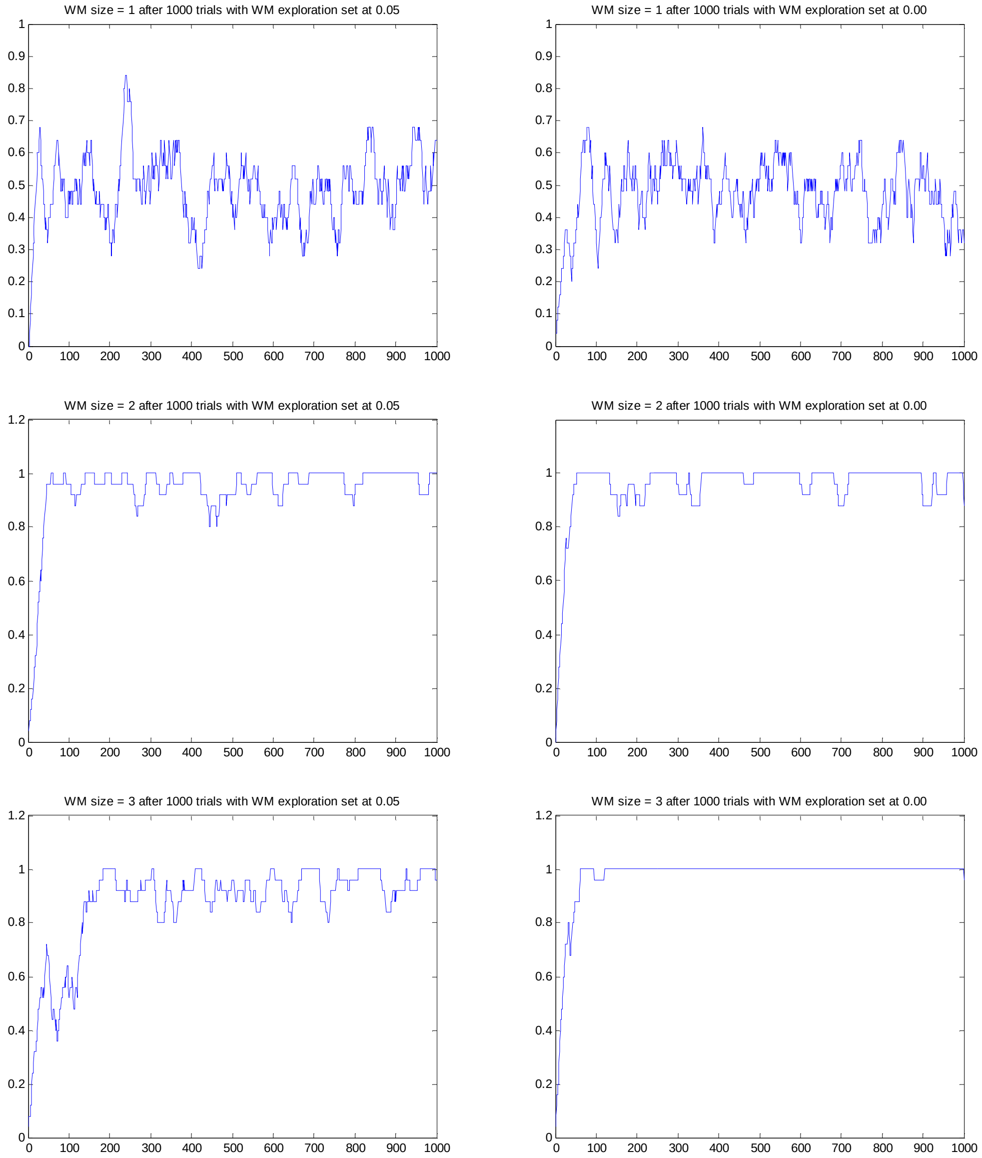


Fig 7.9 25-Point smoothed convolution of correct classification with two WM exploration rates without a distractor

output calculation. On the other hand, when an image of the non-target location is presented after training, if either Object9 or Object10 is chosen by the working memory, its associated small weight will drive the output calculation toward zero.

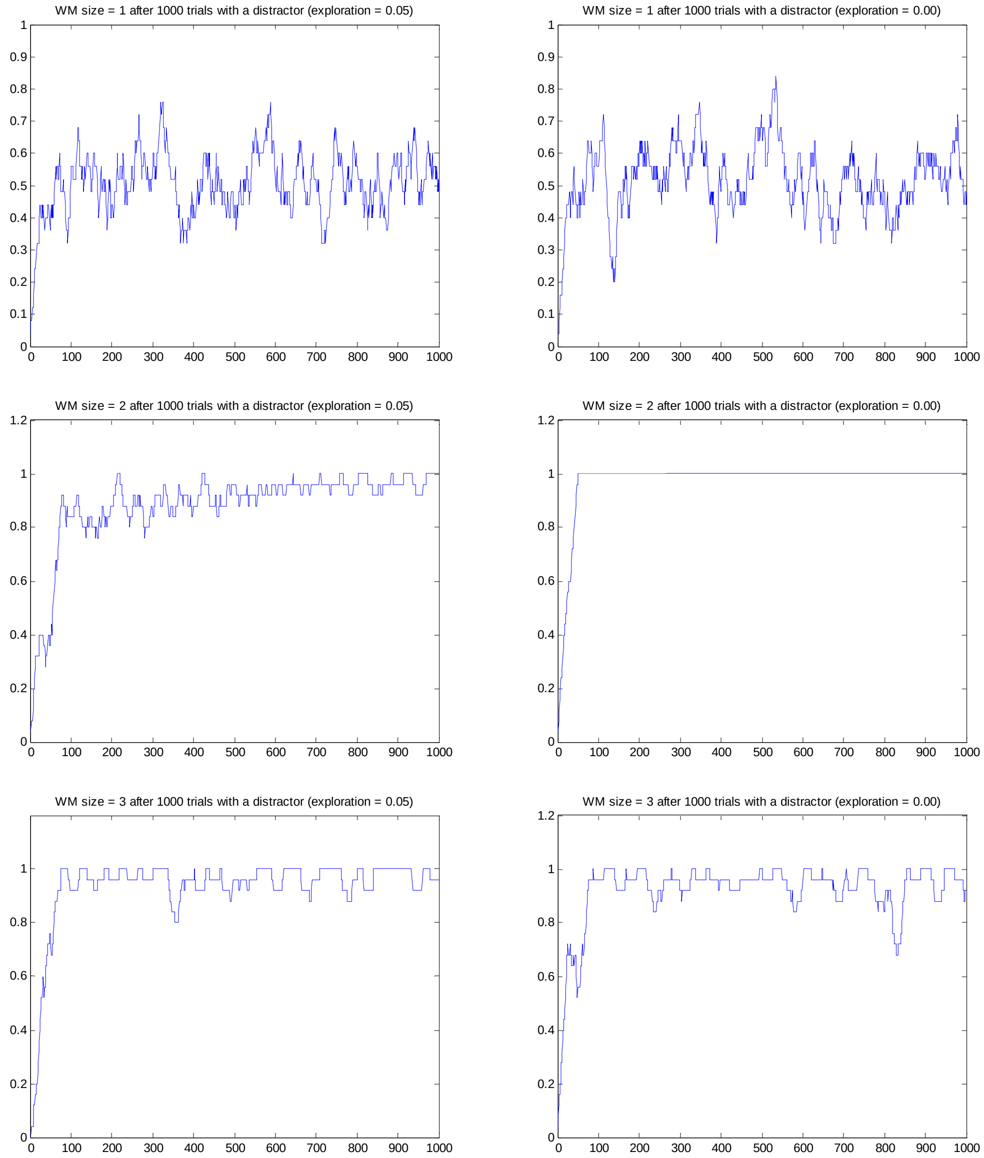


Fig 7.10. 25-Point smoothed convolution of correct classification with two WM exploration rates with a distractor

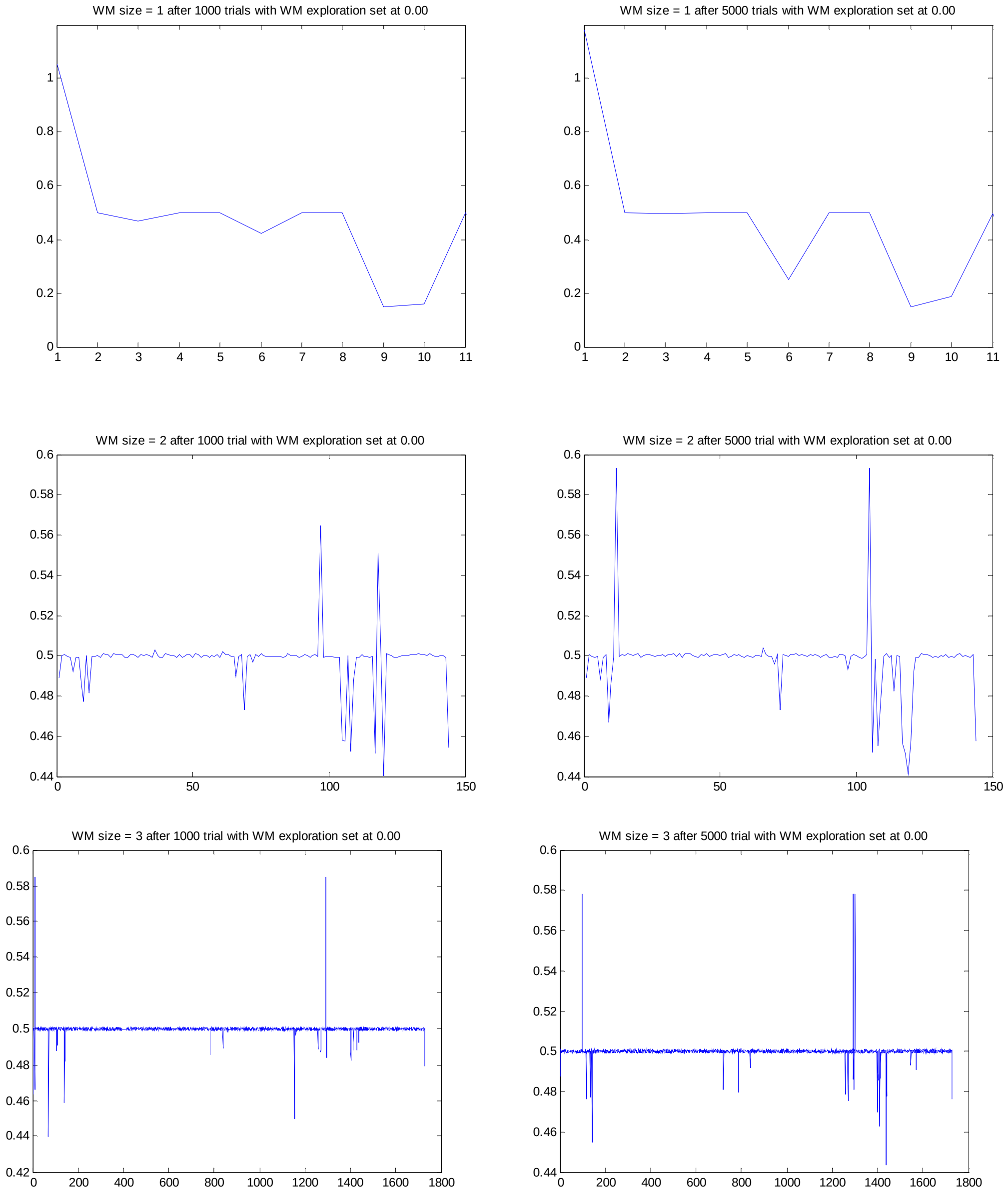


Fig 7.11. The weight distribution at episode 1000 with WM exploration set to 0.00 without a distractor

For working memory sizes larger than one, it can be seen that there are two distinctive weight peaks. The combinations of percepts that correspond to these peak weights are listed in Table 7.4, 7.5, 7.6 and 7.7. From the tables, it can be seen that one peak for each case corresponds to the conjunctive code for the images of a target location, typified by Object1 (BlackStripe), and the other peak corresponds to the non-target locations, typified by either Object9 (WhiteFloor) or Object10 (YellowFloor).

Table 7.4 WM exploration set at 0.00

WM size	1000 trials		5000 trials	
	Peak 1	Peak 2	Peak 1	Peak 2
1	Object1	NA	Object1	NA
2	Object1	Object10	Object1	Object9
	Object9	Object10	Empty	Object9
3	Object1	Empty	Object1	Empty
	Object1	Object9	Object1	Object9
	Object10	Object9	Object9	Object9

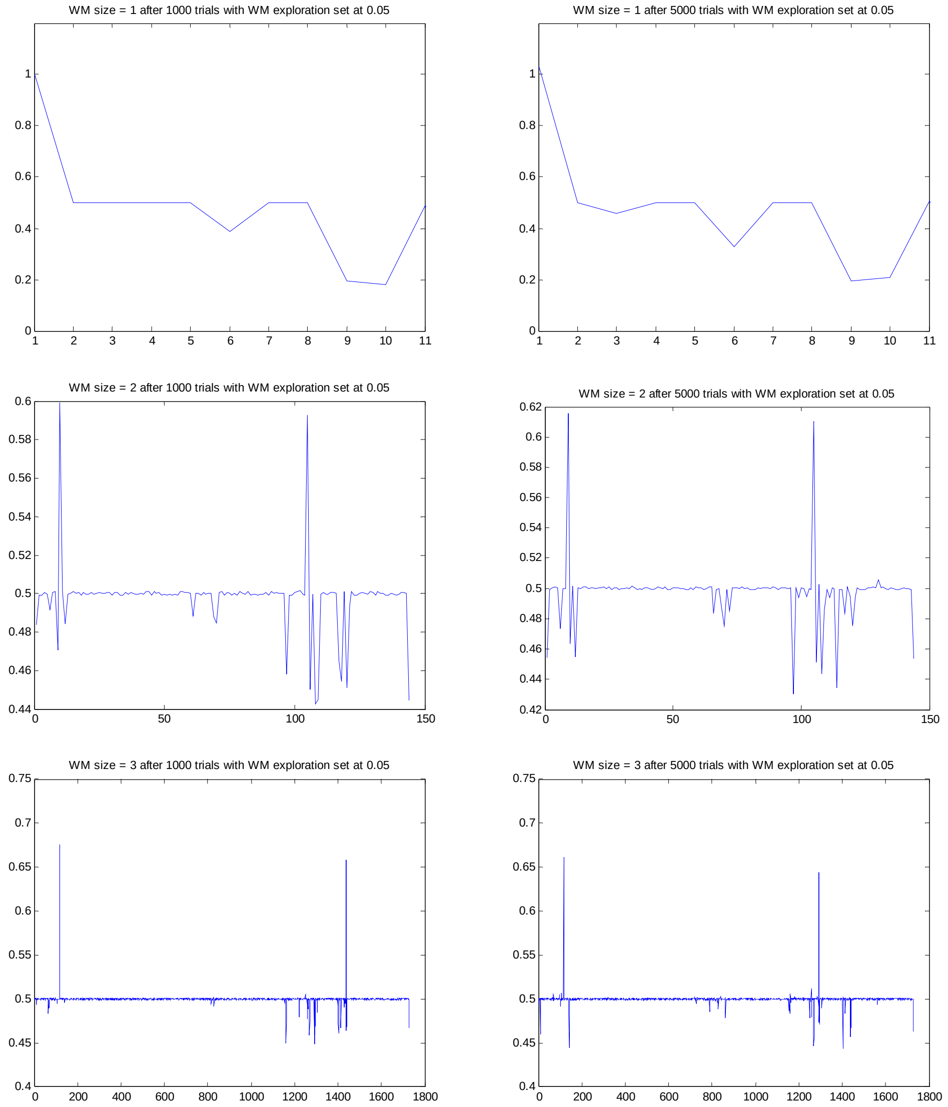


Fig 7.12. The weight distributions at episode 1000 with WM exploration set to 0.05 without a distractor

Table 7.5 WM exploration set at 0.05

WM size	1000 trials		5000 trials	
	Peak 1	Peak 2	Peak 1	Peak 2
1	Object1	NA	Object1	NA
2	Object1	Object9	Object1	Object9
	Object10	Object9	Object9	Object9
3	Object1	Empty	Object1	Empty
	Object10	Object10	Object9	Object9
	Object10	Object10	Object10	Object9

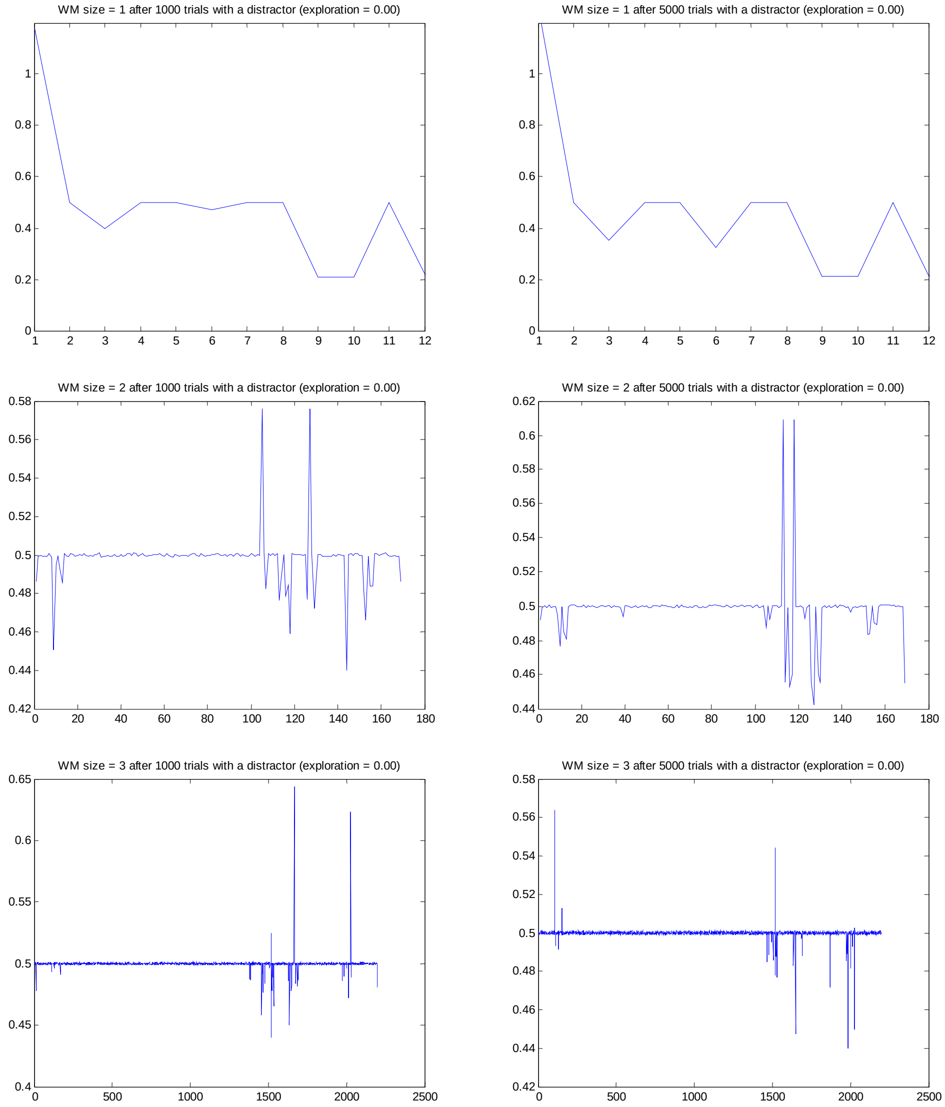


Fig 7.13. The weight distribution at episode 1000 with WM exploration set to 0.00 with a distractor

Table 7.6. WM exploration set at 0.00 with a distractor

WM size	1000 trials		5000 trials	
	Peak 1	Peak 2	Peak 1	Peak 2
1	Object1	NA	Object1	NA
2	Object1	Object10	Object1	Object9
	Object9	Object10	Object10	Object9
3	Object1	Empty	Object1	Empty
	Object10	Object12	Object1	Object9
	Object12	Object12	Object9	Object9

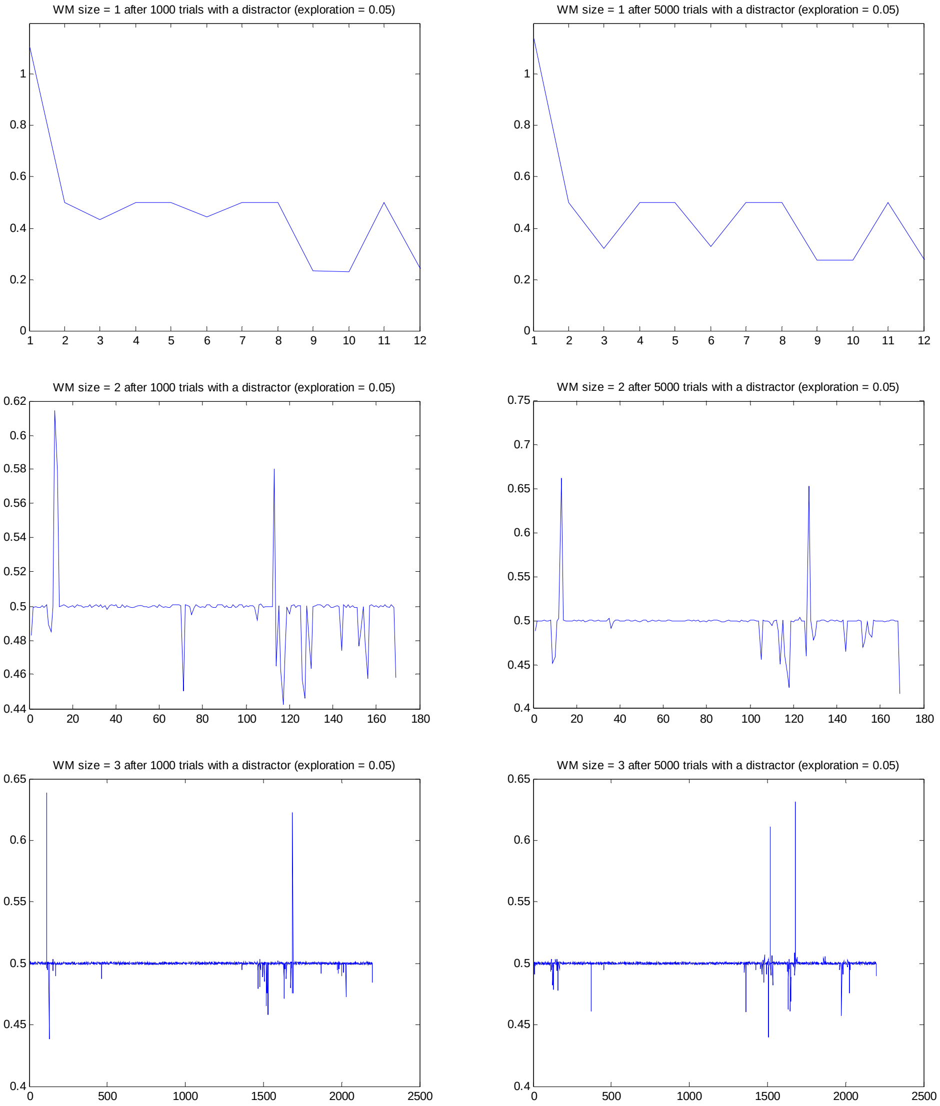


Fig 7.14. The weight distribution at episode 1000 with WM exploration set to 0.05 with a distractor

Table 7.7 WM exploration set at 0.05 with a distractor

WM size	1000 trials		5000 trials	
	Peak 1	Peak 2	Peak 1	Peak 2
1	Object1	NA	Object1	NA
2	Object1	Object9	Object1	Object10
	Object12	Object9	Empty	Object10
3	Object1	Empty	Object1	Empty
	Object9	Object10	Empty	Object9
	Object10	Object10	Object10	Object9

From the results of the second experiment set, i.e., with a distractor, the WMtk performs very well to ignore the distracting feature.

In order to better see the performance of the system, a simple two-class perceptron is used as a comparison. The results are summarized in Table 7.8 and Fig 7.15 and Fig 16. It can be seen that the two-class perceptron has done a better job than the WMtk when the working memory size is set to 1. However, apparently, the WMtk with a size larger than one has superior performance and conjunctive coding is more powerful than a single feature representation.

Table 7.8 Percentage of correct location classification using single perceptron

	Single Perceptron without distractor		Single Perceptron with distractor	
# of trials	1000 trials	5000 trials	1000 trials	5000 trials
% of correctness	87.70%	96.80%	86.50%	97.26%

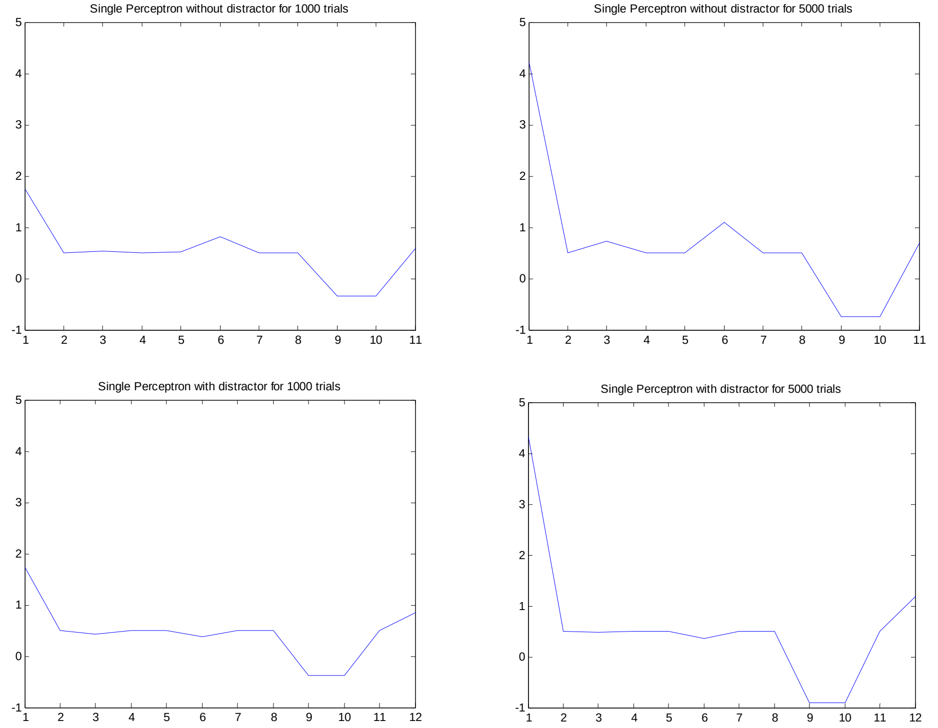


Fig 7.15. The weight distributions for two cases without (upper) and with (lower) a distractor

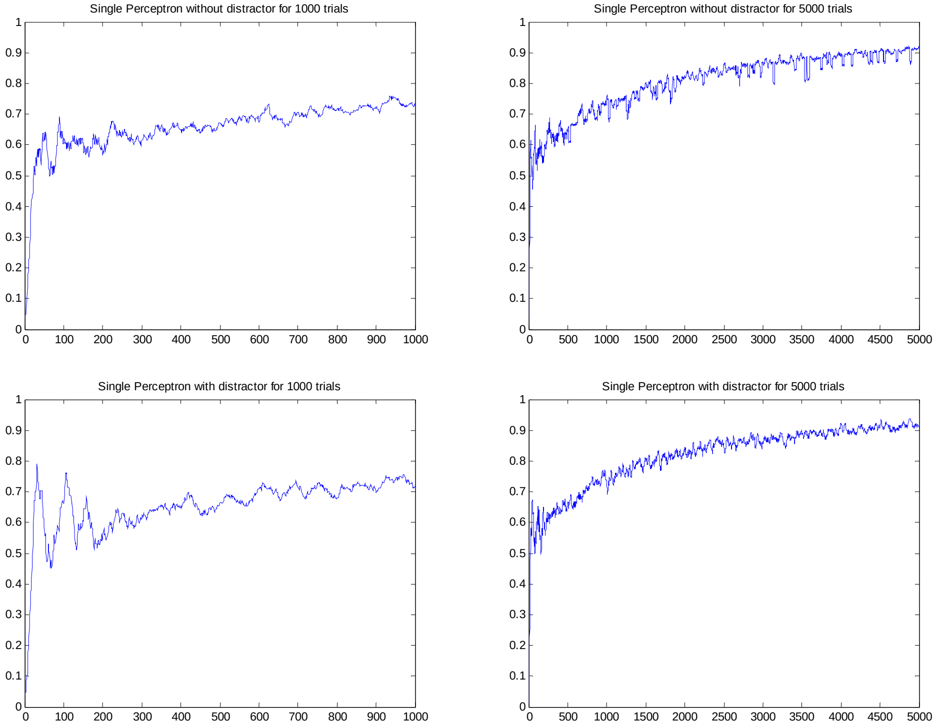


Fig 7.16. 25 Point convolution of correct classification without (upper) and with (lower) a distractor

Experiment 4 Results

To justify the proposed novelty detection methods are realistic and applicable in practice, two sets of experiments in real indoor environments are conducted. In the experiments, it is assumed that only one novel object exists and the novel object is a green ball. In our approach, a SONY Digital Handycam DCR VX2000 digital video camera recorder was used.

On-line novel object detection

The first step in this experiment is to find a suitable threshold. To do so, 80 normal pictures without the novel object in are collected. For each image, the standard deviation associated with the median is calculated as,

$$std = \frac{\sum \left| d_{FV} - d_{median} \right|}{\# FV} \tag{7.1}$$

where d_{FV} denotes the distance between each feature vector (*FV*) and the database tree in the approximate nearest neighbor sense, d_{median} denotes the median of these distances and $\# FV = 3{,}337$ is the total number of feature vectors used per image. This results in multiple medians and multiple standard deviations. The threshold *T* is calculated as the following sum,

$$T = median(d_{median1}, \Lambda\ , d_{medianN}) + median(std_1, \Lambda\ , std_N) \tag{7.2}$$

The threshold value is chosen to be one standard deviation away from the median of the medians, and is shown in Table 7.9.

Table 7.9 Threshold value used in on-line novel object detection

items	median	standard deviation	threshold
values	0.4017	0.2463	0.6480

Then, moving along the hallway, a video, consisting of 19 images with the novel object, is taken as the robot approaches the novel object. In the meantime, each grabbed image is processed and the calculated approximate nearest neighbor distances are thresholded to generate a binary image of size 47-by-71. Next, the lower 25-by-71 part of the processed image is eroded twice by an 8-connected structure element as mentioned in the above. Finally, the largest connected group is identified. When the robot comes nearer and nearer to the novel object, the size of the largest connected group grows. If the number of feature vectors it contains is larger than 100, they are retained and accumulated through temporary storage. As found in the experiment results, the total amount of the added novel feature vectors is no more than 600, 0.91% of the original database.

There are two ways to test the performance of this search tree based novel object detection technique. One method uses the first 18 images out of the total 19 as the training images and the last one as the testing image. The other method uses the total set of 19 images as both the training and the testing images.

In Fig 7.17, the novelty detection results are presented by six of the original 19 pictures. As the number of outliers starts to deviate from the normal images, the novel object is observed in the newly captured images. About 600 feature vectors coming from the novel object were added to the search tree after 2 seconds. Experimental results with these real images show that the system is able to continue learning with only a modest processing delay.

Off-line novel object detection

To obtain stable and permanent novel percept clusters in the database, 10 images are taken. To extract the perceptual feature vectors associated with the novel object, two kinds of distance are calculated for each image during off-line operation. To calculate the first kind of distance, the exact nearest neighbor search is done between each feature vector in the newly collected image and those in the database. To calculate the second kind of distance, a minimum spanning tree

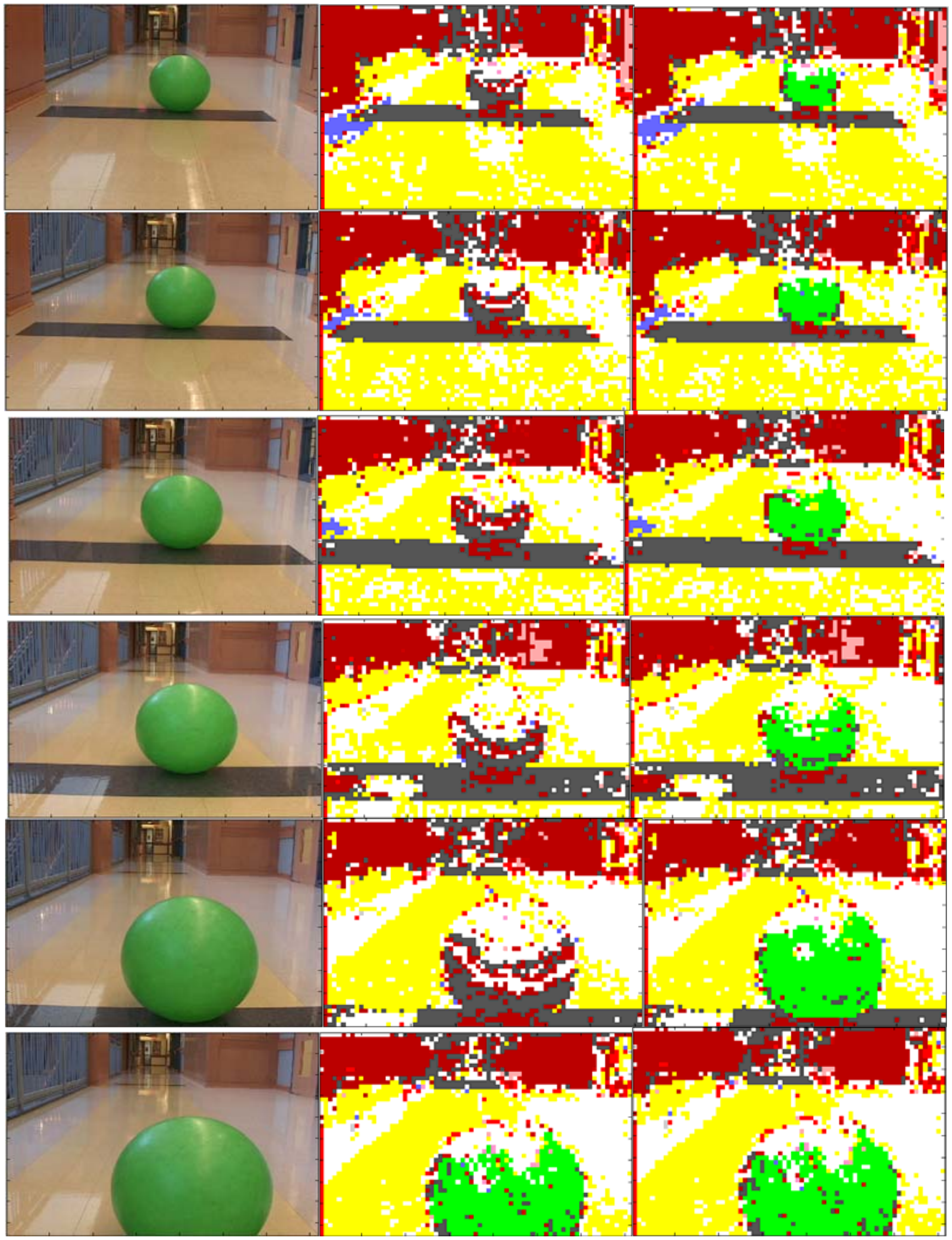


Fig 7.17 (left) the original images, (middle) processed images, (right) processed images after learning.

(MST) is built for each newly collected image and the edge distances of the tree are sorted in an increasing manner. The goal of obtaining these two kinds of distance is to filter out most normal feature vectors and retain only potential candidates for further processing. To do “low-pass” filtering, starting from the smallest edge distance in the MST, for each feature vector, if its smallest distance to the database is smaller than its edge distance in the MST, it is believed to be a normal feature vector and is removed from further consideration. By the term, normal feature vector, we mean one corresponding to a previously learned percept. On the other hand, starting from the second smallest edge in the MST, if its smallest distance to the database is larger than its edge distance in the MST, however, the feature vector connected to by this edge in the MST has been a normal feature vector, the feature vector is believed to be a normal feature vector. After this filtering operation is applied to all the collected new images, the obviously normal feature vectors have been removed and those left are likely candidates for a novel percept. These retained feature vectors are used to form a second MST. To form clusters from this new MST, edges are sorted and cut off in decreasing order until several clusters appear. If there is a significant difference between each cluster’s average distance to the database, the cluster with highest average distance is believed to be associated with the novel object. This process is continued until satisfaction. Though this approach is time-consuming, it is accurate and stable. Following this algorithm, 1,801 new feature vectors, labeled Object 12, are added into the original database. To evaluate this approach, two test images are used as shown in the left side of Fig 7.18. The two processed images in the middle of Fig 7.18 are obtained using pure nearest neighbor search and the two in the right of Fig 7.18 are the results using approximate nearest neighbor search tree constructed from the updated database. The robot has successfully and autonomously learned the novel percept.

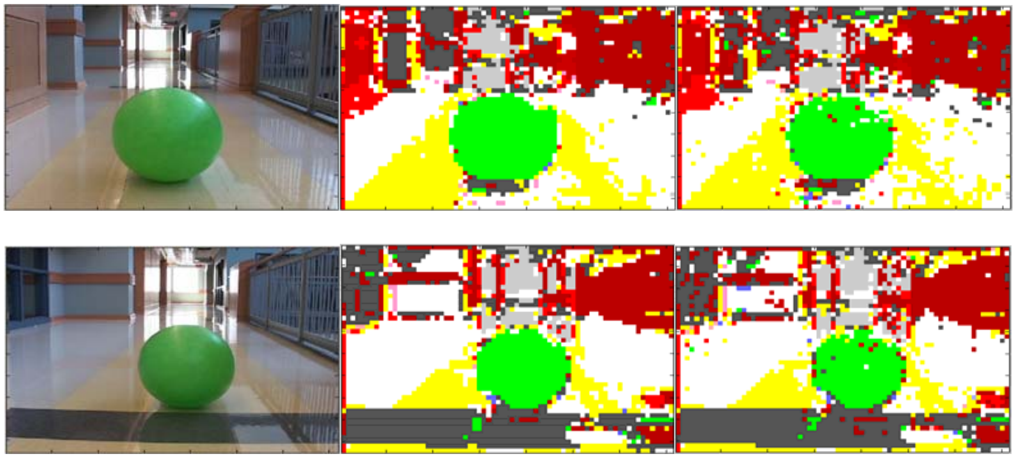


Fig 7.18 (left) original images, (middle) pure nearest neighbor search results, (right) search tree results

Conclusion

In this research, the basic issue of a vision-based autonomous mobile robot system has been addressed by integrating the working memory system into a vision-based mobile robot perceptual system. Particularly, an autonomous perceptual system has been addressed from the feature vector generation, similarity measure selection for natural association, to unsupervised clustering for database construction and fast search implementation using tree structure. It seems that the mixed similarity metric performs better than the Euclidean metric and that fast segmentation with respect to the database search tree works reasonably well. The working memory toolkit, as a biologically inspired model of the PFC and midbrain dopamine system, has been incorporated into the robot control system and validated in the context of a real-world working memory task.

Further, an application of the transverse patterning problem to landmark identification has been explored using the WMtk and a single perceptron. The specific scenario that we are studying assumes that there is a target location that the robot is able to recognize visually. Based on the experimental results, it can be concluded that conjunctive coding using the WMtk can do a better job than perceptron even with the appearance of a distracting percept. This is particularly useful for tasks that require visually-stable landmarks be identified and tracked so as to build a map of the environment. However, questions exist with regard to how to find a reward mechanism so that the peak weights correspond to combinations of non-empty distinctive percepts.

Finally, two non-conventional approaches for novel object detection are proposed. The online approach is fast but may be less reliable, the offline approach is more reliable but computationally expensive. This suggests that the latter method may be used as a so-called reflective process, i.e., run when the robot is not occupied with a task. It has been shown that both approaches can provide an accurate and robust solution to the novel object detection problem in applications. Therefore, the two novel object detection methods are complementary. Experimental results suggest that the proposed novelty detection methods provide a basis for continuous learning, which is important to supporting developmental robotics.

Future Work

Future work includes whether a more accurate cluster selecting criterion should be used, and, finally, whether a better unsupervised method exists for clustering the feature vectors based on the mixed metric. For the application of the transverse patterning problem to landmark identification, questions exist with regard to how to find a reward mechanism so that the peak weights correspond to combinations of non-empty distinctive percepts. As to novelty detection, the computational efficiency and scalability of both methods inspire our confidence to generalize the proposed methods to multiple novel object detection in future work.

APPENDIX A

HUMAN VISUAL SYSTEM

The brain is an incredibly complex structure. It consists of cells called neurons. Neurons are complicated in structure as well, but may be approximated as shown in Fig A.1. The cell receives input from other cells through the dendrites. The input signals tend to activate the cell, and if it is sufficiently activated, it fires. This means that it produces an output which travels along the axon and is passed to the dendrites of other cells through connections called synapses or foot. The elaborate branching structure of the dendrite tree allows a neuron to receive inputs from many other neurons through synaptic connections.

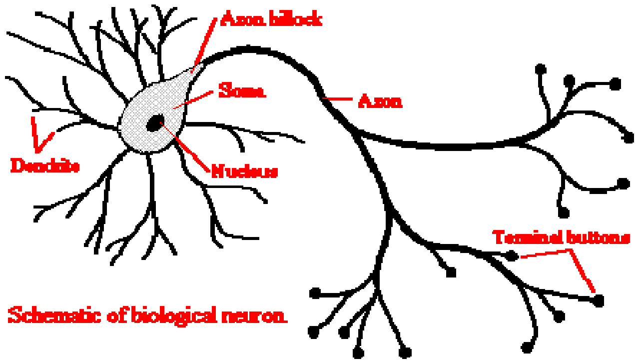


Fig A.1 A simplified model of a neuron

The brain contains around 10^{11} neurons, which have more than 10^{15} synapses and at least 2000 miles of axonal connections [18]. However, as we will see, there are several rules of organization that simplify the explanation of how the brain works. First, neurons with similar patterns of connections and response properties are clustered together to form areas which seem specialized to perform different tasks. Second, many of these different areas themselves are subdivided into smaller processing units such as columns, which is a common feature within the visual system. Finally, there is a duplication of visual areas on either side of brain, a feature called lateralization which allows the brain to carry out a greater variety of tasks with a limited amount of brain tissue.

Vision is the primary sensory modality in human. The human visual system starts from the eye. Its basic structure is shown in Fig A.2. The basic function of the eye is to catch and focus light from the target stimuli onto the retina at the back of the eye. Once the image has been focused on the *retina*, this pattern of light will be transformed into a pattern of neural activity that can accurately represent the image. The retina consists of three main layers: the receptor cell layer, the bipolar cell layer and the ganglion cell layer. The photo receptors form synapses with bipolar cells which in turn synapse onto ganglion cells, whose axons travel through the optic nerves and leave the eye to the brain at a point called the optic disk. There are also two other layers in the retina: the outer plexiform layer, containing horizontal cells, and the inner plexiform layer, containing amacrine cells. These cells transmit information in a direction parallel to the surface of the retina and so combine or subtract messages from adjacent photoreceptors. Therefore, the retina behaves like a high pass filter, emphasizing changes and discontinuities in illumination, and sending the contrast between light and dark spots required for the formation of precise images to the brain via the optic nerve.

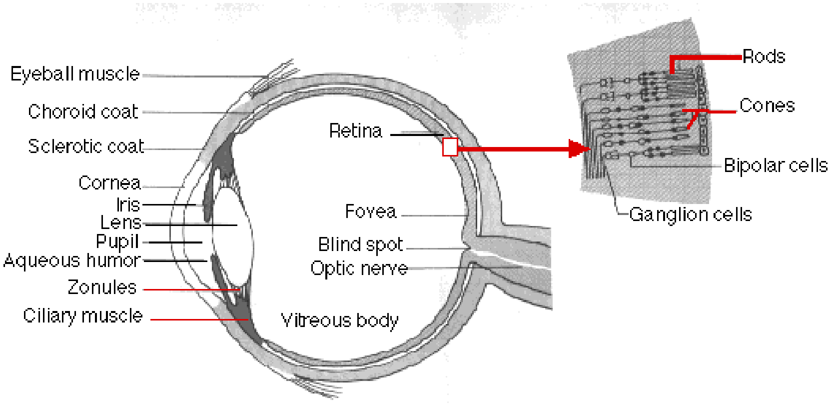


Fig A.2 Cross Section of Human Eye

The transformation or transduction of the light energy into a neural signal is carried out by the light-sensitive receptor cells (photoreceptors) in the retina. There are two types of photoreceptor: the cones and the rods. The cones are concentrated in a small area of the retina called the fovea. They mediate diurnal visual function and provide high-acuity color vision. In contrast, the rods mediate nocturnal vision and provide only low-acuity monochrome vision. Therefore, the range of possible light intensities is divided between these two photoreceptors, with the cones responding to high intensities and the rods to low intensities. There are around 120 million rods and 6 million cones in the human retina, distributed as shown in Fig A.3. Each of the 126 million photoreceptors signals information on how much light is absorbed at a particular point in the retina. However, it is at the level of the retinal ganglion cells that the information relating to retinal regional light differences is extracted. Each ganglion cell is connected to a number of photoreceptors via bipolar cells and its activity is altered by the stimulation of the retinal area corresponding to these photoreceptors. This retinal area is called the ganglion cell's *receptive field*. The photoreceptors in a particular receptive field are arranged in what is called a center-

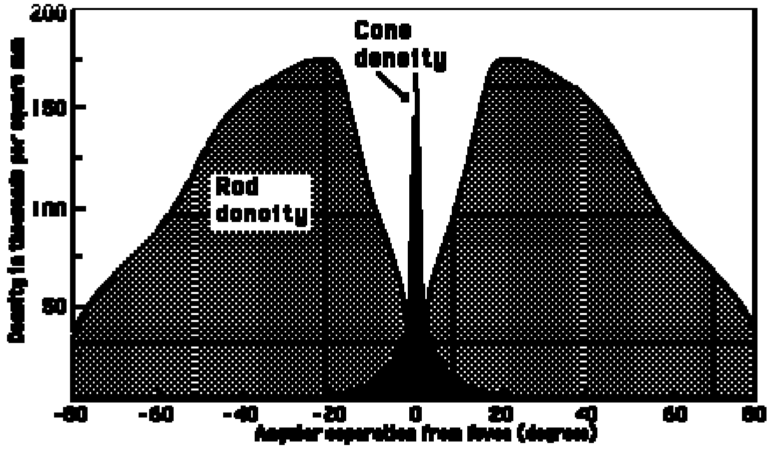


Fig A.3 Distribution of cones and rods around fovea

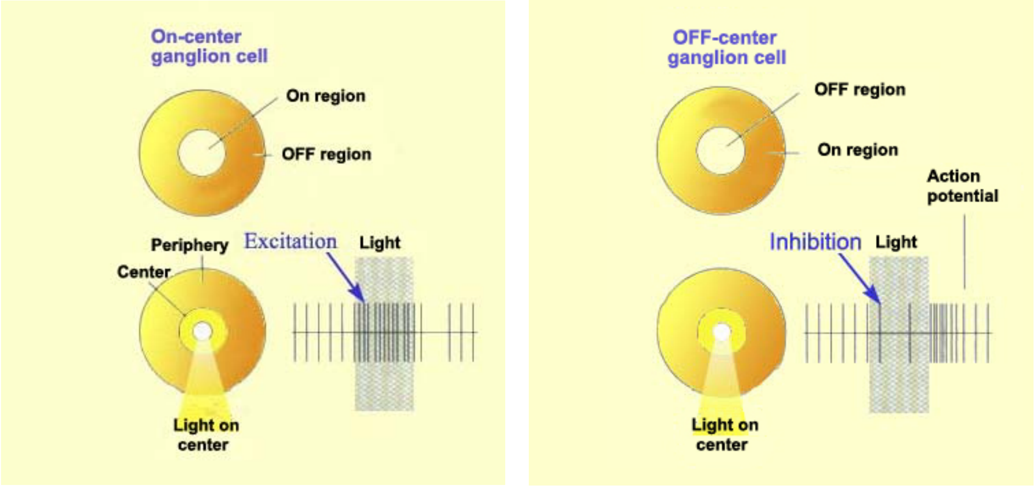


Fig A.4 Center-surround receptive fields

surround organization. There are two kinds of ganglion cell's receptive field, ON-center, OFF-surround cells and OFF-center, ON-surround cells, as shown in Fig A.4. In the first case, light falling on the center of the receptive field might result in the corresponding photoreceptors stimulating the ganglion cell (an ON response), while light falling just on the surrounding ring of photoreceptors might inhibit the ganglion cell (an OFF response). The opposite is true for the other case. When a whole receptive field is illuminated, both the ON-center and the OFF-surround are stimulated. Since the ON-center excites the ganglion cell, but the OFF-surround inhibits it, there is little change in the ganglion cell's response. If an edge passes through the center of a receptive field, the ON-center receives an increase in light and will stimulate the ganglion cell, while the OFF-surround receives a reduced level of light and will not inhibit the cell as much as before. As a result, the ganglion cell is stimulated and will signal the presence of a light/dark boundary. Because the receptive fields of the ganglion cells are usually concentrically arranged, to detect the orientation of the edge, the responses of a number of ganglion cells should be compared (this comparison is made in the first cortical visual area).

For retina color vision to be possible in humans, there are three types of cones: the blue, red and green cones. Blue cones are between 5 and 10% of the total cone population and form a ring or annulus around the edge of the fovea. The rest are red and green cones which are randomly mixed together in small patches and clusters [110]. The blue or short-wavelength pigment absorbs maximally at 445nm, the green or middle-wavelength pigment absorbs maximally at 535nm, and the red or long-wavelength pigment absorbs at 575nm [111]. The sensitivities of these three cones to light wavelengths are shown in Fig A.5. In 1802, Thomas Young correctly proposed the trichromatic (three color) theory, which suggested that, for a human observer, any color could be reproduced by various quantities of three colors selected from various points in the spectrum. The responses from the three different cone classes are compared to allow color discrimination in an opponent manner. The basis of the opponent mechanism is: given that the receptive fields are very small in the fovea and comprise a comparatively small number of randomly arranged red and green cones, there will be an unequal number of these two cones in the center and the surround, resulting in color opponency. In other words, they give opposite responses to different parts of the spectrum in different parts of their receptive field. For example, the center might have an ON-center for green and an OFF-surround for red, and the opposite set of responses exist in the surround.

Even though there are around 126 million photoreceptors, only 1 million ganglion cells exist to carry the information from the retina to the brain via their axons, suggesting a general convergence factor of 126:1. However, in reality, this convergence factor is not constant but varies with retina eccentricity. At the center of the fovea (where the cones dominated), the convergence factor may be as low as 1:1, and the sizes of the ganglion cell's receptive field are at their smallest. As one moves out into the periphery (where the rods dominate), the convergence factor may be as high as several hundred to one and the receptive field size rapidly increases. The degree of convergence determines the spatial resolution of that part of the retina. The resolution (measured as visual acuity) refers to the ability to distinguish differences in the spatial

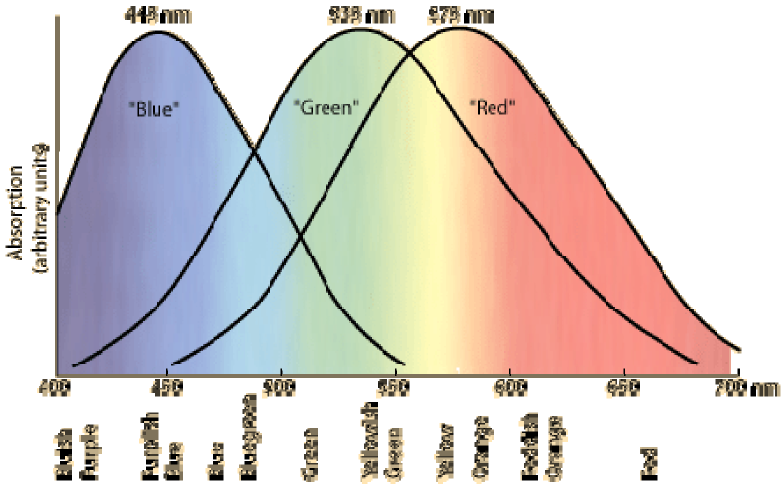


Fig A.5 The color sensitivities of cones

distribution of light in the image. Therefore, at the center of the retina, spatial resolution is at its best. From the above, we can see that the basic purpose of the retina is not just to signal the presence or absence of illumination, but rather to condense and reorganize the information from the photoreceptors into patterns of neural signals from which information relating to the identity of objects and their spatial relationships in the environment can be derived.

Moving up the visual system, from the retina to the lateral geniculate nucleus (LGN) and then on to successive cortical areas, visual neurons become responsive to more and more complex stimuli. But the visual system is not organized in just a serial, hierarchical pathway. Instead, different aspects of a stimulus (such as shape, color and motion) are analyzed in separate, parallel pathways. These pathways are usually divided into two broad categories: ‘what’ and ‘where’ pathways, with the ‘what’ pathway dealing with information about the stimulus features (such as shape and color) and the identity of an object, and the ‘where’ pathway dealing with spatial information about an object (such as motion and form derived from motion).

The division of the visual system into two or more separate pathways starts to become evident at the level of the retina. There are several forms of ganglion cells in the primate retina. The P cells (sometimes called B or Pβ cells), constituting about 80% of the cells, are selective for wavelength and high spatial frequencies and have slow sustained (tonic) responses. The M cells, constituting about 10% of the cells, are not wavelength sensitive but are sensitive to low spatial frequencies, have transient (phasic) responses and have faster conduction velocities. The subsequent pathways from these two cell classes are often referred to as the M or P pathways. At any one eccentricity, the dendritic field of the M cell is three times larger than that of the P cell, which shapes the functions of the subsequent visual areas.

As shown in Fig A.6, the optic nerves or fibers, formed by the axons of all of the ganglion cells coming together, pass out of both eyes through the optic disc, join together and cross over each other at the optic chiasm, and project to the dorsal *LGN*, which can be found on each side of the brain. As a result, each hemisphere of the brain receives information from the opposite side of the visual scene. The LGN consists of six layers. Each layer not only receives input from only one eye (layer 2, 3 and 5 from the eye on the same side as the LGN, and layer 1, 4 and 6 from the eye on the opposite side), but also contains a complete map of the retina. The inner two layers, layers 1 and 2, contain larger cell bodies than those in the remaining four layers and receive input from M ganglion cells, and are therefore called magnocellular or M layers. The outer four layers input from type P ganglion cells and are called parvocellular or P layers. Since there are approximately 1.5 million LGN neurons, they show response properties very similar to the cells from which they receive input.

The LGN neurons mainly project to *the primary visual cortex* which lies posteriorly in the occipital lobe. Also known as striate cortex or V1, the primary visual cortex is the first cortical visual area. The topography of the visual field is preserved in the projection to V1 from the LGN,

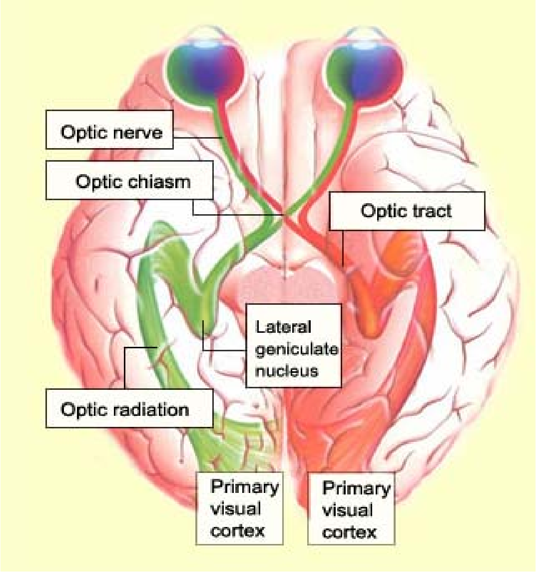


Fig A.6 From retina to primary visual cortex

with relatively more cortex being devoted to the fovea than to the peripheral visual field. However, it is a much larger and more complex structure than the LGN and is composed of around 200 million cells compared with 1.5 million cells for LGN. Incoming bundles of fibers form clear stripes in this area, hence the name striate cortex. It consists of six principal layers (and several sub layers) which are arranged in bands parallel to the surface of the cortex. As shown in Fig A.7, the axons from the LGN cells terminate on cortical neurons in layer 4. This layer is subdivided into three sublayers, A, B and C. Sublayer C is further subdivided into 4Cα and 4Cβ. The neurons in the magnocellular layers send their axons to neurons in layer 4Cα and the lower part of layer 6. This information is then relayed to layer 4B and from there to visual cortex area 2 (V2), visual cortex area 3 (V3) and visual cortex area 5 (V5). The neurons in the LGN's parvocellular layers send their axons to the neurons in sublayer 4A and 4Cβ and the upper part of layer 6. The cells in layer 4Cβ then send their axons to layers 2 and 3 and from there to V2. The separation of inputs from the two eyes is preserved in layer 4. The cells from one LGN layer will project to groups of target cells in layer 4C, which are separate from those supplied by the other eye. These cells group to form alternating stripes or bands in layer 4C. Above and below this layer, most cells are driven by both eyes, although with one eye usually being dominant.

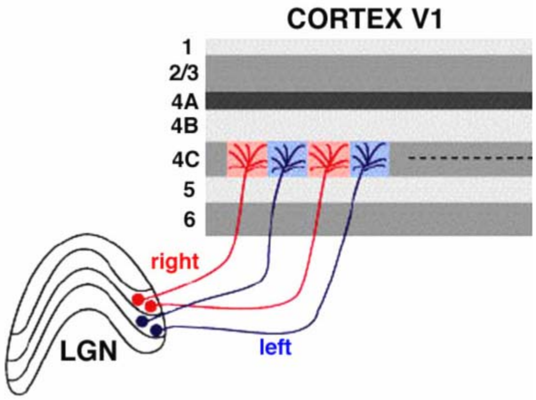


Fig A.7 Projection of LGN cells to layer 4 of V1

According to the shapes of the *cortical receptive fields*, there seem to be two broad categories of neurons in V1: simple neurons and complex neurons. In addition, there is a class of cell found exclusively in layer 4C that has a concentric center-surround receptive field similar to that of the LGN and the retinal ganglion cells.

Simple cells are found mostly in layers 4 and 6, which receive direct inputs from LGN. One type of simple cell has a receptive field that contains an extended narrow central portion flanked by two antagonistic areas. The center can be either excitatory or inhibitory. Such cells have an optimal activation by a bar of light that is no more than a certain width, but entirely fills the center area, and is oriented at a certain angle. The symmetry of the receptive field as well as the preferred orientation of light and dark bar varies with different cells. Another type of simple cell has a receptive field that may just consist of two longitudinal regions facing each other, one excitatory and one inhibitory. There is also a situation where the cell is sensitive to the length of the stimulating bar. For such a case, there seems to be an additional antagonistic area at the top or bottom of the receptive field, and the optimal stimulus is an appropriately oriented bar or edge that stops in a particular place. These cells are named end-inhibited or end-stopped cells.

Complex cells are most found in layers 2, 3 and 5. Like simple cells, these cells require a specific orientation of a dark-light boundary. But there are no longer distinct excitatory and inhibitory areas. There are two main classes of complex cell both of which respond best to moving edges or slits of fixed width and precise orientation. One type of cell responds best to a stimulus bar of a particular length, while the other type responds best to end-stopped stimuli. The best stimuli for these cells not only include a certain orientation but also a discontinuity, such as a line that stops, an angle or a corner.

Unlike the M pathway being preserved in V1, the P pathway splits into two new pathways in the upper layers of V1. One is called the P-B pathway which seems to deal primarily with color. The other is called the P-I pathway, the neurons of which are sensitive to features such as the orientation of the stimulus and seem to mediate high acuity perception.

In V1, there is an arrangement of columns called blobs that extend through layers 2 and 3 and through layers 5 and 6 [112]. The areas surrounding the blobs are referred to as the interblob region. The cells in the blob region are either color or brightness selective but not orientation selective. They receive input from the color-opponent P cells in the LGN, although their receptive field centers are larger and their color coding is double-opponent. In other words, they give opposite responses to different parts of the spectrum in the different parts of their receptive field. For example, the center might give an ON-response to green and an OFF-response to red with the opposite happening in the surround. These cells are therefore part of the P-B pathway. In the interblob region, most cells do not show color-coded responses but respond well to achromatic luminance-contrast borders, and thus stimuli of a particular orientation, such as lines or bars, which are formed by color contrast and used to identify borders. The cells in the interblob region are therefore part of the P-I pathway. From the above, it is easy to see that the blob and interblob systems seem to work in different but complementary ways. Further, both the parvocellular layers and the magnocellular layers send inputs to the blobs. Thus the separation into P and M pathways first observed in the retina is preserved in V1.

The striate cortex is an important area. Partially processed information from the retina and LGN is separated here and packaged up for more elaborate analysis in the specialized visual areas of the extrastriate cortex. Most neurons in V1 respond to more complex stimuli and have very different response properties from those of neurons in the preceding area. Because of this, new response features, such as sensitivity to lines and bars of different orientations and movements, appear along with a specialization of some neurons to an existing visual feature such as color.

Although many neurons in V1 do respond to lines and bars as just mentioned, the optimal stimulus is a sine-wave grating which is classified by its spatial frequency [113]. 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.

There is a class of neurons in V1 which are unresponsive to single lines, bars, or edges but seem to be responsive to texture [114]. Responding preferentially to a sine-wave or square-wave grating of a particular spatial frequency and orientation, these cells do not seem to be frequency analyzers but, instead, respond to the ‘texture’ of the pattern. Nature surfaces have a rough texture. Being sensitive to texture and texture orientation, these cells can not only detect the presence of a surface but also its orientation.

There are also about 10-20% of complex cells in the upper layers of the striate cortex which show a strong selectivity for the direction in which a stimulus is moving. The cell responds strongly to movement in one particular direction, but is unresponsive to movement in other directions. However, other complex cells do not show such a marked direction preference.

As to color, the blobs contain color-opponent and double color-opponent cells, but the neurons will be either red/green opponent or blue/yellow opponent within a blob. The two forms of opponency are not mixed within a single blob [115]. In other words, individual blobs are dedicated to the processing of one color-opponent system. Since there are more red and green cones in the retina than blue cones, different types of blobs are not equally represented in V1. There are more red/green blobs than blue/yellow blobs. The blobs often seem paired by bridges which have been found connecting blobs of different color opponency. However, the responses of neurons in the bridges are neither red/green nor blue/yellow but are mixed in spectral selectivity. To summarize, the cell in V1 is responding not to the perceived color but to the wavelength composition of a surface.

As mentioned before, there are around 200 million cells in V1. These cells are organized into repeating columns and processing modules, which seems to be a standard pattern in all cortical visual areas, due to its efficiency of mapping a multi-dimensional stimulus, such as vision, onto an irregularly shaped piece of two-dimensional cortex. More specifically, each module receives information from other modules, performs some calculation and then passes it on to other modules. The neurons in each module are involved with the analysis of a particular portion of the visual field. As shown in Fig A.8, the modules consist of two units, each of which is centered around a blob. Since blob cells are sensitive to wavelength and interblob cells are sensitive to orientation and, in the case of the complex cells, movement as well, each module contains neurons sensitive to wavelength, movement and line or edge of particular orientation within a specific portion of the visual field. Within a module the response characteristics of cells are arranged systematically as shown in Fig A.9. For example, if a neuron in one position responds best to a line oriented at 45 degrees, then a neuron a short distance away will respond best to a line oriented at 40 degrees, etc. After traveling across both halves of the module, there are two 180-degree rotations in the orientations of lines that best stimulate the neurons. Such an orderly progression of stimulus orientation in columns across the cortex seems to be an idealized version of cortical organization.

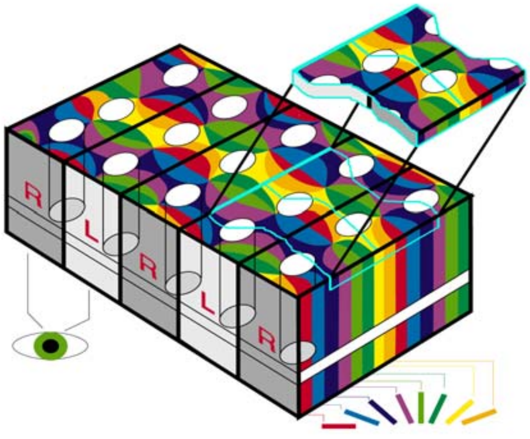


Fig A.8 The ‘ice-cube’ model

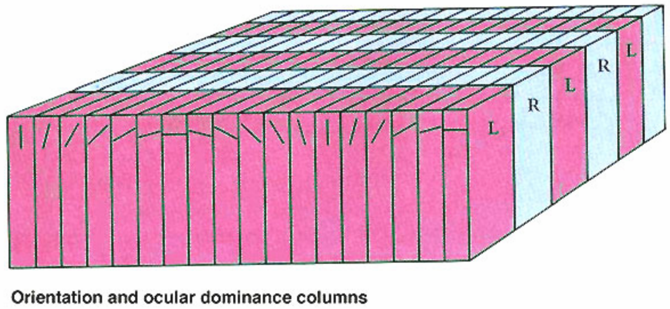


Fig A.9 The ice-cube model of the cortex

Regions of cortex adjacent to V1 are also concerned with vision. The main target of V1 is *Visual Area 2* (V2), which immediately surrounds V1, as shown in Fig A.10, and is sometimes called area 18. Even though V2 receives input primarily from V1, each area contains its own representation of the visual field projected in an orderly manner. In V2, there is not any pattern of blobs or interblobs, but instead a pattern of stripes running perpendicular to the border between V1 and V2 and extending over the entire 8 to 10mm width of V2. There seems to be three types of stripe: thick stripes, thin stripes, separated by interstripes. The neurons in the M pathway project to the thick stripes. As a result, the neurons in the thick stripes show response properties similar to the neurons in layer 4B and are orientation and movement selective. Many of them also show sensitivity to retinal disparity (the basis of stereopsis). The neurons in the blobs project to the thin stripes, whereas the neurons in the interblobs project to the interstripes. More than half the neurons in the thin stripes are color sensitive (mostly double-opponent). But none of them are orientation selective. The neurons in the interstripes are orientation selective but not direction selective, nor do they show color selectivity. It is easy to see that the M pathway projects from layer 4B of V1 to the thick stripes, the P-B pathway projects from the blobs of V1 to the thin stripes, and the P-I pathway projects from the interblob regions to the interstripes. Though the organization of V1 is retinotopic in the sense that the visual field of the retina is mapped onto the surface of the cortex of V1, there seems to be three visual maps in V2 [116]: a visual orientation map within the thick stripes, a color map within the thin stripes, and a disparity map within the interstripes. Adjacent stripes are responsive to the same region of the visual field, so these three visual maps in V2 are interleaved, each representing a different aspect of the visual stimulus. As shown in Fig A.11, it is from V2 that the two pathways project to different cortical areas. However, the separation of the P and M pathways should not be overemphasized. There is some

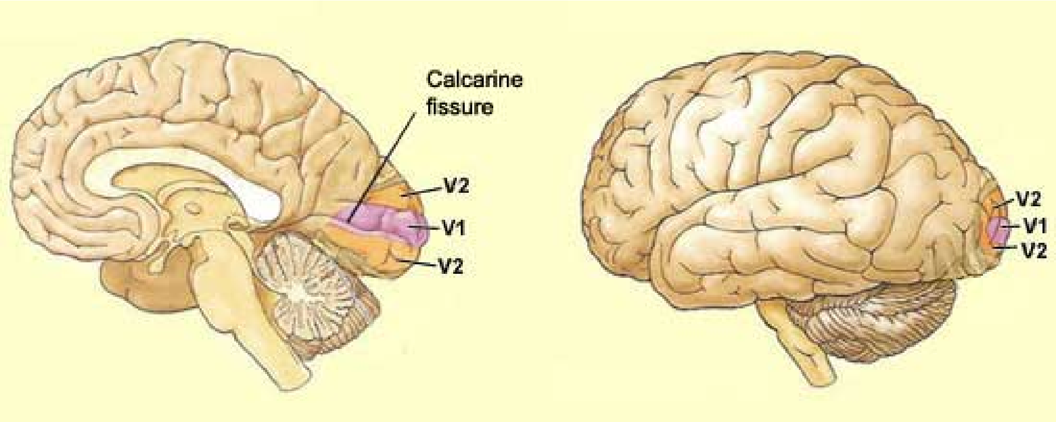


Fig A.10 Area V2

communication between the two pathways, though the M pathway does seem to be more segregated.

The M pathway projects to *visual area 3* (V3) and *visual area 5* (V5), both directly from layer 4B of V1 and through the thick stripes of V2. Most neurons in V3 are orientation selective and are thought to be involved in the processing of dynamic form. V5 is also known as the middle temporal visual area or MT. When motion is perceived, V5 is activated and is believed to process information on motion and stereoscopic depth [117][118]. Neurons in V5 are sensitive to the speed and direction of moving objects, and each neuron has a preferred direction of motion that stimulates its maximal response. Many of these neurons also respond to stereoscopic disparity, which corresponds to the visual plane in which the stimulus occurs. The M pathway then projects to the parietal cortex, which seems to be important for the integration of movement and depth into a representation of space.

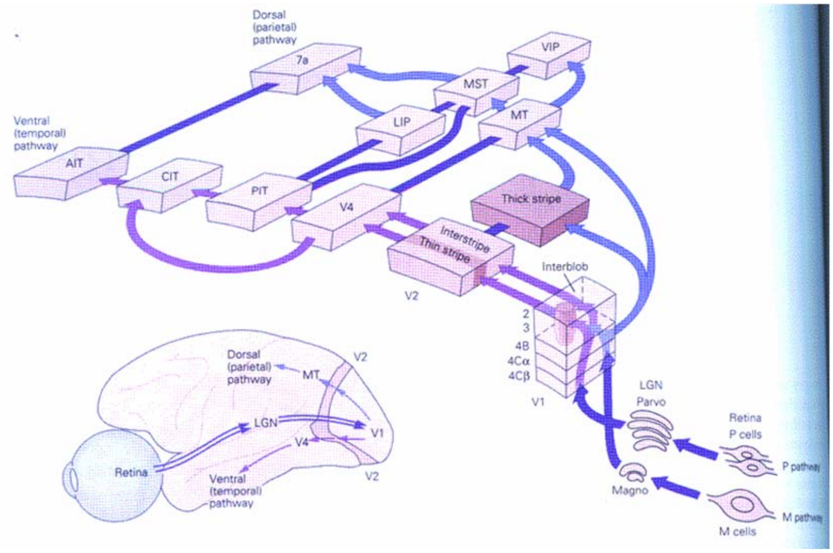


Fig A.11 The M and P pathways

For the P pathway, both the thin stripes (color) and interstripes (form) project to *visual area 4* (V4). The separation of the two subdivisions of the P-pathway continues here. The V4 cells in the P-B pathway are still sensitive to color but seem to display a higher sophistication of response than those neurons in preceding areas. These neurons respond not to the wavelength of light but to its 'color' [95], which is known as color constancy. The V4 cells in the P-I pathway are sensitive to simple shapes and objects. Therefore, V4 is also important for object discrimination. V4 projects primarily to the temporal visual cortex. There seems to be an integration of form and color to give a representation of complex objects in the temporal visual cortex. Neurons in this region are found to respond to complex patterns and objects, such as faces [119]. As we move up the object-processing pathway (V1-V2-V4-posterior IT-anterior IT), the receptive field of a cell gets significantly larger and the response properties of the neurons change. Most of these cells also have receptive fields close to, or including, the fovea (75% of anterior IT cells include the fovea). The increase in receptive field makes possible the development of a visual response that is unaffected by the size and position of a stimulus within the visual field. The cells also respond to more and more complex stimuli. For example, the majority of cells in V4 and in the posterior IT have been found to be sensitive to the 'primary' qualities of a stimulus, such as color, size or orientation, whereas cells in the anterior IT seem to be sensitive to complex shapes and patterns.

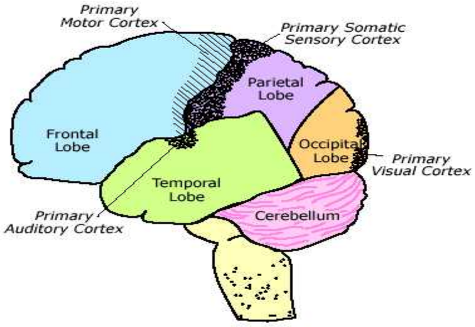


Fig A.12 The human cortex

As mentioned in the above, it has been proposed that visual information is processed in two broad systems: the ‘what’ system (also called the ventral system) and the ‘where’ system (also called the dorsal system). As shown in Fig A.13, the ‘what’ system, through which V1 projects to the temporal lobes, is concerned with the identification of an object, whereas, the ‘where’ system, through which V1 projects to the parietal lobes, is concerned with the relative spatial position of an object [120]. Damage to the ‘what’ system does not impair performance of visuospatial tasks but does impair performance of object discrimination tasks. Damage to the ‘where’ system produces impairments of visuospatial tasks but does not impair the performance of object discrimination tasks. The M and P pathways have been considered to correspond roughly to these two systems [121]. Being responsive to color and shape information and further to a representation of an object, the P pathway would seem to be ideal for the ‘what’ system. On the other hand, being responsive to motion, stereopsis and form derived from motion and further to a representation of spatial relationships in the visual field, the M pathway would seem to be the obvious candidate for the ‘where’ system. Like the M and P pathways, there is considerable communication between the two systems at all levels.

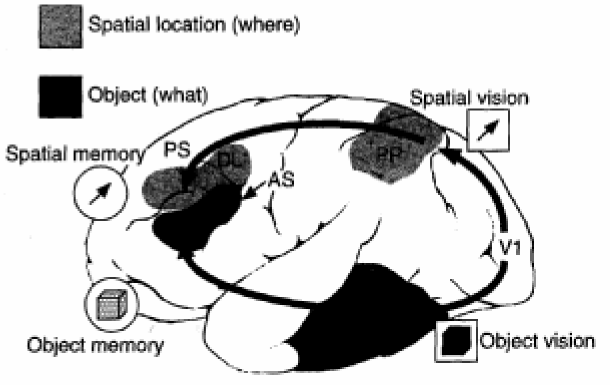


Fig A.13 The ‘what’ and ‘where’ systems

Both streams project to the prefrontal cortical areas. The prefrontal cortex is an important region for short-term working memory. The ‘what’ system projects to the inferior convexity (IC) ventrolateral to the principal sulus and the ‘where’ system projects to the dorsolateral prefrontal region (DL). Experiment results suggest that IC mediates working memory for objects and DL mediates spatial working memory. Although the visual system is composed of a vast number of neurons, its ability to process fully and store in memory distinct, independent objects is strictly limited. It has been suggested that objects must compete for attention and processing ‘space’ in the visual system, and that this competition is influenced by both automatic and cognitive factors.

Now we move from the posterior part, which is mainly sensory, to the anterior part, which is mainly motory. As a matter of fact, the whole cerebral cortex projects to the basal ganglia, the output of which funnels back to the frontal areas of the cortex. In contrast to posterior cortex being ‘sensor’ cortex, the entirety of the frontal cortex, in general, is ‘motor’ cortex devoted to organismic action of one kind or another, such as skeletal movement of various body parts, ocular movement, the expression of emotion, speech, or visceral control and so on, together with considerable functional cooperation between them. The cortex of the occipital, parietal, and temporal lobes largely supports perception and perceptual memory, while the cortex of the frontal lobe supports action and motor memory. Substantial evidence is available that the memory storage and processing of perceptions are functions of large networks of interconnected neurons of the posterior cortex. Particularly, though by mechanisms still poorly understood, the hippocampus seems to play an essential role in the processing of memory formation. Being made of neurons representing the various inputs that, by temporal coincidence, have become associated attributes of that memory, a network represents a memory or an item of knowledge. The facts that perception and memory share to a large extent the same network and that perceptions are processed through pre-established memory networks are the neural foundations for the intimate relationship between perception and memory; we see the world the way we know it, the way we have learned to see it [53][54][55]. At the root of it, the sensory information is processed through hierarchically ordered areas of the posterior cortex and the neuronal networks (memories) they harbor.

Similar to the posterior cortical areas, the frontal areas are hierarchically organized as well for representation and processing of action. At the lowest stage of this frontal hierarchy is the primary motor cortex. Above it lie the premotor areas, above which, in the frontal motor hierarchy, are the dorsolateral areas of the prefrontal cortex. Like perceptual memory, motor memory is also hierarchical and motor memories have an orderly representation in that hierarchy of frontal motor areas. The frontal hierarchy is not monolithic but is separated into different action domains, each providing the neural substrate for the representation and processing of a different category of movement.

In all instances prefrontal functions serve the purpose of ordering sequential action toward a goal, whatever that goal may be. In other words, the prefrontal cortex plays a fundamental role in the temporal organization of behavior because it provides neural support to three cognitive functions that are indispensable for conducting temporal sequences or gestalts of action: (1) short-term memory (working memory), (2) motor attention or set, and (3) inhibitory control of interference [11]. The prefrontal cortex supports these functions not alone but in close cooperation with other brain structures. Working memory is memory in the active state that the organism needs and uses for the performance of acts in short term. The behavior of working memory goes beyond simple stimulus-response reflexes and is typified by delayed-response tasks, which was introduced ninety years ago by W.S. Hunter [13]. In the various forms of delayed-response tasks, the instruction cue disappears before the trigger appears in a given trial and the sensory stimulus and motor response are separated by a brief delay period, during which time the information contained in the instruction needs to be kept in short-term memory until it can be used for the reaction. That is, the prefrontal cortex will sustain activation of neuronal populations during the active retention of sensory information, as during the delay of a delay task. During the process of reaching a goal, the old content of short-term memory can be given renewed timelines and term by the action at hand, which is realized by reinforcement in the form of either reward or punishment. When the goal is attained, the content of short-term memory has outlived its usefulness and must be discarded since its term has expired. A biological correlate for reward learning may be found in the activity of dopamine neurons, which respond rather uniformly to unexpected rewards which are particularly effective for learning. The selected action is maintained by the reinforcer. This kind of learning is thus called reinforcement learning by the machine learning community [66]. Being much more focused on goal-directed learning from interaction than are other approaches to machine learning, reinforcement learning is the basis of neurocomputational models for working memory. The second function of the prefrontal cortex is set, which is conventionally understood as the preparation for action or attention directed to prospective action. If the content of working memory is mainly sensory, the content of set is mainly motor. Being complementary to each other, they help mediate cross-temporal contingencies. The third function of prefrontal cortex, the inhibitory control of interference, is integrative and protects the structure of behavior, speech, or thought from interfering influences, external or internal, that may conflict with it and lead it astray.

APPENDIX B

EIGHTY GABOR WAVELET MASKS

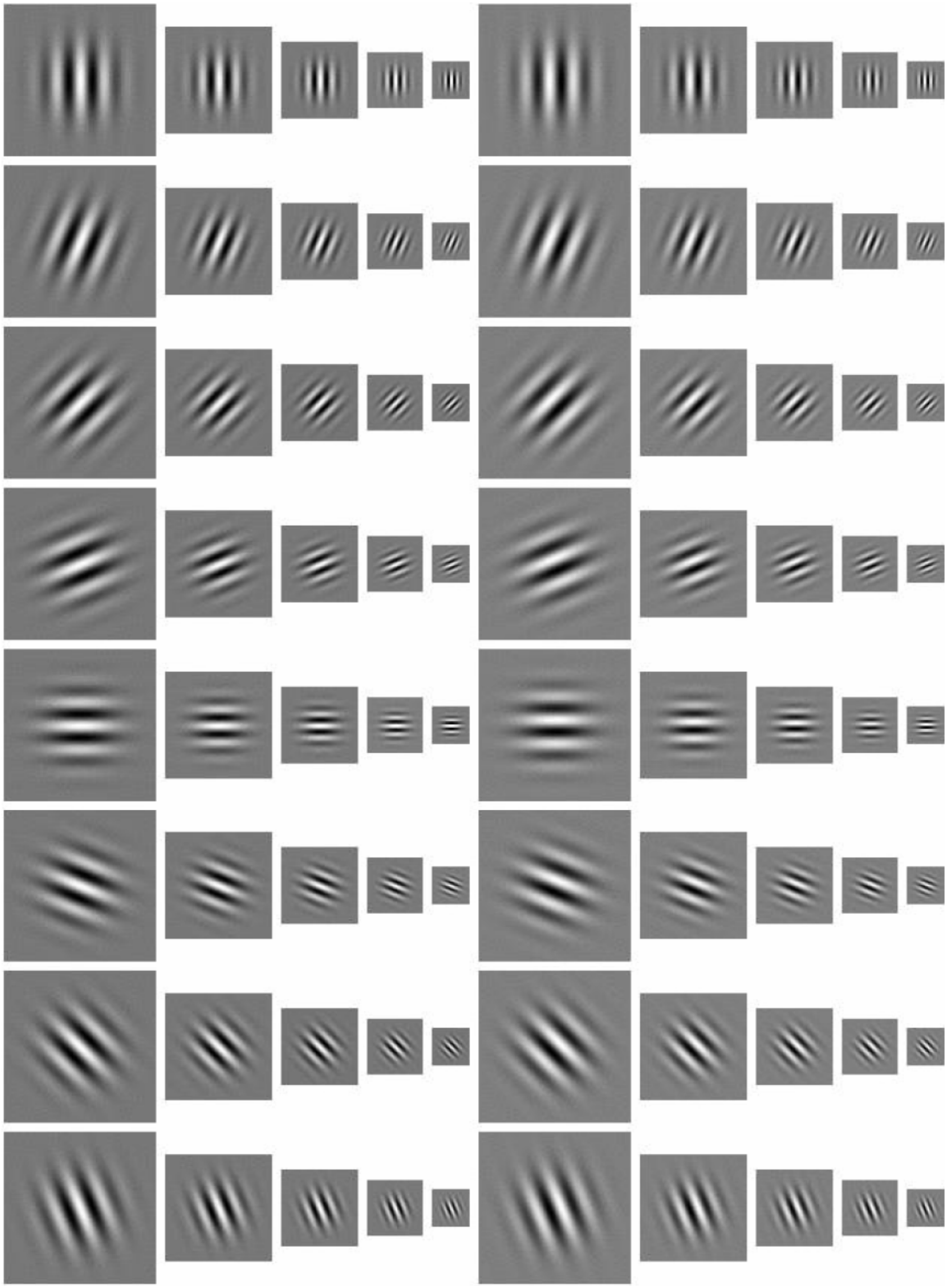
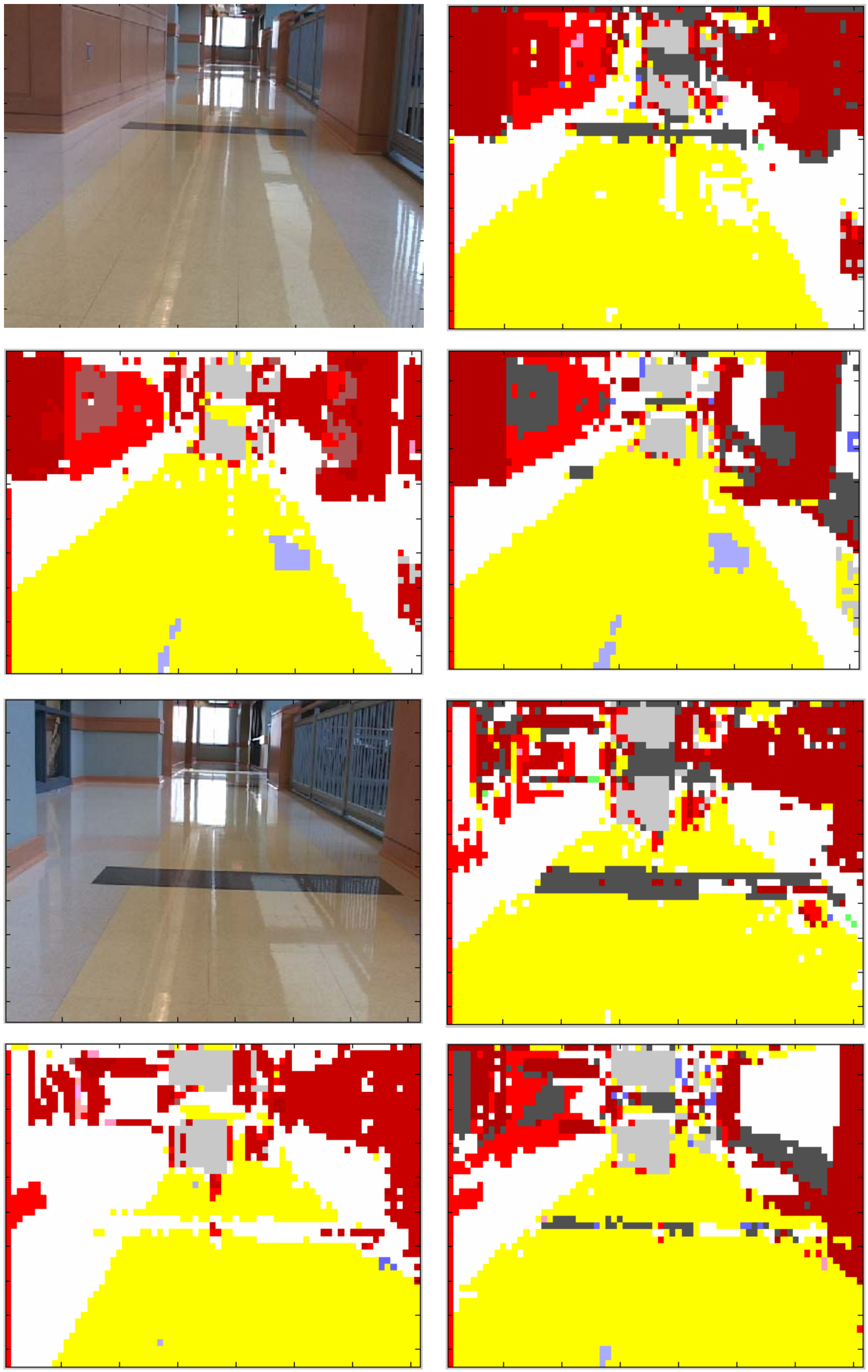
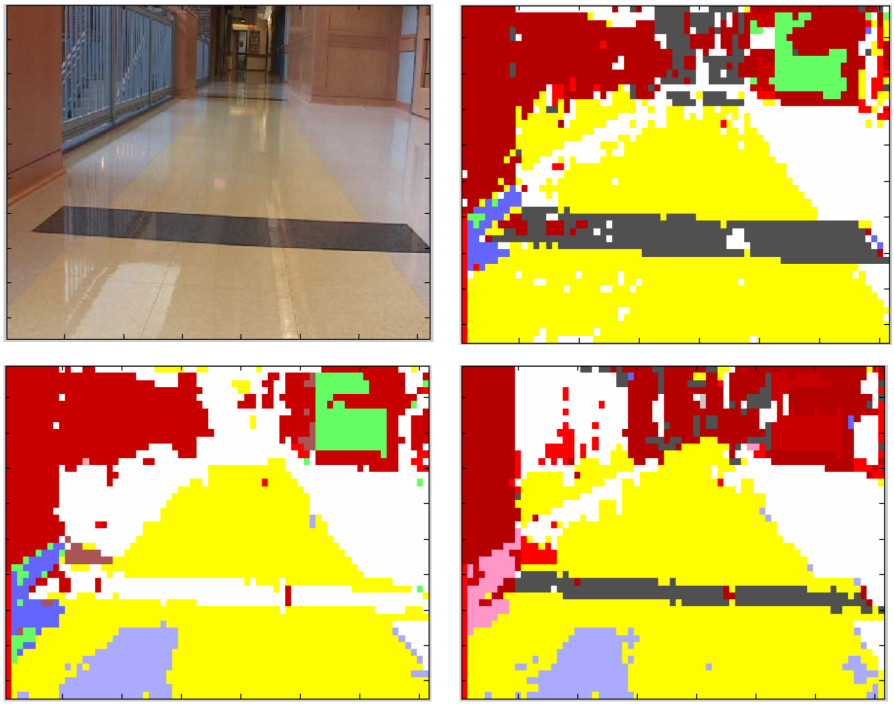


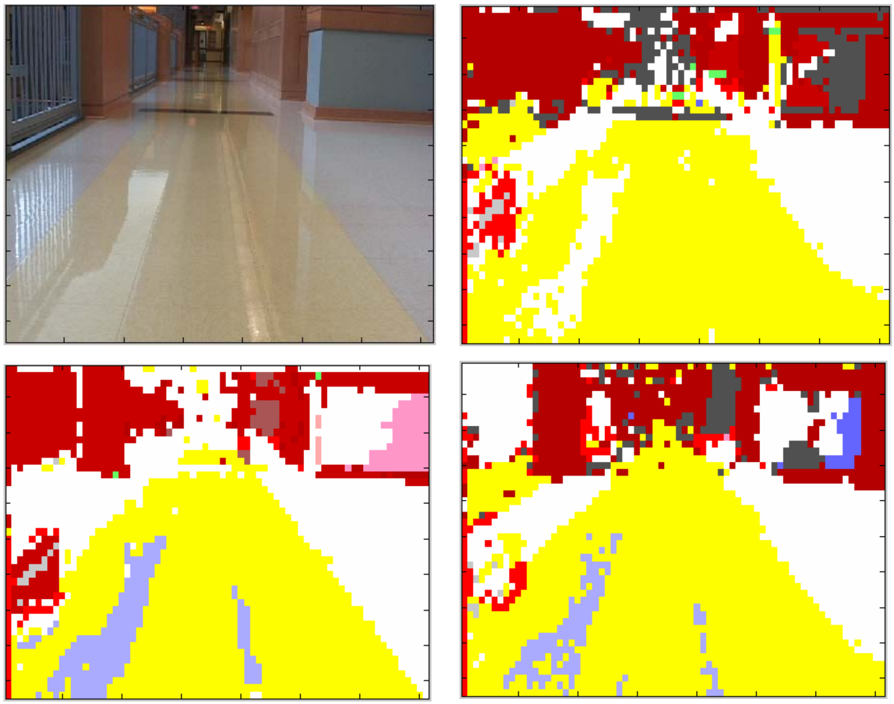
Fig A. 14 Eighty Gabor wavelet masks [122]

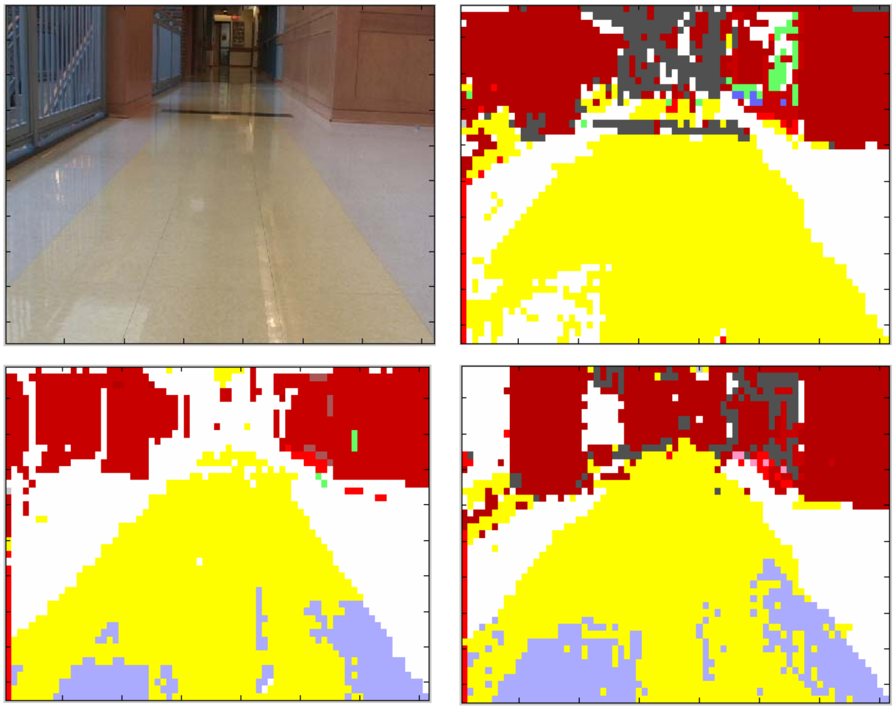
APPENDIX C

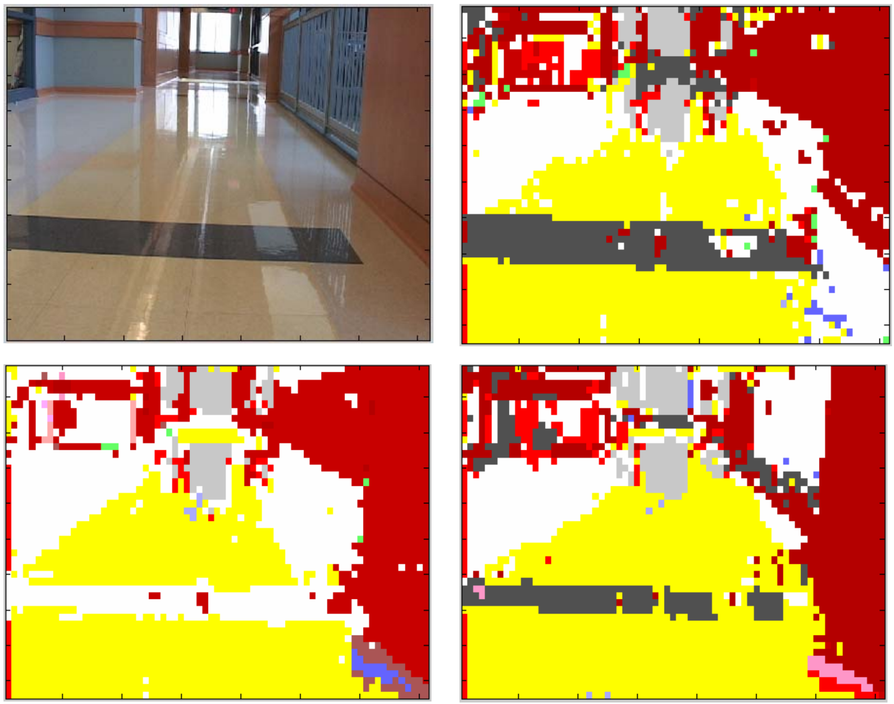
EXPERIMENT I RESULTS

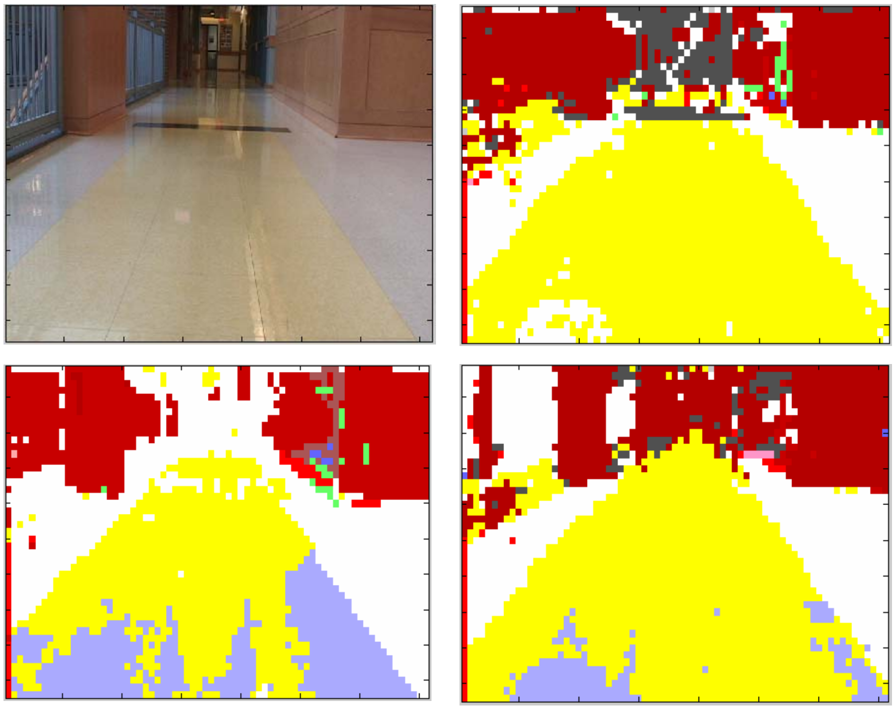


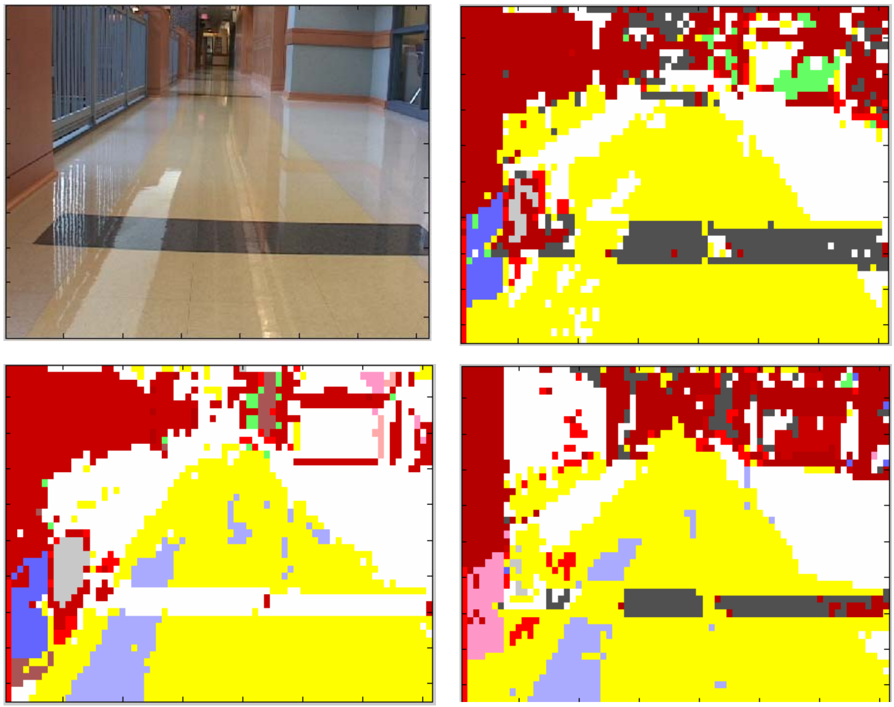












REFERENCES

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

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

[3] Goldstone, R. (1998). Perceptual learning. *Annu. Rev. Psyhchol*, 49:585-612.

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

[5] Hunter, J., Human Motion Segmentation and Object Recognition using Fuzzy Rules, *Proceedings of 14th Annual IEEE International Workshop on Robot and Human Interactive Communication (RO-MAN 2005)*, Nashville, TN, August 13-15, 2005,pp 210-216, 2005.

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

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

[8] Butler, A.B., and Hodos, W. (1996). *Comparative Vertebrate Neuroanatomy: Evolution and Adaptation*. John Wiley & Sons.

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

[10] Baldassarre, G. (2002) A modular neural-network model of the basal ganglia's role in learning and selecting motor behaviors. *Cognitive Systems Research* 1:5-13.

[11] Fuster, J.M. (1997). *The Prefrontal Cortex. Lippincott-Raven Publisher*, Philadelphia, New York.

[12] Schultz, W.S., Apicella, P., Romo, R. & Scarnati, E. (1995) Context-dependent activity in primate striatum reflecting past and future behavioral events. In Houk, C.J., Davis, L.J., and Beiser, G.D. (Eds.), *Models of information processing in the basal ganglia*. Cambridge, MA: The MIT Press.

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

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

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

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

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

[18] Young, M.P., and Scannell, J.W., 1993. Analysis and modeling of the mammalian cerebral cortex. In *Experimental and Theoretical Advances in Biological Pattern Formation*, ed. H.G. Othmer, RK. Main and J.D. Murray, pp. 369-384. New York: Plenum Press.

[19] Tovee, M.J., 1996. *An Introduction to the Visual System*, Cambridge University Press.

[20] Mel, B.W. (1997). SEEMORE: Combining color, shape, and texture histogramming in a neurally inspired approach to visual object recognition. *Neural Computation* 9, 777-804.

[21] Swain, M.J. & Ballard, D.H. (1991). Color indexing. *International Journal of Computer Vision*, 7:1, 11-32.

[22] Pereira, F.C.N., Tishby, N. & Lee, L. (1993). Distributional clustering of English words. In *Proceeding of the 31st Meeting of the Association for Computational Linguistics* pp. 183-190. Columbus, OH.

[23] Puzicha, J., Hofmann, T. & Buhmann, J.M. (1999). Histogram clustering for unsupervised segmentation and image retrieval. *Pattern Recognition Letters*, 20, 899-909.

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

[25] Cesmeli, E. & Wang, D. (2001). Texture segmentation using Gaussian-Markov random fields and neural oscillator networks. *IEEE Transaction on Neural Networks*, 12(2), 394-628.

[26] Geman, D., Geman, S., Graffigne, C. & Dong, P. (1990). Boundary detection by constrained optimization. *IEEE Trans. Pattern Analysis and Machine Intelligence* 12(7), 609-628.

[27] Wersing, H., Steil, J. & Ritter, H. (2001). A competitive layer model for feature binding and sensory segmentation. *Neural Computation*, 13, 357-387.

[28] Bow, S.T., 2002. *Pattern Recognition and Image Processing*. Marcel Dekker, Inc., New York.

[29] Kulkarni, A.D., 2001. *Computer Vision and Fuzzy-Neural Systems*. Prentice Hall.

[30] Zhang, T., Ramakrishnan, R., and Linvy, M., 1996. Birch: an efficient data clustering method for large databases. In *Proc. of 1996 ACM-SIGMOD Int. Conf. on Management of Data, Montreal, Quebec*.

[31] Guha, S., Rastogi, R., and Shim, K., 1998. CURE: an efficient clustering algorithm for large database. In *Proc. of 1998 ACM-SIGMOD Int. Conf. on Management of Data*.

[32] Guha, S., Rastogi, R., and Shim, K., 1999. ROCK: a robust clustering algorithm for categorical attributes. In *Proc. of the 15th Int'l Conf. on Data Eng*.

[33] Jain, A.K., and Dubes, R.C., 1988. *Algorithms for Clustering Data*. Prentice Hall.

[34] Kaufman, L., and Roussseeuw, P.J., 1990. *Finding Groups in Data: an Introduction to Clustering Analysis*. John Wiley & Sons.

[35] Ng, R., and Han, J., 1994. Efficient and effective clustering method for spatial data mining. In *Proc. of the 20th VLDB Conference*, Santiago, Chile.

[36] El-Sonbaty, Y., Ismail, M.A., and Farouk, M., 2004. An efficient density based clustering algorithm for large database. In *Proc. of the 16th IEEE Int'l Conference on Tools with Artificial Intelligence*.

[37] Ester, M., Kriegel, H.P., Sander, J., and Xu, X., 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proc. of the Second Int'l Conference on Knowledge Discovery and Data Mining*, Portland, OR.

[38] Ankerst, M., Breunig, M.M., Kriegel, H.P., and Sander, J., 1999. OPTICS: Ordering Points To Identify the Clustering Structure. In *Proc. of ACM SIGMOD'99 Int'l Conf. on Management of Data*, Philadelphia, PA.

[39] Wang, W., Yang, J., and Muntz, R., 1997. STING: A Statistical Information Grid Approach to Spatial Data Mining. In *Proc. of the 23rd VLDB Conference*, Athens, Greece.

[40] Sheikholeslami, G., Chatterjee, S., and Zhang, A., 1998. WaveCluster: A multi-resolution clustering approach for very large spatial database. In *Proc. of the 24th VLDB Conference*, New York, USA.

[41] Karypis, G., Han, E.H., and Kumar, V., 1999. Chameleon: Hierarchical clustering using dynamic modeling. *IEEE Computer*, 32(8):68--75.

[42] Hopfield, J.J., 1982. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences*, vol. 79, pp. 2554-2558.

[43] Markou, M. & Singh, S. (2003). Novelty detection: a review-part1: statistical approaches. *Signal Processing*, 83, 2481-2497.

[44] Markou, M. & Singh, S. (2003). Novelty detection: a review-part2: neural network based approaches. *Signal Processing*, 83, 2499-2521.

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

[46] Butler, A.B., and Hodos, W. (1996) *Comparative Vertebrate Neuroanatomy: Evolution and Adaptation*. John Wiley & Sons.

[47] Striedter, G. (2004) *Principles of Brain Evolution*. Sinauer Associates.

[48] Nehmzow, U. (2003). *Mobile robotics: a practical introduction*. London; New York:Springer.

[49] Atkinson, R.C. and Shiffrin, R.M., (1968). Human memory: a proposed system and its control process. *The Psychology of Learning and Motivation: Advances in Research and Theory*, (Spence, K.W., ed.) pp. 89-195, Academic Press.

[50] Baddeley, A.D. & Hitch, G.J. (1974). *Working memory, The Psychology of Learning and Motivation* (Bower, G.A., ed.) pp. 47-89, Academic Press.

[51] Baddeley, A.D. (2000). The episodic buffer: a new component of working memory. *Trends Cogn. Sci*. 4. 417-423.

[52] Desimore, R. (1998). Visual attention mediated by biased competition in extrastriate visual cortex. *Nature* 353: 1245-1255.

[53] Helmholtz, H. von, 1925. Helmholtz's Treatise on Physiological Optics (translated from German by J.P.C. Southall). *The Optical Society of America*, G. Banta, Menasha, Wisconsin.

[54] Hayek, F.A., 1952. *The Sensory Order*, University of Chicago Press, Chicago.

[55] Bruner, J.S., 1973. *Beyond the Information Given: Studies in the Psychology of Knowing*. W.W. Norton, New York.

[56] Fuster, J.M., 1997. *The Prefrontal Cortex*, Lippincott-Raven Publisher, Philadelphia, New York.

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

[58] Baldassarre, G. (2002) A modular neural-network model of the basal ganglia's role in learning and selecting motor behaviors. *Cognitive Systems Research* 3, 5-13.

[59] Tulving, E. (2002). Episodic memory: from mind to brain. *Annu. Rev. Psychol*. 53:1-25.

[60] Clayton, N.S. & Dickinson, A. (1998). Episodic-like memory during cache recovery by scrub jays. *Nature* 395: 272-274.

[61] Nadel, L. & Willner, J. (1980). Context and conditioning: A place for space. *Physiology & Behavior*, 8, 218-228.

[62] Nadel, L., Willner, J, & Kurz, E.M. (1985). Cognitive maps and environmental context. In P. Balsam & A. Tomie (Eds.), *Context and learning*. pp. 385-406. Hillsdale, NJ: Erlbaum.

[63] Sutherland, R.J. & Rudy, J.W. (1989). Configural association theory: The role of the hippocampal formation in learning, memory, and amnesia. *Psychology*, 17, 129-144.

[64] Rudy, J.W., & O'Reilly, R.C. (2001). Conjunctive representations, the hippocampus, and contextual fear conditioning. *Cognitive, Affective, & Behavioral Neuroscience*, 1(1), 66-82.

[65] Phillips J.L. and, Noelle, D.C. (2005). "A Biologically Inspired Working Memory Framework for Robots," in *Proc. of the 27th Annual Meeting of the Cognitive Science Society*, Stresa, Italy.

[66] Sutton, R.S. and Barto, A.G., 1990. Time-derivative models of Pavlovian conditioning. In M. Gabriel and J.W. Moore, eds., *Learning and Computational Neuroscience*, pp. 497-537. Cambridge, MA: MIT Press.

[67] Rescorla, R.A. and Wagner, A.R., 1972. 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: Appleton-Century-Crofts.

[68] Dayan, P. and Abbott, L.F., 2001. *Theoretical Neuroscience*, Cambridge, MA: MIT Press.

[69] Wang, R.F. & Spelke, E.S. (2000). Updating egocentric representations in human navigation. *Cognition*, 77, 215-250.

[70] Mittelstaedt, H. (1983). The role of multimodal convergence in homing by path integration. *Fortschr Zool*, 28, 197-212.

[71] Wehner, R. (1983). Celestial and terrestrial navigation: human strategies-insect strategies. *Neuroethology and Behavioral Physiology* (ed. Huber, F. & Markl, H.) Springer-Verlag, Berlin and Heidelberg, New York, pp 366-381.

[72] Tolman, E.C. (1948). Cognitive maps in rats and men. *Psychol Rev*, 55, 189-208.

[73] Wehner, R. (1992). *Arthropods. Animal Homing* (ed. Papi, F.). Chapman and Hall, London, pp 45-144.

[74] Baddeley, A., Aggleton, J.P. & Conway, M. (2002). *Episodic Memory: New Directions in Research*. Oxford University Press, Oxford.

[75] Collett, T.S., Fauria, K., Dale, K. & Baron, J. (1997). Places and patterns-a study of context learning in honeybees. *J Comp Physiol A*, 181, 343-353.

[76] O'Keefe, J. & Nadel, L. (1978). *The hippocampus as a cognitive map*. Clarendon Press, Oxford.

[77] Taube, J.S. (1998). Head direction cells and the neuropsychological basis for a sense of direction. *Prog Neurobiol*, 55, 225-256.

[78] Taube, J.S., Muller, R.U. & Ranck, J.B. (1990a). Head-direction cells recorded from the postsubiculum in freely moving rats: I. description and quantitative analysis. *J Neurosci*, 10, 420-435.

[79] Robertson, R.G., Rolls, E.T., Georges-Francois, P. & Panzeri, S. (1999). Head direction cells in the primate pre-subiculum. *Hippocampus*, 9, 206-219.

[80] Knierim, J.J, Kudrimoti, H.S. & McNaughton, B.L. (1995). Place cells, head direction cells and the learning of landmark stability. *J Neurosci*, 15, 1648-1659.

[81] Burgess, N., Jeffery, K.J. & O'Keefe, J. (1999). Integrating hippocampal and parietal functions: a spatial point of view. *The Hippocampal and parietal foundations of spatial cognition* (ed. Burgess, N, Jeffery, K.J. & O'Keefe, J.). Oxford University Press, Oxford, pp. 3-29.

[82] Burgess, N., Becker, S., King, J.A. & O'Keefe, J. (2001a). Memory for events and their spatial context: models and experiments. *Philosophi Trans R Soc Lond B Biol Sci*, 356, 1493-1503.

[83] Reed, J. & Squire, L.R. (1999). Impaired transverse patterning in human amnesia is a special case of impaired memory for two-choice discrimination tasks. *Behavioral Neuroscience*, 113(1), 3-9.

[84] Siegwart, R., 2004. *Introduction to autonomous mobile robots*. Cambridge, Mass. : MIT Press

[85] Kortenkamp, D. & Weymouth, T. (1994). Topological mapping for mobile robots using a combination of sonar and vision sensing. In *Proceedings of the Twelth National Conference on Artificial Intelligence (AAAI' 94)*, pp 979-984, Seattle, Washington.

[86] Thrun, S. (1998). Bayesian landmark learning for mobile robot localization. *Machine Learning*, 33(1). Pp 41-76.

[87] Vlassis, N., Motomura, Y. & Krose, B. (2000). Supervised linear feature extraction for mobile robot localization. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA'00)*, pp, 2979-2984.

[88] Fleischer, J. & Marsland, S. (2002). Learning to Autonomously Select Landmarks for Navigation and Communication. In *Proceedings of the Seventh International Conference on Simulation of Adaptive Behavior*. Pp, 151-160. MIT Press.

[89] Zimmer, U.R. (1996). Robust world-modeling and navigation in a real world. *Neurocomputing*, Special Issue, 13(2-4):247-260.

[90] Bourque, E. & Dudek, G. (2000). On-line construction of iconic maps. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA'00)*, pp, 2310-2315.

[91] Tani, J. & Nolfi, S. (1999). Bayesian landmark learning for mobile robot localization. *Machine Learning*, 33(1):41-76.

[92] Linaker, F. & Niklasson, L. (2000). Extraction and inversion of abstract sensory flow representations. In From Animals to Animates: *The 6th International Conference on Simulation of Adaptive Behaviour (SAB'00)*, pp, 199-208. MIT Press.

[93] Collett, T.S. & Zeil, J. (1998). *Places and landmarks: an arthropod perspective. Spatial representation in animals* (ed. Healy, S.D.). Oxford University Press. Oxford, pp, 18-53.

[94] Newton, I, 1704. *Opticks*. London: Smith and Walford.

[95] Zeki, S., 1983. Color coding in the cerebral cortex: The reaction of cells in monkey visual cortex to wavelengths and color. *Neuroscience*, vol. 9, pp. 741-756.

[96] Zeki, S. and Marini, L., 1998. Three cortical stages of color proceeding in the human brain, *Brain*, vol. 121, pp. 1669-1685.

[97] Hunter, J.E., 2005. *Object recognition using fuzzy membership function rules.* Vanderbilt University. Nashville, TN.

[98] Copyright © 2000, 2001, 2002. http://en.wikipedia.org/wiki/RGB_color_model, http://en.wikipedia.org/wiki/HSV_color_space. Free Software Foundation, Inc., Boston, MA.

[99] Foldiak, P., 2002. Sparse coding in the primate cortex. In M.A. Arbib (Ed.), *The handbook of brain theory and neural networks* (2nd ed., pp. 1064-1068) Cambridge, MA: MIT Press.

[100] Gabor, D., 1946. Theory of communications. *Journal of Institute of Electrical Engineering*, vol. 93, pp. 429-457.

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

[102] Garner, W.R., 1974. *The processing of information and structure*. Wiley, New York.

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

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

[105] Guttman, A., 1984. R-tree: A dynamic index structure for spatial searching. In *SIGMOD '84, Proceedings of the ACM SIGMOD Conference*. ACM Press.

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

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

[108] O'Reilly, R.C., Braver, T.S., and Cohen, J.D. (1999). A biologically based computational model of working memory. In (Miyake and Shah, 1999), chapter 11, pages 375-411.

[109] Pioneer 2 Mobile Robots Operations Manual, (2000).

[110] Mollon, J.D. and Bowmaker, J.K., (1992). The spatial arrangement of cones in the primate fovea. *Nature*, 360, 677-679.

[111] Dartnall, H.J.A., Bowmaker, J.K., and Mollon, J.D., (1983). Human visual pigments: microspectrophotometric results from the eyes of seven persons. *Proceedings of the Royal Society of London, Series B*, 220, 115-130.

[112] Wong-Riley, M.T.T., (1993). Cytochrome oxidase on human visual cortex. In *Functional Organization of the Human Visual Cortex*, ed. B. Gulyas, D. Ottoson and P.E. Roland, pp. 165-180. Oxford: Pergamon Press.

[113] De Valois, R.L., Albrecht, D.J., and Thorell, L., (1978). Cortical cells: bar detectors or spatial frequency filters? In *Frontiers in Visual Science*, ed. S.J. Cool and E.L. Simith, Berlin: Springer-Verlag.

[114] von der Heydt, R., Peterhans, E. and Durstler, M.R., (1992). Periodic-pattern-selective cells in monkey striate cortex. *Journal of Neuroscience*, vol. 12, pp. 1416-1434.

[115] Ts'o, D.Y. and Gilbert, C.D., (1988). The organization of chromatic and spatial interactions in the primate striate cortex. *Journal of Neuroscience*, vol. 8, pp. 1712-1727.

[116] Roe, A.W. and Ts'o, D., (1995). Visual topography in primate V2: multiple representation across functional stripes. *Journal of Neuroscience*, vol. 15, pp. 3689-3715.

[117] Zeki, S., Watson, J.P.G., and Frackowiak, R.S.J., (1993). Going beyond the information given: the relation of illusory visual motion to brain activity. *Proceedings of the Royal Society of London Series B*, vol. 252, pp. 215-222.

[118] Tootell, R.B.H. et al., (1995a). Visual motion aftereffect in human cortical area MT revealed by functional magnetic resonance imaging. *Nature*, vol. 375, pp. 139-141.

[119] Rolls, E.T. and Tovee, M.J., (1995). Sparseness of the neuronal representation of stimuli in the primate temporal visual cortex. *Journal of Neurophysiology*, vol. 73, pp. 713-726.

[120] Mishkin, M., Ungerleider, L.G. and Macko, K.A., (1983). Object vision and spatial vision: two cortical pathways. *Trends in Neurosciences*, vol. 6, pp. 414-417.

[121] Livingstone, M.S. and Hubel, D., (1988). Segregation of form, color, movement, and depth: anatomy, physiology and perception. *Science*, vol. 240, pp. 740-749.

[122] Bolme, D.S., (2003). Elastic Bunch Graph Matching. *Thesis of Master of Science*.