# Working Memory and Perception

D.M. Wilkes, M. Tugcu, J.E. Hunter, and D. Noelle

*Center for Intelligent Systems*
*Vanderbilt University*
*Nashville, Tennessee USA*
*{Mitch.Wilkes/ Mert.Tugcu/ Jonathan.E.Hunter/ David.Noelle}@Vanderbilt.edu*

***Abstract*** **– The ability to teach a robot new skills and tasks without explicit programming is an important goal in robotics. Such capability tends to imply the ability to learn from experience, much like many biological creatures. Evidence suggests that working memory plays a pivotal role in this learning process, in part by focusing attention on the most relevant data. We describe ongoing research to study the utility of computational neuroscience models of working memory within robotic systems. A system comprised of working memory, short term memory, long term memory, spatial reasoning and perception modules is proposed. The paper focuses on the perceptual module and its interaction with the working memory. Results are given to show the current progress.**

***Index Terms – Working memory, robot vision, computer vision, learning, perception..***

## I. INTRODUCTION

A major goal in robotics has been the development of robots having the ability to learn and reason about new tasks, and thus acquire new skills without explicit programming. The utility of this idea is self-evident, in that it takes the programming of robots away from the unnatural and abstract process of writing large amounts of tedious code, and transforms it into a natural process of learning through experience. Such experience-based learning is used widely by human beings, primates, mammals, and many other animals. Consequently, it is reasonable to expect that we can learn more from biology about this process of learning from past experiences. For example, there is evidence that there are adaptive working memory structures in primate brains [1,2,3]. This type of memory is important to the learning and performing of tasks insofar as it facilitates decision making by focusing on the information necessary for the task and ignoring distracting irrelevant data [4].

### *A. Studying Working Memory*

One of the long-term goals in the Center for Intelligent Systems at Vanderbilt University is to develop an adaptive working memory to support the following*:

- Focus attention on the most relevant features of the current task.
- Support generalization of tasks without explicitly programming the robot.
- Guide perceptual processes by limiting the perceptual search space.
- Provide a focused short-term memory to prevent the robot from being confused by occlusions, i.e., to avoid the "out of sight, out of mind" problem.
- Provide robust operation in the presence of distracting irrelevant events.

A difficulty in studying an adaptive working memory is that it doesn't function in isolation. It must be part of a more complex system, thus we are integrating an adaptive working memory structure into a robot in order to provide the embodiment necessary for exploring the issues of task learning [5]. This will result in a realistic system involving perceptual systems, actuators, reasoning, and short- and long-term memory structures.

The architecture proposed for this system is shown in Figure 1. As can be seen, its structure is that of a fully connected arrangement of five modules. The center of this system is the adaptive working memory. It holds only a limited amount of data, the data most relevant to performing the current task. For this reason it communicates directly with all the other modules, receiving relevant data from the perceptual system, passing data to and from the short-term and long-term memory modules, as well as acting as the primary source of information for the spatial reasoning module. The limited storage capacity of the working memory has the effect of focusing the attention of the robot. The limited capacity also necessitates having some means of deciding which information should enter the working memory as well as which should be removed.

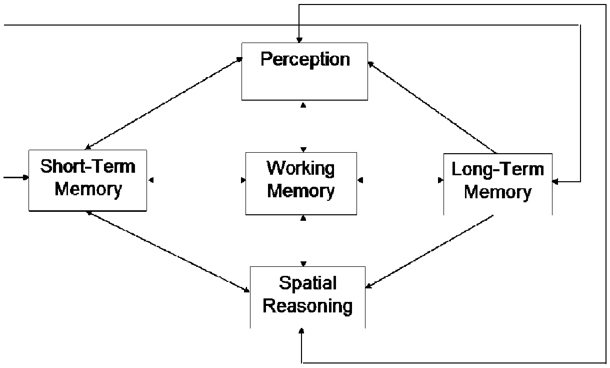

Fig. 1 Proposed system architecture for studying working memory

The short-term memory (STM) contains data acquired within a recent time frame. This time frame is not necessarily fixed but typically is at the scale of minutes. The STM is further decomposed into two parts, a spatial STM called the

---

This work has been supported by the National Science Foundation under Award ID 0325641.

Sensory EgoSphere (SES) [6] and another memory for non-spatial information. The long-term memory (LTM) also is composed of two parts, a spatial LTM called the Landmark EgoSphere (LES) [7] and another for non-spatial data, such as information regarding how to perform the tasks known by the robot. The LES stores data about the permanent objects in the robot's workspace.

The spatial reasoning system is informed by all the other modules and provides the ability to reason on the spatial relationships among the objects the robot knows about. In particular, it provides reasoning about the objects currently within the robot's focus of attention, i.e., the contents of the working memory.

The perception system provides the robot with its observation of the world. It identifies objects and events within the sensory range of the robot using trainable classifiers described in this paper. The perception system supplies data to the spatial reasoning, working memory and STM, and in turn its perception processes may be guided by these three modules and by the LTM. This paper focuses on the perception system (in particular we focus on vision) and its relationship to the adaptive working memory.

### B. *Working Memory*

The primary challenge faced by the working memory system in this architecture is determining what information should be actively maintained in working memory and what information can be safely discarded in the current context. Restricting the capacity of the working memory is intended to provide focus to the various problem solving efforts of the robot, guiding the search for appropriate actions. Thus, it is important that the working memory system select for retention exactly those "chunks" of information that are critical for task success. One approach to the design of this selection mechanism would be to write task-specific procedures that embody knowledge concerning the task-relevance of candidate working memory chunks. We have sought a more general and adaptive approach, however. The working memory system of this architecture is equipped with a mechanism for learning, from experience, which chunks to actively maintain in memory. This learning mechanism is grounded in computational neuroscience models of the working memory circuits of the prefrontal cortex [4].

Computational neuroscientists have discovered that the response properties of certain neurons in the midbrain – cells which communicate using the neurotransmitter dopamine – encode for *changes in expected future reward*, suggesting that these cells may be involved in a kind of reinforcement learning based on reward prediction, such as temporal difference (TD) learning [8]. Neural instantiations of such reinforcement learning algorithms have explained how action sequences might be learned from the occasional delivery of a reward signal [9]. The fact that midbrain dopamine neurons project widely to the prefrontal cortex has suggested that such a reinforcement learning scheme might guide the updating of working memory contents, as well as the learning of motor sequences [10]. Based on the success of computational neuroscience models of working memory updating based on temporal difference learning [11], the working memory system in this robot control architecture utilizes TD learning to determine which informational chunks to retain.

The reinforcement learning mechanism learns to associate features of the current sensory state of the robot and features of a considered informational chunk with an estimate of future reward that might be had if that chunk is actively maintained. This learning process can be greatly facilitated by computing abstract features of the current state and the current chunk that might be indicative of the utility of the chunk. The inclusion of such abstract features increases the likelihood that the reinforcement learning algorithm can focus on a relatively small set of features in order to decide on chunk retention, and this can speed learning. Thus, one of the central jobs of the perceptual system in this architecture is the generation of abstract features of the current sensory state and of sensory chunks that might be indicative of the utility of the currently considered chunk. Noting that an object at a given location is a "tree" might be more informative when trying to decide if that location should be remembered than noting that the object is "brown", for example. In this way, the vocabulary, or semantic meaning, of features provided by the perceptual system can play a central role in assisting the working memory to learn the value of various chunks.

The perceptual system can also provide other forms of guidance to the reinforcement learning processes of the working memory. For example, the detection of novelty in considered chunks may prompt the working memory system to "explore" – retaining the novel chunk just in order to see what happens. Reinforcement learning algorithms must carefully balance "exploration" versus "exploitation" of previously learned knowledge in order to ensure the discovery of better choices, when they exist [12]. By detecting novelty, the perceptual system can inform the quest for this balance.

### C. *Perception System*

The autonomous robot vision problem is exceedingly difficult and has not been solved in general. Therefore, robot vision systems tend to be designed for specific types of problems and circumstances. The resulting systems are typically useful only for these limited environments. Additionally, the results are often fragile, and once trained, it may be cumbersome to train the system to recognize new objects.

The system we are developing for studying working memory is expected to place a relatively wide range of demands on the perception system. Additionally, we wanted the perception system to have utility beyond the working memory study, and thus be useful for other robot systems. Thus, we desired to develop a perception system with broad capabilities and the ability to learn to perform new visual tasks. Some of the guiding principles in the design of this vision system were:

- Use a large set of visual features that have semantic meaning, *i.e.*, can be explained naturally with language.

- The system should be able to explain its reasoning in terms of these semantically meaningful features.
- The system should be taught simply by choosing and labeling examples from acquired images.
- The system should use categorization and classification methods to determine which features have strong identification capability.
- The system has the ability to learn *attentional* features for narrowing the focus of a visual search.
- The system should be able to detect novelty, *i.e.*, to detect that it is viewing something that it has rarely, if ever, seen before.

Some other flexible vision systems that have been reported in the literature include [13,14,15].

The remainder of this paper is organized as follows. Section II describes the basic system structure and Section III presents some of the main details of the perception system. Section IV describes some of our results and Section V provides a discussion and some conclusions.

## II. BASIC SYSTEM STRUCTURE

The requirement that the system have broad capabilities for visual tasks lead to the selection of a large number of basic visual features computed from regions in the image. Our initial set of visual features extracted from a region includes a histogram of 250 colors sampled from the HSV color space based on a human perceptual model of hue (a model developed in our laboratory) and a simple measure of texture, or "roughness," based on the Laplacian of the intensity in the region. We do not believe this to be a complete set of features for this or other visual systems and we expect to add more features as the research progresses. This is merely an initial set for getting started.

We also set down some larger long-term goals that we wanted the system to meet. These goals include the following:

- Learn how to perform segmentation and attention processing from examples and continually refine these capabilities as new data is acquired.
- Eventually become a system able to gather its own data for additional training, learning and refinement, *i.e*., be able to performed unsupervised self teaching.
- Maintain a strong connection to the original features and their semantic meaning.
- Provide a confidence measure or probability of correctness with the segmentation decisions.
- Be able to detect novelty such as new classes or events, and use this to trigger learning, *i.e*., perform "student-driven" learning where the robot asks the user, or "teacher," for information.
- Be able to run the entire learning process in this student-driven mode, where novelty is used to prioritize the student's request for input.
- Keep track of ongoing performance statistics to enable the system to make internal judgments on which classes it can discriminate and which tend to be confused with each other.

A flowchart conception of how the final system will interact with a human user (*i.e.,* a teacher) to learn, process and present its results, is given in Figure 2. The actions of the human teacher are on the left side of the diagram and those of the robot student are on the right.

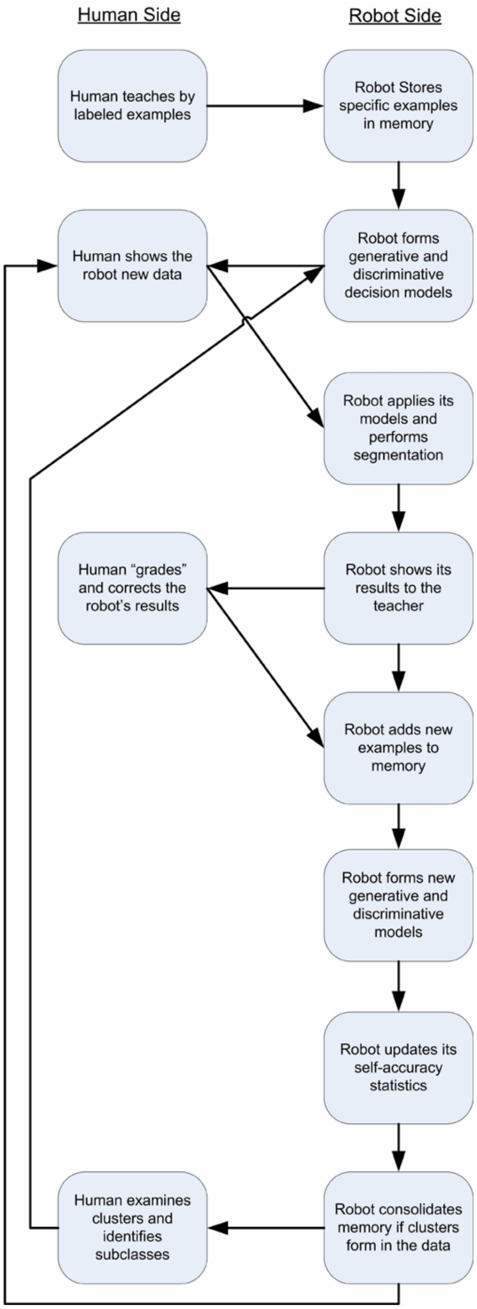

Fig. 2 Diagram of the desired human-robot interaction

## III. PERCEPTION SYSTEM

The structure of the system is presented in Figure 3, and will now be described in more detail.

### *A. Initialize the Memory*

The perception system's exemplar memory is called *Imemory*, for image memory. This database contains all of the specific retained experiences of different objects. The Imemory is set up as an array of structures containing the reference picture (i.e., the picture that contains a certain exemplar), the coordinates for the rectangular area that contains the region from which the exemplar features are computed, the class label for the exemplar, and the computed exemplar feature vector.

Imemory is initialized by the user selecting small rectangular regions from training images. A vector of 251 features is then computed from each region. The vectors are labeled by the user according to the class to which they belong. For example, in an outdoor situation some of the likely classes are "Grass," "Tree Trunk," and "Sidewalk."

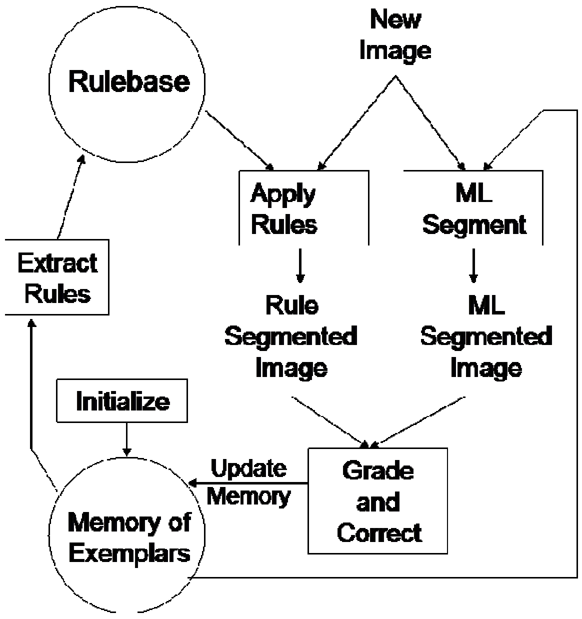

Fig. 3 Perception system architecture

### *B. Extract Rules / Rulebase*

We use the well known machine learning approach of Top Down Induction of Decision Trees (TDIDT) to to give partitions of the training data. The `treefit` function of Matlab is used with its default parameters to fit a decision tree to the training data. The resulting tree is saved, and then we remove the most discriminating feature from the training data (i.e., the feature at the top node of the tree) and compute a new tree. This process is repeated until a predetermined number of trees are formed. Currently we tend to use between 15 and 50 trees. Consequently, we obtain a number of partitions of the training data, from which we desire to extract rules for deciding on the class of the observed feature vector. We use multiple trees in this manner so that a large number of decision rules may be obtained.

In their original form decision trees form partitions via binary decision rules obtained by comparing feature values to thresholds. That is, if a feature vector $\underline{\mathbf{x}}$ has the following form,

$$\underline{x} = [x_1\ x_2\ \cdots x_n]^T \tag{1}$$

then each decision node is of the form $x_i > t_i$ where $t_i$ is a threshold determined by the `treefit` function. Each leaf node of a tree corresponds to a decision rule that selects for one of the class labels in the training data. Additionally, it is the logical AND of the node decision rules traversed from the top node to the leaf node. We use the following notation to denote these individual binary decisions as

$$x_i \quad \longrightarrow \quad i^{\text{th}} \text{ feature} > t_i \tag{2}$$

$$\bar{x}_i \quad \longrightarrow \quad i^{\text{th}} \text{ feature} < t_i \tag{3}$$

As a result, a typical leaf node rule may resemble $x_1\bar{x}_3x_{15}x_{75}$ where the number of terms is determined by the number of decision nodes traversed, and the subscripts are the features used in the associated nodes. The binary decisions at the nodes suffer from being fragile when applied to new data, thus it is desirable to consider a way to avoid a crisp threshold and use a smooth function to indicate a qualitative strength the feature. The optimal form of such functions is a topic of future research, but as a simple first approach we simply use linear functions of the features. Thus we use a different interpretation of the leaf node rules. Each individual term is replaced by the appropriate function below,

$$x_i \quad \longrightarrow \quad k_i x_i \tag{4}$$

$$\bar{x}_i \quad \longrightarrow \quad 1 - \bar{k}_i x_i \tag{5}$$

where $x_i$ denotes the value of the $i^{\text{th}}$ feature, $k_i$ is chosen to yield a result of 1 at the maximum value of the feature, $\bar{k}_i$ is chosen to yield a result of 1 at the origin and 0 at the maximum value of the feature, and the AND operation is replaced by multiplication. The leaf node expression that results from this interpretation is a multidimensional polynomial function of the feature values. The sum of all the polynomial functions pertaining to a particular class, say the $k^{\text{th}}$ class, over all the trees formed, results in a score function $f_k(\underline{x})$ for the $k^{\text{th}}$ class. The larger the value of the function, the more we may attribute it as evidence for the presence of that class. These functions are not optimal in any known sense and have only a qualitative justification at this time. Their utility lies in the fact that they operate on the original semantically meaningful features and attempt to discover the most discriminating features. Additionally, the terms in the polynomial expressions can be interpreted into language statements about how the score, or decision, was made. This is a topic of current research.

### *C. ML Segmentation*

The robot perception system performs a maximum likelihood (ML) segmentation of the image using nonparametric estimates of the *a priori* conditional probability density functions $p_X(\underline{x} \mid C_k)$ where $C_k$ is the $k^{\text{th}}$ class. These density functions are estimated from the system's memory of exemplars, Imemory.

The standard method of population modeling consists of acquiring samples of a class (usually acquired manually), estimating the pdf models (very often the model is a single Gaussian or perhaps a Gaussian mixture model), and using the model to analyze future samples. One advantage of using this method is that a model estimated from a relatively few samples generalizes allowing the identification of many other samples in various proximities of the original samples. After the functional model is created, the calculations are computationally efficient.

A disadvantage of the generalization in such a model is that it may be inaccurate. If the assumed structure of the pdf is incorrect, the model can "lie" to the user creating generalizations among features that should not exist for accurate perception analysis. The estimation of the most accurate models, e.g., the Gaussian mixture models, are usually computationally expensive. This tends to inhibit the robot's learning in practice because the pdf model may not be refined after the acquisition of new samples simply as a matter of computational practicality. Additionally, this method has difficulties when used in high dimensional spaces. In our case the feature vectors of each object sample contain 251 different elements, a very high dimensional space. In practice, researchers may tend to gather a few hundred exemplars from which to estimate the pdf model. For the model to be accurate there must be a sufficient number of samples to create one or more clusters in the feature space. In lower dimensional spaces, a few hundred samples may be sufficient. In higher dimensional spaces, the amount of points needed before clusters form may be much higher.

The pdf estimation approach is a straightforward nonparametric method of using the collected sample exemplars in the Imemory to analyze unknown samples. This method allows for simple updates to the pdf model without the need for recalculation. All that is needed is to add the new samples to Imemory. The value of the pdf at a new sample is estimated by dividing the number of exemplars in that class within a small radius, *r*, of the new sample by the total number of exemplars of that class, and then dividing this ratio by the volume of the hypersphere of radius, *r*. By controlling this radius the amount of generalization obtained by the model can be controlled. The class is estimated by the maximum likelihood principle, i.e., by which of the *a priori* pdf's yields the highest value.

This approach may eventually enable the robot to analyze a picture itself and create many new training samples of objects with little or no interface with user. This method can be computationally expensive as the number of samples grows. This leads to a need for memory consolidation, which is a topic of future research.

### *D. Rule Segmentation*

The score functions, developed from the rules described in Section *III.B.* above, result in a separate score image for each class in the training data. A current topic of research is methods for combining these score images in order to obtain a segmentation of the original image.

### *E. Grade and Correct*

The segmented images, obtained from new testing data, are presented to the human user for evaluation and correction, much as a teacher might grade and correct a student's work. This evaluation results in new labeled data that can be added to the Imemory exemplar memory. Due to the large number of segmented regions that can be evaluated from a single test image, this approach greatly aids in the rapid collection of large numbers of exemplars.

## IV. RESULTS

We have trained the proposed perception system on an outdoor environment in front of Featheringill Hall at Vanderbilt University. The exemplar memory was initialized with approximately 100 exemplars and then tested on training data. Through interaction with the human user via the method of ML segmentation followed by "grade and correct," the Imemory rapidly grew to a total of 1810 exemplars. The camera used is a Sony video camera equipped with a parabolic mirror for a 360 degree field of view. The original unfolded 360 degree camera view and the robot's ML segmentation are shown in Figure 4. The system was specifically trained on the four classes Grass, Sidewalk, Tree Trunk, and Artificial Landmark. The artificial landmark is a cardboard cylinder having three bright saturated strips of color: pink, orange and green. From the interactive training process the system learned to perform good segmentations of the grass and the sidewalk. The performance on the trees is relatively good, but there are some definite misclassifications. The artificial landmark is not in this scene and there are a few false detections of this class.

The ML segmentation approach provides conditional probability density functions that can also be used to compute self information. The computation simply involves taking the negative of the logarithm of the estimated pdf value. This information measure can be used for novelty detection, since events with a low probability of occurrence have high information content. Figure 5 shows an empty table workspace. This image is broken into small regions which in turn are processed to provide a large set of exemplar feature vectors. These are all labeled as Uninteresting Background vectors. Next, three new objects are added to the workspace. The system has not seen these objects before (see Figure 5). Using the exemplars from the empty scene, the self information of the regions in the image in Figure 5 is computed. The resulting information image is shown in Figure 6 and the novel objects are clearly segmented. This novelty detection provides very useful information for the working memory to use in determining whether data should be kept or discarded, and whether it should engage in exploration of novel data.

## V. CONCLUSIONS AND FUTURE WORK

We have presented a larger system for studying models of working memory inspired by neuroscience research. This larger system contained 5 modules: working memory, short term memory, long term memory, spatial reasoning, and perception. Some of the important aspects of the working memory were also presented. The paper then focused on the perception system, its interaction with the working memory, and the requirements for flexibility and learning imposed on the perception. This research is ongoing and not all objectives have been met at this time. However, the results presented show good performance in learning segmentation and novelty detection tasks.

## REFERENCES

[1] Fuster, J. M. and Alexander, G. E. (1971). Neuron activity related to short-term memory. Science, 173:652–654.

[2] Funahashi, S., Bruce, C. J., and Golman-Rakic, P. S. (1989). Mnemonic coding of visual space in the monkey's dorsolateral prefrontal cortex. Journal of Neurophysiology, 61:331–349.

[3] Miller, E. K. and Desimone, R. (1994). Parallel neuronal mechanisms for short-term memory. Science, 263:520–522.

[4] 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.

[5] Brooks, R.A., "A robust layered control system for a mobile robot", *IEEE Journal of Robotics and Automation*, vol. 2, no. 1, pp. 14-23, 1986.

[6] Peters, R.A., Hambuchen, K.E., Kawamura, K., and Wilkes, D.M., "The sensory egosphere as a short-term memory for humanoids," *Proc. IEEE-RAS Int. Conf. Humanoid Robots,* Tokyo, pp. 451-459, November, 2001.

[7] Kawamura, K., Koku, A.B., Wilkes, D.M., Peters, R.A., and Sekmen, A.,"Toward egocentric navigation," *Int. J. of Robotics and Automation,* vol. 17, no. 4, pp. 135-145, 2002.

[8] Sutton, R. S. (1988). Learning to predict by the method of temporal differences. Machine Learning, 3:9–44.

[9] Montague, P. R., Dayan, P., and Sejnowski, T. J. (1996). A framework for mesencephalic dopamine systems based on predictive hebbian learning. Journal of Neuroscience, 16:1936–1947.

[10] Braver, T. S. and Cohen, J. D. (2000). On the control of control: The role of dopamine in regulating prefrontal function and working memory. In Monsell, S. and Driver, J., editors, Control of Cognitive Processes, volume XVIII of Attention and Performance, chapter 31, pages 713–737. MIT Press.

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

[12] Sutton, R.S. and Barto, A.G., *Reinforcement learning: an introduction.* MIT Press, Cambridge, Massachusetts, 1998.

[13] Oliver, N.M., Rosario, B., and Pentland, A.P., "A Bayesian computer vision system for modeling human interactions," *IEEE Trans. PAMI*, vol. 22, no. 8, pp. 831-843.

[14] Hongeng, S., Bremond, F., and Nevatia, R., "Bayesian framework for video surveillance application," *Proc. ICPR 2004*, pp. 164-170, August, 2004.

[15] Piater, J.H. and Grupen, R.A., "Feature learning for recognition with Bayesian networks," *Proc. ICPR 2004,* pp. 17-20, August, 2004.

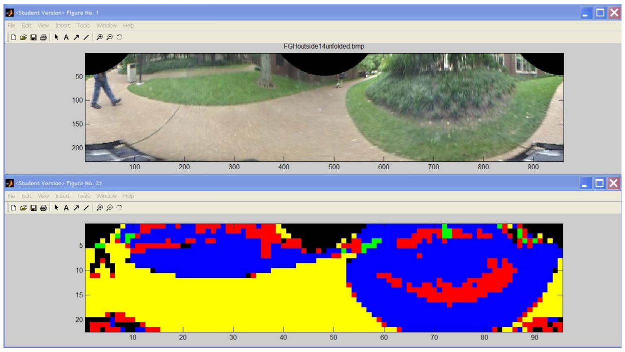

Fig. 4 Original 360 degree camera image and ML segmentation. Blue represents grass, yellow represents sidewalk, and red represents tree segmentations. The few green pixels represent false segmentation of an artificial landmark not present in the scene.

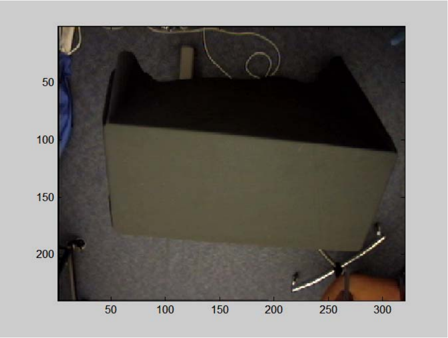

Fig. 5 Empty workspace scene

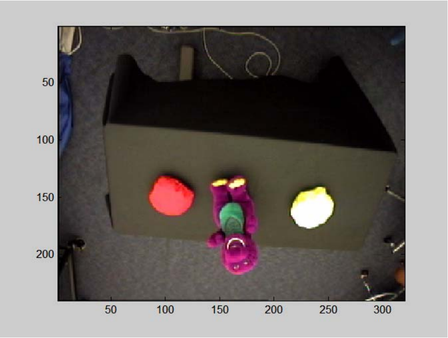

Fig. 4 Workspace scene with novel items

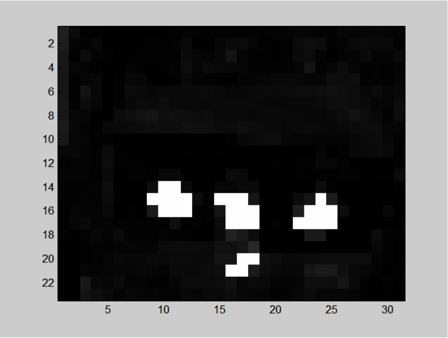

Fig. 6 Detection of novel items