A biologically inspired memory in a multi-agent based robotic architecture

Tomas Arredondo*, Patricio Castillo*, Pablo Benapres*, Javiera Quiroz* and Miguel Torres†
*Electronic Engineering Department
Universidad Tecnica Federico Santa Maria, Valparaiso, Chile
tomas.arredondo@usm.cl
†Department of Electrical Engineering
Pontificia Universidad Catolica, Santiago, Chile
mtorrest@ing.puc.cl

Abstract—We describe a biologically inspired memory in a multi-agent based robotic architecture. In this approach, memory and pattern recognition are intertwined to form a cognitive memory that is used for recognition of objects in a robotics environment.

This memory is implemented in a multiple agent behavior based blackboard architecture as an object recognition agent. The agent performance is tested against a standard dataset with satisfactory results.

The system is currently installed in a mobile robotic platform where its capabilities and applications are explored.

I. INTRODUCTION

Memory is a fundamental component of all mental processes, specially used to store, retain and recall information and/or experiences. Considering the remarkable capability of higher organisms for pattern recognition, this leads naturally to an understanding of memory that is based on behavioral and performance models, rather than to formulate an approach toward understanding memory that is based on brain anatomy and physiological models [1]. Given this perspective, memory and pattern recognition are intertwined.

The cognitive memory system performs pattern recognition, but differs significantly from common pattern recognition systems. Most pattern recognition systems learn to classify patterns by finding some parameters that generate the expected classification output for a given set of training samples in the so-called training stage. Once the training stage is complete, the training patterns are discarded and new input patterns are classified according to the fixed set of parameters found [16], [17].

Cognitive memory, on the other hand, stores previously presented training patterns and when prompted by a new input pattern produces an output that is an aggregate recollection of its contents. If no patterns in the memory are associated to the input pattern, then a new label or tag is created and the pattern is stored as a new class. Thus the cognitive memory approach to recognition stores much more information than typical classifiers. As storage technologies become faster, memory limitations are no longer a strong restriction for devices of this kind.

Cognitive memory can have many kinds of input patterns (e.g. visual, auditory, etc.). As part of a general philosophy toward biomorphic robotic systems, and given the richness of information available in images, motivates the use of visual cues in the present work.

Robotic vision and specifically human-centered vision systems have shifted from specific vision techniques toward fundamental principles on how cognitive abilities, like vision, emerge and how specific knowledge can be acquired from the interaction with the environment [2], [7], [9], [15].

Autonomous robotic systems must concurrently handle sensors, actuators and data processing tasks such as planning, mapping and environment perception using pattern recognition. This multitude of requirements calls for a multi-agent architecture [3].

In this context, cognitive memory is currently proposed to be included into an agent in a multiple agent architecture driven by a set of user-defined motivations. The cognitive memory is implemented in a novel multiple-agent architecture and shown to yield excellent recognition levels of a variety of objects, with low false positives and misdetections.

This paper is organized as follows, section II presents an introductory background on cognitive memory. A review of biologically inspired computer vision schemes is presented in section III, followed by a description of the cognitive memory agent-based architecture in section IV. The experimental setup and the results are discussed in sections V and VI, respectively. Final remarks are given in section VII.

II. COGNITIVE MEMORY

In a computer memory, data is stored in numbered registers or addressed locations. Subsequently, the CPU retrieves this data by instructing the memory to deliver the contents of specific registers. It is very unlikely that the human brain has either addressed memory locations or numbered registers. Though not fully understood yet, human memory is probably “content addressable”, where data is stored permanently as patterns (visual, auditory, tactile, etc.) in storage locations as required. Data is retrieved in response to “prompt” patterns that may be inputs (external stimuli) from sensors such as eyes, ears, etc., or patterns that may be generated internally (e.g., via thought processes).
There are two main objectives in pursuing research on cognitive memory as pointed by Widrow [1]:

- To develop “human-like” memory for computers that would complement existing forms of computer memory in order to facilitate solutions to problems in AI, pattern recognition, speech recognition, control systems, etc.
- To advance cognitive science with new insights into the workings of the human mind.

Fig. 1 [1] shows architectural elements and structures of a preliminary mechanistic memory system that could behave to some extent like human memory.

Memory patterns may not be stored in the brain’s neurons because it takes too long to train and they must be continually trained because synaptic memory is likely to vary over time and some neurons die continually. After training a neural network, the training patterns are generally discarded, but human memory doesn’t discard the training patterns as biological long-term memory is capable of recording patterns of interest in great detail over an entire lifetime. The hypotheses concerning the operation of human memory underlying the cognitive model in fig. 1 can be found in [1].

Most adaptive pattern recognition systems learn to classify patterns by adapting to a set of training patterns. Once trained, the training patterns are discarded. When new input patterns are applied to the trained classifier, these patterns are classified in accordance with the training experience [17]. Cognitive memory, on the other hand, stores training patterns (visual, auditory, etc.) in folders and recalls the entire contents of the folder when prompted by a new input pattern. The folder that produces the best match has one or more associated tags that give cognitive meaning to it. Depending on the cognitive policy being used, the new image may be stored in a folder and tagged to give meaning to the new input pattern by induction or association with existing tags (or possibly new ones).

### III. Biologically Inspired Vision Systems

Robot vision refers to the capability of a robot to visually perceive the environment and use this information to execute various tasks [4]. Robot vision extends methods of computer vision to fulfill the tasks given to robots and robotic systems. Typical tasks are to navigate toward a given target location while avoiding obstacles, to find a person and react to the person’s commands, or to detect, recognize, grasp and deliver objects.

Even if biological vision systems are not perfect in their performance, most common tasks that seem trivial may be impossible for artificial systems to perform. For example, biological vision systems have the ability to attend to parts of a scene and rapidly remove irrelevant regions so that it can concentrate on what it is important for the task at hand.

**A. Previous Approaches**

Recent neuroscientific findings show that such tasks such as object grasping and manipulation are realized through distributed information flows between multiple regions within the nervous system [5], [6], [7].

A review of biological vision system was published by Marr [2]. He introduced a theory based on a series of abstraction levels that guide the processing of the image from a 2D sketch to a structured and object-centered 3D model. The view put forward by Marr is that vision is seen as a grouping and reconstruction process of 3D shape models. Another model of biologically centered vision, is the idea that there are different streams of processing tailored to specific tasks. An example, is the processing in what and where streams [8] or recognition of objects from parts of the objects [9].

Bajcsy argued that the problem of perception was not of image processing nature but of control of data acquisition [10]. The active vision paradigm has been pushed forward by the works of Aloimonos et al. [11] and Ballard [12], who proposed approaches to combine the different visual observations with apriori information in such a way that the process achieved a common task.

Following the paradigm of active vision, two lines of research emerged: work on visual attention and work that more closely integrated a robot’s action with visual feedback. A closer integration of the robot mechanism with the vision processing is manifested in a series of works on Visual Servoing, where the robot actively follows the object motion for tracking, navigation, or grasping. Continuous vision-based feedback control of the position of the robot is referred to as Visual Servoing (VS) [13], [14].

There is also a trend in the integration of findings from biological vision into computer vision approaches such as object recognition [15].

In summary we can say that there are three mayor usages of vision systems for robotics:

- Detection
- Tracking
- Recognition

1) **Detection:** When detecting an object, we can find points of interest that can provide information to describe what the robot is looking at. These can be landmarks, edges, surface textures, or any feature that allows the determination of an object location and/or pose. This is the basic task for most of the more advanced applications of vision systems.

2) **Tracking:** Toward tracking objects, common techniques for this application are model-based, maximum-velocity, and interest-point tracking. The first method is about the determination of the 3D pose in which an object might be and the detection of structures to reduce the combinatorial complexity. The second technique is used to estimate the velocity of a target in the image and is very sensitive to latency in the image refresh. The third method has different approaches from model based to appearance based and its goal is to enable recognition of objects in different poses and conditions (including occlusion). These tasks can be used to measure an object movement or to find and follow a target.

3) **Recognition:** Recognition in one of the most widespread usages of vision systems, as images can provide information as rich and detailed as the system can process. For recognition, common techniques are model-based, appearance-based and
finding structures in data. In model based recognition, the algorithm tries to find a match between a known model of the object and the features in the camera. The second technique is usually trained with segmented images (in different poses if necessary) and then tries to find those textures with PCA, SVM or similar methods. The third one commonly uses a mixture of the previous methods to obtain a representation of the object in as many dimensions as possible.

**B. Cognitive Vision**

There has been a significant progress in the fields of artificial intelligence, computer perception, machine learning and robotics. Yet there has not been major progress on truly cognitive systems. Cognitive vision is a novel area of research in vision systems linked to biological vision. Cognition is here interpreted as generation of knowledge based on perception, reasoning and learning. One important difference between most object recognition vision systems and a truly cognitive vision system is the ability not only to recognize objects, but also to perform recognition by means of categorization and being able to "learn" new objects.

As seen in fig. 3, images are captured using a stereo camera system and different objects are extracted using an OpenCV based Blobber. Then both left and right eye blobs are sent to the cognitive agent for recognition. The cognitive agent implements the system seen in fig. 1 which takes the left and right eye images and transmits them through the multiplexer directly to the neural network for recognition. In all segments, the following steps occur:

- The neural network tries to replicate the input in its output.
- A classifier using Mean Squared Error (MSE) matches the input image with the neural network output and determines a hit or no hit.
- The Buffer Memory within the memory segment is used to store the input image, this image is compared with the image(s) in each of the memory folders in each segment and the folder in all segments with the lowest MSE is selected as a match.
- A tag in this folder is used to give categorization to the image, currently this tag is simply a unique integer much like a key in a relational database.

**IV. AGENT ARCHITECTURES**

Different methodologies have been proposed in agent architectures, these include four general classes: deductive approaches, deliberative toward means-ends approaches, deliberative motivation based, and behavior based [3], [18], [19].

**A. Deductive Architectures**

In the view of agents as theorem provers, an executable specification through a set of deduction rules is applied when certain conditions are satisfied. Hence, deductive architectures require a symbolic representation of the environment. The symbols are manipulated using logical operations in a manner that corresponds to logical deduction.

There are several difficulties with these type of methods [3], [18], [19], the main ones being:

- Transduction - the difficulty in translating the real world into an accurate and valid symbolic representation.
- Representation/reasoning - the issue of getting agents to manipulate the symbolic information correctly and in time for these results to be useful.
- Dynamic and noisy environments - if the information which the agent uses is inaccurate or has changed then reasoning may err seriously (e.g. a robot in a world with arbitrarily moving objects like a crowded hallway), it is potentially dangerous to rely on old information that may no longer be valid.
- Highly hierarchical - involving additional time lags due to information flows up and down the decision hierarchy.
in order for the agent to deliberate on a new course of action whenever environment conditions change. Because of the time lags during this processing, hierarchical control systems are best suited for structured and highly predictable applications (e.g. a manufacturing cell).

- Difficulty in engineering a complete system - as incremental competency proves difficult to achieve, the entire system needs to be built before system testing can be done. This is contrary to current engineering methods (e.g. agile, iterative or incremental engineering processes).

B. Deliberative Towards Means-Ends Architectures

Reasoning that is directed towards actions is one of the main characteristics of deliberative toward means-ends reasoning [3]. The Belief Desire Intention (BDI) paradigm pursues this type of reasoning and contains explicitly represented data structures that loosely correspond to these three mental states.

The Procedural Reasoning System (PRS) is the earliest and most prevalent implementation of the BDI agent architecture [3], [22], [23], [24]. PRS consists of: (i) a database (i.e. beliefs) containing facts about the environment and internal states, (ii) desires which represent the agent’s goals, (iii) an intention structure containing those plans that the agent has committed to achieve, and (iv) a set of plans describing how sequences of actions that have their preconditions satisfied are to be executed to fulfill certain desires.

Plans in PRS always involve: (i) a goal - the postcondition of the plan, (ii) a context - the precondition of the plan, and (iii) a body - the recipe part of the plan, i.e. the set of actions to execute. In PRS, an interpreter runs the entire system by selecting the appropriate plans based on current beliefs and desires, putting those that are selected on the intention structure so that they may be executed in order.

C. Deliberative Motivation Based Architectures

In deliberative motivation based schemes, associated with each motivation is a numeric measure of motivational value. Motivations may also assist in modeling the context of an agent. What is contextual depends upon an agent’s sensors, actuators, size, internal states and so on. The context of an autonomous planning agent is captured partly by representing capabilities. What is contextual depends upon an agent’s sensors, actuators, size, internal states and so on. The context of an autonomous planning agent is captured partly by representing capabilities. What is contextual depends upon an agent’s sensors, actuators, size, internal states and so on. The context of an autonomous planning agent is captured partly by representing capabilities.

D. Behavior Based Architectures

Behavior based systems were first introduced in the realm of robotics (e.g. subsumption architecture), but these concepts [18], [19] have extended into general agent based systems [3]. Behavior based (i.e. reactive) agent architectures in general do not use world models, representative or symbolic knowledge and there is a tight coupling between sensing and action. This design philosophy promotes the idea that agent based systems should be inexpensive, robust to sensor and other noise, uncalibrated and without complex computers and communication systems.

Two key concepts of behavior based robotic systems (e.g. subsumption) that can be extended into agents are situatedness and embodiment. Situatedness refers to the ability of an agent to sense and readily react to its current surrounding without the use of abstract representations (i.e. maps, symbolic representations) and embodiment means that the agent(s) (e.g. robot, control system, web search engine, etc.) should experience the world directly. Behavior based learning systems may typically include softcomputing based methods such as reinforcement learning, neural networks, genetic algorithms, fuzzy systems, case and memory based learning and others [3], [18], [19].

Planning actions based on complex internal world representations in not seen as something beneficial because of its inherent error and associated temporal costs. In general, a purely reactive agent \( (Ag_r) \) will produce a response out of all possible responses \( (R) \) based on current stimulus \( (S) \) which can be represented by the following map:

\[
Ag_r : \beta(S) \rightarrow R
\]  

Each individual stimulus or percept \( s_i \) (where \( s_i \in S \)) is a tuple consisting of a particular type or perceptual class \( (p) \) and a strength value \( \lambda \):

\[
s_i = (p, \lambda)
\]

The agent \( (Ag_r) \) chooses a current response \( (r_i) \) out of the set of possible responses available \( (R) \) whenever \( \lambda \) is greater than a threshold \( \tau \):

\[
R = \{r_1, r_2, \ldots \}.
\]

1) Behavior Layering and Coordination: Behavior based systems tend to be developed in an incremental manner in which multiple behavior functions are layered on top of one another in an iterative process and without changing previous layers. In behavior based systems layering is done in a different dimension than what traditionally had been pursued in the AI community, the conventional layering model being vertical (fig. 2.a) and the behavior based model being horizontal (fig. 2.b).

Layered behavior based systems require a mediation mechanism (i.e. a coordinator) that produces an appropriate overall output for the agent at any point in time given current stimuli (fig. 2.c). There are competitive and cooperative types of coordination mechanisms for behavior based systems. In competitive methods when conflicts arise as a result of two or more behaviors being active, a coordinator typically will select a winner take all response in which the winning response is chosen to be executed by the robot. This can take the form of strict hierarchical methods through the use of suppression of lower layers (e.g. subsumption) [19], action-selection methods which arbitrarily select the output of a
single layer based on activation levels determined by the agent’s goals and current stimulus, and voting based methods in which behaviors generate votes and the action with the most votes is chosen. Cooperative methods require that behaviors be fused or somehow added, this in general requires an output representation that is amenable to such fusion [18].

V. IMPLEMENTATION IN A ROBOTICS CONTEXT

The cognitive memory system was implemented in a medium-size mobile robot (1m tall) with stereo cameras, two Single Board Computers (SBCs), sensors and motors as described in Table I. The system has a behavior-based multiple agent architecture as shown in fig. 3.

A. Blackboard Architecture

The system relies on a blackboard architecture [26] for message passing as implemented in [20], [25]. The idea of a blackboard architecture is to solve a given problem by using expert agents to solve jointly subproblems together. Each agent has some area of expertise which it is responsible for. The results of each expert are presented on a shared data repository called blackboard. Each expert can read the results of other experts using the blackboard. The most important is that solving all subproblems must lead into solving the given problem.

In our current implementation, the system consists of eleven agents. As the interface between the experts a Common Data Area (CDA) serves as the blackboard previously described. The system can be categorized as hybrid because it includes both behavior-based and non-behavior-based agents. Synchronization of memory access is arranged through semaphores to avoid undesired interactions. These agents run on two separate SBCs with a common memory (i.e. CDA) as shown in fig. 3.

The eleven agents and their tasks are summarized in Table II.

B. Cognitive Memory for Object Recognition

As previously mentioned, the cognitive agent takes the image information stored by the Blobber in the CDA shared memory. Then scales the images of each blob and uses them as inputs to the segments of the cognitive memory. The system checks if any of these objects is recognized. Depending on the mode of operation defined for the experiment, this agent can add new objects to the cognitive memory.

The cognitive memory is composed of different segments, each segment contains a neural network trained to recognize a specific set of objects presented to the robot, the array of registers with the image of the blob, and the tag (label) that identifies this set (i.e. class) of objects. There is a limit on the maximum number of registers that can store a segment, as increasing the number of objects which the neural network is trained, also increases the probability of recognizing an image
TABLE II
AGENTS AND THEIR MAIN TASKS

<table>
<thead>
<tr>
<th>Agent</th>
<th>Main Task</th>
</tr>
</thead>
<tbody>
<tr>
<td>Init-CDA</td>
<td>Initializing and managing the Common Data Area.</td>
</tr>
<tr>
<td>Network</td>
<td>Handling communications with the control PC for retrieving position information.</td>
</tr>
<tr>
<td>Executive</td>
<td>Interfacing between hardware and software and the calculations of actual robot positions out of Odometry-Data and Virtual-GPS-Data.</td>
</tr>
<tr>
<td>Mapper</td>
<td>Mapping based on Fuzzy ART algorithm [27] and also representing objects in 3D using the Open Dynamics Engine.</td>
</tr>
<tr>
<td>Close Range Nav. (CRN)</td>
<td>Navigating around nearby obstacles in a purely reactive behavior based manner using fuzzy logic to convert sensor signals into actuation data [28].</td>
</tr>
<tr>
<td>Long Range Nav. (LRN)</td>
<td>Planning routes from one point to another depending on a set of motivations and a set of fuzzy fitness scores using a Genetic Algorithm (full details are outside the scope of this current paper).</td>
</tr>
<tr>
<td>Feature Navigator (FN)</td>
<td>Recognizing visual features in the environment obtained with the camera and updating the feature-data-base. This process was idle in the experimental evaluation of this paper.</td>
</tr>
<tr>
<td>Monitor</td>
<td>Selecting between LRN and CRN navigation modes. As previously described this corresponds to a competitive coordination mechanism.</td>
</tr>
<tr>
<td>Blobber</td>
<td>Processing the images to extract “blobs”, thus focusing attention on elements that outstand from the background, and storing the new information in the shared memory.</td>
</tr>
<tr>
<td>Cognitive Agent</td>
<td>Performing the cognitive memory process to recognize the objects captured by the blobber.</td>
</tr>
<tr>
<td>Memory Sync</td>
<td>Synchronizing the shared memory (CDA) employed as a repository for the cognitive memory process of each single-board computer (SBC) over an ethernet link.</td>
</tr>
</tbody>
</table>

incorrectly. Each item is distinguished from another by means of these tags. If two elements are similar enough then they are labeled with the same tag, otherwise a new tag will be created for the new element. Some aspects pertaining to recognition thresholds, the neural network implementation and the modes of operation of the cognitive system that we have developed are discussed next.

1) Recognition Thresholds: There are three thresholds involved in the recognition process, named as $\alpha$, $\beta$ and $\gamma$. These three thresholds determine the values of the Mean Square Error (MSE) for which an object is classified as well-known, known, familiar or unknown as shown by the intervals in fig. 4. Basically, if the MSE between a new test object and a previously stored object is below $\beta$, it is considered as recognized, otherwise is rejected or stored in a new register as a new object according to an operation mode described below.

The values of these thresholds was initially estimated as follows:

$\gamma$: An untrained neural network was used to measure the minimum MSE of a test image, which was typically close to an error of $1.5 \cdot 10^{-2}$ and rarely lower than this value, with the same neural network architecture. Thus $\gamma$ is empirically chosen to be $1.5 \cdot 10^{-2}$.

$\beta$: After training the network with images of some objects, the average MSE was measured with new images of the same class of training objects, which typically gave an MSE of less than $9 \cdot 10^{-3}$. This value is sensitive to the variation between trained and tested images, so $\beta$ is chosen as the one that yields a desired false positive rate.

$\alpha$: The average MSE for images very similar to those used for training (and therefore that do not provide new information). An adequate value for this threshold was found experimentally to be $7 \cdot 10^{-3}$.

The estimation of appropriate thresholds was performed with 400 different images representative of a broad range of conditions. Thresholds can be optimized to achieve improved recognition of reduced classes of objects instead of broad and general sets of objects. The computation of new thresholds can be done using the previously mentioned values as starting point by means of the analysis presented in the next section.

2) Squashing Parameter: As seen in equation 4, the neural network in the cognitive memory uses neurons with a sigmoid or logistic curve as an activation function. In order to adjust the behavior of the network, a “squashing” or smoothing parameter $s$ was determined empirically. As this values increases, the training time also goes up, therefore $s$ is increased systematically until the recognition rate is maximized.

$$g(x) = \frac{1}{1 + \exp^{-x}}$$ (4)

3) Operational Modes: Once the cognitive memory has been trained, it is necessary to select one of the following cognitive memory modes:

- **Idle**: Only performs recognition and does not store any images. It only recognizes images with MSE below $\alpha$.
- **Lazy**: Performs recognition and stores new images of recognized objects, i.e. it considers images with MSE below $\beta$.
- **Curious**: Performs recognition, stores new images of known objects and adds new objects that look familiar. It takes into consideration images with MSE below $\gamma$.
- **Greedy**: Performs recognition, stores new images of known objects and adds any new objects.
VI. EXPERIMENTS

The performance of the cognitive memory was tested using a subset of images from the Amsterdam Library of Object Images (ALOI) [29]. This is a color image collection of one-thousand small objects, systematically recorded with a varying viewing angle, illumination angle, and illumination color for each object, and additionally captured wide-baseline stereo images.

The experiments were conducted with different orientations of the objects and the color information was reduced as the images were converted to gray scale. From the entire database, we selected the following training, known and unknown tests:

- **Training set**: 10 objects with 5 orientations each were used. Thus a total of 50 images were used to train the agent.
- **Known set**: The set of known objects are the same as those of the training set but in 5 different orientations from those in the training set. A total of 50 images were used that should be recognized.
- **Unknown set**: This set included 25 untrained objects in 2 different orientations. A total of 50 images that should not be recognized.

A. Training

The training process was carried out in two stages:

**First stage**: The cognitive memory was trained with one image of each object (10 images) in Greedy mode. This is so that every image was added as a new object. Then backpropagation training was run for 100 cycles using the 10 images previously added.

**Second stage**: The remaining 40 images were presented to the cognitive memory in Lazy mode, which adds to the memory only the images of already known objects. A high recognition threshold \( \beta = 0.05 \) was used so that these would be recognized and added as additional information to the already known objects. The memory was trained until the Mean Square Error (MSE) of the entire training set exceeds by a third the most demanding recognition threshold \( \beta \). The value of \( \alpha \) was established empirically as \( 7 \cdot 10^{-3} \) in previous experiments.

B. Experimental Results

Two sets of results are presented, Receiver Operating Characteristic Curves and a confusion matrix.

1) **Receiver Operating Characteristic Curves**: In our results, three squashing \( s \) values are presented \( (s = \{40, 100, 200\}) \) with values of \( \beta \) for the entire range of recognition between \( 0.1 \cdot 10^{-3} \) to \( 20 \cdot 10^{-3} \). These results are presented as Receiver Operating Characteristic (ROC) curves for these three squashing values and are compared to the Color Co-occurrence Histogram method by Chang-Krum as in [30], which also uses the ALOI image set. Fig. 5, shows the cognitive memory (CM) curves and the color co-occurrence histogram (CCH) curves for different parameter values.

As seen in fig. 5, as the recognition threshold \( \beta \) increases, more images are recognized but this also increases the number of false positives. CM results are in general an improvement over CCH. Also, the highest recognition rate was achieved with \( s = 100 \) and \( \beta = 11.4 \cdot 10^{-3} \).

2) **Confusion Matrix**: The confusion matrix for \( \beta = 11.4 \cdot 10^{-3} \) and \( s = 100 \) can be seen in table III. This was the experiment with the best results, with a True Positive Rate \( TPR = 0.96 \), False Positive Rate \( FPR = 0.06 \) and accuracy of 95%.

3) **Recognition Error Analysis**: False positives resulted from images that have no major resemblance with the recognized object but as the most important recognition features are the shape and texture of the image rather than color, false positives would occur mainly with images with similar textures or shapes in some parts of the object. The direction of the illumination is also a major factor that highlights different characteristics depending on the direction of the light.

False negative resulted from objects that were more difficult to recognize than others, i.e. required a higher recognition threshold \( \beta \). Object (a) in fig. 6 usually required a lower recognition threshold than the rest of the trained objects. In the other hand, object (b) usually required higher \( \beta \) in order to be recognized and produced false negatives.

False positives resulted from objects that were similar to others and that lacked distinctive features. Object (c) in fig. 6 was trained and is similar (for the cognitive memory) to untrained object (d).

![Fig. 5. Receiver Operating Characteristic (ROC) comparison.](image-url)

<table>
<thead>
<tr>
<th>TABLE III</th>
<th>COGNITIVE MEMORY PERFORMANCE STATISTICS.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actual</td>
<td>Prediction</td>
</tr>
<tr>
<td>Known</td>
<td>48</td>
</tr>
<tr>
<td>Unknown</td>
<td>3</td>
</tr>
</tbody>
</table>

As shown in fig. 5, and as the recognition threshold \( \beta \) increases, more images are recognized but this also increases the number of false positives. CM results are in general an improvement over CCH. Also, the highest recognition rate was achieved with \( s = 100 \) and \( \beta = 11.4 \cdot 10^{-3} \).
This paper presented a biologically inspired memory in a multi-agent blackboard based robotic architecture. This approach was applied for recognition of objects in a robotics environment. The object recognition agent performance was tested against a standard dataset with very satisfactory results. The approach was implemented on a compact mobile robot and worked successfully with a tight amount of computational resources.

Results discussed in the previous section show that the cognitive memory approach to recognition yields high correct classification rates (96%) with low (6%) false positives. It is to be noted the approach is scalable in a natural way, i.e. new objects can be added to the system as the bank of memory segments is dynamically increased. This scalability is an advantage over other classifiers that have less modular structures, and whose parameters need to be retrained completely to achieve good interclass discrimination capability. This project is open-source and source code can be found at: http://sourceforge.net/projects/marrbot/.

Ongoing research is concerned with the development of a Bayesian approach to the computation of the cognition thresholds that maximize the likelihood of correct object recognition.

ACKNOWLEDGMENTS

This research has been supported by the UTFSM under DGIP Grant 231138, and by the National Commission for Science and Technology Research of Chile (Conicyt) under Fondecyt Grant 1110343. We would like to thank the UTFSM Electronic Engineering Department Workshop for their help with robot construction.

REFERENCES


Fig. 6. Examples of common false negatives and false positives.