ACKNOWLEDGMENTS

The authors wish to specifically thank Ken Gant, Patricia Crossno, David Gardener, Randall Smith, Rob Bugos, Johnny Biffle, Carl Digert and the many others at Sandia that have worked on the system and models, and have generally helped and supported this effort.

BIOGRAPHY

CREVE MAPLES **{{photo 1}}** obtained his B.S. in Chemistry from MIT and his Ph.D. in Nuclear Science from the University of California, Berkeley, in 1970. He is the author of over 100 articles in the field of computing and is recipient of awards in parallel computing, real-time systems, and computer architecture and scientific visualization.

He was previously Chief Scientist and director of software development at VITESSE Electronics Corp.; president of CARD Associates (Computer Application Research and Development); and head of the Advanced Computer Architecture Laboratory at the University of California's Lawrence Berkeley Laboratory.

Currently he is a Senior Scientist at Sandia National Laboratories in Albuquerque, where he heads the Synthetic Environment Laboratory. He is the principal investigator of the MERLIN (Memory Routed, Logical Interconnection Network) Project, an ultra high-speed, minimum latency, heterogeneous communication system, and the developer of

A VIRTUAL ENVIRONMENT INTERFACE TO COMPLEX AUTONOMOUS PERCEPTUAL SYSTEMS

Thomas P. Caudell
Interdisciplinary Computational Systems Group
Dept. of Electrical & Computer Engineering
University of New Mexico
Albuquerque, N.M. 87131
email: tpc@eece.unm.edu

ABSTRACT

This paper describes the on-going development of a novel interface approach to understanding complex systems. We present a description of an interface, referred to as a Homunculus, which allows an experimenter to explore complex systems through immersive virtual reality technology. We describe an initial application under development where the Encephalon, a biologically motivated neural architecture, is used to control a robotics system. Encephalon modules are represented in the Homunculus as 3-D icons. Information flow between modules of the neural network is represented as graphical animations. Virtual tools will be available to view, manipulate, model, diagnose, analyze, and navigate through the software and multi-dimensional data. We discuss many important research questions revealed by this work.

1. Introduction

This paper describes work in progress in the development of an immersive virtual reality interface to a robot controlled by a simulation of a complex artificial neural network. We label this interface a "Homunculus". The term *Homunculus* derives from the Latin diminutive form of the word "Homo", literally meaning "little man" or "dwarf". In the 16th and 17th centuries, medical scientists believed an exceedingly small body, called the homunculus, was contained in certain germinal cells, from which the human body formed [1]. Later on, in discussions of mammalian vision, the concept of a homunculus was invoked to disprove the existence of a "neural screen" in the visual cortex [2]. If the vision system actually represented the visual scene as a 2-D image on a screen-like structure in the brain, then perception would require another "eye" within the brain to comprehend the neural image. This "inner eye" was attributed to a homunculus. Of course, this line of reasoning leads to infinite regress, since the homunculus would require yet another smaller homunculus within its head to view its "neural screen" and so on - *ad infinitum*.

In modern times the term is used in neural anatomy to refer to the mappings of the external senses and motor activity onto the cerebral cortex of the brain [3, 4]. For example, Fig. 1 shows the mapping of the body's afferent sensory signals onto specific locations of the cortex, and the mapping of efferent motor signals onto the body's muscles. Note the distortion of portions of the body onto these regions, indicating variable allocation of cortical resources to important regions of the body's senses and musculature. The connection with the work described in this paper is established through the concept of representing a person within the brain, in our case, the robot's "brain".

In this paper we describe our work in developing the Homunculus virtual reality interface to a robotics system controlled by a complex software artificial neural network. The goals for our work in this environment are to 1) extract and refine *engineering principles* for the design of more advanced autonomous perceptual systems based on studies of artificial and biological systems, 2) design and simulate autonomous perceptual systems based on these principles and guidance from theoretical formalisms of neural computation [5], 3) conduct extensive, controlled laboratory tests of the simulations using robotics systems that feature sensory and environmental interactions, and 4) explore human-computer interface issues associated with this application domain. We postulate that the Homunculus environment will expedite research in the field of autonomous perceptual systems, while simultaneously exposing many new research issues in the field of human-computer interfaces.

With virtual reality (VR) technology, the scientist can immerse his/her senses into a virtual environment that contains a representation of complex dynamical system [6]. When the VR system is interfaced to an autonomous robot, we can metaphorically think of this environment as an interface directly to the robot's "brain" (Fig. 2). The interface will allow modification of the software code and configuration, the setting or adjustment of parameters, the monitoring of information flow, the visualization of intermediate computational results, the viewing of raw input and output data streams, and the performance of statistical analysis on sets of system variables. Within the virtual environment, multi-sensorial modalities can be used to represent and understand information. For example, spatially localized 3-D

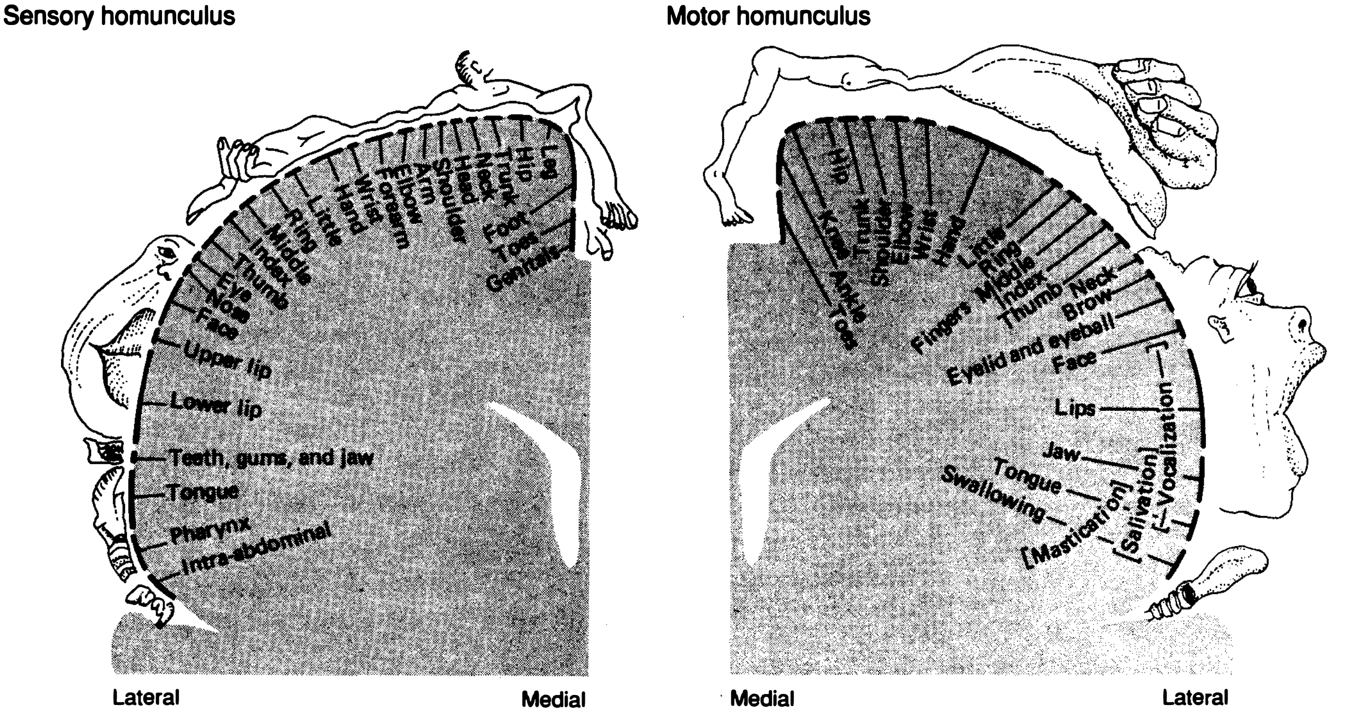

Figure 1: The sensory and motor homunculus for the human brain. This shows the mapping of the body's afferent senses onto specific locations of the cortex, and the mapping of efferent control of the body's muscles from the cortex. (Reprinted with permission from Kandel, Schwartz, and Jessel, *Principles of Neural Sciences*, 3rd Ed., Appleton and Lange, Norwalk, CT. 1991. Adapted from Penfield and Rasmussen, *The Cerebral Cortex of Man: A Clinical Study of Localization of Function*, Macmillian, 1950)

sounds can be used to indicate the status of each software module, encoding erroneous performance as discordant chords.

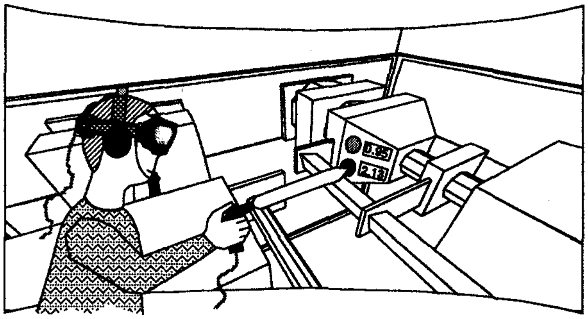

Figure 2: The concept of a person immersed within a virtual environment representing the software that controls a robot.

Why is the creation and study of such an interface necessary for the study of complex dynamical systems? A common direction of the fields of neural networks and neurobiology is to develop mathematically detailed models of the behaviors of biological neural systems. Due to their intrinsic complexity and nonlinearity, these studies are usually confined to simple animals such as marine mollusks and insects [7]. In principle, the knowledge produced in the study of simpler systems can, in turn, be used to deepen our understanding of more complicated biological systems and as an aid in the engineering of artificial neural systems. In practice, though, understanding the dynamics of even simple biological or artificial systems is currently very difficult, due primarily to the quantity, the temporal dynamics, and the high intrinsic dimensionality of the data.

To illustrate this point, we will discuss a class of artificial systems called autonomous perceptual systems. We define an *autonomous perceptual system* as an automatic sensor-based system that processes and combines multiple types of sensor data, provides task-based interpretations of the sensor data, and interacts with the environment to accomplish its overall objectives. Later, we will present an example of a autonomous perceptual system under development, called the *Encephalon.*

Many examples of autonomous perceptual systems exists in the literature [8-17]. Bachelder and Waxman [18] have demonstrated on the robotics platform MAVIN, visual navigation of simplified 3-D environments and the learning of 2-D mappings that represent spatially distributed visual landmarks. A common characteristic of all of these systems is the bi-directional interaction of a macrocircuit of complex nonlinear neural modules with an uncertain sensory environment. In isolation, these neural modules can usually be represented and understood as either mathematical models or algorithms. When combined into an autonomous perceptual system, the resulting system becomes difficult, if not impossible, to characterize mathematically, and therefore is usually studied empirically. This is particularly true when the system interfaces with the sensors and actuators of a real-world robot, adding new levels of nonlinear interaction and noise. In some cases, the combined system can lead to chaotic and unpredictable behaviors not found in the component modules [19]. Investigating and understanding the dynamics of complex, high dimensional systems is a major hurdle in the path between basic research and actual implementation of practical, robust autonomous perceptual systems.

The Homunculus will provide an efficient VR environment in which we can hypothetically conduct empirical studies of complex, interacting systems. Modules of the neural network (e.g. the Encephalon) will be represented in the Homunculus as 3-D Khoros-like glyphs [20] or icons, which can be developed external to the virtual environment on conventional computer workstations. A glyph in this context is an executable software process, represented in iconic form as part of a visual programming language in a graphical user interface. Glyphs are graphically connected together by lines representing information channels to produce programs. Information flow through the network will be represented in the virtual environment as graphical animations, allowing visual monitoring of the computation during execution. Auditory, haptic, and proprioceptive human sensory modalities, and telepresence will also be integrated into the virtual environment to aid in interpretation of the learned neural memories and the control and monitoring of the robot. The next section will introduce virtual reality technology. Sec. 3 will discuss the Khoros software engineering tool we are using for neural network system software integration and testing. Sec. 4 gives the details of the complex biologically motivated neural architecture called the Encephalon under development, illustrates how it is being coded in Khoros, and shows one possible 3-D representation in the Homunculus. In addition, this section will discuss the small robot to be controlled by the Encephalon and its operational environment. In Sec. 5 we will discuss how this approach blends the best of human perception and machine automation to better our comprehension of complex dynamical systems and multidimensional data. Several research issues and open problems generated by our approach are presented as well. Sec. 6 wraps up the paper with a summary.

2. Virtual Reality Interfaces

When software modules are connected into complex nonlinear systems, it can become very difficult to characterize and visualize their emergent behaviors. Typically, empirical methodologies are used to study these systems, leading to large numbers of numerical experiments with differing sets of parameters and input conditions. With complex macrocircuits of modules and high dimensional state spaces, specifying these experiments and presenting the results can be a daunting task with current computer interfaces and visualization tools.

In general, investigating and understanding the content of multidimensional information is a major impediment in the application of basic research results. The scientist remains separated from his/her data, with the computer acting as a recalcitrant intermediary. For example, as our understanding of neuroanatomy has become more sophisticated, scientists have had to deal with more and more information in order to accurately understand and represent real biological systems. Complex, multi-dimensional data sets arise experimentally, from a variety of sensors or measuring devices, or computationally, from the results of computer calculations or models. Automated analysis techniques, such as pattern recognition, image segmentation, and marching cubes [21], designed to help with the analysis process, are not typically robust enough to automatically decipher this complex data. On the other hand, when human experts study a data set, they naturally focus-in on details important to their investigation and ignore irrelevant information. What is required of human-computer interfaces is an approach that allows humans and machines to do what they each do best. We believe that future developments in experimental sciences will *critically* depend on the development of more effective human-computer interfaces that are designed to use significantly more of our natural human reasoning and perception in the analysis process.

Virtual reality provides a technological basis for such an improved human-computer interface. VR is defined as an advanced human-computer interface technology that embodies a sense of immersion, interactivity, navigation, and exploration of computer generated virtual worlds [22]. The immersion of human senses into data sets, improving a person's ability to explore, interact with, perceive, and understand computer-based information is one goal of VR technology development. VR systems typically involve high-performance graphics computers, opaque head-mounted displays that provide real-time stereo views of the graphics, six degree-of-freedom head tracking, tracking of the hand and fingers or hand-held virtual tools, 3-D binaural sounds, and force or haptic feedback devices.

Current human-computer interfaces are functional but, in many ways, still primitive [23]. Almost all interaction takes place through 2-D screens, keyboard and mouse input, and window-based menus. For example, PCs and workstations provides a simple "desktop" interface that attempts to enable the same kind of functionality in software that is available on a real desktop, with folders and icons for different file types. Although a big improvement over punch cards and paper tape, most computer interface technologies available today are restrictive, unnatural and relatively inefficient for human use. In such an interface, the mapping of multi-dimensional data or real world inputs to the human senses is primarily 2-D visual, with interactivity restricted to the mouse or keyboard (see Fig. 3).

In contrast, a virtual environment has the potential to allow the scientist to view, manipulate, model, analyze, and navigate through complex data sets and systems as though actually in the experimental domain. Consider the dimensional richness in this interface, both for input and output (Fig. 4). In addition to the older methods of interaction, the user can now manipulate data and modify software architectures by gesturing with his/her hands or hand held virtual tools, moving his/her head or eyes, or perhaps most importantly, using speech. The user can view data in 3-D graphical representations via stereo head-mounted displays, hear sounds corresponding to 3-D events [24], or feel the virtual world with tactile and force-feedback [25,26] in a "natural" environment recreated by the computer.

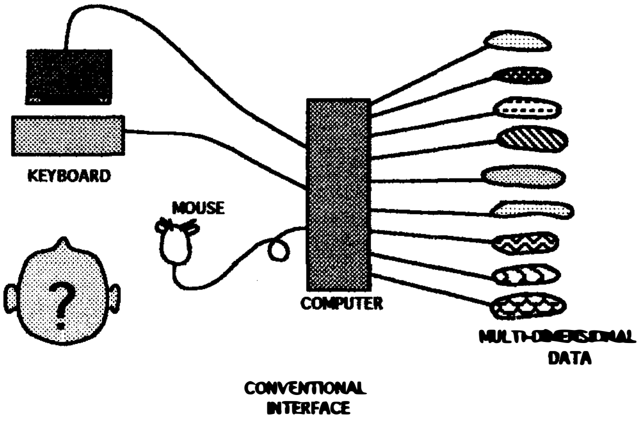

Figure 3: A conventional interface with mouse, CRT, and keyboard. For multi-dimensional data and complex software architectures, conventional interfaces are not always adequate.

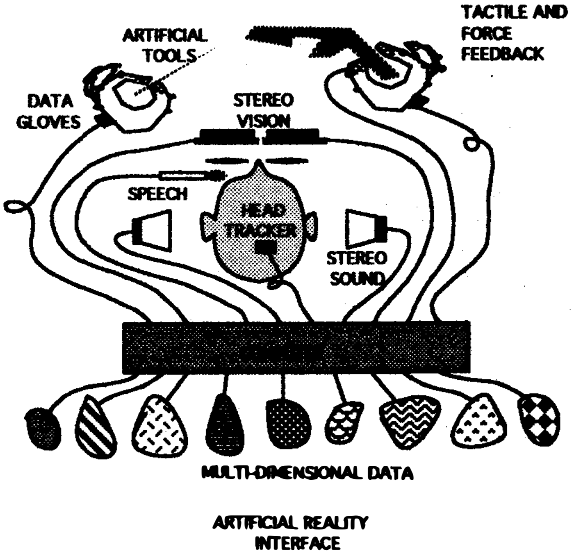

Figure 4: An ideal Virtual Reality interface for multi-dimensional data visualization. On top, electromagnetic trackers on head and gloves report X, Y, Z, roll, pitch, & yaw information. Data gloves give the position of fingers and thumb, while force and tactile feedback provide extra sensitivity. Stereo head mounted displays and headphones place the user in a computer-generated environment which can be manipulated in a direct, natural manner.

The unnatural interface of today's computers is a significant bottleneck to efficient use in applications that involve interpretation of high dimensional spatio-temporal data sets and the control of sophisticated dynamical systems. For example, consider the case of a robot arm and accompanying control computer. In many applications, a robot arm is controlled using commands or sequences of commands typed from a keyboard or with numerous joysticks and buttons. Not only is this an unnatural interface, it is difficult to use, even for trained individuals. On the other hand, using head-mounted display, body tracking, and speech recognition in a virtual environment puts the computer in the domain of the user rather than the user in the domain of the computer. By mapping the movement of arm joints and end effectors to the position and orientation of the user's arm, hand, and fingers, the user can program the robot's arm as if it were his/her own [27]. Another form of interaction using speech recognition is being explored in the control of remote robots (called telepresence or telerobotics) for applications in toxic waste cleanup [28]. The efficiency of use of a computer may be substantially improved through virtual environments that bring the computer and human together in an environment natural to the human.

Given that VR offers a more natural interface to the computer, a software environment must be developed that takes advantage of the richness of this interface. The *Khoros* software development environment is one such tool.

3. Khoros 2.0 for Modular/Hierarchical Design

The virtual environment of the Homunculus offers a place to conduct empirical studies of complex software systems. To effectively use this environment, tools must be available to represent, modify, diagnose, and visualize the software system [6]. Khoros is such a tool that lends itself to this application.

Khoros is an integrated software development environment for information processing and visualization. Khoros was developed at the Department of Electrical and

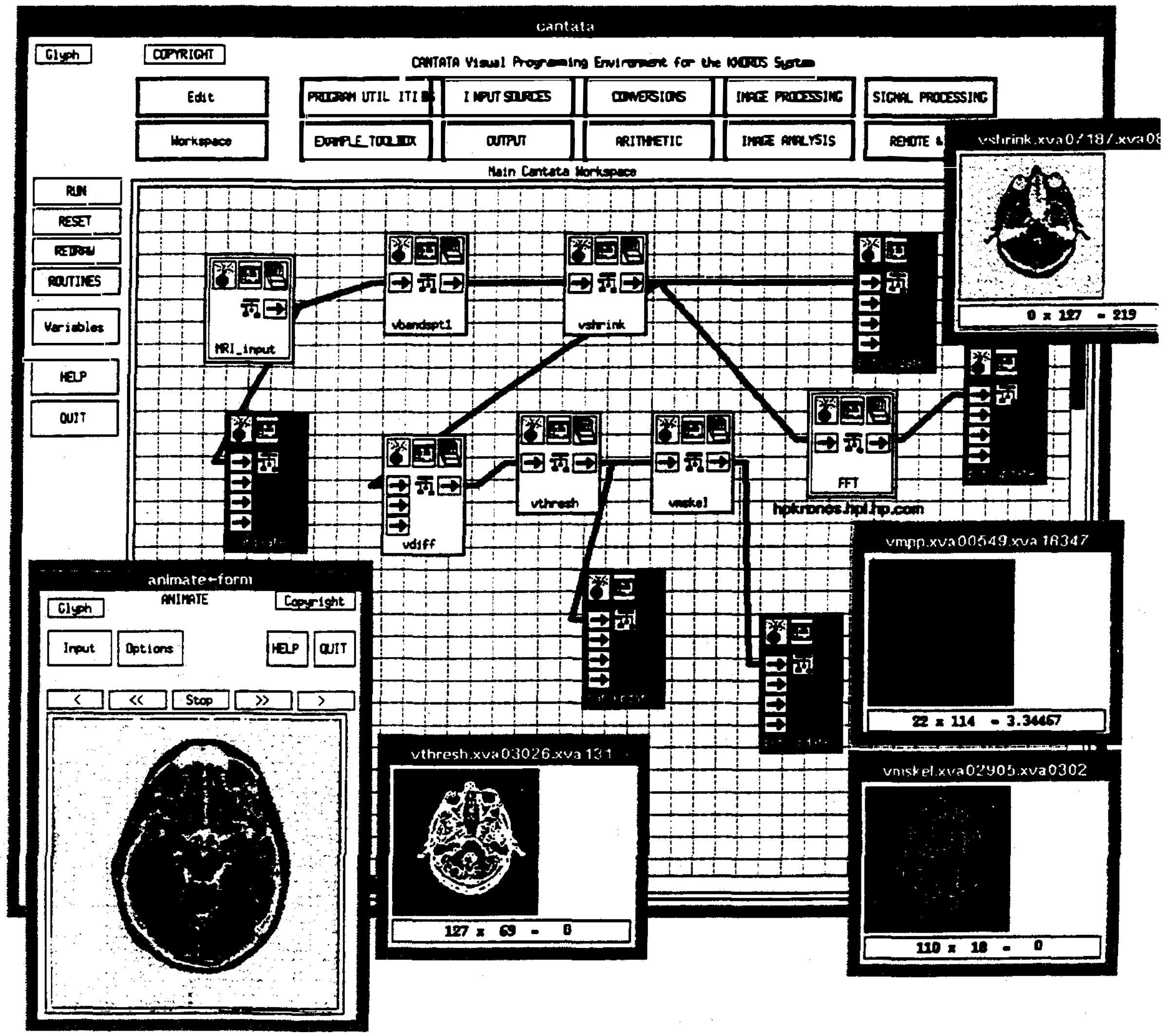

Figure 5: A block diagram of an image processing application in Khoros.

Computer Engineering at the University of New Mexico between 1987 and 1993, and continues today at Khoral Research Incorporated. The system includes a visual programming language, called *cantata* [29], code generators for extending the visual language and adding new application packages to the system, an interactive user interface editor, interactive image display programs, surface visualization, an extensive library (over 260 routines) of image processing, numerical analysis, signal processing routines, and 2-D/3-D plotting packages. In a networked environment, Khoros allows distributed computing and provides support for collaborative computing by allowing multiple users to simultaneously share data and workspaces.

Fig. 5 shows a snapshot of *cantata*, the programming environment for the Khoros system user. Cantata is a general-purpose programming language based on a data flow paradigm with additional support for conditionals, iteration, and sub procedures. The term "data flow" is used rather loosely here since it communicates the basic idea behind the cantata visual language but does not account for all of its features [29]. In cantata, coarse-grained UNIX processes are graphically represented as glyphs in a 2-D workspace and communicate through interconnection links to build application programs. Glyphs can communicate via several user selectable transport mechanisms, such as temporary files, sockets, or shared memory. Temporary files are the default transport, because all architectures support them and they provide data permanence. Permanence is important for saving the state of the visual program so that a program can be resumed after it has been stopped. In Khoros, the entire workspace, the data flow, and intermediate results can be saved and restored for later use. If the computer where Khoros is running supports shared memory, then substantial processing speedup can be achieved. UNIX sockets are used for remote or distributed transport (file-to-file or shared-memory-to-shared-memory) as well as for local transport without permanence.

During execution, glyphs and their input/output ports are highlighted as they are running to provide the programmer with an execution view. For example, in Fig. 5 the "active" glyphs (the ones that display data) are shown in reverse video. (These glyphs are considered "active" in the sense that they are still running UNIX processes). If an error occurs, the glyph is highlighted with an error icon, and the error message can be displayed on the screen. Since cantata supports permanent data transport, single stepping is provided. The user can execute one operator or glyph at a time. Break points can also be set using the flow control glyphs (if-then, trigger, etc.). For example, an if-then glyph inserted in the data flow path will block the flow of execution if the expression evaluated by the if-then statement is not true. After the execution is stopped, the user can view the latest state of data and control values at any node of the flow graph.

Cantata allows an hierarchy of workspaces so that the visual complexity of large data flow graphs can be reduced. Multiple glyphs can be encapsulated into a cantata workspace which can be used as a regular cantata glyph at the next higher level of abstraction. This paradigm maps well onto the neural design principle of hierarchies and modularity [5, 30]. The visual hierarchy is not implemented as a "graphical grouping", as in some other systems, but as a sub-procedure that supports local and global data and variables. Clicking the mouse on the middle icon of a sub-procedure opens the sub-procedure workspace and allows a user to edit and execute glyphs in that workspace as usual. Another feature of Khoros is distributed computing. Applications that are built using Khoros tools will run as a network application, utilizing resources as appropriate.

These features of Khoros makes it an ideal software engineering tool to support the visualization of the neural network software within the Homunculus interface. We will now discuss how all these pieces fit together in the current prototype system under development.

4. Details of System Under Development

In this Section we shall discuss in more detail the Homunculus system under development at the University of New Mexico. The components of the prototype are shown in Fig. 6, and are at varying stages of completion. They consist of a) the Encephalon neural architecture, b) the Khoros workstation environment for code development and specification of architectural configurations, c) an X protocol translator to allow communication between Khoros and the virtual environment, d) the virtual environment, where the human experimenter is immersed for the purpose of interacting with and visualizing the 3-D representation of the Encephalon, e) the robot and it's physical environment, and f) a video telepresence system to allow monitoring of the robot from within the virtual environment. We will now describe each of these components in greater detail.

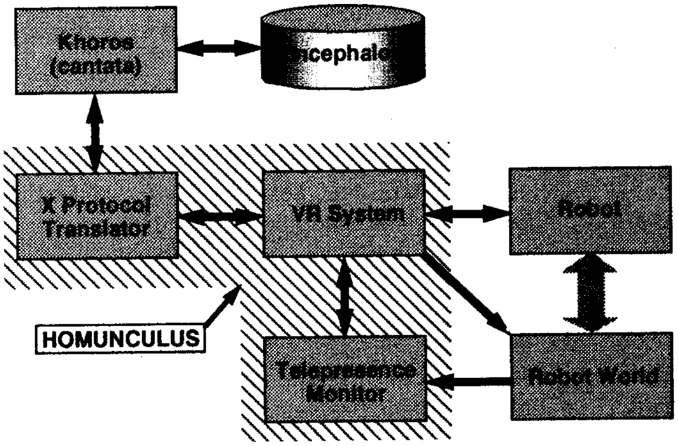

Figure 6: The components of the system currently under development consist of the Encephalon coded in Khoros, a X protocol translator, the virtual reality system, the robot and it's simplified environment, and the telepresence monitor. The Homunculus is indicated by the cross hatched area of the diagram. The broad arrow between the robot and it's world indicates a physical interaction.

4.1 THE ENCEPHALON AUTONOMOUS PERCEPTUAL SYSTEM

In this subsection, we present a neural-based autonomous perceptual system called the Encephalon [31] (Fig. 7), a version of which will serve as the first complex software system to be represented in the Homunculus. The Encephalon makes extensive use of adaptive resonance theory (ART) artificial neural networks and their operational principles [32]. This is a hybrid system consisting of neural network modules for sensor processing, code compression, working memory, inference, control, and a 3-D model database. It is unique in that the major image understanding functions are all performed by the neural network modules. The system is multilevel, with a separate chain of modules for each individual sensor modality, except at the uppermost levels where information fusion takes place. The neural network modules are designed to function with other neural networks as system components for composite networks. Some of them have been investigated as stand-alone networks in other applications [33, 34].

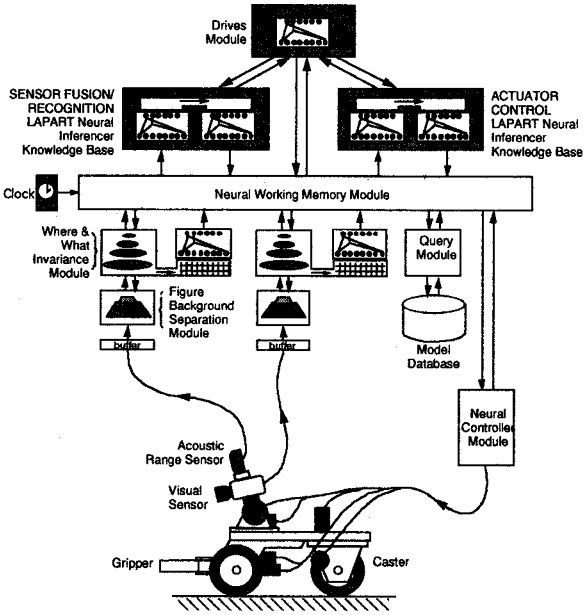
Drives Module
SENSOR FUSION/ RECOGNITION LAPART Neural Inferencer Knowledge Base
ACTUATOR CONTROL LAPART Neural Inferencer Knowledge Base
Clock
Neural Working Memory Module
Where & What Invariance Module
Query Module
Figure Background Separation Module
buffer
buffer
Model Database
Neural Controller Module
Acoustic Range Sensor
Visual Sensor
Gripper
Caster

Figure 7: A block diagram of an Encephalon architecture interfaced to the robot. ART class networks and their design principles are used throughout the various modules.

The system modules are as follows, beginning at the sensor level and working up through the system. Each sensor (e.g.. vision, range, sound) provides input to a Figure / Ground Separation Module (FGSM). This module pre-processes the raw data to provide the equivalent of object boundary contours and color and/or textural fill inside boundaries. Noise suppression, completion of partially formed features, and filtering of background clutter occurs at this level [35]. This is aided by feedback from the next higher level module in the sensor's individual multilevel chain. Each sensor's processing is isolated at this stage, except for the indirect influence of other sensors in the top-down feedback pathways. In the current visual channel the FGSM is an enhanced implementation of the CORT-X network, described in [36]. The CORT-X system is a multilayer network whose output layers represents an input image as a collection of noise reduced, spatially registered objects isolated from background.

The FGSM output activation pattern is input to the Where / What Invariance Module (WWIM) at the next higher level. The WWIM selects objects in a manner that is invariant to spatial location, scale, and orientation in the image [37]. It is a multilayer hierarchical network providing two major pathways to the Working Memory Module (WMM), where selected objects are recombined to form the highest-level abstraction representing the sensor image. The WWIM contributes to efficient object recognition by splitting the tasks of feature extraction and recognition into parallel channels. The extraction component is a multilayer neural network called the Feature Mass Detector [38].

The Feature Mass Detector (FMD) is a neural architecture for efficient extraction of invariant features placed in arbitrary position, scale, and rotation in the field. The architecture provides versatility in invariant selection with minimal computation and storage requirements. The FMD autonomously extracts fixed-size subsets of input pattern components based upon total activation (i.e.. "mass") in the receptive fields of detector nodes. (For example, mass could be defined as contrast density.) It extracts a subset of input pattern features by sampling a fixed number of nodes in the input pattern through any of a set of pre-defined subset selection masks (Fig. 8). A mask is a set of connections from the input layer to the mass detection nodes, which are activated in proportion to the amount of input they receive. Each mask's coverage of the input pattern defines the scale at which it "sees" an object, somewhat mimicking the magnocellular and parvocellular circuits in mammalian vision systems [39]. All masks have the same, fixed number of connections to their respective detector nodes. Hence a large mask contracts the size of an object while a small mask expands the size, "zooming in" on the object.

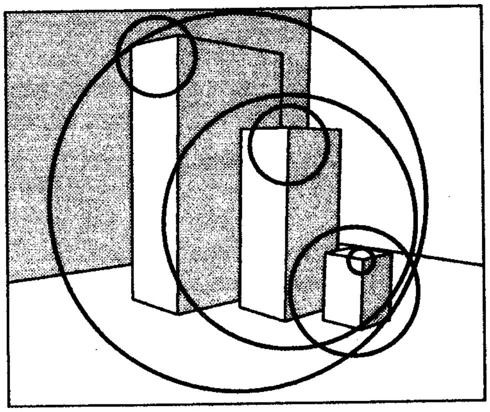

Figure 8: A hierarchy of receptive fields for detector node masks, leading to the extraction of structured features by the Feature Mass Detector.

The FMD has both a "dumb" and a "smart" operating mode. "Dumb" is the nominal mode, in which mass activation determines the order in which features in the image are examined. The "Smart" mode occurs in active sensing; feedback from the higher pattern recognition systems operating upon working memory is used to guide the sequence of feature examinations. The mass sensitive nodes in the FMD can be activated preferentially or primed by feedback, based upon previous pattern recognition decisions. Alternatively, the examination of a preferred feature region can be boosted by feedback.

The current subset selection is based upon competition among the detector nodes, which are contained in multiple layers representing different object scales. The winning detector node has maximal activation, representing the currently most active set of masked nodes in the FGSM image. It primes an associated set of mask nodes. These act as relays, transmitting the currently-masked portion of the input pattern to an ART network when primed by a detector node. The ART network codes the object's feature, which is now centered and represented at a fixed scale in its input layer. The ART network's input layer has the same number of nodes as the fixed sample of each mask, relayed through the mask nodes. A coding node in the ART network represents the currently-recognized object feature through its activation, which is relayed to the WMM. The corresponding spatial location and scale is transmitted separately to the WMM by the active detector node. Rotation is also abstracted from the object before reaching the ART network.

Thus, the FMD is actually "grabbing" icons that are organized in hierarchies and coded by the output patterns of the ART modules. Each winning detector node in the FMD is stunned by a system analogous to the ART vigilance subsystem [32], once its input field region has been forwarded to the classification network. In "dumb" mode this allows the detector node with the next most active input field region to become the new winner and forward its portion of the input pattern through the mask layer. In this way a sequence of object features is masked and coded, with the codings and the corresponding spatial locations, scales, and rotations transmitted to the WMM. The WMM consists of a multiple layer system of self-excitatory working memory nodes. The self-excitation enables a working memory node, once excited, to remain so until deactivated by a signal from the inferencing network (the Laterally-Primed Adaptive Resonance Theory (LAPART) network [40]) or the top-level Drive Module.

The WMM consists of a system of nodes representing each sensor's abstracted features, their locations, scales, and rotations within the sensor field, and the class to which each has been assigned by the associated ART network. Here the information is processed through the interaction of four modules: a LAPART inferencing system for hypothesis generation, a LAPART inferencing system for sensor and actuator control, a pre-programmed 3-D model database, and the top-most level Drive Module. The WMM patterns are interpreted by the first LAPART inferencer to form hypotheses that are placed back in the WWM. This action, combined with top-down feedback, leads to a chain reaction of hypothesis confirmation-disconfirmation cycles. The second LAPART inferencer interprets the current state of the WMM and provides feedback through a controller to the appropriate actuators. For example, sensor pointing can be modified to improve registration of separately-identified objects, based upon an object hypothesis generated by the first LAPART. The pre-programmed database stores 3-D model based information to aid in hypothesis confirmation.

LAPART is a neural network architecture capable of recognizing sequences of patterns (LAPART is similar to ARTMAP in many ways [41].) This architecture defines a class of networks that can be trained to associate classes of patterns appearing in a sequence. This is made possible by an adaptive neural inferencing mechanism that uses a pattern from a class in a learned sequence to predict the next class. The basis for the sequence recognition function of the LAPART architecture is the coupling of two pattern classifier networks through a system of lateral interconnects. The interconnects implement a dual system of inference rules. The LAPART system learns the inferences during presentation of training pairs of patterns through (1) ART pattern classification involving synaptic learning within each classifier network, and (2) synaptic learning of the class-to-class inferences through interconnects. A LAPART system can remain adaptive following training, continuing to learn from observed patterns. Distinctly novel inputs are automatically classified separately from those encountered in the past. Loosely speaking, the network "knows" when it has encountered a pattern sequence that lies outside its trained generalization capability. LAPART has many potential applications, including supervised learning [40].

The Drive Module is a network that provides top-down direction to the LAPART inferencers. Through a competitive interaction, a specific goal for the system is selected, based on bottom-up inputs and preprogrammed priorities. For example, competition between the drive for food (energy) and the drive for sleep (off-line learning) determines the extent and degree of top-down priming that will lead to either a "search for food" or a "return to home" behavior in the robot. An ART network serves this role in the current system.

4.2 THE ROBOT AND ITS WORLD

Because our research goals are focused on extracting and refining *engineering principles* for the design of more advanced autonomous perceptual systems, it is essential to capture the variability and noise of real sensor data. Therefore the Encephalon will be interfaced to a physical robot that is under development in our lab (Fig. 9). Since we are not concerned at this time with free roaming behaviors in unstructured environments, no effort has been expended to make this robot computationally self-contained. It basically acts as a sensor and actuator platform, connected to the computer network and power source through a set of cables. In addition to simplifying the robot's design, this approach allows it to be physically small compared to its environment. The robot has two fixed wheels and one steerable powered caster. The sensors will ultimately consist of a CCD color camera, an acoustic ranger, four microphones, a gripper pressure sensor, and a set of whiskers around the base. Actuators will consist of caster motors, pan/tilt motors for the camera/acoustic sensors, and gripper control motors.

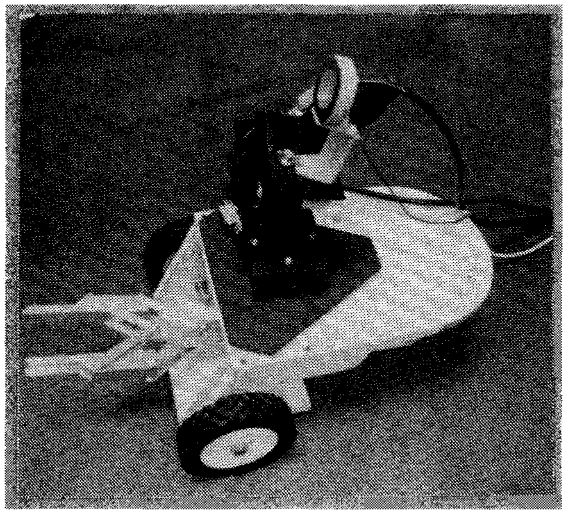

Figure 9. Photograph of Valentino One, a three wheeled robot under development which will provide sensory data to and receive control commands from the Encephalon. In order to make a compact system, the robot has no on-board computing. The sensors will ultimately consist of a CCD color camera, an acoustic ranger, four microphones, a gripper pressure sensor, and a set of whiskers around the base. Actuators will consist of caster motors, pan/tilt motors for the camera/acoustic sensors, and gripper control motors. (Photograph by Greg Donohoe)

In order to conduct highly controlled experiments on the Encephalon, the robot will be confined to a finite and structured world [42], illustrated in Fig. 10. This is a simple "blocks world", with the potential for introducing artificial day/night lighting cycles, sounds, and external objects. In experiments that involve the ramifications of imposed higher level "drives", such as foraging and nesting behaviors, "food" objects can be introduced into the robot's world at random locations and times. The primary requirement for this world is the ability to *reliably reproduce* the robot's sensory environment from experiment to experiment.

Since the experiments planned for the Encephalon and the robot are to be conducted from within the virtual environment of the Homunculus, a telepresence system will be used to monitor the actions of the robot. The device, pictured in Fig. 11, is called a *Molly* from FakeSpace Inc. (Menlo Park, CA)., and consists of a pair of CCD cameras on a high torque pan/tilt/roll drive. Using head tracking information, the Molly servos the gaze direction of the cameras to match that of the user, allowing a stereoscopic view of the surrounding environment from within the immersive head-mounted display. One advantage of incorporating telepresence in the Homunculus is that the user may conduct experiments remotely from the robot lab. This will also allow the experimenter to observe the robot's behavior without "contaminating" the sensory inputs during a learning session.

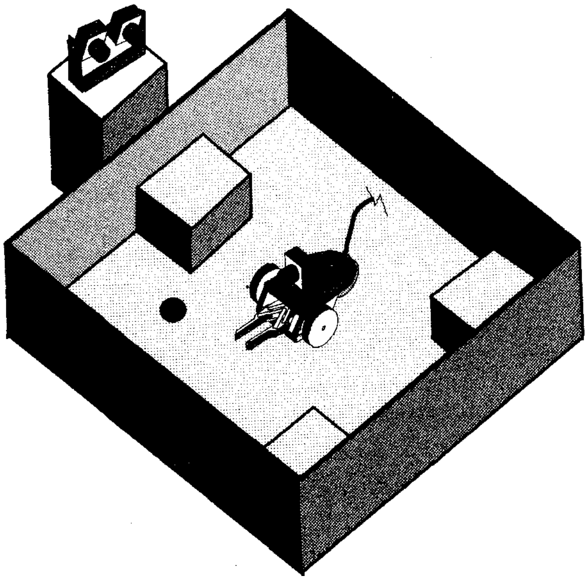

Figure 10: Illustration of the simplified world within which the robot will operate. Lighting and sound levels will be controlled. A cable support system will servo to a position above the robot. The telepresence system allows monitoring of the robot and it's environment from within the Homunculus.

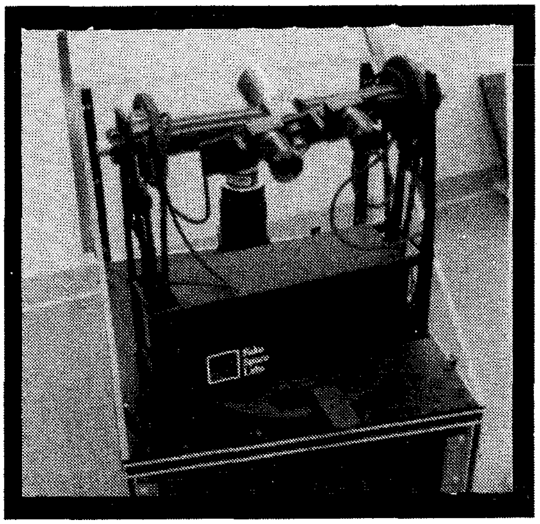

Figure 11: Photograph of the Molly, a telepresence system that servos a pair of CCD cameras.to point in the direction of the users head. (Photograph by Greg Donohoe)

4.3 KHOROS 2.0 REPRESENTATION OF ENCEPHALON

A version of the Encephalon has been coded in Khoros and one possible graphical layout is shown in Fig. 12. Each glyph represents a functional block in the algorithm as depicted in Fig. 7, and the links between blocks represent data flow pathways. The UNIX process represented by each glyph may be executed anywhere on a network of computers, which could include workstations, parallel computers, and special purpose hardware, such as vector processing systems.

4.4 THE X PROTOCOL INTERPRETER

The cantata graphical user interface for Khoros 2.0 is based on the X protocol client/server model [43]. This allows platform independent graphical user interfaces to be developed and interfaced with the basic system code of Khoros. In order for the VR environment to reflect changes in the cantata visual representation of the executing code, the normal client/server communication pathway of cantata must be tapped. A similar system has been developed at Sandia National Laboratories to make collaborative environments possible for design engineering [44]. We plan to use a variant of this technology to intercept X protocol requests and acknowledgments between the cantata client and server, parse the requests, and translate them into commands to the VR system. This approach decouples the VR system from Khoros, allowing for independent software development.

4.5 THE VR SYSTEM

Just as the desktop is used as a metaphor for today's computer interfaces, the Homunculus will initially use a 3-D "engine-room" model of the environment. Raw sensor data will be projected on the walls of the room, while the neural network software, represented as a blocky engine, will float in the middle of the room. A rendering of this can be seen in Fig. 13. The experimenter will be able to navigate around the room with a set of diagnostic tools, for example, probing the neural network, while visualizing intermediate computational results and the resultant physical behavior of the robot.

There are many ways to represent software programs in virtual environments [6] just as there are in conventional interfaces. One possible form of the Encephalon is shown in more detail Fig. 14. In this model, rectangular boxes stand for Khoros-like glyphs with the interconnecting pipes representing data flow pathways. Cantata workspace hierarchies are represented as nested rectangular boxes into which the user can navigate, allowing multiple levels of abstraction of the software.

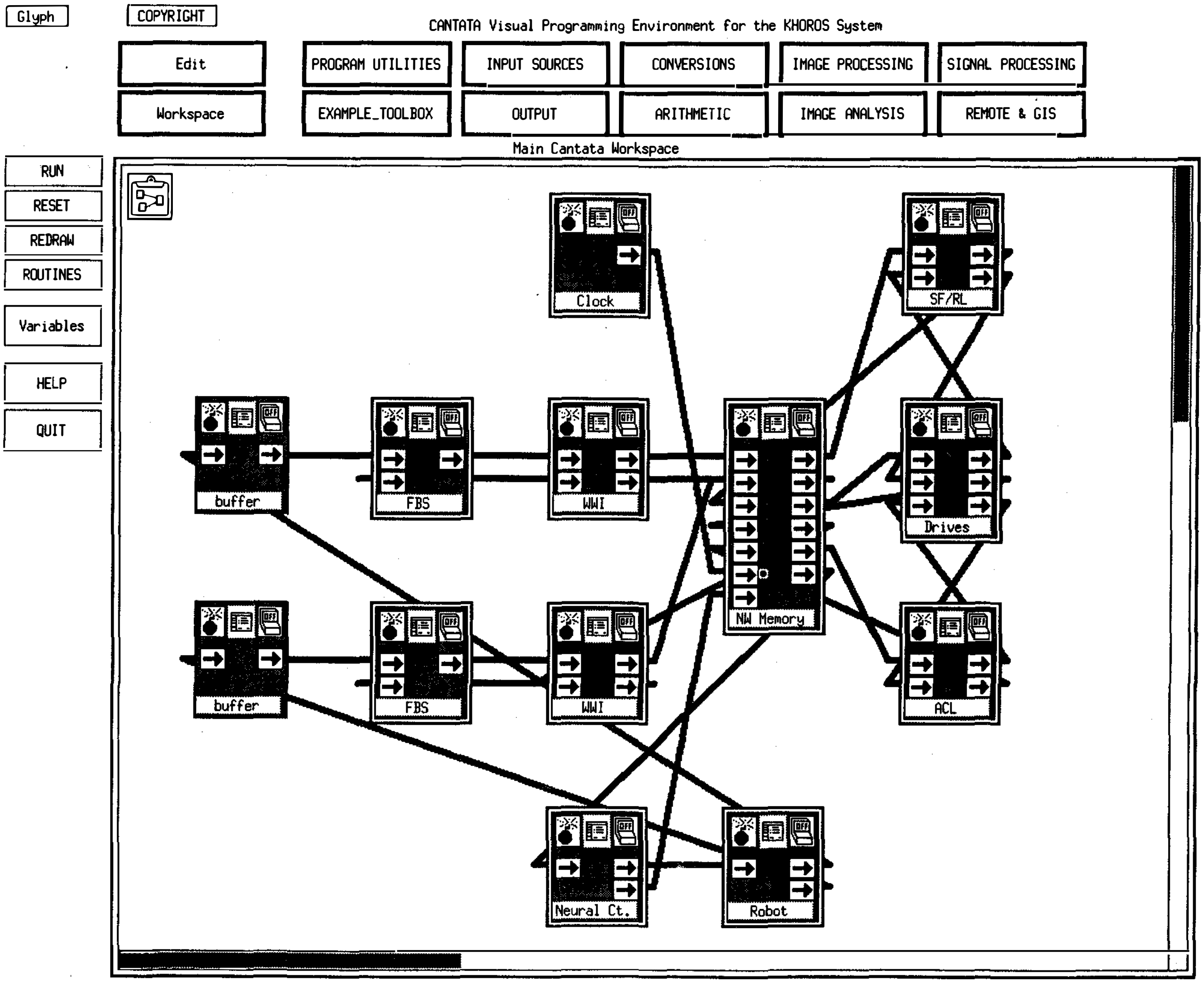

Figure 12: A cantata screen snapshot showing the top level modular layout of one version of the Encephalon.

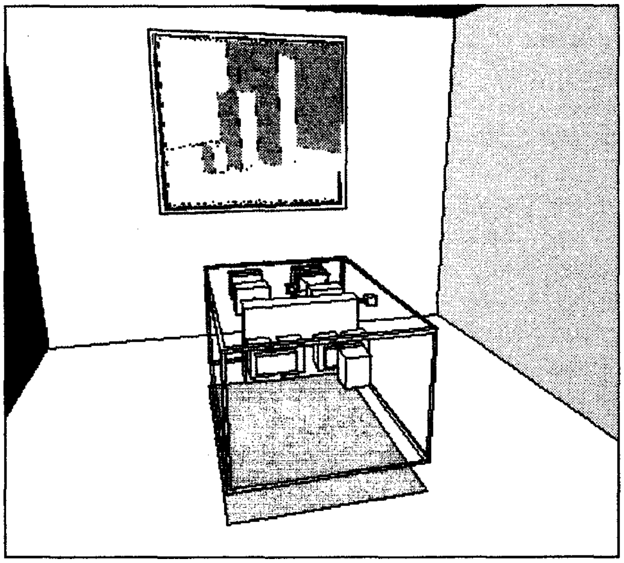

Figure 13: A view of a 3-D graphical model of the Encephalon within the Homunculus. Each block represents a glyph in the main cantata workspace. Information transfer between blocks will be represented as animated pulses of color moving down pipes. Raw sensor data will be viewed on the walls in this "engine room" metaphor for the Homunculus.

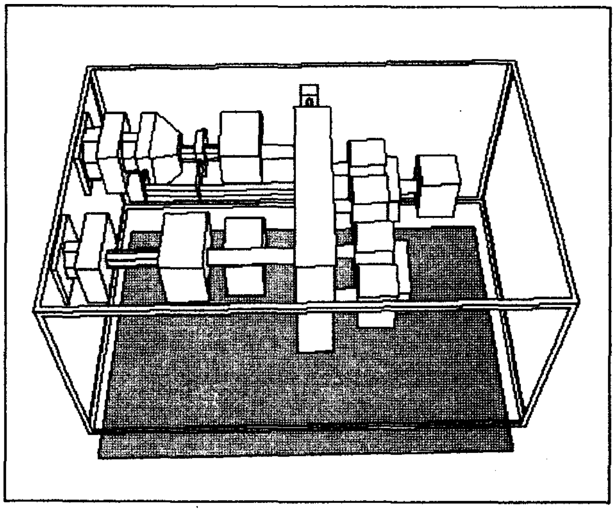

Figure 14: A close up on the representation of the Encephalon in the "engine room". Each block can be entered by the experimenter to expose sub-components of the UNIX process. In cantata, these sub-components are workspaces in their own right.

The flow of information through the system will be modeled after the cantata interface in Khoros. Blocks will change color as data enters and exits the module. The flow of data packets will be represented as moving blocks of color along the pipes. In addition, 3-D binaural sound will be incorporated into this environment. Operations of the program may be associated with sounds spatially stabilized within any block. The type and quality of the sound may code operational conditions of the algorithm. For example, the block may emit a resonant sound when operating normally, but become discordant when an error occurs or the data moves out of bounds.

This virtual environment is being designed to empower the experimenter to view, manipulate, model, diagnosis, analyze, and navigate through the robot's controlling neural network as though actually in the experimental domain, in this case, the robot's "brain". Considering the richness of this domain, we are faced with many questions as to how it should be best used. This will be further discussed in the next Section.

5. Discussion

Our understanding of autonomous perceptual systems is limited by the computational tools available today. Complex programs and multi-dimensional data sets frequently arise experimentally, from a variety of sensors or measuring devices, or computationally, from the results of computer calculations or models. Investigating and understanding the content of such information is currently a major roadblock between basic research and actual application of these results to future artificial neural systems. The scientist still remains separated from his/her simulations and data, with the computer acting as a recalcitrant intermediary.

In this paper, we have outlined an approach where the scientist will be able to view, manipulate, and move through the "brain" of a robot as though actually immersed in that abstract world. This approach blends the best of human perception and machine automation to better our comprehension of complex dynamical systems and multidimensional data sets.

Using this approach, an experimenter can fly through the virtual environment consisting of raw sensory data, observe inter-connectivity of blocks within the program, dissect individual elements from the code, and examine the resulting change in performance. As the experimenter moves closer to an element or group of software blocks, the volume could be augmented to include sub-components of the blocks and additional information on parameter settings. The multimedia feature of the environment will provide 3-D sound cues corresponding to operational activity. In the virtual environment an experimenter would listen for synchrony to find blocks that drive other blocks with phase delays. Thus, he/she can discover nuances that make the circuitry function by listening, for example, for beat frequencies. Only in such an environment could these types of experiments and visualizations be effectively performed.

As we develop this tool to improve our understanding of one domain, we inevitably expose a multitude of research questions in another domain. - Many research issues must be addressed before VR technology can be efficiently applied in the ways discussed above. Because these issues are not commonly presented, a number of them are listed here:

a) *Accurate Unobtrusive Position Tracking.* Current VR position tracking systems are bulky, limited to roughly room sized volumes, and require "clean" environments for accurate operation.

b) *Effectiveness of Force Feedback.* What role does force feedback play in the use of virtual tools within the Homunculus, and how do we build the actuators.
c) *Fusion of Data and Graphics.* How do we bring live raw data, such as video images, into a virtual environment and paste them onto the walls of a room?
d) *Efficient Graphics Render.* Given the potential graphic complexity of the "engine room" metaphor combined with the requirement to display live data, how can we improve the efficiency of the graphics rendering engines?
e) *Representation of Tools in VR.* Given the need for a set of virtual tools, how should we represent the tool boxes or menus within the virtual environment? Should we use a pallet, pull down menus, voice commands, or some combination?
f) *Coding of Information in Sound.* What types of sounds are best for representing information, abstract or reality based, useful to the experimenter?
g) *Navigation Methodologies.* An old issue in virtual environments: what is the best way to move your virtual head and body (often referred to as your "avatar") around in the 3-D volume and not get lost or disoriented?
h) *Requirements for Avatar.* How much of your body should be represented in the virtual environment, what should it look like, and how should it behave?
i) *Representation of Programs.* Is a 3-D block glyph the best way to represent software modules? What other models are possible?
j) *Effectiveness of "Engine Room" Metaphor.* How natural and effective will people find the engine room metaphor of the Homunculus? What room shapes and layouts are most efficient in getting the job done?
k) *Human 3-D Perception and Reasoning.* How do we best match the optics in the displays to the characteristics of human vision, such as depth of focus, optical pupil size, and eye strain? What role will color play in the display of abstract information?
l) *Human Task Performance Measurement.* How does VR technology affect a persons ability to understand the operation of complex dynamical systems? What types of tasks does it help or hinder? What system characteristics, such as computational time delays or position errors, affect measures of task performance?
n) *Human qualitative measurements.* How do people perceive the Homunculus workplace? Given a choice, will they use it or revert back to older methods of flat screen interfaces? Do these factors effect the quality and quantity of their work? Finally, can these effects be quantitatively measured?

Clearly there is much research to be performed. The development of the Homunculus will provide a testbed for experimenting with many approaches to practical virtual reality as well as broaden our understanding of autonomous perceptual systems.

6. Conclusion

This paper has described ongoing research in the development of a immersive virtual reality interface to a robotics system controlled by a complex artificial neural network. The interface is referred to as a Homunculus in that it allows a person to move freely about and interact with the robot's "brain". The neural network will be represented in the Homunculus as 3-D Khoros-like glyphs, which can be developed externally in a conventional Khoros environment. Data flow through the network will be represented as graphical animations, allowing visual monitoring of the computation. Auditory, haptic, proprioceptive, and telepresence modalities will also be integrated into the virtual environment to aid in interpretation of the learned neural memories and the control of the system. As a example, a complex biologically motivated neural architecture called the Encephalon was presented that makes extensive use of adaptive resonance theory networks. The Encephalon is a software perception system that autonomously learns object classification inference rules, and makes extensive use of the interplay between the bottom-up and top-down flow of information. A robot that will be interfaced to the Encephalon and its environment was also presented. In the end, we discussed the many research issues left to be addressed in the application of virtual reality interfaces to complex software simulations and as a place to conduct empirical studies.

7. Acknowledgments

I wish to acknowledge and thank John Rasure of Khoral Research Incorporated in Albuquerque for his valuable additions, corrections, and suggestions to this manuscript. I also want to acknowledge the contributions of Greg Donohoe and Chaouki Abdallah of the Department of Electrical and Computer Engineering of the University of New Mexico; Mike Healy for help with the Encephalon, Scott Smith, and Henry Sowizral of the Boeing Company; Mike Daily and Mike Howard of Hughes Research Laboratories; Nadine Miner, Sharon Stansfield, and Creve Maples of Sandia National Laboratories; Ray Hitts and Han Lin of the Department of Electrical and Computer Engineering at the University of New Mexico; and Kathryn Ann Caudell of the Department of Nursing of the University of New Mexico.

REFERENCES

[1] *Stedman's Medical Dictionary*, 24th Ed., Williams and Wilkins, Baltimore, 1982, p. 656.

[2] S. Ramachandran, "Perception: A Biological Perspective", *Neural Networks for Vision and Image*

Processing, G. Carpenter and S. Grossberg, Ed., MIT Press, Cambridge, 1992, Chapt. 3.

[3] W. Penfield and T. Rasmussen, *The Cerebral Cortex of Man*, Macmillian Pub. Co, 1950, p 57.

[4] E. R. Kandel, J. H. Schwartz, and T. M. Jessel, *Principles of Neural Science*, 3rd Ed., Elsevier, N. Y., 1991, p. 372.

[5] M. J. Healy, Boeing Comp. Services, TR# BCS-RT-CS-ACS-91-001, T.P. Caudell Ed., 1991, Ch. 3.

[6] H. Amari, T. Nagumo, M. Okada, M. Hirose, T. Ishii, "A Virtual Reality Application for Software Visualization", *Proc. of Virtual Reality Annual Int'l Sym. (VRAIS)*, Seattle, WA, 1993, pp. 1-6.

[7] J. Moya, M. Wilcox, G. Donohoe, "Modeling the first Two Neural Layers in the Common Housefly Vision System", *Proc. of World Congress on Neural Networks*, Portland 1, 1993, pp. 3-10.

[8] M. Siebert and A. M. Waxman, "Adaptive 3D object recognition from multiple views", *IEEE Trans. on Pat. Anal. & Mach. Intel. 14*, 1992, pp. 107-124

[9] A. A. Baloch and A. M. Waxman, "Visual learning, adaptive expectations, and behavioral conditioning of the mobile robot MAVIN", *Neural Networks 4*, 1991, pp. 271-302.

[10] H. P. Moravec, *Robot rover visual navigation*, UMI Research Press, Ann Arbor, MI, 1981.

[11] Z. Zhang and O. Faugerhaus, "A 3D world model builder with a mobile robot", *The Int'l. Jour. of Rob. Res. 11*, 1992 4, pp. 269-285.

[12] J. Tani and N. Fukumura, "Learning goal-directed sensory-based navigation of a mobile robot", *Neural Networks* 7, 1994, 3, pp. 553-563.

[13] K. P. Venugopal, A. S. Pandya, and R. Sudhakar, "A recurrent neural network controller and learning algorithm for the on-line learning control of autonomous underwater vehicles", *Neural Networks 7*, 1994, 5, pp. 833-846.

[14] J. A. Walter and K. J. Schulten, "Implementation of self-organizing neural networks for visuo-motor control of industrial robot", *IEEE Trans. of Neural Nets. 4*, 1993, 1, pp. 86-95.

[15] *Autonomous Robot Vehicles*, Springer-Verlag, NY., I. J. Cox and G. T. Wilfong, Eds, 1990.

[16] S. Nagata, M. Sekiguchi, K. Asakawa, "Mobile robot control by a structured hierarchical neural network", *IEEE Contr. Syst. Mag.*, April 1990, pp. 69-76.

[17] W. R. Uttal, "An integrated computational model of an autonomous, visually guided, undersea vehicle", *Proc. 3rd IEEE Int'l Workshop on Robot and Human Communication*, Nagoya, JAPAN, 1994, pp. 38-43.

[18] A. Bachelder and A. M. Waxman, "Mobile robot visual mapping and localization: a view-based neurocomputational architecture that emulates hippocampal place learning", *Neural Networks 7*, 1994, 6/7, pp. 1083-1099.

[19] H. Bai-Lin, *Chaos*, 1984, World Scientific, Singapore.

[20] K. Konstantinides and J. R. Rasure, "The Khoros software development environment for image and signal processing", *IEEE Trans. on Image Proc. 3*, 1994, 3, pp. 243-252.

[21] W. E. Lorensen and H. E. Cline, "Marching cubes: a high resolution 3D surface construction algorithm", *Comp. Graph. 21*, 1987, 4, pp. 163-169.

[22] C. Maples, Invited Session, *Virtual Reality Annual Int'l Sym. (VRAIS)*, Seattle, WA 1993.

[23] M. Daily, M. Howard, T. P. Caudell, Hughes Res. Lab. Tech. Rep, 1990.

[24] E. M. Wenzel, "Localization in virtual acoustic displays", *Presence 1*, 1992, 1, pp. 80-107.

[25] M. Ishii and M. Sato, "A 3D Interface Device with Force Feedback: A Virtual Workspace for Pick-and-Place Tasks", *Proc. of Virtual Reality Annual Int'l Sym. (VRAIS*, Seattle, WA, 1993, pp. 331-335.

[26] S. Tachi, T. Maeda, R. Hirata, and H. Hoshino, "A Construction Method of Virtual Haptic Space", *Proc. of 4th Int'l. Con. on Artificial Reality and Tele-Existence*, Tokyo, Japan, 1994, pp. 131-138.

[27] S. Tachi and K. Yasuda, "Evaluation Experiments of Tele-Existence Manipulation System", *Proc. of 3rd Int'l. Con. on Artificial Reality and Tele-Existence*, Tokyo, Japan, 1994, pp. 17-26.

[28] N. E. Miner and S. A. Stansfield, "An interactive virtual reality simulation system for robot control and operator training", *Proc. of IEEE Int'l. Conf. on Robotics and Autom*, San Diego 2, 1994, pp. 1428-1435.

[29] J. Rasure and C. Williams, "An integrated data flow visual language and software environment", *J. of Visual Lang. and Comp. 2*, 1991, 3.

[30] B. L. M. Happel and J. M. J. Murre, "Design and evolution of modular neural network architectures", *Neural Networks* 7, 1994, 6/7, pp. 985-1004.

[31] T. P. Caudell and M. J. Healy, "Adaptive resonance theory networks in the Encephalon autonomous vision system", *Proc. of the IEEE World Cong. on Comp. Intell. Orlando*, 2, 1994, pp. 1235-1240.

[32] G. A. Carpenter, and S. Grossberg, "A massively parallel architecture for a self-organizing neural pattern recognition machine", *Comp. Vision, Graphics, and Image Proc. 37*, 1987, pp. 54-115.

[33] T. P. Caudell, S. D. G. Smith, R. Escobedo, M. Anderson, "NIRS: Large scale ART-1 neural architectures for engineering design retrieval", *Neural Networks 7*, 1994, 9, pp. 1339-1350.

[34] T. P. Caudell and D. S. Newman, "An adaptive resonance architecture to define normality and detect novelties in time series and databases", *Proceedings of the INNS World Cong. on Neural Networks*, Portland 4, 1993, pp. 166-176.

[35] S. Grossberg and L. Wyse, "Invariant Recognition of Cluttered Scenes by a Self-Organizing ART Architecture: Figure-Ground Separation", *Proc. of the*

Int'l Joint Conf. for Neural Networks, Seattle, 1, 1991, pp. 633-638.

[36] G. Carpenter, S. Grossberg, and C. Mehanian, "Invariant Recognition of Cluttered Scenes by a Self-Organizing ART Architecture: CORT-X Boundary Segmentation", *Neural Networks 2*, 1989, pp. 169-181.

[37] G. Carpenter, S. Grossberg, and G. Lesher, "A What-and-Where Neural Network for Invariant Image Preprocessing", *Proc. of the Int'l. Joint Conf. for Neural Networks,* Baltimore, 3, 1992, pp. 303-308.

[38] M. J. Healy and T. P. Caudell, "An efficient neural network architecture for recognition of spatial pattern invariants", *Int'l. Joint Conf. on Neural Networks,* Baltimore, 4, 1992, pp. 208-213.

[39] E. R. Kandel, J. H. Schwartz, and T. M. Jessel, *Principles of Neural Science*, 3rd Ed., Elsevier, N. Y., 1991, Chapt. 29, p 425.

[40] M. J. Healy, T. P. Caudell, and S. D. G. Smith, "A neural architecture for pattern sequence verification through inferencing", *IEEE Trans. on Neural Networks 4,* 1993, 1, pp. 9-20.

[41] G. Carpenter, S. Grossberg, and J. Reynolds, "ARTMAP: Supervised Real-Time Learning and Classification of Nonstationary Data by a Self-Organizing Neural Network", *Neural Networks 4*, 1991, pp. 565-588.

[42] V. Braitenberg, *Vehicles, experiments in synthetic psychology*, MIT Press, Cambridge, Mass, 1987.

[43] W. Scheifler and J. Gettys, "The X Window System", *ACM Trans. on Graphics,* 5, 1986, 2, pp. 70-109.

[44] H. M. Abdel-Wahab and M. A. Feit, "XTV: a framework for sharing X Window clients in remote synchronous collaboration", *Proc. of the IEEE Conf. on Comm Software*, Chapel Hill, 1991, pp. 159-167.

BIOGRAPHY

Dr. Thomas P. Caudell **{{photo 1}}** is an Associate Professor of Electrical and Computer Engineering at the University of New Mexico, and a member of the newley formed University of New Mexico Interdisciplinary Computational Systems (ICS) Group. His current research is in virtual reality and artificial neural networks. He is studying the use of virtual reality technology for engineering, manufacturing, and construction applications in industry, and as an environment for conducting empirical studies on neurobiologically-based autonomous attentive perception systems and mobile robots.

Prior to moving to New Mexico he was Senior Principal Scientist and Principal Investigator of the Boeing Computer Services' Adaptive Neural Systems Research and Development project, which has developed and deployed several very large multiuser neural networks applications within the Boeing Company. He was also involved in the initial efforts to import virtual reality technology into The Boeing Company. Previously, he held the position of Senior Staff Physicist at the Hughes Research Laboratories in Malibu, where he lead their neural networks project and started their first virtual reality project. Dr. Caudell has been professionally active in the field of virtual reality and neural networks since 1986 and has over 75 publications in these areas. He received his Ph.D. in Physics from the University of Arizona in 1980.