

# VISUAL PERCEPTION PROCESSING FOR THE FLIGHT TELEROBOTIC SERVICER CONTROL SYSTEM

*Marilyn Nashman and Karen Chaconas*  
*National Institute of Standards and Technology*  
*Building 220, Room B127*  
*Gaithersburg, Maryland 20899*

## I. ABSTRACT

This paper describes the visual perception system of the NASA/NIST Standard Reference Model (NASREM) Architecture for the Flight Telerobotic Servicer Control System. It describes the hierarchical organization of the system, the interfaces, and the functionality of the visual perception branch of the real-time control system. It includes a description of the scope of the processing performed at the lowest level of perceptual processing, and it defines the interfaces and the information exchanged between the modules at this level, as well as interfaces to a camera, a human operator, and to higher levels of the system.

## II. INTRODUCTION

The control system architecture discussed in [2] describes a hierarchical framework to control complex robot systems. It has been partially implemented in various versions of the Real-time Control System (RCS) at the National Institute of Standards and Technology (NIST) including the Automated Manufacturing Research Facility (AMRF), the Army Field Material-handling Robot (FMR), the NBS/DARPA Multiple Autonomous Undersea Vehicle project (MAUV), and the Army TEAM (Technology Enhancement for Autonomous Vehicles) project. The hierarchy decomposes plans spatially and temporally to meet system objectives, monitors the environment with system sensors, and maintains the status of system variables.

The control system is composed of three parallel systems, task decomposition, world modelling, and sensory processing, that cooperate to perform telerobot control (Figure 1). The task decomposition system breaks objectives into simpler subtasks to control physical devices. The world model supplies information and analyzes data using support modules. It also maintains an internal model of the state of the environment in the global data system. The sensory processing system monitors and analyzes sensory information from multiple sources in order to recognize objects, detect events and filter and integrate information. The world model uses this information to maintain the system's best estimate of the past, current, and possible future states of the world.

Each device or sensor of the telerobot has support processes in each of the three columns of the control system, as shown in Figure 2. For example, the task decomposition functions associated with planning the actions for processing camera data reside in the task decomposition hierarchy; the world modeling functions for supporting those plans reside in the world model hierarchy, and the image processing techniques required for executing those plans reside in the sensory processing hierarchy. The modules can be logically configured according to their function in the system, as shown in Figure 3. The system pictured consists of two main branches; the left branch contains the perception processes and the right branch contains the manipulation processes. The perception branch of the tree supports processes which provide sensory feedback to the manipulator system such as cameras, range sensors, tactile array sensors, acoustic devices, etc. [4]. The manipulator branch of the tree supports processes which are responsible for planning and executing manipulator trajectories [5,9]. The two branches decompose tasks independently and communicate via the global data system.

The world modeling support modules communicate asynchronously with the task decomposition and sensory processing systems [7]. Information flows bidirectionally between adjacent levels within any given hierarchy. The interfaces to the sensory processing system allow it to operate in a combination of bottom-up (data driven) and top-down (model driven) modes. Bottom-up processing involves the extraction of knowledge from sensory data, and top-down processing is used to correlate predicted information from the world model with extracted information from the environment. The interfaces between the sensory processing system and the world model allow updated information to be sent to the world model and predicted information or sensory processing parameters to be sent to the sensory processing system.

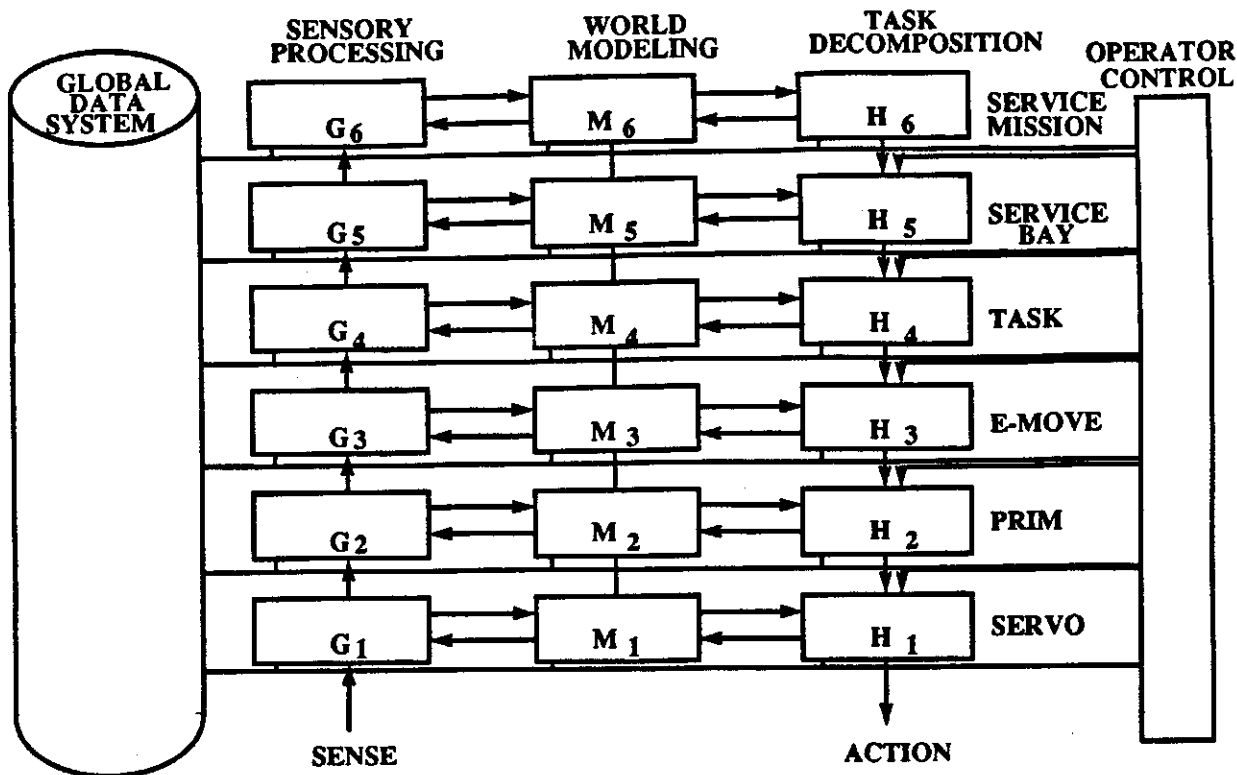


Figure 1. The NASREM Architecture

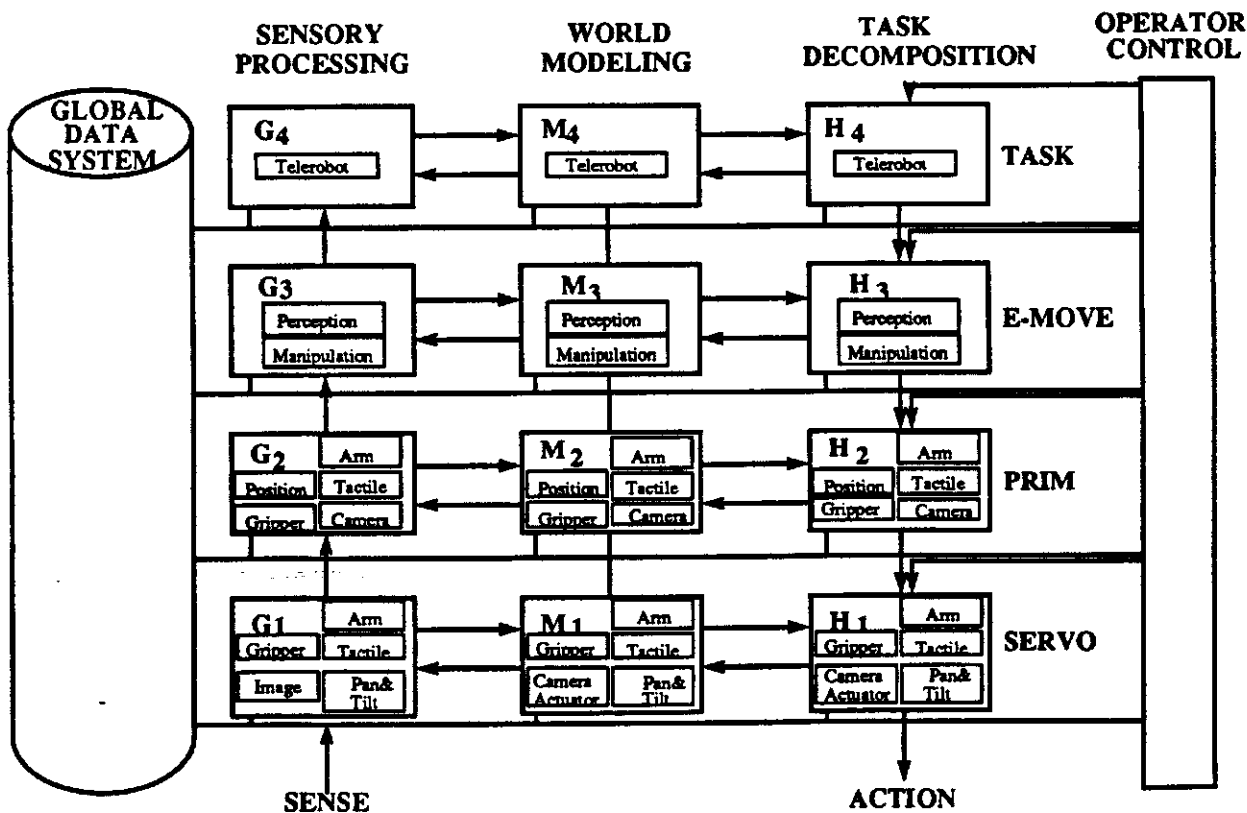


Figure 2. The NASREM Architecture for Control of a Telerobot

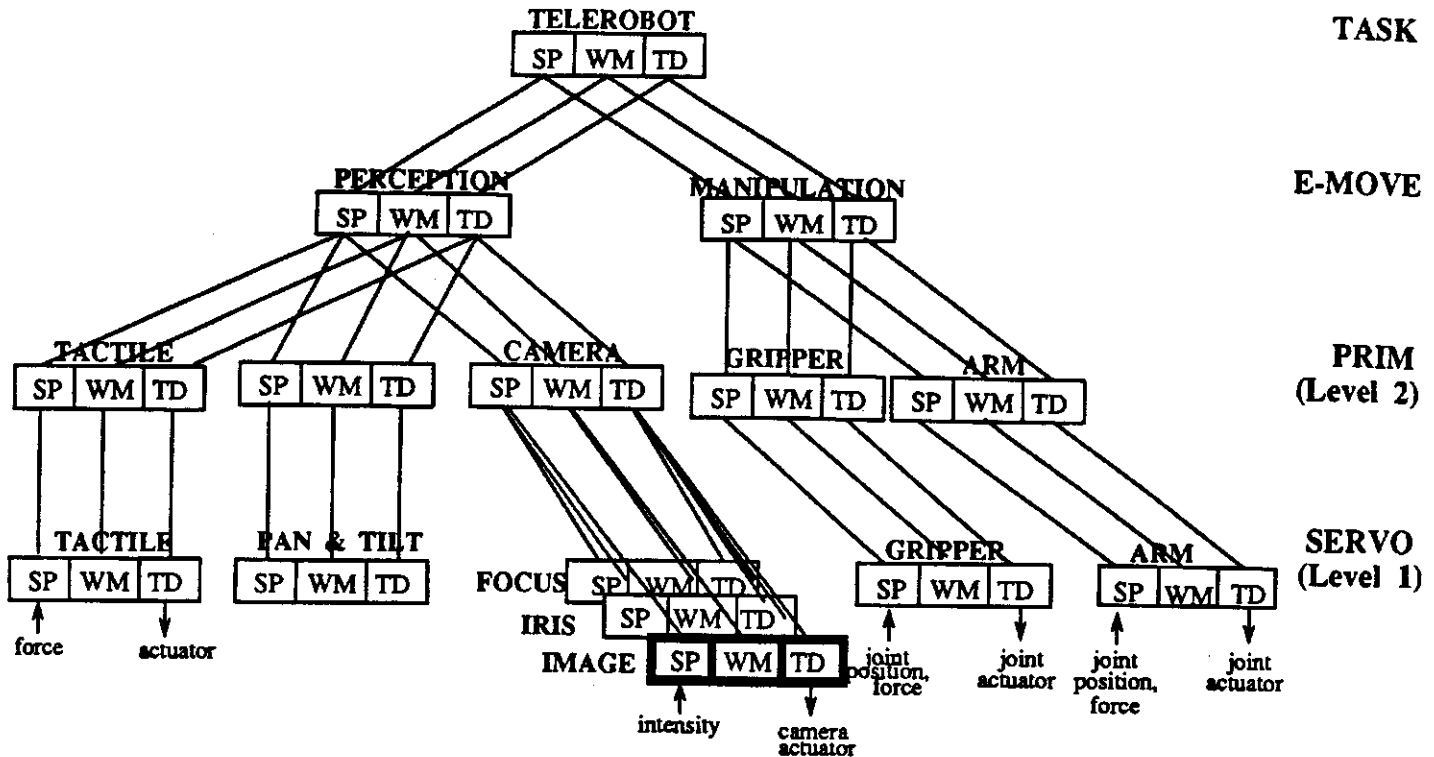


Figure 3. Functionality of the NASREM Architecture for Control of a Telerobot

This document describes the interfaces and functionality of the perception branch for a camera that is part of a telerobotic control system. Level 1 processing (the one highlighted in Figure 3) is used to explain the detailed interfaces. Processing at this level is performed on individual image elements or pixels. Level 1 gathers raw information (readings) from each camera, filters the information, and, when applicable, enhances it. It then extracts edge points, surface patches, and information relevant to the optical flow of pixels. Section III discusses the general architecture of a computational level of the system and defines the functions and the interfaces of the task decomposition, world model, and sensory processing modules. Section IV provides an example of the interactions between modules in performing a typical telerobotic task.

### III. GENERAL PERCEPTION SYSTEM ARCHITECTURE

Each level within a branch of the Telerobot Control System consists of a task decomposition module, a world model support module, and a sensory processing module (Figure 4). The task decomposition module bases its decisions on information extracted by the sensory processing module. The sensory processing module is driven by predictions of the state of the world provided by the world model. The world model maintains the best estimate of the past, current and possible future states of the world [1]. The characteristics of these modules for the perception branch are described below.

#### A. Task Decomposition

As shown in Figure 3, the Telerobot Control System contains four levels of control: task, e-move, prim and servo. Task decomposition consists of a Job Assignment (JA) module, Planner (PL) modules, and Execution (EX) modules. These modules have the same general functions at each level of the system. The Job Assignment module accepts and queues commands from the world modeling support module, the level above, and the operator. The commands are passed to the Planner modules, which analyze the request and select the most appropriate sensory processing algorithm for achieving the desired output. The Execution modules obtain confidence factors from the world model, update and modify algorithm parameters, and pass this information through the world model to the sensory processing system. The interaction of the Execution modules and the world model serves as a learning tool for sensory processing to improve the performance of algorithms.

Each of the three modules execute cyclically to process commands and pass information. They read input, perform computations, and generate output independent of the other modules. This type of processing allows the system to operate quickly and efficiently. It also allows the system to respond to new information without being explicitly commanded to do so.

The Planner and Execution modules are directed by one Job Assignment module. The single Job Assignment module interacts with a Planner module for camera control and  $s$  Planner modules, where  $s$  is the number of classes of processing algorithms at a given level of the system. At Level 1, there are five classes of algorithms: filtering, enhancing, edge point extraction, surface patch extraction, and optical flow. Each of the Planner modules communicates with an Execution module as shown in Figure 4. Figure 5 lists the information requirements of Level 1 modules from the world model, from Level 2, and from a human operator.

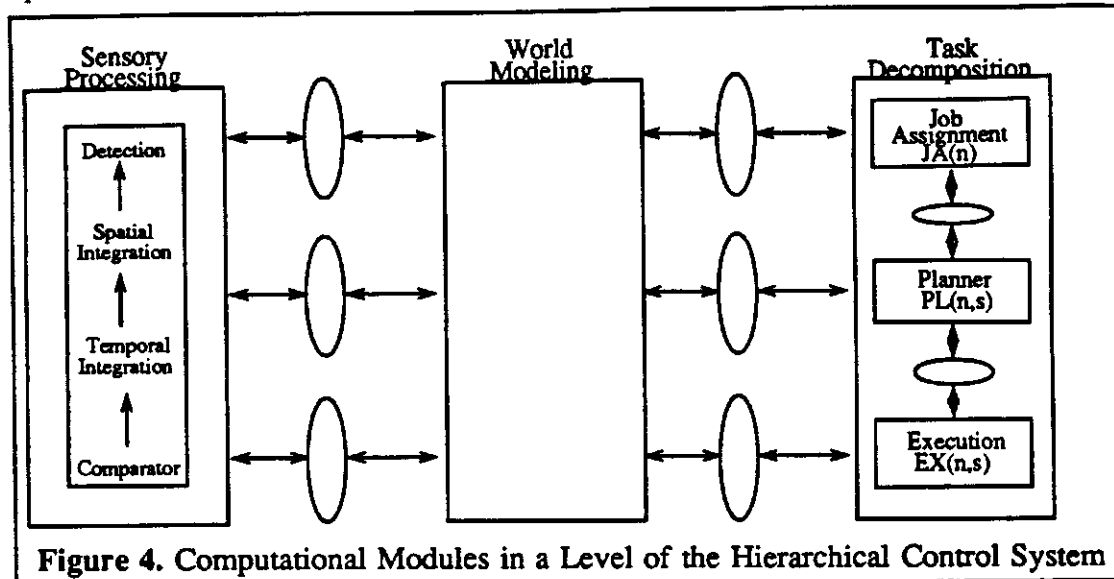


Figure 4. Computational Modules in a Level of the Hierarchical Control System

### A.1 Job Assignment Module

Within a computational level, the Job Assignment module maintains a queue of commands received from the world model, the next higher level and the operator. It accepts all incoming commands and assigns them a position in the queue according to the priority level assigned to the command. The priority level is based on the requirements of the plan developed by the Planner at the next higher level. For example, when task decomposition requires information about an object's position, it activates a plan to detect the identifying features of that object and to update their positions. The activation of a plan raises the priority level assigned to the class of algorithms responsible for extracting the required information.

The use of a queue enables incoming commands to be prioritized as they are received. In this way, the information needed immediately is serviced first, but all information buffers are updated at specified time intervals. At the completion of execution, the Job Assignment module returns status to the requesting process.

At each level of the hierarchy, the operator interfaces only with the Job Assignment module. He/she may request a specific type of output, output from a particular algorithm, or termination of execution of an active process. He/she may also request a change of parameters for a specific algorithm or request processing in a special window of interest. The Job Assignment module writes parameters supplied by the operator into the world model global memory where they can be read by the Execution module at any level. The operator also specifies the mode of operation for each command he/she issues: either continuous operation until a "halt processing" command is received or execution for a fixed number of times. In all cases, an operator request is assigned the highest priority. Output from an operator's command is returned in the form of graphic displays, ASCII strings, or other easily understandable formats.

### A.2. Planner Modules

The Planner modules read commands from the top of the Job Assignment queue. A single Planner is dedicated to camera control, e.g. iris or focus settings. The remaining Planners process commands to initiate sensory processing algorithms. The camera control Planner interprets and passes activation commands to the Execution module. Each remaining Planner determines which algorithm within its general class of algorithms capable of being performed in the sensory processing module at the given level is best suited for providing results. Since each class of algorithms contains many methods of computing the required output, the Planner modules act as rule based systems to choose the most appropriate algorithm for a given situation. Decisions

are based on criteria such as timing requirements, precision requirements, statistical analysis of sensed information, and knowledge about the environment (lighting conditions, power constraints, etc.). The world model global memory contains this information, and the Planner module reads and analyzes it. At completion of the command, it returns status information to the Job Assignment module.

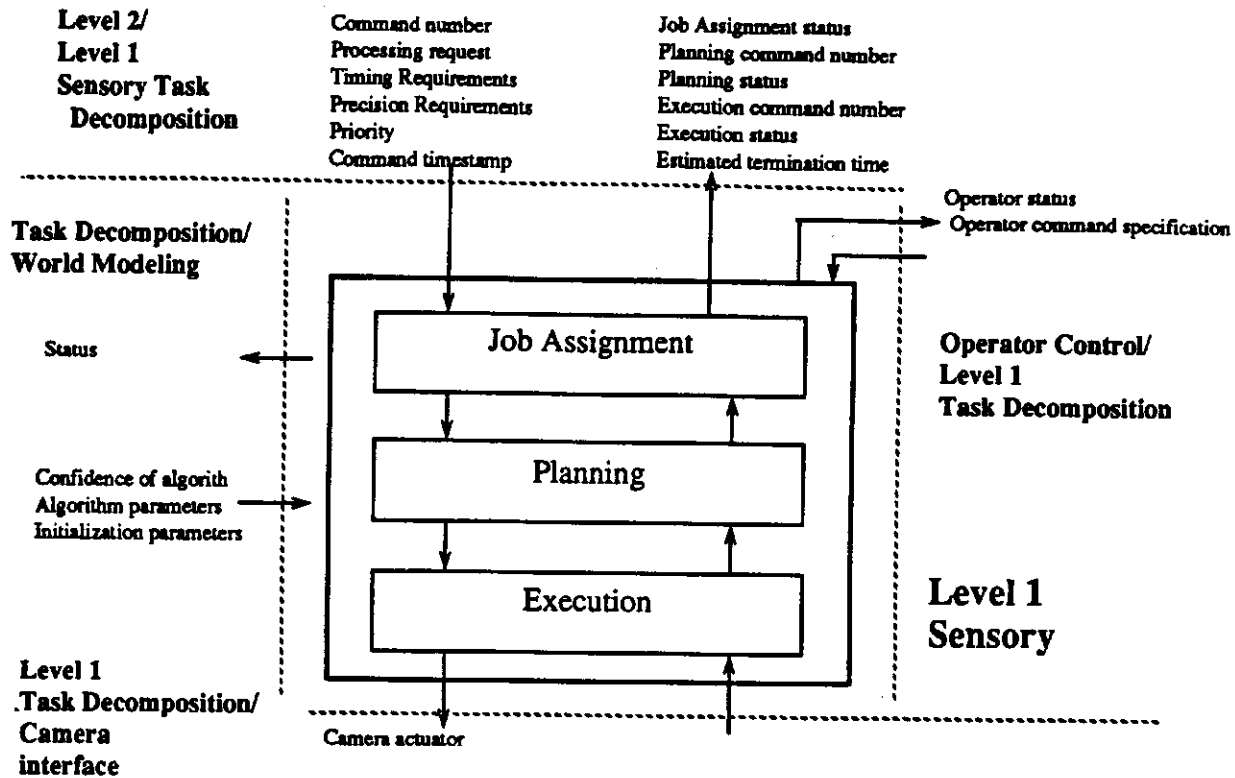


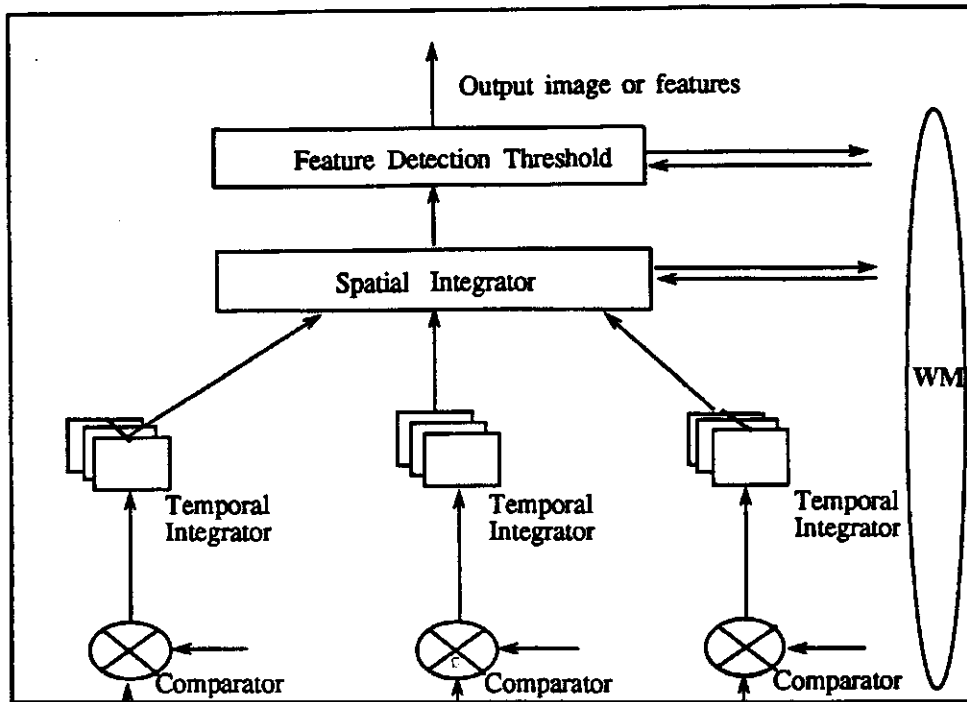
Figure 5. Level 1 Task Decomposition Module Interfaces

### A.3. Execution Modules

The Execution modules receive their commands from the Planner modules in the same level of the hierarchy. They are responsible for issuing commands to control a physical device or sensor or passing algorithm parameters to sensory processing and activating the sensory processing system to execute the selected algorithm. When a particular algorithm is chosen for execution, the appropriate Execution module reads the parameters required for its execution from the world model global memory. These parameters include threshold values, histories of past performance, and sensor model information such as physical sensor parameters, initial conditions, etc. The Execution module then passes the algorithm command and all parameters needed for its execution to the world modeling module.

### B. Sensory Processing

The sensory processing modules of the real-time control system compare incoming data with predicted information, integrate sensory data over space and time, and determine the detection of an event. At each level of the hierarchy, this information is used to update the world model. Each sensory processing module consists of four components: comparators, temporal integrators, a spatial integrator, and a detection threshold (Figure 6). A specific example of how these modules interact at a given level is shown in Figure 7, where the



**Figure 6. Submodules in the Sensory Processing System**

sensory processing module at Level 1 is discussed.

Sensory processing at Level 1 accepts pixel brightness values as input and processes each pixel according to the algorithm chosen by the task decomposition Planner module. Each pixel is passed through the comparator module, the temporal integrators, the spatial integrator and the detection module. The pixels can be enhanced or filtered using any of the algorithms available for preprocessing pixels, or they can be categorized according to their gray level characteristics into edge pixels, surface patch pixels, or pixels of motion. To clarify the type of processing done at Level 1, Figure 7 depicts the functions of the sensory processing modules for labeling pixels as edge or non-edge points using the Sobel edge detection method.

In the comparator module, pixel brightness value input is received from the camera sensor. Conceptually, there is a dedicated comparator for each pixel in the image array. The prediction supplied by the world model, the 3 x 3 Sobel convolution mask, is a model of the feature to be tested at that pixel location. As shown in Figure 7, each input pixel in the image is multiplied by the appropriate element of the Sobel edge detector mask.

The output generated by the multiplication of the pixel's intensity value and its corresponding value in the Sobel mask is passed to the temporal integrators. The results from an averaged sequence of pixels at the same location in the image are gathered over a time span defined in the world model.

The temporally integrated pixels are passed to the spatial integrator. The size of the spatial window is defined in the world model. The pixels are summed over this window.

Lastly, the results of the spatio-temporal information are evaluated by comparing the results to a threshold parameter stored in the world model. Pixels which exceed this threshold value are labelled edge points, while those pixels falling below the threshold are labelled non-edge points.

The order of the integrator modules can be reconfigured a priori depending on the algorithm. It may be appropriate for a specific application to perform temporal integration after spatial integration, such as when tracking a centroid of a moving object, or it may be unnecessary to do either spatial or temporal integration.

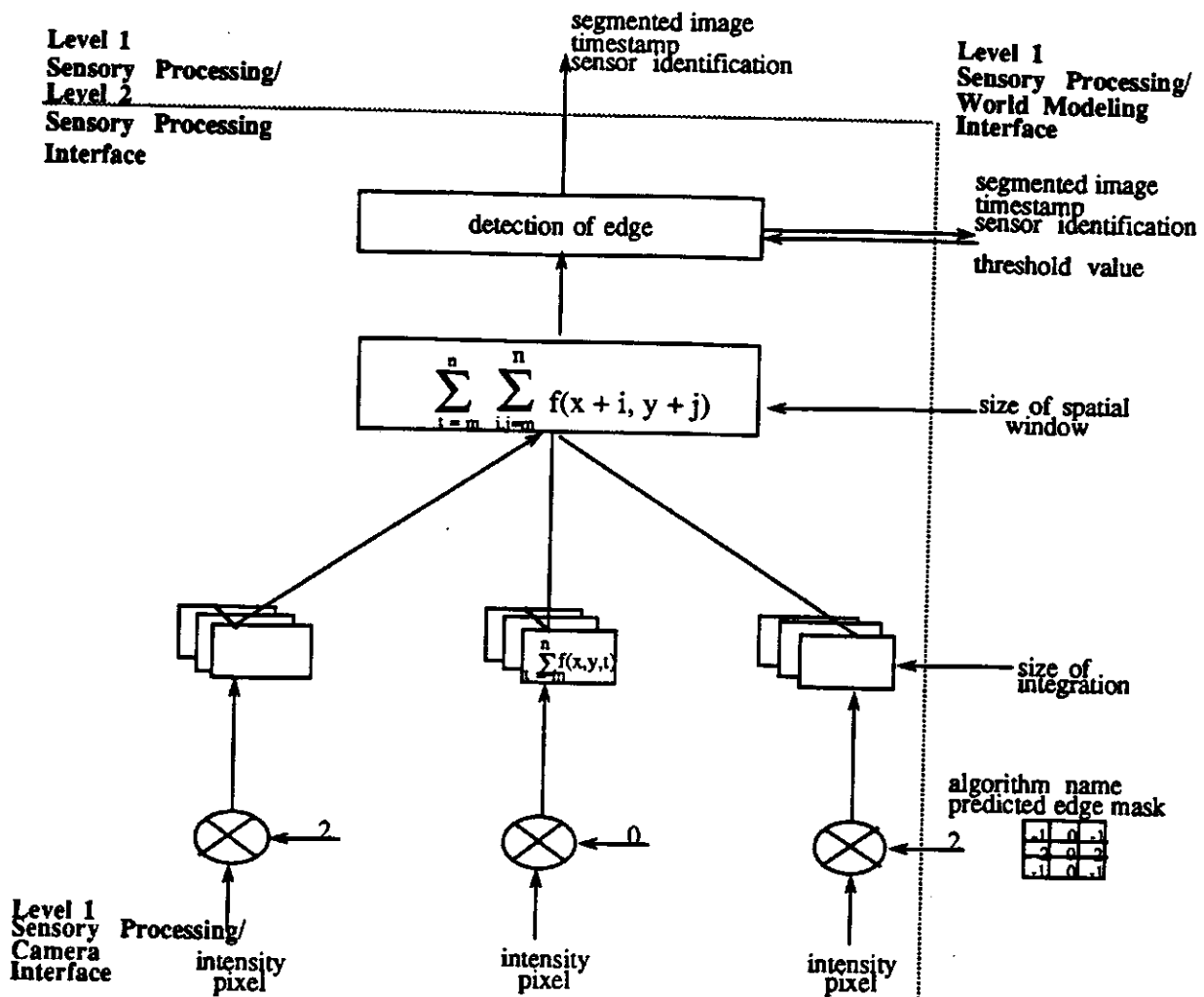


Figure 7. Sensory Processing for a Sobel Edge Detection Algorithm

### B.1. Comparator Module

The comparator modules receive input from two sources: the world model and the sensory processing module at the next lower level. The input from the world model is a model of the expected output. The input from the level below in the sensory processing hierarchy consists of the results generated by that level. The comparator modules perform algorithm specific computations using these two inputs to generate values which are passed either to the temporal integrators or the spatial integrator.

### B.2. Temporal Integrators

Each temporal integrator combines its inputs over a given time window. The length of the time interval is supplied by the world model and depends upon factors such as timing and accuracy requirements. In addition, the window usually covers a shorter interval at lower levels of the control hierarchy and a longer interval at higher levels. The output from the temporal integrators is passed to both the world model and to the spatial integrator.

### B.3. Spatial Integrator

The spatial integrator module integrates values over space to produce a single response value. The range

of the spatial integral is supplied by the world model, and the results of the spatial integration are sent to the model to update confidence factors.

#### **B.4. Detection Module**

The output from the spatio-temporal integration process is passed to the detection module for evaluation or event detection. When the output surpasses a prespecified threshold indicating correspondence between observations and the prediction of the world model, event detection occurs. An event can be defined to be the detection of an edge point, the fit of a line, or the recognition of an object, depending on the level in the control hierarchy at which the detection is occurring. The correspondence of a prediction occurs when, for example, a moving object's centroid is within a small distance from its prediction based on a past centroid measurement and the object's velocity. The results of event detection are passed to the world model to update global memory.

#### **C. World Modeling**

World modeling maintains the system's internal model of the world by continuously updating the model based upon sensory information. It consists of two components: support processes or functions which simultaneously and asynchronously support sensory processing and task decomposition and the global data system which is updated by the world modeling support processes. The term *world model* refers to the two hierarchies of support processes together with the global data system. Throughout this document, the terms world model, world model support, and global database will be used interchangeably. Any of these terms implies the combined function of the world modeling Level 1 support module and the global data system.

The world model provides decision-making criteria to the task decomposition system. It allows the Planner module to access global memory in order to select the optimal algorithm in a given situation. The Planner uses histories of performance, timing criteria, lighting conditions, expected range, etc. to choose an algorithm. This information is stored in the world model database. The Execution module selects the parameters or initialization conditions required for sensory processing. These parameters are also stored in the world model.

The world model global database provides sensory processing with the algorithm selected by the task decomposition Planner, the parameters selected by the Execution module, and any additional command parameters, such as integral ranges. The world model support module analyzes the selected algorithm in order to provide the model required by the sensory processing comparator. In addition to providing sensory processing with an algorithm and its parameters, the world model also provides a prediction to the detection module. The prediction is a range of acceptable values that are used to determine whether an event has been successfully detected. A threshold value used in edge detection or a window for the centroid value of a moving object are two examples of predictions. The results of the sensory processing integration and detection processes are written to the world model where they are used to update confidence factors and global memory.

### **IV. A VISION SYSTEM APPLICATION**

To clarify the operation of the computational triple at Level 1 for a camera, this section provides a specific example of system interfaces. Assume that Level 3 initiates a command for information required for tracking a particular surface of a moving part. Further, assume that task decomposition has selected a particular instance of a camera and positioned it appropriately. The world model contains information about the object model and an initial prediction of the location of the part. Level 2 is directed to extract the symbolic information required to define the features of the object surface and their attributes (the centroid of the object aspect, the equations of its boundaries, their length and their orientation). The input it requires for these computations resides in buffers written by Level 1. Similarly, Level 1 is directed to segment its data in accordance with priority levels set by its Planner.

The remainder of this section describes in detail the role of the Level 1 processing unit associated with the particular camera in this example. The sensor plan stored in the world model generates requests to the sensory processing module in response to the need for updated information. These requests are created by assigning priority levels to the classes of output produced by Level 1. For the sake of example, we assume that the plan requires an edge point image 20 times per second and a surface patch image of the same scene 10 times per second. When these commands are received by the Job Assignment module, they are prioritized and placed on the queue.



The boundary detection Planner module receives its commands from the top of the Job Assignment queue. In order to obtain an edge point image, it must decide which gradient extraction algorithm is most likely to provide satisfactory results in a particular situation. The Planner module determines the best algorithm based on a performance history residing in the world model. It also considers the update rate required by the plan. The execution time of each algorithm is known to the system and is stored in a world model parameter. In addition, the Planner module also must take into account the accuracy required. For example, when the camera is far from the object being tracked, interior texture information is not visible and does not have to be smoothed from the image. However, texture could be visible in a closer view of the same object and must be removed in order to avoid false edge information. Thus, different algorithms are used depending on the distance between the camera and the object, lighting conditions and object surface reflectivity.

The Planner module performs the same type of analysis in choosing an appropriate region classification routine. Assume that after analyzing all factors, it requests output from a series of two Sobel edge detection algorithms followed by a thresholded image for a period of thirty seconds. The Execution module reads any parameters that are required by the individual algorithms from the world model global memory area. In this example, the edge point algorithm needs a threshold value to suppress low magnitude edges, and the thresholding algorithm needs a parameter for converting grey scale pixels to binary pixels.

The Execution module passes the algorithm name and the associated parameters to the world model. This results in the sensory processing module activating the chosen algorithms from each class of algorithms: Sobel edge detection from the class of gradient extraction techniques and thresholding from the class of region classification techniques. These algorithms are cyclically executing processes, and are continuously reading raw camera data. The output from each algorithm is written into a predefined buffer area where it is available to be read by the requesting process. The sensory processing module appends a timestamp to the results based on the system time at which the algorithm was initiated. Status information is sent to the world model module at the completion of processing.

## V. CONCLUSION

This document has described Level 1 of the perception branch of a real-time control system hierarchy for the Flight Telerobotic Servicer. The components and functions of the computational triple of task decomposition, world modeling, and sensory processing were defined, and the specific functions of each component were discussed. Interfaces between the modules, including the operator, have been defined. The concept of grouping these algorithms into classes of algorithms allows the flexibility of adding or deleting algorithms at any stage of implementation without changing the structure of the system.

Although specific hardware requirements are not defined in this document, the amount of array information required to be processed at this level (~64 K bytes of information per image), suggests the use of parallel processing machines for realtime output. Many algorithms implemented at Level 1 operate on image data by using local information in a non-sequential manner. Because of the large amount of data to be processed and the need to process that data as close to video rate as possible, most serial computers cannot meet the requirements of Level 1 processing. Parallel computers have been developed in recent years to specifically fulfill the need of real-time processing of image data [3,6,8], and although the machines differ in architectural design and implementation, they share the goal of being able to process an entire image or a region of an image in real-time.

## ACKNOWLEDGMENTS

The authors would like to thank all of the members of the Robot Systems Division who have contributed to the design of the NASREM control system, especially Ron Lumia, John Fiala, Laura Kelmar, Stephe Leake, John Michaloski, Barry Warsaw, Al Wavering, and Tom Wheatley.

#### REFERENCES

- [1] J. S. Albus, *Brains, Behavior, & Robotics*, Petersborough, NJ, Byte Books, Subsidiary of McGraw-Hill, 1981.
- [2] J. S. Albus, H. G. McCain, R. Lumia, *NASA/NBS Standard Reference Model Telerobot Control System Architecture (NASREM)*, NASA Document SS-GSFC-0027, NASA GSFC, Greenbelt, MD, July, 1987.
- [3] Aspex, Inc., *PIPE--An Introduction to the PIPE System*, NY, 1987.
- [4] K. Chaconas, M. Nashman, *Visual Perception Processing in a Hierarchical Control System: Level 1*, NIST Technical Note 1260, National Institute of Standards and Technology, Gaithersburg, MD, June 1989.
- [5] J. Fiala, *Manipulator Servo Level Task Decomposition*, NIST Technical Note 1255, National Institute of Standards and Technology, Gaithersburg, MD, October 1988.
- [6] T. Gross, H. T. Kung, M. Lam, and J. Webb, "WARP as a Machine for Low Level Vision.", *IEEE Journal of Robotics and Automation*, pp. 790-800, 1985.
- [7] L. Kelmar, *World Model Level 1*, NIST Technical Note 1258, National Institute of Standards and Technology, Gaithersburg, MD, December 1988.
- [8] R. Lumia, M. Shneier, and E. Kent, "A Real-Time Iconic Image Processor", *IEEE International Conference on Robotics and Automation*, pp. 873-878, 1985.
- [9] A. Wavering, *Manipulator Primitive Level Task Decomposition*, NIST Technical Note 1256, National Institute of Standards and Technology, Gaithersburg, MD, October 1988.