COMPUTER VISION, GRAPHICS, AND IMAGE PROCESSING 33, 293–306 (1986)

Model-Based Strategies for High-Level Robot Vision*

MICHAEL O. SHNEIER, RONALD LUMIA, AND ERNEST W. KENT

National Bureau of Standards, Gaithersburg, Maryland 20899

Received September 18, 1984; revised December 31, 1985

The higher levels of a sensory system for a robot manipulator are described. The sensory system constructs and maintains a representation of the world in a form suitable for fast responses to questions posed by other robot subsystems. This is achieved by separating the sensing processes from the descriptive processes, allowing questions to be answered without waiting for the sensors to respond. Four groups of processes are described. Predictive processes (world modellers) are needed to set up initial expectations about the world and to generate predictions about sensor responses. Processes are also needed to analyze the sensory input. They make use of the predictions in analyzing the world. A third essential function is matching, which involves comparing the sensed data with the expectations, and provides errors that help to servo the models to the world. Finally, the descriptive process constructs and maintains the internal representation of the world. It constructs the representation from the sensed information and the expectations, and contains at all times everything known about the world. The sensory system is responsive to changes in the world, but can also deal with interruptions in sensing, and can supply information that may not be available by sensing the world directly.

1. INTRODUCTION

The goal of a robot sensory system is to understand the environment sufficiently to enable the robot to accomplish its assigned tasks. The processes involved in understanding must operate fast enough so that the robot control system, which moves the robot, does not have to pause in performing its tasks to wait for sensory information. Usually, however, the information supplied does not have to be completely accurate. For example, the position of an object can be given roughly when the robot is far away from the object, and can be refined as the robot approaches the object.

Other factors that characterize the robot sensing problem and differentiate it from the general vision problem are more task-specific. A robot operating in a factory has a much more constrained environment than a robot moving through open countryside. Both robots need models of the objects they will encounter, but, for the robot in the factory, the models can usually be much more complete and specific than those for an autonomously navigating robot. The work outlined in this paper is aimed at the problems of a robot working in a factory environment. The approach is, however, applicable to more general situations, if various modifications are made.

Consider a robot engaged in servicing a machine tool. Many features in this environment are fixed: the machine tool does not change its position, and the worktable usually is either stationary or moves with a known velocity. For well-defined tasks, the appearances of parts to be manipulated are known in advance, and their changes in appearance throughout the machining process are also known. This knowledge provides powerful constraints that can be used to enhance the perfor-

0734-189X/86 $3.00

mance of the sensory system. It is also possible, however, for unknown objects to appear in the workspace, and a mechanism must be provided to incorporate such objects into the evolving understanding of the environment.

The above scenario describes the situation for which the sensory system discussed below was designed. The sensory system is one part of a comprehensive automated factory project whose goal is the complete automation of a small-batch metal-machining facility (Simpson *et al.* [15]). This paper discusses only the higher levels of the sensory system. The lower levels, as currently implemented, are described elsewhere (Shneier [13]; Kent [6]).

Information available to the sensory system has two sources. The first is a set of sensors that examines the environment, and the second is the a priori information about the world and the objects in it. While the a priori knowledge base may be very large, only a small part of it will be needed for any particular task. This is made available to the sensory system when the task is initiated. A certain amount of processing may be performed on the information before the task begins. This processing need not occur in real time. Once the task begins, however, all responses from the sensory system must occur in real time.

The initial processing involves acquiring models of the expected objects from a database, setting up initial expectations about the world, and configuring the sensory system to identify the expected objects. The initial expectations represent the best guess about what is in the world. As sensory data are acquired, the best guess is constantly updated to reflect the real situation. At all times, the sensory system attempts to maintain its internal representation in registration with the world. The sensory system runs asynchronously with respect to the control system, which is permitted to ask questions of the sensory system at any time. The control system of the robot is informed about the world based on information in this best-guess workspace representation. This removes the need for pauses while the sensors directly sample the world to respond to control system queries.

Thus, two parallel processes are needed to accomplish a task. First, there is the control process that plans and executes the steps of the task. Second, there is the sensory system that examines the world and maintains the workspace representation using information about the goals of the task and the responses from the sensors. Both of these processes are organized hierarchically. The task is recursively broken down into subtasks, until primitive subtasks that can be executed directly are reached. Similarly, the sensory processing is divided into simpler and simpler tasks. Here, however, the division is along semantic lines. The high levels of sensory processing concern themselves with recognizing objects and relationships in the environment. Lower levels extract and group features, while yet lower levels perform generic sensory-processing tasks.

The rest of this paper describes the representations and the processes used for the higher levels of the sensory system. The next section provides an overview of the system. It is followed by a brief description of the sequence of events during the execution of a task, and each module is then described in more detail.

2. OVERVIEW OF THE SENSORY SYSTEM

The sensory system consists of a number of components, each of which has a specific role in the processing. Components communicate with other components and cooperate to construct and maintain the internal model of the workspace. The

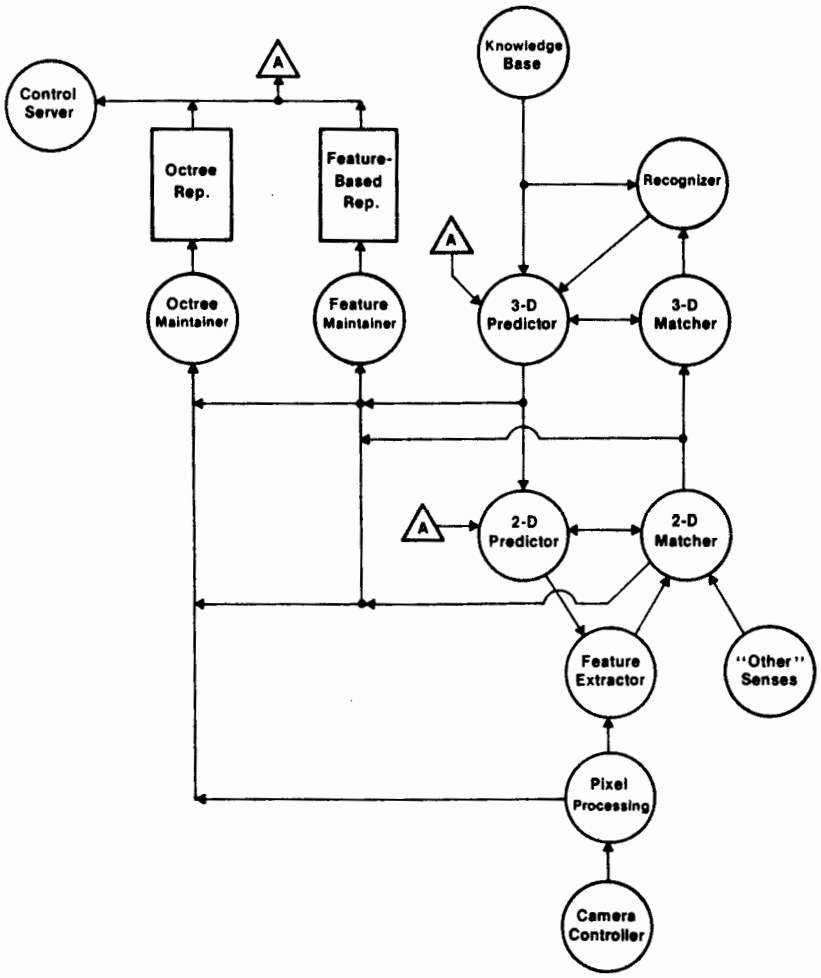

FIG. 1. A schematic representation of the higher levels of the sensory system.

sensory system modules must predict the information the sensors will detect, analyze the sensor data to produce meaningful features, match the sensor data with models, and continuously update the model of the world. Figure 1 shows the modules and their interconnections.

The knowledge base is external to the sensory system, and stores information about all objects that have been entered into the computer-aided design system. For any specific task, only a small subset of these objects will be active. The knowledge base also stores information about fixed objects in the workspace, such as machine tools, buffer areas, and the robots themselves.

There are two levels of world modelling in the system. The higher level modeller maintains three-dimensional representations of the objects in the workspace, and uses them to predict the three-dimensional structure of the scene at any time. The lower level modeller produces two-dimensional projections of the objects, and uses

them to predict the two-dimensional image features expected at each cycle. It is possible to consider yet higher levels of modelling, such as models of relationships between objects, but this has so far not been investigated.

Images and other sensor data are preprocessed by the low-level system, not described here, in order to extract a set of meaningful features. The results of processing the data are stored first in a two-dimensional database, and, after suitable processing, in a three-dimensional database. The two-dimensional database has an associated processor that reconciles data from multiple sensors, and attempts to match the real two-dimensional data with the expectations. When it is successful, it passes the labeled features to the three-dimensional database. This database also has an associated processor, which is used to servo the three-dimensional models to the sensory data. This results in updated poses for the objects, which are sent back to the three-dimensional world modeller to improve its future hypotheses.

When objects or groups of features cannot be matched with the expectations, they are sent to the recognition module. The task of this module is to attempt to identify subsets of features with generic objects, using the full three-dimensional models stored in the world model. This involves a complete graph search, and is much slower than the verification matching in the usual situation. When an object is recognized, it is instantiated in the world model, and appears in all subsequent hypotheses.

The system has to deal with both expected and unexpected objects, and cannot wait until an object has been recognized before interacting with it. This means that unrecognized objects must be represented in the same way as recognized objects, at least to the extent necessary for trajectory computations, or for closer examination. The workspace representation fulfills this need. It consists of two parts, a spatial representation and a feature-based representation. The spatial representation records the parts of the workspace that are empty, full, or about which nothing is known. The feature-based representation records information about instances of objects and features. Links exist between the two representations and between the feature-based representation and the world models.

The next section describes the sequence of events in the various levels of the sensory system.

3. SEQUENCE OF EVENTS

The first stage in the execution of a task involves initializing the sensory system. The knowledge base sends description of the objects expected to be found during the task to the three-dimensional world modeller. It also sends information about other (non-task-specific) structures in the world, and expectations about the initial configuration of the workspace.

The three-dimensional world modeller distributes this information to the spatial representation module which creates the initial map of the workspace. The information is also sent to the three-dimensional matcher, which uses it as a starting point for its verification of the scene, and to the feature-based representation, which sets up its initial best guesses about the objects and features in the workspace.

Processing during the task is an iterated procedure whose goal is to maintain the workspace representation in registration with the world. Sensors that move about in

the workspace and supply information about the parts that are visible slowly build up a comprehensive picture of what is really in the world, and where all the objects really are. Objects can move or be moved by the manipulator, and this, too, must be taken into account. Deciding which sensors to use and how best to use them to maintain the workspace representation is the task of a supervisor processor.

At the start of a cycle, the supervisor decides what kind of sensors to use. In the case of vision, for example, it may choose between a floodlight or structured light sensor. The supervisor commands the lower-level system to take the picture, and the low levels perform a standard thresholding and connected-components analysis for floodlit images, or a less-standard approximation using Chebyshev polynomials, in the case of structured-light images (Shneier *et al.* [14]).

In the meantime, the two-dimensional world modeller uses information about the expected position of the camera, and about the positions of objects expected to appear in the scene to construct windows in the image, and features to be sought in each window. The sizes of the windows reflect the uncertainties in the positions of the features. Each expected feature is labeled using information from the model. The two-dimensional world modeller also tracks the features in the image over time. The tracker maintains windows in the image where each feature can be expected, given its appearance in previous views and the expected motion of the robot. The process of tracking feature locations aids the feature labelling process done in the two-dimensional database and plays a vital role in the treatment of unexpected features.

When the processed image arrives at the processor responsible for extracting features, the windows are placed over the image, and the hypothesized features are extracted if they appear. The results of the processing are sent to the two-dimensional data base, where they are reconciled with information from other sensors.

At the two-dimensional database level, the two-dimensional matching process is carried out. This is greatly simplified because the processing of features was already model driven. Where features were hypothesized, and the hypothesis was confirmed by the feature detector, the labeling is straightforward. The matched and unmatched features are sent to the three-dimensional matcher, which uses the labelled features and its three-dimensional expectations to match objects. The results are new estimates of the poses of the objects,which are sent to the three-dimensional world modeller for use in the next iteration of hypotheses. Features that cannot be matched are sent to the recognition module.

The recognition module takes the unmatched features and performs a search through all the object models in an attempt at recognition. On success, the results are sent to the three-dimensional world modeller, which then includes the objects in later computations. Even without recognition, however, the features will be transformed by the known motion parameters when their two-dimensional projections are predicted in later views.

In parallel with the matching operations used for naming objects, a separate process attempts to describe the environment in terms of the space that is occupied. This is done using an octree representation to decompose the volume of the work area. Each camera image is projected into the octree. The objects are intersected with objects already represented, while the background is projected to carve out volumes known to be empty. There are links between the spatial representation and a feature-based representation. They allow object names and attributes to be associated with regions in space.

The whole process of hypothesis and test, and of describing the world, is repeated throughout the duration of the task. As more and more parts of the world are sensed, the workspace representations become more and more accurate, and the expectations converge towards the true situation. The various processes that carry out the computations are described in more detail in the next section.

4. PROCESSES

The processes can be divided into four groups. There are predictive processes (world modellers), processes that analyze sensor data, processes that match sensor data with models, and processes that use the sensed data and the predictions to describe the world.

4.1. Prediction

There are two predictive processes in the system. One predicts three-dimensional poses of objects, while the other predicts how the sensors will respond to the world in the next cycle. Initially, the three-dimensional modeller receives its input from the external database. Input consists of geometric descriptions, or models, of parts expected to appear in the environment during the task, and initial expectations of their positions. Also passed to the modeller are descriptions of fixed surfaces in the workspace, such as buffer tables and machine tools.

The three-dimensional modeller acts as a repository of information about individual objects. For each instance of each object, it constructs a pose matrix describing the position and orientation of the object in the world. There is, however, only one copy of the generic model for each object. The three-dimensional world modeller is only responsible for predicting the three-dimensional nature of the world. The workspace representation attempts to *maintain* an explicit representation of what is actually in the world in a form more amenable for responding to questions about the world.

The three-dimensional modeller predicts how the world will appear at the next cycle of sensing. Its predictions are used in matching and describing the world. It also acts as the foundation for the two-dimensional matching process. The two-dimensional matching process predicts how each individual sensor will perceive the world when it next takes a sample. For a touch sensor, the prediction is a simple binary response, either touching or not touching. For a visual sensor, it is an image of projected features with their locations. It is not necessary, however, to construct a whole image, because typically only small regions in the image are important.

The predictions are computed from the three-dimensional models, and the predicted positions of the sensors. They also make use of the workspace representation. In the case of vision, the image to be constructed must have hidden features removed, and must include both recognized and unrecognized objects. The image is constructed by projecting the field of view of the sensor into the octree and finding the set of objects that will appear. The hidden feature relationships are derived partly from an aspect graph, or set of generic two-dimensional views of the object (derived off-line from the models), and partly from the relationships between objects in the octree. The aspect graphs (Koenderink and van Doorn, [9]) give information about the appearance of each object from the expected sensor position, and the octree provides information about the relationships between objects in the scene.

This information allows the predicted image to be computed. Only certain parts of the image are, in fact, constructed. These are the projections of features considered important for recognizing or verifying the objects' identities. For more detail on the three-dimensional world model see Lumia [10].

The two-dimensional modeller includes the ability to track features in images as the sensors (or the objects) move. This is particularly important for dealing with unexpected objects for which predictions are not available. The appearances and positions of features are predicted based on their previous history, and the known motion of the sensors. By identifying features across images, it is possible to extract three-dimensional information about them, which will help the three-dimensional modeller and the recognizer to interpret the environment, and allow more accurate predictions for future images.

The two-dimensional modeller thus acts as a source of predictions of what the sensor should attempt to extract from its input, and where in the input it should look for information. In the case of vision, for example, it might predict corner features, and would be able to define a window in the image within which each corner could be expected to be found. The size of the window is related to the confidence of the prediction. The predictions are sent to the processors for each sensor, and used to guide their analysis.

4.2. Analysis

The processes that analyze the sensory input are usually sensor specific. Each sensor produces data that must be modified or sampled to produce useful information. Ultimately, however, the information from the various sensors must be integrated and reconciled into a single description. After this has been done, there is usually no need to know how the information was obtained, and questions about the world can be answered without reference to the means by which the answers were obtained.

In the case of a visual sensor, low-level operations are those that are image-independent. For example, smoothing and edge-detection operators are usually applied to all images, regardless of the information to be extracted. High-level operations start at the point at which model-based information first interacts with the image. In the system described here, this occurs at the stage of extracting features from the image. It should, however, be mentioned that it is possible to propagate model-derived information all the way down to the lowest-level operations. A hardware image-processing machine has been designed to accomplish this, and is described elsewhere (Kent *et al.* [7]). The machine is currently under construction, and is expected to replace the low-level processing at a later date.

In the current system, low levels of processing extract connected components from images, and compute various properties of each component. The components are then sent to the first level that makes use of model knowledge. This is the level at which features are extracted. The modelling process predicts which features should be visible in the image, where they should occur, and provides a confidence measure about the existence and location of each feature. The image is processed in the windows where features are expected, and the feature detectors are tailored to the expectations. For example, if a corner is expected with a particular included angle, it can be sought explicitly, and lower thresholds can be set for accepting a corner with the right characteristics. This can, of course, lead to hallucination of features where

they do not actually appear, but in a non-malevolent world this should happen only rarely.

The parts of the image outside the windows of expectation must also be processed. The windows were chosen to detect the features of the expected objects. It is not reasonable to suppose that the features for unexpected objects will also fall within these windows. Consequently, these non-windowed areas must also be checked. Furthermore, if an expected feature does not appear within its window, either the expectation is incorrect due to lack of knowledge about the world, or the object detected is different from the expected object. If this occurs, the system applies all its feature detectors to the component, in the hope that the recognition module will be able to name the object on the basis of its features. Other processing that is done for each component is to approximate its boundary using straight line segments. This is to aid the description processing in building the workspace representation.

4.3. Matching

Matching can be divided into two processes: verification and recognition. Verification involves comparing expectations with observations, and computing the degree of similarity. Recognition is more difficult, because the expectations are much less specific. Recognizing an object involves matching a set of features with a set of models and computing the best fit. In verification, usually only a single model need be matched, and the orientation and visible features are already hypothesized. As a result, verification is a simpler and faster process than recognition.

The aim of verification is to confirm the hypotheses of the modelling process, and to compute the differences between the expectations and the observations. These differences allow the world modellers to servo their expectations to the real world. Verification occurs at two levels. The first is at the two-dimensional level, corresponding to the two-dimensional modelling process, and the second is at the three-dimensional level, corresponding to the three-dimensional modelling process.

Two-dimensional verification matching is greatly simplified by the model-driven feature-extraction processes. When a feature hypothesized by the modeller is found, the labelling is already known. When more than one feature is found within the same window, or a feature is found that could have more than one label, an arbitration process is invoked to select the best choice. Missing features can reduce the confidence of a match, but many fewer features are necessary to successfully verify an expectation than to recognize an object.

The result of a two-dimensional match is a new feature with ideal properties, such as the included angle in a corner, being taken from the model, but with position and orientation being taken from the data. These labelled features are sent to the three-dimensional verifier, while the errors between the expected and actual positions are sent to the two-dimensional modeller to be used to update its expectations. Unmatched features are sent to the recognition module.

A similar process occurs at the three-dimensional verification level, using the ideal labeled features extracted in the two-dimensional level. Here, an iterative procedure is used to pull the model and the sensed data into congruence. The result is a new pose matrix for each object which describes its location. This is sent to the three-dimensional modeller to be used in constructing the next round of expectations. The algorithm used to perform the matching is described in detail in (Rutkowski *et al.*

[11]). It takes as input labelled features from the sensed data and from the model, and uses a least squares technique to find the best global match.

The recognition module has a much harder task to perform. Unlike the verification processes, however, it is does not have to operate in real time to find matches. This is because the spatial representation will have enough information to enable the system either to avoid an object or to study it more closely, and, because the object is unexpected, the system has to take special action to deal with it which is not part of its explicit task. Input to the recognition module consists of generic descriptions of all the objects, and features extracted by the sensors that were found not to belong to any expected object. Features are organized according to the components to which they belong. The recognition module performs an exhaustive search of all object models, attempting to find subsets of features that are arranged in the same way in the data as in a model. When a unique match is found, the features used to find it and any others that are consistent with the interpretation are passed to the feature-based representation, which reorganizes its features, and the discovered object is made known to the world modeller for use in later expectation generation.

The recognizer might not be able to match all features, either because an object appearing in the scene is not one for which there is a model, or because insufficient features are available to disambiguate the matches. As more sensory data are acquired, however, most objects will be recognized. Those for which no recognition is possible are kept in the system, but can only be described by the descriptive processes and transformed by the modellers. No non-sensed information can be brought to bear for such objects.

4.4. Description

The aim of the descriptive process is to construct and maintain an accurate and up-to-date representation of the workspace and its contents. The purpose of the representation is to act as a buffer between the control system and the sensors, so that questions about the environment can be answered immediately, without having to wait while the sensors analyze the workspace.

The workspace representation is divided into two parts, a spatial representation and a feature-based representation. Together, they describe what is known about the workspace, including objects that have been recognized and those that have not. The workspace representation is described in Shneier *et al.* [12].

The spatial representation is designed to allow answers to questions about empty and occupied regions. It is organized as an octree, a regular decomposition of a cube into octants. Each octant may be split again, if it is not homogeneous, giving rise to a tree representing the workspace. Homogeneous nodes in the tree, called leaves, represent empty space, where it is known that no objects lie, object volumes, where it is possible that an object exists, or unknown regions, where no sensory information is available.

The spatial representation is constructed directly from the visual sensory data. Each component of each image is projected into the octree as a cone centered at the focus of the camera, with a cross section defined by the silhouette of the component. As the camera moves about in the world, each object may be seen from many viewpoints. The intersections of the resulting cones constrain the shapes and locations of the objects. Similarly, the empty parts of each image project into the workspace, and give rise to regions known to be empty. The octree construction

process is described in Hong and Shneier [5]. Other sensors, such as proximity or structured light sensors, can also supply information to constrain the positions of objects. Each object node in the octree contains pointers to the feature-based representation, described below.

The spatial representation is useful in computing free paths for trajectory analysis, and for answering questions about the identities of objects or features in given locations. The representation is also used to simplify the task of the two-dimensional modeller by explicitly representing the spatial relationships between objects.

The feature-based representation is linked to both the octree and the world models. It has entries for each object known or hypothesized to be in the workspace, including objects that have not yet been recognized. Each object is associated with the set of features that verifies its identity, and recognized objects are linked to their geometric models.

The feature-based representation is especially suited to answering questions about objects or features by name or by description. Some questions rely on both the representations for their answers. For example, deciding if an object is occluded from a particular viewpoint involves first finding the entry for the object in the feature-based representation, and then following the links to the spatial representation to find the answer.

For both the spatial and feature-based representations, interactions with the matching and recognition modules enhance accuracy of the information that is stored. When an object is recognized, its features and properties can be extracted directly from the model, allowing missing information to be filled in and errors to be corrected. For the spatial representation, recognition removes the uncertainties about the shapes of objects, allowing a more reliable description. The initial entries of both the spatial and feature-based representations are derived from the initial expectations received from the knowledge base.

5. IMPLEMENTATION

Large parts of the sensory system have been implemented, with varying degrees of complexity. In some cases, simplified modules exist to allow testing of more complex modules, and the modules have not all been integrated into a unified system. The inputs and outputs of the various modules have, however, been matched, so that it is possible to run the same data through all implemented parts of the system. Of course, even those parts that are considered implemented are subject to change as other modules interact with them, and as experience suggests improvements.

Integrating the system is a non-trivial task. It is being performed incrementally as the necessary hardware and sensor systems are constructed. A special-purpose operating system has been developed to take care of interactions between modules that may reside on different processor types, with different word sizes, and on different backplanes. In the following paragraphs, brief descriptions are given of the states of the various modules.

Currently, the knowledge base consists of a computer-aided design (CAD) data base containing boundary representations of objects (in an object-based coordinate system). It also includes individual octree representations for each object. The knowledge base also provides task-specific information, such as the number of

instances of each generic object, and their expected initial positions (Lumia [10]). A program exists to read in this data and instantiate the initial conditions for the world modeller and the feature-based representation, although the initial octree representation has not yet been incorporated. (The problem of initially instantiating the octree is treated in part by Ahuja and Nash [1] and Boaz and Roach [2]).

The supervisory process, which decides what sensors to use and ensures that the rest of the system is running, has also been partly implemented. As each process is added, the supervisor is modified to include any necessary additional capabilities. The two-dimensional world modeller is in the early stages of development. A simple tracking capability is implemented, and work is being conducted on generating projections of features, and making use of the aspect graphs of objects to speed up the predictions.

A version of the two-dimensional matcher also exists, although it is tailored somewhat to a specific task being performed in the automated factory. The recognition module, however, has so far not been implemented. The three-dimensional matcher, and its associated object pose finder have been implemented, although they are not yet integrated with the rest of the system (Rutkowski *et al.* [11]). The octree maintainer, too, has been implemented, and has also not been integrated into the system (Hong and Shneier [5]). Other tasks awaiting completion are the construction of the maintainer for the feature-based representation, and the integration of responses from multiple sensors (Kent *et al.* [8], describes the integration of structured light and reflectance ranging systems). The interface to the robot controller also needs extensions and enhancements. Currently, a major obstacle to integrating the various parts of the system is the acquisition of suitable hardware. The low-level modules of the system are implemented on a network of 16-bit microcomputers, while the higher levels are being developed on 32-bit machines. It was also found necessary to design and construct custom hardware for performing fast matrix operations. The two dimensional matcher and predictor are scheduled to be integrated into the current system in the coming year, and those modules such as the octree maintainer, three-dimensional modeller, and three-dimensional matcher, which exist on a mainframe computer, will be moved to individual 32-bit processors.

6. DISCUSSION

The higher levels of the NBS sensory system have been described in terms of the kinds of processes involved. A different perspective on the system arises from viewing the sensory system as four functional planes: the input plane, the interpretative plane, the representation plane, and the interface plane. There are currently seventeen functional modules in the NBS sensory system. An extensive network of interplane connections links the modules in the various planes. The system employs data-flow paths between modules which may be dedicated links, shared memory, or common bus transfers. The mode of transfer is transparent to the system modules. There is no necessary correspondence between hardware modules and functional modules. Hardware choices are dictated by speed and economics, while functional modules are organized so as to best carry out the sensory processing.

The input plane contains the sensors and their associated preprocessing hardware. Currently, two forms of vision (a floodlight-based system and a structured light

system) share the camera hardware module, and constitute the major sensory input to the system. While other sensors are actively under study, only exploratory work has been done on integrating these into the system.

The interpretative plane contains modules implementing low-level vision algorithms, as well as the higher level modules described in this paper. In this plane, the hierarchical structure of the sensory system is evident, as well as the servo-loop interaction of the prediction and analysis modules described above. This plane contains the processes that acquire and process sensor data, that predict the appearance of the world, and that match the predictions with the sensed input.

The representation plane contains modules concerned with the permanent, long-term, or integrated knowledge of the system. The a priori knowledge base contains information about the world such as the description of each generic part, the physical location of the machine tool jig, etc. The octree maintainer and the table maintainer are processes which build and update the volume-indexed and feature-indexed portions of the world model database.

The interface plane contains modules which handle communications between the sensory system and the rest of the world. A module called the server receives and interprets requests from the control system, and formulates answers based on the data currently stored in the representation plane. Another module, the ambassador, is responsible for communications with the external factory database which informs the sensory system which parts should be present during the task.

In addition to the modules in the planes discussed above, there are others that do not fit into any plane. One module, the supervisor, spans all of the planes. It is responsible for starting the system, monitoring and maintaining system operation, communicating with the operator, and determining which resources should be used in order to minimize the difference between the model of the world and reality. Another processor, the recognizer deals with unexpected features resulting from unexpected objects, or from known objects in unexpected locations. It tries to recognize unexpected objects by comparing them with all machining stages of all parts known to the system, not just those thought to be currently present.

A system that performs a similar function to that presented in this paper is the ACRONYM system of Brooks [3], although the ways in which the systems operate is rather different. ACRONYM also comprises four components. They are object models, predictions from the models, interpretation of images in terms of models, and descriptions. ACRONYM describes objects using primitives based on generalized cylinders. It constructs an object graph that has these volume elements as nodes, with arcs describing relationships between the nodes. In addition to the object graph, a restriction graph provides constraints on the volumetric models, and also defines a hierarchy of specializations of objects. ACRONYM uses the graphs to construct predictions of features expected in the scene using a constraint manipulation system, a powerful tool that not only provides expected matches of object and model features, but also provides three-dimensional information about the object instances. The system described in this paper also computes expectations, but they are more specific to the particular instances of objects expected in the scene, and are computed in a less general manner. Sacrificing generality for speed is justified because of the constrained domain and the real-time responses required. The construction of an explicit spatial representation also provides an alternate way of extracting the three-dimensional information. ACRONYM does not explicitly

construct windows in which to search for features, but rather attempts to find constraints on their allowable ranges. In our system, constraints in the spatial representation are directly represented by the cones in the octree, while those for features are explicit in the window sizes and the ranges allowed for feature values.

ACRONYM shares with our system the hierarchy of complexity of description. In our system, this is achieved by conglomerating features into objects, and objects into assemblies. In ACRONYM, conglomerations are constructed by attempting to find groups in the data that are related in a way consistent with a more general model (e.g., a set of aircraft at an airport).

The similarities in the two systems arise mostly from the fact that both are solving similar problems (although ACRONYM is solving a more general form of the problem). Our approach is strongly based on servoing the models to the data, and constructing a description of the world, while ACRONYM is attempting to solve a set of constraint equations that together explain the world, but may individually be incomplete. ACRONYM does not explicitly represent the object instances, and has no mechanism for dealing with unexpected objects.

Crowley [4] describes a system for representing the world to a mobile robot that has many features in common with that proposed here. In his system, each sensor modality has its own low-level preprocessors, which operate on the current data from the sensor. The results of the processing are integrated into a description of the world which is valid for longer times. The differences between the sensed information and the model information are used to servo the position of the robot, and to update the internal representation of the world. The differences between the two systems arise partly from the task domains for which they were developed, and partly from the kinds of a priori information available about the world. Crowley's system constructs a very simple description of the layout of the world, without any a priori information about the objects present. Our system constructs a more detailed description. While it can operate without a priori knowledge, it is particularly designed to use such information both to construct more meaningful descriptions of the world, and to better constrain the positions of the objects in the world.

7. CONCLUSIONS

This paper has described the design of the higher levels of a sensory system for a robot manipulator. The goal of the sensory system is to construct and maintain a representation of the workspace in a form suitable for very fast responses to questions about the world posed by other robot subsystems. This goal is achieved by separating the sensing process from the descriptive process, allowing questions to be answered without having to wait for the sensors to respond. The sensory system has to integrate and reconcile all incoming data from sensors and from a priori expectations. It has to servo the world model based on errors between the sensed data and the expectations constructed from earlier information about the world.

Four groups of processes were described to carry out these tasks. Predictive processes (world modellers) are needed to set up initial expectations about the world and to generate predictions about sensor responses for each sensory modality. Processes are also needed to analyze the sensory input. They can usually be broken into sensory-specific processes and sensory-independent processes, and can make use of the predictions in analyzing their inputs. Matching is a third function essential to the task. Matching involves comparing the sensed data with the expectations, and

provides the error signals that help register the internal representation with the world. The fourth process is the descriptive process that actually constructs and maintains the internal representation of the world. It constructs a spatial and a feature-based representation from both hypothesized and sensed information, and attempts to ensure that the representations contain as much as is known about the world at all times.

The four processes, plus other processes for supervising the operation of the sensory system and communicating with the other robot modules, together constitute a sensory system that is responsive to changes in the environment, but is also able to cope with interruptions in sensing and with unexpected objects in the world. It can respond rapidly to queries about the world even when the current positions of the sensors would make direct inquiries of the world impossible.

ACKNOWLEDGMENTS

We are grateful to the members of the Sensory-Interactive Robotics Group at the National Bureau of Standards, all of whom contributed substantially to the content of this paper.

REFERENCES

1. N. Ahuja and C. Nash, Octree representations of moving objects, *Comput. Vision Graphics Image Process.* **26**, 1984, 207–216.
2. M. Boaz and J. Roach, "*An Oct-tree Representation for Three-Dimensional Motion and Collision Detection*," Virginia Polytechnic and State University Technical Report, 1984.
3. R. A. Brooks, Symbolic reasoning among 3-D models and 2-D images, *Artif. Intell.* **17**, 1981, 285–348.
4. J. L. Crowley, Dynamic world modelling for an intelligent mobile robot, in *Proc. 7th Int. Conf. Pattern Recognit.*," Montreal, 1984, pp. 207–210.
5. T-H. Hong and M. Shneier, *Describing a Robot's Workspace Using a Sequence of Views from a Moving Camera*, *IEEE Trans Pattern Analysis and Machine Vision* 7, 6, 1985, pp. 721–726.
6. E. W. Kent, A hierarchical, model-driven, vision system for sensory-interactive robotics, in *Proc. Compsac '82*, Chicago, November 1982.
7. E. W. Kent, M. O. Shneier, and R. Lumia, PIPE–Pipelined image processing engine, *J. Parallel Distrib. Comput., J. Parallel Distrib. Comp.* **2**, 1, 1985, 50–78.
8. E. W. Kent, T. Wheatley, and M. Nashman, Real-time cooperative interaction between structured light and reflectance ranging for robot guidance, *Robotica* **2**, 4, 1984.
9. J. J. Koenderink and A. J. van Doorn, The internal representation of solid shape with respect to vision, *Biol. Cybernet.* **32**, 1979.
10. R. Lumia, Representing solids for a real-time robot sensory system, in *Proc. Prolamat 1985*, Paris, June 1985.
11. W. S. Rutkowski, R. Benton, and E. W. Kent, *Model-driven Determination of Object Pose for a Visually Servoed Robot*, Robot Systems Division, National Bureau of Standards, Washington, D.C., 1984.
12. M. O. Shneier, E. W. Kent, and P. Mansbach, Representing workspace and model knowledge for a robot with mobile sensors, in *Proc. 7th Int. Conf. Pattern Recogn.*, Montreal, 1984, pp. 199–202.
13. M. Shneier, 3-D robot vision, in *Proc. Int. Conf. Cybernet. Soc.*, Seattle, Wash., October 1982, pp. 332–336.
14. M. O. Shneier, W. S. Rutkowski, and T-H. Hong, *Using Chebyshev Polynomials for Interpreting Structured Light Images*, in Proc. International Conference on Robotics and Automation, St. Louis, MO, March 1985, pp. 17–20.
15. J. A. Simpson, R. J. Hocken, and J. S. Albus, The automated manufacturing research facility of the National Bureau of Standards, *J. Manuf. Systems* **1**, No. 1, 1983, 17–31.