Developing Robotic Systems with Multiple Sensors

MOHAN MANUBHAI TRIVEDI, SENIOR MEMBER, IEEE, MONGI A. ABIDI, MEMBER, IEEE, RICHARD O. EASON, MEMBER, IEEE, AND RALPH C. GONZALEZ, FELLOW, IEEE

Abstract—**Intelligent robots should be able to acquire, interpret, and integrate information from a variety of sensor modalities. Research presented in the paper deals with the development of robotic systems that utilize multisensory information to perform a variety of inspection and manipulation tasks. Design of multisensor systems is a complex and difficult task requiring resolution of several subtasks, including sensory modality selection, processing and analysis methods for information acquired by individual sensors, and integration of independently derived information in an accurate and efficient manner. A general approach for the integration of vision, range, proximity, and touch sensory data to derive a better estimate of the position and orientation (pose) of an object appearing in the work space is presented. Efficient and robust methods for analyzing vision and range data to derive an interpretation of input images are discussed. Vision information analysis includes a model-based object recognition module and an image-to-world coordinate transformation module to identify the three-dimensional (3-D) coordinates of the recognized objects. The range information processing includes modules for preprocessing, segmentation, and 3-D primitive extraction. The multisensory information integration approach represents sensory information in a sensor-independent form and formulates an optimization problem to find a minimum error solution to the problem. The capabilities of the multisensor robotic system are demonstrated by performing a number of experiments using an industrial robot equipped with several different sensor types.**

I. INTRODUCTION

A. On Developing Intelligent Robotic Systems

DEVELOPMENT OF INTELLIGENT robotic systems requires consideration of two different types of tasks. The first task deals with the design and development of individual components that in the system, whereas the second task is related to the proper integration of the individual components to form a complete system. Obviously, these two tasks are interrelated and successful development of a complete robotic system requires a *systems engineering* perspective. Specifications of the overall system must be utilized to guide the design of individual components and the framework for their integration.

Manuscript received March xx, 1989; revised January 17, 1990. This work was supported by the DOE's University Program in Robotics for Advanced Reactors (Universities of Florida, Michigan, Tennessee, Texas, and the Oak Ridge National Laboratory) under Grant No. DOE DE-FG02-86NE37968.

The authors are with the Electrical and Computer Engineering Department, The University of Tennessee, Knoxville, TN 37996-2100.

IEEE Log Number 9037593.

Over the past several years, the main emphasis of research studies has been on the development of individual components that can be utilized in a larger robotic system. These studies have contributed to the development of useful image processing, analysis, and interpretation schemes and various robot control, path planning algorithms. Research studies with the primary focus on the development of complete robotic systems have been comparatively much fewer in numbers. This may be due to the fact that such studies typically require extensive laboratory resources. Some of the noteworthy system development studies reported in the literature include, autonomous land vehicle (ALV) project related works [1], [2], automatic welding system developed at SRI [3], and a sheep shearing robot [4]. It should also be noted that whereas both theoretical and experimental approaches were utilized in the research associated with the individual component design, the main approach followed in system development research has been experimental. The complexity of most practical robotic scenarios required a systematic experimental research effort for system design and performance evaluation.

The research described in this paper is directed towards the development of an integrated robotic system capable of performing a variety of inspection and object manipulation tasks autonomously. Most of the industrial robots currently in use utilize very limited or no external sensory feedback, a fact that limits their ability in performing complex tasks [5], [6]. External sensory information derived from a variety of sensor modalities is critically important for robots operating in complex, unstructured and dynamic environments. Design of autonomous systems which effectively utilize multisensor inputs is a very challenging research task [7]–[11]. It involves consideration of issues such as sensor modality selection, low-level processing of sensor data, interpretation of information from a single as well as multiple sensory domains, decision making with noisy, uncertain or incomplete information and efficient and robust implementations for on-line operation of robotic systems. It should also be noted that in developing successful robotic solutions for a given problem, careful consideration of the specifications, requirements and constraints of a particular robotic work environment is essential.

0018-9472/90/1100-1285$01.00

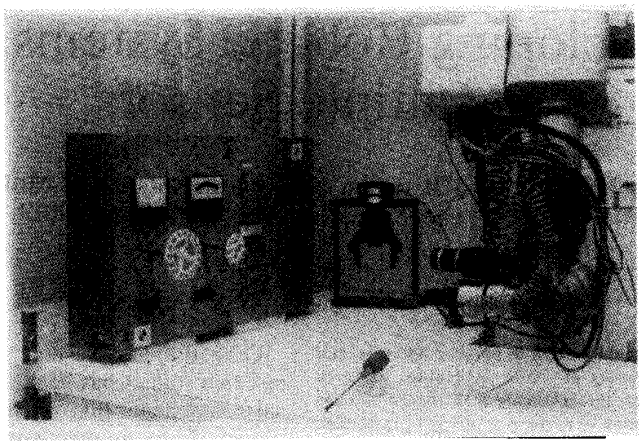


Fig. 1. Test panel and industrial robot with vision, range, touch, force, and proximity sensory capabilities. Test panel includes a variety of displays, meters, valves, controls, and switches.

B. Research Study Objectives and Overview

In this paper, we address two important classes of problems encountered in developing sensor-driven robots. These are: *how to analyze and interpret sensory information, in a particular mode, for object recognition and location*, and secondly, *how to integrate information acquired from a variety of sensor modalities to determine the exact location of a particular (already recognized) object*. Specifically, we discuss approaches for analyzing information acquired by vision and range sensors and integrating object location information acquired with vision, range, proximity, and touch sensors to derive a better estimate of the object pose.

The research is conducted in the context of developing robotic systems for advanced nuclear reactors. Use of autonomous systems in such hazardous environments can enhance safety and minimize operational costs. The experiments performed, using a specialized testbed, were designed to address various inspection and manipulation task requirements of industrial plants. Development of such robotic systems is indeed a complex task. We have undertaken an approach, which we believe allows making incremental progress towards the eventual development of such a system. Our efforts are directed towards researching issues associated with acquisition and analysis of multiple sensory data using a robotic system. This is accomplished by focusing on the development of an autonomous system that is capable of performing various inspection and manipulation tasks associated with a control panel. For example, these tasks can range from reading of various meters and displays, to operating different types of valves, switches, and controls. This study provides an implementation of a laboratory based robotic system that may serve only as the initial effort in the development of a field deployable system capable of autonomous operation in unstructured and dynamic environments.

Our experimental set-up includes a test panel, a robot having multiple sensory capability, computers, and various manipulation tools. The test panel and the robot with various sensors mounted on the arm are shown in Fig. 1. The industrial robot consists of a Cincinnati Milacron T^3-726 robot with enhanced sensory mechanisms. The sensors implemented on the robot include vision, range, and proximity as non-contact devices and touch and force/torque as contact devices. All of these sensors are mounted on the gripper itself. The camera and range sensor point in a direction parallel to the fingers, while the proximity and touch sensors are mounted on the fingers. The force/torque sensor is mounted between the gripper and the face plate of the end effector for measurement of the forces acting on the gripper.

The vision sensor considered is a simple black-and-white camera mounted on a robot arm. We describe the image processing and interpretation hierarchy for recognizing various objects appearing in a scene. Input images of the work space of a robot are analyzed to extract spectral, spatial, and relational domain features. These features are matched with object models stored in the system's knowledge base. Once the identities of various objects in the work space are accurately derived, their exact positions in the 3-D space needs to be determined. For this, we present an efficient image-to-world coordinate transformation procedure that utilizes knowledge about the locations of four control points and the camera to derive a transformation matrix. The range data analysis is presented as a step-wise approach utilizing efficient, parallel processing algorithms for low level image processing, segmentation and 3-D primitive extraction. The algorithms are implemented on a parallel processing hypercube computer. The multisensory integration approach presented utilizes a representation scheme to assimilate information derived from various sensors in a standard, sensor-independent form. We formulate an optimization problem to merge information from disparate sources to provide a

minimum error solution. The specific robotic application considered is that of accurate pose determination of an object appearing in the work space. Such information is critical to perform various manipulation and inspection tasks autonomously. Applicability of the analysis tools and techniques described in the paper is verified by performing a number of experimental studies utilizing a laboratory-based testbed including an industrial robot equipped with multiple sensors.

II. A Vision System for Object Recognition

The main goal of a robot vision system is to provide an accurate interpretation of a scene utilizing visual information as the primary source of input. This interpretation can be provided in a variety of forms and at different levels of abstraction. A useful form of interpretation may include an object location map where different types of physical objects appearing in the scene are independently recognized and accurate locations of the objects in the scene are determined. Also of utility is the information regarding the status or condition of an object. Design of a computer vision system that can perform such object recognition and scene interpretation is a complex and challenging task. The main difficulty underlying this task stems from the images being two-dimensional (2-D) projections of the three-dimensional (3-D) real scene and innumerable combinations affecting the illumination source, scene and sensor parameters can result in the same observable value of recorded image intensity.

In order to make the above problem computationally tractable, a model-based approach to computer vision is proposed [12]. The approach requires knowledge of a set of models associated with objects that are expected to appear in the scene. These models are recorded in the knowledge base of the system. Various features from the input images are extracted using low-level, general-purpose operators. These operators are robust in extracting image features that characterize various object properties. Finally, a correspondence is sought between the image derived features and scene domain models to recognize the objects. This is accomplished by utilizing various decision making schemes in the matching module. Successful design and implementation of a vision system utilizing the model-based paradigm is affected by a number of factors. These include, the ability to derive suitable object models, the nature of image features extracted by the operators, a computationally effective approach to handle the task of matching, schemes utilized for representing knowledge about scene and other data structures, and an effective control mechanism for guiding the overall operation of the system.

A. Overview of the Vision System

Development of model-based vision systems has been an active area of research in robotics [13]–[15]. Several of these studies involve direct utilization of 3-D features and models for object recognition. The 3-D features are typically extracted using passive or active approaches such as stereo, structured light or time-of-flight sensors [16], [17]. These approaches are generally computationally expensive or require specialized hardware. In our application, we use the eye-in-hand configuration for the camera. This allows the robot to position the camera in a specified location and orientation using its six-degrees-of-freedom arm. After determining the pose of the control panel, the robot arm can be directed to acquire images using an orthogonal viewing geometry. This allows extraction and use of 2-D features from the images which are later matched with the 2-D models of the objects. In order to calculate the locations of the recognized objects in the 3-D workspace, we utilize a transformation matrix affecting the image-to-world coordinate transformation.

Robustness and ease of expandability to accommodate changes in the task environment are two key features guiding the development of the vision system. The system is compartmentalized in two basic groups of procedures. The first group consists of general-purpose procedures for camera calibration, image acquisition, knowledge acquisition, image segmentation, matching, and robot arm movements. The second group consists of special-purpose procedures designed primarily for determining status of individual objects. The main functions supported by the first set of procedures in the system are:

- to move the robot arm to any desired position in the environment,
- to acquire images of different resolutions,
- to perform segmentation of input images,
- to allow a user to input object attributes in spectral, spatial and relational domains, and to encode this information in the system knowledge base,
- to extract spectral, spatial and relational domain features from the acquired images,
- to perform matching of image-derived features with object attributes to recognize various objects, and
- to determine the location of objects in the field of view of the camera.

The system is developed in such a fashion that the previous functions are performed by procedures that are general-purpose, i.e., they rely on minimal knowledge about the scene and its constituent elements. For example, the procedures employed for recognizing and locating a meter in the panel is basically similar to that of recognizing and locating a valve. A flow chart showing the sequence of processing steps included in this stage of the vision system is presented in Fig. 2. Derivation of the transformation matrix to convert the image coordinates into world coordinates is also accomplished as a vision system task. Details of the exact procedure followed to derive the transformation matrix are presented in Section II-B. The region growing procedure utilized for segmentation employs uniformity of gray-level intensities to identify distinct homogeneous regions in the input image. The region uniformity calculation is based upon the mean and variance of the gray-level intensities of the pixels within a

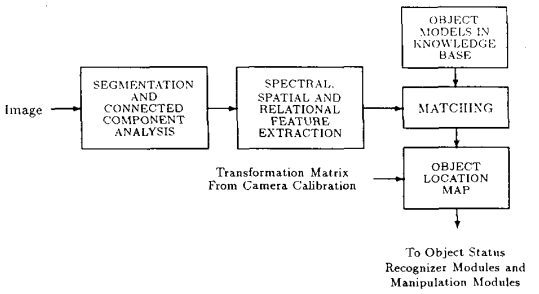
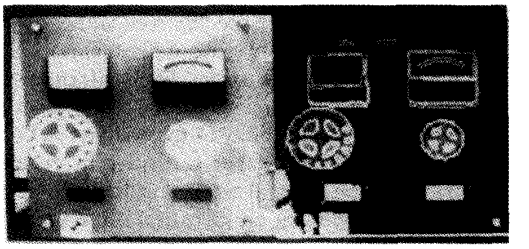


Fig. 2. Functional modules used in the vision system to recognize objects and to determine their locations.

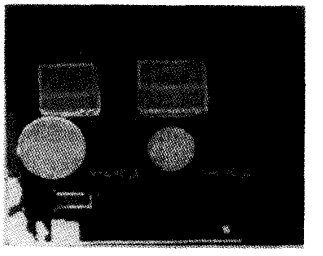
region. The technique has proven to be quite robust and efficient in processing a variety of images. The segmented images are analyzed to extract spectral, spatial and relational domain features of the detected blobs. The only spectral domain feature utilized is the relative mean gray level of a blob. The spatial features extracted from the blobs include: size, shape (using Fourier shape descriptors), height, width, and aspect ratio. Relational features can be derived using the coordinates of the smallest enclosing rectangle associated with the blobs. The matching module utilized for object recognition is based on a bottom-up approach. The module first searches for the subobjects in an image, then examines the appropriate relational constraints to see if a valid object can be constructed by the detected subobjects. Initial search for the subobjects involves examination of the spectral and spatial properties of the blobs and their matching with the object attributes stored in the system's knowledge base. The attributes associated with the objects expected to appear in the scene are supplied by the user. The user is directed through a question-answer sequence to acquire the description of spectral, spatial and relational properties of various objects and their subparts. The matching process utilizes relative values of objects and region descriptors to add generality. The system can be easily extended to recognize additional types of objects. Details of the vision system architecture are presented in [18].

As opposed to the above described functions and procedures, the second group of procedures in the system addresses specialized requirements for dealing with individual objects. The main function supported by procedures in this group is to determine the status of various objects that are recognized and located using the procedures from the first compartment. Depending upon the type and nature of the object, the camera mounted on the arm is moved to take images using orthogonal viewing geometry. These images are analyzed to determine the status of an object. Detailed discussion of the routines developed for object status recognition is provided in [19].

The vision system has been tested in the laboratory environment utilizing the testbed described earlier. The system has performed successfully in recognizing various objects mounted on the test panel and in determining their locations and status. Typical results of vision system



(a) (b)



(c)

Fig. 3. (a) Image of the left half of the test panel acquired by the camera mounted on the robot arm. (b) Segmentation of the image shown in (a). (c) Results showing the recognized objects and their locations after performing matching.

processing are illustrated in Fig. 3. In Fig. 3(a) the input image of the left half of the test panel is shown. Segmentation results are shown in Fig. 3(b). Finally, in Fig. 3(c) we show the results of the object recognition phase. The system has been operational in our laboratory for about one year and over one hundred tests were conducted. Recently, the vision system software has been transported to another laboratory employing a different robot, camera, and illumination conditions. The system performance has been quite satisfactory and robust.

B. Image-to-World Coordinate Transformation

In order to perform various inspection and manipulation tasks autonomously, the robotic system must determine the exact locations of various objects appearing in the work space. In this section, we describe derivation of a transformation matrix which can accurately map the image coordinates into 3-D world coordinates. The need for accurate and efficient positioning of a robot arm with respect to a given target has been addressed in many ways using painted lines, infrared beacons, and more sophisticated 3-D vision techniques [20], [21]. By utilizing knowledge of the geometrical structure associated with the scene, one can often extract 3-D information from the sensed 2-D images. Pose recovery using landmark approaches has shown to be accurate, fast, efficient, and robust [22], [23]. For a broad range of robotic work environments, introduction of a series of landmarks can be made easily and cost-effectively.

Here, we present a new pose estimation technique which we have implemented on our robotic system. In this

algorithm, we show how the relative position of the robot arm can be determined up to a desired accuracy using a four light targets mounted on the test panel. The method is purely analytic and requires the use of fairly simple image analysis operations, which makes it an attractive solution for landmark-based pose estimation. With reference to Fig. 4, we would like to recover the 3-D position of each of the point-targets, P_1, P_2, P_3, and P_4 knowing their respective images, p_1, p_2, p_3, and p_4 as well as their relative position with respect to each other ($P_i = (X_i, Y_i, Z_i)^T$ and $p_i = (x_i, y_i)^T$). The solution is obtained by direct computation and is more efficient and accurate than other iterative techniques [24].

In an augmented coordinate system, the image coordinates (x, y) of an image point are related to the world coordinates (X, Y, Z) by

$$\begin{bmatrix} c_{h1} \\ c_{h2} \\ c_{h3} \\ c_{h4} \end{bmatrix} = \begin{bmatrix} a_1 & a_2 & a_3 & a_{10} \\ a_4 & a_5 & a_6 & a_{11} \\ a_7 & a_8 & a_9 & a_{12} \\ a_{13} & a_{14} & a_{15} & a_{16} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

$$x = \frac{c_{h1}}{c_{h4}} \tag{1}$$

$$y = \frac{c_{h2}}{c_{h4}} \tag{2}$$

where c_{h1}, c_{h2}, c_{h3}, and c_{h4} represent the elements of the homogeneous coordinates. In this system, we use a pinhole camera model for the vision sensor (Fig. 4). To compensate for the abberations of the lens system, the linear relationship between image coordinates and sensor coordinates is approximated by a cubic, the coefficients of which are determined experimentally.

The image coordinates (x, y) and the world coordinates (X, Y, Z) are related as follows [6]:

$$x = \frac{\mathcal{F} X^c}{\mathcal{F} - Z^c}, \tag{3}$$

$$y = \frac{\mathcal{F} Y^c}{\mathcal{F} - Z^c}, \tag{4}$$

$$W^c = AW, \tag{5}$$

where

$$W = (X, Y, Z, 1)^T, \tag{6}$$

$$W^c = (X^c, Y^c, Z^c, 1)^T, \tag{7}$$

$$A = \begin{bmatrix} a_1 & a_2 & a_3 & a_{10} \\ a_4 & a_5 & a_6 & a_{11} \\ a_7 & a_8 & a_9 & a_{12} \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{8}$$

The intermediate parameters X^c, Y^c, and Z^c represent positions in the camera coordinate system, and the elements a_1 through a_{12} represent the matrix transformation A from world coordinates to camera coordinates. The parameter $\mathcal{F}$ represents the focal length of the camera. By the very nature of the problem, the parameters a_1 through a_9 characterize the effect of rotation only,

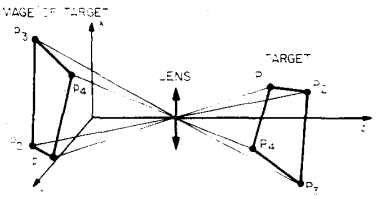


Fig. 4. Target points and their images using a pinhole camera model.

whereas a_{10}, a_{11}, and a_{12} characterize the effect of translation. The intent here is to compute the elements of the transformation matrix, a_1 through a_{12}, using the image of the target points and information about their relative positions to determine the 3-D locations of those targets with respect to a coordinate system fixed relative to the camera. Since these target points are rigid with respect to the panel, this allows full recovery of their pose.

Each point provides a pair of linear equations involving the unknown parameters a_1 through a_{12}; therefore a minimum of six points is required to uniquely define the matrix A. However, we have shown that four is the minimum number of points required to completely compute A. The target can form a general quadrangle. In our pose estimation system, first we use three target points to find a multiple solution to the pose problem, then add a fourth point to disambiguate this solution. The output of this procedure is the complete 3-D pose of the target, $(X_0, Y_0, Z_0, \alpha, \beta, \gamma)$, with respect to a fixed coordinate system [24].

The procedure transforming image coordinates into 3-D world coordinates was incorporated in the control panel experiment described earlier. The four points required for the transformation were associated with the lights which were mounted on the border of the panel. Location of the lights corresponding to the panel and the actual distances between the lights are measured. A simple procedure utilizing two images of the panel taken from the robot mounted camera was used to detect the image coordinates of the four lights. One of the images was taken with the lights turned on and other with the lights off, these two spatially registered images were subtracted followed by a simple thresholding and region growing operation in order to detect the lights. Through the experiments that we have conducted in our laboratory, this procedure has proven to be quite robust and efficient. Typical results of the light detection procedure are shown in Fig. 5.

III. Analysis of Range Information

The recovery of 3-D information from 2-D images has always been a difficult and often computationally-intensive task. Our investigations include development of efficient and robust algorithms for stereo image analysis [17] as well as for extracting shape information from range images generated by active sensing. In this section, we concentrate on range information analysis. This information will again help in the tasks of object recognition and

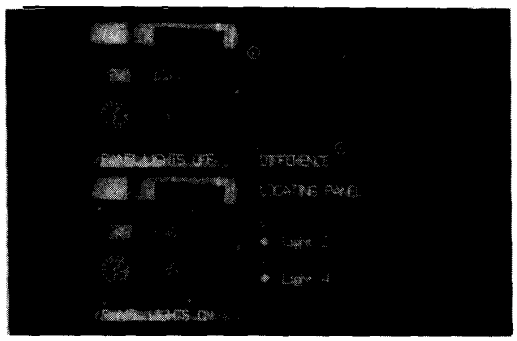
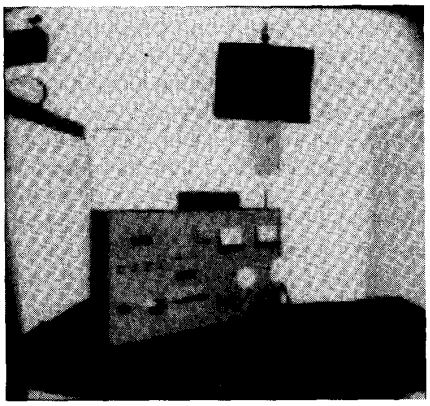


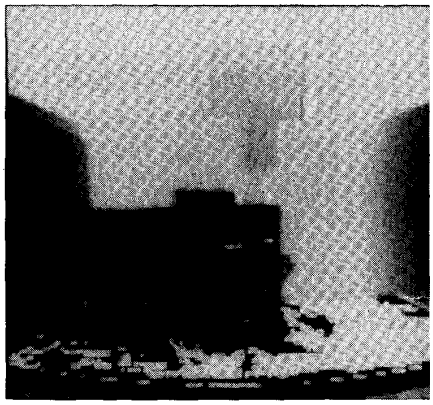
Fig. 5. Image-to-world coordinate transformation determination. To determine the 3-D location of the control panel, two images of the panel, one with the lights turned off and another with the light on are taken. These images are analyzed to detect the 4-lights to derive the transformation matrix.

pose determination [25]–[28]. As an example, consider the intensity/range pair shown in Fig. 6. The left image shows the intensity image of a panel scene. The right image shows the corresponding depth map where the displayed image intensity represents the distance from the sensor to the nearest object detected. The laser scanner used to generate these pairs of images maps the measured distance nominally between 3 and 30 ft to an image intensity between 0 and 255. Since the purpose of our experiments is to detect, identify, and manipulate objects which are mounted on the panel, only a small fraction of the total dynamic range of the depth map is used. We use the algorithms described in the previous section to determine the pose of the panel. Once the panel is identified, the depth intensity is mapped so that the intensity of the area of interest occupies most of the dynamic range of the image. Examples of such operation are shown in Fig. 7 where by linear scaling, the intensity of the panel region is increased from 0.1 to approximately 0.25, 0.50, 0.75, and 1.0. As we will see later, the dynamic range in depth maps is a critical factor in achieving accurate surface characterization.

In order to accurately and reliably identify objects in a scene, range images must be segmented into basic components, such as smooth patches. A segment is a maximal connected component of pixels that exhibit some degree of spatial or surface coherence. Segmentation is typically followed by a 3-D primitive extraction phase to identify a set of symbolic primitives such as planes, spheres, cylinders, ellipsoids, and cones. In the following, we describe a five-step procedure that implements an efficient range image analysis scheme. Tasks which are computationally intensive, such as segmentation and primitive extraction, are performed using parallel algorithms. This segmentation system includes both existing techniques as well as newly developed methods [29]. As a first step, the image is processed to eliminate sparse high amplitude noise. In the second step, local geometric parameters are computed using partial derivatives of the image data. Next, the necessary parameters for range segmentation are



(a)



(b)

Fig. 6. A pair of intensity and range images ($128 \times 128 \times 256$) of the control panel acquired using the Odetics range scanner. Darker gray levels in the range image correspond to objects closer to the sensor and brighter to those farther away. (Original image is courtesy of the Oak Ridge National Laboratory.)

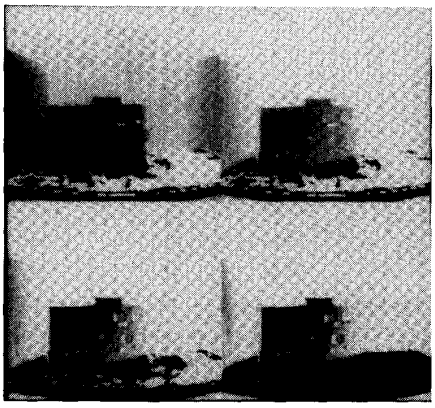


Fig. 7. Linear mapping of range intensity from dynamic range of 0.1 (Fig. 6) to 0.25 (top left), 0.5 (top right), 0.75 (bottom left), and 1.0 (bottom right).

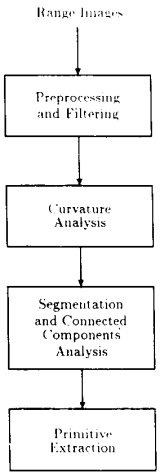


Fig. 8. Steps involved in the low level analysis of range images.

evaluated then used to group pixels within areas of smoothness. This is followed by a fourth step where regions that satisfy some consistency checks are extracted from the image. Finally, the extracted patches are fitted to a number of primary surfaces then labeled accordingly (Fig. 8). These steps are iteratively repeated on the image until no change occurs. The objective is to minimize the number of connected components. The tools used during this process are linear filters (Gaussian), nonlinear filters (median), surface evaluation of normals and curvatures. The surface fitting procedure is performed using a QR factorization algorithm. For this purpose, we have developed two new algorithms, one for a shared-memory parallel system and the other for a message-passing parallel system [30]. The step following primitive extraction is generally associated with actual object recognition. This is typically a model-based approach where 3-D object models stored in the knowledge base are matched against the extracted primitives. Our implementation does not yet include the matching component.

Range data is often very noisy. This noise is caused by imperfections in the mechanical scanning system and signal conditioning and processing of the laser range finder. Meaningful use of this data requires a filtering preprocessing step in order to reduce the effect of quantization and sensor noise. This is accomplished through median and gaussian filtering. There are two types of errors to be considered: measurement errors and gross errors. Measurement errors are normally distributed. Gross errors (classification errors), on the other hand, are unbalanced and often violate the smoothness assumption to a great extent. Measurement errors are filtered out using gaussian mask operators, whereas gross errors are filtered out using median filters.

Local geometry depends largely on partial derivatives. Differentiation is not robust against noise, i.e., even a low level of noise may disrupt differentiation severely. For example, the two functions $f(x)$ and $\hat{f}(x) = f(x) + \epsilon \sin \omega x$ can be made arbitrarily close by decreasing ϵ while the amplitude of the term $\hat{f}' - f' = \epsilon\omega \cos \omega x$ increases linearly with ω. Moreover, the objective of local geometry analysis is the location of discontinuities (step and roof edges), where differentiation is less reliable. Hence, numerical estimation of partial derivatives of digital surfaces is noise prone and is an ill-posed problem in the sense of Hadamard. Well-posedness and numerical stability of the differentiation step requires the regularization of the input image by a regularizing filtering operation preceding differentiation. Differentiation can be regularized using low-pass filters. Gaussian masks are used as regularization operators when calculating derivatives, for reasons of flexibility and optimality [31], [32].

Once the partial derivatives are evaluated, they are used to link pixels within areas of smoothness and to characterize geometric changes. Derivatives of different type and order may be needed, possibly at different scales, so that the range image can be analyzed at different resolution levels, depending on the amount of relevant detail. In a general-purpose segmentation system, it is necessary for the operator to be rotationally symmetric, hence nonresponsive to object orientation. Furthermore, in order to facilitate its parallel implementation, an ideal operator should have no data dependencies. Gaussian masks satisfy these conditions.

Surface curvature estimates are extremely sensitive to noise because they require the evaluation of second order partial derivatives over a discrete grid of sample points. This process is also mathematically ill-defined as well as noise-sensitive, since high frequency noise is amplified. In fact, the 8-bit quantization noise by itself seriously degrades the quality of curvature estimates unless large windows are used. The segmentation procedure that we have implemented therefore utilizes the parameters of the lowest order before analyzing the higher order terms. This multistep segmentation involves analysis of the range data in the first step, next it examines the surface orientation features and if the segmentation is still not complete then utilizes surface curvature for classification as a last resort. This multi-step examination of the features allows us to develop a more efficient and accurate overall approach than one which utilizes all of them together or does not consider some of the features [29].

Surface orientation is given by the normal vector to the surface at each point

$$n = \frac{(-g_u, -g_v, 1)^T}{(1 + g_u^2 + g_v^2)^{1/2}}.$$

The orientation of the normal vector is characterized here by the following function:

$$\tan^{-1}(g_u / g_v).$$

Surface curvature is given by the Mean (H) and Gaussian

(K) curvatures:

$$H = \frac{(1+g_v^2)g_{uu} + (1+g_u^2)g_{vv} - 2g_u g_v g_{uv}}{2(1+g_u^2+g_v^2)^{3/2}},$$

$$K = \frac{g_{uu}g_{vv} - g_{uv}^2}{(1+g_u^2+g_v^2)^2}.$$

One of the most crucial issues associated with the use of such techniques in real-time systems, is that of efficient implementation. For this reason, we have developed parallel processing implementations of range processing modules, including filtering, curvature analysis, connected components analysis, and primitive extraction. All operations—except for the connected components analysis—are neighborhood operations. Their parallel implementation on a hypercube-connected parallel computer is a relatively straightforward task [33]. We have developed a new algorithm for the connected components module implemented on a message-passing system [29]. It can be summarized as follows:

- *Step 1:* The feature image is divided among the processors available by standard broadcasting.
- *Step 2:* A raster scan connected component analysis is done on each node and a linked list of adjacent regions generated for merging.
- *Step 3:* The bottom and top rows of each node and the linked lists are interchanged among the nodes using cube doubling.
- *Step 4:* The linked lists are merged and updated.
- *Step 5:* The final linked list is used to update the raster scan connected component analysis.

The resulting connected components are fitted to a general quadric using the Givens transformation that is solved for using the QR factorization. Details of the characteristics of the quadrics, the new parallel algorithm implementing the Givens transformation, as well as the least-squares solution to the fitting problem are given in [30]. This implementation has been carefully tested using a series of experiments. Efficiency gains achieved by this algorithm are obviously dependent on the number of nodes available for computing. The speed-up ratios as a function of parallel computer cube size are illustrated in Fig. 9. Note, that remarkable efficiency gain is achieved initially, and the drop in the gain is due to the limits of parameters associated with the interprocessor communication.

In the following, the complete segmentation process is illustrated using a 256×256 real range image (Fig. 10(a)) containing simple objects. The reason behind using such a scene instead of the one shown earlier is to carefully test the algorithms developed using objects of known shape. Research is underway to apply this methodology to real range data. Data used here is coded from 0 (near) to 255 (far). The resulting image after the median filtering is shown in Fig. 10(b). Results of the segmentation step are presented in Fig. 10(c). Using increasing higher order

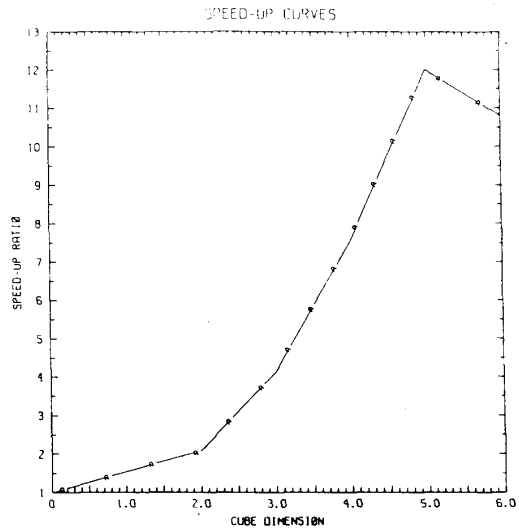


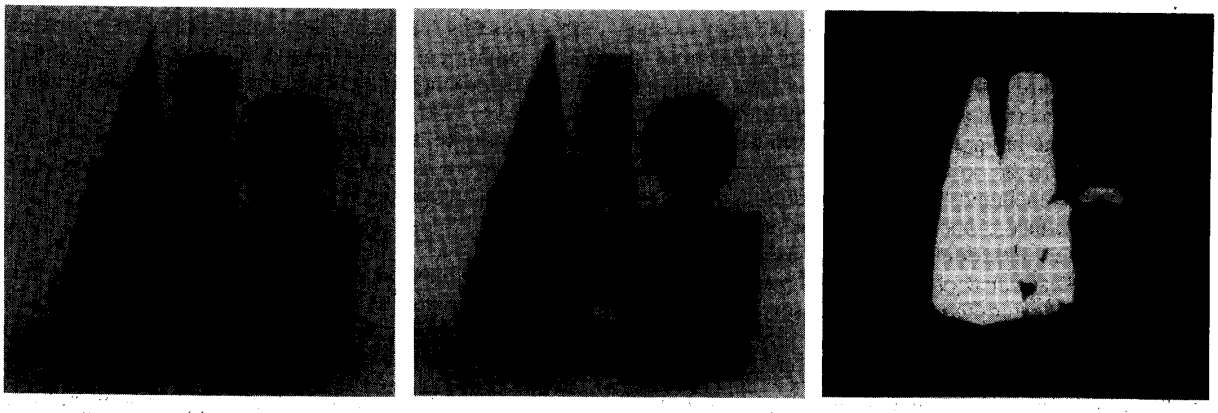
Fig. 9. Efficiency gains as a function of the parallel computers cube size. This curve corresponds to the surface fitting phase of range analysis.

surface characteristics, the individual components of the scene are isolated from the rest of the image then fitted. In the order they were detected, they are plane 1, sphere, cylinder 1, plane 2, cylinder 2, and cone. All components (except for the cone) were segmented using zero and first order parameters. For the cone, however, both the Mean and Gaussian curvatures are used to separate it from some of the noise generated by the table on which the scene is resting.

Work is underway to apply this methodology to panel images (Fig. 6). This is a more tedious task that the one just described because of the resolution of the range scanner and the complexity of the depicted scene. The objective is to recover the surface primitives that best describe the panel itself and its components: valves, switches, controls, etc. The panel will be described by a series of planar patches which can be matched against primitives stored in a CAD database. Components of the panel, such as the emergency knob, will be decomposed into planar, cylindrical, and spherical patches which will be matched also against its own primitives. Reasoning about the connectivity of those patches need to be addressed at this level. The underlying difficulty in using the data shown in Fig. 6 is the low resolution in both depth and spatial components. Detection and recognition operations in our multisensor robotic system are based on these primitives.

IV. Integration of Multiple Sensor Information

In the previous two sections, we discussed issues related to the analysis of data acquired by vision and range sensors. Such analysis provides two types of information: the identity of objects, and the position and orientation



(a) (b) (c)

Fig. 10. (a) Range image (256×256×256) of a scene composed of four objects with distinctive geometrical shapes acquired using the DSC scanner. (b) Resulting image after median filtering. (c) Results of range analysis; image was segmented into six regions. (Original images are courtesy of Digital Signal Corp.)

(pose) of these objects in their 3-D environment. In this section, we discuss integrating information acquired from several sensory modalities. The use of multiple sources of sensory information can be beneficial in improving the accuracy of the pose estimate and in lowering the cost of acquiring information by choosing an optimum sensing strategy. The sensory information integration approach is discussed for a specific robotic task, that of accurate identification of the 3-D position and orientation (pose) of an object. Pose information is critical to the success of various inspection and manipulation tasks. The overall integration approach is, however, applicable to a variety of other problems in which parameterized information is required (e.g., determining object or feature size, surface shape, or object status information).

Sensory information for determining pose can come from a variety of sources. Common examples include single camera images, depth maps produced by stereo vision or laser ranging devices, single point data generated by touch, proximity, or range sensors, and information provided by structured lighting or calibration marks used with visual images. Such data only provides partial information about the pose of an object, it may be noisy, and it comes in a variety of forms. The problem is to merge such information to arrive at the best estimate of object pose.

We address the problem of determining the pose of a known object, given partial pose information from different types of sensors. From our viewpoint each sensory measurement provides geometric information constraining the position and orientation of an object feature; e.g., if depth is unknown, then visual sensing of an object vertex can reveal that the vertex lies somewhere on a certain line in space. Different sensors provide different forms of these constraints and also involve different measurement accuracies. We assume object features have already been matched to sensory data. (See, for example, work by Stockman and Chen [34] regarding the problem of matching geometric features to sensory data.)

The vast majority of other research involving the fusion of information for determining object pose has involved only the use of a single sensor, traditionally vision, and more recently range. For example, Bolle and Cooper [35], Faugeras and Hebert [36], Bilbro and Snyder [37], and Grimson et al. [38] have each described methods for combining multiple range measurements, but have not investigated how these methods might be used with arbitrary types of sensors. One notable exception is the work of Durrant-Whyte [11]. Using a widely applicable approach, he uses Bayesian techniques to combine probabilistic descriptions of sensory data to build and maintain a single world model. In determining object pose, however, he treats position and orientation separately, without addressing the complexities of their nonlinear interrelationship. This makes the method ill-suited to problems where the data is sparse and only partial pose information is provided by each sensory measurement. Smith and Cheeseman [39] have described a method for chaining and merging uncertain transformations and have applied it to 2-D mobile robot positioning. Their merging process is similar to what is accomplished with our method in three dimensions; however, they too do not allow the use of partial pose information. To our knowledge, our method is the only one which handles partial noisy data from a variety of sensing modalities.

A. Determination of Object Pose Using Multiple Sensors

Each sensory measurement involves a set of parameter vectors $\{\boldsymbol{P}, \boldsymbol{M}_i, \boldsymbol{X}_i, \boldsymbol{Y}_i\}$, where $\boldsymbol{P}$ represents the global pose parameters, $\boldsymbol{M}_i$ represents the parameters of an object feature in model space, $\boldsymbol{X}_i$ represents the parameters of the object feature in transformed space (i.e., the transformed object feature), $\boldsymbol{Y}_i$ represents parameters of the

sensory feature, and i is the index of the sensory measurement. Although, we have primarily considered parameters that are purely geometric (e.g., the position of points, the direction of lines and surface normals), the generality of the integration method given below does not require them to be so. For example, optical properties could also be included: the object model could contain surface reflectance properties, the pose vector could contain parameters describing environmental lighting conditions, the transformed object features could contain the intensity of reflected light, and the sensory features could contain gray level values.

These parameter sets are related to one another by the double mapping

$$\boldsymbol{M}_i \xrightarrow{\mathscr{P}(\boldsymbol{P})} \boldsymbol{X}_i \xrightarrow{\mathscr{S}_i} \boldsymbol{Y}_i \tag{9}$$

where $\mathscr{P}(\boldsymbol{P})$ represents the transformation of the object feature from model space to the transformed space (a function of global pose) and $\mathscr{S}_i$ represents the sensory process that maps a transformed object feature to sensor feature space (a many-to-many mapping). From this perspective, this section addresses the problem of determining $\mathscr{P}(\boldsymbol{P})$ when given matched (sub)sets of $\{\boldsymbol{Y}_i\}$ and $\{\boldsymbol{M}_i\}$ and knowledge of the associated sensing processes $\{\mathscr{S}_i\}$.

With the approach presented here, each of the parameter vectors, $\boldsymbol{P}$, $\boldsymbol{X}_i$, and $\boldsymbol{Y}_i$, is described using a distance function rather than explicit values in order to incorporate estimates of sensing errors. In this manner, global pose is described by $E(\boldsymbol{P})$, (local pose estimates by $E_i(\boldsymbol{P})$), a transformed object feature by $f_i(\boldsymbol{X}_i)$, and a sensory feature by $g_i(\boldsymbol{Y}_i)$. Evaluation of each function for a given set of parameters indicates our estimate of "error" for that set, and the location of a function's minimum represents our estimate of the most likely values. In particular, our best guess of global pose is given by the minimum of E. Note that the set $\{g_i(\boldsymbol{Y}_i)\}$ is extracted from the sensory data. Weighting of one sensory measurement over another is done by scaling of the individual functions.

Our task is to perform the inverse mapping given in the above relationship and create the error function for pose, $E(\boldsymbol{P})$, given the set $\{g_i(\boldsymbol{Y}_i)\}$ and models of the corresponding mappings. We begin by defining the functions $\{f_i(\boldsymbol{X}_i)\}$; i.e., for the object feature corresponding to each sensory measurement, we define an error as a function of its transformed feature parameters. We note that for a given instance of $\boldsymbol{X}_i$, there are possibly an infinite number of instances of $\boldsymbol{Y}_i$ that it could map to according to $\boldsymbol{S}_i$ (e.g., for a given position and orientation of a plane, there are an infinite number of points which could sample it). The $\boldsymbol{Y}_i$ that we choose to associate with that $\boldsymbol{X}_i$ is the one which is most likely (i.e., the one where the function g_i is minimum), and the error that we assign for that $\boldsymbol{X}_i$ will be the error for the $\boldsymbol{Y}_i$ so chosen. Thus we define

$$f_i(\boldsymbol{X}_i) = \min_{\boldsymbol{Y}_i} g_i(\boldsymbol{Y}_i)$$

$$\text{subject to } \boldsymbol{X}_i \xrightarrow{\mathscr{S}_i} \boldsymbol{Y}_i. \tag{10}$$

In words this states that the error assigned to a particular $\boldsymbol{X}_i$ is equivalent to the least error possible according to the corresponding sensory measurement and sensing model.

Because the relationship $\mathscr{P}(\boldsymbol{P})$ defines $\boldsymbol{X}_i$ uniquely for a given $\boldsymbol{P}$, we can write $\boldsymbol{X}_i$ as a function of $\boldsymbol{P}$, and define the distance function for our local pose by

$$E_i(\boldsymbol{P}) = f_i(\boldsymbol{X}_i(\boldsymbol{P})). \tag{11}$$

This states that the local estimate of error for a given pose is the same as the error assigned to the resulting transformed object feature parameters. Globally, the error for a given pose is defined as the total error defined by the local pose estimates. Formally, this is written as

$$E(\boldsymbol{P}) = \sum_i E_i(\boldsymbol{P}) \tag{12}$$

where i varies from one to the number of sensor measurements involved.

Evaluation of this global pose function for an arbitrary pose reflects our estimate of how unlikely that pose is, based on the sensory measurements. Several aspects of this function are significant. The location of its minimum provides the best guess of pose, while its value at the minimum provides a measurement of agreement among the sensory data. Although we do not address the matching problem in this section, bad correspondences will result in "abnormally" high evaluations of the associated local pose function at the minimum. In addition, the second partial derivatives of the function at the minimum (how quickly it accelerates in value) indicates how tightly constrained the global pose estimate is likely to be. Detailed development of this method is presented in [40].

Advantages of this method are its unified treatment of a wide variety of geometric sensory information and its incorporation of sensory errors into the pose estimation process. It does require that each sensory measurement be put in the form of a distance function, a process requiring a suitable model of the associated errors; however, any method which performs a similar function would require equivalent information.

A disadvantage is that in general it requires numerical techniques to evaluate the minimum of the required functions, so that the usual problems of convergence and global minimums can apply. The nature of the solution method makes it more applicable to sparse problems where single-sensor techniques are not available.

B. Classification of Sensory Data

The classification we will use requires that each sensory feature describe either a point, a line, or a plane in the world coordinate system. According to the sensory models we consider, each sensory feature is related to an object feature (described in world coordinates) by one of the following relationships: *contains*, *is contained in*, or *is*. Examples of each of the resulting nine classes of sensing are given in Fig. 11. Because each of the nine classes involves a particular relationship between the object fea-

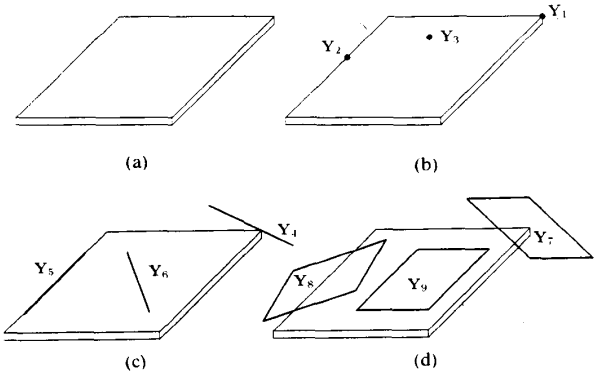


Fig. 11. (a) Simple object. (b) Point sensing of some object features. (c) Line sensing of some object features. (d) Plane sensing of some object features.

tures and sensory features, each will involve its own particular equations in generating the local pose distance function. The individual problems will be referred to using a two word descriptor, the first word referring to the type of sensory feature and the second referring to the type of object feature. For example, the *point-plane* problems represent those in which a sensor reports information about the coordinates of a point which lies on a planar surface of the object. Although this classification of sensing may seem rather artificial and restrictive, a wide variety of sensing methods are actually represented. The following paragraphs give examples of each of the sensing classes.

- *Point-Point Sensing*: The point-point problems are those in which the sensor reports an estimate of the 3-D coordinates of a point feature on an object. An example of this type of sensing is where stereo vision is used to locate an identifiable point on the object such as a vertex, small hole, or light. Knowing the position and orientation of the two cameras and the direction of the object point from each, the 3-D coordinates of the point can be computed.
- *Point-Line Sensing*: Point-line problems are used when the sensor can estimate the coordinates of a point which lies on an object line. An example of this type of sensing is given by the detection of an edge on an object (formed by the intersection of two planar surfaces) by a "point sensor" such as a proximity or touch sensor.
- *Point-Plane Sensing*: The point-plane problems occur when a sensor, such as a touch or range sensor, reports the coordinates of an arbitrary point on the surface of the object. Note that if a *known* point on the surface is sensed, then it would be better to use the point-point method, as that method places more restrictions on pose.
- *Line-Point Sensing*: Line-point problems are used when a sensor determines the parameters of a line which contains a point feature. An example of this type of sensing is given by the visual sensing of a point object feature. The image coordinates of the point, together with the position and orientation of the camera, can be used to determine the parameters of the line sensory feature. The image-to-world coordinate transformation described in Section II-C is a good example of this type of sensing.
- *Line-Line Sensing*: Line-line problems are used when a sensor is able to directly report an estimate of the parameters of a line object feature. An example of this type of sensing is given by the use of stereo vision to sense an edge of an object. Here the parameters of the line object feature can be computed from the depth information provided by the two images and the known positions of the two cameras [17].
- *Line-Plane Sensing*: Line-plane problems involve sensing in which an estimate can be made of the parameters of an arbitrary line which lies on a planar surface of the object. An example of this is where a sheet of light is projected on a surface, creating a light stripe, and this stripe is viewed by a camera. The parameters of the line can be computed from the intersection of two planes: one defined by the position and orientation of the camera, the location of the projection of the stripe on the camera image plane, and the camera model, and the other defined by the position and orientation of the source of the sheet of light.
- *Plane-Point Sensing*: With plane-point sensing a point object feature is determined to lie in a particular plane in space, as when a large planar surface on a robot makes contact with a vertex.
- *Plane-Line Sensing*: Plane-line problems are those in which a particular plane in space is known to contain a certain line object feature. An example of this is when lines on an object are observed with a visual sensor. Here, the sensory plane is defined by the projection of the object line on the image plane, the center of the lens (position of the pinhole using a pin-hole camera model), and the position and orientation of the camera.
- *Plane-Plane Sensing*: Finally, the plane-plane problems are used when a sensor is able to directly report the parameters of a plane object feature. This occurs, for example, when planar surface of the object is sampled several times by a touch or range sensor (e.g., a range image). Although each individual measurement could be treated as a separate point-plane problem, it would be simpler to fit a plane through these points and replace these multiple problems with a single plane-plane problem. The range image analysis presented in Section III corresponds to this type of sensing.

1) Parametrization of Sensory Features and the Sensing Models: The points, lines and planes of both the object features and the sensory features are parameterized using a point and a direction vector: point features are described by the point and an arbitrary unit vector, line

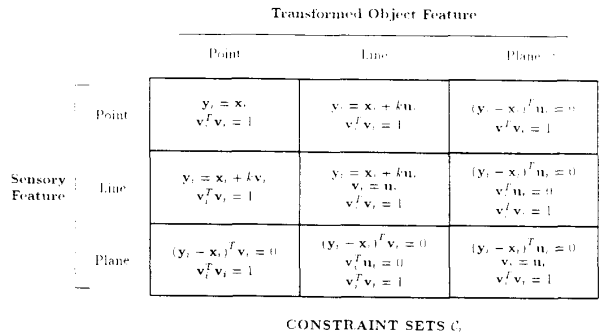


Fig. 12. Constraint relationships between sensory features and object features.

features are described by a point on the line and a unit vector in the direction of the line, and plane features are described by a point on the plane and a unit normal to the plane (either sense may be used). Each of these values is with respect to the world coordinate system. We therefore parameterize object features using $\boldsymbol{X}_i = (\boldsymbol{x}_i, \boldsymbol{u}_i)^T$ and sensory features using $\boldsymbol{Y}_i = (\boldsymbol{y}_i, \boldsymbol{v}_i)^T$, where $\boldsymbol{x}_i$ and $\boldsymbol{y}_i$ are the points and $\boldsymbol{u}_i$ and $\boldsymbol{v}_i$ are the unit vectors.

The sensing model for each class of sensing describes a geometric relationship between an object feature (in world coordinates) and its corresponding sensory feature (also in world coordinates). Each of these relationships will be written as a set of constraint equations involving the parameters of the corresponding pair of object and sensory features. If we consider point features to have a dimensionality of zero, line features to have a dimensionality of one, and plane features to have a dimensionality of two, then these constraints in general require the lower-dimensional feature to be contained within the higher-dimensional feature. For our parameterizations, the resulting constraint sets are given in Fig. 12. Each of the constraint sets shown in this figure also includes the constraint that $\boldsymbol{v}_i$ have unit length. (In our fusion method, the length of $\boldsymbol{u}_i$ will be defined externally to these constraint sets.)

2) Distance Functions for Sensory Features: The method of sensor fusion requires each sensor to report a distance function of the parameters of the sensory feature rather than specific values. The evaluation of this function for a given set of parameters produces a measure of how unlikely we estimate that parameter set to be. This provides a unified method of incorporating estimates of measurement errors into the problem. This subsection describes the distance functions we are using.

Each of our point, line, and plane sensory features is parameterized using a point, $\boldsymbol{y}_i$, and a unit vector, $\boldsymbol{v}_i$. For each sensory feature, regardless of its type, we use a distance function of the form

$$g_i(\boldsymbol{y}_i, \boldsymbol{v}_i) = (\boldsymbol{y}_i - \boldsymbol{\mu}_i)^T \boldsymbol{C}_i^{-1} (\boldsymbol{y}_i - \boldsymbol{\mu}_i) + \boldsymbol{v}_i^T \boldsymbol{M}_i \boldsymbol{v}_i \quad (13)$$

where the 3-D vector $\boldsymbol{\mu}_i$ and the 3 by 3 matrices, $\boldsymbol{C}_i^{-1}$ and $\boldsymbol{M}_i$, contain coefficients that are generated as a combination of the sensory measurement and its estimated errors. (We represent this first matrix by $\boldsymbol{C}_i^{-1}$ rather than by $\boldsymbol{C}_i$ because of its close association to the inverse covariance matrix used with multivariable Gaussian probability functions and with the Mahalanobis distance used in pattern recognition.) We require that $\boldsymbol{C}_i^{-1}$ and $\boldsymbol{M}_i$ be real, symmetric matrices. This means the eigenvalues of each will be real, and the set of eigenvectors for each will be orthogonal. We also require that $\boldsymbol{C}_i^{-1}$ be positive definite ($\boldsymbol{x}^T \boldsymbol{C}_i^{-1} \boldsymbol{x} > 0$ for all nonzero vectors $\boldsymbol{x}$) and that $\boldsymbol{M}_i$ be positive semidefinite ($\boldsymbol{x}^T \boldsymbol{M}_i \boldsymbol{x} \geq 0$ for all nonzero vectors $\boldsymbol{x}$, with the equality holding for at least some nonzero vectors $\boldsymbol{x}$).

The distance function given previously consists of a sum of two quadratic functions, one of $\boldsymbol{y}_i$ and the other of $\boldsymbol{v}_i$, each representing a "square error" function of its variables. The coefficients of these functions are generated along with each sensory measurement. They are chosen so that the minimum of the function is zero, the minimum occurs at the most likely values of the parameters, and the function increases in value quickest in the directions in which the estimated error is least. (A logical choice of these values would be to use the mean and covariance.) Further information on the generation of these coefficients is given in [40].

C. Multisensor Integration for Object Pose Determination: Experimental Results

A series of experiments was performed to demonstrate the ability of the fusion algorithm to determine object pose when given information from various combinations of sensors. The test panel (see Fig. 1) was placed in a known position and orientation and a number of sensory measurements were made using a human operator to provide the sensing strategy and feature matching. The data was then passed to the fusion algorithm one measurement at a time, and with each new measurement, the pose was computed and the errors in position (in inches) and orientation (in degrees) were recorded. To evaluate the capability of the algorithm of handling bad data, the initial measurement involved an improperly matched sensory feature.

In reporting our results, we use a two character descriptor (see Table I) to refer to each particular type of sensing. The first character is a digit which represents the class of sensing, and the second character is a letter indicating the particular sensor used. The results of the experiment are given in Table II. In this table, each row represents a run of the fusion algorithm with one new measurement added. For the added measurement, the value of its local pose function at the actual pose is given as an indication of how good that measurement is (note that these values are independent of the other measurements). Also given in each row are the errors in position and in orientation for the computed pose.

Note that the position and orientation errors given in the first few rows are almost meaningless, as the algo-

TABLE I
SUMMARY OF THE TYPES OF SENSING WE PERFORM AND THE DESCRIPTORS FOR EACH

Descriptor	Sensing Class	Sensor
$1p$	Point-point sensing	Proximity
$1t$	Point-point sensing	Touch
$1v$	Point-point sensing	Stereo vision
$2p$	Point-line sensing	Proximity
$2t$	Point-line sensing	Touch
$3p$	Point-plane sensing	Proximity
$3t$	Point-plane sensing	Touch
$4v$	Line-point sensing	Vision
$5v$	Line-line sensing	Stereo vision
$6p$	Line-plane sensing	Proximity
$6t$	Line-plane sensing	Touch
$7r$	Plane-point sensing	Range
$8v$	Plane-line sensing	Vision
$9p$	Plane-plane sensing	Proximity
$9t$	Plane-plane sensing	Touch

TABLE II
IMPROVING THE POSE ESTIMATE BY INCLUDING ADDITIONAL DATA GIVEN AN INITIAL BAD MEASUREMENT[a]

Number of Measurements	Data Added[b]	Local Pose Function Value[c]	Position Error (in)	Orientation Error (degrees)
1	$1v$*	321.0	7.14	168.0
2	$1t$	0.41	10.8	97.5
3	$2p$	0.11	4.27	156.0
4	$8v$	0.004	2.70	75.8
5	$9t$	0.48	2.52	39.1
6	$4v$	0.59	2.17	2.18
7	$6p$	0.002	2.16	2.06
8	$3t$	0.49	1.50	3.80
9	$5v$	0.35	1.49	3.73
10	$7r$	0.06	1.46	3.82
11	$3p$	0.10	1.46	3.82
12	$1v$	1.38	1.46	3.57
13	$6t$	0.004	0.98	3.24
14	$9p$	0.01	0.84	2.85
15	$2t$	1.10	0.78	1.37
16	$1p$	0.49	0.63	0.63

[a]Each row represents one run of the fusion algorithm and includes all data in the rows above it.
[b]The bad measurement is marked with an asterisk.
[c]This represents the evaluation of the local pose function at the actual object pose for the added data.

rithm does not yet have enough data to constrain the six degrees of freedom of object pose; the pose determined by the algorithm is largely dependent on the particular search path taken by the optimization algorithm.

In this experiment, the steady improvement in the solution with additional data is apparent. This is because the initial piece of bad data (marked with an asterisk) forced the early pose estimates to a value far from the actual pose, and each new piece of good data then places a pull on the solution point in approximately the same direction in pose space, the direction in which the actual pose lies. By the last measurement, the pose estimation is reasonably accurate as the good data has outweighed the bad to a large extent. If the data is unbiased (zero mean) and the actual pose was correctly determined, then the errors should average to zero as more measurements are taken. (In this experiment, the final error was close to the accuracy of the determined value of the actual pose [40].)

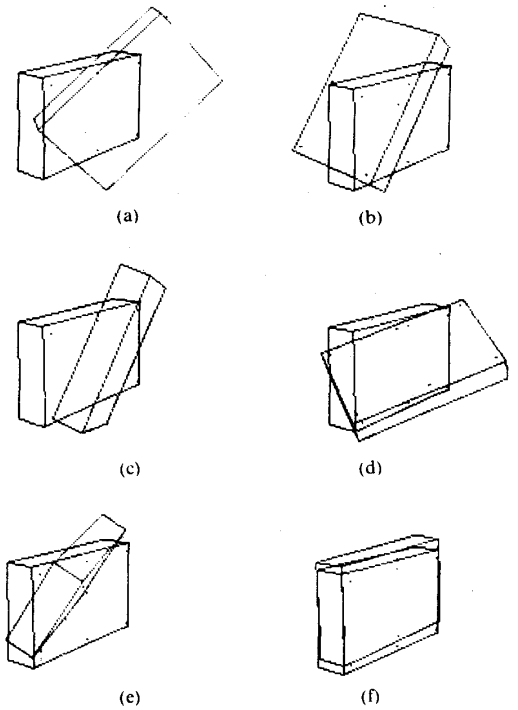


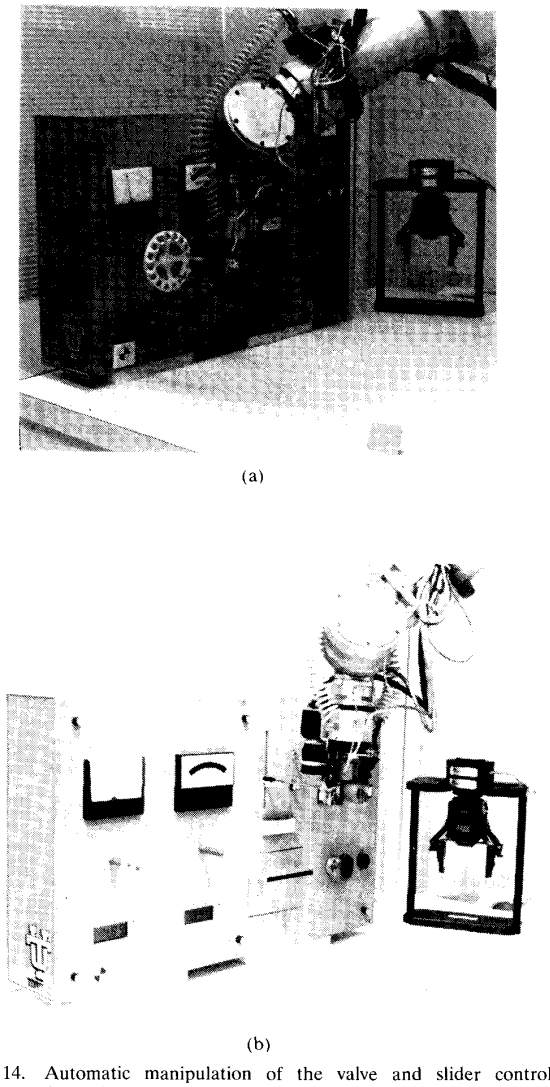
Fig. 13. Example line drawings of panel pose. These drawings show the actual and computed panel poses after each of the first six runs of the experiment.

In order to help the user visualize the pose computed by the fusion algorithm, the implementation included an option for creating line drawings of the panel in both its actual and estimated poses. In addition to displaying the edges of the panel, the six panel lights are also included to help distinguish between its front and back sides.

Fig. 13 shows six examples of these line drawings that correspond to the first six poses computed in our experiment. As can be seen in this figure, the poses computed after the first few measurements are far from the actual pose. This, of course, is due to the initial bad measurement and the small number of additional good measurements. It wasn't until the sixth measurement that the fusion algorithm had enough good data to arrive at a "reasonable" pose estimate in spite of the bad measurement.

V. CONCLUSION

Intelligent robots should be capable of acquiring and analyzing information from a variety of sensor modalities. The task of providing such capability to robots is quite complex and challenging. In this paper, we have described research studies directed towards some of the problems associated with the analysis of information acquired by vision and range sensors and through the combination of multi-sensory information. The sensory modalities of vi-



(a)

(b)

Fig. 14. Automatic manipulation of the valve and slider control mounted on the control panel. Complete sequence includes panel pose determination, object recognition, and localization, tool grasping, and object manipulation.

sion and range are selected since they provide useful descriptors of a typical work environment and involve some very important research issues. Specifically, we discussed the following. The vision information analysis includes a model-based object recognition module and an image-to-world coordinate transformation module to identify the 3-D coordinates of the recognized objects. The range information processing includes modules for preprocessing, segmentation, and 3-D primitive extraction. We have presented efficient algorithms for these range image analysis which are implemented on a parallel hypercube computer. The multi-sensory information integration approach represents sensory information in a sensor-independent form to formulate an optimization problem to find a minimum error solution to the integration problem.

The performance of various methods discussed in the paper, was evaluated by conducting a series of experimental studies. These laboratory studies utilized a testbed including a test panel and an industrial robot with multiple sensors. It involves the capability for object recognition, panel pose determination, object localization (in 3-D workspace), tool handling and trajectory planning and control. In Fig. 14 we illustrate the system's ability to manipulate the valve and slider control autonomously. Results of the studies are promising and will be useful in expanding our studies in sensor driven robotics. The time required to perform such tasks is also not excessive. In our non-optimized implementation, we can perform the complete sequence of operation, starting from initial image acquisition to object manipulation, in less than 3 minutes using a multiuser VAX 11/785 computer as the host. The paper discussed details associated with components which constitute parts of an overall multi-sensory robotic system with a specific application in focus. There are a number of research issues which need further investigations. Some of the important ones include, ability to directly analyze and utilize 3-D features in the vision system using a stereo approach, develop a matching module for range image analysis, and to extend and evaluate the multi-sensory integration approach to other robotic tasks.

Development of an autonomous system which can be deployed in an unstructured, complex and dynamic environment to perform useful tasks within acceptable bounds of accuracy, reliability, time, and cost is indeed a very challenging goal. We view our research as only a step towards the eventual field deployable autonomous system. We have attempted to address a number of issues, as model-based object recognition, image-to-world coordinate transforms and multi-sensory integration for a specific task domain. We have also attempted to evaluate the performance of the integrated system using a laboratory based robotic testbed. We recognize the limits of such a testbed in addressing the complexities of the real world work environment. We believe that laboratory based research such as the one reported in this paper provides discussion of relevant issues and solution approaches which can be useful.

Acknowledgment

B. Bernhard, C. Chen, S. Marapane, P. R. Mukund, and A. Perez helped in developing experimental set-up and also made other important contributions. We thank the CESAR group of the Oak Ridge National Laboratory and the Digital Signal Corporation for providing the range images. We would also like to thank the reviewers for their valuable suggestions for improving the quality of this manuscript. Mrs. Janet Smith assisted in the preparation of the manuscript.

REFERENCES

[1] C. Thorpe, M. H. Hebert, T. Kanade, and S. A. Shafer, "Vision and navigation for the Carnegie-Mellon Navlab," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 10, no. 3, pp. 363–373, May 1988.

[2] M. A. Turk, D. G. Morgenthaler, K. D. Gremban, and M. Marra, "VITS-A vision system for autonomous land vehicle navigation," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 10, no. 3, pp. 342–361, May 1988.

[3] D. Nitzan, "Three-dimensional vision structure for robot applications," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 10, no. 3, pp. 291–309, May 1988.

[4] J. P. Trevelyan, "Sensing and control for sheep-shearing robots," *IEEE Trans. Robotics Automat.*, vol. 5, no. 6, pp. 716–727, Dec. 1989.

[5] A. C. Kak and J. S. Albus, *Handbook of Industrial Robotics*, chapter on Sensors for Intelligent Robots, New York: Wiley, 1985, pp. 214–230.

[6] K. S. Fu, R. C. Gonzalez, and C. S. G. Lee, *Introduction to Robotics: Control, Sensing, Vision, and Intelligence*. New York: McGraw-Hill, 1987.

[7] R. Bajcsy, "Active perception," in *Proc. IEEE*, Aug. 1988, pp. 996–1005.

[8] K. M. Andress and A. C. Kak, "Evidence accumulation and flow of control in a hierarchical spatial reasoning system," *AI Mag.*, vol. 9, no. 2, pp. 75–94, Summer 1988.

[9] H. P. Moravec, "Sensor fusion in certainty grids for mobile robots," *AI Mag.*, vol. 9, no. 2, pp. 61–74, Summer 1988.

[10] P. K. Allen, "Object recognition using vision and touch," Ph.D. thesis, Univ. Pennsylvania, Philadelphia, PA, Sept. 1985.

[11] H. F. Durrant-Whyte, *Integration, Coordination, and Control of Multi-Sensor Robot Systems*. Boston: Kluwer, 1988.

[12] H. G. Barrow and J. M. Tenenbaum, "Computational vision," in *Proc. IEEE*, May 1981, pp. 572–595.

[13] T. O. Binford, "Survey of model-based image analysis systems," *Int. J. Robotics Res.*, vol. 1, no. 1, pp. 18–63, Spring 1982.

[14] R. T. Chin and C. R. Dyer, "Model-based recognition in robot vision," *Computing Surveys*, pp. 67–108, Mar. 1986.

[15] B. Bhanu, Ed., *Computer*, vol. 20, no. 8, Aug. 1987 (Special issue on CAD-Based Robot Vision).

[16] A. C. Kak, "Depth perception for robots," in *Handbook of Industrial Robotics*, S. Nof, Ed. New York: Wiley, 1986, pp. 272–319.

[17] S. B. Marapane and M. M. Trivedi, "On developing region-based stereo for robotic applications," *IEEE Trans. Syst. Man Cybern.*, vol. SMC-19, no. 6, Nov./Dec. 1989.

[18] M. M. Trivedi, C. Chen, and S. B. Marapane, "A vision system for robotic inspection and manipulation," *IEEE Computer*, vol. 22, no. 6, June 1989.

[19] M. M. Trivedi, S. B. Marapane, and C. Chen, "Automatic inspection of analog and digital meters in a robot vision system," in *Proc. Fourth Conf. Artificial Intell. for Space Appl.*, pp. 233–242, NASA, Nov. 1988.

[20] R. M. Haralick, "Using perspective transformation in scene analysis," *Computer Graphics and Image Processing*, vol. 13, pp. 191–221, 1980.

[21] I. Fukui, "TV image processing to determine the position of a robot vehicle," *Pattern Recog.*, vol. 14, nos. 1–6, pp. 101–109, 1981.

[22] M. R. Kabuka and A. E. Arenas, "Position verification of a mobile robot using standard pattern," *IEEE J. Robotics Automat.*, vol. RA-3, no. 6, pp. 505–516, 1987.

[23] K. Mandel and N. Duffie, "On-line compensation of mobile robot docking errors," *IEEE J. Robotics Automat.*, vol. RA-3, no. 6, pp. 591–598, 1987.

[24] M. A. Abidi and R. C. Gonzalez, "The use of multisensor data for robotic applications," *IEEE Trans. Robotics Automat.*, vol. 6, no. 2, pp. 159–177, Apr. 1990.

[25] P. J. Besl and R. C. Jain, "Segmentation through variable-order surface fitting," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 10, no. 2, pp. 33–80, Mar. 1988.

[26] B. Bhanu and L. A. Nuttall, "Recognition of 3-D objects in range images using a butterfly multiprocessor," *Pattern Recognition*, vol. 22, no. 1, pp. 49–64, 1989.

[27] P. J. Besl and R. C. Jain, "Three-dimensional object recognition," *Computing Surveys*, vol. 17, no. 1, pp. 75–145, Mar. 1985.

[28] R. Hoffman and A. Jain, "Segmentation and classification of range images," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 9, no. 5, pp. 608–620, 1987.

[29] A. Pérez, *Parallel Segmentation of Range Images on a Hypercube-Connected Distributed Computer*. Ph.D. thesis, Dept. Elect. Comput. Eng., Univ. Tennessee, 1989.

[30] A. Pérez, M. A. Abidi, and R. C. Gonzalez, "Parallels fitting of quadric patches for structural analysis of range images," in *Proc. Visual Communications and Image Processing II Conf.*, (Cambridge, MA), pp. 89–96, SPIE, Oct. 1987.

[31] J. S. Wiejak, H. Buxton, and B. F. Buxton, "Convolution with separable masks for image processing," *Computer Vision, Graphics and Image Processing*, vol. 32, pp. 279–290, 1985.

[32] B. K. P. Horn, *Robot Vision*. New York: McGraw Hill, 1986.

[33] P. Gemmar, *Intermediate-Level Image Processing*, chapter entitled "Considerations on parallel solutions for conventional image algorithms." New York: Academic, 1986.

[34] G. Stockman and S. Chen, "Detecting the pose of rigid objects: A comparison of paradigms," in *Proc. Optical and Digital Pattern Recognition, SPIE*, vol. 754, pp. 107–116, 1987.

[35] R. M. Bolle and D. B. Cooper, "On optimally combining pieces of information, with application to estimating 3-D complex-object position from range data," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-8, no. 5, pp. 619–638, Sept. 1986.

[36] O. D. Faugeras and M. Hebert, "The representation, recognition, and locating of 3-D objects," *The Int. J. Robotics Res.*, vol. 5, no. 3, pp. 27–52, Fall 1986.

[37] G. L. Bilbro and W. E. Snyder, "Linear estimation of object pose from local fits to segments," in *Int. Conf. Robotics Automat.*, (Raleigh, NC), pp. 1747–1752, Mar. 1987.

[38] W. E. L. Grimson and T. Lozano-Pérez, "Model-based recognition and localization from sparse range or tactile data," *The Int. J. Robotics Res.*, vol. 3, no. 3, pp. 3–35, 1984.

[39] R. C. Smith and P. Cheeseman, "On the representation and estimation of spatial uncertainty," Tech. Rep., SRI Int., September 1985.

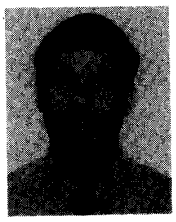
[40] R. O. Eason, "Determining the pose of three-dimensional objects by multisensor fusion." Ph.D. thesis, University of Tennessee, Knoxville, TN, Aug. 1988.

Mohan Manubhai Trivedi (S'76–M'79–SM'86), for a photograph and biography, please see page 1243 of this TRANSACTIONS.

Mongi A. Abidi (S'81–M'87) was at "L'Ecole Nationale d'Ingenieur de Tunis," Tunisia from 1975 to 1981. He received the Principal Engineer Diploma and the First Presidential Engineer Award in 1981. He received the M.S. in 1985 and Ph.D. in 1987 both in electrical engineering from the University of Tennessee, Knoxville.

In 1987, he joined the Department of Electrical and Computer Engineering at The University of Tennessee, Knoxville. His research and teaching interests include digital signal processing, image processing, data fusion, and robot sensing. He has published over 30 papers in these areas.

Dr. Abidi is a member of several honorary and professional societies including Tau Beta Pi, Phi Kappa Phi, and Eta Kappa Nu.

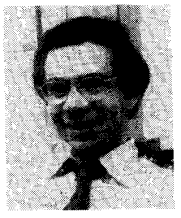


Richard O. Eason (S'77–M'79–S'80–M'80–S'82–S'85–M'85–M'86–S'86–M'88) received his BS, ME, and Ph.D. degrees in electrical engineering from the University of Tennessee, Knoxville in 1978, 1980, and 1988, respectively.

From 1980 to 1982 he worked as a VLSI design engineer at Zilog, Inc. in Cupertino, CA. While there, he designed Zilog's Z8581 Clock Generator and Controller. Since 1988, he has been an Assistant Professor of Electrical Engineering at the University of Maine, Orono. He

spent the summer of 1989 as an invited researcher at Kyushu Institute of Technology in Kitakyushu, Japan. His current research interests include multisensor fusion, computer vision, and intelligent robotics.

Dr. Eason is a member of Phi Kappa Phi, Tau Beta Pi, Eta Kappa Nu, and the IEEE Computer Society.



Ralph C. Gonzalez (S'65–M'70–SM'81–F'84) received the B.S.E.E. degree from the University of Miami, Miami, FL, in 1965 and the M.E. and Ph.D. degrees in electrical engineering from the University of Florida, Gainesville, in 1967 and 1970, respectively.

He has been affiliated with the GT&E Corporation, the Center for Information Research at the University of Florida, NASA, and is presently President of Perceptics Corporation and Distinguished Service Professor of Electrical and Computer Engineering at the University of Tennessee, Knoxville. Dr. Gonzalez is a frequent consultant to industry and government in the areas of pattern recognition, image processing and machine learning. He received the 1978 UTK Chancellor's Research Scholar Award, the 1980 Magnavox Engineering Professor Award, and the 1980 M.E. Brooks Distinguished Professor Award for his work in these fields. He was elected a Fellow of the IEEE in 1983 and was named Alumni Distinguished Service Professor at the University of Tennessee in 1984. He was awarded the University of Miami's Distinguished Alumnus Award in 1985, the 1987 IEEE Outstanding Engineer Award for Commercial Development in Tennessee, and the 1988 Albert Rose National Award for Excellence in Commercial Image Processing. He is also the recipient of the 1989 B. Otto Wheeley Award for Excellence in Technology Transfer and the 1989 Coopers and Lybrand Entrepreneur of the Year Award.

Dr. Gonzalez is co-author of the books *Pattern Recognition Principles*, *Digital Image Processing*, and *Syntactic Pattern Recognition: An Introduction* (Addison-Wesley), and is co-author of *Robotics: Control, Sensing, Vision and Intelligence* (McGraw Hill). He has been an Associate Editor for the IEEE TRANSACTIONS ON SYSTEMS, MAN AND CYBERNETICS and the *International Journal of Computer and Information Sciences*, and is a member of several professional and honorary societies, including Tau Beta Pi, Phi Kappa Phi, Eta Kappa Nu, and Sigma Xi.