To the Graduate Council:

I am submitting herewith a dissertation written by Hrishikesh P. Gadagkar entitled "Active Exploration of a Robotic Workcell Using Contact and Non-contact Sensors." I have examined the final copy of this dissertation for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Doctor of Philosophy, with a major in Electrical Engineering.

Mohan M. Trivedi, Major Professor

We have read this dissertation
and recommend its acceptance:

Accepted for the Council:

Associate Vice Chancellor
and Dean of The Graduate School

ACTIVE EXPLORATION OF A ROBOTIC WORKCELL USING CONTACT AND NON-CONTACT SENSORS

A Dissertation

Presented for the

Doctor of Philosophy

Degree

The University of Tennessee, Knoxville

Hrishikesh P. Gadagkar

May 1992

DEDICATION

This dissertation is dedicated to my father Mr. Pandurang R. Gadagkar and my late mother Mrs. Rukmini P. Gadagkar.

ACKNOWLEDGEMENTS

I would like to thank all of my teachers who have taught me since I began my educational career and motivated me to seek the best possible education. I express my sincere gratitude to Prof. Mohan M. Trivedi, who extended tremendous support, encouragement, and fruitful advice. Among others who made this work truly rewarding are my committee members, Dr. M. A. Abidi, Prof. D. W. Bouldin, Prof. S. J. Handel, Dr. R. V. Dubey, and I appreciate their patience and support. I would like to thank all of my fellow graduate students who at some time or another provided useful help, advice, and enjoyable discussion. In addition, I would like to thank Dr. C. Chen for assisting in developing an interface to display the results of this research through the simulator developed by him, Mr. Brian Bernhard, and Mr. Nils Lassiter for their technical support at various stages of this research.

I would like to thank the Department of Energy for supporting this research under the DOE's University Program in Robotics for Advanced Reactors through Contract No. DOE DE-FG02-86NE37968. Finally, I would like to thank my father, brother, and sister for their constant encouragement throughout my educational career. I also thank my wife Swati, for her support and particularly for her understanding and accommodating approach during the final phase of this research.

ABSTRACT

Robotic manipulatory systems are being developed with an ever increasing vigor for a variety of applications in hazardous or unstructured environments. Their successful utilization relies heavily on the use of sensory feedback mechanisms to guide their motions to perform a variety of tasks. The sensory feedback mechanism typically includes one or more sensors, such as vision, range, proximity, tactile, and force/torque. This research focuses on developing a robotic exploration system with emphasis on using both *contact* and *non–contact* sensors. Particularly, we intend to emphasize the distinctive qualities of haptic exploration that need to be exploited to perform a variety of robotic manipulatory tasks. When the robotic work environment is not conducive to acquiring vision, and range measurements, or tasks requiring grasping of an object, haptic exploration holds the potential for providing the necessary feedback to the robotic system to accomplish its tasks. This research presents a computational framework for an active exploration system that is characterized by a modular architecture, closed–loop control, incremental data integration scheme, flexibility, and graceful degradation. The exploration system involves acquisition, processing, and integration of sensory inputs acquired from vision, point laser range, tactile and force/torque sensors. For the purpose of active exploration, no a priori information associated with the composition of the workcell (e.g., information available from object or world models) is utilized. The exploration process results in the building of three–dimensional *Half–space* models of various polyhedral (convex and concave) objects encountered in the workspace. A set of sensor specific exploratory mechanisms has been designed to allow systematic and repeated use of sensory modalities for the acquisition of various descriptors. Subsequent

to the process of building object models using active exploration mechanisms, the object models allow extraction of essential information required for performing tasks such as object identification, grasping, and manipulation.

The experiments include convex, concave, thin planar rigid objects, and also compliant or non–rigid objects. The experiments bring out the advantages associated with having modular architecture, closed–loop control mechanism, incremental data integration, flexibility, and graceful degradation. The quantitative analysis of the experimental results show that the 3–D models are similar to the original object whenever it is possible to easily access the individual surfaces of the object. For thin planar objects, a composite tactile imprint is generated, instead of a 3–D half space model. The experiments also demonstrate the unique features of flexibility and graceful degradation.

TABLE OF CONTENTS

CHAPTER PAGE

CHAPTER PAGE

LIST OF TABLES

LIST OF FIGURES

FIGURE PAGE

FIGURE PAGE

CHAPTER 1

INTRODUCTION

Robotic manipulatory systems are proving to be increasingly effective and efficient for applications in unstructured and hazardous environments. The challenges posed in such complex environments necessitate the use of sophisticated sensory feedback mechanisms. In particular, for the robotic manipulatory systems to be able to handle substantially complex tasks, such as inspection and manipulation, the robotic systems have to employ multiple sensors [70]. Especially, when a task demands an object to be grasped or picked up, as in case of assembly, part handling, or object recognition the robotic system requires the ability to identify the position/orientation (*pose*) of the object and to generate a strategy for grasping or picking up the object. Such tasks invariably need repeated use of both *perception* (i.e., sensing and interpreting the sensed information) of the work environment and *motor* (i.e., motions associated with the robotic manipulator) functions to achieve the goals of the task. Throughout the description of this study certain terms will be utilized repeatedly, and it is necessary to define these terms from our research standpoint.

1. **Exploration:** The process of deriving an environmental model of the work cell.

2. **Perception:** The process of identifying and localizing an object using model–based paradigm [71].

3. **Manipulation:** The process of changing the state of one or more objects in the work cell, including that of the robot itself.

4. **Active Exploration:** The goal of active exploration is the same as for exploration, that is, to develop an environmental model of the work cell. However, the steps required to acquire sensory information are not preprogrammed. They are selected on the basis of the analysis of the sensory information already acquired.

5. **Active Perception:** Involves identifying and localizing objects in the work cell, where the sequence of operations associated with sensory data acquisition task are not preprogrammed but are based on the analysis of sensory information acquired so far.

It should be noted here that active exploration consists of a series of tasks directed towards developing an environmental model, where as, active perception consists of tasks that utilize object or environmental models to localize and identify objects in the workcell. Exploration requires the sensing and manipulatory activities to be employed repeatedly depending on the complexity of the task. This task of exploring a robot's work environment is essentially an "active" task requiring the physical motion of the manipulator (i.e., "*active body*"). At the same time, the sensed data during exploration is integrated either with certain a priori information, such as the robot's world model or previously acquired sensory information (i.e., "*active mind*"). This indicates that the two *active body* and *active mind* activities are tightly coupled and complement each other. Thus, it cannot be assumed that these two activities function independently.

In the past, researchers have studied issues associated with utilizing active exploration strategies in robotic manipulatory systems [41, 6, 67]. These studies addressed

only a small portion of the entire active exploration problem and were able to show to some degree that active exploration methodologies can indeed be utilized for model based object recognition types of tasks (i.e., active perception). However, to show a broad applicability of active exploration techniques for multi–sensory robotic application, it becomes necessary to have a perspective that encompasses various interrelated issues pertaining to the selection of sensory modality, data acquisition, data processing, and integration. The research discussed in this dissertation is focused on utilizing active exploration methodologies in robotic systems employing multiple sensors. The strength of this research is derived on the basis of conducting a series of experimental studies to obtain and to validate methodologies and algorithms constituting the *computational framework* and the *architecture* for multisensor based active exploration.

1.1 Motivation for the Study of an Active Exploration System

Object manipulation and assembly tasks require the robotic manipulator to grasp the object. In such cases the ability to acquire and process contact information becomes quite vital. This research effort was initiated to understand important issues underlying the acquisition, processing, and analysis of *contact* (or tactile) information that is normally required for the two tasks mentioned earlier. The initial phase was directed towards analyzing the sensing and processing requirements for tactile exploration and to evaluate its use as a pertinent sensing modality for robotic manipulatory systems. The review of tactile sensor technology [35] and tactile data processing methodologies [36, 33] clearly indicate that *haptic* (i.e., *cutaneous* and *kinesthetic*) exploration holds the ability to provide the vital information that is indeed required for object manipula-

tion and assembly tasks. Haptic exploration essentially combines both the *cutaneous*, or skin level, impulses and *kinesthetic*, or joint *pose*, and force impulses. The initial experimentation with tactile sensors clearly bring to light some of the distinctive advantages and disadvantages of utilizing haptic exploration.

- **Advantages of haptic exploration:** Haptic exploration is able to provide rich localized three dimensional (3–D) information so that object recognition, surface characterization, and contour following tasks could be accomplished quite easily [33]. Object recognition can be performed using either surface-based techniques (e.g., complex moment invariants) or shape-based techniques (e.g., Fourier descriptors). However, for shape recognition of objects that are larger than the sensor size, it becomes important to build a composite imprint (i.e., the generation of a larger imprint by combining several smaller imprints) in order to increase the field–of–view of the sensor [33]. Composite imprint generation based on the least–squared cross–correlation technique provides a sound technique of increasing the field–of–view of the sensor.

- **Disadvantages of haptic exploration:** First, the most important observation, however is that due to the lack of large field–of–view for the tactile sensors, the robotic system requires either an alternate sensing modality or some sort of world model to provide certain a priori information regarding object *pose*. This observation essentially implies that tactile sensors are not able to acquire all the required information without additional assistance from a secondary source. Second, the exploration process or the methods utilized to acquire various object features require a sound "computational framework", and utilization of ad hoc techniques

cannot ensure accuracy and repeatability. This observation arises from the fact that haptic exploration needs to be performed systematically in order to reduce the negative effects of limited field–of–view, low output resolution, and slow speed of exploration. One of the serious limitations is the slow speed of exploration, and to improve that, sensors that have better sensing rates and sensors that can be moved over the surface while maintaining continuous contact need to be designed and developed.

In short, additional sensing modalities are generally required to assist haptic exploration and to achieve improved performance. This additional facility coupled with a well defined computational framework would provide a means to efficiently implement haptic exploration for a variety of robotic applications. The eventual desire is to experimentally validate the premise that by using active exploration strategies, a robotic manipulatory system would be able to gain detailed information about the previously unknown composition and the constitution of its workspace.

1.2 Description of Active Exploration Tasks

The goal of active exploration is the same as that of exploration, that is, to develop an environmental model of the work cell. However, the steps required to acquire sensory information are not preprogrammed. They are selected on the basis of the analysis of the information already acquired. The exploration process needs to proceed in a systematic fashion, and generally, the approach adopted is determined based on an intuitive understanding of the entire process. However, the complexity of the work cell can be such that it would require the exploration system to verify previously acquired

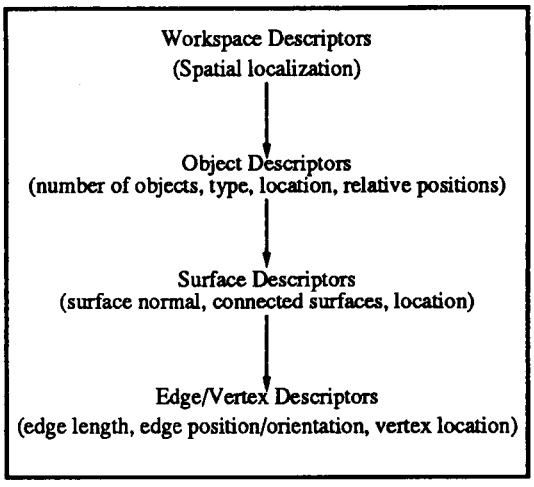


Figure 1.1: *A hierarchical workspace description technique is utilized to identify the descriptors required for constructing an object model using active exploration.*

information or acquire additional sensory information from a different perspective to strengthen the hypotheses based on the analysis of acquired information. The specific steps involved in the entire active exploration task need to be defined as the exploration proceeds. The research presented here is structured around utilizing a hierarchical approach, where the exploration begins with sensors that have a larger field–of–view and then gradually narrow down the focus of attention onto localized information. Figure 1.1 shows the workspace description technique, where descriptors have been identified at four different levels. The scheduling of the active exploration tasks is such that we need to extract the descriptors at the higher level before proceeding to the descriptors in the lower levels.

Active perception basically involves identifying and localizing objects in the work cell, where the sequence of operations associated with the sensory data acquisition task are not preprogrammed but are based on the analysis of sensory information acquired so

far. Note that in case of active exploration which requires us to completely explore the work cell of the robot, it is necessary to acquire *all* of the descriptors shown in Figure 1.1. However, for the purpose of identifying an object, only a sub–set of the descriptors shown in Figure 1.1 can be extracted and used for the matching phase to identify the object. Thus, exploration is lot more extensive in terms of the descriptors that need to be extracted to accomplish its task. By selecting exploration as the goal of the system, it becomes possible to utilize the same system for active perception without having to modify the system.

1.3 Research Scope and Objectives

Keeping the above observations in mind, we then need to focus on specific issues that will be emphasized in this research. As stated earlier, this research is motivated towards designing an exploration system that is not tailored to a specific robotic application. Having the "generality of design" in the focus, this research is going to address issues associated with the integrated system architecture, control mechanism, data integration, nature of sensory modalities, data acquisition and processing modules, sensor ordering, and system performance. The solutions provided here to various issues related to an active exploration system will be verified experimentally. In addition, an attempt will be made to bring out the difference between the approach discussed here and the approach utilized by researchers who have attempted to build similar exploration systems.

- **Architecture:** The architecture of a system indicates what the system is capable of achieving. It is thus essential to address the question of what the structure of the overall system architecture should be. This question can be answered by

reviewing the ultimate goals set out for the active exploration system. The constraints that would be dictating the eventual architecture are the ability of the system to be able to handle a variety of applications, the ability to allow the knowledge already acquired (or available from known models) to be utilized to guide subsequent acquisition and extraction of additional information, and the ability to add new sensors or remove existing sensory modalities. It is believed here that these features can be provided by developing an architecture that is modular and allows implementation of a "closed–loop" control mechanism.

- **"Closed–loop" Control Mechanism:** For an active exploration system, it is essential to have a means to monitor the performance of the system. This would allow the next set of actions to be determined based on how the system has been performing. In order to ensure such control over the system performance, it is essential to investigate whether the system should be set up to execute a predetermined set of sensing and manipulation actions independent of the composition of the workcell or implement a closed–loop control mechanism that can be tailored to suit the composition of the workcell. In our case, the active exploration is structured such that the task of choosing between available sensing modalities and of deciding the next feature or descriptor to be extracted are accomplished interactively by a *user*. Note that the interactive scheme is used to provide the link, which essentially *closes the loop* for the entire system. The user is required to monitor the success/failure of exact exploration task, and assist in making the selection of the next exploration task and an appropriate sensor. Note that the user is not required to interfere in the sensing and analysis tasks performed by the exploration system. Particularly for haptic exploration, the user typically plays

the role of verifying and refining the system performance. Such a closed–loop control would provide a means for repeating sensory and/or manipulatory activities as many times as they would be required.

- **Incremental Data Integration:** For a modular system design, it is necessary to identify how the information acquired by various modules (and sub–modules) will be put together to accomplish the eventual goal of the system. The choice here is to have a data integration scheme that is either incremental or non–incremental in nature. An incremental mechanism would imply that the integration process continuously updates the knowledge of the system as new information is made available. This updated information is then available to the control mechanism to decide the information that should be extracted next. For a non–incremental mechanism, the system would begin the integration process only after all the information that is believed to be necessary is indeed available. By having an incremental data integration mechanism, it would be possible to show *visually* that after each data acquisition and processing phase the exploration system would have a better "picture" of the robot's work cell.

- **Nature of Sensory Modalities:** The primary emphasis here is on showing the utility of haptic exploration capabilities in being able to characterize or reconstruct 3–D objects in the robot's work cell. Subsequent to the initial experiments with tactile sensors, certain shortcomings that seriously hamper the utility of tactile sensing for robotic applications are identified. These shortcomings arise due to the unavailability of certain a priori information, such as object *pose*, object surface orientation, etc. These shortcomings can be alleviated by utilizing additional

sensory modalities (preferably *non–contact*) that are capable of providing the necessary a priori information ensuring efficient haptic exploration. This brings us to the question of the nature of *non–contact* sensors that would allow the system to overcome the disadvantages associated with haptic exploration and that would also fit well into the overall objective of exploring the robot's work cell.

- **Modality Dependent Data Acquisition and Processing Modules:** The basic question at this point, then, concerns the determination of the best level of modularity of the entire system that would allow for maximum flexibility in the system. The modularity concept was emphasized earlier in the hypothesis associated with the system architecture. If the active exploration system is required to be inherently flexible in terms of being able to add a new sensing modality or remove an existing sensing modality, then we have to investigate the structure and design of sub–modules associated with data acquisition and processing for each individual modality. Generally, modules that are designed to utilize a single modality are easier to add or remove at any time.

- **"Ordering" of Contact and Non–contact Sensors:** This research intends to emphasize the use of haptic exploration and to study the extent to which haptic exploration can be utilized to solve real life problems. However, due to inherent strengths and weaknesses of *non–contact* and *contact* sensing, there exists certain levels of mutual dependence. This introduces the question of how to derive an explicit ordering among a diverse set of *non–contact* and *contact* sensors. In order to address this issue it is necessary to identify the features or descriptors that are best suited for *non–contact* sensing and those best suited for *contact* sensing.

- **Graceful Degradation:** In case of multi–sensory systems, it is often noticed that an object feature or descriptor information can be extracted from more than one sensory modality. Such redundancy can be quite useful since it helps to avoid a complete system failure when one of the sensors is unable to provide the necessary feedback due to reasons such as unfavorable environmental constraints. For example, if vision and range sensors are available to extract the number of objects in the scene, then when the local lighting conditions are unfavorable for vision, the system will continue to operate using the range sensor. It is also important to pay attention to the performance of these sensory modalities that are capable of extracting the same object feature or descriptor. It is quite likely that when the sensor with a better performance is rendered unusable, by utilizing an alternate sensing modality there will be some degradation in the overall performance, and this is generally referred to as "graceful degradation". Such a feature would help demonstrate that when one or more sensors are unavailable for use, the active exploration system would continue to function at a lower performance level as against encountering a complete system failure. Note that, for any active exploration system to produce a minimum amount of useful information, it is necessary to fulfill the minimum sensory requirements.

1.4 Computational Framework for Active Exploration

Several researchers in the past have implemented systems that were generally designed for model based object recognition tasks. The active exploration system is being designed with a premise that having the ability to demonstrate the task of building

a detailed 3–D description (3–D model) of an unknown object is relatively more challenging. The data processing and data integration requirements for exploration (i.e., 3–D model building) are more stringent than for object recognition tasks. The tasks involved in the object recognition process are indeed a subset of the tasks comprising the object model building process. Through experimental implementation, an attempt is being made to sufficiently address the design issues associated with the architecture and closed–loop control scheme.

The active exploration system will be implemented using a $T^3 - 726$ industrial robot, incorporating vision, point laser range, tactile, and force/torque sensors. The simulator for the entire robotic system exists on a Silicon Graphics 4D25TG machine, and the active exploration system incrementally updates this world model as more and more information is extracted from the robot's work environment. The structure of the overall systems is such that the sensory modalities and the descriptors that need to be extracted can be selected interactively as the exploration process proceeds. The unique quality of the testbed is that the data extracted by the above mentioned diverse sensing modalities can be integrated consistently to develop a reasonable model by utilizing both the autonomous and user guided operations.

After describing in detail the goal for the eventual exploration system, attention is now diverted toward a means for evaluating the overall performance of such an active exploration system. The performance evaluation of the active exploration system will be based on the **accuracy** of the final result. Keeping the emphasis on haptic exploration in focus, vision, point laser range (PLR), and force/torque sensors were selected to assist the tactile sensor based haptic exploration. Having highlighted the main experimental focus and the sensory modalities being utilized, it is then necessary to take a

closer look at the active exploration implementation details. This primarily includes the architectural design and the design of the closed–loop control scheme. The entire active exploration implementation task has been divided into a *primary task* and two distinct *secondary tasks*. From the point of view of the experimental verification of the active exploration system, the *primary task* is the most crucial and important one. However, the *secondary tasks* are required to make the entire implementation possible, and thus lesser importance is being attached to these.

1. The *primary task* consists of implementing the data acquisition, data processing, and integration modules.

    - **Data acquisition module:** This module consists of data acquisition procedures which provide specific object features or descriptors using sensed information acquired from a specific sensor. It then becomes essential to begin considering the sensing requirements which would include availability of certain a priori information and the need for the manipulator motion either before or during the data acquisition process.
    - **Data processing module:** Since there will be specific modules performing data acquisition for specific object features, a data processing module would be required that will consist of sub–modules corresponding to each sensory modality. These specialized sensor specific modules would have the knowledge as to which object features or descriptors can be extracted and correspondingly which data acquisition procedure should be utilized. Note that these sensor specific modules should be such that they do not communicate information among themselves.

- **Data integration module:** In order to have a closed–loop control, it is necessary to have a module that continuously monitors and updates the state of the exploration system. This module would be required to utilize the knowledge that is already available and to determine what additional information is required, and then, when the required information is available, the knowledge of the system needs to be updated further. This is generally called an "incremental data integration mechanism", and the data integration module would essentially perform each of these aforementioned functions.

2. The *secondary task* consists of designing and developing a sensor selection module and a 3–D Half–space modeler.

   - **Sensor selection module:** For the purpose of demonstrating the implementation of the entire system, the order in which sensory modalities would be invoked and the information that would be extracted from the selected sensory modality needs to be determined. We acknowledge that the task of developing an application dependent automatic sensor selection scheme would be fairly complex and continues to remain a challenging research problem. The research presented here is intended to utilize a fairly simple strategy for selecting the most suitable sensor at any point in time during the active exploration process.

   - **3–D Half–space modeler:** The objects that will be used for testing the performance of the exploration system will be polyhedral objects. Polyhedral objects can be modeled very easily using a Half–space modeler, where each surface of an object is represented by a 3–D plane or half–space. The

Half-space modeler should be designed such that an object model can be generated using sensor independent object information made available by the data integration module. This would provide a modeling scheme that does not impose any restrictions on the sensory modalities available to the active exploration system.

1.5 Major Contributions of this Research

This research presents a computational framework and architecture for active exploration using a multisensory robotic system. The architecture is modular and utilizes an closed-loop control scheme. The exploration system design was developed after performing a series of experiments designed to study the usefulness of *contact* sensing. Some of the haptic perception techniques implemented in the past borrow insights from human exploration mechanisms [67, 6]. Certain other techniques are based on mathematical models such as cost functions [27, 60, 65]. Some of the major weaknesses in these techniques have been identified, and this research provides alternative and more general schemes for the purpose of active exploration. In this section the specific issues that are quite unique and significant in the implementation of an active exploration system are being emphasized.

- **Architecture and Control Scheme**

  1. **Multisensory Integrated System:** This research describes an integrated system architecture that utilizes *non-contact* and *contact* sensors. The architecture is not only modular but also facilitates the implementation of an closed-loop control mechanism. Henderson *et al.* [41, 42] in their implemen-

tation of a multi–sensory system have alluded to the fact that a closed–loop architecture enforces better control over the execution of the overall system, thereby enhancing its efficiency. From the perspective of developing active exploration systems with an emphasis on utilizing haptic exploration, researchers in the past [68, 6] have restricted themselves to open–loop systems that were sufficient for addressing the object recognition problem. Through this research , by emphasizing closed–loop architecture, it has been possible to show that haptic exploration can be implemented in a highly efficient manner to go much beyond addressing just the object recognition problem.

2. **Incremental Data Integration Scheme:** An important consideration, which ensures robotic systems are able to function in unstructured and hazardous environments, is the design of their integration scheme. The level of intelligence of the integration scheme essentially decides the complexity of task that can be accomplished with substantial success by a robotic system. Ellis [27] and Allen [6] have attempted to highlight the importance of data driven haptic exploration approach. However, they have not been able to show how to utilize the data driven approach when there are multiple sensors involved. Henderson *et al.* [41, 42] have noted that to be able to improve the efficiency of their multi–sensory system they would indeed require an incremental integration scheme. The notion of incremental data integration and the data driven exploration approach are the key factors of this exploration system and do not coexist in most other integrated multi–sensory systems.

3. **Flexible Exploration System:** The two issues noted above provide an improvement over earlier attempts to design active exploration systems. One of

the distinctive flavors of the active exploration system design is the ability to reorder the exploration tasks to comply with the requirements of a different task. The system allows the sensors and the order in which they need to be used to be tailored based on the application at hand. Researchers in the past have always concentrated their efforts towards solving partial problems associated with active exploration systems and have not been able to fully address all the issues associated with the entire problem. The exploration system described in this research looks at the design of active exploration systems from a slightly different approach, where the emphasis is on the ability to choose between *non–contact* and *contact* sensors at any stage during the active exploration, thereby providing maximum flexibility to the exploration system.

4. **"Graceful Degradation":** In addition to flexibility of the exploration system, we introduce the concept of "graceful degradation". By "graceful degradation" it is implied that the system is able to tolerate (i.e., remain functional) when one of its sensory modules is unusable. In the past, most robotic systems have refrained from ensuring that the entire system continues to function at a reduced performance level when a certain sensory modality is unable to function due to unfavorable working conditions. The design of architecture and the control scheme is such that redundancy among a certain set of sensors is used to keep the system operational when one of the sensors is unusable due to unfavorable working conditions. This facility adds an additional component to the exploration system described here, and this has always remained alien to most of the present day multi–sensory robotic

systems.

- **Sensory Data Acquisition and Analysis**

  1. **Tactile and Force/Torque Data Acquisition and Analysis:** In order to be able to explore a variety of convex and concave polyhedral objects, the tactile data acquisition and data processing modules have to be designed so that they will retain their generality to a great extent. Further, the data acquisition and processing techniques are required to have a sound computational background, for instance, against relying on ad hoc intuitive procedures [67]. The composite imprint generation technique provides a means for increasing the tactile imprint size for objects that are larger than the sensor size. This technique provides a means for generating a 3–D representation of thin planar objects. The techniques utilized for extracting surface and contour information can handle surfaces that are arbitrarily oriented by first extracting the surface normal using the force/torque sensor and then using the tactile sensor to explore the surface.

  2. **Versatile Multimodal Point Laser Range System:** The addition of point laser range sensor enables the extraction of 3–D data in three distinct modes: The *point mode* is useful for extracting random point data; the *line mode* allows acquiring a linear profile with variable resolution; and the *image mode* is particularly important to acquire the proximity of 3–D images. The versatility of the system is in its ability to extract information from arbitrary *pose*; variable size scans, and variable spatial resolutions. The versatility and the wide variety of spectral and geometric information that can be extracted

from PLR data makes the entire active exploration system quite unique.

1.6 Organization of the Dissertation

The multi–sensory robotic system includes both *Non–contact* and *Contact* sensory modalities. Here the challenging features of an active exploration system are pointed out by providing a detailed description of the individual components of this task. The structure and design of various modules that would be required to accomplish active exploration tasks will be discussed in detail. A review of the research conducted over the past several years helps in identifying some of the drawbacks of earlier implementations of active exploration systems. In this dissertation we describe our approach towards implementing an active exploration system, which not only inherits some of the features from earlier attempts but also provides several additional features that have been overlooked by most of the earlier implementations.

Reviewing some of the earlier related research reveals the striking differences between similar research efforts in the past and the research goals described here. This review provided in chapter 2 is essential to highlight the contributions of this research. A detailed description of the architectural framework of the active exploration system is provided in chapter 3. Chapter 3 presents a detailed discussion of the structural and functional aspects of the individual modules constituting the overall active exploration framework. The specific details pertaining to each of the exploratory mechanisms are described in chapter 4. The experimental results will be presented in chapter 5 along with a discussion describing the reconfigurability and graceful degradation features of the exploration system. Finally, the summary and conclusion will be presented in chap-

ter 6. The future research directions will be discussed in chapter 7.

CHAPTER 2

OVERVIEW OF RELATED RESEARCH

The problems associated with tactile sensor based active perception for the purpose of object recognition, localization, and manipulation have been studied by several researchers. Particularly, for object recognition, several researchers have succeeded in combining tactile data with known object models. The problem here boils down to extracting appropriate object features from tactile data which are then matched with corresponding features from the object model domain. The availability of object models reduces the burden on elaborate tactile sensing. However, for generating a detailed object description, researchers have utilized information available from vision and tactile data. This section describes some of these efforts in detail in an attempt to bring out the inherent distinctions and the complexity of the problems associated with active perception and active exploration tasks.

This review of other related research is primarily categorized in four distinct groups. The first group provides an overview of the tactile sensor technology. This review is provided here to emphasize the efforts put forth by numerous researchers to provide a sensory modality that is quite different from the traditional *non–contact* sensory modalities often used in robotic applications. In the second group, the research dealing with the use of *active tactile perception* techniques is discussed. This category deals with the various attempts made in trying to combine tactile sensing with available object models to recognize 2–D and 3–D objects. The emphasis here was to reaffirm the hy-

pothesis that tactile data can prove to be a rich source of localized 3–D information. The third group deals with research efforts that added vision as an additional component to circumvent some of the drawbacks associated with tactile sensing. The research efforts from the second group led many researchers to perform experiments on implementing active perception techniques using vision, touch, and object models for object recognition and building 3–D object descriptions. The success of experiments in the second group verified the hypothesis that additional sensing modalities can provide the initial information that is required to commence haptic exploration. This led to the investigation of the kinds of additional sensing modalities that would be useful and how extensively they would be required to be used to assist haptic exploration. To probe into these issues, the fourth group deals with research attempts, where the emphasis is on multi–sensory robotic systems. The study discussed in the third group inclines more towards proposing the architectural framework for multi–sensory systems, which is equally important to the active exploration system that is presented here. Note that the architectural issues are not emphasized in any of the studies reported in either the second or third group.

2.1 Review of Tactile Sensor Technology for Robotic Applications

A tactile sensor is a collection of transducers that convert force signals into measurable electrical responses. These transducers are commonly known as *forcels (or taxels, or tactels).* A variety of devices, materials, and sensing techniques have been explored for use in developing these transducers. The primary factors influencing a tactile sensor's overall performance, ruggedness, size, and spatial resolution are the sensing (or

transducing) mechanism and the chosen sensing element. A number of survey articles present comparative discussions of tactile sensor technology [54, 26, 55, 1, 44, 38]. Most of the earlier reviewers have concentrated on reviewing tactile sensor technology and have not emphasized on the amount and type of processing required for using the acquired tactile information efficiently. In our review of the tactile sensor designs, an attempt was made to provide a framework that could be used to select the appropriate tactile sensor, based on the sensing technique and the performance indices of the sensor.

Most of the tactile sensors described in the literature can be categorized into five broad categories. This classification is primarily based on the methodology of transducing force into measured signals. For instance, force inputs can induce changes in capacitance, resistance, magnetic field, or optical signals.

1. Optoelectronic Tactile Sensors ([63, 77, 64, 55, 26, 49, 55, 26, 23, 12])

2. Capacitive Tactile Sensors ([16, 66, 21, 16, 29])

3. Piezoresistive Tactile Sensors ([69, 43, 58])

4. Magnetoinductive Tactile Sensors ([39, 22, 73])

5. Piezoelectric Tactile Sensors ([25])

A list of sensor features and performance indices is shown below, and it can be used for the evaluation of individual sensors and for comparing different sensors against a common scale. The evolution of this list has been greatly inspired by Harmon [40] and Vranish [73]. The ultimate selection of one sensor over the other is primarily dependent on the application at hand.

1. *Sensor Shape*

2. *Compliance*

3. *Spatial Resolution*

4. *Sensitivity and Dynamic Load Range*

5. *Response Time*

6. *Hysteresis and Linearity*

7. *Cross Talk*

8. *Tactile Data Output*

9. *Communication Interface*

Over the years each new design attempted to provide either additional desirable features or overcome some of the limitations of earlier designs [56, 61, 62, 10, 78, 76, 51, 72]. Research in tactile sensors has led to experimentation with various types of materials and new technologies. If optical technology has provided high resolution sensors, then use of VLSI and CMOS technology has enabled smaller designs and higher response rates. Although not many tactile sensors are commercially available, a conscious attempt is being made to develop tactile sensors that would be able to satiate the demands of robotic applications. In spite of a pressing need to design and develop tactile sensors that would allow robotic manipulations to be more dexterous, the underlying sensor technology still leaves much to be desired. The present state of tactile sensor technology is hindering the efficacy of tactile sensing in robotic applications. Optical fibers

and capacitive sensing methods have proven advantageous for both planar and finger shaped tactile sensors. However, neither methods have encountered much success in determining thermal properties and surface texture of objects in contact. Apparently, to provide all the sensing capabilities of human skin, it is extremely difficult to provide a straightforward engineering solution. An ideal tactile sensor can be conceptually visualized as a fusion of various sensing techniques that would collectively provide the outstanding sensing capabilities of human skin. It seems the combination of CMOS technology and optical fibers would allow such a sensor to be miniaturized, with high resolution and very small response times.

Tactile sensing technology is a relatively new technology which is yet to offer a practical and rugged commercial sensor. However, tactile sensing is highly critical for many robotic applications, and we shall see more practicable commercial tactile sensors made available in the not too distant future. It is expected that these sensors will be geared toward addressing the requirements of specific robotic tasks. In addition, these sensors will be more effective in a multisensor environment rather than as a stand alone system. Finally, we point out that in order to make an effective utilization of the tactile sensor technology in robotic applications, one also needs to examine the tactile data processing and analysis requirements.

2.2 Active Perception Using Tactile Sensor and Object Models

Active object perception using tactile data along with information from object model data–base has been studied by Cameron [17], Grimson *et al.* [37], Ellis [27], Schneiter [65] and Roberts [60]. Grimson *et al.* [37] designed a technique where corresponding to

each sensed position, a set of interpretations are generated based on the local constraints that are satisfied. Given a set of sensed positions and unit normals at these positions, there can be numerous interpretations that can associate these positions (or several positions) to the surfaces of known object models. The distance between each pair of positions, the range of angles between measured normals at individual positions, and the direction of the vector between two sensed positions are used as local constraints. The sensed information satisfying the local constraints is tested for consistency with the surface information available from the object model. If a rotation and translation transformation is determined, such that the sensed positions lie inside the object face and not just on the surface then, the interpretation is considered acceptable. The local constraints are used to prune an interpretation tree, and the knowledge available from the object model is then utilized to reduce the set of possible interpretations in order to recognize the object. The technique was successfully demonstrated for two object models. However, for a larger set of object models certain heuristics were needed in order to restrain the number of possible interpretations for each set of sensed positions. For instance, it is fairly simple to distinguish between large and small objects based on the maximum distance between any two sensed positions. In addition, Grimson *et al.* [37] have pointed out that the knowledge of angles between surfaces of objects can be another feature that can help in reducing the number of possible interpretations.

The general research theme of Schneiter [65], Ellis [27], and Roberts [60] was based on the work reported by Grimson *et al.* [37]. Ellis [27] was able to show that tactile data already acquired can be utilized to select a new path through free space for moving the tactile sensor in order to acquire additional tactile data. This additional tactile data would either uniquely determine the identity of the object or reduce the number of plau-

sible interpretations (candidate objects). The computed paths are designed to intersect an unexplored face of each of the candidate interpretations. The sensor traverses along a new computed path (based on existing tactile information), and new tactile data is acquired. If a unique face is identified corresponding to the newly acquired tactile information, then the identity of the object can be determined. Alternatively, if more than one face from the set of plausible interpretations fits the description from the newly acquired tactile data, then only a smaller number of interpretations survives. This process results in either identifying an object or reducing the number of interpretations at the end of tracing each new path. Ellis [27] describes this process as traversing an ambiguity tree, where the initial set of interpretations corresponds to the root node of the tree. Each new path results in either adding a terminal node (unique object) or a non–terminal node (fewer interpretations).

Roberts [60] combined the exploratory techniques used by Stansfield [3] and the techniques used by Ellis [27]. Roberts was able to show that for object recognition, the techniques of surface exploration [3] and optimization of sensor placement [27] could be used for object recognition using the approach outlined by Grimson *et al.* [37]. An additional constraint that defined the distance of the nearest approach between two edges was added to the set of constraints used by Grimson *et al.* Similar to Grimson *et al.* [37] an interpretation tree was traversed for matching sensed features with the available model features using a set of localized geometrical constraints. `Edge move`, `vertex move`, and `face move` were three primitive moves designed to explore an edge, a vertex, and a face respectively. Explicit rotation and translation transformations were not derived to match the data points with the model points. Instead, an interpretation tree would generate anticipated information for each of the interpretations based on

the specific move (edge, vertex, face) performed. Subsequent to the motion performed, the new information acquired would satisfy fewer interpretations and thereby reduce the set of possible interpretations. A cost factor is computed based on the information available from the models for each primitive move. Further, the efficiency of a primitive move is measured based on its inherent ability to reduce the set of interpretations. The next move selection strategy used by Roberts worked quite efficiently for 3–D polyhedral objects.

Schneiter [65] emphasized how to localize a 2–D object using perfect measurements of position and normals. Paths that guarantee distinguishing measurements (or optimal) and sub–optimal paths were shown to exist for a given set of interpretations. On the other hand Cameron [17] used a statistical decision theory to locate the next best position for acquiring new tactile information. A probabilistic membership function, representing the current knowledge of the environment, and a utility function, which provides a measure of the ability of the new sensing location to discriminate between the known objects and their possible locations, are utilized to make a Bayes decision regarding the next best sensing location. The approach in both cases [27, 17] is to identify the object with as few tactile measurements as possible. Such techniques that utilize the model library to provide useful feedback for the purpose of acquiring additional tactile data make active tactile sensing more efficient.

Reviewing these research efforts directed towards recognizing objects using tactile sensing and known object models, it can be noticed that the environments required in each of these studies was quite structured. That is, the *pose* of the object had to be fixed in some sense to enable the tactile sensing process to begin acquisition of the required information. The small field–of–view of typical tactile sensors influenced strongly the

size of the object that could be used in the experimentations. These earlier attempts fell short in their ability to show how tactile sensors could be used to extract a global view of a given object. These observations led to the formulation of the new hypothesis that additional sensing modalities that inherently possess a larger field–of–view would help in extracting a global view and the *pose* of an object, thus providing the necessary information to initiate haptic exploration in unstructured work environments.

2.3 Active Perception using Vision and Tactile Sensors

Due to the larger field–of–view provided by vision and the localized 3–D information that can be extracted easily using touch, the combination of vision and touch is quite fruitful. Visual data is typically used to provide the initial driving force for the subsequent tactile exploration. Further, object exploration can begin by utilizing the coarse information made available from vision. Although vision can provide a substantial amount of detailed object features, the emphasis here on using vision is only to provide assistance to the tactile exploration. Allen [68] and Stansfield [3] have successfully demonstrated the use of vision and touch for active exploration of planar and non–planar objects. The information acquired from vision and touch is then integrated for object recognition.

The task of object recognition requires various primitives and features of the object under investigation to be extracted systematically. In addition, these primitives and features need to be represented in a manner well suited for the subsequent process of matching with object models. Stansfield [67, 68] designed a haptic system that interacted with a stereo vision system to develop a hierarchical (frame–based) representation

by actively exploring an object. The vision system primarily guides the haptic system in the exploration task designed to extract certain object primitives and features. Initially, using the stereo images, various approach planes for an object for haptic perception are defined. Further, the visual images are processed to compute an edge–map and a region–map. The design of haptic exploration was based on the experimental observations from perceptual psychology studies of human haptic exploration system [50, 47, 48]. Lederman and Klatzky proposed that the human haptic exploration is characterized by well defined exploration techniques called as Exploration Procedures, and are used for extracting specific object information. Haptic exploration begins by exploring the known edges from the edge–based stereo analysis. The haptic system is mainly composed of two subsystems, the body expert and the part expert. The body expert consists of exploratory procedures for the extraction of object surface primitives, such as compliance, elasticity, and roughness. Further, features such as surface shape and contour (edge and corner) are extracted using specialized haptic exploratory procedures. The part expert extracts features and primitives associated with a sub–part of the entire object. For instance, when an object can be split into sub–parts, which inherently have distinct shapes and surface characteristics, the representation of the entire object using such sub–parts becomes simpler in a hierarchical system. The exploratory procedures (EP) describe a set of specialized hand movements designed to extract certain object properties, such as contours, texture, edges, etc. Thus a contour EP is an exploratory procedure to extract the outer contour of an object. This is accomplished by defining two other EPs, edge EP and corner EP. The knowledge of regions from the region–map facilitates exploration of uniform surfaces. Frames for each high–level feature (contour, extent, surface–patch, and sub–part) corresponding to the approach planes are generated with intermediate

(edges, corners, and surface shape) and low–level (compliance, elasticity, and roughness) haptic features as slots. The corresponding EPs provide the necessary information to fill in the slots of the frames. The frames for the entire object and sub-parts are made using the frames generated for the various approach planes. Objects such as a basket, box, cup, jar, sponge, tray, etc. were used to derive their hierarchical representation. The experimental results showed that the system was repeatable and quite robust.

Allen[3, 2, 4, 5] integrated vision and touch for object recognition, where the active tactile system is guided by vision. When a region (from visual input) is selected for exploring by touch, the sensor is aligned normal to the least squares plane obtained from the 3–D stereo points that form the contour of the region. The sensor is then moved towards the region along the normal, and depending on the tactile feedback the region can be classified as a hole, cavity, or a surface. The contour of a hole or cavity is traced out by moving the tactile sensor along the periphery. Surface descriptions (bicubic surface patches) of the object can be formed using such sparse sensed points. To construct bicubic surface patches (or Coon's patches), high curvature points along the contour are selected as knot points. The tangent and twist vectors at such knot points are then used to generate the surface patch. A single patch requires only 4 knot points; however, the surface description thus obtained will not be the best one. In order to generate the best fitting surface, additional knot points are selected at mid-points of successive knot points in either direction from the previous set of sensed points. Objects can be recognized by matching corresponding holes or cavities located and the generated surface descriptions. First, the transformations required for matching holes, cavities, and surface patches are computed. Using these transformations, location and approach axis of holes and cavities can be computed. These transformations can also be used

to match surface descriptions. The matching process yields multiple interpretations based on the perceptual information derived from vision and touch. Using the model information for the plausible interpretations, touch can serve as a tool for hypothesis verification. For instance, existence of a handle, extent (depth) of a cavity, etc. can be easily verified by touch. Objects, such as a plate, bowl, mug, pitcher, etc., were successfully recognized.

Addressing the suitability of the shape representation scheme for data acquired as a result of haptic exploration, Allen[6] has suggested the use of three distinct shape representation schemes along with their corresponding exploratory procedures. A four fingered articulated robotic hand was used to perform exploratory motions designed specifically for extracting tactile information, which is then transformed into a specific shape representation scheme. Superquadrics, Face–Edge–Vertex, and Generalized cylinders were selected to be the three shape representation schemes. The *grasping by containment* procedure allows a 3–D object to be grasped by all four fingers and obtain tactile information from more than one face of the object. Repeated grasping of this nature provides substantial information from all the faces of the object. The superquadric shape recovery process works quite well when the available data points are equally distributed on all the faces of the object. Superquadric shape representation provides volumetric information, and the shape recovery can be performed using sparse data. The *lateral extent* procedure was designed primarily to acquire extents of faces, locations of edges and vertices, surface normals, etc. The fitting of planes is performed on a set of sensed points, acquired from a pair of orthogonal directions, to extract the actual face information. The knowledge of intersecting planes is used to extract edge and vertex information. The *contour follower* procedure uses two fingers

to extract contour points along a known axis. Generalized cylinder representation primarily requires an axis of sweep and the cross–sectional area, which is being swept along the known axis. A two fingered grasp around an object and along an axis is used to provide data points along the known axis. The distance between the fingers at each grasped location is used as the diameter of a circle. This circle approximates the cross–section at a particular location lying orthogonal to the axis. When a number of points is acquired along the axis, the generalized cylinder representation can be constructed using the known axis and the fitted circles at each sensed location along the axis. The axis of the object was computed using edge–based stereo analysis. Although each of the representation schemes provide an approximate representation, the combination of these procedures can yield a very well defined representation scheme. For instance, the information available from superquadrics can be refined by performing additional lateral extent exploration or contour following.

It can be observed that over the past several years substantial effort has been made by researchers to utilize haptic information for tasks such as object recognition and object manipulation. The systems developed by Stansfield [68] and Allen [6] emphasize utilizing exploratory procedures that are derived from empirical observations from studies in human tactile exploration techniques. Also, as in case of Allen [3], the system developed by Stansfield [68] utilizes vision to provide certain specific object information, which is then used to assist haptic exploration. Note that the 3–D information available through edge–based stereo is generally insufficient to determine the orientation of individual object surfaces.

Both Stansfield and Allen assume to a large degree that the object being explored is placed such that it facilitates both vision and tactile sensing processes. In particu-

lar, no systematic attempt is made to determine before hand the orientation of object surface that will be explored using touch. Typically, the tactile sensing process requires continuous monitoring of tactile data to detect and correct unexpected errors due to improper sensor alignment or contact with the object surface. During contour following, the orientation of the tactile sensor relative to the contour is quite crucial. If there exists an error in aligning the sensor along a contour, then as the contour is followed the error gets magnified, leading to a situation where the contour moves totally out of the sensor's field–of–view. The other form of error is caused when there is a significant difference between the orientation of the tactile sensor (i.e., the sensitive portion) and the object surface that needs to be touched. In either case (i.e., aligning with contour or aligning with surface), the exploration system should not only be able to detect such errors, but should have the inherent capability of correcting the errors.

The approach utilized by Stansfield has no provision for any form of feedback from the integrated information to inform the exploration process, the nature of information that is yet to be acquired. In Allen's [3] case, only when all the sensory information has been integrated, it is then possible to identify the missing information. This missing information is generally hypothesized and subsequently verified if possible. This serious drawback arises due to the utilization of a fixed procedure for acquiring and integrating sensed information. There always exists a possibility that vision sensing may be required again to acquire a specific object feature. The systems developed by Stansfield and Allen do not provide a means for either allowing such additional sensing or for selecting the sensing modality at any given time during the process of exploration. Note that, there is no provision for either replacing an existing or adding a new sensory modality to the exploration system. Such facilities can be incorporated, provided the system

architecture has the provision of specifying the sensory modalities that need to be used and the order in which they would be used. This brings us to the issue of architectural requirements of an active exploration system to provide for flexibility.

2.4 Multi–sensory Robotic Systems

Henderson *et al.* [41, 42] have addressed some of the issues mentioned earlier in a slightly different approach. The emphasis in their research is on developing a well structured modular system. Henderson *et al.* define a group of logical sensors that are specialized to extract certain specific object information. The logical sensors are essentially computational units operating on raw sensory information to provide a specific object feature. The logical sensors can accept inputs from a set of possible sensory modalities and have the capability to switch between these sensors when required. The system consists of an interactive mechanism, where the user would specify which logical sensor needs to be used. The features extracted by the selected logical sensors are then integrated with the existing knowledge of the system. In their system they also provide a means for having a closed–loop structure, where the data integration module sends feedback commands to the logical sensors based on the state of the system. This feature is specifically meant for the logical sensors to switch to an alternate sensing modality when the inability of a certain sensor to provide the required information is detected. The option for users to select sensors allows the sensor ordering to be determined based on the state of the system.

The active exploration system described in this research borrows some of the interesting aspects of Henderson's logical sensor framework. The system developed by

Henderson *et al.* [41, 42] does not address various issues concerned with integration of multi–sensory information. Their major emphasis is on being able to acquire and process multi–sensory information. When multiple sensors are being utilized, the question of relating extracted information across diverse modalities requires considerable attention. For example, the edges or regions available from visual data need not be clearly identifiable from tactile data. This requires data available from each sensory modality to be transformed into a common frame of reference. An additional factor is that extraction of certain features is dependent on the availability of certain other features. That is, there are inter–dependencies between features , and the feature extraction process needs to follow an implicit order. For example, the position or location of an object needs to be extracted before its vertices can be located. Although Henderson *et al.* mention the inclusion of tactile sensors as one of the sensory modalities being used in their system, they are not motivated towards emphasizing the use of tactile exploration. The various data acquisition and processing modules (or *logical sensors*) were not designed baring in mind the stringent requirements of tactile exploration. The emphasis on haptic exploration of this research has required us to be conscious of the needs for tactile exploration associated with the data acquisition and processing tasks. In addition, issues such as error diagnosis and recovery have to be addressed while developing data acquisition and processing modules for *contact* sensors, which was not addressed by Henderson *et al.*.

CHAPTER 3

COMPUTATIONAL FRAMEWORK AND ARCHITECTURE OF MULTISENSORY ACTIVE EXPLORATION

The previous chapters discussed the role of multi–sensory active exploration for robotic applications and described the issues that would be emphasized as part of our overall research focus. This chapter presents a computational framework of the proposed multisensory active exploration framework. The design of our active exploration scheme borrows certain empirical observations from the study of human perceptual framework available from perceptual psychology [8, 28, 45], neurophysiology [74, 7], and tele–robotics studies. The active exploration system is modularized into five distinct modules, namely, the data acquisition module, sensor selection module, input systems module, data integration module, and the Half–space modeler. Figure 3.1 shows the overall structural layout of the active exploration scheme, and each of these modules serves specific needs of the active exploration system.

3.1 Sensor Selection Module

The sensor selection process is primarily based on the nature of the information that needs to be extracted from the robot's workspace. A hierarchical workspace description process is utilized to guide the exploration process, particularly to facilitate selection of appropriate sensors at different times during the process of exploration. The hierarchical workspace description process allows the identification of "descriptors" that need to be

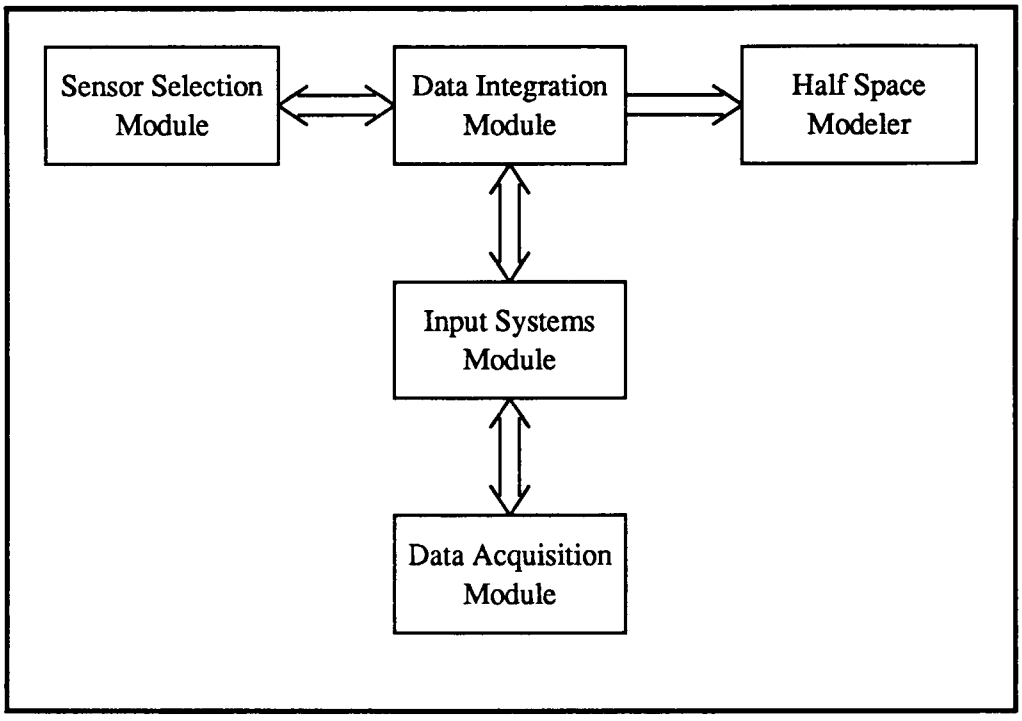


Figure 3.1: *Schematic layout of the active exploration system.*

acquired to promote a data–driven exploration. A *descriptor* is defined as any physical parameter that describes the structure of an entity, such as the workspace, an object, or any portion of an object. The descriptors required for building 3–D object models using the Half–space modeling scheme have been classified into four groups (Figure 3.2). The *Workspace Descriptors* describe the spatial localization and extent of the workspace. The *Object Descriptors* provide information such as the number of objects in the workspace, their type (i.e., convex or concave), location, and relative positions. The *Surface Descriptors* provide the surface normal, connected surfaces, and location corresponding to each surface of an object. The *Edge/Vertex Descriptor* provides the edge length, edge *pose*, and vertex location. It should be noted that the hierarchical

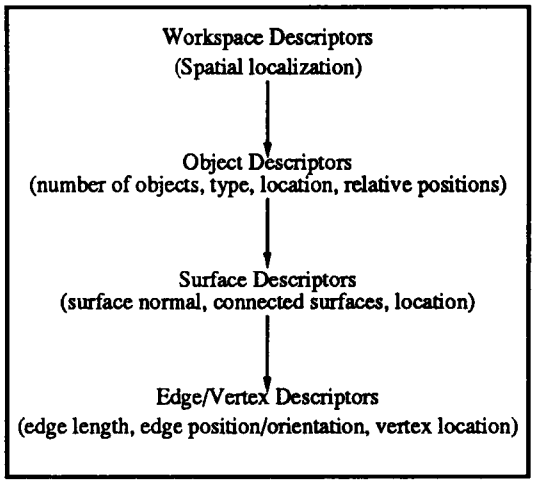


Figure 3.2: *The descriptors required for building 3-D object models using the Half-space modeling scheme have been classified into four groups.*

description process makes efficient utilization of the implicit ordering in the descriptor extraction process. For example, edge/vertex descriptors can be extracted provided the surface descriptors are available, which in turn require the object descriptors. Thus, the exploration process begins with acquiring and analyzing global information of the workspace and then proceeds towards acquiring more and more localized information.

3.1.1 Sensor selection process in the human perceptual mechanism

From a perceptual psychological perspective, Marks [53] explains that the human sensing modalities are characterized by intrinsic differences in their individual capacities to discriminate small variations in the stimulus. In addition, the information presented through individual modalities can be correlated and connected, or the information may be orthogonal and independent. Along a similar line of thought, it can be observed that in the active robotic work cell exploration task, information provided by sensors, such as PLR, tactile and force/torque, can be seen to be correlated and connected

to a certain degree since these sensors provide localized 3–D information of objects. Further, extraction of certain descriptors (e.g., number of objects, spatial location of objects, etc.) requires sensory modalities that can offer a large field-of-view and for certain other descriptors (e.g., surface normal, 3–D coordinates of vertices, etc.) there exists a need for sensors that acquire localized 3–D information. That is, we have sensors that provide information that can be seen as being "orthogonal and independent", as put forth by Marks. After having established the similarity in the functional features of the sensory modalities in the human perceptual system and the multi –sensory robotic system, the next step would be to study the criteria that are utilized by the human perceptual mechanism, while determining or selecting a sensor that will be used at any given point in time.

The process of attaching relative importance to sensory modalities is termed as *intersensory bias.* Intersensory bias in the case of the human perceptual mechanism was explained by Welch *et al.* [75], based on three distinct hypotheses, namely, *modality precision, directed attention,* and *modality appropriateness.* Welch *et al.* [75] anticipate these hypotheses to act complementary to one another rather than being mutually exclusive. According to the modality precision hypothesis, the sensing modality that is most precise in its ability to extract a given feature tends to bias the other modalities capable of extracting the same feature. For instance, in the human perceptual system, vision tends to dominate over the remaining sensing modalities in its ability to provide spatial location information. The directed attention hypothesis proposes that the amount of attention devoted to each of the sensing modalities, while attempting to extract the same object feature, determines dominating sensing modality. For example, the "seen position" might dominate the "felt position" if the directed attention towards

vision is greater than touch. Note that even though the "felt position" might be the correct one, the bias towards vision (with higher directed attention) leads the person to rely on the "seen position". The third hypothesis states that the modality that is most suitable for the extraction of a particular feature dominates the others. Each sensing modality is believed to extract certain features better than any other sensing modality, and this property according to the third hypothesis, is responsible for determining its appropriateness. Note that, in the human perceptual mechanism, all the sensory modalities are active at any given time, although, the directed attention towards each sensing modality is different.

These three hypotheses provide useful hints for designing an intelligent sensor selection scheme. This necessitates the identification of the strengths and weaknesses of the sensors being used in the robotic exploration system. This would eventually help us to determine the sensor appropriateness. Further, empirical results can be used to help us determine sensor precision. Note, the reason the sensor selection process becomes an extremely important task for active exploration is that no a priori information pertaining to the composition of the workcell is available. Further, the data acquisition process corresponding to each sensory modality requires certain "initial information" to begin its tasks. Here, the "initial information" refers to information such as a starting point on an object surface for haptic exploration, or for computing the surface normal using either PLR or Force/Torque sensors. Thus, the data–driven approach becomes quite challenging, and the information that is extracted at the current step strongly influences what can be extracted in the subsequent steps of exploration.

3.1.2 Designing a sensor selection scheme

In this section, a sensor selection scheme is described that can be used for robotic systems performing active exploration. Similar to modality appropriateness, modality precision, and directed attention hypotheses, *reliability, accuracy,* and *complexity* are the three *qualitative* measures that can provide a means to compare sensory modalities by the sensor selection module. Reliability is a measure used to indicate the ability of a sensory modality to successfully extract specific features each time they are required to be extracted from sensory inputs. Accuracy indicates the quality or precision of the extracted feature. Complexity reflects the combination of factors, such as the time involved in the sensing process and the computational effort. Note that time is proportional to the amount of sensed data that needs to be acquired. In addition, the computational effort implies the complexity of the algorithm used to extract the required feature, which in turn controls the computational time.

A set of *quantitative* measures corresponding to each of these qualitative measures would enable the sensor selection module to be able to select the most reliable, accurate, and least expensive sensor for any given task. This requires identification of various descriptors along with the most suitable sensor corresponding to each descriptor, required for the workspace exploration process. Note that for the development of our active exploration system, an *interactive* sensor selection scheme is utilized, instead of an *automated* sensor selection scheme. At any time during the active exploration process, the sensor selection is performed by a user based on empirically determined quantitative measures that are available to him/her. In the subsequent paragraphs the empirical technique utilized for determining quantitative measures is described in detail.

The entire set of descriptors has been categorized into *primary* and *secondary* de-

scriptors. *Primary* descriptors can be extracted using a sensing modality without depending on any other descriptor. For instance, the number of objects in a visual image can be extracted easily by thresholding the image and locating disjoint regions representing distinct objects in the scene. *Secondary* (or *Dependent*) descriptors require at least one other descriptor, which can be either a *primary* or *secondary* descriptor. For example, a descriptor such as the 3–D location of an object vertex requires prior knowledge about the surface of the object to which the particular vertex is associated with and the location of the object itself.

Let $\mathbf{s}$ be the vector representing the set of n sensors, and $\mathcal{D}$ be the vector representing the set of m descriptors that are required for object model generation.

$$\mathbf{s} = \begin{bmatrix} s_1 & s_2 \cdots s_n \end{bmatrix}^T \tag{3.1}$$

$$\mathcal{D} = \begin{bmatrix} d_1 & d_2 \cdots d_m \end{bmatrix}^T \tag{3.2}$$

Let us denote the $(m \times n)$ cost matrix $C_{\mathbf{d}}$, which indicates the cost of extracting each of the descriptors, $\{d_j \mid j = 1,\ m\}$, (assuming the corresponding *secondary* descriptors are already available) using the available sensors $\{s_I \mid I = 1,\ n\}$.

$$C_{\mathbf{d}} = \begin{bmatrix} C_{d_1}^{s_1} & C_{d_1}^{s_2} & \cdots & C_{d_1}^{s_n} \\ C_{d_2}^{s_1} & C_{d_2}^{s_2} & \cdots & C_{d_2}^{s_n} \\ \vdots & \vdots & & \vdots \\ C_{d_m}^{s_1} & C_{d_m}^{s_2} & \cdots & C_{d_m}^{s_n} \end{bmatrix} \tag{3.3}$$

$C_{d_j}^{s_I}$ implies the cost of extracting a descriptor d_j, independently using sensor s_I. It is essentially a combination of the three sensor selection quantitative measures associated with the process of extracting the descriptor d_j using the data provided by sensor s_I. Note that the corresponding cost element will be either zero or non–zero depending on whether sensor s_I is capable of extracting descriptor d_j

.

$\mathcal{D}_\mathbf{d}$ matrix $(m \times m)$ provides the descriptor dependency information, where each row k of $\mathcal{D}_d$ matrix indicates dependency of descriptor d_k on all of the other descriptors d_j, where $\{d_j \mid j = 1,\ m;\ j \neq k\}$.

$$\mathcal{D}_\mathbf{d} = \begin{bmatrix} D_{d_1}^{d_1} & D_{d_1}^{d_2} & \cdots & D_{d_1}^{d_m} \\ D_{d_2}^{d_1} & D_{d_2}^{d_2} & \cdots & D_{d_2}^{d_m} \\ \vdots & \vdots & & \vdots \\ D_{d_m}^{d_1} & D_{d_m}^{d_2} & \cdots & D_{d_m}^{d_m} \end{bmatrix} \tag{3.4}$$

Thus if descriptor d_k requires the knowledge of descriptor d_j, then the corresponding element $D_{d_k}^{d_j}$ will be set to d_j, else it will be set to zero. Note that the $\{D_{d_k}^{d_j} \mid j = k\}$ elements are always set to d_k.

Let vector $\mathbf{D}_d^k$ represent the *secondary* descriptors $\{d_j \mid j = 1,\ m;\}$ required for extracting a descriptor d_k (equation 3.5). Note that vector $\mathbf{D}_d^k$ is a subset of descriptor matrix $\mathcal{D}_\mathbf{d}$.

$$\mathbf{D}_d^k = \left[D_{d_k}^{d_j}\right]^T \tag{3.5}$$

The product of $\mathcal{D}_\mathbf{d}$ and $\mathcal{C}_\mathbf{d}$ yields the cost matrix $\mathcal{C}_{\mathbf{D}_d}$.

$$\mathcal{C}_{\mathbf{D}_d} = \mathcal{D}_\mathbf{d}\, \mathcal{C}_\mathbf{d} \tag{3.6}$$

The elements $C_{D_d}(I,j)$ of this new cost matrix denote the cost of extracting a descriptor d_I, along with each of its dependent descriptors given by the vector $\mathbf{D}_d^I$, using sensor s_j.

$$C_{D_d}(I,j) = \sum_{r=1}^{m} D_{d_I}^{d_r} C_{d_r}^{s_j} \tag{3.7}$$

Thus, the elements across each row i in matrix $C_{D_d}(I,j)$ represent the cost of extracting descriptor d_I and its dependent descriptors using sensors 1 through n respectively.

The set of descriptors that need to be extracted for the active exploration process discussed in this research is shown in Table 3.1. Recall that the hierarchical description of the workspace is quite useful since the exploration process does not utilize any known object models or world models. The descriptor acquisition process thus requires proceeding in a manner such that each new descriptor facilitates the subsequent acquisition of other descriptors. Table 3.2 shows the available sensors, and Table 3.3 shows the set of descriptors. For the set of descriptors shown in Table 3.3, Table 3.4 shows the descriptor dependency relationships (with the primary descriptors emphasized in bold).

The cost of extracting each descriptor along with any required dependent descriptors will enable us to select sensors based on the quantitative measures discussed earlier. Table 3.5 shows an example of a typical *normalized* cost matrix (where the zeros represent

Table 3.1: *Object descriptors and their corresponding sensing modalities.*

Object descriptors	Sensing modalities
Number of objects	Vision, PLR
Relative positions of objects	Vision, PLR
Position of an object	Tactile, PLR, Vision
3–D Bounding box	PLR, Vision
Object surface type	PLR, Tactile
Object compliance	Force/Torque
Surface normal	Force/Torque, PLR
Adjacent surface	Tactile, PLR
Edge position/orientation	Tactile
Edge length	Tactile
Vertex location	Tactile

Table 3.2: *List of sensors used in the active exploration system.*

Sensor Number	Sensor Name
s_1	Vision
s_2	PLR
s_3	Force/Torque
s_4	Tactile

Table 3.3: *List of descriptors used in the active exploration system.*

Descriptor Number	Descriptor Name
d_1	Number of objects
d_2	Relative positions of objects
d_3	Position of an object
d_4	3–D Bounding Box
d_5	Object surface type
d_6	Object compliance
d_7	Surface normal
d_8	Adjacent surfaces
d_9	3–D Edge position/orientation
d_{10}	Edge length
d_{11}	Vertex location

Table 3.4: *Descriptor dependency relationships.*

Primary and Secondary Descriptors	Descriptors Required										
	d_1	d_2	d_3	d_4	d_5	d_6	d_7	d_8	d_9	d_{10}	d_{11}
d_1	1	0	0	0	0	0	0	0	0	0	0
d_2	1	1	0	0	0	0	0	0	0	0	0
d_3	0	0	1	0	0	0	0	0	0	0	0
d_4	0	0	0	1	0	0	0	0	0	0	0
d_5	0	0	1	1	1	0	0	0	0	0	0
d_6	0	0	1	0	0	1	0	0	0	0	0
d_7	0	0	1	0	0	0	1	0	0	0	0
d_8	0	0	1	1	0	0	1	1	0	0	0
d_9	0	0	1	0	0	0	1	0	1	0	0
d_{10}	0	0	1	0	0	0	1	0	1	1	0
d_{11}	0	0	0	0	0	0	1	0	1	0	1

a sensor's inability to extract the corresponding descriptor) and the cost values can be empirically determined. An empirical method would require evaluating the qualitative measures (i.e., reliability, accuracy, and complexity) and then determining normalized values for the cost matrix. Here, initial values are set for the *maximum tolerable values* associated with each of the qualitative measures that are then used to normalize the empirically determined values. Note that values for reliability and accuracy are intuitively assigned. However, in order to determine a quantitative measure for complexity, the time involved in the sensing process and the time taken to extract (or compute) the descriptor are used. These cost measures do not take into account the cost involved in extracting the dependent descriptors.

Table 3.5: *Cost of extracting each descriptor using the available sensors.*

Sensors	Descriptors										
	d_1	d_2	d_3	d_4	d_5	d_6	d_7	d_8	d_9	d_{10}	d_{11}
s_1	1	3	3	6	0	0	0	0	0	0	0
s_2	6	7	7	2	5	0	3	6	0	0	0
s_3	0	0	0	0	0	2	4	0	0	0	0
s_4	0	0	0	0	4	0	0	4	2	4	5

3.2 Data Acquisition Module

The data acquisition module consists of sensor specific sub–modules that are responsible for extracting data from the corresponding sensor. These sub–modules function independently of each other, and this allows us to add or remove sensors without disturbing the operation of other sensory modalities. For the purpose of maintaining modularity in the system, a distinct coordinate frame is defined for each of the sensory modalities. The position and orientation of these coordinate frames is generally known with respect to the robot coordinate frame. The coordinate frames serve two important purposes. First, the motions associated with each of the sensors for the purpose of data acquisition are defined with respect to the sensor's coordinate frame. Second, the data acquired by each sensor is defined in terms of its own sensory coordinate frame. The *input systems* module performs the necessary transformations to have the extracted data represented with respect to the robot coordinate frame. Thus each data acquisition sub–module performs data acquisition associated with the sensor and data transformation to provide the data in terms of robot coordinate frame to its corresponding input systems module.

3.3 Input Systems Module

In the previous section the descriptors necessary for the 3–D model building process were identified along with the sensory modalities that can provide the respective descriptors. This section provides the structure of the computational modules responsible for interacting with their corresponding data acquisition modules to acquire necessary sensory information. Further, these modules extract the required descriptor from the sensed information and communicate the extracted information to the data integration module. Fodor [31] proposes that the human cognitive system consists of "*input systems*" that mediate between transducer outputs and central cognitive mechanisms by providing representations that are interpreted as characterizing the structure of entities of the world. He further explains that the *input systems* are restricted to a rigid domain and are responsible for transducing the information available from the sensory modality to which they are tied to as well as compiling the transduced information into representations that are accessible by a central cognitive system. Fodor believes that each *input system* is further sub–divided into task specific modules. For example, in the case of vision, mechanisms for analyzing color, shape, and spatial relationships are considered to be sub–divisions of the *input system* for vision. It appears that robotic systems employing multiple sensing modalities would benefit from Fodor's modularity concept.

Along a similar line of thought, the active exploration architecture design includes a data acquisition module, data integration module and the *input systems* module. Such a scheme allows the processing and analysis of sensed data provided by each modality to be independent of any other modality. The primary role of the *input systems*

module is to extract a given descriptor that is required for the data integration process. The *input systems* module consists of four input systems, one for each sensing modality. Each sensor specific input system is generally capable of extracting a set of descriptors. Associated with each such descriptor there exists a unique *Exploratory Mechanism* (EM). Each time when the data integration module requires a specific descriptor, the corresponding EM in the *input systems* module is activated to extract the required information. The tasks of each EM include,

(a) Invoke the appropriate data acquisition sub–module to acquire the necessary sensed data from the sensory modality to which the EM is associated with.

(b) Perform necessary manipulatory motions that are necessary either before or during the data acquisition process.

(c) Perform preprocessing (or filtering), transformation from sensor coordinate frame to the robot coordinate frame, and extraction of descriptor information.

(d) Decide whether additional sensing and manipulatory motion is required or terminate its process and report success or failure, depending on the extracted information.

The motor functions of the manipulator generally vary based on the particular EM. As stated earlier, each EM coordinates the sensing and manipulatory motions depending on the sensor to be used and the descriptor that will be extracted. Each EM will either have *alternating* or *simultaneous* sensing and manipulatory motion operations. For example, while extracting surface normal using PLR sensor, the `surface normal`

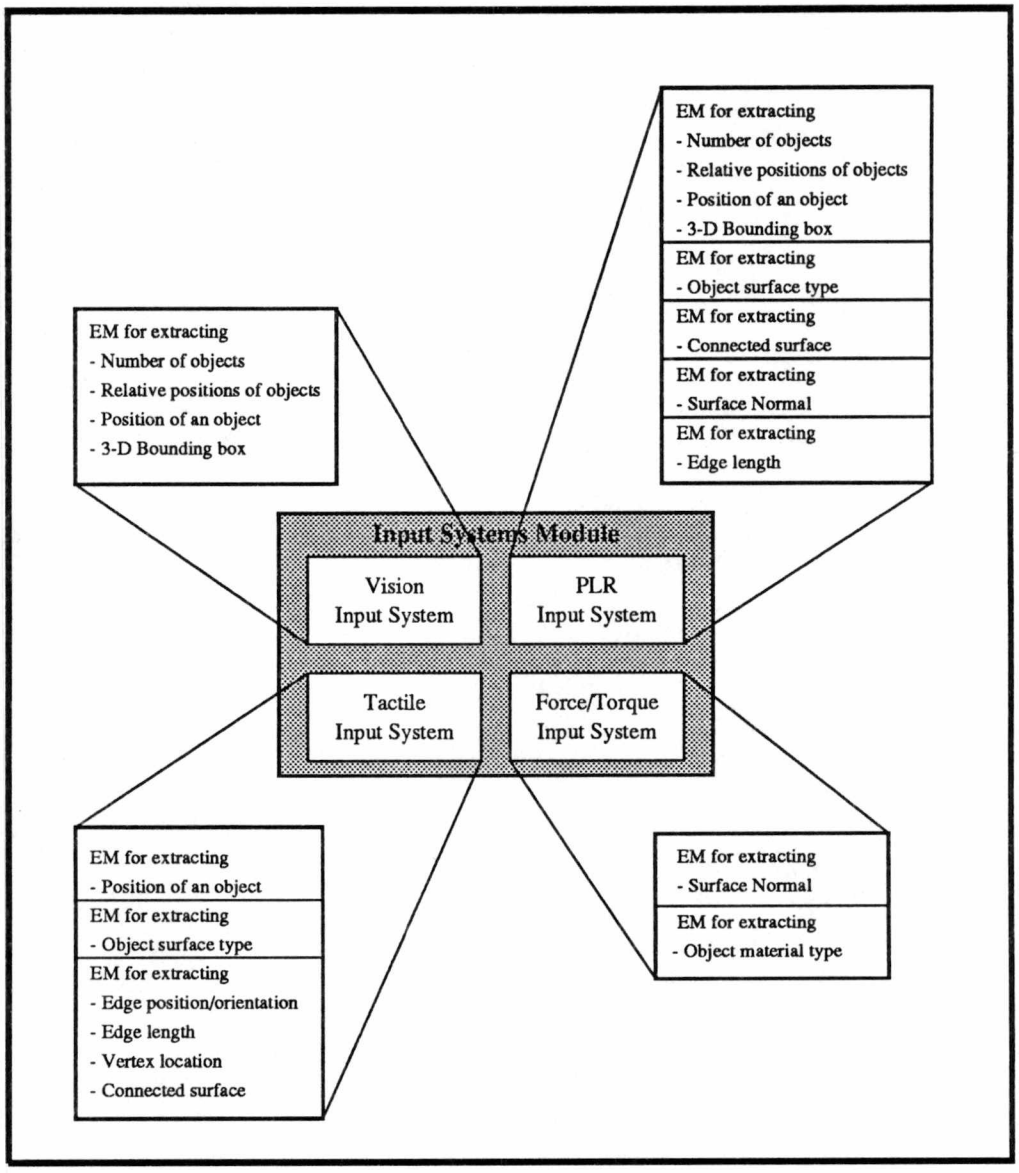


Figure 3.3: *The input systems module consists of sensor specific independent modules. Note that the input system corresponding to each sensor is capable of extracting certain specific descriptors.*

EM will move the sensor to the first location, which will be followed by the acquisition of a point measurement. The arm will then be moved to the second and then the third location to repeat the same set of actions. Here the manipulatory motion and the sensing action are alternating. However, while extracting the surface type using the PLR sensor, the `surface type` EM needs to have the manipulator to move along a specific linear trajectory while the sensor is continuously acquiring point samples. In such cases it would be necessary to have the manipulatory motion and sensing actions taking place simultaneously.

The EM not only consists of the procedural knowledge specifying the algorithm being used for extracting a given descriptor, but in addition, certain EMs need to monitor and correct, if necessary, the sensing process (i.e., alter the manipulator and sensor *pose*). For example, the exploration of an object contour requires continuous monitoring of the tactile sensor orientation with respect to the object contour. When the corresponding EM detects a sensor misalignment or a deviation from the actual object contour, the sensor is repositioned. The data processing task for each EM includes low–level processing or filtering operations such as cross–talk removal, random noise removal, etc. Following which the higher–level descriptor extraction is performed. Generally, depending on the descriptor required, various sensing modalities will impose distinct low–level processing and high–level processing requirements.

3.4 Data Integration Module

The data integration module controls the overall operation of the active exploration system. The data integration module is required to communicate with the sensor selec-

tion module when either a specified descriptor needs to be extracted or a descriptor is to be verified using an alternate sensing modality. Higher level representations passed on to the data integration module by the input systems module require transformation into a representation more akin to the input required by the Half–space modeler. The integration module is required to piece together such independent higher level representations in an incremental fashion as and when these representations are provided by the input systems module. The integrated information is then made available to the Half–space modeler to reconstruct the object model. The incremental approach acts like a feedback mechanism allowing the exploration process to be directed towards extracting information relevant to the particular application. Clearly, a closed–loop incremental data integration mechanism expedites the exploration process and augments its effectiveness in practical robotic applications. Applications that do not require detailed modeling (e.g., object recognition, grasping, etc.) would terminate the exploration process as soon as the requisite amount of information is available. Note that whenever accurate information for modeling an object is not already available, the integration module hypothesizes such information, and when necessary the hypothesized information can be easily verified with additional exploration.

Table 3.1 shows various sensing modalities, and the order in which they appear indicates their relative superiority in providing a particular descriptor. These descriptors were selected with a view that there should be some way of relating the sensed information provided by distinct sensing modalities with one another to aid in the incremental data integration process. The design of each EM is such that they require certain known information as "input" to guide their actions to provide the necessary descriptors as "output". For example, the EM for computing surface normal associated

with PLR sensor would require the descriptor information corresponding to the location of the object and the location of the particular surface of the object. These two descriptors will help determine the identity of the object (if there is more than one) and the identity of the surface of the object. Before invoking the input systems module, the data integration module ascertains that the necessary a priori information or "input" for the particular EM is available. Prior to the commencement of the active exploration process, the only information available to the system is the location and size of the workspace that requires exploration. Subsequently, the data integration module augments the knowledge of the system with more and more specific descriptor information. Such an exploration mechanism that more or less completely relies on sensory input to plan subsequent exploratory tasks is called a data–driven exploration. Note that for certain applications, the "input" required for a given EM can be made available from pre–existing sources such as object or world models, and in such cases the exploration is called a model–based exploration.

3.5 Half–space Modeler

The explored objects in the robotic work cell must be reconstructed to create a better understanding of their physical structure in 3–D. The reconstruction information is quite useful for tasks such as object recognition and object manipulation. For the active exploration task, it is assumed that the robotic work cell is composed of polyhedral objects. The active exploration task results in extracting object descriptors (Table 3.1) using both *non–contact* (vision and PLR) and *contact* (tactile and force/torque) sensors. The half–space modeling technique allows incremental and uniform integration of object

descriptors to build object models. This modeling technique does not require detailed and accurate object information. Although modeling techniques such as superquadric [6] and generalized cylinders [30] have been used predominantly for representing objects in robotic system applications, we believe that boundary representation schemes [15, 52] tend to be more favorable for multisensory systems. In addition, once an object representation is built for tasks such as object recognition and grasping, it is necessary to compute several geometric descriptors from the representation either to match with the sensed descriptors or to decide the appropriate grasping pattern [24]. The Half–space modeling technique can provide information pertaining to a variety of 3–D geometric descriptors, such as number of edges and vertices, *pose* of edges and surfaces, distance between surfaces, edges, and vertices, etc.

The reconstruction of polyhedral objects using the Half–space modeling technique requires the number of surfaces, individual surface equations, and surface adjacency relationships for each object. Uniqueness of the object reconstruction is ascertained when along with the number of surfaces and surface equation, the surface adjacency relationships for the object is also accurately known. Two surfaces are considered to be adjacent if they share a common edge. The surface adjacency relationship is a graph that typically specifies the set of adjacent surfaces for each individual object surface (Figure 3.4). Using the surface adjacency relationships, the individual vertices of a particular surface are computed.

The data integration process provides the surface equations of the object being explored by either extracting the surface normals or by hypothesizing the surface equations. The number of surfaces of an object are also extracted from sensory information. However, the extraction of surface adjacency information for each surface of an object

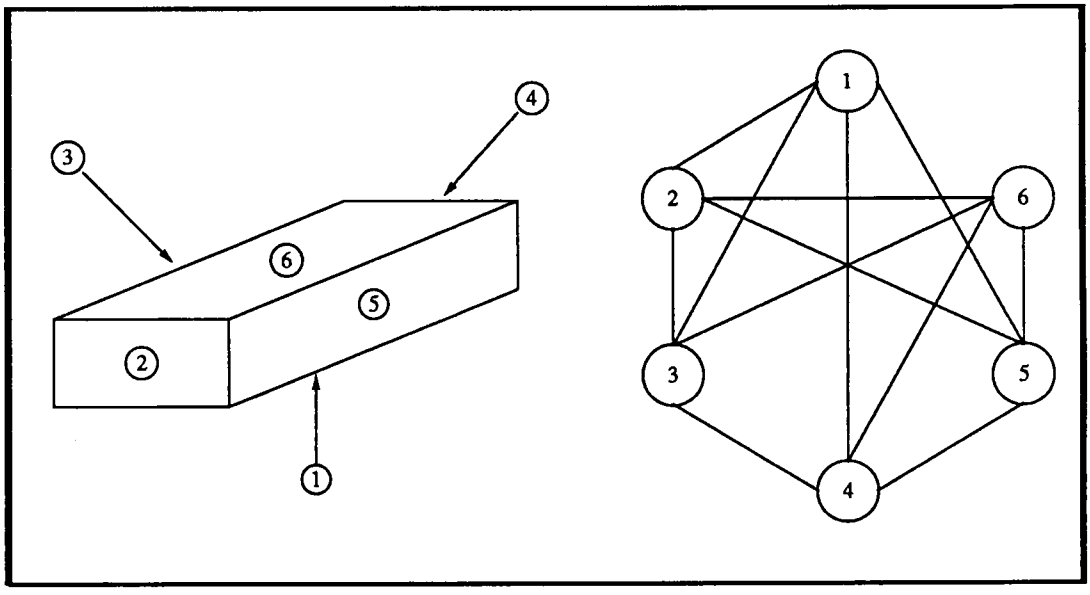


Figure 3.4: *The surface adjacency graph for a rectangular block indicates all the adjacent (connected) surfaces.*

requires extensive exploration. Alternatively, the surface adjacency information can be extracted by testing the intersection between various combinations of a set of three surfaces. Generally, three mutually adjacent surfaces intersect at a common vertex and such intersections can be computed by solving simultaneously the three surface equations. Consider three surfaces $\{S_1 \mid a_1x + b_1y + c_1z + d_1 = 0\}$, $\{S_2 \mid a_2x + b_2y + c_2z + d_2 = 0\}$, and $\{S_3 \mid a_3x + b_3y + c_3z + d_3 = 0\}$. Note that S_1, S_2, and S_3 are not known to be adjacent to each other. The common intersection point for these three planes then yields the common vertex belonging to each of the three planes. In order to solve the three equations of the surface simultaneously, the determinant of the matrix S would be required, where S is given by equation 3.8.

$$S = \begin{bmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ a_3 & b_3 & c_3 \end{bmatrix} \tag{3.8}$$

$$S = U\,W\,V \tag{3.9}$$

When the matrix S is singular, it implies that the three surfaces do not intersect at a unique point. The test for singularity is performed using the singular value decomposition (SVD) technique [57]. Using the SVD technique the matrix S is decomposed into U, V, and W matrices, satisfying the relation given by equation 3.9. The matrix W is a diagonal matrix, and if any diagonal element is *zero* or very close to *zero* then the matrix S can be assumed to be singular. Generally, computing various intersections between sets of three surfaces does not guarantee a unique object model. One way of achieving a unique solution is to specify a bounding box for the vertices. This would imply that the vertices that lie within the specified bounding box are retained as acceptable vertices of the object and the three surfaces are considered to be adjacent to each other. Thus

Table 3.6: *The table shows both geometrical and relational information that can be extracted from the model of an object.*

Geometrical and Relational features
Number of surfaces
Number of Edges
Number of Vertices
Distance between surfaces
Distance between vertices
Angle between surfaces
Relative orientation of edges

for each surface, the number of adjacent surfaces and the labels (identity of a surface) of the adjacent surfaces are utilized along with surface equations to reconstruct the object model. Note that the bounding box solution works quite well with convex objects; however, for concave objects, a unique solution is not guaranteed. In such cases, it is necessary to rely on sensed information to verify the existence of each of the computed vertices.

The data integration module provides the bounding box parameters for each object. The Half–space modeler begins with checking the surface adjacency relationships and determines if the information available is sufficient for the reconstruction task. This check involves verifying that with each identified surface there exists at least two additional adjacent surfaces. Once the individual vertices for each of the object surfaces are available, the object can be graphically reconstructed. Figure 3.5 shows Half–space models of objects generated using synthetic data. Table 3.6 shows the various geometrical and relational information that are extracted once the model of an object is built. Such information can be used to transform the Half–space model into any other form of geometrical model using a different representation scheme.

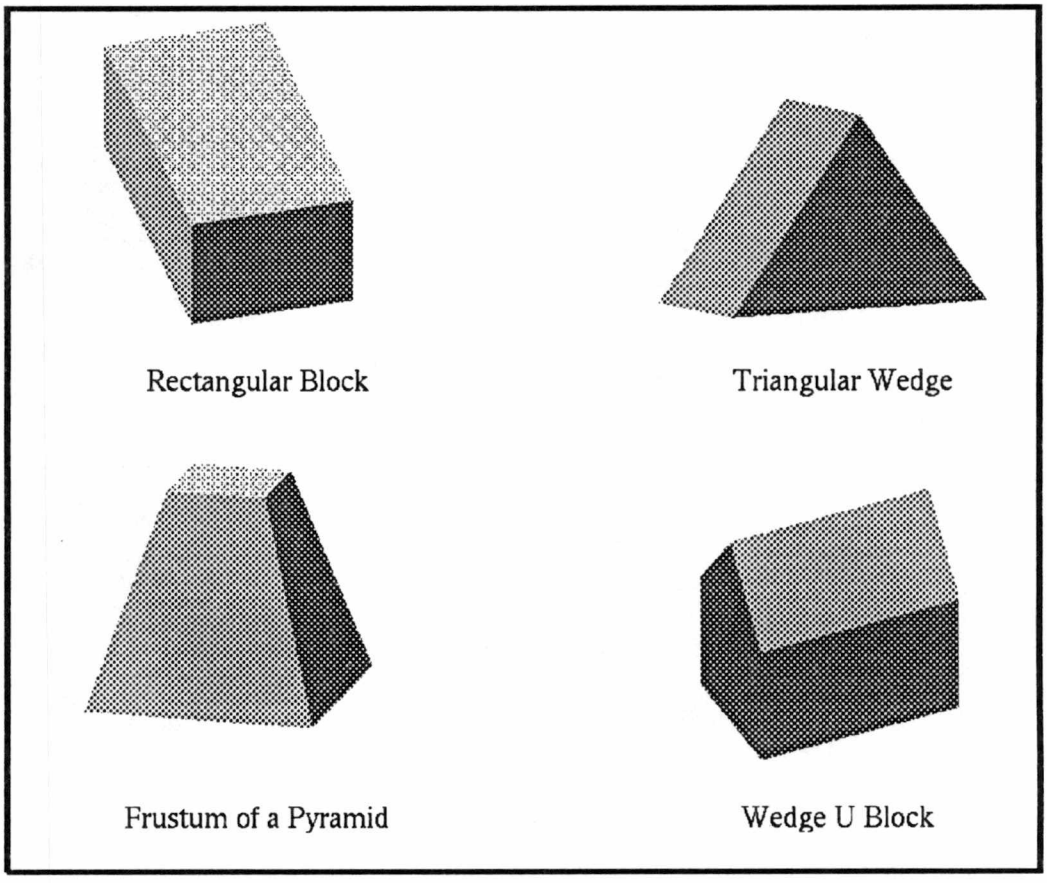


Figure 3.5: *Object model generation using the Half–space modeler. The modeling scheme preserves the actual physical size thus allowing distinction to be preserved between a small rectangular block from a large one.*

CHAPTER 4

DESCRIPTION OF DATA ACQUISITION AND INPUT SYSTEMS MODULES

The data acquisition and input systems modules need to interact with each other to a large extent in order to extract various object descriptors. The level of interaction varies depending on the complexity of the descriptor extraction process. The sensor specific EMs possess the necessary knowledge for extracting specific descriptors. The EMs also need to invoke the corresponding motor modules necessary for the data acquisition process. The EMs are also responsible for monitoring and controlling the descriptor extraction process. In this chapter, we elaborate on describing the specific details of each of the EMs that are designed to extract various descriptors required for the model building task. Note that the data acquisition module is composed of sensor specific data acquisition modules. In this section, we will be describing the algorithms used for extracting various descriptors, which in reality require the data acquisition and input systems module. Explicit description of data acquisition modules is discussed only where it is required.

4.1 Object Localization in Visual Images using Dynamic Thresholding

The hierarchical workspace description technique enforces an implicit ordering to the descriptor extraction process. We are thus required to begin with workspace descriptors and then to proceed to object, surface, and edge/vertex descriptors, in that order.

However, we do not intend to imply that each of the descriptors from the previous level in the hierarchy should be extracted before proceeding to extract descriptors in the next level. The initial view of the work cell is provided by vision, and before we proceed to explore individual objects in the work cell, we need to find the number of objects and their relative positions *wrt* the robot. The relative positions typically describe the relationships between the centroids of the object regions as they appear in the visual images. For the purpose of isolating individual objects in the visual image, we have developed a dynamic thresholding technique that can provide multiple thresholds for images with multimodal histograms.

Input: *Pose* for the camera to take the image, camera calibration information.

Output: Number of objects, approximate positions of objects,
relative positions of objects, and 2–D bounding box parameters.

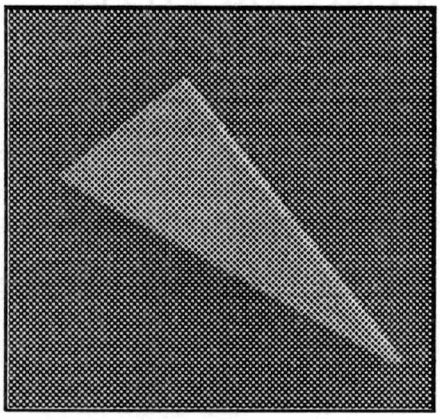
- *Step 1:* Smooth the original histogram using a 1×3 window.
- *Step 2:* Detect zero crossings from the first order derivative of the histogram.
- *Step 3:* Detect prominent peaks at negative going zero crossings and valley points at positive going zero crossings.
- *Step 4:* Detect valley points that are located between two peak points separated by a preset distance. Further, the difference in the height between the valley point and each peak should be larger than a preset value.

Using the computed thresholds, a binary image is generated, and each region corresponding to an object is extracted using connected component analysis. The centroids of these regions serve as approximate object positions. The position of the centroid is

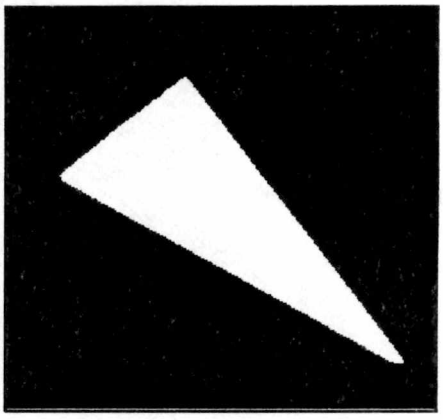
then transformed into the *robot frame* using the camera calibration process. Although, the histogram of the original image is smoothed, the original shape is still preserved, allowing easy detection of peaks and valleys (Figure 4.1(c)). For the example shown in Figure 4.1(a), the dynamic thresholding technique selects a single threshold value at 114 (Figure 4.1(c)). The thresholded image (Figure 4.1(b)) clearly extracts the object portion from its background. In case of multiple objects (Figure 4.2(a)), we sometimes require more than one threshold if individual objects have different colors, resulting in large differences in their gray levels. The thresholded image in Figure 4.2(b) was created using single threshold at 174 (Figure 4.2(c)), provided by the dynamic thresholding technique.

4.2 Results of High–level Planar Descriptor Extraction from PLR Data

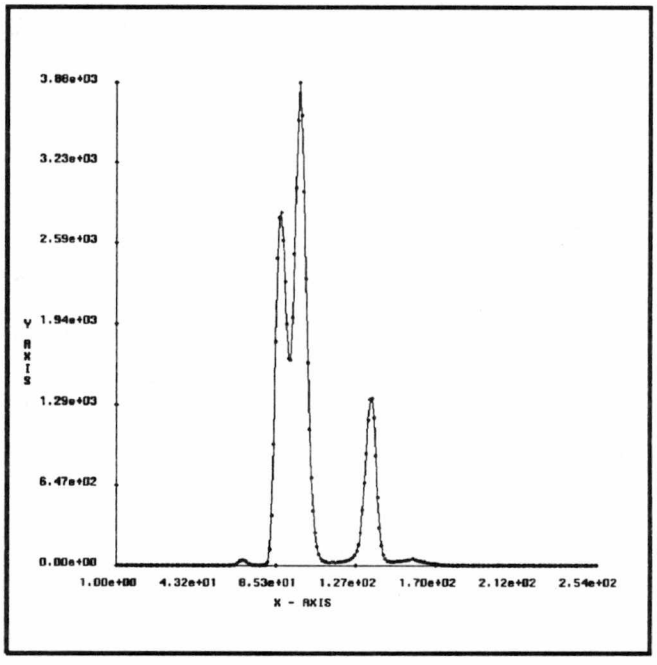
In the past couple of years, researchers have actively begun investigating point laser range (PLR) measuring devices that can be conveniently mounted on robotic end–effectors [59, 9, 34]. The PLR sensor being used in our active exploration system provides discrete point measurements, and when it is mounted on a robotic manipulator, it can be moved to any convenient *pose* to acquire range measurements. The PLR sensor has a dynamic range of 80mms with a standoff distance of 60mms [34]. With a small standoff distance, the PLR sensor allows the robotic manipulator to be quite close to the object being sensed. This allows the exploration process to focus on a single object at a time and not to be distracted by other neighboring objects. When the PLR sensor is mounted on a robotic manipulator, we can perform data acquisition in three distinct modes.



(a)

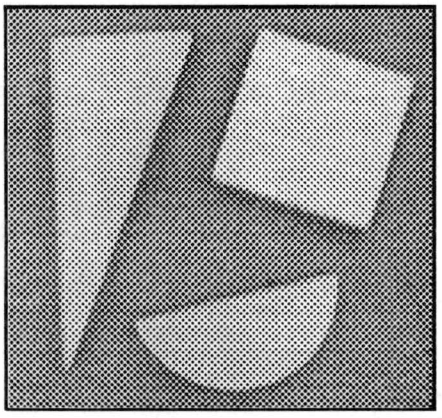


(b)

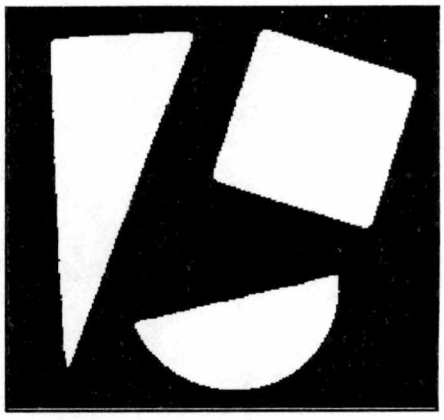


(c)

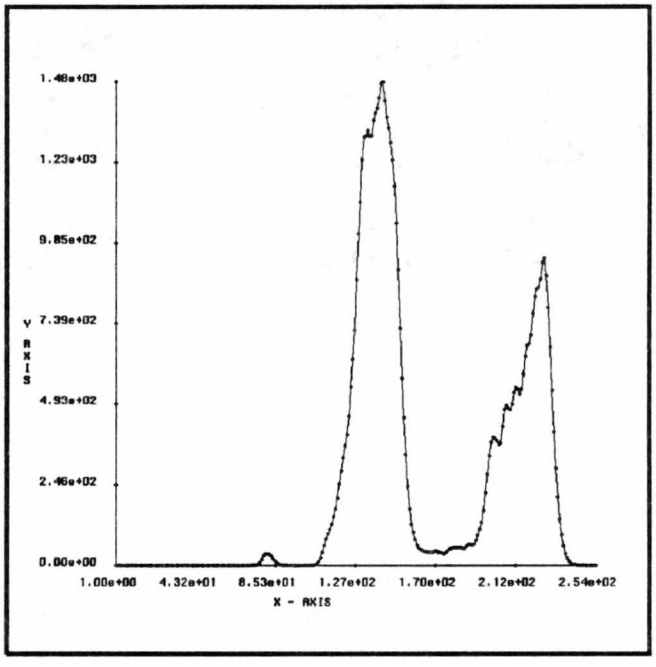
Figure 4.1: *The original visual image (a) is thresholded using a threshold selected at 114 (c) enabling extraction of the object from its background.*



(a)



(b)



(c)

Figure 4.2: *The original visual image (a) is thresholded using a threshold selected at 174 (c) enabling extraction of the object from its background.*

1. *Point:* If the measurements of a single point in the 3-D workspace is required.

2. *Line:* If measurement profile along a line in the 3-D workspace is required.

3. *Image (or Grid) :* If measurements associated with a 2-D grid are required. Sensors operating in this mode of operation are often called imaging, field–of–view, sensors [14].

These three sensing modes are applicable to a range sensor viewing the object from a *single*, selected perspective or from *multiple* perspectives. Single view provides characterization of a scene from a restricted perspective, and, therefore, only a portion of the object surface in the workspace can be sensed. Multiple views can provide a capability to derive a more complete 3-D description of an object. Robotic applications requiring complete 3–D object characterization need to use multiple sensors, viewing the object from different position and orientations. Obviously, the point mode operation provides the minimal depth information, whereas, the grid mode provides the most information. It can also be observed that the line mode operation subsumes the point mode, and the grid mode operation subsumes both line and point mode operations. This section focuses on different algorithms that have been implemented in order to extract high–level descriptors associated with planar surfaces. These include type of surface (planar or non–planar) from 2–D range data, surface normal using three–point measurement, and segmentation of range image into homogeneous regions.

4.2.1 Surface type computation using line fitting technique

Generally, surface fitting techniques are utilized to characterize various types of object surfaces encountered in range images. However, the problem of surface fitting becomes

fairly complex if the object is composed of a variety of surfaces. In such cases, the range image needs to be segmented into individual regions, which are then processed to compute the best fitting surface. A typical 2–D range profile (Figure 4.3) can be seen to reflect the nature of surfaces that are included in the scan. For instance, in the profile shown in Figure 4.3(a), the four linear segments indicate that the object is composed of four planar surfaces, while the profile in Figure 4.3(b) indicates that the object is composed of 6 planar surfaces with one curved surface in the middle. Although the 3–D physical extent of the individual surfaces (or size of each surface) appearing in the profile cannot be computed, the knowledge of the nature of individual surfaces itself is quite useful. Subsequent to the determination of the surface type, the exact orientation, spatial extent, etc. can be determined using 3–D range data.

An algorithm was developed to recursively fit lines (using minimum squared error (MSE) line fitting technique) to 2–D data points. The algorithm begins fitting a line for the first two points and then proceeds to test and add subsequent points in the 2–D scan. Following each addition of a new point, the new slope and intercept for the current line are recomputed. Whenever a large change in line orientation is encountered the current line is terminated and the computation of slope and intercept for a new line begins.

Input: Initial and final *pose* for the PLR sensor to specify the 2–D profile.

Output: Approximate number of surface located in the scan, surface type for each surface located.

- *Step 1:* Compute slope and intercept for the current line using two consecutive points of the line profile using the MSE technique. If all the points in the scan have been tested, then go to *Step 5.*

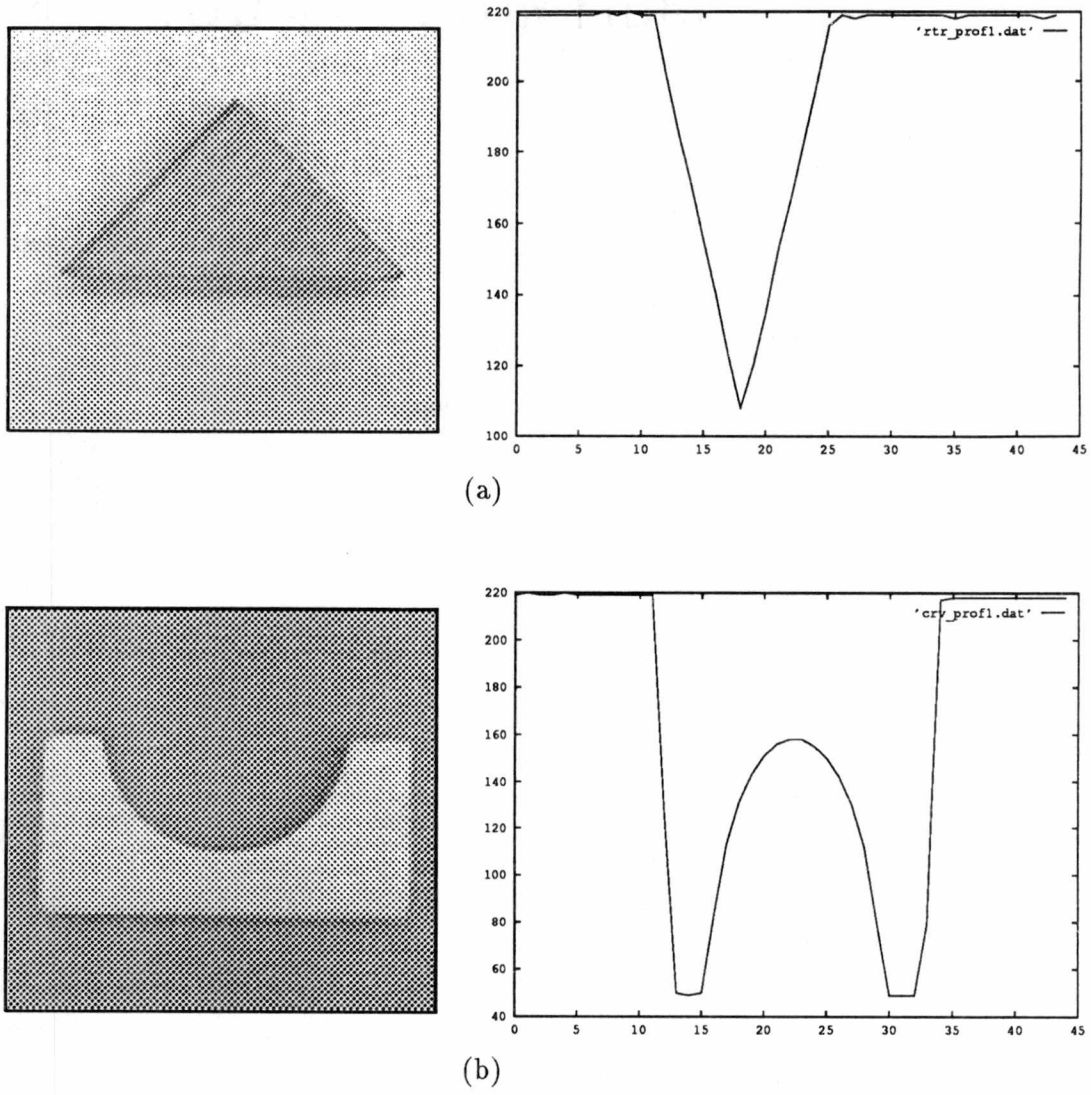


Figure 4.3: *The 2–D range measurements for (a) Wedge (triangular surface) and (b) Roman arch (semi–circular concave surface) show the output linearity for the PLR sensor which is quite essential in order to distinguish between linear and curved portions in the scanned profile.*

- *Step 2:* Compute the new slope and intercept (for the current line) by including the next point in the profile using MSE technique.

- *Step 3:* If the change in slope and intercept does not exceed a threshold, then the slope and intercept for the current line are updated to the new values computed in *Step 2*, and go to *Step 2*. Else go to *Step 4*.

- *Step 4:* Terminate the current line and begin computation of slope and intercept for a new line. Go to *Step 1*.

- *Step 5:* Eliminate lines consisting of less than four scan points.

- *Step 6:* Group successive "small" line segments with change in slope lesser than a predetermined threshold as being part of a single curved surface.

The algorithm sequentially tests each point in the line profile to verify if it should be included in the current line or if it should be used to commence a new line. The thresholds for slope and intercept are chosen such that small deviations due to random noise in the data does not result in terminating a line unnecessarily. The robustness of this algorithm was verified at the lowest and highest possible spatial resolutions of the scanning system. In order to acquire a 2–D profile, the robotic manipulator is required to move along a predetermined path. Note that the determination of sensor *pose* for 2–D scanning can itself be quite complex. We utilize 2–D or 3–D bounding box parameters for determining the initial and final location for the 2–D scan with the sensor orientation maintained along the robot's negative Z–axis.

Figure 4.3 shows two objects along with the 2–D profiles. Note that the 2–D profiles are inversely proportional to the shape of the original object. For example, the object

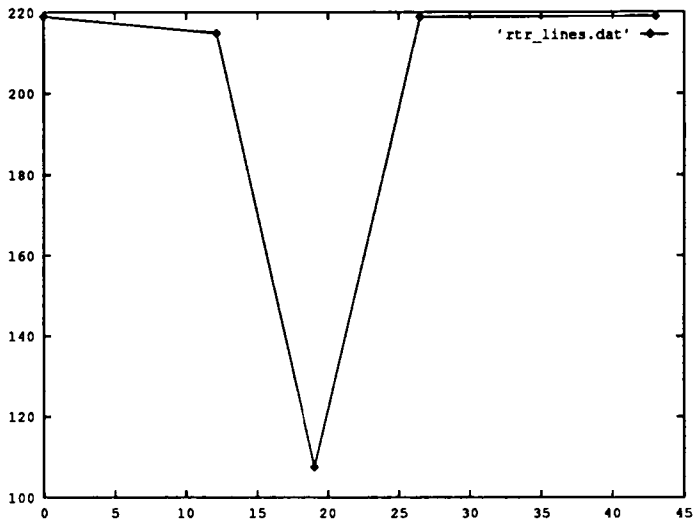
shown in Figure 4.3 (b) has a concave surface in the middle, whereas the corresponding scan shows a convex curve. The size of the two objects is such that the entire height of the object is within the dynamic range of the PLR sensor. Figure 4.4 shows the results of implementing the algorithm on the line profiles of objects in Figure 4.3. The results clearly indicate that linear portions of the profile were detected consistently and the non–linear (curved) portions are broken into numerous smaller linear segments. Note that the rate of change of slope over the curved portion is a function of the curvature which determines the number of line segments. It can be observed that each linear segment extracted using the above algorithm corresponds to a planar surface of the object.

4.2.2 Hypothesize number of surfaces of an object

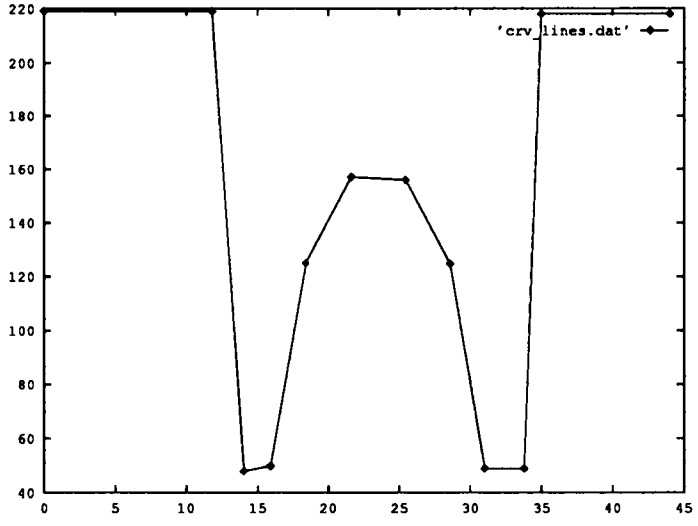
In the previous section we presented an algorithm that is being used to compute the surface type based on a single 2–D range scan data. The process, when repeated over the entire extent of the object, can yield the total number of surfaces of an object. In our active exploration process, we are interested in generating hypotheses and then attempting to verify the hypotheses at a later time using additional sensed data. Note that we intend to produce an approximate value rather than an exact value for the number surfaces because the time taken to acquire high resolution scans is quite large and also the data processing requirements will be quite severe. Thus, we resort to coarse scanning, which is less time consuming.

Input: 2–D bounding box of the object, PLR sensor *pose* for scanning.

Output: Approximation of total number of object surfaces.



(a)



(b)

Figure 4.4: *Line fitting technique extracts single line segments corresponding to planar surfaces and multiple line segments corresponding to curved surfaces for (a) Wedge (triangular surface) and (b) Roman arch (semi-circular concave surface).*

- *Step 1:* Compute the number of scan lines required along the X–axis of *robot frame*, with distance between each scan line along Y–axis being 0.25in.

- *Step 2:* Acquire and process each of the 2–D scans. For each scan line determine the number of surfaces located. Compute the histogram to indicate how many times a certain number of surfaces was located. Select the value for number of surfaces that was found the maximum number of times. In case of a tie, select the smaller of the two for estimating the number of surfaces.

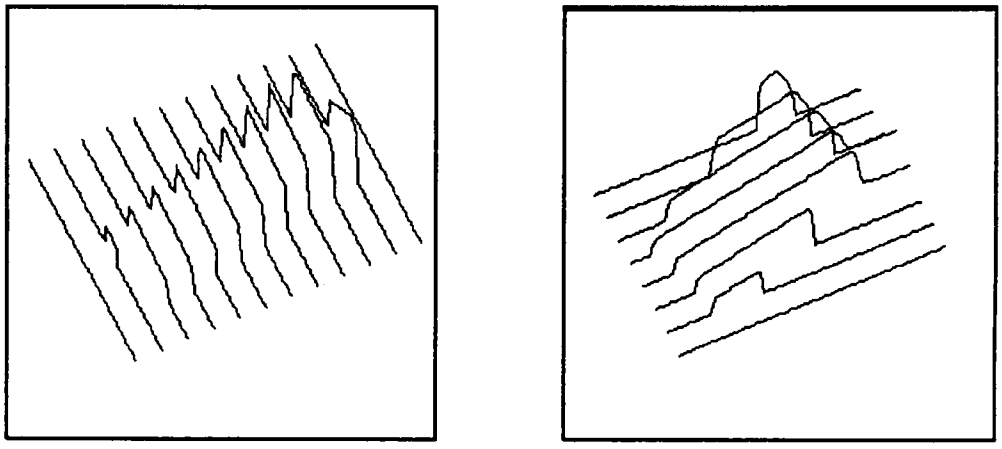
- *Step 3:* Reposition the sensor to begin acquiring scan lines along the Y–axis of the *robot frame.* Go to *Step 1:*

- *Step 4:* Compute the average of the two estimated values for the number of surfaces.

Figure 4.5 shows an orthogonal pair of 2-D scan lines used for estimating the number of surfaces. Note that each scan line does not include a substantial portion of the object to provide a good enough measure for the total number of object surfaces. In addition, by utilizing orthogonal sets of scan lines we ensure that each object surface is encountered in one scan line or the other. The histogramming and averaging processing allow us to account for surfaces that appear in, at the most, one or two scan lines and also correct for the fact that there exist surfaces that are encountered in several distinct scan lines.

4.2.3 Surface normal computation using three–point measurement

For polyhedral objects, determining the *pose* of an object *wrt*, the robot coordinate frame requires the knowledge of the *pose* of its individual surfaces. In the previous



(a) (b)

Figure 4.5: *Horizontal (a) and vertical (b) scans are used to generate the initial hypothesis for number of object surfaces.*

section an algorithm was presented which can be used as a precursor to the task of determining the *pose* of the object surfaces. For any planar surface, the position of the surface is generally specified in terms of its centroid or a vertex of the surface. In order to locate the centroid or one of its vertices, we need to acquire 3–D range data of the surface. In this section we will focus on extracting the orientation of a surface, and the computation of surface position will be considered in the next section. Using the knowledge of the number of planar surfaces computed subsequent to analyzing a line profile, we can then proceed to locate three non–collinear points on each of these surfaces and to compute the orientation (surface normal) of individual surfaces.

Input: Location of a point on the surface, PLR sensor *pose* for acquiring point measurements.

Output: Surface normal or equation of the surface.

- Move to a known location on the surface and acquire a point measurement.
- Locate two additional non–collinear points within a small neighborood of $\pm$ 0.5 $in.$
- Transform the PLR measurements from these three points to the world coordinate frame, and compute the 3–D components of the surface normal (direction cosines) using the equations (4.1)–(4.3).

Note that equations (4.4)–(4.6) provide the euler angles required for the robotic arm to realign the laser beam of the PLR sensor along the computed surface normal.

$$n_x = \sum_{i=1}^{N}(y_i - y_{i+1})(z_i + z_{i+1}) \tag{4.1}$$

$$n_y = \sum_{i=1}^{N}(z_i - z_{i+1})(x_i + x_{i+1}) \tag{4.2}$$

$$n_z = \sum_{i=1}^{N}(x_i - x_{i+1})(y_i + y_{i+1}) \tag{4.3}$$

$$d = \begin{cases} \frac{\pi}{2} + tan^{-1}(n_z/n_{xy}) & n_{xy} \neq 0 \\ \pi & n_{xy} = 0 \end{cases} \tag{4.4}$$

$$e = \begin{cases} tan^{-1}(n_y/n_x) & n_x \neq 0 \\ r & n_x = 0 \end{cases} \tag{4.5}$$

$$r = \begin{cases} -90 & n_x > 0 \\ +90 & n_x < 0 \\ 0 & n_x = n_y = 0 \end{cases} \tag{4.6}$$

Where n_x, n_x, n_x = 3–D components of the surface normal

x_i, y_i, z_i = 3–D coordinates of the i^{th} point

N = modulo 3 integer

$n_{xy} = \left|\sqrt{n_x^2 + n_y^2}\right|$

Table 4.1 shows measured and computed values of the *d* and *e* euler angles for System II [32], required for the robotic manipulator to realign the PLR sensor such that the laser beam is orthogonal to the object surface. For each of the measurements, the orientation of the object was changed affecting the *d* and *e* angles while the third euler angle *e* was kept constant. The measured values correspond to physically measured angles, whereas the computed angles correspond to the angles computed using three discrete range measurements. The error between the measured and computed values also includes physical measurement errors. The error also indicates that for arbitrary orientations, the surface normal computation is not very accurate. The accuracy of computed values of surface normals is accurate enough to realign the PLR sensor without any discernible difference in the resulting alignment.

4.2.4 Region based segmentation of 3–D range image data

Input: 2–D or 3–D bounding box of the object, and PLR sensor orientation.

Output: Regions corresponding to visible object surfaces, and background.

The field of view is much larger for 3–D image data acquired by scanning a 2–D region. A larger field-of-view allows extraction of much more information pertaining to the objects being scanned. However, by narrowing down the focus of attention while acquiring data, localized information can be extracted easily and reliably (sections 4.2.1 and 4.2.2). In this section we discuss and describe the technique utilized to extract

Table 4.1: *d and e angles (in degrees) for the robot arm to orient itself along the surface normal.*

d and *e* angles computed using surface normals acquired from			
Physically measured values		Range measurements at three non-collinear points	
d	*e*	*d*	*e*
159.0	-90.00	159.19	-90.00
159.0	-80.00	158.96	-81.03
159.0	-56.00	158.47	-59.53
159.0	78.00	157.81	78.69
159.0	64.00	157.61	61.95
150.0	75.00	150.00	75.96
150.0	66.00	151.35	66.25
150.0	-57.00	150.49	-57.99
150.0	-42.00	151.05	-40.60
150.0	-77.00	150.20	-77.91

homogeneous surfaces based on the range segmentation technique proposed by Besl and Jain [13]. Homogeneous surfaces in this case are the surfaces that are characterized by constant surface normal orientations. Further, abrupt changes in surface normal directions are easily detected. This technique is quite useful for isolating individual object surfaces from the background. The surface normals at each point in the range image may be characterized by their orientation given by the scalar function of equation (4.7).

$$n = tan^{-1}(\frac{d_y}{d_x}) \tag{4.7}$$

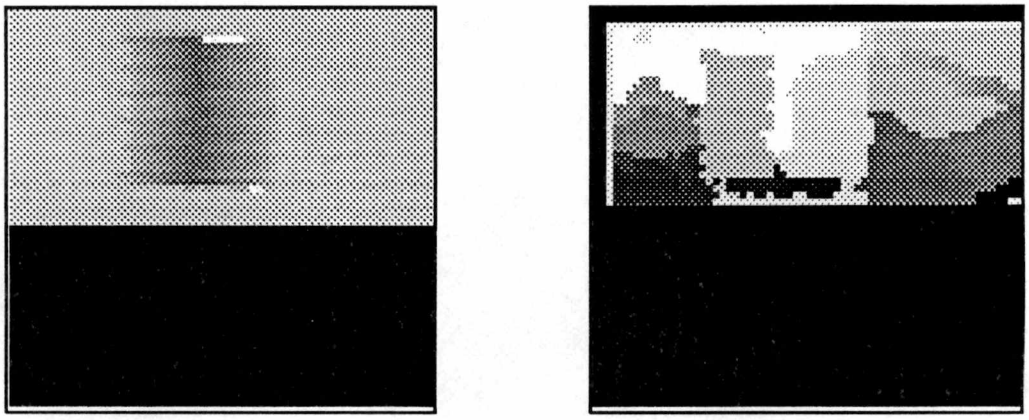
where d_x and d_y are the x and y gradients computed using 3 $\times$ 3 Sobel operator. A technique similar to region growing clusters groups of connected points over which the

magnitude and direction of the scalar function n varies over a small range of values. The use of both magnitude and direction allows detection of object surfaces with relatively smaller variations in surface normals. Since this technique is quite sensitive to noise and since the Sobel convolution operation tends to highlight fairly low levels of gradients, the technique makes the region growing operation break homogeneous regions into separate smaller regions. A median filtering operation prior to the Sobel convolution operation has been found to reduce the random noise and the false contouring effect produced by the Sobel operator.

Figure 4.6 shows the original and the processed images for the Wedge shown in Figure 4.3(a). The final result shows that the background is split into several regions, and this occurs primarily due to the Sobel operator highlighting small gray level changes in the background. The two object surfaces are isolated without much difficulty. This segmentation process does not yield the surface equations or relative orientations of individual surfaces. The techniques described in the previous two sections (4.2.1 and 4.2.2) can be used to accurately determine the surface normal and surface orientation informations.

4.3 Experimental Verification of the Active Haptic Exploration Modules

The individual haptic exploration modules are fairly specialized, where every subsequent motion is a direct consequence of the previous motion or previously acquired tactile and force/torque data (i.e., data driven). The primary objective of the exploration strategy is to extract enough details regarding the object descriptors to allow a fairly accurate representation of the object. This entails being able to extract informa-



(a) (b)

Figure 4.6: *A 3–D scan for the object shown in Figure 4.3(a) was acquired and the original image (a) for a wedge is segmented into individual regions (b) in order to isolate the object surfaces.*

tion, such as the *pose* of all the accessible surfaces, edges, and position and connectivity information of all accessible vertices (i.e., which two vertices are connected). Thus, to facilitate the active exploration of objects to extract surfaces, edges, and vertices in a data driven framework, we have designed our algorithms to perform systematic exploration. Also the exploration scheme discussed here is consistent with the sensor selection scheme discussed in section 3.1.2. The hierarchical dependency between the three descriptors, namely surface, edge, and vertex, can be seen very cleary from Table 4.2. For instance, to extract the length of an edge, we begin with exploring the corresponding surface and move along the surface in an arbitrarily selected direction till the object edge is encountered. After locating the edge, we travel along the edge in either direction until the two vertices along the edge are isolated.

We now discuss the tasks associated with the EM used for generating composite

Table 4.2: *Hierarchical dependency relationship in tactile descriptor extraction schemes.*

Object descriptors being extracted	**Object descriptors being tracked**
Composite Surface	Surface
Edge *pose*	Surface
Edge length	Edge
Vertex Position	Edge

tactile image, computing wrist orientation through surface normal computation and edge/vertex exploration.

4.3.1 Composite image generation using overlapping tactile imprints

Generally, objects larger than a tactile sensor can be recognized or identified by exploring their entire surface. Although by having an object model library, objects can be uniquely identified by acquiring tactile images from selected locations [27, 17]. The entire surface of objects that are relatively larger than the sensor can be reconstructed using active sensing. This process might be quite time consuming and thus wasteful for fairly large objects. For instance, tasks such as machining and inspection often require exploration or investigation of portions of object surfaces that are relatively larger than the sensor size. Further, for pick–and–place types of tasks, the robotic end–effector can achieve a better grip on the object by exploring specific portions of interest for gripping. In addition to providing the surface characteristics of the explored portion of the object, the reconstructed portion when transformed into world coordinates can be used to extract relational information. The relational information is most essential since it provides a global perspective of the location and orientation of the object in relation to other known objects in the vicinity of the robotic manipulator.

It is extremely important for robotic manipulators to be able to acquire the best possible contact in order to utilize tactile sensing most effectively. The non–planar objects have to be gripped each time with the plane of the tactile sensor maintained orthogonal to the surface normal. This pre–condition can be maintained by continuously monitoring the torques measured by a force/torque sensor placed at the wrist of the robotic manipulator. Ideally, the torques measured at the wrist of a robotic manipulator would normally be zero when the tactile sensor makes a flush contact with a planar surface. In order to select the direction of motion for the sensor relative to the object, we have developed a scheme which provides the robotic manipulator the 3-D world coordinates of the next position of contact based on the 3-D position and the tactile data acquired from the current location.

- Initially tactile data from a known position is acquired, ascertaining that at least a small portion of the object comes in contact with the sensor.

- From any given location there can be eight possible directions of motion. However, one of these directions will be the one already used to get to the current location. The possible directions of motion are determined from the current tactile data. The eight directions are numbered from 0 to 7 in counter clockwise direction (i.e., North is 0, North–West is 1, and so on). Whenever the object extends to the extreme rows (first or last) or the extreme columns (first or last), it implies that the object can be explored in the direction of the corresponding row or column. Further, the diagonal directions are determined by observing if the object seems to stretch beyond two adjacent extreme rows and columns.

- The current 3-D position and all the possible directions of motion are first saved on

a last–in–first–out (LIFO) stack. The possible directions computed at the current location are pushed on the stack in descending order, which ascertains that the object is explored completely and in a systematic fashion.

- The stack is then popped, and for each direction the next 3-D position (in world coordinates) of the arm is computed for further object exploration. Thus the object is explored in all the directions in which further exploration is needed, and a composite image is then generated.

- Using the newly generated composite image, the subsequent directions for further exploration are determined and pushed on the LIFO stack.

The process of generating a composite tactile image from a set of tactile images with overlapping portions between successive images requires knowledge about the overlapping portions. Due to smaller spatial resolution of the tactile sensor, it is quite difficult to maintain consistent overlap between successive images. The process of reconstruction has to ascertain the best possible overlap, in addition to compensating for possible mechanical errors introduced by the robotic manipulator while exploring the object. Further, surface and shape characteristics have to be matched while constructing the composite image. Cross–correlation between two successive images can be used to locate the overlapping portions (or common portions) between the two images. A simple cross–correlation technique is insufficient for this task for the following reasons.

- The response of cross–correlation increases with the area of overlap, which makes the detection of overlapping regions more complex.

- Regions with similar shape and different sizes are difficult to discriminate.

- Regions with similar sizes and shapes but with different surface characteristics are difficult to discriminate.

The two portions of images that belong to a common region of the object can be located by computing the difference (or error) in their surface and shape characteristics. Although two regions that are symmetrical about their axes can pose a problem to such a technique. For instance, for a rectangular overlap it will be difficult to know the exact size of the overlap since the error remains constant as the overlap is increased along one of its principal axes.

We developed a least–square cross–correlation technique to compute the best possible overlap between successive tactile images. The technique compensates for errors occurring due to orthogonal shifts (translation orthogonal to the direction of sensor motion) between successive tactile images. The following steps explain in detail the procedure of acquiring and processing multiple images for generating the composite image of an object larger than the sensor pad.

Input: 2–D Object location, initial tactile sensor *pose.*

Output: Composite 3–D tactile imprint of the object.

- *Step 1:* Move the manipulator holding the tool housing the tactile sensor to the predetermined position. This position is determined such that the first tactile image shows substantial overlap with the object being explored.

- *Step 2:* Acquire the tactile image and compute the next position for the manipulator to acquire the subsequent image using the technique described above. If no new position can be selected, then terminate.

- *Step 3:* Move the manipulator to the next position computed in *Step 2*, and repeat *Step 2*.

- *Step 4:* Perform the tactile image to world coordinate frame transformation, described in [33], for the newly acquired image in *Step 2*.

- *Step 5:* By sliding the second image over the first (one taxel at a time), compute the squared–error associated with the corresponding taxel of the two images.

- *Step 6:* The squared–error is then divided by the actual number of overlapping object taxels (4.8). Compute the least–squared–error for the corresponding rows and columns of overlap by dividing the squared–error by the area of overlap (4.9).

$$SQE_{i,j}(p) = \frac{1}{t_{ov}(p)} \left(\sum_{q=1}^{C_{i,j}} (t_i(m-p+1,q) - t_j(p,q))^2 \right) \tag{4.8}$$

$$LSE_{i,j}(R,C) = \frac{1}{R_{i,j}C_{i,j}} \left(\sum_{p=1}^{R_{i,j}} SQE_{i,j}(p) \right) \tag{4.9}$$

Where $SQE_{i,j}(p)$ = Squared error associated with p^{th} overlapping row between images i and j

$LSE_{i,j}(R,C)$ = Least-squared error between images i and j with $R_{i,j} \leq n$ overlapping rows and $C_{i,j} \leq m$ overlapping columns

t_i and t_j = i^{th} and j^{th} tactile images (with n rows and m columns) respectively

$t_{ov}(p)$ = Number of overlapping taxels in row p

- *Step 7:* Locate the global minima (or the valley point) in the two–dimensional

least–squared–error function. Append the corresponding images using the appropriate overlap. Move the manipulator to the next position computed in *Step 2*, and repeat *Step 2*.

A two dimensional array associated with the least–square error is generated as an outcome of the cross–correlation. The error is maximum (infinity) when none of the object taxels overlap. The least–squared error gradually decreases with increasing overlap, eventually reaching its minima indicating the location of best overlap and increasing thereafter with the additional increase in rows or columns.

In order to evaluate the validity and effectiveness of our active exploration approach, a series of experiments were conducted. However, if the shapes of object portions being cross–correlated are quite similar (e.g., rectangular), then the best overlap will be obtained based on their surface characteristics. The least–square cross–correlation technique selects the best overlap based on both surface and shape characteristics. This ensures that two similar shaped regions with widely varying surface characteristics are distinguished quite easily. Surface or shape descriptors corrupted with noise are always difficult to manipulate. Figure 4.7 shows a set of five tactile imprints acquired by bringing the tactile sensor in contact with a long key and maintaining sufficient vertical pressure. The initial imprint is analyzed to determine the 3–D locations for acquiring the subsequent four tactile imprints. The composite tactile imprint is then generated using these five tactile imprints, and the result is shown in Figure 4.7. The overlap between successive images was maintained 50% or 5 rows. The disadvantage in generating a composite image by utilizing the a priori knowledge (50% overlap) is that due to poor spatial resolution, it is quite difficult to assume that the overall size subsequent to the merging of all the acquired images would remain close enough to the actual size of the

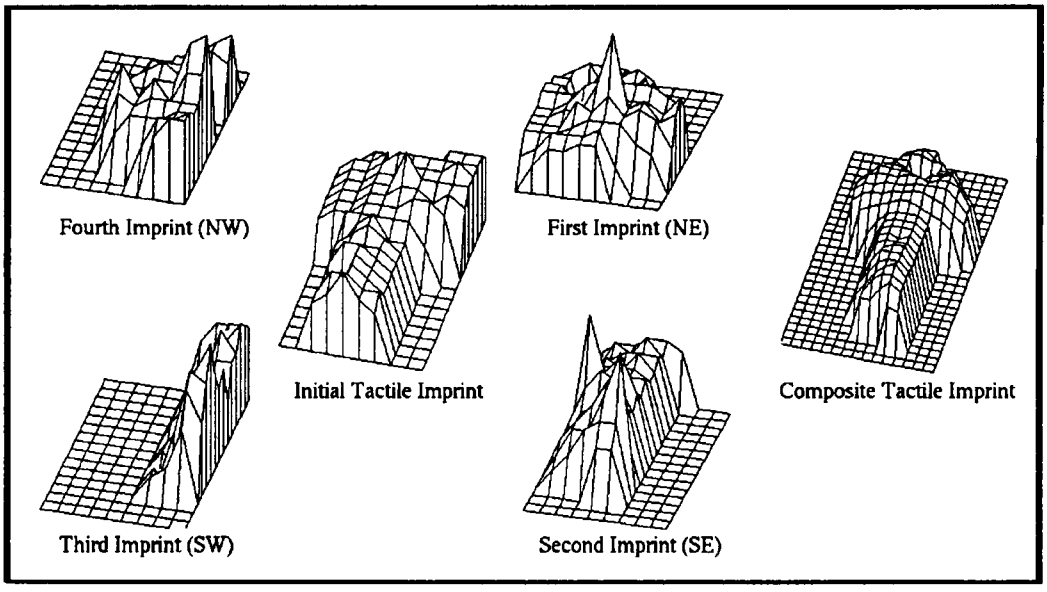


Figure 4.7: *The individual tactile imprints are combined using least-squared cross-correlation technique to generate the composite tactile imprint.*

explored object (or object portion).

4.3.2 Wrist reorientation through surface normal computation

Due to the planar construction of the tactile sensor being used, extraction of tactile data necessitates a planar contact between the sensor and the object surface. In order to ascertain a flush contact between the tactile sensor surface and the object surface, it is necessary that the surface normal of the object surface being explored be aligned with the Z–axis of the *sensor coordinate frame*. In order to realign the sensor, the object surface normal needs to be computed. The mechanism designed to determine the surface normal of any surface utilizes the principles of rigid body statics. When a vertical load L is applied onto any surface using a tool (Figure 4.8), then one of

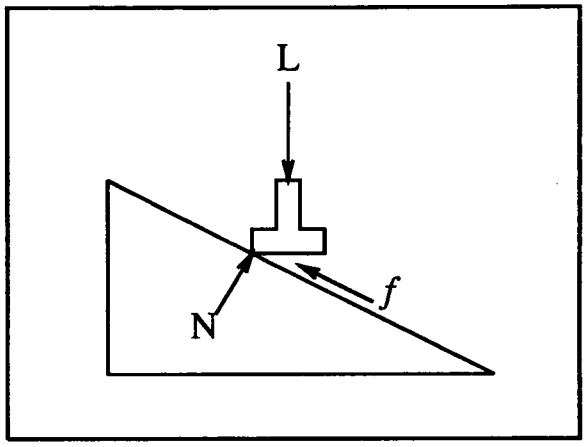


Figure 4.8: *For a load L applied on to an inclined plane results in a normal reaction force N and frictional force f.*

three conditions exist, namely the tool is stable and stationary at the point of contact; the tool is on the verge of sliding along the surface; or the tool is sliding along the surface, depending on the existing frictional force f [11]. For the second and third scenarios, the reaction force is the resultant of normal reaction N and the frictional force f. Alternatively, for the first scenario, the reaction force is equal to the normal force N. When the tool housing the tactile sensor makes a contact with any surface, the 3–D components of force made available from the force/torque sensor clearly reflect the orientation of the object surface. However, the inherent vibrations of the robotic arm and the inaccuracies associated with force/torque sensor require a large number of readings in order to average out these errors.

An alternate scheme would be to utilize three additional points on the object surface, and then the surface normal components can be computed by substituting the 3–D coordinates of the four points in the equations (4.10)–(4.12).

$$n_x = \sum_{i=1}^{N}(y_i - y_{i+1})(z_i + z_{i+1}) \tag{4.10}$$

$$n_y = \sum_{i=1}^{N}(z_i - z_{i+1})(x_i + x_{i+1}) \tag{4.11}$$

$$n_z = \sum_{i=1}^{N}(x_i - x_{i+1})(y_i + y_{i+1}) \tag{4.12}$$

Where n_x, n_x, n_x = 3–D components of the surface normal

x_i, y_i, z_i = 3–D coordinates of the i^{th} point

N = modulo 4 integer

If the first location of contact is assumed to be at (x, y, z) then the subsequent three locations are selected to be (x+dx, y, z), (x+dx, y+dy, z), and (x, y+dy, z) (Figure 4.9). However, the actual Z coordinate for each of the three additional points is determined by first moving the tool housing the tactile sensor to the above mentioned points and then moving it towards the object (i.e., along Z–axis) until a contact is made with the object. The directions (or signs) of dx and dy increments along the X and Y axes of the robot are such that the four points are traversed in a counter clockwise direction, guaranteeing the surface normal pointing away from the object. Further, as mentioned earlier, the surface orientation can be determined using the 3–D components of force measurements using the force/torque sensor, which enables us to decide the signs on dx and dy increments.

The force/torque sensor used for determining surface orientations needs to be calibrated since the 3–D components of the force vector are not normalized and are also multiplied by internal scaling factors, which themselves need to be computed. We designed a technique for computing these scale factors using a set of training and test

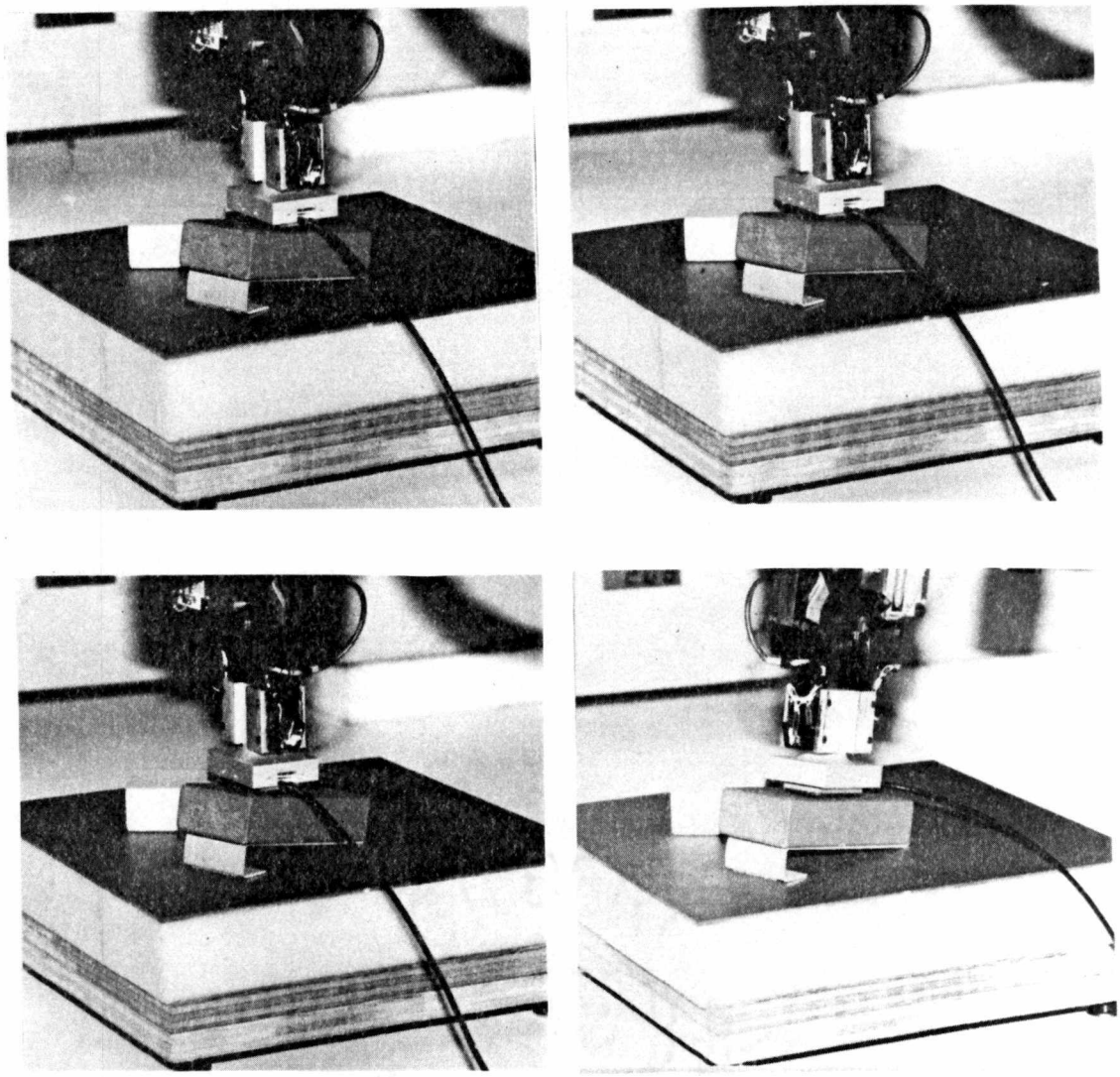


Figure 4.9: *Force/torque readings at four different sensed position are used to orient the Robotic arm from the initial position to the final position which is along the surface normal.*

sample readings of the force vector. Ideally, for any inclined surface, the surface normal components available directly from the force components of the force/torque sensor should agree with the surface normal values computed using four (or at least three) known points on the surface. The training set of samples consisted of two sets, namely, 3–D force vector components acquired through force/torque sensor and 3–D surface normal components computed using four known points on a surface (N_p), for 10 distinct surfaces. The normalized 3–D force components are then used for computing the three scale factors using (4.13)–(4.18).

$$sf(x) = F^n_{training}(x)/N^p_{training}(x) \tag{4.13}$$

$$sf(y) = F^n_{training}(y)/N^p_{training}(y) \tag{4.14}$$

$$sf(z) = F^n_{training}(z)/N^p_{training}(z) \tag{4.15}$$

$$N^f_{testing}(x) = F^n_{testing}(x)/sf(x) \tag{4.16}$$

$$N^f_{testing}(y) = F^n_{testing}(y)/sf(y) \tag{4.17}$$

$$N^f_{testing}(z) = F^n_{testing}(z)/sf(z) \tag{4.18}$$

where

sf(x), sf(y), sf(z)	=	The X, Y, and Z scale factors
$F^n_{training}(x), F^n_{training}(y), F^n_{training}(z)$	=	Normalized 3–D components of the force vector (training sample)
$N^p_{training}(x), N^p_{training}(y), N^p_{training}(z)$	=	Calculated 3–D components of the surface normal using position

information (training sample)

$N^f_{testing}(x), N^f_{testing}(y), N^f_{testing}(z)$ = Calculated 3–D components of the normal using force vector (testing sample)

The set of testing samples consisted of similar 10 readings and the normalized force vector components were divided by the corresponding scale factors to get the corrected 3–D components of the surface normal. In order to check the accuracy of the two sets of surface normals, the two angles necessary for the robotic arm to align with the surface normal, i.e., θ and ϕ, were physically measured. These angles were also computed using the two sets of 3–D surface normal components. Table 4.3 shows the three sets of θ and ϕ angles and the θ_p and ϕ_p values computed using surface normal components acquired from position information consistently agree quite closely with θ_m and ϕ_m the measured values. The θ_f and ϕ_f values computed from the surface normal components acquired using the force vector exhibit fairly large error for certain readings and this can be attributed to the calibration procedure which assumes ideal repeatability for the force/torque sensor. Further, the end–effector, being unevenly loaded due to the presence of a CCD camera and an ultrasound range sensor, using a single scale factor, may not be the best solution.

4.3.3 Exploration of edges/vertices associated with an object surface

Edge and vertex extraction is a fairly complex process. Due to inherent noise in the tactile imprints, the precise location of *pose* orientation of edges and locations of vertices can be quite tedious. The edge and vertex extraction process isolates all the edges and vertices associated with the face of the object being currently explored. The process

Table 4.3: *θ and ϕ angles (in degrees) for the robot arm to orient itself along the surface normal.*

θ and ϕ angles computed using surface normals acquired from				Physically measured values	
Normalized and corrected force vector components		Four points on the surface			
θ_f	ϕ_f	θ_p	ϕ_p	θ_m	ϕ_m
114	54	112	53	110	54
116	49	122	53	122	54
68	62	69	54	68	54
54	61	41	54	42	54
59	67	45	60	45	60
39	72	35	60	34	60
128	61	126	59	125	60
125	55	140	59	139	60
60	73	58	68	56	68
125	62	119	68	117	68

terminates when the robotic arm returns to a vertex already isolated. Since the objects used in the experimentation are restricted to only planar faces, we can expect to detect only linear or straight edges. The coarse level edge isolation is performed using Hough transform, and the finer level edge isolation is performed using a minimum squared error line fitting technique. The edge and vertex extraction process can begin from any location on the face of an object as long as the tactile sensor surface is positioned parallel to the face being explored.

Input: *Pose* of the object surface.

Output: For each edge the length and *pose*, vertex locations, hypothesizing of adjacent surfaces.

- Obtain an initial tactile imprint from the current location of contact on the object surface. Check/examine if the tactile imprint exhibits presence of an edge. This test is a fairly simple test since we only need to verify the presence of a background (or non–object) region, which is not enclosed within the object region.

- Once an edge(s) of a face is encountered, the robot arm is moved to a new position such that the edge passes through the origin of the sensor frame.

- A new tactile imprint is then acquired and the following operations are performed.

  - Compute the Hough transform to isolate the lines in the tactile imprint. The Hough transform is computed in the (ρ, θ) domain in order to include lines oriented in $(-\pi/2, \pi/2)$ range [32].

  - Line clusters centered around prominent lines are then extracted. In the Hough domain a large value corresponding to a specific ρ and θ indicates

the presence of a line with Y–axis intercept ρ and slope θ. By allowing a certain amount of flexibility in the slope and intercept parameters, it is possible to locate relatively smaller lines in the vicinity of a large line. Once a line cluster is located, although it represents a family of lines, the taxels (or tactile elements) corresponding to each line cluster are then considered as forming a single line.

- These line clusters are sorted based on the number of taxels located on the largest line in each cluster.
- The line clusters are then assigned labels and the corresponding taxels are also labeled. Those taxels that belong to more than one line cluster are assigned modified labels. For instance taxels belonging to lines 1 and 2 are labeled 12, and taxels belonging to lines 2, 5, and 6 are labeled 256. These taxels with modified labels indicate intersection between all the isolated lines by Hough transform.
- The first line cluster (longest line) from the sorted line clusters is then selected and line fitting based on the minimum squared error (MSE) technique is then performed [57]. The slope of the fitted line is used to compute the new wrist orientation.
- The robot arm is then reoriented such that the Y–axis of the sensor frame is aligned with the longest line. The second line fitting technique provides a much better line as compared to the one provided using Hough transform.

• A new tactile imprint is acquired to check/examine if the sensor is already at one of the end–points (corner or vertex) of the isolated line. This test is performed by

repeating the steps indicated above for the longest line isolation and then looking for taxels with modified labels on the line oriented along the Y–axis of the sensor frame. If more than one taxel that includes eight connections is found, then the centroid of all these taxels provides the location of the vertex. If the isolated taxels are not eight-connected and separated by more than one taxel, then both end–points (vertices) are assumed to be located. By transforming the isolated end–point(s) from sensor frame to robot frame, we can avail ourselves with the 3–D location of the corner(s) of the object.

- If no corner (vertex) is located, then the sensor is moved along the edge in either direction of the edge till both the corners or end–points are located.

- From one of the known vertices the next edge is then followed in a similar manner.

- The 3–D locations of the isolated corners (vertices) are saved and continuously monitored to check if a newly detected corner is already included in the list of corners, which terminates the edge/vertex extraction process.

During edge/vertex exploration when an edge is encountered in the tactile imprint, Hough transform enables the location of all possible lines. We are interested in lines that consist of at least four contour points. This condition helps eliminate a large set of unwanted lines. From the remaining set of lines, clusters of lines, which have their respective ρ and θ values within a preset threshold, are formed. The MSE line fitting is then performed on the line from each cluster that consists of the largest number of contour points. Figure 4.10 shows the plots of the lines computed using the MSE line fitting. The parameters of the line in Figure 4.10 are then used to align the Y–axis of

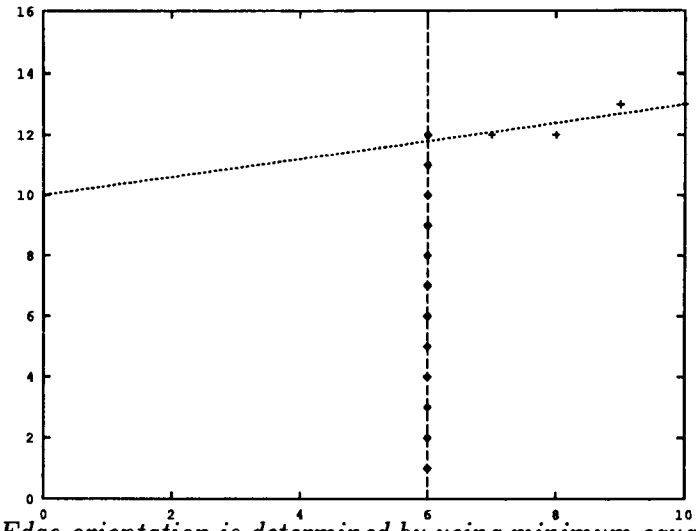


Figure 4.10: *Edge orientation is determined by using minimum squared error line fitting technique.*

the tactile sensor along the object edge. A second tactile imprint is then acquired to verify the alignment process.

Once the tactile sensor is aligned along the object edge, the two edge end–points or object vertices need to be located. The tactile sensor is moved along the edge, and tactile imprints are acquired at regular intervals to determine if a vertex is encountered. After locating one vertex, the robotic arm returns to the first position on the edge and begins to move along the edge in the opposite direction to locate the second edge. Figure 4.12 shows a set of tactile images acquired at regular intervals (Figure 4.11) during edge tracking. After aligning with an edge, the first imprint serves to verify the alignment. Subsequently, while tracking the edge, each new tactile imprint is examined to locate the end–points of the edge.

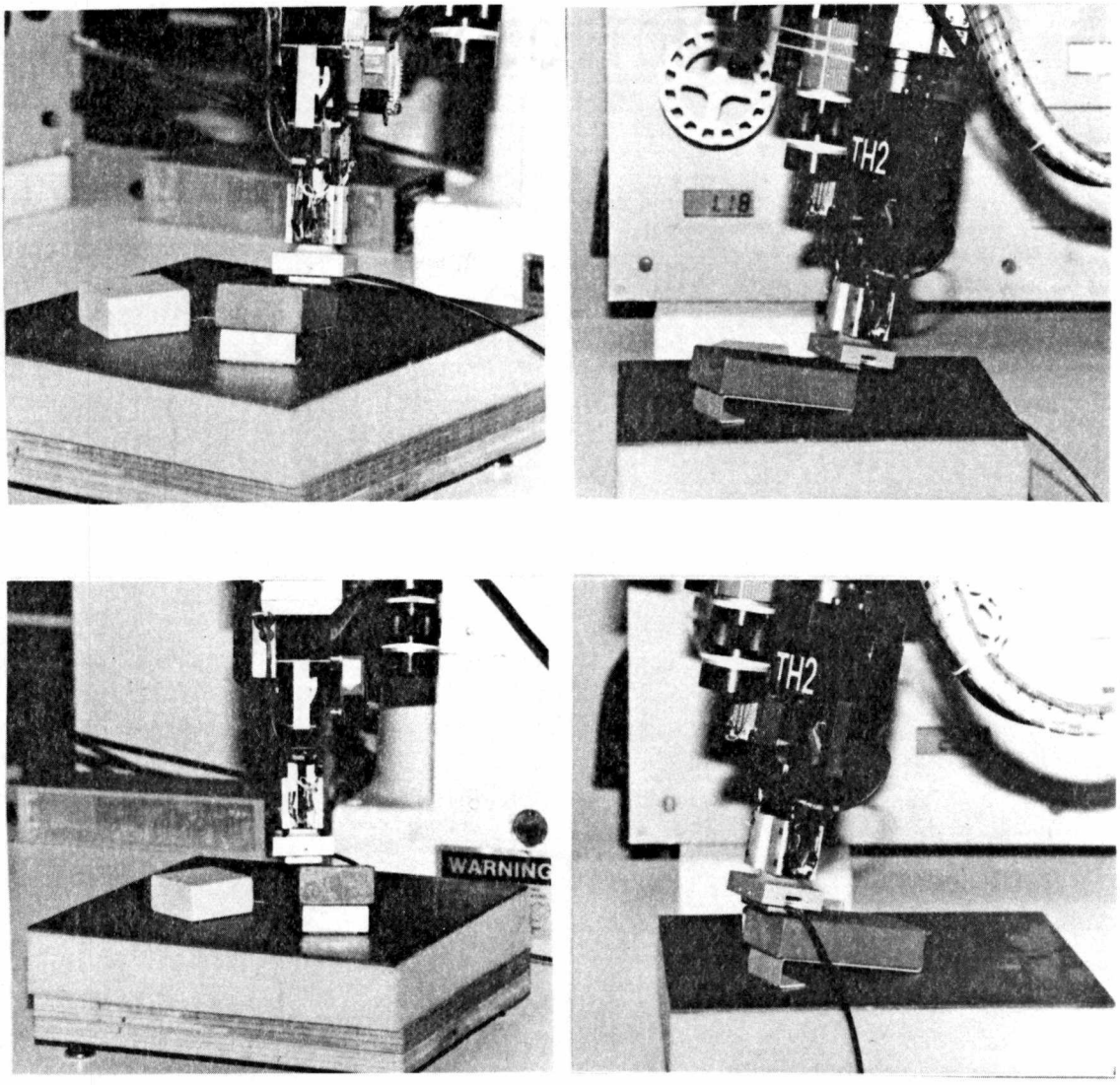
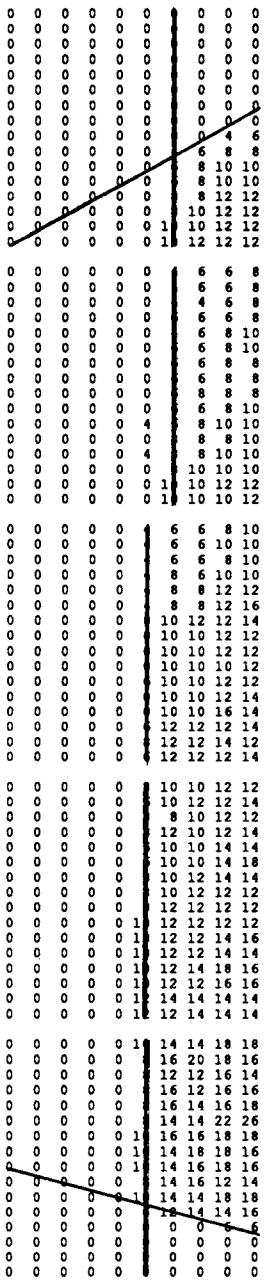


Figure 4.11: *The figure shows tactile acquisition process involved in systematic object edge isolation and edge tracking.*



Imprint 5: Orientations of the detected lines were 90° and 30° *wrt* X–axis.

Imprint 4: Orientation of the detected line was 90° *wrt* X–axis.

Imprint 1: Orientation of the detected line was 90° *wrt* X–axis.

Imprint 2: Orientation of the detected line was 90° *wrt* X–axis.

Imprint 3: Orientations of the detected lines were 90° and $-20°$ *wrt* X–axis.

Figure 4.12: *Tactile imprints acquired at regular intervals while performing edge exploration.*

CHAPTER 5

EVALUATION OF THE INTEGRATED ACTIVE EXPLORATION SYSTEM: EXPERIMENTAL STUDIES

In chapter 4, various algorithms utilized for the extraction of the required descriptors was described in detail. This chapter attempts to provide the motivation behind designing a set of experiments, along with, the inherent differences between various experiments. In addition, this chapter emphasizes the way in which the motivation and unique features of each experiment essentially help to bring out the major contributions of this research. There are certain *limitations* associated with the experimental set up that required the experiments to be carried out after making some *assumptions* pertaining to the robotic system, the sensors, the objects, and the work cell where the objects to be explored are located.

5.1 Constraints on the Experimental Study

1. The robot itself has limitations associated with its reach or its ability to reorient itself to attain arbitrary configurations. This limitation will limit the exploration process when attempting to access all the surfaces of an object, especially while using touch. This accessibility factor also requires the robot to pick up the tactile sensor only when it is necessary and then to replace the sensor to its default position. Such a scheme is required because the tactile sensor is housed in a special tool. Particularly, for PLR sensing, it is necessary to ascertain that the

entire object height is within the dynamic range of the PLR sensor. While carrying the tactile sensor tool, this pre–condition for PLR sensing cannot be guaranteed, and moreover, while scanning an object there is a possibility of the object being dislodged by the tactile sensor tool.

2. Each of the sensors has certain requirements in terms of working conditions. For example, while using vision the assumption is that the lighting conditions are favorable. Similarly, it is necessary for the tactile and force/torque sensors that the force applied by the robotic manipulator is sufficient to acquire reliable contact information.

3. The objects selected for the experimental verification phase are polyhedral objects. For the purpose of haptic exploration, it is necessary to make sure that the surfaces are planar. Further, the surfaces need to be large enough for the tactile sensor to move along the contour of a given surface without encountering a situation where two non–intersecting edges are visible in the tactile imprint. This assumption requires the distance between any pair of surfaces to be larger than the tactile sensor size.

4. The workcell, where the objects being explored are placed, is of 1 square ft. area. This region is sufficient for placing 4 to 5 objects in the scene. The objects need to be rigidly placed, and the assumption here is that during *contact* sensing the object does not get physically moved. It is also assumed that when the work cell consists of multiple objects, the objects do not have any overlapping portions and are not located too close to each other, thereby hindering the haptic exploration process.

5.2 On Designing Experimental Studies

In the previous chapters, details pertaining to the primary objective, the design of the overall system, and the specifics of individual modules were discussed. The primary objective included developing the computational framework for a multisensory active exploration system. The active exploration system is characterized by certain highly desirable qualities, such as modular architecture, a closed–loop control mechanism, an incremental data integration system, modality specific data acquisition and processing modules, a sensor selection scheme, flexibility, and graceful degradation. In order to demonstrate the advantages associated with these desirable features incorporated in the active exploration system described here, a set of experiments needs to be designed. The set of objects selected for the experimental verification were limited to polyhedral objects. The set of objects selected for the experiments can be broadly classified into

1. Convex non–compliant objects

2. Concave non–compliant objects

3. Thin planar non–compliant objects (i.e., objects with planar surfaces, where the distance between the planar surfaces is small)

4. Convex/Concave compliant objects

The active exploration experiments have been designed to demonstrate the capabilities of the system using objects from the above mentioned four distinct types of polyhedral objects. The next consideration would be to have a means by which the system performance can be evaluated quantitatively and qualitatively. For the purpose

of qualitative analysis *accuracy of results* will be utilized. For quantitative analysis, the eventual 3–D model will be analyzed to verify various geometrical features, namely

1. Number of surfaces
2. Number of vertices
3. Number of pairs of parallel surfaces
4. Number of edges located by haptic exploration
5. Relative orientations of edges located by haptic exploration

Note that the *accuracy of results* parameter used for quantitative analysis will essentially reflect the deviation of the results provided by the exploration system from real values.

The entire experimental verification task has three independent objectives. The first one is to demonstrate the functionality of the integrated system by having objects from the above mentioned four distinct categories. Through the first set of experiments it is possible to highlight the modular architecture, closed–loop control mechanism, incremental data integration, modality dependent data acquisition and processing aspects of the active exploration system. The objects from each of these four categories have to be handled quite differently based on their respective complexities. The second and the third objectives deal with demonstrating the unique contributions pertaining to flexibility and graceful degradation, respectively.

Experiment 1: *Exploration of a simple polyhedral object*

The first experiment involves exploring a relatively simple object; however, due to the arbitrary position and orientation of the object, a substantial amount of

complexity is introduced in the exploration process. The complexity deals with the ability to identify the surface orientations so that sensor realignment would be required for exploring each surface. Through this experiment, the detailed description of various steps involved in the entire exploration process are provided. Using *a simple polyhedral block placed at an arbitrary position/orientation*, the experiment allows us to demonstrate the distinctive advantages associated with the proposed computational framework and architecture for active exploration.

Experiment 2: *Haptic exploration of multiple surfaces*

The structure of the object used in the first experiment is such that only the top surface is easily accessible for haptic exploration. This is due to the fact that the robotic manipulator itself has certain limitations pertaining to its ability to orient itself such that surfaces at any arbitrary orientation can be explored using the tactile sensor. The object being utilized for the second experiment has *more surfaces and two surfaces have to be explored using touch.* This allows us to demonstrate the robustness of tactile exploration and the Half–space modeler's ability to handle objects with more than six surfaces.

Experiment 3: *Exploration of concave objects*

In the first and second experiments, the object used was convex non–compliant and polyhedral. The *concave objects* are generally much more difficult to model as compared to the convex objects because the modeling technique generally computes additional vertices for the concave objects that are erroneous. A vertex is considered erroneous if the modeler identifies it to exist as a consequence of computing the intersecting point of three adjacent surfaces. However, haptic ex-

ploration results can be used to verify or reject the vertices computed by the modeler. Such a verification process is made possible only through the closed–loop control mechanism and the modular architecture. The number of such vertices is dependent on the complexity of the structure of the concave object itself.

Experiment 4: *Exploration of thin planar non–compliant objects*

The third type of non–compliant polyhedral objects are those that are generally called thin planar objects. Objects such as keys, wrenches, etc., are typical examples of objects in this category. The Half–space modeler is not capable of developing a 3–D model of such objects. However, the composite tactile imprint generation technique will allow us to show that a localized 3–D imprint of the object can be constructed once the active exploration identifies that the particular object cannot be modeled using the Half–space modeler. This experiment consists of *two thin planar non–compliant objects* and the system explores the objects haptically and builds localized composite imprints of the objects.

Experiment 5: *Exploration of compliant objects*

The use of *contact* sensors, particularly force/torque, becomes quite essential when the robot's work cell consists of objects that are compliant in nature. Typically, *non–contact* sensors are unable to decipher whether a given object is compliant or non–compliant. This additional feature becomes important when there are two objects that are physically similar placed next to each other and the system is required to recognize which one is made of non–compliant material and which one is not. The fifth experiment is essentially designed to show that an approximate model of *two compliant polyhedral objects* is built using the *non–contact*

and *contact* sensors. Haptic exploration tasks such as contour following cannot be performed as easily as in the case of non–compliant objects. This limitation causes the system to develop an approximate model rather than an exact model of the object.

Experiment 6: *Exploration of workcell containing multiple objects*

In the previous experiments the purpose was to show how the system would handle either single or multiple objects belonging to a particular category of polyhedral objects placed in the robotic work cell. The challenge posed in this experiment is to deal with four objects, where *the objects belong to different categories of polyhedral objects.* Such a set up would require the active exploration system to identify the category to which the particular object is associated and then determine the extent and nature of exploration that will be performed. It should be noted here again that exploration of various types of polyhedral objects is feasible provided there exists a modular architecture, closed–loop control, and modality dependent (or modality specific) data acquisition and processing modules.

Experiment 7: *Flexibility of the exploration system*

The set up utilized for the sixth experiment requires repeated use of various sensors. The order in which these sensors are utilized can be changed, resulting in either a better, similar, or poorer result, depending on the sensors and the order in which they are utilized. This experiment is designed to show that the active exploration system is not tied to a unique sequence of exploratory tasks. The ability to *select the sensors and the order in which they are utilized* provides a unique feature to this system which has not been demonstrated by any other active ex-

ploration or perception system. Using the same objects utilized in the previous experiment, this experiment will indicate that the sequence of exploration events can be changed quite easily without having to make any modifications to the existing exploration system.

Experiment 8: *Graceful degradation of the exploration system performance*

The second feature that is unique to this research is the ability to demonstrate *graceful degradation.* The question here is that if the exploration system is unable to utilize all the available sensory modalities, then would the system still be able to continue its operation with a certain degree of degradation in its performance or would the system be rendered disfunctional. The active exploration system ideally utilizes vision, PLR, tactile, and force/torque sensors. In this experiment to study the graceful degradation feature of the system, the exploration will be performed using only the PLR and tactile sensors. The objects used will be the same as for sixth experiment and the eventual integration result will show that due to the fewer number of sensors, the system performance has degraded in terms of the *accuracy of result.*

The details and results at various stages of exploration associated with first six experiments will be presented in section 5.3. The seventh and the eighth experiments will be discussed in sections 5.4 and 5.5 respectively. For each of these experiments the order in which the sensors will be utilized depends to a large extent on the composition of the robot's work cell. A detailed discussion of the experimental results is provided in section 5.6.

The data acquisition and processing modules associated with the individual sensing modalities have been tested thoroughly to verify their ability to extract various object descriptors. The experimental setup utilized for the testing of various algorithms consists of a six degree–of–freedom Cincinnati Milacron $T^3 - 726$ robotic manipulator, a Cohu CCD camera, a Keyence LB–70 PLR sensor [46], a Lord LTS–210 (10×16) tactile sensor pad [49], and a Lord F/T–30/100 force/torque sensor. The schematic diagram of the entire robotic manipulatory system consisting of various sensing modalities along with their respective controllers is shown in Figure 5.2, and Figure 5.1 shows a closer view of the robotic manipulator housing various sensors. The exploration results are displayed incrementally on a simulator [18, 19, 20, 44] that contains the simulated models of the robot and its work cell (Figure 5.3).

5.3 Exploration of Polyhedral Objects

Each of the experiments that are described here were performed using the robotic testbed described earlier. The sensor and the exploration task selection was performed "interactively" based on the particular experiment. Because exploration is a fairly involved and complex task, user assistance would allow the system to perform only those exploration tasks that are required for the particular experiment. An alternative would be to exhaustively try all the possible exploration tasks with all the available sensors. The active exploration system implementation allows the sensor and the task to be performed using the selected sensor to be specified by the user at each stage of exploration. At the end of each exploration step, the sensor and the task can be changed by the user. The extent to which each object is explored is also determined

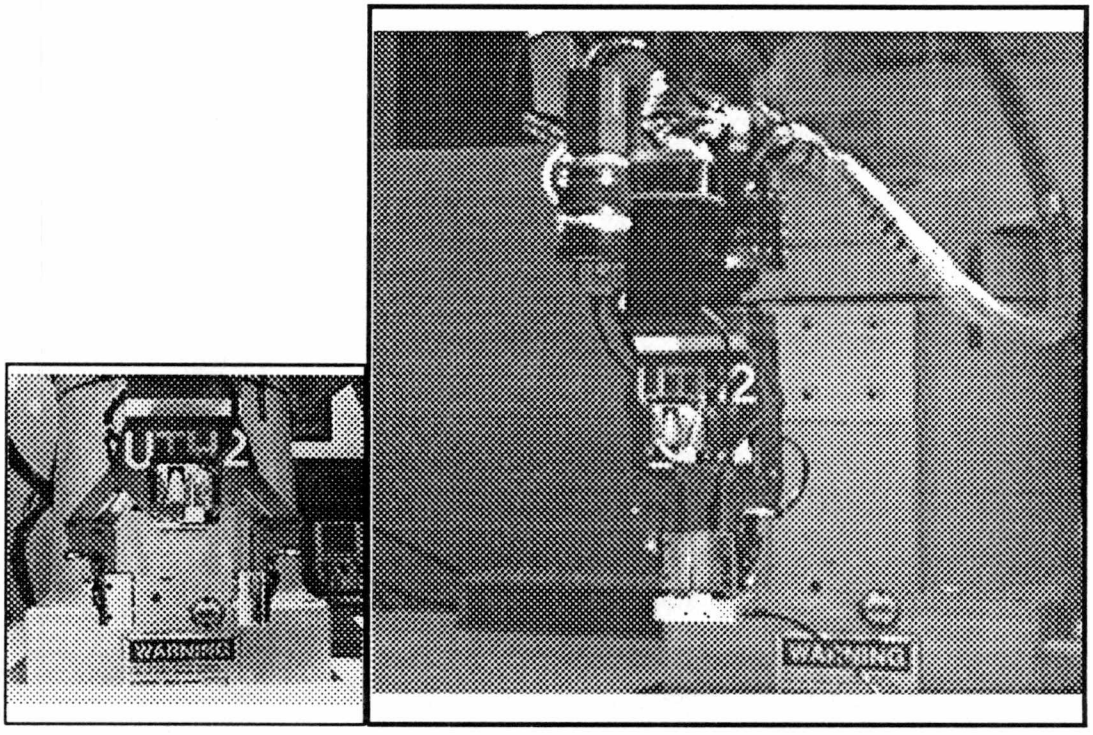


Figure 5.1: *Robotic manipulator holds the tool housing the tactile sensor, the vision and PLR sensors are located on the end-effector, and the force/torque sensor is located in the wrist.*

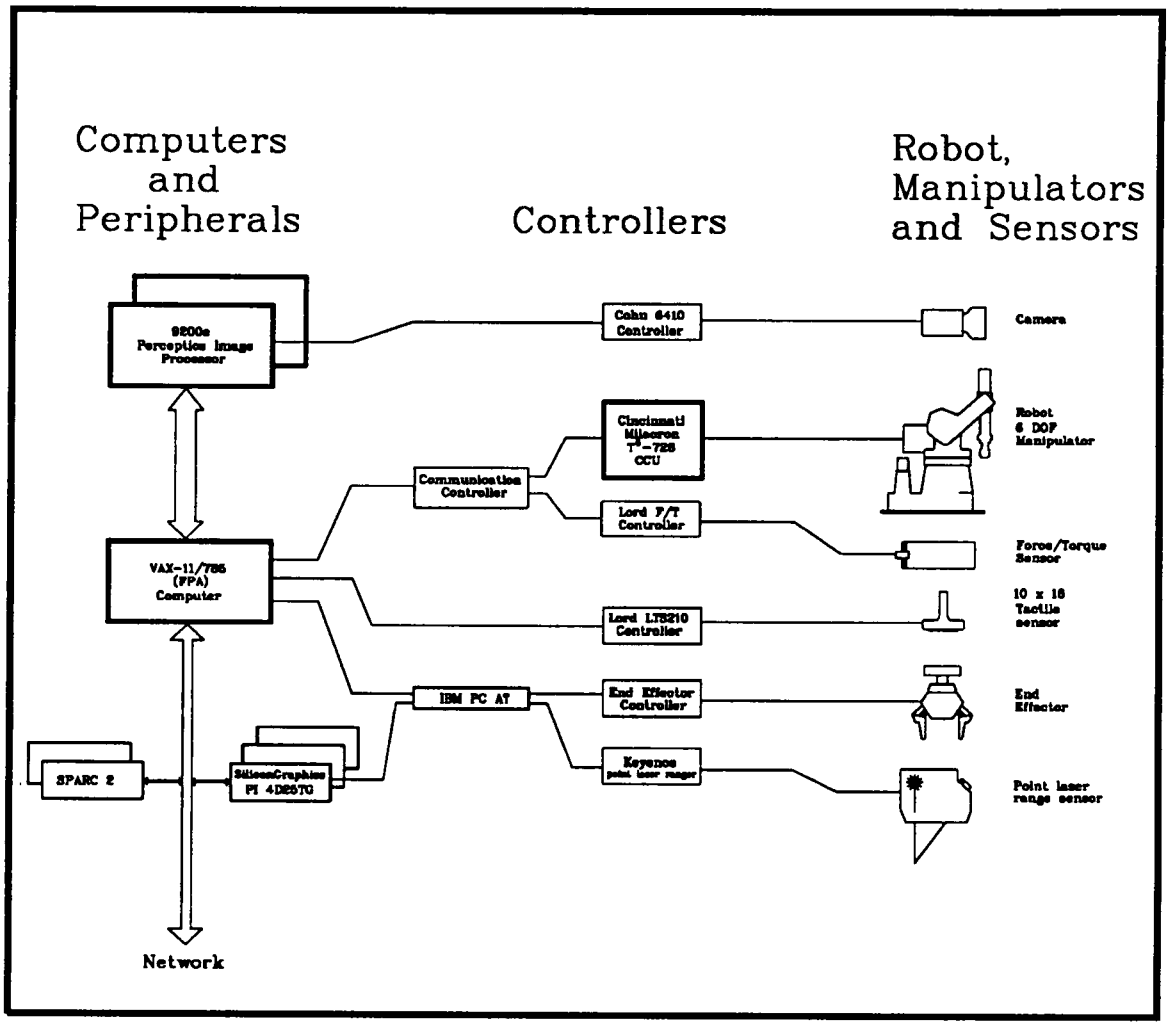


Figure 5.2: *Schematic layout of the robotic exploration system.*

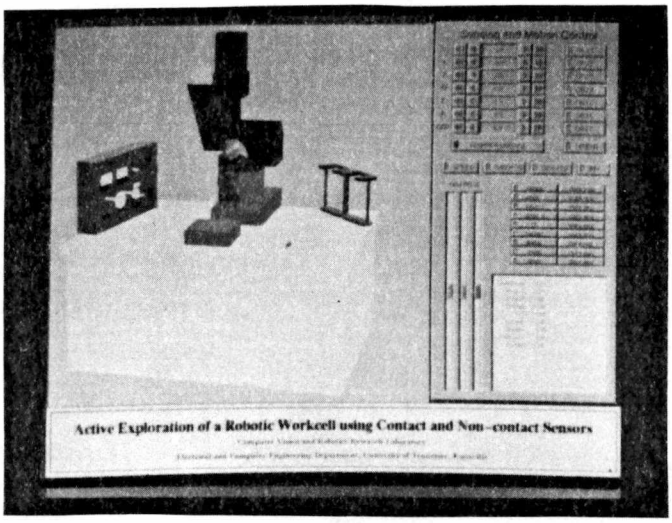


Figure 5.3: *The exploration results are displayed incrementally on a simulator that contains the simulated models of the robot and its work cell.*

interactively, because the present system does not include a means for analyzing the state of the generated model to automatically decide whether further exploration would be required or not.

5.3.1 Exploration of a simple polyhedral object

The object selected for this experiment is a simple polyhedral object consisting of eight surfaces, with no two surfaces parallel to each other (Figure 5.4). Figure 5.5 shows four distinct groups for the entire set of exploration tasks. The role played by the user in the sensor and exploration task selection is to ascertain that the exploration process proceeds in a manner that is best suited for the particular experiment. The user is not required to provide his/her judgment to assist in the data processing or data integration tasks of the exploration system. For example, if the actual number of objects in the scene is three and if the EM for extracting the number of objects using vision locates four objects, then such errors are left for the exploration system to rectify.

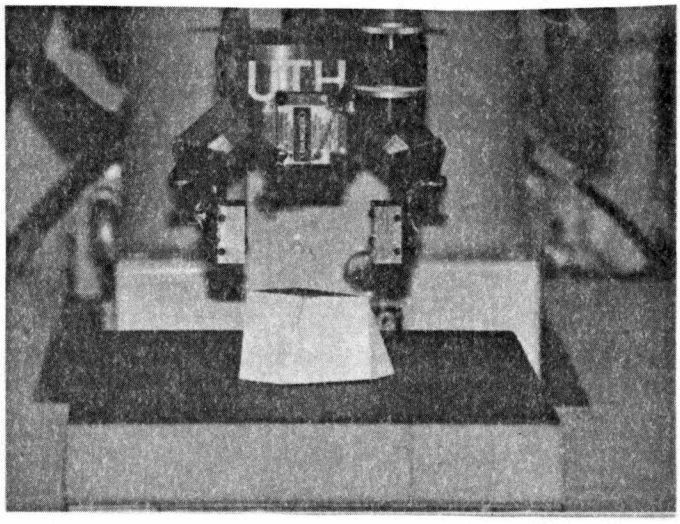


Figure 5.4: *Figure shows a polyhedral object with eight surfaces, where no two sides parallel to each other.*

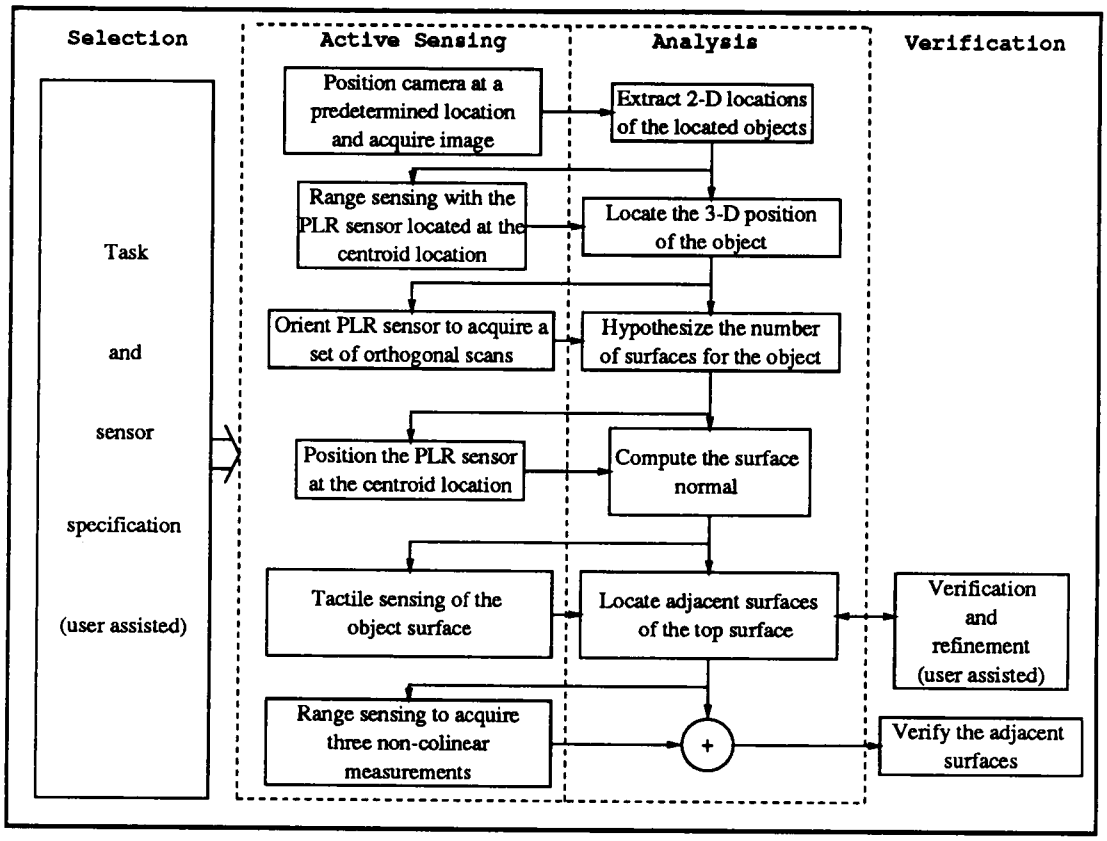


Figure 5.5: *Exploration tasks that were utilized to explore the simple polyhedral object. Note that the tasks are grouped into four categories based on their functionality.*

The exploration begins with acquiring a visual image of the work cell. To acquire the visual image the camera is moved to a predetermined location, which guarantees the entire work space to be included in the visual image (Figure 5.6). The visual image (Figure 5.7(a)), undergoes a thresholding operation (Figure 5.7(b)), and the 2–D bounding box and the location of the object (2–D) are determined. Once the object region is extracted by the thresholding process, the object model shown in the simulator consists of only the 2–D bounding box, indicating the localized rectangular region where the object is located (Figure 5.8).

Sensor selected: Vision

Number of objects located: 1

For object: 1

Bounding Box:

x_{min} = 23.563 x_{max} = 26.483

y_{min} = -3.282 y_{max} = 0.527

z_{min} = -19.870 z_{max} = -19.870

Centroid: 25.484 -1.657 -19.870

Note that the z_{min} and z_{max} values are equal after the bounding box computation. The next step is to determine the 3–D location of the centroid of the object region extracted by the thresholding process. The PLR sensor is moved to the 2–D centroid location, and the vertical height is chosen such that the entire object will be included in the dynamic range of the sensor. A point measurement is acquired to extract the 3–D location of the centroid.

Sensor selected: PLR

For object: 1

Centroid: 25.484 -1.657 -17.641

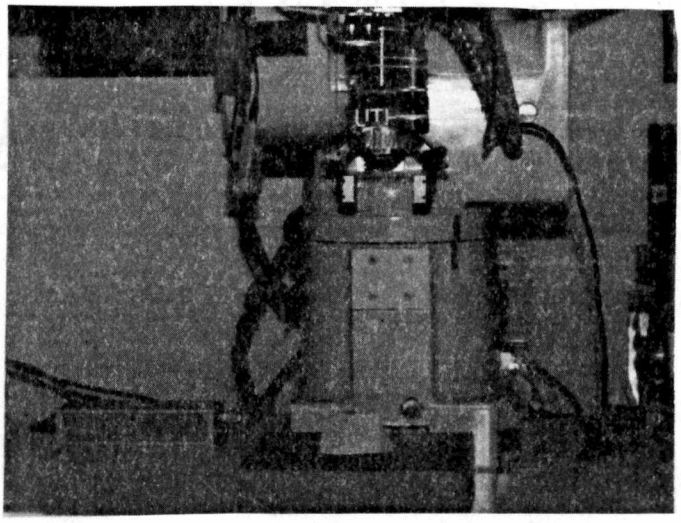
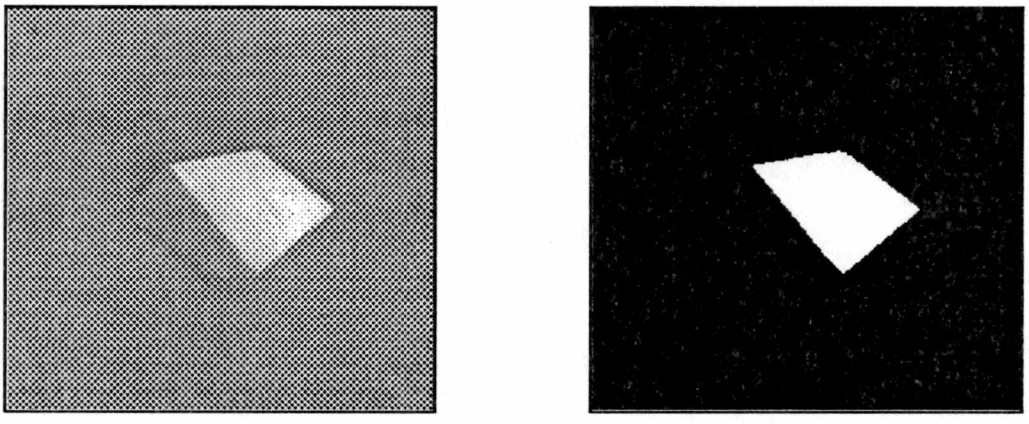


Figure 5.6: *To acquire the visual image the camera is moved to a predetermined location which guarantees the entire work space to be included in the visual image.*



(a) (b)

Figure 5.7: *(a) The original visual image is thresholded as shown in (b) thus enabling extraction of the object region from its background.*

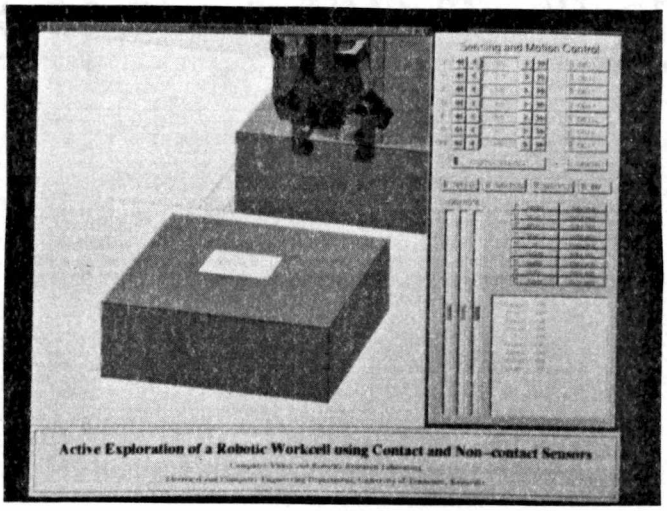


Figure 5.8: *The 2–D bounding box is displayed in the robot simulator, indicating the localized rectangular region where the object is located.*

To verify the 2–D bounding box and to update it to a 3–D bounding box, horizontal and vertical range scans are acquired using the PLR sensor. The horizontal and vertical scans are fairly coarse in resolution and are also used to provide an estimate for the number of object surfaces present (Figure 5.9).

Sensor selected: PLR

For object: 1

Number of surfaces located: 5

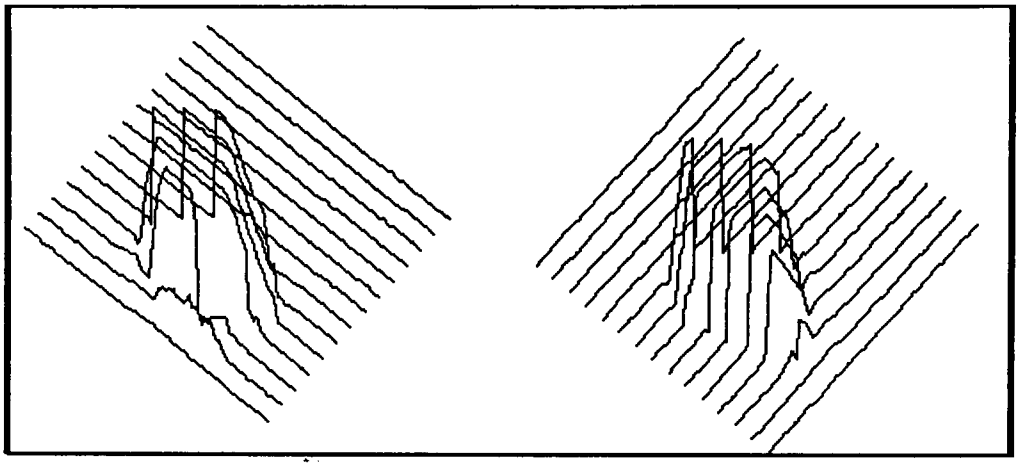
Bounding Box:

$x_{min} = 23.563$ $\quad$ $x_{max} = 26.483$

$y_{min} = -3.282$ $\quad$ $y_{max} = 0.527$

$z_{min} = -19.870$ $\quad$ $z_{max} = -17.560$

One of the preconditions for haptic exploration requires the tactile sensor to be aligned parallel to the surface that needs to be explored. This requires the surface normal to be computed. The surface normal was computed using PLR sensor measurements



(a) (b)

Figure 5.9: *To verify the 2-D bounding box and to update it to a 3-D bounding box, (a) horizontal and (b) vertical PLR scans are acquired.*

(Figure 5.10) from three non-collinear points.

Sensor selected: PLR

For object: 1

Surface Number: 1

Surface Normal:

$n_x = -0.084$ $n_y = 0.024$ $n_z = 0.996$

Surface Number: 2

Surface Normal:

$n_x = 0.00$ $n_y = 0.00$ $n_z = -1.00$

At this stage, the object model can be updated further to reflect the 3-D bounding box, where the upper surface has the appropriate orientation as computed by the surface normal extraction process (Figure 5.11). Now, the tactile sensor housed in its tool is picked up, and the top surface is explored haptically to extract the *pose* of the adjacent surfaces (Figure 5.12). The object centroid is used as the starting point for haptic

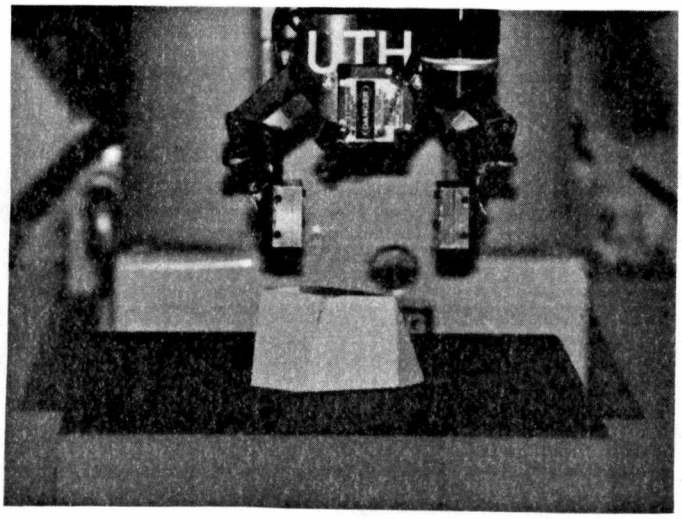


Figure 5.10: *The PLR sensor is moved over to the location of the centroid and measurements from three non–collinear points are acquired to compute the surface normal.*

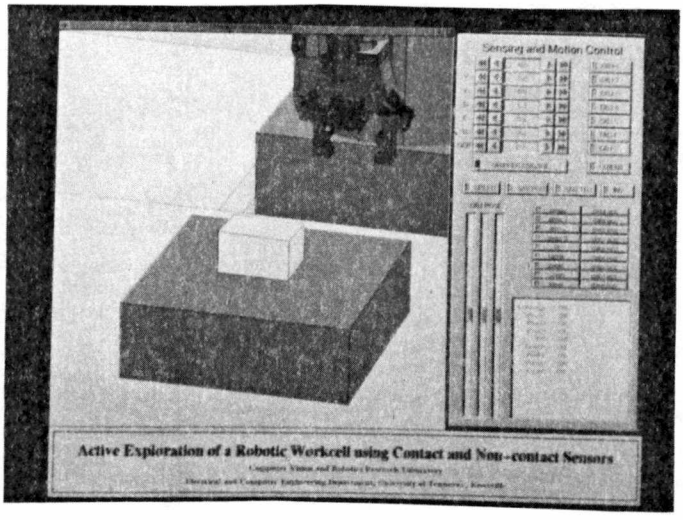
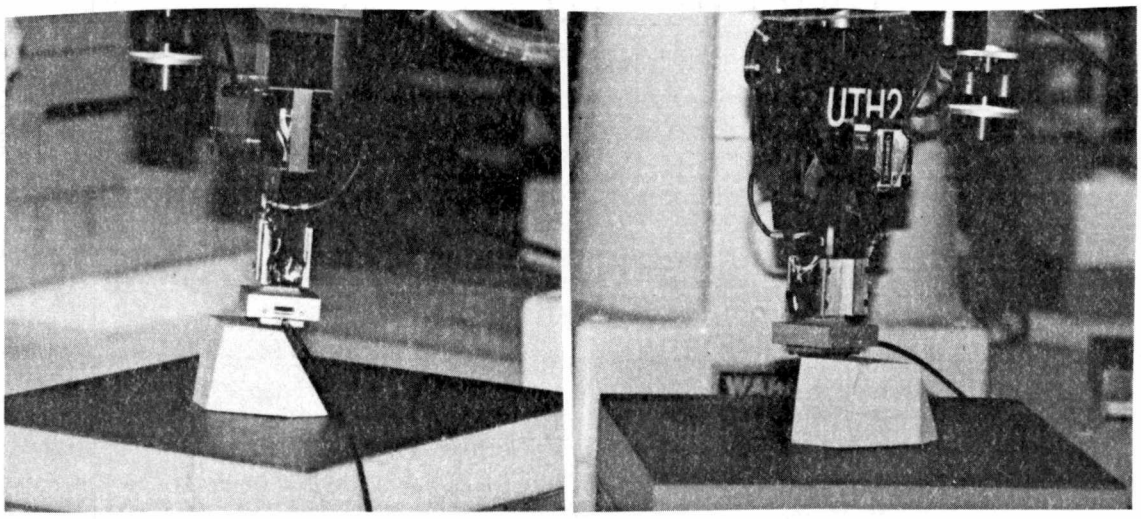
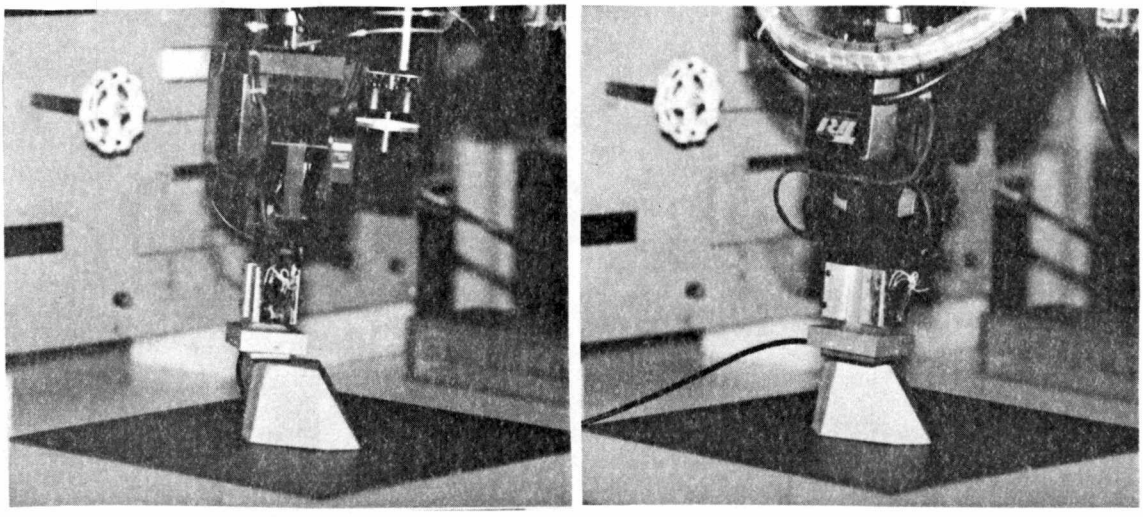


Figure 5.11: *The object model can be updated further to reflect the 3–D bounding box, where the upper surface has the appropriate orientation as computed by the surface normal extraction process.*

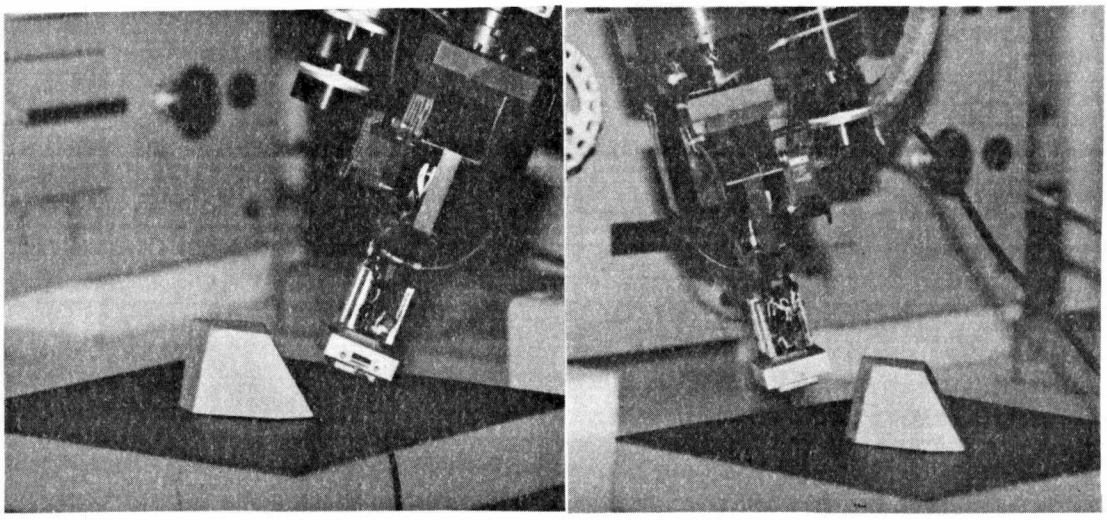


(a) (b)



(c) (d)

Figure 5.12: *Haptic exploration of the top surface of the object enables extraction of the pose of the adjacent surfaces.*



(a) (b)

Figure 5.13: *The end-effector is aligned along an orientation that would allow surface normal computation using the PLR sensor.*

exploration. The adjacent surfaces are hypothesized to be orthogonal to the surface being explored, and this hypothesis is later verified or corrected using the verification process using the PLR sensor. The closed-loop control allows verification of these hypothesized surfaces, where the robot is aligned along an orientation that would allow surface normal computation using the PLR sensor (Figure 5.13).

Sensor selected: PLR

For object: 1

Surface Number: 1

Surface Normal:

$n_x = -0.084$ $n_y = 0.024$ $n_z = 0.996$

Surface Number: 2

Surface Normal:

$n_x = 0.00$ $n_y = 0.00$ $n_z = -1.00$

Surface Number: 3

Surface Normal:

$n_x = -0.608$ $n_y = 0.605$ $n_z = 0.514$

Surface Number: 4

Surface Normal:

$n_x = -0.733$ $n_y = -0.679$ $n_z = -0.045$

Surface Number: 5

Surface Normal:

$n_x = 0.813$ $n_y = -0.577$ $n_z = 0.083$

Surface Number: 6

Surface Normal:

$n_x = 0.990$ $n_y = 0.118$ $n_z = 0.080$

The object model is updated again, where the final result is displayed at its appropriate *pose* (Figure 5.14).

The final result shown in Figure 5.14 and quantitative results shown in Table 5.1 indicate that the model is similar to the actual object in terms of size, position, and orientation. The number of surfaces and vertices located for this object model are not the same as the ones for the actual object. The error exists due to the inability of the robotic manipulator to reposition/reorient itself to perform haptic exploration of all the known surfaces to and check for any additional surfaces. However, the error associated with the adjacent surface orientations are within $\pm 12^o$. Note that these errors can be reduced further by verifying the edge orientation after each contact along an edge.

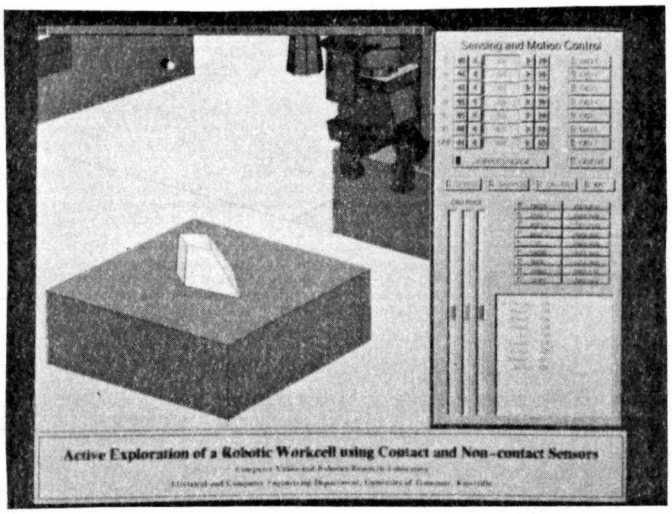


Figure 5.14: *The final result shows the object pose in the simulator corresponds to the actual object pose.*

Table 5.1: *Experimental results for the exploration of a simple polyhedral object.*

Geometric Properties Results	**Exploration from 3–D Model**	**Actual Values**
Number of surfaces	6	8
Number of vertices	8	10
Number of pairs of parallel surfaces	0	0
Number of edges located using haptic exploration	4	4
Relative orientation between edges located by haptic exploration (in degrees)	69.0, 82.32, 65.6, 134.0	60.0, 88.0, 76.0, 135.0

This would enhance the quality of results at the cost of the speed of exploration. By incorporating the PLR based adjacent surface verification process, the extraction of the true orientations of these surfaces is made possible.

5.3.2 Haptic exploration of multiple surfaces

The data acquisition and processing routines especially for *contact* sensors have been designed such that surfaces at arbitrary orientations can be explored. Surfaces oriented at arbitrary orientations necessitate the tactile sensor to be realigned to provide a flush contact with that surface. For the purpose of realignment, the surface normal can be computed either using the sensory feedback from force/torque sensors or the PLR sensor. In the previous experiment only the top surface was accessible for haptic exploration and the intent of this experiment is to show the ability of the system to haptically explore multiple surfaces, provided they are accessible for exploration. Further, all the surfaces of an object are not visible in the visual image or as a result of haptic exploration of the top surface alone. The complexity of an object might be such that certain surfaces can be located only after extensive exploration. The object (Figure 5.15) selected for this experiment has two surfaces that are accessible from the top and unless both of these surfaces are explored haptically, the true model of the object cannot be generated. The exploration of this object was carried out using the procedure shown in Figure 5.16.

The visual image (Figure 5.17(a)) is used to extract the location and the 2–D bounding box of the object (Figure 5.17(b)). The 2–D bounding box model is supplied to the simulator as shown in Figure 5.18. The 3–D location of the centroid of the object region is then extracted using the PLR sensor. The horizontal and vertical scans using PLR are then used to update the 2–D bounding box to a 3–D bounding box and to estimate

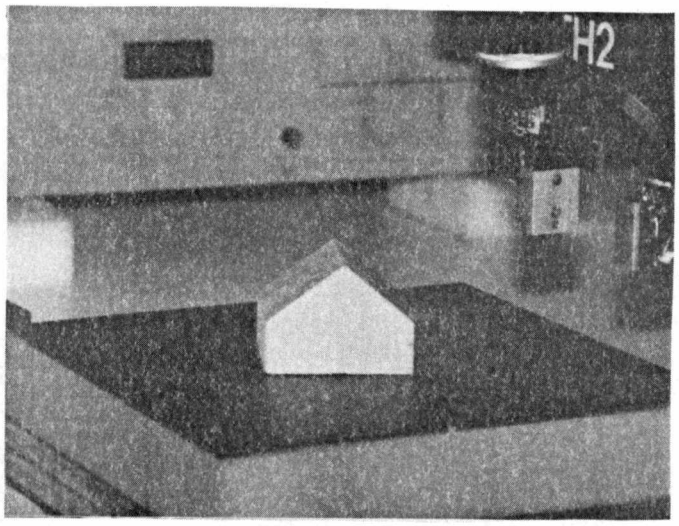


Figure 5.15: *The object has seven surfaces of which two of the upper surfaces require haptic exploration.*

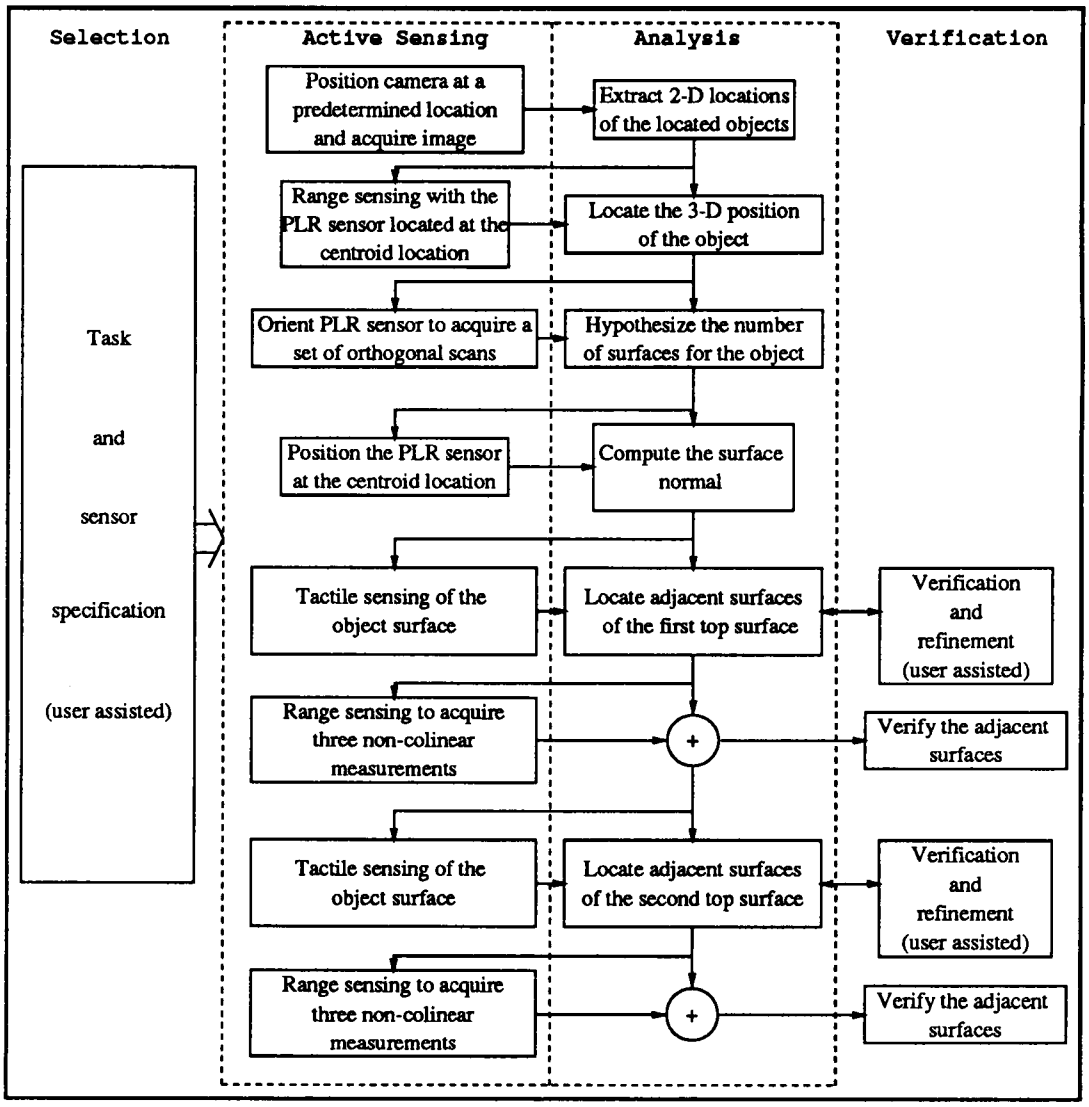
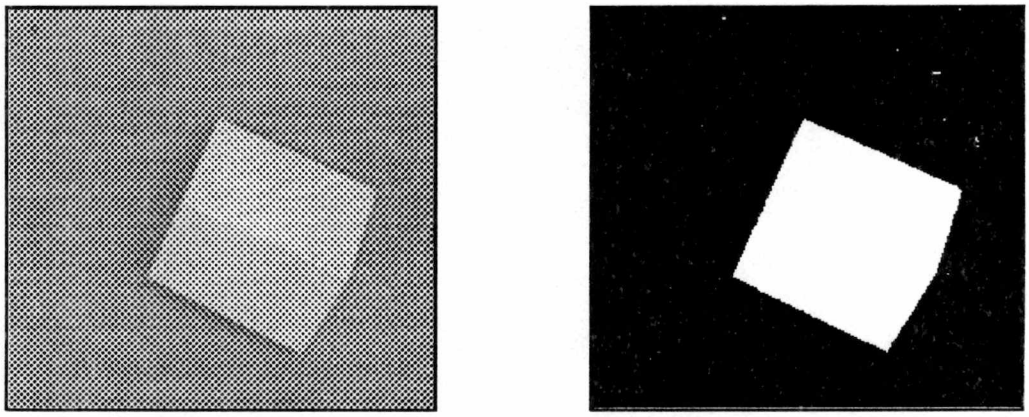


Figure 5.16: *Exploration tasks show that the adjacent surfaces for the two upper surfaces would be required to explore the polyhedral object completely.*



(a) (b)

Figure 5.17: *For the second polyhedral object, (a) the original visual image is thresholded as shown in (b) to extract the object region from its background.*

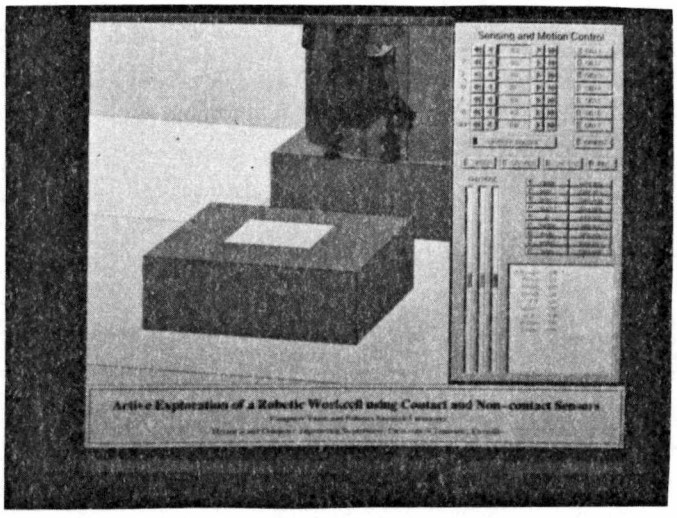
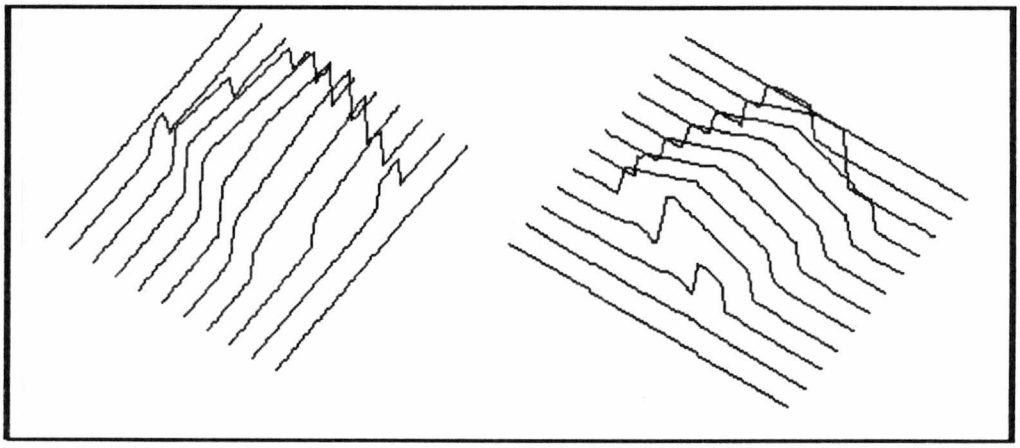


Figure 5.18: *The 2–D bounding box for the second polyhedral object, provides the location of the object in the workspace.*

the number of surfaces (Figure 5.19). The surface normal computation extracts the surface normal of one of the top surfaces of the object. Again, at this stage the simulator is updated to reflect the current model of the object, as shown in Figure 5.20.

Haptic exploration of this top surface provides the adjacent surface locations and orientations which are also verified by the PLR surface normal computation process (Figure 5.21). Following this step, the other top surface becomes visible in the model, and it can now be explored haptically. Haptic exploration of the second top surface isolates one of the side surfaces that was totally missing in the previous models that were generated (Figure 5.22). Thus, the final result shows all the surfaces that could be isolated by the combination of haptic exploration and PLR verification processes. Haptic exploration of the top surface yields the *pose* of adjacent surfaces. The *pose* of each individual adjacent surface is verified using the PLR sensor. Figure 5.21 shows the results generated by the Half–space modeler.

The final result is shown in Figure 5.22, and the quantitative results shown in Table 5.2 indicate that the final result is fairly accurate. The physical size, the number of surfaces, and the orientations of surfaces have been extracted quite accurately. However, as compared to the first experiment, the haptic exploration performed much better in terms of its accuracy. An important observation here is that haptic exploration allowed localization of a surface that was not located in the *non–contact* sensor based exploration. The verification of two of the smaller adjacent surfaces of the two sloping surfaces required additional assistance from the user to ascertain that the surface normals were computed accurately. This was required since these surfaces are relatively smaller and there exists a possibility that the end–effector would hit the platform dur-



(a) (b)

Figure 5.19: *The number of surfaces of the second object is greater than the first one, and here again the (a) horizontal and (b) vertical PLR scans are used to update 3–D bounding box and estimate the number of surfaces.*

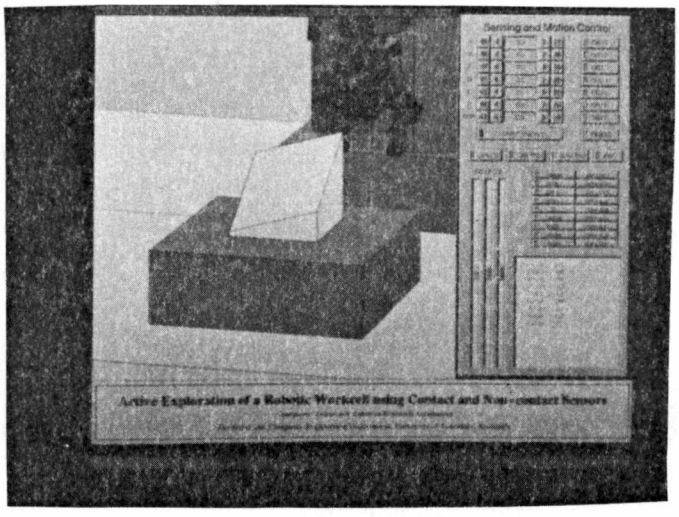


Figure 5.20: *The 3–D nature of the object can be observed following the surface normal extraction process.*

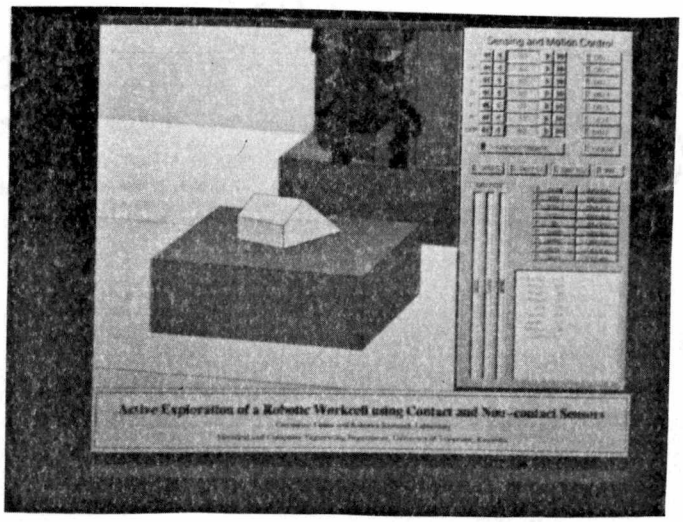


Figure 5.21: *The 3–D model for object clearly shows that subsequent to the haptic exploration of the first top surface, all the surfaces of the object have not been identified.*

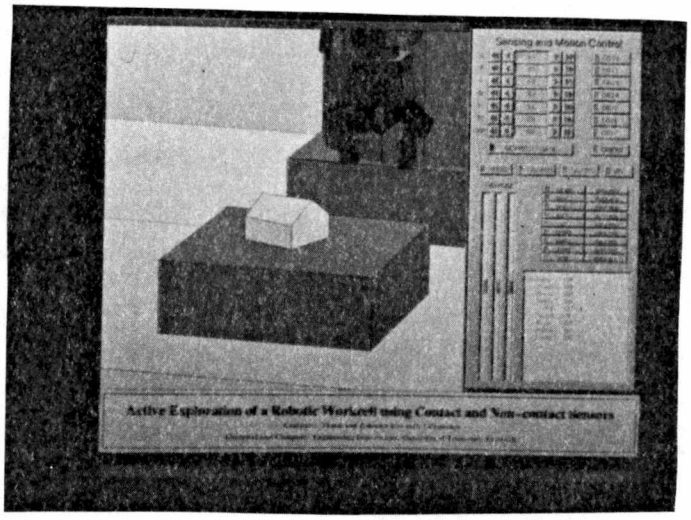


Figure 5.22: *The final result shows all the surfaces at their appropriate positions and orientations.*

Table 5.2: *Experimental results for the exploration of the polyhedral object requiring haptic exploration of two surfaces.*

Geometric Properties Results	**Exploration from 3–D Model**	**Actual Values**
Number of surfaces	7	7
Number of vertices	10	10
Number of pairs of parallel surfaces	2	2
Number of edges located using haptic exploration	7	7
Relative orientation between edges located by haptic exploration (in degrees)	$(1^{st} surface)$ 89.98, 88.76.0, 90.4, 88.35 $(2^{st} surface)$ 85.10, 89.8, 93.0, 92.65	$(1^{st} surface)$ 90.0, 90.0, 90.0 90.0 $(2^{st} surface)$ 90.0, 90.0, 90.0, 90.0

ing the verification process. Further, during the haptic exploration it was necessary to make sure that the object was not physically moved from its position when the robotic manipulator attempted to apply external force for acquiring tactile imprints.

5.3.3 Exploration of concave objects

Most geometric modelers model complex objects by performing either a union or intersection of a set of primitive object types. These primitive object types are generally convex objects. The purpose of this experiment is to demonstrate that both convex and concave objects can be explored using the active exploration system. Neither the exploration system nor the Half–space modeler has any a priori knowledge about the object is being either convex or concave. The object is determined to be concave when the modeler generates more vertices during the model generation process than what the exploration system can locate on a given surface. The concave object selected for this

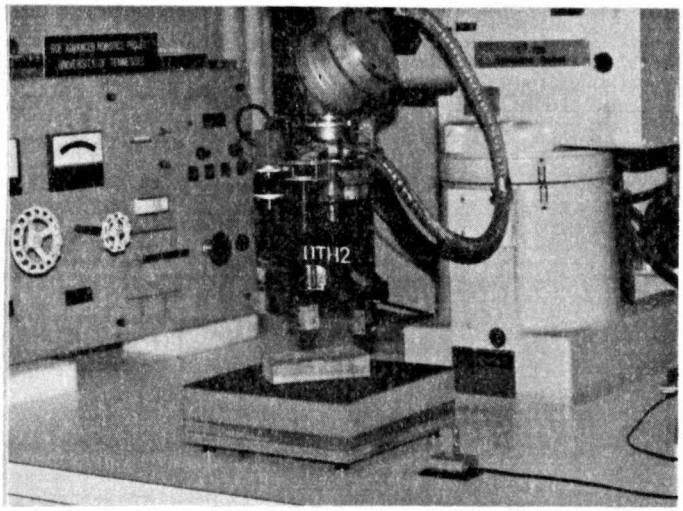


Figure 5.23: *The concave object selected for the exploration has nine surfaces.*

experiment (Figure 5.23) has nine surfaces. The 3–D model of this object is built using the exploration described in Figure 5.24.

The visual image of the workspace, consisting of a single concave object, is shown in (Figure 5.25(a)). The thresholding process provides the object region to compute the 2–D location and bounding box for the object (Figure 5.25(b)). The simulator shows (Figure 5.26) a large 2–D bounding box corresponding to the object region from the thresholded image. The next step involves determining the 3–D location of the centroid and the material type. This is accomplished by using the combination of tactile and force/torque sensors.

The tactile sensor is moved to the 2–D location of the centroid, and the sensor is lowered until the force/torque sensor detects a large reaction force and stops the manipulator motion. At this point if the tactile imprint shows a flush contact, then the 3–D location of the object surface is extracted. The manipulator then applies additional force to test if the object is compliant or non–compliant. The horizontal and vertical

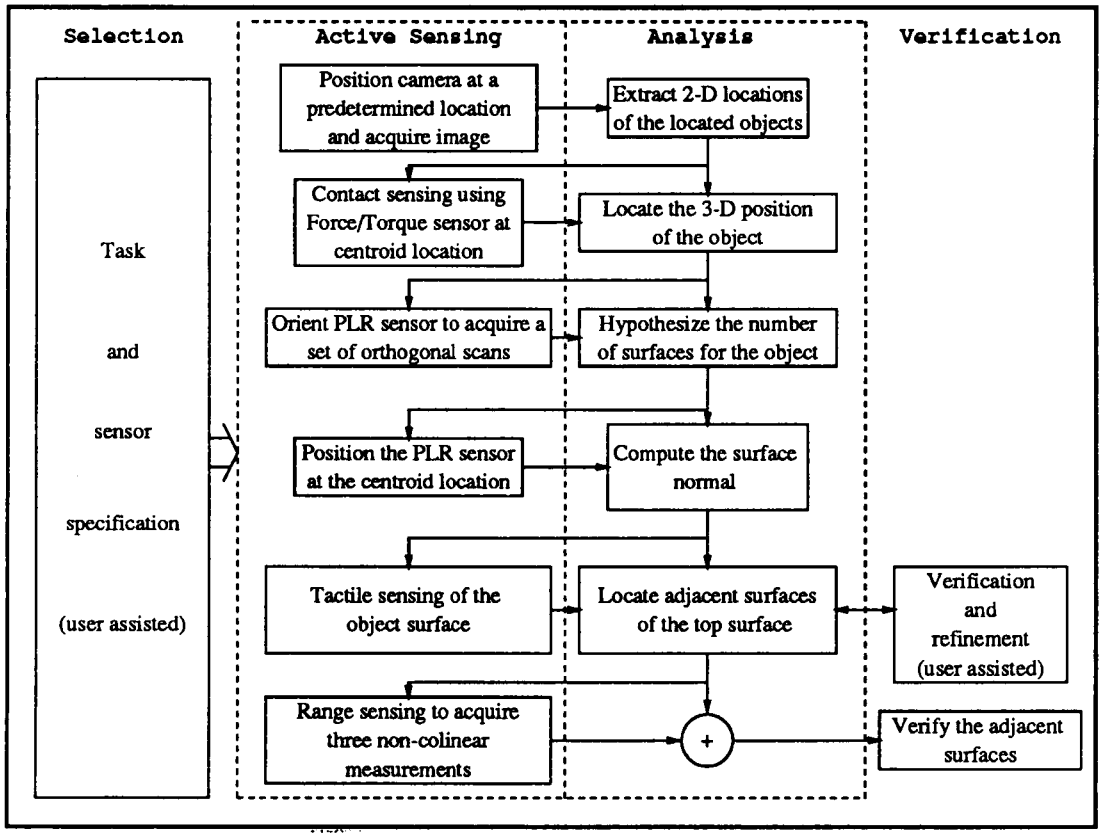
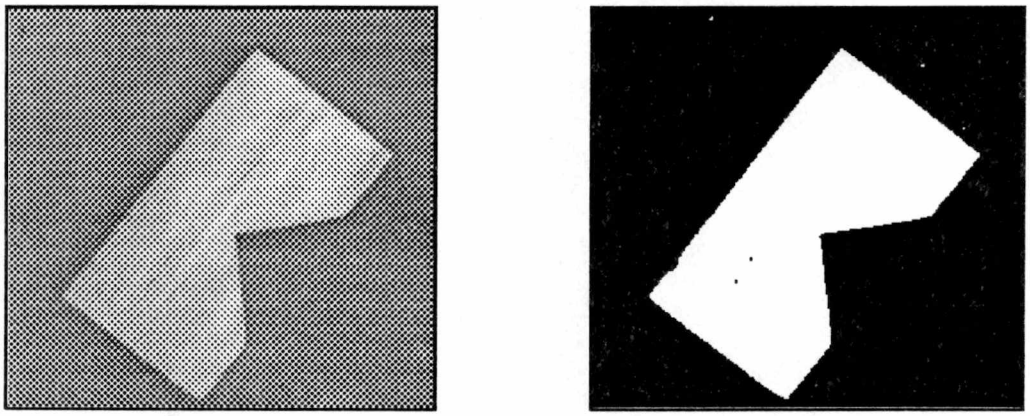


Figure 5.24: *Figure shows the sensing and analysis task sequence during the active exploration of a concave object. Note that this sequence is different from the one utilized for experiment 2.*



(a) (b)

Figure 5.25: *(a) The original visual image is thresholded as shown in (b) thus enabling extraction of the object region from its background.*

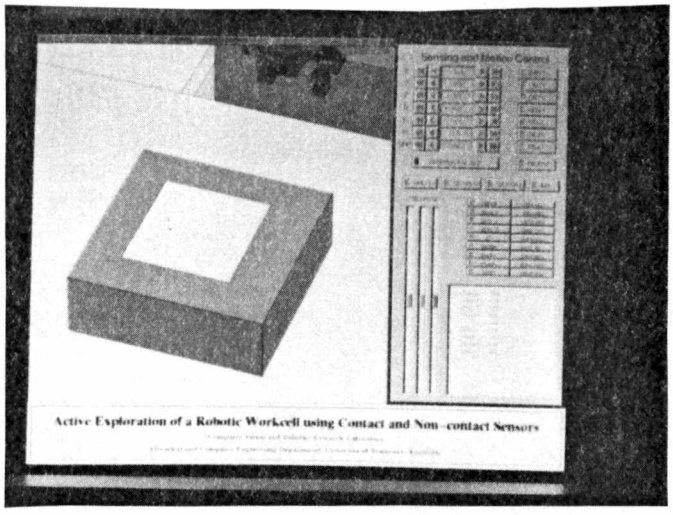
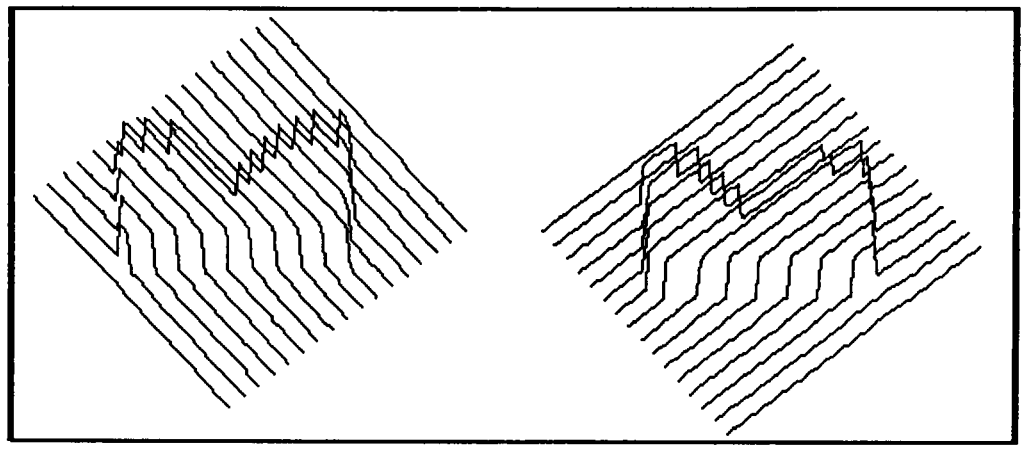


Figure 5.26: *The 2-D bounding box provides the localized rectangular region where the object is located.*



(a) (b)

Figure 5.27: *The (a) horizontal and (b) vertical PLR scans are used to update the 3–D bounding box and estimate the number of surfaces.*

scans (Figure 5.27) provide an estimate for the number of surfaces of the object, and also the 3–D bounding box gets updated. The surface normal is then computed using the PLR sensor and the updated model at this stage is shown in Figure 5.28.

Haptic exploration of the top surface yields the *pose* of adjacent surfaces. The *pose* of each individual adjacent surface is verified using the PLR sensor. Figure 5.29 shows the final result generated by the Half–space modeler.

The exploration of the concave object shows that the final result has preserved the size, number of surfaces, and orientation. Table 5.3 shows various parameters computed from the model and the corresponding values of the actual object. Depending on the complexity of the concave object, additional verification is sometimes required.

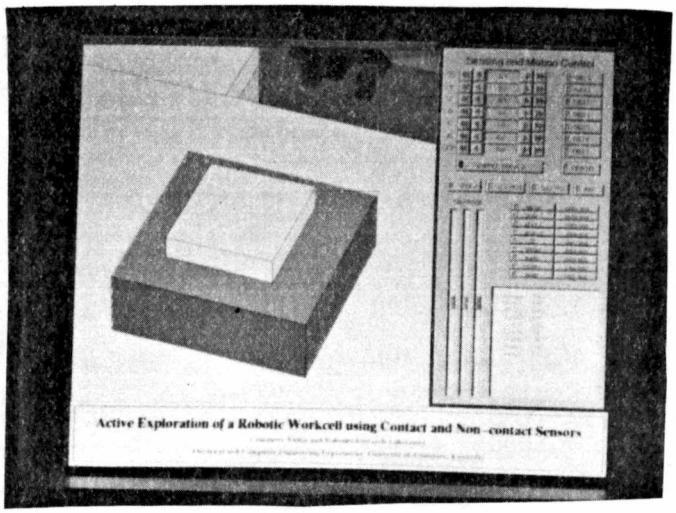


Figure 5.28: *The 3–D nature of the concave object is visible in the simulator.*

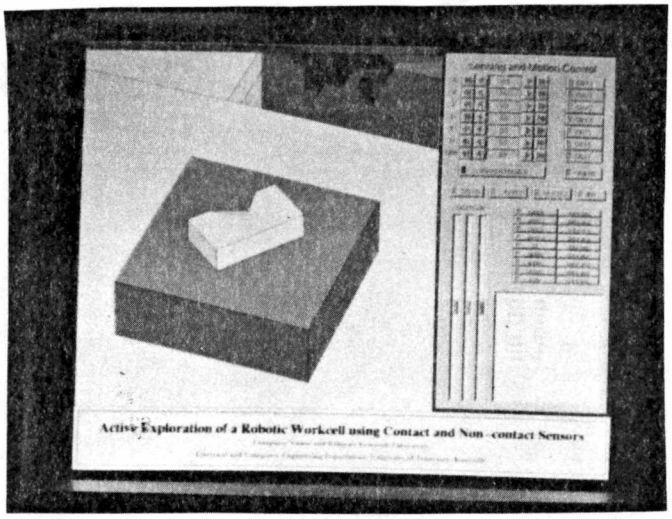


Figure 5.29: *The final result for the concave object after haptic exploration and verification of adjacent surfaces.*

Table 5.3: *Experimental results for the exploration of concave objects.*

Geometric Properties Results	Exploration from 3–D Model	Actual Values
Number of surfaces	9	9
Number of vertices	14	14
Number of pairs of parallel surfaces	4	4
Number of edges located using haptic exploration	7	7
Relative orientation between edges located by haptic exploration (in degrees)	89.98, 135.3.0, 90.4, 135.1, 89.3, 90.0, 90.7	90.0, 138.0, 93.0, 136.0, 90.0, 90.0, 90.0

5.3.4 Exploration of thin planar non–compliant objects

The sensors utilized in the active exploration system are such that a single sensory modality cannot uniquely determine whether a given object is concave, convex, compliant, non–compliant, or any combination of these. Thus, by employing multiple sensors, the versatility of a robotic system is enhanced even further. In the past, researchers have not studied the problem of modeling 3–D objects that are thin or lack significant height, but this experiment is directed towards emphasizing the fact that once an object is determined to be a thin planar polyhedral object, then certain information can still be displayed on the simulator. However, the localized 3–D information can be represented in terms of a composite tactile imprint. Two thin planar objects were selected (Figure 5.30), and the exploration procedure for this experiment is shown in Figure 5.31.

A wrench and a key were arbitrarily placed in the workcell, and the visual image (Figure 5.32(a)) after thresholding (Figure 5.32(b)) provides the 2–D bounding boxes.

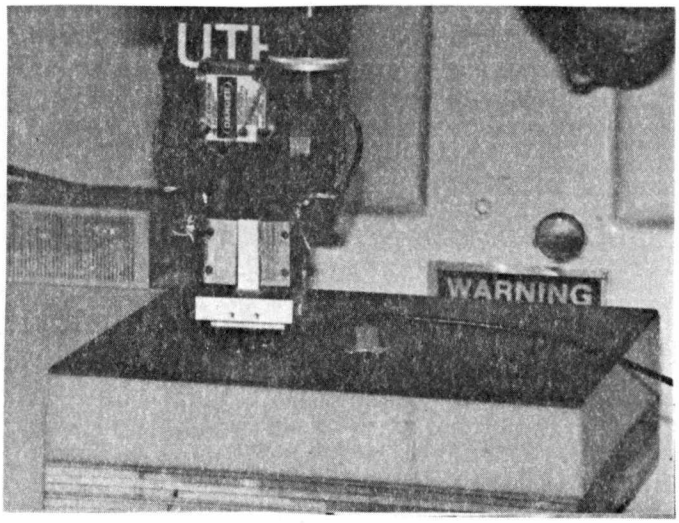


Figure 5.30: *The two thin planar objects were placed arbitrarily in the robotic workcell.*

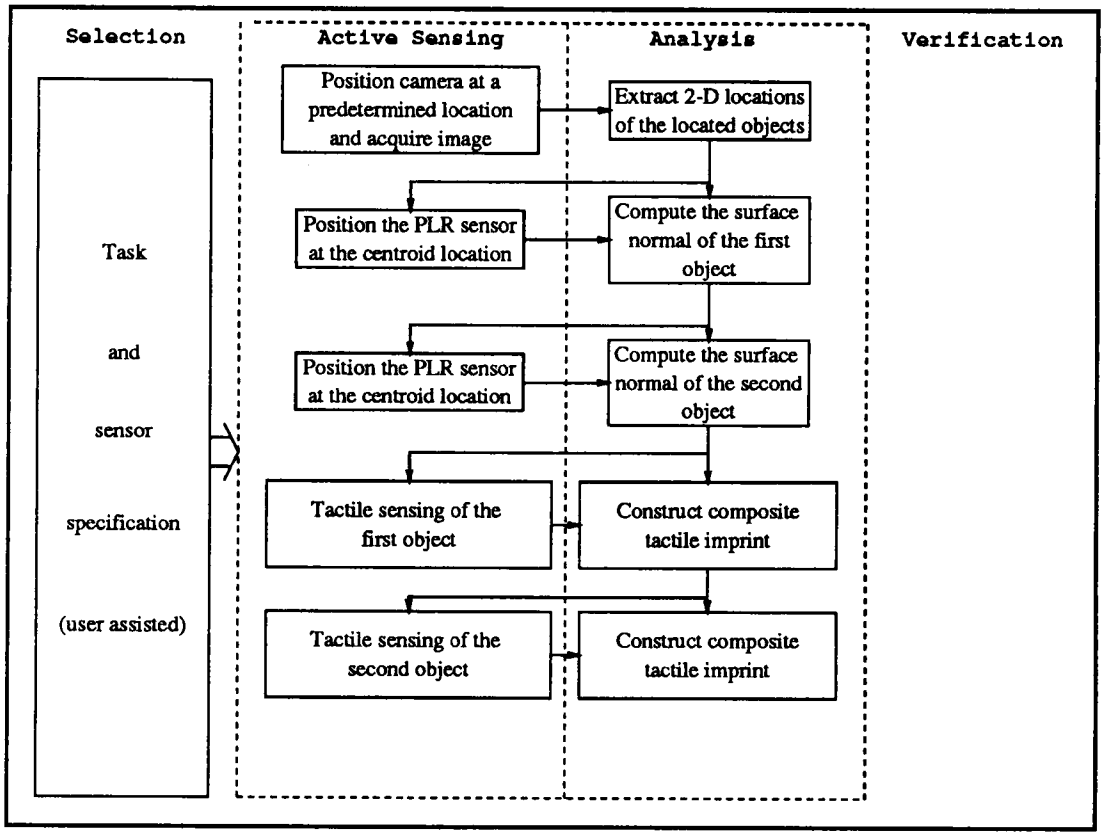
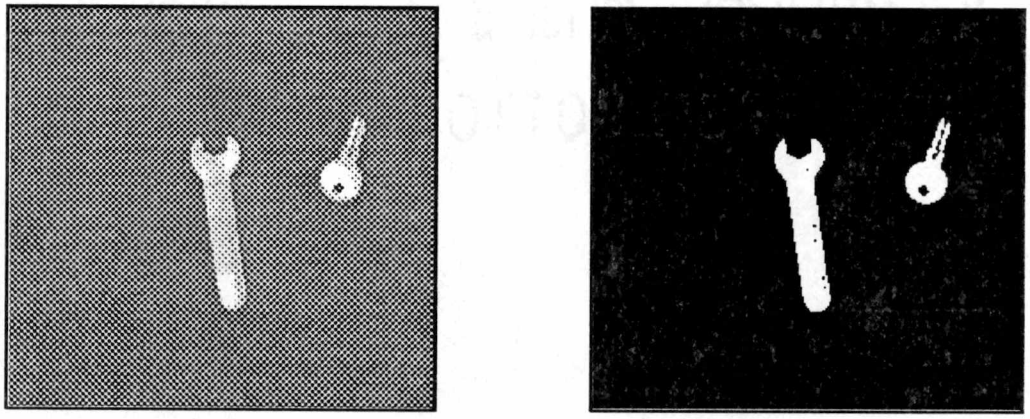


Figure 5.31: *Exploration tasks utilized to explore the thin planar polyhedral objects. Note that the nature of haptic exploration utilized here is different from the one used for the previous three experiments.*



(a) (b)

Figure 5.32: *(a) The visual image is thresholded as shown in (b) thus enabling extraction of the object region from its background.*

The simulator is updated at this time to show the 2–D bounding box models (Figure 5.33). Using the PLR sensor, the surface normals of each of the objects are extracted, and based on the height of the objects, it is then determined that further exploration cannot be performed (Figure 5.34). The composite tactile imprints for the two objects are then generated by acquiring multiple tactile imprints (Figures 5.35 and 5.36).

5.3.5 Exploration of compliant objects

Haptic exploration requires objects to be rigidly placed, and they should not deform when external force is applied during exploration. The objects that deform, i.e., non–rigid or compliant objects, do not provide clear tactile imprints that can then be used to extract object contours. Thus, for the contour following process, the objects need to be rigid and be able to provide a clear tactile imprint. However, non–contact sensors are incapable of determining the material properties of an object. Thus, the *non–contact* sensor based exploration processes can be performed on objects that are compliant or non–compliant, thereby, implying that in case of compliant objects, the 3–D model building will be based only on the information provided by the *non–contact* sensors. Such a model will be an approximate model rather than a fairly accurate model. The objects used in this experiment are shown in Figure 5.37, and these objects are made of a soft foam–like material. One of the objects has a rectangular cross–section, whereas the other has a circular cross–section. The steps involved in the exploration of these objects is shown in Figure 5.38.

The visual image (Figure 5.39(a)) allows extraction of two distinct object regions (Figure5.39(b)). The two object models are now updated and the simulator indicates the corresponding object regions (Figure 5.40).

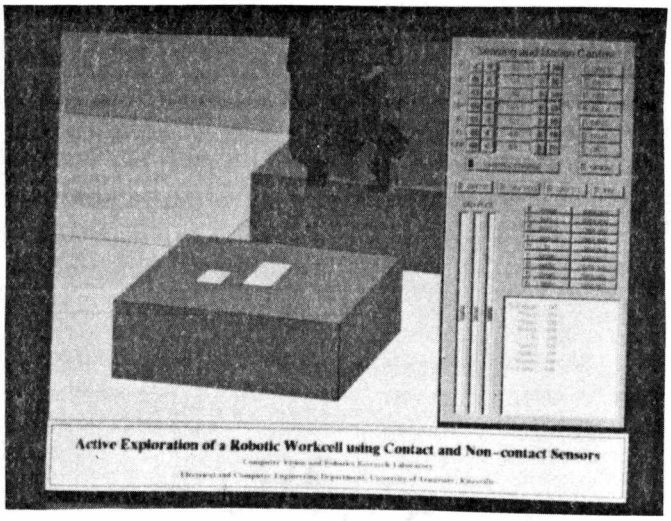


Figure 5.33: *The 2–D bounding box model is shown in the simulator to indicate the location of the object.*

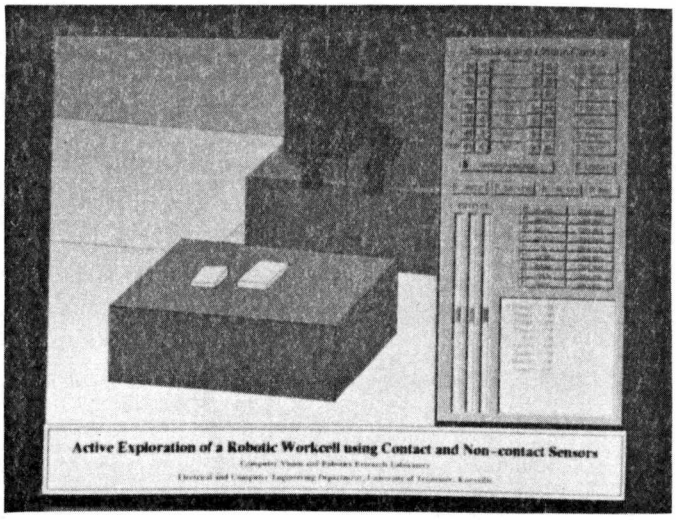


Figure 5.34: *The simulator shows the 3–D bounding box for the two thin planar objects.*

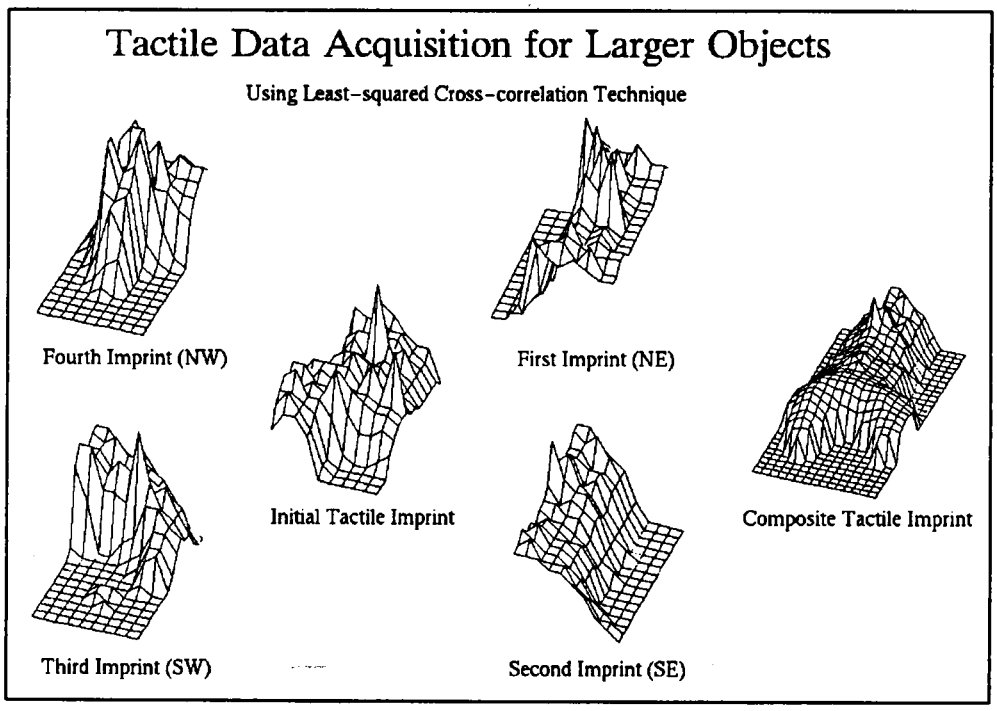


Figure 5.35: *The composite tactile imprint for the wrench.*

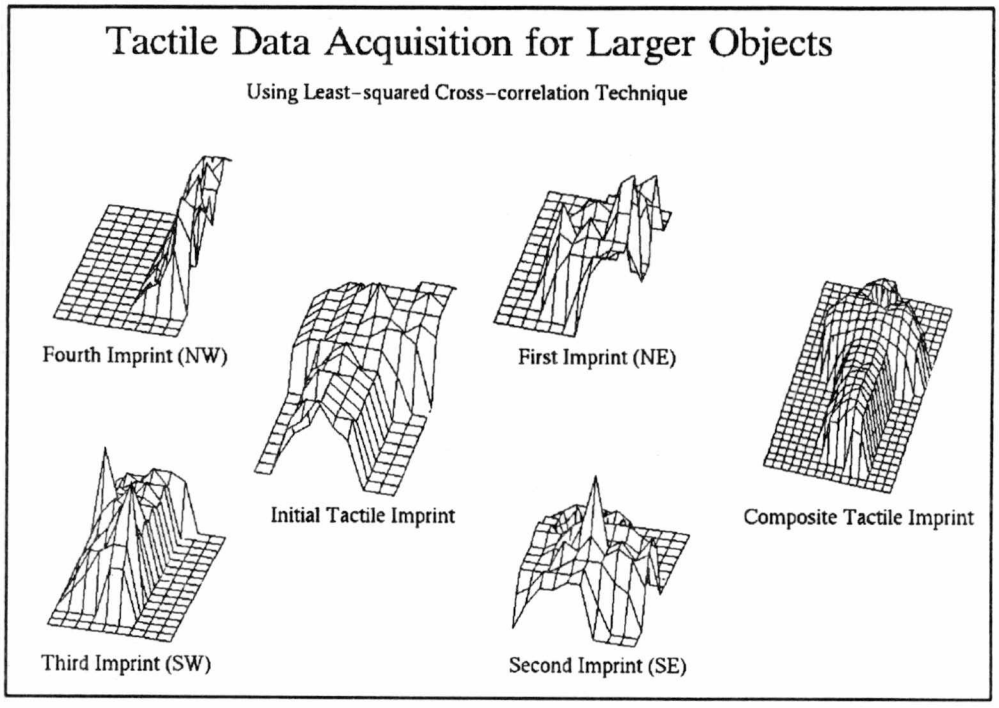


Figure 5.36: *The individual tactile imprints are combined using least-squared cross-correlation technique to generate the composite tactile imprint.*

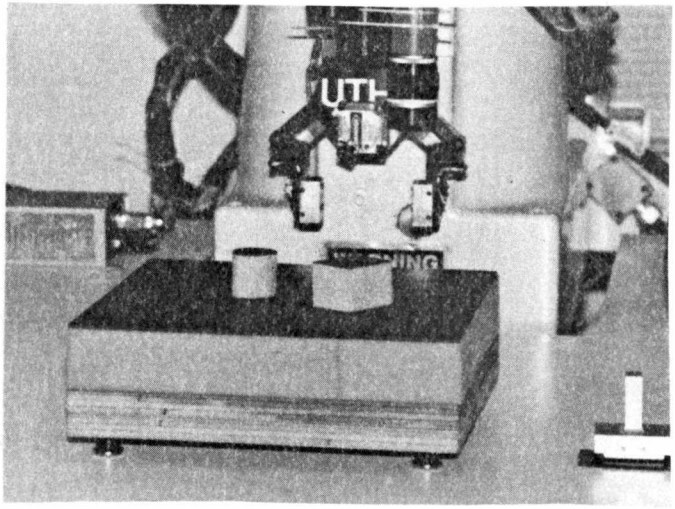


Figure 5.37: *The two compliant objects were placed at arbitrarily locations in the robotic workcell.*

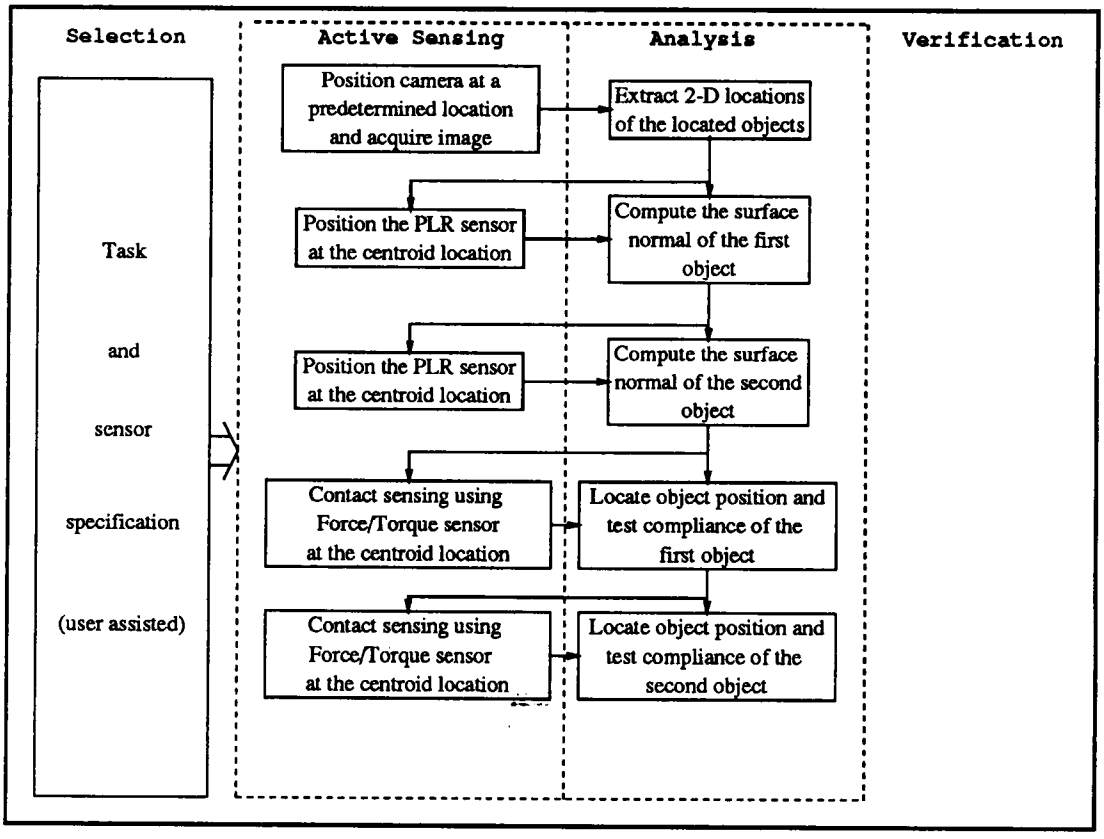
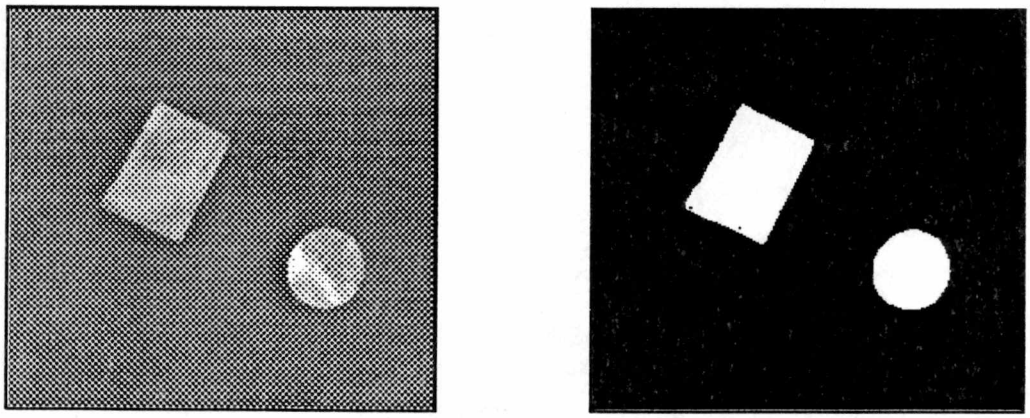


Figure 5.38: *The exploration tasks indicate that extensive exploration is not utilized to explore the compliant objects.*



(a) (b)

Figure 5.39: *(a) The visual image is thresholded as shown in (b) thus enabling extraction of the two object regions from their background.*

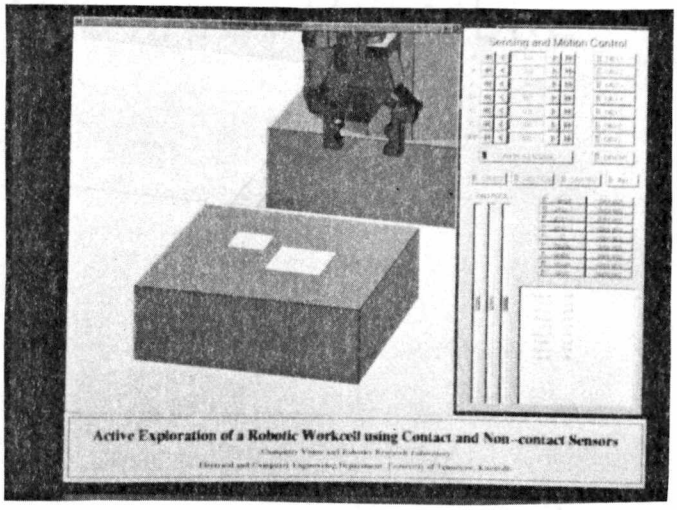


Figure 5.40: *The 2-D bounding box corresponding to the two objects are then displayed in the simulator, indicating their respective locations.*

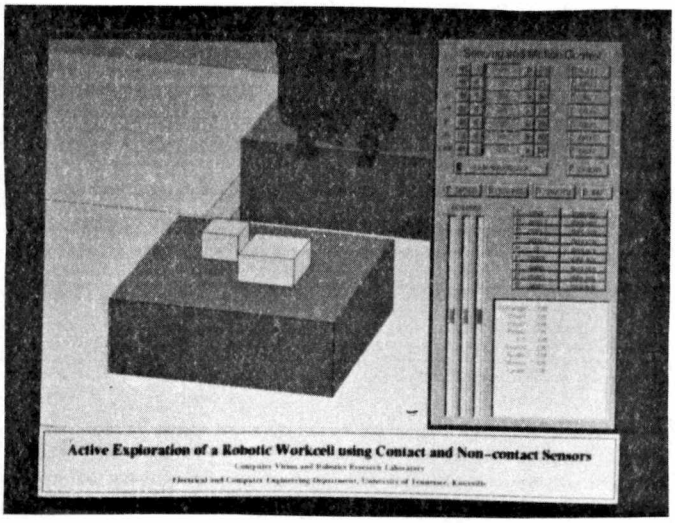
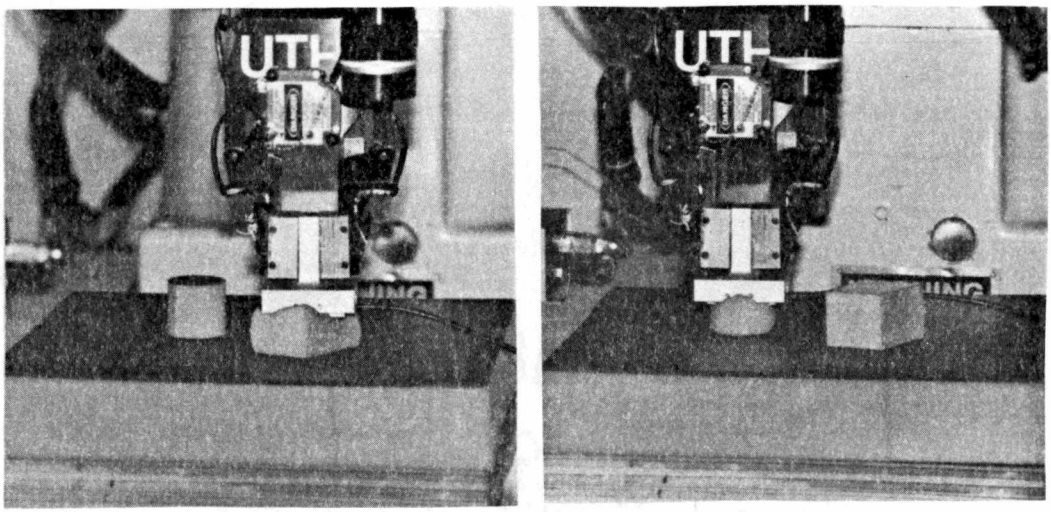


Figure 5.41: *The surface normals of the two objects are then extracted using the PLR sensor measurements.*

The surface normals of the two objects are then extracted using the PLR sensor measurements. Figure 5.41 shows the updated view of the models after surface normal computation. The 3–D location of the top surface can be verified using the combination of tactile and force/torque sensors. The tactile sensor is moved to the centroid (2–D) location of the object region, and then the arm gradually lowers the tactile sensor. The force/torque sensor stops the motion of the manipulator as soon as the reaction force, due to the contact with the object, exceeds a predetermined threshold. The arm attempts to exert additional force to verify if the object in contact is compliant or non–compliant (Figure 5.42). This identification is based on the force/torque sensor measurements and displacement of the tactile sensor if any. When the manipulator applies additional force after the initial contact, if the object is compliant, then there will be further displacement of the tactile sensor. For the objects used for this experiments the displacement was 0.25 in. The reaction force values along z direction changed from 87 oz to 128 oz. Once the object is determined to be compliant (or non–rigid), the



(a) (b)

Figure 5.42: *The arm attempts to exert additional force to verify if the object in contact is compliant or non–compliant.*

object is not explored any further.

5.3.6 Exploration of workcell containing multiple objects

In the above experiments the details associated with the exploration of objects belonging to each of the four categories of polyhedral objects was provided. Typically, the workcell of a robotic system does not include objects from only one of these four categories, and a more probable scenario would be where there are objects belonging to more than one category. In such scenarios, the active exploration system would be required to perform different exploration tasks based on the category of the object. The modularity of the architecture, sensor specific exploration modules, closed–loop control mechanism, and incremental data integration are the key principles behind the ability to explore a variety

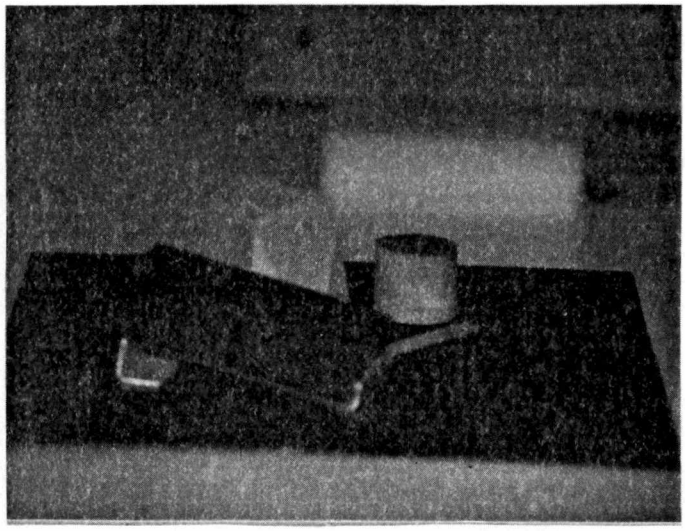


Figure 5.43: *The four objects selected for this experiment include two rigid convex objects, one compliant object, and one thin planar object .*

of objects using the active exploration system described in this research. Four objects were selected for this experiment, which include two rigid convex objects, one compliant object, and one thin planar object (Figure 5.43), and the exploration sequence is shown in Figures 5.44–5.46.

The four objects are isolated using the visual image (Figure 5.47(a)) and performing the thresholding operation (Figure 5.47(b)). The 2–D bounding boxes for each of the four individual objects (Figure 5.48) are now displayed through the simulator.

The *non–contact* and *contact* sensor based exploration procedure outlined in Figures 5.44–5.46 is performed on each of the four objects. Except for the thin planar object the horizontal and vertical range scans are acquired for the three larger objects (Figures 5.49–5.51). The results available after each exploration task are incrementally integrated, and the results of integration are shown in Figures 5.52–5.54. Note that for the thin planar object only its 2–D bounding box model is displayed in the simulator.

The quantitative results for the two rigid objects are shown in Table 5.4 and Table

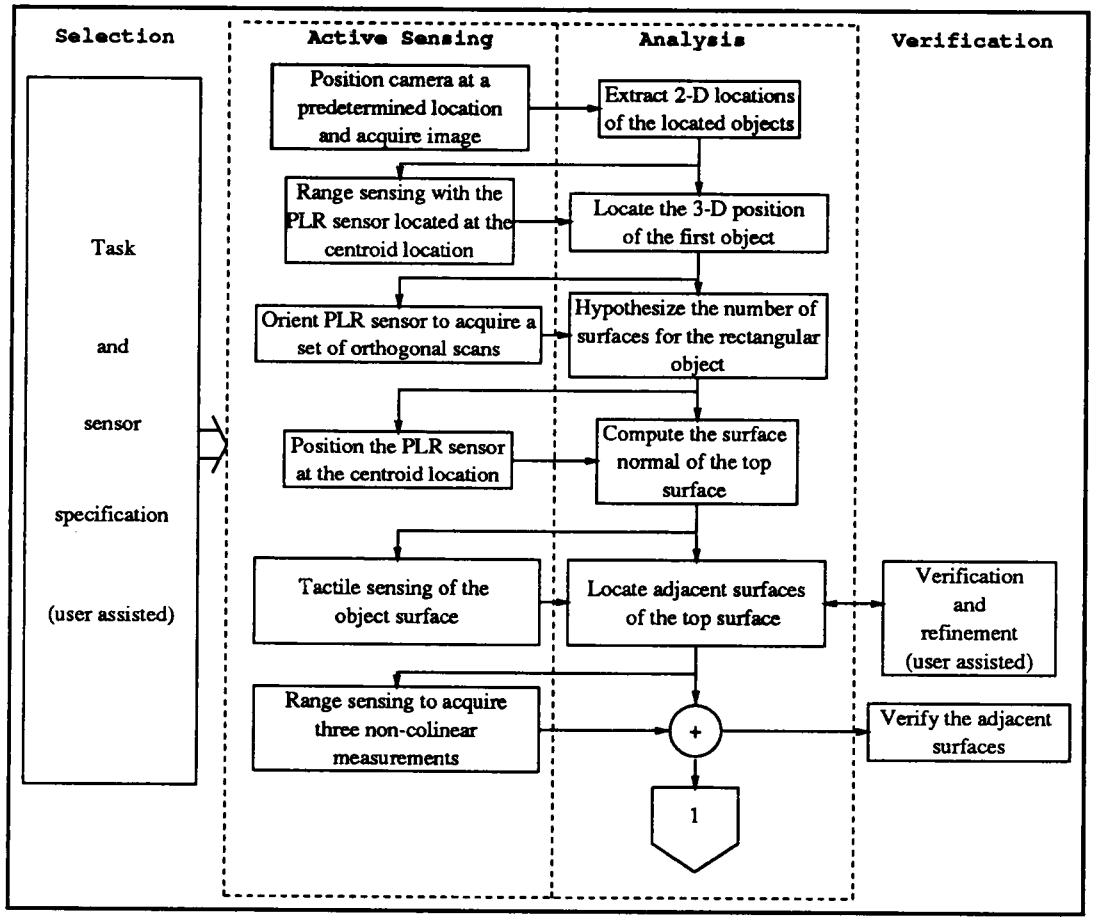


Figure 5.44: *Exploration tasks that were utilized to explore the rectangular object in the workspace containing multiple polyhedral objects.*

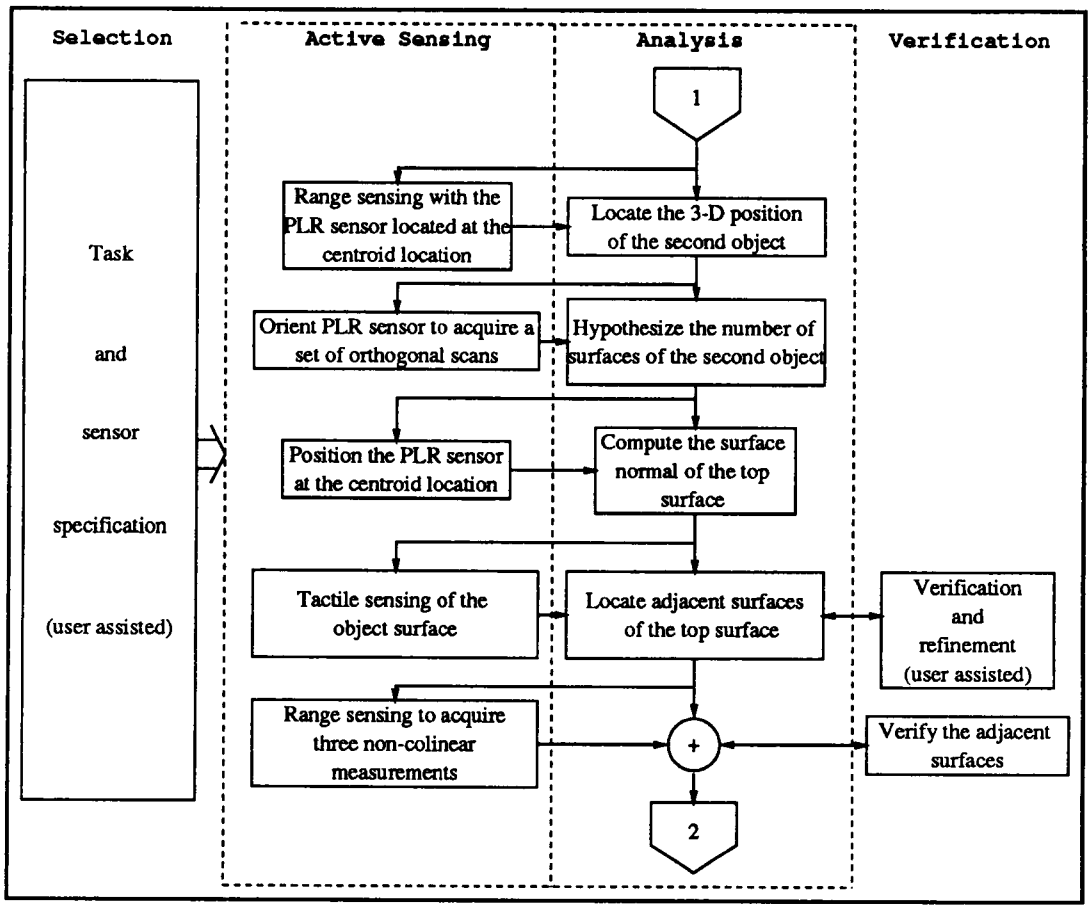


Figure 5.45: *Figure shows the exploration sequence used to explore the second object.*

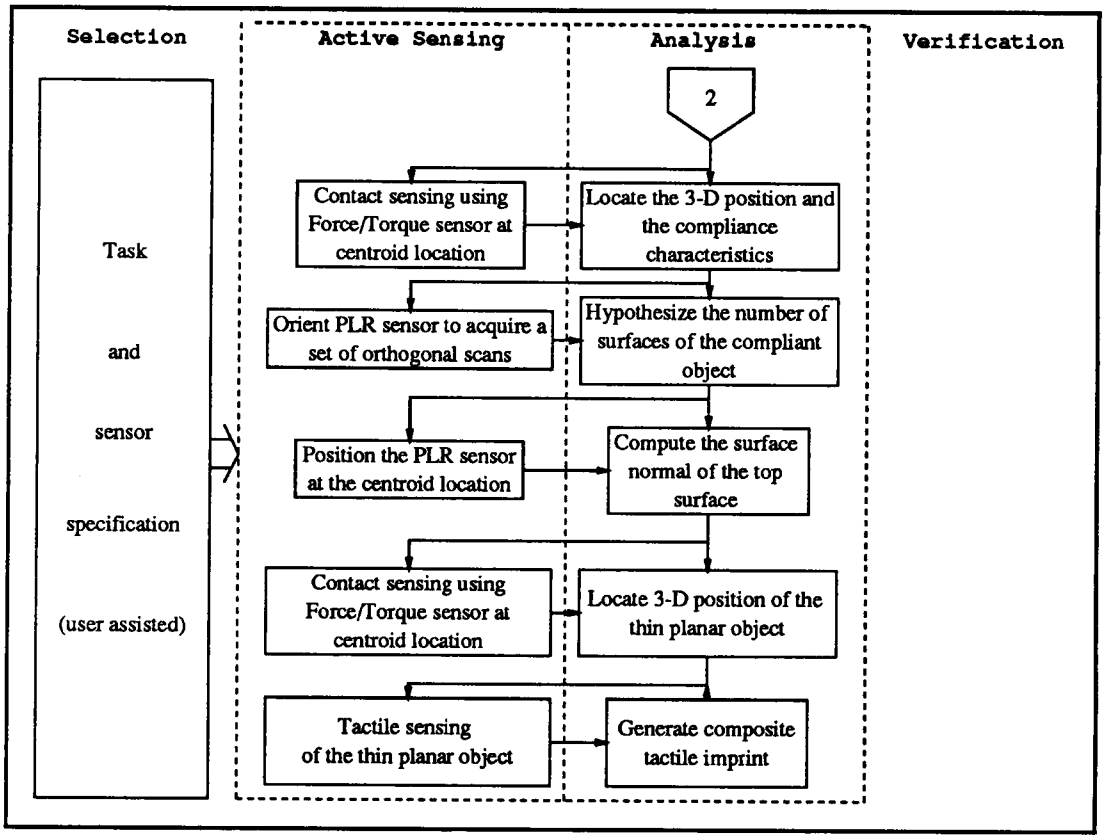
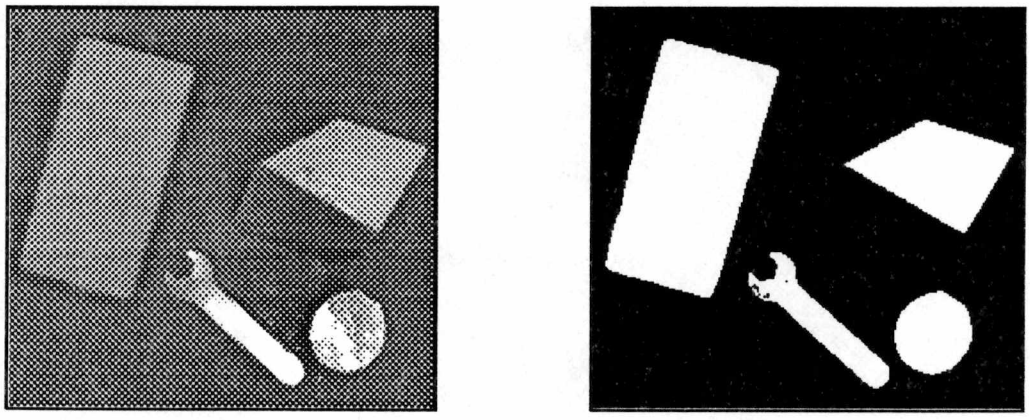


Figure 5.46: *The third object is a compliant object and cannot be explored extensively using tactile sensor. The exploration for the fourth object involves in generating a composite tactile imprint.*



(a) (b)

Figure 5.47: *(a) The original visual image is thresholded as shown in (b) thus enabling extraction of all four object regions from their background.*

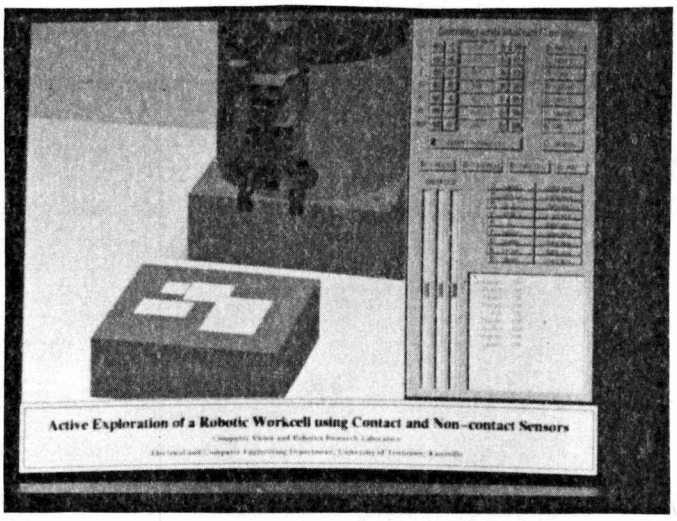
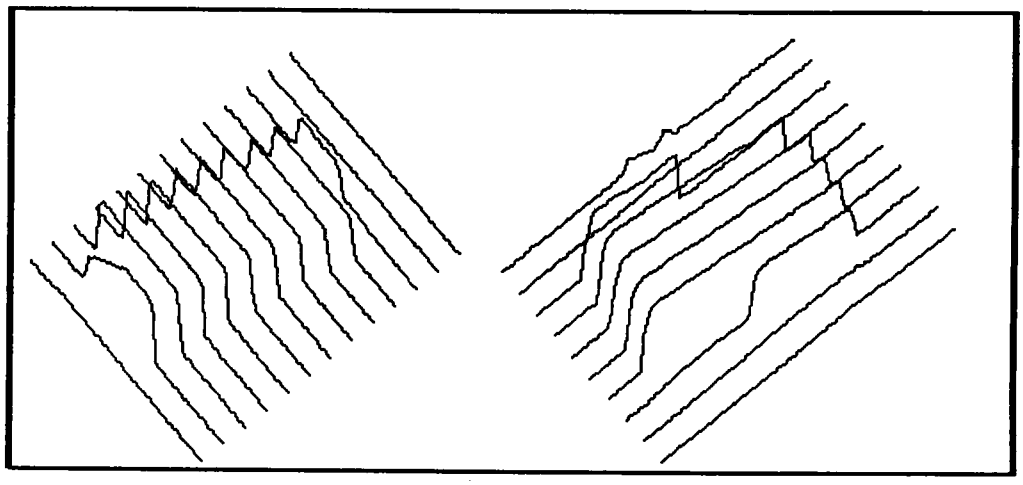
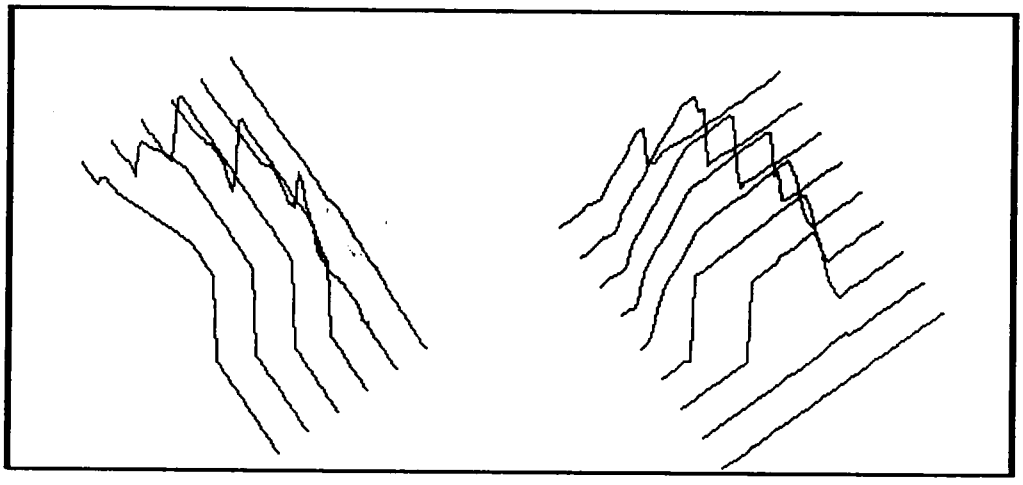


Figure 5.48: *The 2–D bounding box for each of the objects is shown in the simulator.*



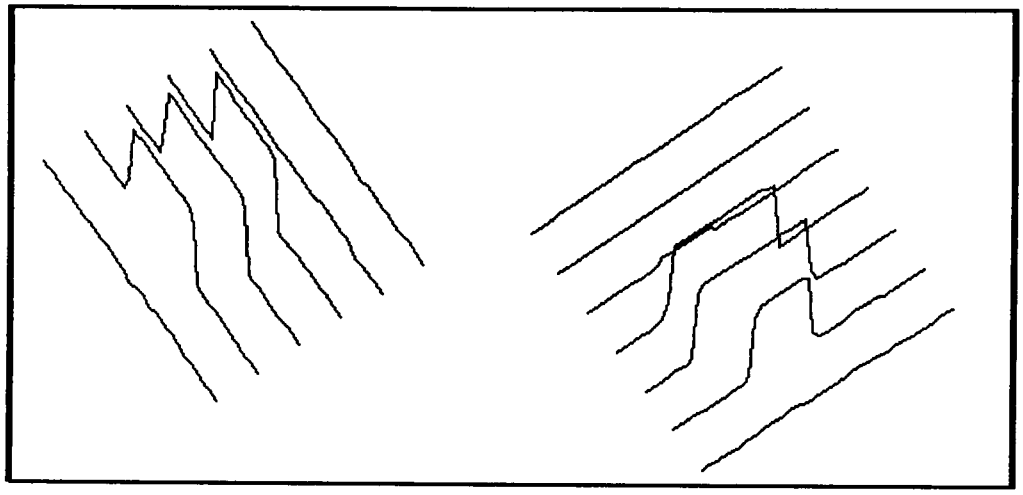
(a) (b)

Figure 5.49: *The (a) horizontal and (b) vertical PLR scans for the rectangular (non-compliant) object.*



(a) (b)

Figure 5.50: *The (a) horizontal and (b) vertical PLR scans for the second (non-compliant) object.*

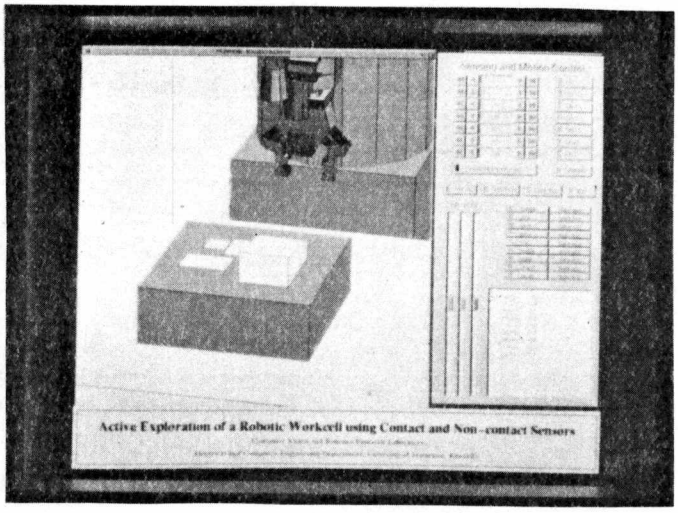


(a) (b)

Figure 5.51: *The (a) horizontal and (b) vertical PLR scans for the compliant object.*

Table 5.4: *Experimental results for the exploration of the rectangular object in the work-cell containing multiple objects.*

Geometric Properties Results	**Exploration from 3–D Model**	**Actual Values**
Number of surfaces	6	6
Number of vertices	8	8
Number of pairs of parallel surfaces	3	3
Number of edges located using haptic exploration	4	4
Relative orientation between edges located by haptic exploration (in degrees)	89.0, 94.7, 84.8.0, 91.6.0	90.0, 90.0, 90.0, 90.0



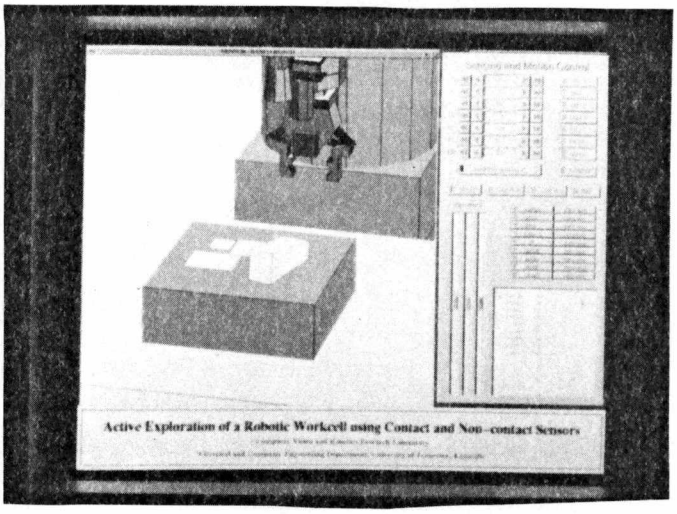


Figure 5.52: *Incremental exploration results for the rectangular object.*

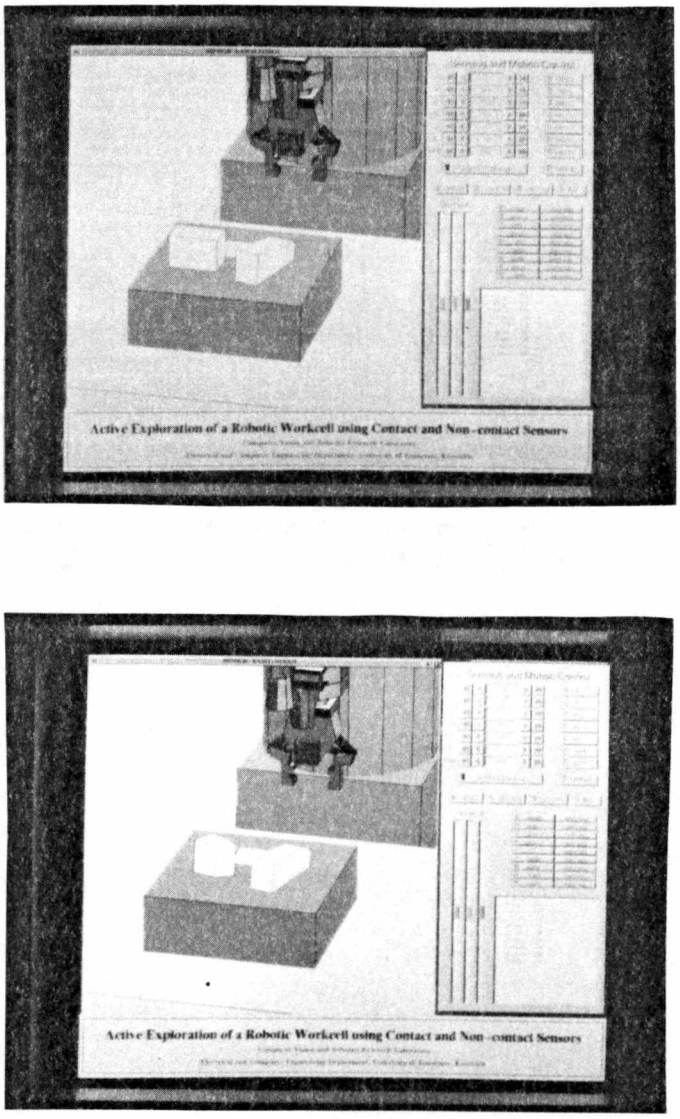


Figure 5.53: *Incremental exploration results for the second object.*

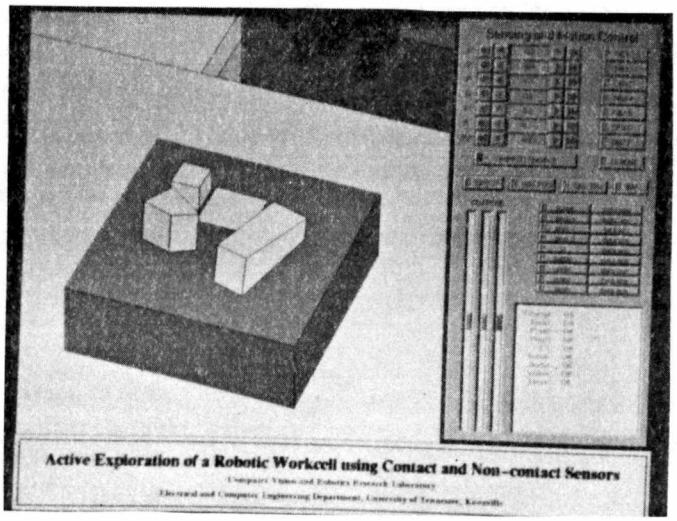


Figure 5.54: *Incremental exploration results for the compliant object.*

Table 5.5: *Experimental results for the exploration of the second non-complaint object in the workcell containing multiple objects.*

Geometric Properties Results	**Exploration from 3-D Model**	**Actual Values**
Number of surfaces	6	6
Number of vertices	8	8
Number of pairs of parallel surfaces	3	3
Number of edges located using haptic exploration	4	4
Relative orientation between edges located by haptic exploration (in degrees)	89.0, 94.7, 84.8.0, 91.6.0	90.0, 90.0, 90.0, 90.0

5.5. The error associated with the haptic exploration of the second object is relatively larger than the rectangular object. The important aspect being the exploration requirements and thus the eventual model developed for the rigid objects, the compliant object and the thin planar object are quite different. For some of the exploration tasks, such as approximation of number of surfaces and surface normal computations, the object category is not affected in any way. However, the haptic exploration tasks are different for different categories of objects.

5.4 Flexibility of the Active Exploration System

One of the noteworthy features of the active exploration system described here is that the system is capable of allowing the sensors and the exploration tasks that will be performed using the selected sensor to be specified at any stage of the exploration. Such a capability can be provided only when the architecture is modular and the data integration scheme is not rigidly tied to a particular set of predefined exploration tasks. The exploration is a dynamic process that may be required to be varied depending on the workspace composition and object type. Further, the closed–loop control mechanism allows us to study the performance of the exploration system when the exploration task sequence is changed. Particularly, for object recognition tasks, the exploration process can be structured such that only the most important features are extracted in contrast to exploring the finest details. However, for resolving conflicts, if finer details are necessary, then the exploration process can be continued till the conflicts are satisfactorily resolved. The same set of objects were used to compare the exploration steps, and the results. The sequence of exploration tasks used for this experiment is shown in Figures 5.55–5.57.

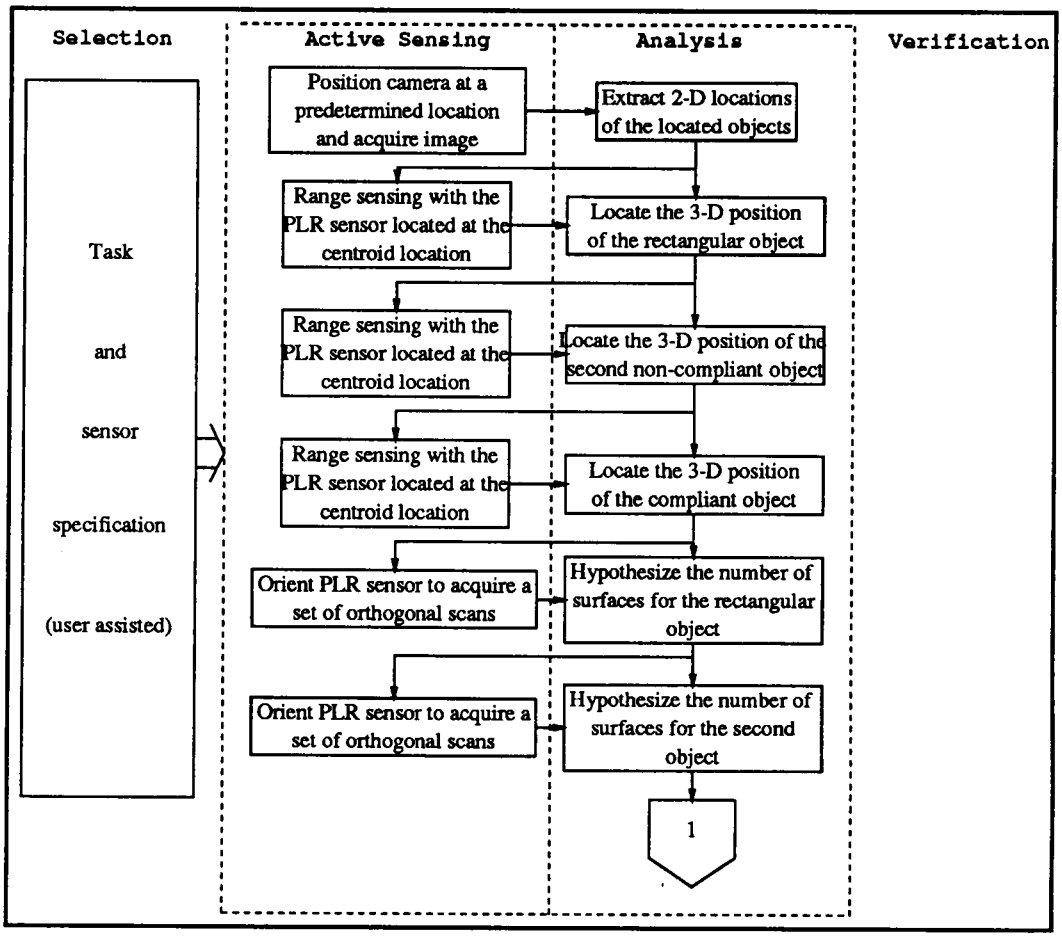


Figure 5.55: *Exploration task ordering was changed considerably to explore the same set of objects used in the previous experiment.*

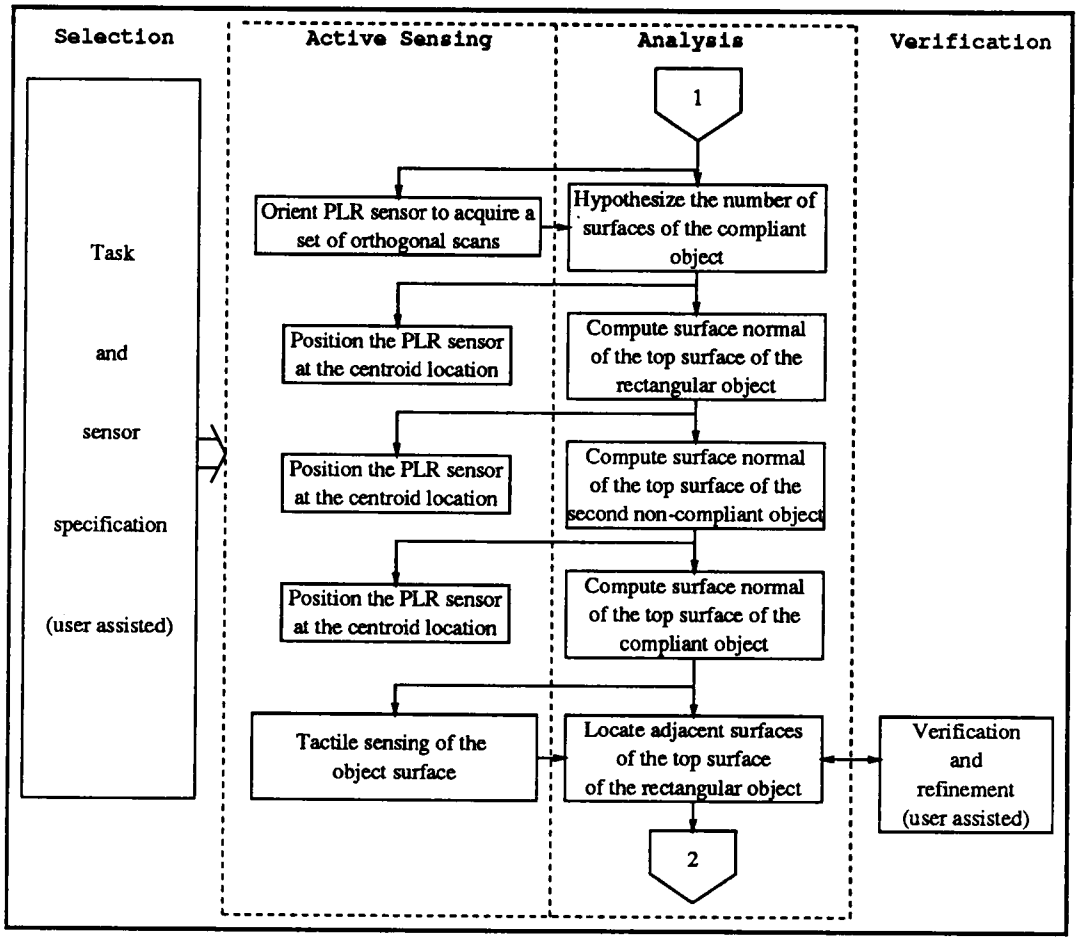


Figure 5.56: *The exploration task sequence shown in the figure has not been changed arbitrarily, because of interdependencies among descriptors. The exploration sequence shown here indicates that it is not necessary to complete the exploration of the first object before proceeding to the second object.*

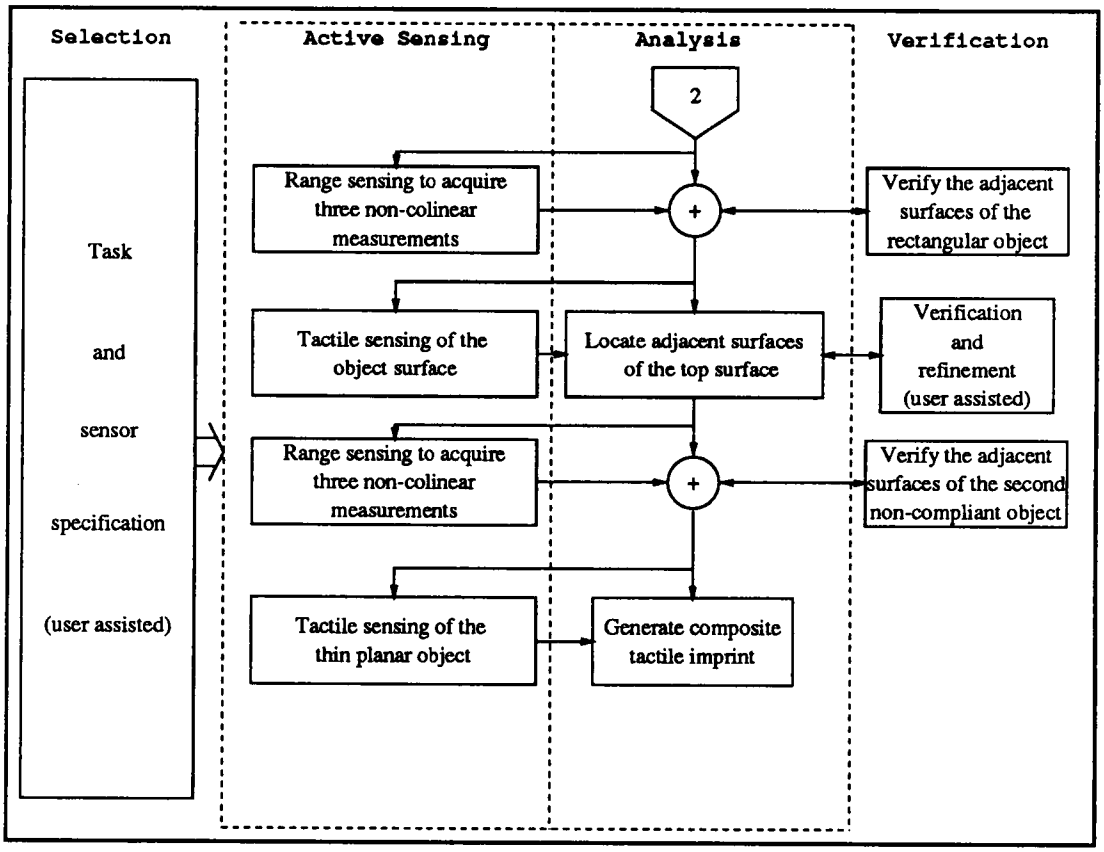


Figure 5.57: *The exploration sequence shown here indicates that it is not necessary to complete the exploration of the first object before proceeding to the exploration of a second object.*

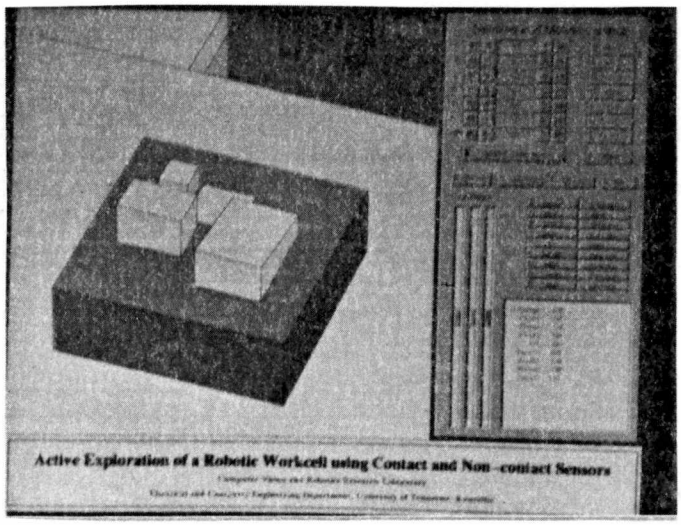


Figure 5.58: *Exploration results for all the objects after the computation of surface normals.*

It should be noted that several tasks can be reorganized depending on the application at hand. Figure 5.58 shows the incremental integration results using the exploration procedure outlined in Figures 5.55–5.57. In this particular case the final outcome (Figure 5.59 has not changed in any sense due to the change in the ordering of the exploration tasks. Note that there always exists a certain implicit ordering among the exploration tasks. For instance, the surface normal computation task is required to precede the tactile exploration of that particular surface. Such mutual dependencies cannot be disturbed arbitrarily.

The exploration results are similar to those of the previous experiment. However, it should be noted that depending on the objects themselves, a change in exploration procedure can affect the quality of the results.

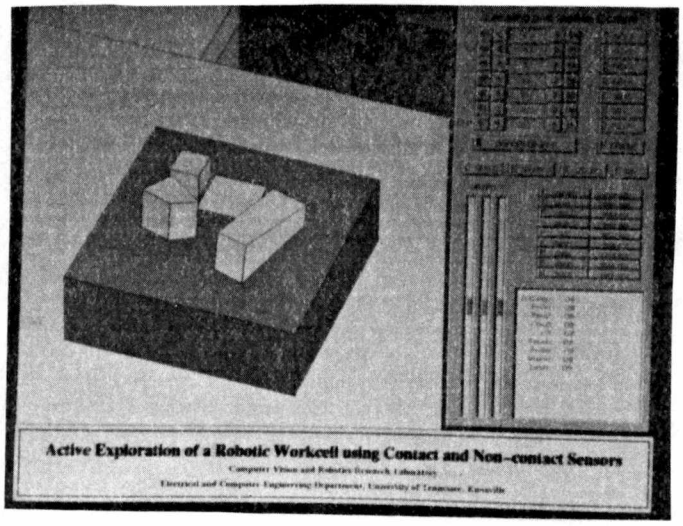


Figure 5.59: *Exploration results for all the objects after haptic exploration.*

5.5 Experimental Validation of the Graceful Degradation Feature of the Active Exploration System

The active exploration system described in this research exhibits a certain amount of redundancy, that is, certain descriptors can be extracted by more than one sensory modality. Such redundancies prove to be beneficial when the exploration system is unable to utilize some of its sensors. The active exploration system essentially consists of *non–contact* (i.e., vision and PLR) and *contact* (i.e., force/torque and tactile) sensors. Based on the objects present in the work cell, at least one of the *non–contact* and *contact* sensors is available to the system. For example, if the PLR sensor is available to explore non–compliant objects, the tactile sensor would be required and vision and force/torque sensors may not be required. Further, if compliant objects need to be explored, then the system would require the force/torque sensor along with one of the *non–contact* sensors and would not require the tactile sensor to be available. In this experiment, vision

and force/torque sensors are assumed to be unusable and the exploration system relies totally on the PLR and tactile sensors. For the same set of objects used in the previous two experiments, the exploration system performs the tasks indicated in Figures 5.60 and 5.61.

The number of objects, their location, and bounding box parameters are extracted by acquiring a coarse 3–D scan of the entire workspace (Figure 5.62). Note that the thresholding process is unable to extract the thin planar object due to the limitations associated with the workspace.

The final results are comparable to the models developed for the previous two experiments (Figures 5.63–5.64). However, the degradation in the overall performance, is in terms of exploration time and accuracy. The time associated with scanning the entire workspace is much greater than the time associated with vision based processing. Also, the thin planar object was located quite easily in the visual image, whereas in the PLR 3–D scan the thresholding technique is unable to extract it successfully. Thus, the exploration system is capable of functioning even in the absence of two important sensory modalities.

5.6 Discussion of the Active Exploration Results

The objectives of this research were described in detail in chapter 1, where an attempt was made to emphasize designing and developing an active exploration that is modular in architecture, employs a closed–loop control mechanism, provides a means for incremental data integration, is able to modify the sequence of exploration tasks (flexibility) to be able to explore a variety of objects, and exhibits graceful degradation.

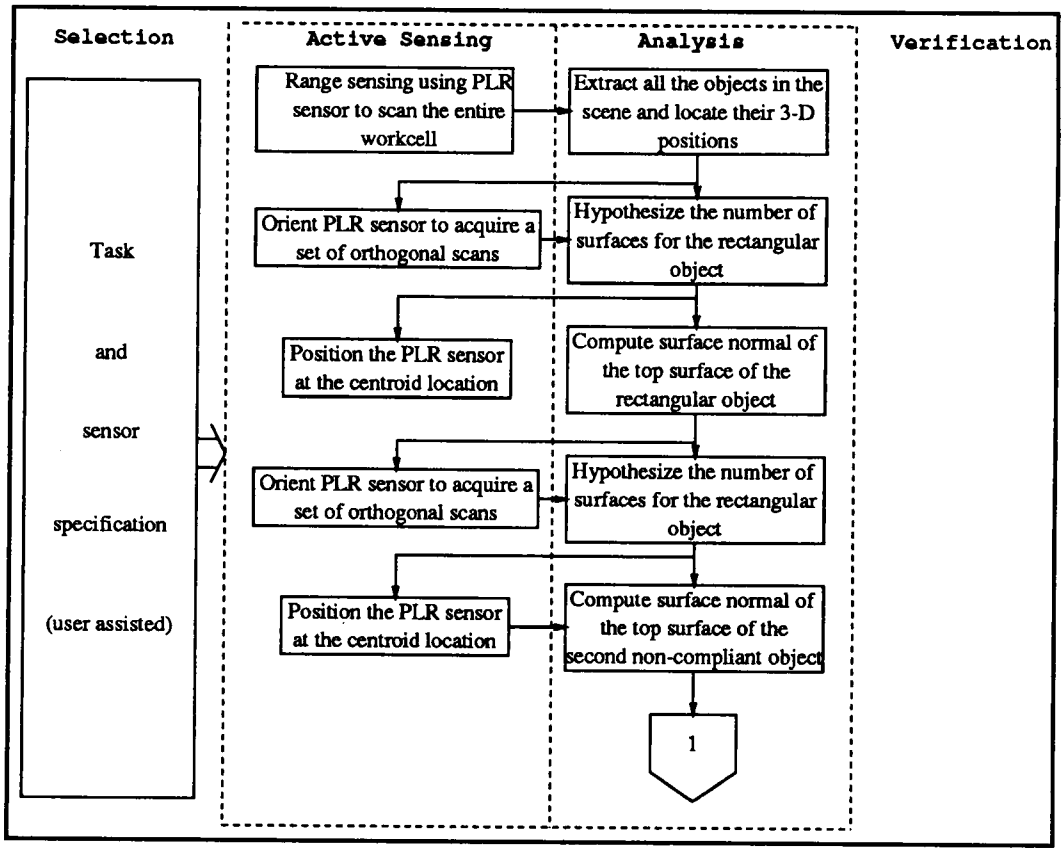


Figure 5.60: *Exploration tasks that were utilized to demonstrate graceful degradation feature of the system.*

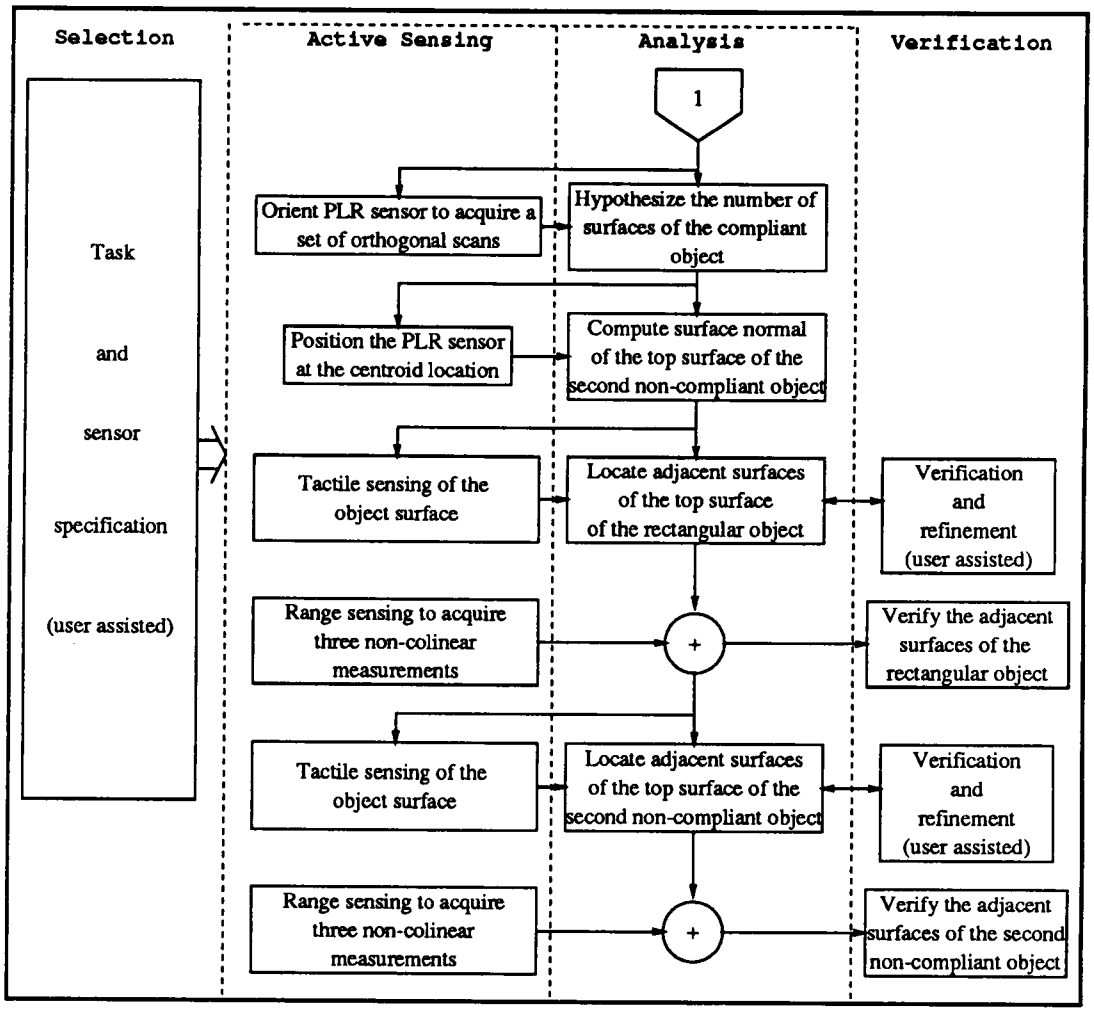
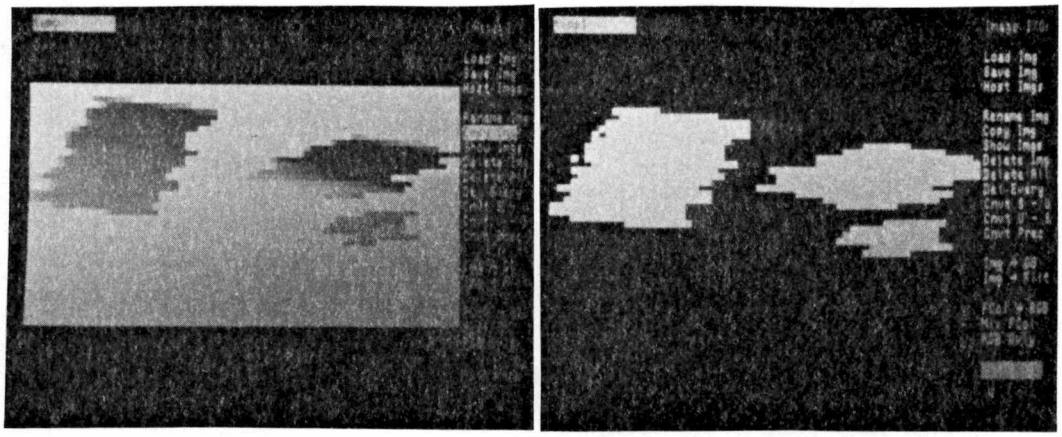


Figure 5.61: *The 3-D location of the centroid for each of the objects located is readily available from the PLR scan of the entire workcell. Thus, the exploration tasks show that the 3-D location of each of the objects is not measured again using PLR sensor.*



(a) (b)

Figure 5.62: *In this experiment, the PLR sensor is used to replace the task performed by vision in the earlier experiments. The figure shows the 3–D scan of the entire workspace consisting of four objects. Note that only three objects have been successfully extracted in the thresholded image (b).*

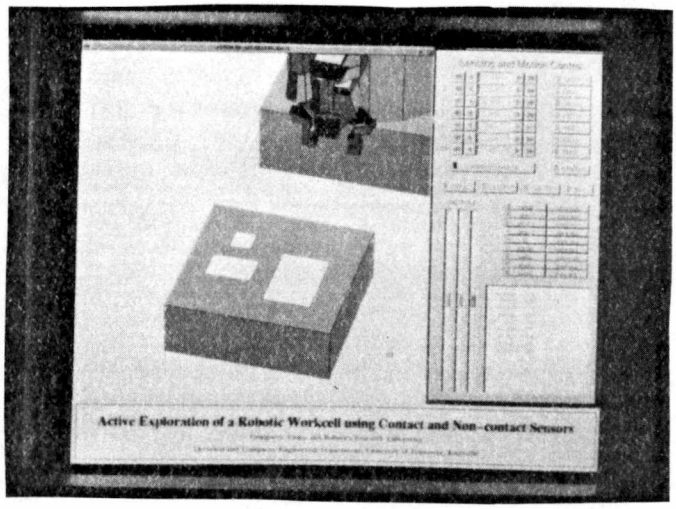


Figure 5.63: *The 2–D bounding boxes of only three objects are being displayed in the simulator following the thresholding operation.*

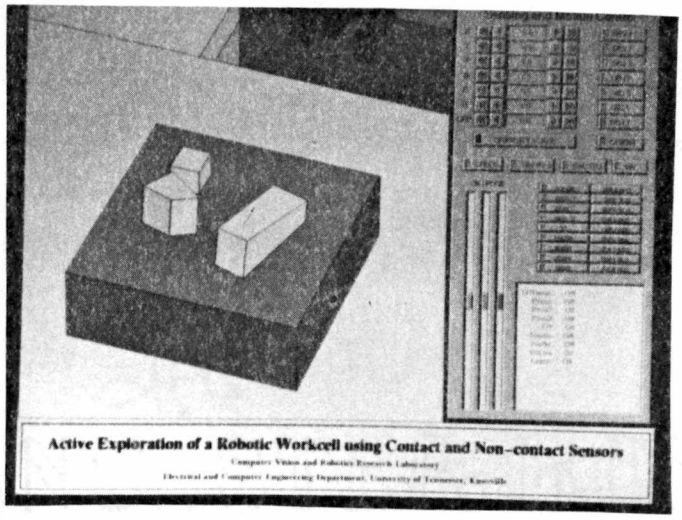


Figure 5.64: *Final exploration results for all the objects localized in the workspace.*

The above experiments clearly bring out the distinctive qualities of the active exploration system designed and implemented in this research. The ability to utilize either *non–contact* or *contact* sensors at any stage during the exploration enables the system to explore a variety of objects. The first six experiments exploit the key features of the active exploration system, namely, the modular architecture, closed–loop control, incremental data integration, and sensor selection. The model generated subsequent to the exploration of objects presented in the first, second, third, and the sixth experiment, when compared with the real object, provides a fairly good estimate. Particularly, as in the case of the first experiment, two surfaces remained undetectable due to their position. Further, the haptic exploration results indicate that the angles between edges that are explored vary over a range of $\pm$ 12°. Such errors can be eliminated by performing repeated checks on the orientation of the edge and applying corrections to reorient the tactile sensor along the edge being followed. Note that this would increase the time of exploration considerably. The model for the objects in the first two experiments and the first object in the sixth experiment were quite close to their real values.

The experiments associated with flexibility and graceful degradation were intended to show the system performance due to changes in either the exploration sequence or the availability of sensors. The flexibility of the system for the experiment described here did not deteriorate or improve the result considerably. The main idea behind demonstrating the flexibility feature is that the work environment of present day robots is typically quite dynamic. The conditions present on one day may not remain the same in the future. This implies that as the work environment undergoes changes or modifications, the robotic system should possess a certain degree of flexibility to accommodate for such changes. Also, for telerobotic applications, different users may want to utilize the

system in different ways, and in such scenarios flexibility becomes an important issue.

The experiment demonstrating graceful degradation brings out two important factors. First, the experiment clearly shows that different "sensory systems" (i.e., data acquisition and data processing) have varying capabilities. For example, both vision and PLR were able to locate and extract the larger objects in the scene; however, unlike vision, PLR data processing techniques did not allow the thin planar object to be extracted from the background. Second, sensory modalities are interdependent, and thus, multi–sensory systems should be designed keeping these interdependencies in mind. For example, the tactile sensor depends on some sensor that is able to provide a starting point for haptic exploration. Also, the tactile sensor has the sensor alignment requirements that necessitates a sensor that can provide the equation or the surface normal of the surface to be explored.

During each experiment certain unfavorable circumstances were encountered that necessitated the experiment to be either repeated or to be terminated and the experiment resumed from the point of termination. Situations where the haptic exploration results in losing contact with the object surface due to a failure in determining edge termination can cause the system to fail. Also, improper computation of surface normal can cause the haptic exploration to fail. Another source of failure is when the object gets dislodged from its position by the end–effector carrying the tactile sensor while going from one location to the other. Particularly, for the second experiment, the exploration had to be performed several times because the object was getting dislodged from its position each time the end–effector applied external force while acquiring tactile imprints. In such situations, the user can terminate the exploration and resume the exploration process by repeating the exploration task that caused the system to fail or restart the

entire exploration from the beginning.

The inaccuracies of the generated model are not due to the way in which exploration is performed or the exploration tasks are selected and their mutual ordering. The accuracy of the model depends on

- *Accuracy of the manipulator:* The accuracy of the manipulator, i.e., the ability of the manipulator to achieve the required configuration.

- *Sensory coordinate frame:* The local sensory coordinate to the world coordinate transformation techniques. Particularly, when the local sensory coordinate frames are defined, the inherent assumption is that the *pose* of these coordinate frames is "accurately" known with respect to the robot coordinate frame.

- *Sensor calibration:* Among the four sensors used in this research, vision has the most stringent constraints on sensor calibration. Sensor calibration is closely related to the issue of sensor coordinate frame position and orientation. However, the sensor calibration also takes into account the sensor performance factors in terms of the sensor accuracy, noise, etc.

- *Sensor resolution:* The vision and PLR have quite good resolution, whereas the tactile and force/torque sensors have fairly poor resolution. Particularly, the tactile sensor resolution is such that given a single tactile imprint, the contour orientation can vary over a range of $\pm$ 10°. This particular factor is a function of sensor technology and with better sensor designs, the error can be reduced further.

The composite tactile imprints for thin planar objects provide a means of representing the 3–D structure of these objects. The least–square cross–correlation technique

allows accurate identification of the object's shape and size. Note that the composite tactile imprint generation scheme being purely data driven, arbitrarily shaped objects can be explored. However, for compliant objects, 3–D model building can be performed only by using *non–contact* sensors. For such objects, additional *non–contact* sensor based data acquisition and processing schemes need to be developed which will provide the object descriptors that were otherwise provided by contact sensors.

CHAPTER 6

SUMMARY AND CONCLUSIONS

Robotic manipulatory systems are being developed with an everincreasing vigor for a variety of applications in unstructured environments. Their successful utilization relies heavily on the use of sensory feedback mechanisms to guide their motions to perform a variety of tasks. The feedback mechanism typically includes one or more sensors such as vision, range, proximity, tactile and force/torque. A variety of robotic applications, such as object recognition, manipulation, and workspace characterization, require repeated use of perception and manipulatory tasks. This research focuses on developing a robotic exploration system with particular emphasis on using both *contact* and *non–contact* sensors. In the past, researchers were motivated towards developing robotic systems performing active perception using *non–contact* sensors as a means of facilitating the data acquisition process of the *contact* sensors. The exploration system discussed here goes far beyond fulfilling the needs of an active perception system. By designing and implementing an active exploration system, this research shows that an active exploration system can be reconfigured to suit the requirements of the above mentioned applications.

In addition, the strength of this research is derived on the basis of conducting a series of experimental studies to derive and validate methodologies and algorithms constituting the computational framework and the architecture for multisensor based active exploration. This research presents an architecture for active exploration that is modu-

lar and utilizes an intelligent closed–loop control scheme. The exploration system design was developed after performing a series of experiments designed to study the usefulness of *contact* sensing. The specific issues that are quite unique and significant in the implementation of the active exploration system can be categorized into those associated with the architecture and control scheme and those associated with sensory data acquisition and analysis.

- **Architecture and Control Scheme**

  1. *Multisensory Integrated System:* This research describes an integrated system architecture that utilizes *non–contact* and *contact* sensors. The architecture is not only modular but also facilitates the implementation of an intelligent closed–loop control mechanism. Through this research by emphasizing a closed–loop architecture it has been possible to show that haptic exploration can be implemented in a highly efficient manner to go much beyond addressing just the object recognition problem.

  2. *Incremental Data Integration Scheme:* An important consideration that ensures robotic systems to be able to function in unstructured and hazardous environment is the design of their integration scheme. The notion of incremental data integration and the data driven exploration approach are the key factors of this exploration system and are not seen to coexist in most other integrated multi–sensory systems.

  3. *Flexible Exploration System:* The system allows the sensors and the order in which they need to be used to be tailored to the application at hand. The exploration system described in this research looks at the design of active

exploration systems from a slightly different approach, where the emphasis is on the ability to choose between *non–contact* and *contact* sensors at any stage during the active exploration, thereby providing maximum flexibility to the exploration system.

4. *Graceful Degradation:* The design of the architecture and the intelligent control scheme is such that redundancy among a certain set of sensors is used to keep the system operational at a lower performance rate when one of the sensors is unusable due to unfavorable working conditions. This facility adds an additional component to the exploration system described here, and this has always remained alien to most of the day multi–sensory robotic systems.

- **Sensory Data Acquisition and Analysis**

1. *Tactile and Force/Torque Data Acquisition and Analysis:* In order to be able to explore a variety of convex and concave polyhedral objects, the tactile data acquisition and data processing modules have to be designed such that they would retain their generality to a great extent. The techniques utilized for extracting surface and contour information can handle surfaces that are arbitrarily oriented by first extracting the surface normal using the force/torque sensor and then by using the tactile sensor to explore the surface.

2. *Versatile Point Laser Range System:* The versatility of the PLR sensory system is in its ability to extract information from arbitrary *pose*, variable size scans, and variable spatial resolutions. The versatility and the wide variety of spectral and geometric information that can be extracted from PLR data makes the entire active exploration system quite unique.

The motivation of this research is towards designing an exploration system that is not tailored to a specific robotic application. Having the "generality of design" in the focus, this research is intended to address issues associated with the integrated system architecture, control mechanism, data integration, nature of sensory modalities, data acquisition and processing modules, sensor ordering, and system performance. The design and development of the computational framework and the architecture includes a *primary task*, comprising of data integration, input systems, and data acquisition modules; and two *secondary tasks*, namely, sensor selection module and Half–space modeler. The emphasis on the primary task is relatively greater than the secondary tasks because the implementation of the primary task essentially determines the nature and extent of experiments that can be performed. The exploration system utilizes an interactive scheme for sensor selection and exploration task selection. The Half–space modeler is capable of modeling convex and concave objects. In addition, models of complex objects can be built by performing a repeated union operation of simpler convex or concave object models.

The experimental verification process includes eight experiments designed to emphasize the above mentioned issues and the major contributions of this research. The experiments include convex, concave, and thin planar rigid objects and also compliant or non–rigid objects. The experiments were successful in bringing out the advantages associated with having a *modular* architecture, *closed–loop* control mechanism, *incremental* data integration, *modality specific* data acquisition and processing modules, *flexibility* and *graceful degradation.* The quantitative analysis of the experimental results show that the 3–D models are similar to the original object. However, in case of compliant objects, the model building is not pursued extensively due to certain limitations

associated with haptic exploration.

The active exploration system described in this research allows the exploration tasks to be reorganized so that it can be utilized for applications such as object recognition, object manipulation, grasping, assembly, etc. The sensor selection and exploration task selection can be automated by developing mathematical models representing sensory and data processing capabilities. However, for automating scheduling of exploration tasks, there exists a need for developing a means by which it would be possible to monitor the progress of the 3–D model building task. Such a facility would allow determination of whether further exploration is needed, and if so, what should be the next exploration step.

CHAPTER 7

FUTURE RESEARCH SCOPE AND DIRECTION

This dissertation describes our efforts in developing a computational framework for active exploration using *contact* and *non-contact* sensors. The experimental analysis shows that the design and implementation of the active exploration system has the ability to explore complex polyhedral objects, and at the same time exhibits certain distinguishing qualities such as flexibility and graceful degradation. However, the experiments have also helped bring out certain limitations associated with the task of exploration and the present implementation of the active exploration system. These limitations can be addressed to great lengths and are being posed here as interesting research issues, some of which are being actively studied by other researchers.

- *Sensor Selection:* The task of sensor selection for this research was accomplished interactively. For telerobotic systems, an interactive scheme can prove to be quite advantageous. However, development of an *automatic* sensor selection scheme remains to be a challenging problem. An empirical technique has been suggested in section 3.2.2 of this dissertation, which need not be the best or an optimal solution. In order to reliably determine the appropriateness of a sensor for a given task, it would be required to design specific experiments that can evaluate sensor performance. Particularly, when multiple sensors are capable of extracting the same information, it becomes quite critical to derive a set of criteria that will allow comparison of various sensory modalities along the same scale.

- *Exploration Task Selection:* Exploration task scheduling is another challenging issue, since exploration assumes that very little knowledge about the object being explored is available prior to the commencement of exploration. Having a fixed set of exploration tasks for all types of objects may prove to be wasteful or time consuming. Thus, in order to make the exploration process efficient, there exits a need for developing a means by which the exploration or the integrated result can be monitored continuously to determine if further exploration would be required or not. Thus, automating the task of exploration task selection would require the exploration system to evaluate the change or the amount of modification subsequent to each exploration step.

- *Error Diagnosis and Recovery:* Particularly, *contact* sensor based exploration requires a means of detecting any errors associated with the sensor positioning or object descriptor extraction. Further, research is needed in this area, where an alternative sensory modality would be used to verify the the information acquired by a particular sensory modality.

- *Active Manipulation of Objects:* The experimental results indicate that certain surfaces remain unexplored or sometimes even undetected by the exploration system either due to the inaccessibility of the surfaces themselves or inaccessibility of at least one of their adjacent surfaces. Such scenarios could be handled if it is possible to manipulate the object and reposition/reorient the object to ascertain accessibility of all of its surfaces. This task of manipulation requires computation of a grasping pattern and computation of a new *pose* for the object. Certain cases might also warrant not only grasping but also regrasping before the object can be

made to achieve its desirable *pose*.

To pursue further research along these research directions, essentially, it would require a clear understanding of the entire task of exploration itself in a broader sense. That is, the exploration procedures should be quite general so that the exploration system will be able to handle a variety of complex objects. Exploration is certainly a time consuming process; however, the knowledge, experience and insight gained through building an active exploration system is extremely useful while trying to build an application specific multi–sensory robotic system.

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] R. Agrawal and R. Jain, "An overview of tactile sensing, technical report," tech. rep., The University of Michigan, Robot Systems Division, Center for Research on Integrated Manufacturing, Ann Arbor, MI, 1986.

[2] P. K. Allen, "Surface Description From Vision and Touch," in *Proceedings of the International Conference on Robotics and Automation*, (Atlanta, GA), pp. 394–397, 1984.

[3] P. K. Allen, *Object Recognition using Vision and Touch.* PhD thesis, University of Pennsylvania, 1985.

[4] P. K. Allen, "Sensing and Describing 3-D Structure," *Proceedings of the International Conference on Robotics and Automation*, pp. 126–131, April 1986.

[5] P. K. Allen, "Integrating Vision and Touch for Object Recognition Tasks," *The International Journal of Robotics Research*, vol. 7, no. 6, pp. 15–33, December 1988.

[6] P. K. Allen, "Mapping Haptic Exploratory Proceedures to Multiple Shape Representations," in *Proceedings of IEEE International Conference on Robotics and Automation*, (Cincinnati, OH), pp. 1679–1684, 1990.

[7] M. A. Arbib, "Perceptual Structures and Distributed Motor Control," in *Handbook of Physiology - The Nervous System II* (M. Brady and R. Paul, eds.), pp. 1449–1480, Cambridge, MA: MIT Press, 1984.

[8] M. Arbib and A. R. Hanson, *Schemas that Integrate Vision and Touch for Hand Control, in Vision, Brain, and Cooperative Computation.* Cambridge, MA: The MIT Press, 1987.

[9] C. Archibald and S. Amid, "Calibration of a Wrist–mounted Range Profile Scanner," in *Proceedings of Vision Interface*, (London, Ontario), pp. 24–28, June 1989.

[10] C. M. Bastuscheck, "Area Touch Sensor for Dextrous Manipulation," in *Proceedings of IEEE Internaltional Conference on Robotics and Automation*, pp. 151–156, May 1989.

[11] F. P. Beer and E. R. Johnston, *Vector Mechanics for Engineers: Statics.* McGraw–Hill Book Company, 1977.

[12] S. Begej, "Planar and Finger-Shaped Optical Tactile Sensors for Robotic Appplications," *IEEE Journal of Robotics and Automation*, vol. 4, no. 5, pp. 472–484, October 1988.

[13] P. J. Besl and R. C. Jain, "Segmentation Through Variable–Order Surface Fitting," *IEEE Trasanctions on Pattern Analysis and Machine Intelligence*, vol. PAMI-10, no. 2, pp. 167–192, March 1988.

[14] P. J. Besl, "Active, Optical Range Imaging Sensors," *Machine Vision and Applications*, vol. 1, no. 2, pp. 127–152, 1988.

[15] C. R. Bidlack and M. M. Trivedi, "Integrated Vision System for Object Identification and Localization using 3-D Geometrical Models," in *Proceedings of the Applications of Artificial Intelligence VIII Conference*, (Orlando, FL), pp. 270–280, April 1991.

[16] R. A. Boie, "Capacitance Impedence Readout Tactile Image Sensor," in *Proceedings of IEEE International Conference on Robotics*, (Atlanta, GA), pp. 370–378, March 1984.

[17] A. Cameron, "Optimal Tactile Sensor Placement," in *Proceedings of IEEE International Conference on Robotics and Automation*, pp. 308–313, May 1989.

[18] C. Chen, M. M. Trivedi, C. R. Bidlack, and T. Lassiter, "An Environment for Simulation and Animation of Sensor-Based Robots," in *Proceedings of the Applications of Artificial Intelligence IX Conference*, (Orlando, FL), pp. 354–366, April 1991.

[19] C. Chen, *Planning, Coordination and Execution of Perceptual and Motor Actions in Sensor-Based Intelligent Robots.* PhD thesis, Department of Electrical and Computer Engineering, University of Tennessee, Knoxville, 1991.

[20] C. Chen, M. M. Trivedi, and C. R. Bidlack, "Simulation and Graphical Interface for Programming and Operation of Sensor-based Robots," in *Proceedings of IEEE International Conference on Robotics and Automation*, (Nice, France), May 1992.

[21] K. J. Chun and K. D. Wise, "A Capacitive Silicon Tactile Imaging Array," in *Proceedings of IEEE International Conference on Solid-State Sensors and Actuators*, (Philadelphia, PA), pp. 22–25, June 1985.

[22] J. J. Clark, "A Magnetic Field Based Compliance Matching Sensor for High Resolution, High Compliance Tactile Sensing," in *Proceedings of IEEE International Conference on Robotics and Automation*, vol. 2, pp. 772–777, April 1988.

[23] A. S. Collins and W. A. Hoover, "A Prototype for an Image-Based Tactile Sensor," in *Proceedings of IEEE International Conference on Robotics and Automation*, (Chapel Hill, NC), pp. 1760–1765, 1987.

[24] M. R. Cutkosky, *Robotic Grasping and Fine Manipulation.* Kluwer Academic Publishers, 1989.

[25] P. Dario, A. Bacchi, F. Vivaldi, and P. C. Pinotti, "Tendon Actuated Exploratory Finger With Polymeric Skin-Like Tactile Sensor," in *Proceedings of IEEE International Conference on Robotics and Autoamtion*, (Philadelphia, PA), pp. 701–706, April 1985.

[26] P. Dario and D. D. Rossi, "Tactile Sensors and the Gripping Challenge," *IEEE Spectrum*, vol. 22, no. 8, pp. 46–52, August 1985.

[27] R. E. Ellis, "Acquiring Tactile Data for the Recognition of Planar Objects," in *Proceedings of The International Conference on Robotics and Automation*, (Chapel hill, NC), pp. 1799–1805, 1987.

[28] E. V. Evarts, "Brain Mechanisms of Movement ," *Scientific American*, vol. 241, no. 3, pp. 164–169, September 1979.

[29] R. S. Fearing, "Simplified Grasping and Manipulation with Dextrous Robot Hands," *IEEE Journal of Robotics and Automation*, vol. RA-2, no. 4, pp. 188–195, December 1986.

[30] R. S. Fearing, "Tactile Sensing for Shape Interpretation," in *Dextrous Robot Hands* (S. T. Venkataraman and T. Iberall, eds.), pp. 209–238, New York, NY: Springer-Verlag, 1990.

[31] J. A. Fodor, *Modularity of Mind.* MIT Press, 1983.

[32] K. S. Fu, R. C. Gonzalez, and C. S. G. Lee, *Robotics Control, Sensing, Vision, and Intelligence.* New York, NY: Mc-Graw Hill Book Company, 1987.

[33] H. P. Gadagkar and M. M. Trivedi, "Tactile Sensory Analysis for Robotic Applications," in *Proceedings of the Applications of Artificial Intelligence VIII Conference*, (Orlando, FL), pp. 788–800, 1990.

[34] H. P. Gadagkar, M. M. Trivedi, and T. N. Lassiter, "A Versatile Approach for Measuring Proximity and Surface Profile Using a Wrist Mounted Point Laser Sensor," Tech. Rep. TR-ECE-91-32, The University of Tennessee, Electrical and Computer Engineering Department, Knoxville, TN, March 1991.

[35] H. Gadagkar and M. M. Trivedi, "Tactile Sensor Technology for Robotic Applications," in *Concurrent Engineering* (C. T. Leondes, ed.), Orlando, FL: Academic Press, 1992.

[36] H. P. Gadagkar and M. M. Trivedi, "Computational Approaches for Tactile Information Processing and Analysis," in *Advances in Computers* (M. Yovits, ed.), Orlando, FL: Academic Press, 1992.

[37] W. E. L. Grimson and T. Lozano-Perez, "Model-Based Recognition and Localization from Sparse Range or Tactile Data," *The International Journal of Robotics Research*, vol. 3, no.3, pp. 3–35, Fall 1984.

[38] R. A. Grupen, T. C. Henderson, and I. D. McCammon, "A Survey of General Purpose Manipulation," *The International Journal of Robotics Research*, vol. 8, no. 1, pp. 38–62, February 1989.

[39] S. Hackwood, G. Beni, L. A. Hornak, *et al.*, "A Torque-Sensitive Tactile Array for Robotics," *The International Journal of Robotics Research*, vol. 2, no. 2, pp. 46–50, Summer 1983.

[40] L. D. Harmon, "Automated Tactile Sensing," *The International Journal of Robotic Research*, vol. 1, no. 2, pp. 3–32, Summer 1982.

[41] T. Henderson and E. Shilcrat, "Logical Sensor Systems," *Journal of Robotic Systems*, vol. 1, no. 2, pp. 169–193, 1984.

[42] T. Henderson, C. Hansen, and B. Bhanu, "The Specification of Distributed Sensing and Control," *Journal of Robotic Systems*, vol. 2, no. 4, pp. 387–396, 1985.

[43] W. D. Hillis, "A High-Resolution Imaging Touch Sensor," *The International Journal of Robotics Research*, vol. 1. no. 2, pp. 33–44, Summer 1982.

[44] J. M. Hollerbach, "Robot Hands and Tactile Sensing," in *AI in the 1980's and Beyond* (W. E. L. Grimson and R. S. Patil, eds.), pp. 317–343, Cambridge, MA: MIT Press, 1987.

[45] D. H. Hubel, "The Brain ," *Scientific American*, vol. 241, no. 3, pp. 45–53, September 1979.

[46] KEYENCE Corporation of America, Fair Lawn, NJ, *KEYENCE LB-70 Laser Displacement Sensor, Instruction Manual*, 1990.

[47] R. L. Klatzky and S. Lederman, "Intelligent Exploration by the Human Hand," in *Dextrous Robot Hands* (S. T. Venkataraman and T. Iberall, eds.), pp. 66–81, New York, NY: Springer-Verlag, 1990.

[48] R. L. Klatzky, R. Bajcsy, and S. J. Lederman, "Object Exploration in One and Two Fingered Robots," in *Proceedings of the International Conference on Robotics and Automation*, pp. 1806–1809, 1987.

[49] Industrial Automation System, Cary, NC, *LORD Corporation LTS-210 Tactile Sensor System*, 1989.

[50] S. J. Lederman and R. L. Klatzky, "Hand Movements: A Window into Haptic Object Recognition," *Cognitive Psychology*, vol. 19, pp. 342–368, 1987.

[51] P. P. Lin and P. Datseris, "Development of A Position and Force Sensor for Robotic Applications," in *Proceedings of IEEE Internaltional Conference on Robotics and Automation*, pp. 1798–1805, April 1986.

[52] M. Mantyla, *An Introduction to Solid Modeling.* Rockville, Maryland: Computer Science Press, 1988.

[53] L. E. Marks, "Multimodal Perception," in *Handbook of Perception, Perceptual Coding* (E. C. Carterette and M. P. Friedman, eds.), pp. 321–339, New York, NY: Academic Press, 1978.

[54] M. Ogorek, "Tactile Sensors," *Manufacturing Engineering*, pp. 69–77, February 1985.

[55] K. E. Pennywitt, "Robotic Tactile Sensing," *Byte*, pp. 177–200, January 1986.

[56] K. Petersen, C. Kowalski, J. Brown, H. Allen, and J. Knutti, "A Force Sensing Chip Designed for Robotic and Manufacturing Automation Applications," in *Proceedings of IEEE International Conference on Robotics and Automation*, (Philadelphia, PA), pp. 30–32, April 1985.

[57] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, *Numerical Recipes in C.* Cambridge University Press, 1988.

[58] M. H. Raibert and J. E. Tanner, "Design and Implementation of a VLSI Tactile Sensing Computer," *The International Journal of Robotic Research*, vol. 3, no. 18, pp. 3–18, Fall 1982.

[59] M. Rioux, F. Blais, and J. A. Beraldin, "Laser Range Finder Development for 3–D Vision," in *Proceedings of Vision Interface*, (London, Ontario), pp. 1–9, June 1989.

[60] K. Roberts, "Robot Active Touch Exploration: Constraints and Strategies," in *Proceedings of IEEE International Conference on Robotics and Automation*, (Cincinnati, OH), pp. 980–985, 1990.

[61] R. A. Russell, "A Thermal Sensor Array to Provide Tactile Feedback for Robots," *The International Journal of Robotics Research*, vol. 4, no. 3, pp. 35–39, Fall 1985.

[62] R. A. Russell, "Compliant-Skin Tactile Sensor," in *Proceedings of IEEE International Conference on Robotics and Automation*, (Chaple Hill, NC), pp. 1645–1648, 1987.

[63] J. S. Schoenwald, A. W. Thiele, and D. E. Gjellum, "A Novel Fiber Optic Tactile Array Sensor," in *Proceedings of IEEE International Conference on Robotics and Automation*, (Chapel Hill, NC), pp. 1792–1797, 1987.

[64] J. L. Schneiter and T. B. Sheridan, "An Optical Tactile Sensor for Manipulators," *Robotics and Computer-Integrated Manufacturing*, vol. 1, no. 1, pp. 65–71, 1984.

[65] J. L. Schneiter, "An Objective Tactile Sensing Strategy for Object Rocognition and Localization," in *Proceedings of IEEE International Conference on Robotics and Automation*, pp. 1262–1267, April 1986.

[66] D. M. Siegal, S. M. Drucker, and I. Garabieta, "Performance Analysis of a Tactile Sensor," *Proceedings of The International Conference on Robotics and Automation, Chapel hill, NC*, pp. 1493–1499, 1987.

[67] S. A. Stansfield, "Visually-Aided Tactile Exploration," in *Proceedings of the International Conference on Robotics and Automation*, (Chapel Hill, NC), pp. 1487–1492, 1987.

[68] S. A. Stansfield, "A Robotic Perceptual System Utilizing Passive Vision and Active Touch," *The International Journal of Robotics Research*, vol. 7, no. 6, pp. 138–160, December 1988.

[69] B. Tise, "A Compact High Resolution Piezoresistive Digital Tactile Sensor ," in *Proceedings of IEEE International Conference on Robotics and Automation*, vol. 2, pp. 760–764, April 1988.

[70] M. M. Trivedi, M. A. Abidi, R. O. Eason, and R. C. Gonzalez, "Developing Robotic Systems with Multiple Sensors," *IEEE Transactions on Sytems, Man, and Cybernetics*, vol. 20, no. 6, pp. 1285–1300, November/December 1990.

[71] M. M. Trivedi and A. Rosenfeld, "On Making Computers See," *IEEE Transactions on Sytems, Man and Cybernetics*, vol. 19, no. 6, pp. 1333–1335, November/December 1989.

[72] C. S. Vaidyanathan and H. C. Wood, "A New Tactile Sensing System," in *Proceedings of IEEE Internaltional Conference on Robotics and Automation*, pp. 1311–1312, April 1988.

[73] J. M. Vranish, "Magnetoinductive Skin for Robots," in *Proceedings of IEEE International Conference on Robotics and Automation*, pp. 1292–1318, April 1986.

[74] P. D. Wall, "The Sensory and Motor Role of Impulses Travelling in the Dorsal Columns Towards Cerebral Cortex ," *Brain*, vol. 93, pp. 505–526, 1970.

[75] R. B. Welch and D. H. Warren, "Intersensory Interactions," in *Handbook of Perception and Human Performance*, pp. 25–1:36, New York, NY: Wiley, 1991.

[76] R. M. White and A. A. King, "Tactile Array for Robotics Employing A Rubbery Skin and A Solid -State Optical Sensor," in *Proceedings of IEEE Internaltional Conference on Robotics and Automation*, (Philadelphia, PA), pp. 18–21, April 1985.

[77] J. G. Winger and K. M. Lee, "Experimental Investigation of a Tactile Sensor Based on Bending Losses in Fibre Optics," *Proceedings of IEEE International Conference on Robotics and Automation*, vol. 2, pp. 754–759, April 1988.

[78] K. Wong and J. V. der Spiegel, "A Shielded Piezoresistive Tactile Sensor Array," in *Proceedings of IEEE Internaltional Conference on Robotics and Automation*, (Philadelphia, PA), pp. 26–29, April 1985.

VITA

Hrishikesh P. Gadagkar was born on March 27, 1964, in Bombay India. He then joined the Sardar Patel College of Engineering of Bombay University in 1981, where he received his Bachelor of Engineering in Electrical Engineering in 1985. In August of 1985 he joined the Electrical Engineering Department at Louisiana State University to pursue his graduate studies. He then received his Master of Science degree in May 1987. Since January 1987 he has been a research assistant at the Electrical and Computer Engineering Department of the University of Tennessee, Knoxville. He is currently a candidate for the Doctor of Philosophy degree and will be graduating in May 1992.

He is a member of the electrical engineering honor society of Eta Kappa Nu. He is also a student member of IEEE, the IEEE Computer Society, the Society of Manufacturing Engineering/Robotics Institute (SME/RI), and the National Service Robots Association (NSRA).