INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps. Each original is also photographed in one exposure and is included in reduced form at the back of the book.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.
Visually guided tactile and force-torque sensing for object recognition and localization

Rafia, Nader Iskander, Ph.D.
Case Western Reserve University, 1991

Copyright ©1991 by Rafia, Nader Iskander. All rights reserved.
VISUALLY GUIDED TACTILE AND FORCE-TORQUE SENSING FOR OBJECT RECOGNITION AND LOCALIZATION

by

NADER ISKANDER RAFLA

Submitted in the fulfilment of the requirements for the degree of Doctor of Philosophy

Thesis Advisor: Dr. Francis L. Merat

Department of Electrical Engineering and Applied Physics
Case Western Reserve University
August 1991
CASE WESTERN RESERVE UNIVERSITY

GRADUATE STUDIES

We hereby approve the thesis of

______________________
Nader Iskander Rafla

candidate for the Ph. D.

degree.*

Signed:

______________________
Francis Marat

(Chairman)

______________________
Ray D. Adelsperger

______________________
Dale M. Phillips

______________________
Behnam Matabi

Date ________________

*We also certify that written approval has
been obtained for any proprietary material
contained therein.
I grant to Case Western Reserve University the right to use this work, irrespective of any copyright, for the University's own purposes without cost to the University or to its students, agents and employees. I further agree that the University may reproduce and provide single copies of the work, in any format other than in or from microforms, to the public for the cost of reproduction.

Nader F. Raffie
VISUALLY GUIDED TACTILE AND FORCE-TORQUE SENSING
FOR OBJECT RECOGNITION AND LOCALIZATION

Abstract

by

NADER ISKANDER RAFLA

A model-based object recognition and localization system has been developed to recognize and locate three dimensional solid objects using vision range images and touch data. This system performs three main tasks: extraction of surface characteristics, integration of vision and touch data, and object recognition and localization.

Surface characteristics of the sensed object are extracted from vision range images and touch data independently. From vision range images, surface normals are calculated. Points which have similar properties of their normals are grouped into regions called surface patches. Touch data is acquired using a probe gripped by two imaging tactile and force-torque sensors (Lord Corporation gripper SE-680 and
Lord Corporation sensors LTS-200) mounted in a five degree of freedom robot arm (Intelledex 605T). The operation of this tactile system is similar to that of a commercial robot, q.v. a coordinate measuring machine. From touch data, normals are determined using feedback from the tactile and force-torque sensors to orient the probe normal to the surface. Surfaces are located using the inverse kinematics of the robot probe.

The extracted vision and touch features are combined into vision-touch surface patches on the basis of surface normals and position. These vision-touch patches are processed for classification and surface equation determination. The surface equations are calculated for each surface patch using a least-square minimization method that used only a few touch and vision surface points within the same patch.

A series of experiments was done to test the different components of the system individually and as a system for object recognition and localization. The touch data was acquired from a variety of physical objects. Vision range images corresponding to these physical objects were computer generated. Normally distributed noise was added to the vision range images to simulate errors due to timing jitter, etc. The vision and touch data analysis successfully extracted normals for planar, cylindrical, and spherical surfaces. The resulting surface features were integrated into vision-touch surface patches. Some vision points on cylindrical and spherical surfaces were not integrated into the
vision-touch surface patches due to the noise in the vision images but all vision points were correctly identified for planar surfaces.

The object recognition algorithms correctly recognized and located test objects. The objects used in these experiments were chosen to include many different types of surfaces in the same object. The same surfaces were also included in different objects. Recognition was accomplished by finding an object in the model data base that is consistent with the surface characteristics discovered by the sensors, satisfying the adjacency relations between surfaces, and imposing constraints on the relation between the object centroid and each surface. These constraints are used with touch data points in order to locate the object. All objects were recognized correctly using two recognized surfaces. The present system is restricted to objects bounded by planes, cylinders and spheres. Although there is no reason why the system would not be extended to general second order surfaces such as hyperboloids and paraboloids. The results of this research can be applied in intelligent robotics and machine parts inspection.
Dedicated to my wife Hala, 
my children, 
and 
my mother Isabelle
Acknowledgements

"Neither he who plants is anything, nor he who waters, but God who gives the increase." [1 cor 3:7]

I wish to express my deepest thanks to my advisor, Prof. Francis L. Merat. His expert guidance, sincerity, interest, enthusiasm and above all, the inordinate amount of time he spent shaping me into a productive and hard working researcher enabled me to achieve the results presented in this dissertation. I truly appreciate my other committee members, Dr. Paul Claspy, Dr. Stephen Phillips, and Dr. Behnam Malakooti, for their valuable contributions to the dissertation.

I also wish to express my appreciation to the staff of the Medical Center Company for giving me the time to write this manuscript.

This dissertation has been dedicated to my wife and my parents. Without their love and patience, this work would not have been possible.
# Table of Contents

Abstract .................................................................................. ii
Acknowledgements ..................................................................... vi
Table of Contents ....................................................................... vii
List of Figures ............................................................................ xi

Chapter 1 - Introduction ............................................................. 1
  1.1 Overview ............................................................................. 1
  1.2 Problem Definition ............................................................. 2
  1.3 Model-Based Recognition .................................................... 3
  1.4 Integration of Multiple Sensors .......................................... 4
  1.5 Robotic System Performance .............................................. 5
  1.6 The Approach ................................................................. 6
  1.7 System Description ........................................................... 8

Chapter 2 - Related Research and Background ......................... 12
  2.1 Introduction ........................................................................ 12
  2.2 Object Models For Recognition .......................................... 12
  2.3 Image Processing ............................................................. 14
  2.4 Low-Level Object Characteristics ..................................... 15
  2.5 Segmentation ..................................................................... 16
  2.6 Calculation of Surface Equations ....................................... 18
Chapter 5 - Touch Sensing

5.1 Introduction

5.2 Experimental Touch System

5.3 Organization of Touch Data Processing

5.4 The Approach Phase

5.5 The Alignment Phase

5.6 Summary

Chapter 6 - Integrating Vision and Touch

6.1 Introduction

6.2 Composing Tactile and Vision Surface Normals

6.3 Surface Classification

6.4 Surface Equation Estimation

6.4.1 Estimation of Planar Surface Equation

6.4.2 Estimation of Cylindrical Surface Equation

6.4.3 Estimation of Spherical Surface Equation

6.5 Estimation of the Normalized Surface Equation

6.6 Summary

Chapter 7 - Matching

7.1 Introduction

7.2 Design of the Matcher

7.3 Initial Matching

7.4 Sensed Surface Orientation
List of Figures

Figure 1.1 - System overview ----------------------------------------------- 10

Figure 3.1 - Model features domain ----------------------------------------- 27
Figure 3.2 - Model surface representation ------------------------------- 32
Figure 3.3 - The coordinate systems and angles of rotation ----- 32
Figure 3.4 - Position Representation of surfaces ----------------------- 24
Figure 3.5 - The object centroid location in case of
               a spherical surface-------------------------------------- 35
Figure 3.6 - The object centroid location in case of
               a cylindrical surface----------------------------------- 36
Figure 3.7 - The object centroid location in case of
               a planar surface------------------------------------- 37
Figure 3.8 - Common surface arrangement ------------------------------- 38
Figure 3.9 - Tree structure arrangement ------------------------------- 38
Figure 3.10 - A model object ---------------------------------------- 39
Figure 3.11 - The model representation of
               the object "Example"--------------------------------- 40

Figure 4.1 - Block diagram of the feature extraction
               process and the feature space ----------------------- 43
Figure 4.2 - Point level algorithm -------------------------------------- 35
Figure 4.3 - Decomposition of \( v_j \) ---------------------------------- 38
Figure 4.4 - Patch level algorithm -------------------------------------- 49
Figure 7.7 - Matching of cylindrical surfaces .......................... 93
Figure 7.8 - The centroid geometric primitive and the
discriminant for a planar surface ................................. 94
Figure 7.9 - An incorrect matching after the discriminant ...... 95
Figure 6.10 - Determination of object location ..................... 97

Figure 8.1 - Surface normals on a planar surface
    a. Surface normal from vision analysis ...................... 101
    b. Touch data for surface normal calculations .......... 101

Figure 8.2 - Surface normals on a cylindrical surface
    a. Surface normal from vision analysis ...................... 103
    b. Touch data for surface normal calculations .......... 103

Figure 8.3 - Surface normals on a planar surface
    a. cross section showing surface normals
       from vision analysis ......................................... 104
    b. Touch data for surface normal calculations .......... 104

Figure 8.4 - Image of a plane with added Gaussian error
    with standard deviation of 1 mm ............................ 107

Figure 8.5 - Planar vision - touch patch (within borders)
    using touch point at (32, 32) ............................... 107

Figure 8.6 - Image of a cylinder with added Gaussian error
    with standard deviation of 1 mm ............................. 108

Figure 8.7 - Cylindrical vision - touch patch (within borders)
    using touch point at (32, 13) ............................... 108
Figure 8.8 - Image of a sphere with added Gaussian error with standard deviation of 1 mm 109

Figure 8.9 - Spherical vision - touch patch (within borders) using touch point at (35, 40) 109

Figure 8.10 - Test object 1 "cylinder & sphere"
    a. Object configuration 111
    b. Object range image 111

Figure 8.11 - Model description for test object 1 "cylinder & sphere" 114

Figure 8.12 - Object recognition and localization results
    For test object 1 115

Figure 8.13 - Test object 2 "cylinder & block" 116

Figure 8.14 - Model description for test object 2 "cylinder & block" 119

Figure 8.15 - Object recognition and localization results
    For test object 2 120
CHAPTER 1

INTRODUCTION

1.1 Overview

The central idea of this dissertation is to use tactile information in conjunction with visual information to locate and recognize three dimensional opaque solid objects. The vision system provides three dimensional information about regions of interest on viewed surfaces and the touch system provides measures about sensed surface points of the same object. A robust procedure has been developed to integrate the visual and tactile data into an accurate three dimensional object description. This integration results in one set of data representing the combined geometric visual and tactile measures of the sensed surfaces. These measures are used to select model objects that are consistent with the sensory data. A number of experiments have been performed using real experimental tactile sensors, force-torque sensors and real (noisy) range image data to demonstrate the utility of such a system to recognize and locate objects that would be difficult for a system using vision alone.

This chapter defines the problem addressed by this research and focuses on the approach that has been taken to solve the problem. It also
outlines the present state of robotic system performance for object recognition. An overview of the system hardware and software is included with succeeding chapters describing the system and its performance in detail.

1.2 Problem Definition

The goal of this research is to develop a model-based object recognition and localization system using vision range image data and touch data acquired using force-torque sensing.

A range image contains a set of elements which records distances from surface points to the image plane. These distances are calculated before an image is formed and stored directly as the image data. The touch data consists of a set of surface points and a corresponding surface normal, a directional vector perpendicular to the surface at the associated surface point. Surface points and surface normals are called surface characteristics. Vision data processing extracts these surface characteristics from a range image. Touch data analysis obtains surface characteristics, for the same viewed object, from information provided by tactile and force-torque sensors.

Given a set of surface characteristics extracted from the vision system and another set (for the same object) derived from the touch data, the process that combines these two sets into a single set is called vision-taction integration. The new set of data consists of vision-touch surface
patches which contain surface points extracted by the vision system correlated with at least one touch surface point extracted by the touch system.

The object recognition task is to generate a description of the sensed object in terms of surfaces and their spatial relationships. A description of a known object is called a model description and the known object is called a model. The model data base is a library of model object descriptions. The recognition task is to match the extracted vision-touch patches against the stored model surface descriptions to select a matched model. This matched model gives the description of the object explored for recognition.

The object localization task is to find the position of the object centroid (the geometric center) in a real world coordinate frame. This frame is the coordinate system defined by the robot manipulator used in collecting the touch data. The xy plane of this coordinate frame is the plane containing the base of the manipulator and the z axis is the directional vector perpendicular to that plane.

1.2 Model-Based Recognition

If robots are to use sensory data, they must know how the data relates to the perceived environment. Higher level knowledge about the world needs to be invoked to put the lower level sensory data into context. Model based object recognition is the paradigm that allows higher level
knowledge about the object to be encoded and assist the recognition process. Recognition has two components, a data driven or bottom up component that supplies low level sensory primitives and features, and a high level that utilizes these features to understand an object.

High level knowledge about the object is usually contained in models that relate the observables to the actual objects. This model information must be computable from the sensor information and must contain criteria that facilitate efficient matching of the model to the sensed object [14,21,23]. Matching is to find a model with a description similar to the extracted descriptions from vision and touch sensing.

1.3 Integration of Multiple Sensors

Most sensor related work in robotics has tried to use a single sensor to determine environmental properties [17,30,37,40,52,61,68]. This is a serious limitation because present robotic sensor systems are not able to determine many important environment properties. For example, a vision system using a 2-D camera has difficulty determining 3-D shape. The approach taken here is to use multiple sensors in a complementary fashion to extract more information from the environment than a single sensor [31,49,64].

Many different operators and methods have been developed by various researchers trying to extract depth and surface information from visual data [8,12,31]. Examples of these are descriptions from
stereo [36], descriptions from shading [42], descriptions from contour [41,65] and descriptions from texture [7,71]. A potentially promising idea is to use all of these separate operators together in a system that will integrate their results. Unfortunately, the operators all have different sets of constraints on the object’s structure, reflectance, and illumination [9,27,52,57]. The integration of these many visual methods is still not well understood. A much more promising approach is to supplement the visual information with other sensory information that directly measures some of the desired object properties. The strategy of trying to obtain enough information for object recognition from a single sensor may fail due to the limitations of that sensor as is typically the case with machine vision. If this vision sensing can be supplemented with tactile and force-torque information that directly measures object descriptors, more robust and error free descriptions of object structure can result as demonstrated through the research presented here.

1.4 Robotic System Performance

Robotic systems have limited capabilities for performing complex tasks [9,33,49]. Typical robotic tasks such as those which consists of pick and place type operations depend on a pre-taught series of movements according to the task at hand. Continued operation of the robot assumes no change in the work environment from the teaching sequences. Some of the complex tasks for which robots are needed include industrial applications, parts inspection, and manipulation in which pre-
programmed sequences cannot be used. Object recognition and localization is a central issue in such tasks [14,35,52,69]. The intent of this research is to achieve the task of object recognition and localization using multiple sensors. The strategies used for selecting such sensors are based on resolving the conflict information acquired from the different sensors and to optimize the use of the data provided by them in order to produce a complete three dimensional description of the object under investigation.

1.5 The Approach

The first step taken by this research was to identify the domain of objects as a subset of objects bounded by quadric surfaces. That is, object surfaces bounded by planar, cylindrical and spherical surfaces and are described by a quadratic equation. This domain includes about 85% of presently manufactured objects [50] and can be easily expanded to include other quadric surfaces.

The second step was to establish a criteria to build a model database. The model descriptions represent view independent object geometric properties and tactile features. The model description contains a complete surface adjacency graph of an object. Such a graph has a surface description as a graph node and the adjacency relation between surfaces as a graph edge. Each edge of the graph contains a normalized surface equation for that surface. In addition, the node
contains a geometric primitive which describes the relationship of that surface to the object centroid. Each model contains an object name and constraints upon the object centroid imposed by the model surfaces which will be used to efficiently recognize and locate objects.

The third step was deciding how to extract surface characteristics from range images. A raw range image provides only distance information with no information about how to group image elements into features. The method developed in this thesis is to extract surface normals at surface points and to group points having similar properties of their normals into regions called surface patches.

The fourth step was to develop methods for finding surface characteristics of the same object from touch data. These surface characteristics should be similar, in description, to those extracted from range images. Sensory information provided by an integrated tactile / force-torque sensors was used to achieve this goal.

The fifth step was to integrate the extracted features from vision and touch into combined vision-touch surface patches. These patches are processed for classification and surface equation determination. The surface equations were calculated for each surface using a least-square method that needs only a few touch and vision surface points.

The sixth step was to develop methods for recognizing and locating objects using extracted vision-touch surface patches and model descriptions. The object recognition task is to find a model such that
there exists a mapping from sensed surface patches to model surfaces satisfying: First, each surface patch maps to a surface of this model with the same surface properties and second, the constraint upon the relative location of the object centroid to that surface is maintained. These constraints are referred to as the discriminant and are defined as the geometric primitive that relate each surface to its object centroid. The mapping mentioned above is called matching.

After the object had been recognized its location in the robot world coordinate frame can be readily determined. Since the touch system provides accurate coordinates of surface points, the problem is reduced to finding the location of the object centroid relative to any touch point. This relative location was determined during the recognition process using the discriminant. Simple coordinate transformations are then applied to calculate the absolute location of the object centroid in the real world coordinate frame, q.v. the robot coordinate frame.

1.6 System Description

Current object recognition systems can deal with three types of objects: polyhedral in which objects can be described as polyhedra; objects bounded by curved surfaces; and objects containing curved surfaces but with some constraints; for example, objects bounded by spheres and cylinders [15]. The objects used in this research are objects bounded by planar, spherical and cylindrical surfaces. This work is
significant since existing recognition systems do not provide satisfactory solutions for them [14]. Hidden surface details pose serious problems for many visual systems in recognizing such objects. Chapter 2 reviews previous approaches for object recognition.

The experimental hardware is described as follows: The object to be explored is rigidly placed in the workspace specified by the robot manipulator. A pair of tactile and force-torque sensors are mounted on a two finger gripper attached to a six degree of freedom robot manipulator arm. The gripper fingers are closed on a probe pointing in the direction of the tool z-axis of the manipulator. The gripper and the sensors are controlled by commands and software written in "C" running under the Unix operating system. The vision experiments have been done on a Silicon Graphics IRIS 1400 Workstation and a Vax11/780. Real and synthetic range images were used to test the ability of the recognition system described above. Real images came from the University of Utah range image database\(^1\). Synthetic images with added noise were generated on the IRIS 1400 Workstation. The touch data was collected and processed by the touch system. Both vision and touch data were transferred to the VAX for vision-touch integration and object recognition and localization.

Figure (1.1) is an overview of the system architecture. The system consists of six subsystems: the model data base, the vision

\(^1\) University of Utah, Salt Lake City, Utah 84112 USA
subsystem, the touch subsystem, the control subsystem, the integration subsystem and the matching subsystem.

Figure 1.1 - System overview
Chapter 3 describes the object model form used in this dissertation. Chapter 4 describes how surface descriptions are generated from vision range images. Chapter 5 describes both the control module and the touch subsystems. The control module controls the touch subsystem. This is not a feedback control system; rather it synchronizes the robot manipulator movements with the data collected from the tactile and force-torque sensors. The touch subsystem processes the touch data to provide the required tactile surface characteristics. The procedures developed and used to build high level surface and feature descriptions of the sensed objects by integrating the results of vision and touch sensor systems are described in Chapter 6. The matching subsystem matches the integrated data to the model data. The system fulfills its goal when the object is recognized and located using the algorithms described in Chapter 7. Finally, conclusions and recommendations for future work as an extension to this research are made in Chapter 8.
CHAPTER 2

RELATED RESEARCH AND BACKGROUND

2.1 Introduction

Issues in object recognition and localization relevant to this research include three dimensional modeling (description of known objects), object description (characterizing sensed objects from sensory information), extraction of surface characteristics (surface normals), tactile recognition (extracting object descriptions from tactile data), and object recognition and localization (classifying a sensed object as a known object and estimating its location). There has been much research done in solving these problems using vision or touch individually. This chapter discusses previous approaches for each of the above issues.

2.2 Object Models For Recognition

The task of object recognition extensively uses object models for recognition and localization. Most existing systems have similar descriptions for sensed objects and models. These models describe the object in terms of its geometry, topology and include relational
information about the objects. Robert [59] created one of the first model representations for vision systems by modeling an object as vertices and edges, called a wireframe description. This method is simple but not applicable to non-polyhedral objects. A sphere for example, does not have a vertex nor an edge.

Later, as researchers explored shape classification, generalized cylinders or cones [2,48,60] were used as primitives. This type of description can characterize surfaces of revolution easily and compactly. Yet it is not suitable for describing many real world objects since it uses a 3-D axis and a 2-D cross-section function which are very hard to recover from images. Moreover, there can be many generalized cylinder representations of the same object which causes problems in object recognition.

A good and efficient approach for model description in a hierarchical manner is Constructive Solid Geometry (CSG) [58] which describes the object in the form of a tree containing 3D volumetric primitives at its leaves and a set of operators at non-leaf nodes. Each sub-tree can be viewed as a description of a part of the object. The disadvantage of this method is that the description is not unique; an object can be described by different CSG trees. Since CSG primitives are volumetric primitives which often include a large invisible part of the object, they are very hard to extract from sensed data.

A widely used model is based upon the Surface Boundary
Description [19,37] in which objects are described by their bounding surfaces and the spatial relations between them. This type of description is straightforward and useful for recognizing and locating objects. However, it is hard to relate models to others that share some of their surface descriptions.

Another approach is the spatial occupancy description [8] which is a 3D finite element approach. It uses cubes to approximate an object with the object described as the union of these cubes. This description does not directly provide object characteristics or spatial information helpful for locating objects but may be good for matching.

2.3 Image Processing

Image processing may be required to remove noise from the image elements so that accurate object characteristics can be extracted. This noise can be removed using a Gaussian filter [20,34,43]. However, the use of Gaussian filter may make edge points harder to distinguish. Burt [20] proposed an iterative Gaussian filtering algorithm which increases the standard deviation of the Gaussian filter at each iteration. This algorithm gives better results than a simple Gaussian filter [20,34,43] but it is computationally intensive [72].
2.4 Low - Level Object Characteristics

Surface normals, curvatures and edge points are common local characteristics of objects that are used as low level object characteristics [43]. These properties can be estimated from the first and second order derivatives at object points. If these derivatives are approximated by a difference operator, error will be associated with each pixel carrying inaccuracy in the calculated local object characteristics. Therefore, many methods have been developed to extract local object characteristics more accurately. Yang and Kak [73] calculate surface normals and curvatures by fitting spline patches to surface patches. The partial derivatives of the spline patches are calculated and used to determine surface normals and curvatures. This method is simple but not accurate for curvature measurement because the patches which fit object points are distorted. Besel and Jain [11,12] fit orthogonal polynomials to a surface patch and calculate the Gaussian and mean curvatures without using differentiation; however, the computation is rather complex. Ittner and Jain [43] use difference operators to estimate surface normal and curvatures, but the accuracy breaks down rapidly as data noise increases. Brady [19] applied Terzopolou's dynamic smooth operators derived from differential geometry to calculate surface normals and curvatures. This method is not reliable since differentiation was used. Wu and Merat [72] approximated partial derivatives by finite difference operators and convolved a Gaussian filter with them to calculate surface normals. This method is inaccurate in case for sparse data.
There are many methods to detect edge points. One method is to detect an edge point from an abrupt change in depth, surface normals, or surface curvatures between neighboring points [11,44]. A major problem with this approach is the error in estimating surface normals and curvatures. Another method to extract edge points is to use structured light to project the object and locate points which have sudden depth changes or break a light stripe [10,16,18]. This approach is relatively robust for edge detection but can only extract edge points and provide range data for edge points. It can extract the depth everywhere if multiple sources of structured light are used [19].

2.5 Segmentation

Segmentation is the process of breaking up an image into regions with some similar characteristic(s). For example, a region might correspond to a surface (similar geometry and topology) of an object being imaged. Further processing can then be separately applied to each of these image regions. Segmentation can be achieved by grouping points having similar low level properties together. There are two methods commonly used. The first method is region growing in which segments are extracted by merging neighboring points with similar characteristics together. The second method is the closed contour segmentation method in which edges are connected into contours enclosing the segments.

The following discussion pertains to the region growing
segmentation method. Snyder and Bildro [63] have developed a method that can grow regions from three points and avoid points where the surface normal direction changes abruptly. This method is very sensitive to noise. Oshima and Shirai [52] approximated surface patches by planar patches. Points with the same normal vector direction were merged together, first into small regions, then into large regions. Incorrect segmentation may occur using this method. Other researchers [25,26,28,29] have used the finite element method to approximate curved surfaces. Neighboring patches were merged if they could be fit to a quadric surface equation and satisfy some error criteria. This method is computationally costly because the surface equation has to be calculated using eigenvalue computations in each iteration.

Segmentation methods using closed contours need to connect edges into contours. Fan, Navatia and Medioni [24] locate points containing discontinuities and link them into curves then into closed contours. The 3DPO system [17,18] segments a light stripped image into regions by extracting edge points and grouping smoothly connected edge points into edges. Neighboring edges are connected into closed contours enclosing surface regions. Tomita and Kanade [68] used a similar method. They first found edge points and then grouped smoothly connected edge points into edges. A line or a curve is fitted to each edge which are connected to form contours. However, all methods of this type use many heuristics to extract contours which may result in incorrect segmentation when applied to different problem domains.
2.6 Calculation of Surface Equations

Many characteristics can be extracted from a surface, such as the surface size, the derivative of surface normals, the centroid, the surface class, lines of curvatures, and the surface equation \([13,23]\). Most properties, except the surface equation, can be calculated in a straightforward manner from low-level information. Therefore, this discussion is limited to methods for calculating surface equations.

The extraction of a planar surface equation has been well studied \([38]\). Non-planar surfaces are described by parametric equations or polynomials. Potmesil \([56]\) fits a parametric bicubic surface patch to each surface region. Vermuri and Aggarwal \([70]\) fit a tension spline patch to each window and merge smoothly connected points to form surfaces. Dane and Bajcsy \([23]\) fit quadric surfaces to each region using the least squared error surface fitting techniques. Solina and Bajcsy \([64]\) fit a superquadric surface with minimal volume to each region based on a minimization technique. However, their method requires quite accurate initial estimation and very complex computation. Brady et. al \([19]\) extract symbolic surface descriptions from curvature properties. The problem is that the surface view-independent information, which is very important in locating objects, may be lost during processing.

2.7 Touch Sensing Capabilities

While vision remains the primary sensing modality in robotics,
interest in tactile sensing is increasing. Harmon [39] has surveyed many researchers in the field of robotics and reported that most viewed that tactile sensing concurrently with vision should form a better recognition system than vision alone. Touch sensing is important for recognition tasks, assembly, parts fitting work, and inspection tasks. Tasks that call for close tolerances or low absolute error can benefit from a tactile approach.

Tactile sensors vary in their ability to sense a surface. At the lowest level, simple binary contact sensors such as microswitches report three dimensional coordinates of a contact point. The next level of sensors report gray scale values that are proportional to the force or displacement on the sensor. Useful properties that remained unexploited are temperature and hardness sensing. The geometries of these sensors vary from a single sensor to planar array of sensors to a finger like array covered with sensors. Most research in tactile sensing has centered on the transduction technology. A number of technologies including microswitches, strain gauges, piezoelectric materials, and conductive elastomers have been utilized. For a thorough review of these technologies see Harmon [38]

2.8 Tactile Recognition

Four different approaches to object recognition using tactile sensing have been taken by recent researchers. The first approach uses
a statistical pattern recognition method based on the assumption that only statistical parameters are invariant. The measured statistics are compared to reference statistics for known objects. An early effort in this approach was the work of Kinoshita, Aida, and Mori [44]. They utilized a five fingered hand containing twenty two binary sensors to discriminate between objects. Each object was grasped from a number of different angles, the binary pattern was recorded, and a discriminating plane was calculated in the sensor space from these learning samples. A similar approach was used by Okada and Tsuchiya [51] who used an eleven degree of freedom three fingered hand to grasp objects and form binary patterns with the hands contact sensors. Ma [45] and Togai [67] used a flat tactile pad to detect 2D geometrical shapes such as a triangle, trapezoid, rectangle, square, hexagon, and circle. The problem with this approach is that it is limited in capability and can only discriminate among a few simple type of objects.

The second approach uses kinesthetic information to compute the coordinates of contact points from which the object shape is derived and compared to a stored model. Bajcsy and Goldberg [6] used cutaneous information to detect contact. The set of contact points is processed to obtain volume and centroid information for identification. Bajcsy and Hager [5] suggested three tactile primitives: surface normal, surface curvature and hardness. Based on these primitives, a Gaussian spherical representation for objects can be developed and used as basis
for matching against models. The problem with these methods is that they neglect of the importance of the cutaneous information which can greatly simplify the task of recognition.

The third approach is a feature extraction based methodology in which tactile images are processed to identify object features which are compared to stored models. The object with the best match is the object identified. Ozaki et al [54] treated objects as containing parallel slices sensed by a special gripper. The gripper consisted of seven contact surfaces with tactile sensors which were wrapped around an object’s contour and reported the unit normal distribution along the contour. This distribution was then matched with a set of model distributions to try to discriminate shapes. This system did not work well with objects that could not easily be described as a series of slices. Overton [53] described a tactile sensor organized in a tactile array capable of yielding gray value information proportional to force extracted on each sensor in the array. Simple vision array operators were used to distinguish patterns of tools from static sensing. A similar effort was reported by Hillis [40] who used a very high spatial resolution tactile sensor to distinguish small objects as screws, clips, bolts etc. His approach was to use traditional gray level image processing techniques on the array values to find bumps and holes on the surface. Because the sensor was larger than the object, static sensing was used. An attempt was made by Bajcsy [4] and Allen [3] to interpret sensor imagery over time and integrate the results, a departure from static sensing. They used a flat
pad array of conductive elastomer sensors manufactured by the Lord Corporation.

The fourth approach combines more than one of the above mentioned approaches. Gaston [32] and Grimson [36] defined a set of constraints (distance, angle and direction) and used a tree pruning technique to eliminate models with no feasible interpretation. The model which satisfied all these constraints was the right one. Chen and Loparo [22] used a 16x16 element tactile pad to get images of solid objects. They represented these objects as a set of enclosing surfaces. Local surface features were extracted and a surface interpolation technique was used to build the surfaces. Surface matching arranged as a multi-level decision tree was then used for object recognition. The problem with this approach is that it is only applicable for simple polyhedral objects.

2.9 Object Recognition and Localization

Model-Based object recognition has become a popular approach in recent years because this method provides a flexible and general solution for object recognition [14,21]. This approach usually extracts structured viewed object descriptions and recognizes objects by finding a model with a similar structural description. The problem with this approaches is the complexity of calculations involved in checking the similarity between two structural descriptions. Some previous
approaches organized the model objects in a hierarchical structure. Recognition is based upon finding an end node of the hierarchy using matched object characteristics [17]. The problem with this method is the dependency of the recognition process on important object characteristics. Unfortunately, it is possible that some of these characteristics may be lost during the processing. Other approaches impose constraints to reject incorrectly matched models then verify that the object found is the correct match [13,17,35,37,47]. The problem with these approaches is that much effort is spent on rejecting incorrectly matched objects. Yet other approaches recognize objects by checking whether a constrained object location could be found for the viewed and the model objects [9,10,13] but they are limited to polyhedral objects.

In this research, the object is described in terms of bounding surfaces represented in the form of surface patches. This type of description is suitable for the object recognition task developed here since the data processing extracts only surface patches. The model based object recognition method matches the extracted surface patches against the stored model surface descriptions to select a matched model. Incorrectly matched models are rejected early in the process to reduce the effort normally spent in these techniques. The arrangement of the model objects is chosen to efficiently recognize and locate objects. This research contributes to methods for extracting surface patches from vision and touch, and to methods of recognizing and locating their corresponding objects.
CHAPTER 3

MODEL DATA BASE

3.1 Introduction

The model data base encodes the high level knowledge about the objects which are to be recognized. The object structure which is encoded in the models is used to understand and place in context the low level sensing information. The design of the object models was influenced both by the object domain and the task of object recognition. The object domain consists of objects with planar, cylindrical and spherical surfaces. The task of recognition employs sensors that see surfaces and touch surface points. The models include surface descriptions and contain relational information that constrain matches between sensed and model objects for easier and faster recognition.

This chapter describes the modeling procedure used in this dissertation along with the criteria used to build the models. The order in which the models are arranged in the model data base is presented and an example of an object model is given.
3.2 Criteria for a Recognition Model

As discussed in the previous chapter, there is a wide range of primitives and methods to organize them into three-dimensional models for recognition. Because no one model is necessarily best, it is important to establish a good criteria in deciding upon the structure of an object recognition model. The following subsections describe the criteria that have been established and used as a basis for the design of the object models used in this research.

3.2.1 Computability from sensors

A model must be in some way computable from the sensory information provided by the low level sensors. If the model descriptions are very different from the sensory information, then transformations, which may not be information preserving, are necessary. This lost information may make the recognition process slow and inefficient. A better situation occurs when the model primitives are directly related to the sensing information.

3.2.2 Preserving Relations Between Object Parts

Objects to be modeled must be broken down into manageable parts. In this dissertation, the object is decomposed into portions of uniform geometric characteristics. Such object portions usually form some geometric shapes called features. Accordingly, the object is
represented as a structural combination of features and object
ccharacteristics associated with them. Maintaining relationships
between the parts in the model is essential. A complete surface
adjacency graph of objects is used to maintain such relationships in this
dissertation. Such a graph has a surface description as a graph node
and an adjacency relation between surfaces as a graph edge. In
addition to the surface adjacency graph, a geometric relation between
each surface in the object and the object centroid is encoded in the object
models. These geometric relations add more geometric information
about the objects and are used to guide the object recognition process and
help in the object localization procedure.

3.2.3 Feature Specification

Feature based matching has been a useful paradigm in
recognition tasks. If these features are computable, then they need to be
modeled explicitly as an aid in the recognition process. The more
computable features that are modeled, the better the chance of correct
matching.

In this dissertation, the features used in modeling the objects are
defined as a closed set of surface points which have similar geometric
characteristics that involve vision and touch surface points. These sets
of points are referred to as vision-touch patches, VTP, and have specific
formats explained later in this chapter. If two VTP's belong to two
different surfaces, the manner in which they are connected together
describes how the object is formed. Accordingly, the model features consist of objects, surfaces, VTP's, neighboring relations, and the geometric relation between each surface and the object centroid. These features are arranged as shown in Figure (3.1)

![Diagram showing object relationships with surfaces, vision-touch patches, and relations]

Figure 3.1 - Model features domain

3.2.4 Ability to Model Curved Surfaces

Some domains may be constrained enough to allow "blocks world" type polyhedral models; however most domains need the ability to model objects that contain curved surfaces. To provide such an ability, object's curved surfaces must be mathematically presented in a unified coordinate frame. This representation eases the process of distinguishing the surfaces. The model is described in a coordinate system such that transformations between this frame and the surface
coordinate frame can be easily calculated. This reduces the amount of information encoded in the model and the calculations involved in matching an object against models.

Two coordinate frames are defined for each object: a model object coordinate frame and a surface coordinate frame. The model object coordinate frame has its origin at the model object centroid and has its axes parallel to a main coordinate frame, i.e. the robot coordinate system. The surface coordinate frame is the coordinate frame with axes coincident with the principal axes of the quadric surface. The origin of this system is located at a specific point defined by each type of surface. The main coordinate frame is a frame representing the robot manipulator with the z axis aligned parallel to the tool z axis.

An analytic surface description in the format represented by Equation (3.1) is used to define the surface in the object coordinate frame. A translation matrix giving the transformation of the object to the main coordinate frame is calculated and associated with each object.

\[
\text{Surface} = (\text{Eq.}, \ C, \ \theta_{x,y,z}) \tag{3.1}
\]

where:

Eq: is the normalized quadric surface equation

C: is a specific point described by each surface type.

\(\theta_{x,y,z}\): represents the angles of rotation used to rotate the surface coordinate frame to coincide with the object coordinate frame. These angles specify the orientation of the surface.
3.3 The Model Description Form

The criteria discussed in Section 3.2 has been used to build model descriptions that represent specific position and orientation invariant properties for objects. These descriptions contain only geometric features similar to the features extracted from the sensed objects. The models are described in terms of surfaces and surface adjacency relations.

The adjacency between surfaces is a complete surface adjacency graph that indicates whether two surfaces are connected. A surface is described by its identifier, class, position, and orientation. Another geometric primitive that defines the location of the object centroid with respect to the surface is also included. Three types of surfaces are used: planar, cylindrical and spherical. The model is discussed in more detail in the following subsections.

3.3.1 Description of Model Surfaces

A surface is represented by the form shown in Figure (3.2). Each surface is identified by a unique identification number. This number is used by the surface adjacency graph to show the relation between that surface and its neighbors. This relation appears as a list of surface identifier pairs of neighboring surfaces on the model description shown in Figure (3.2)
Any quadric surface can be represented by a quadratic equation. This equation is calculated in a coordinate frame whose origin is at a point on the surface and has a general form given by Equation (3.2).

\[ f(x,y,z) = q_1 x^2 + q_2 y^2 + q_3 z^2 + q_4 xy + q_5 xz + q_6 yz + q_7 x + q_8 y + q_9 z + q_{10} = 0 \quad (3.2) \]

or in matrix notation:

\[ P^T Q P = 0 \quad (3.3) \]

where:

- \( P \) is a point on the surface represented by the homogeneous coordinates

\[ P = (x, y, z, 1) \quad (3.4) \]

and \( Q \) is the coefficient matrix given by

\[
Q = \begin{pmatrix}
q_1 & q_4' & q_5' & q_7' \\
q_4' & q_2 & q_6' & q_8' \\
q_5' & q_6' & q_3 & q_9' \\
q_7' & q_8' & q_9' & q_{10}
\end{pmatrix} \quad (3.5)
\]

If this equation is calculated in a coordinate frame having its
origin at the center when the surface is spherical, or at a point on the central axis when the surface is cylindrical, or at a point on the surface when the surface is planar, it is reduced to its normalized form. These specific points will be referred to, without further explanation, as the surface centroid. In general, the axes of the coordinate frame are coincident with the principal axes of the surface where the z axis is always pointing in the direction of the surface normal and the y axis is in the central axis direction (the line in the center of the cylinder and perpendicular to its surface normals), if exists. The Q matrix given by Equation (3.5) reduces to $Q_n$ (the normalized coefficient matrix) as shown in Equation (3.6) under these conditions.

$$Q_n = \begin{pmatrix} q_1 & 0 & 0 & 0 \\ 0 & q_2 & q_{8/2} & 0 \\ 0 & 0 & q_3 & 0 \\ 0 & q_{8/2} & 0 & q_{10} \end{pmatrix} \quad (3.6)$$

The orientation of the surface is represented by three angles defined as $(\alpha_m, \beta_m, \gamma_m)$. The angle $(\alpha_m)$ represents a rotation about the x axis and the angle $(\gamma_m)$ a rotation about the z axis. The third angle $(\beta_m)$ is used to specify the angle of rotation about the new rotation axis defined by $(\alpha_m)$ and $(\gamma_m)$. The choice of the surface coordinate frame described previously makes the new rotation axis the y axis of the coordinate frame after the first two rotations. Figure (3.3) shows the coordinate axes and the angles of rotations.
Surface I.D. /* number that identifies the surface */
Surface Class /* the type of the surface */
Normalized Equation /* the Qn matrix for the surface */
Surface Orientation /* three angles of rotation ($\alpha_m$, $\beta_m$, $\gamma_m$) */
Surface Position /* three fields: */
  Type: (0, 1, 2)
  Point: (x, y, z)
  Vector: (v_x, v_y, v_z)
Object Centroid /* geometric representation of object centroid */
  Point: (x, y, z)
  Vector: (v_x, v_y, v_z)
Adjacency Relation /* Neighboring surface list */

Figure 3.2 - Model surface representation

Figure 3.3 - The coordinate systems and angles of rotation
The rotation angles can be determined from the matrix $Q$ as follows. Let the upper-left 3x3 submatrix of $Q$ defined in Equation (3.5) be denoted by $Q_{ul}$. This matrix contains the coefficients of all second order terms as shown in Equation (3.7). The eigenvectors of the matrix $Q_{ul}$ give the three rotational angles $\alpha_m, \beta_m, \gamma_m$.

$$Q_{ul} = \begin{pmatrix} q_1 & q_{4/2} & q_{5/2} \\ q_{4/2} & q_2 & q_{6/2} \\ q_{5/2} & q_{6/2} & q_3 \end{pmatrix} \quad (3.7)$$

The position of the surface is represented by three fields: a surface type, a vector and a point. Three surface types are used: type 0 for spherical, type 1 for cylindrical and type 2 for planar surfaces. In the case of spherical surfaces the vector is not used and the point is chosen to be the center of the sphere. In the case of cylindrical surfaces the vector represents the central axis orientation while the point is chosen to be any point on the central axis. If the surface is planar, the vector gives the surface normal and the point is any point on the plane. (This point will be considered the origin of the plane's coordinate.) Figure (3.4) shows the location representation of the three surface types in a tabular form.
<table>
<thead>
<tr>
<th>Surface</th>
<th>Type</th>
<th>Vector</th>
<th>Point</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spherical</td>
<td>0</td>
<td>Not used</td>
<td>Centroid</td>
</tr>
<tr>
<td>Cylindrical</td>
<td>1</td>
<td>Central axis</td>
<td>Any point on central axis</td>
</tr>
<tr>
<td>Planar</td>
<td>2</td>
<td>Surface normal</td>
<td>Any point on the plane</td>
</tr>
</tbody>
</table>

Figure 3.4 - Position representation of surfaces

3.3.2 The Object Centroid Geometric Primitive

For each surface description in a model there is an associated geometric primitive which relates the surface to the object centroid. It consists of two fields: a point and a vector as shown in Figure (3.2). The point field has the x, y and z coordinates of the object centroid in the model object coordinate system; and the vector field has the components of the normalized directional vector from the surface centroid to the object centroid. This representation differs for each surface type. In the case of spherical surfaces, the distance between the center of the sphere and the object centroid can be defined uniquely. The object centroid is then located as a point as shown in Figure (3.5).
Cylindrical surfaces are specified by a central axis. In this case, the surface centroid could be any point on the central axis. In order to locate the object centroid relative to the surface, the perpendicular distance between the object centroid and the central axis is calculated and used to define a line parallel to the central axis. This line contains the object centroid and is referred to as the centroid geometric primitive for cylindrical surfaces. The constant distance, defined as the *discriminant*, will be used to constrain the matching process. This situation is shown in Figure (3.6).

Planar surfaces are specified by their surface normal direction. The surface centroid is any point on the plane and the object centroid is at a constant distance from the plane. This constant distance, the discriminant, defines possible locations for the object centroid as a plane parallel to the surface that contains the object centroid. This plane is the centroid geometric primitive for planar surfaces as shown in Figure (3.7).

![Diagram](image)

**Figure 3.5** - The object centroid location in case of a spherical surface
$P_s$: Surface centroid

$P_m$: Object centroid

$C_a$: Central axis

The Discriminant is the distance $d$

The Object Centroid Geometric Primitive is the line CGP

Figure 3.6 - The object centroid location in case of a cylindrical surface

$P_s$: Surface centroid

$P_m$: Object centroid

$N$: Surface normal

The Discriminant is the distance $d$

The Centroid Geometric Primitive is the plane CGP

Figure 3.7 - The object centroid location in case of a planar surface
3.4 Arranging the Models in the Model Data Base

All the models stored in the model database form the domain of objects to be recognized. Each object has an identification number and a name followed by the description of the object surfaces. There exist two means by which objects are arranged and linked together. The first is by the existence of a common feature between objects: a normalized surface equation. All objects that have at least one surface described by the common surface equation are combined. The second defines a tree structure which links an object model called the root model to several models defined as the tree models. These tree models can be derived from the root model by adding more surfaces or another object primitive to the root model. Examples of the two types of arrangements are shown in Figures (3.8) and (3.9).

The common feature structure has been developed to ease the process of matching the sensed objects with the model objects. The tree structure is used to reject model surfaces that are connected to different surfaces. This process is triggered by matching a surface equation and considering all base models as a possible match. Most of these models are rejected using the centroid geometric primitive stored in the models during the process of matching which is fully described in Chapter 7.
Figure 3.8 - Common surface arrangement

Figure 3.9 - Tree structure arrangement
3.5 Example model

Several model objects are calculated and stored in the model data base system using the description form and the procedures described above. Here we present a model for one object as an example. Other model objects will be shown in Chapter 8. The example model object is a right circular cylinder with a sphere on the top. This model object contains the three types of surfaces used in this research: planar, cylindrical and spherical. The object is shown graphically in Figure (3.10). The central axis of the cylindrical surface is in the y axis direction of the object coordinate frame. The object centroid is on the central axis at (0, 0, 0) which is the origin of the coordinate frame. The diameter of the sphere is 4.00 cm; the diameter of the circular base is 2.50 cm. The total height of the object is 6 cm. Figure (3.11) shows the detailed model representation for the example object.

Figure 3.10 - A model object
/** Object information **/

object id: 1
object name: Example

/** Surface Information **/

surface id: 1
surface class: Spherical

normalized equation: \( Q_n = \begin{pmatrix} 1.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 1.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 1.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & -4.00 \end{pmatrix} \)

Orientation: (0.00, 0.00, 0.00)
Surface centroid: (0.00, 0.00, 0.00)
Position: type (0)
  point (0.00, 0.00, 0.00)
  vector (0.00, 0.00, 0.00)
Object Centroid: point (0.00, 0.94, 0.00)
  vector (0.00, -1.00, 0.00)
Adjacency relation (1, 2)

surface id: 2
surface class: cylindrical

normalized equation: \( Q_n = \begin{pmatrix} 1.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 1.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & -1.625 \end{pmatrix} \)

Orientation: (0.00, 0.00, 0.00)
Position: type (1)
  point (0.00, 0.00, 0.00)
  vector (0.00, 1.00, 0.00)
Object Centroid: point (0.00, 2.06, 0.00)
  vector (0.00, 1.00, 0.00)
Adjacency relation (1, 2); (2, 3)

surface id: 3
surface class: plane

normalized equation: \( Q_n = \begin{pmatrix} 0.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 5.00 \\ 0.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 5.00 & 0.00 & 0.00 \end{pmatrix} \)

Orientation: (0.00, 0.00, 0.00)
Position: type (2)
  point (0.00, 0.00, 0.00)
  vector (0.00, 1.00, 0.00)
Object Centroid: point (0.00, 3.06, 0.00)
  vector (0.00, 1.00, 0.00)
Adjacency relation (3, 2)

Figure 3.11 - The model representation of the object "Example"
CHAPTER 4

VISION PROCESSING

4.1 Introduction

The partial machine vision problem addressed in this research is to extract surface characteristics in form of surface points and surface normals. This problem is solved by developing a method to extract low-level object characteristics. Depending on these characteristics, the image is partitioned into regions containing only surface points. These regions are called surface patches. These patches are used with tactile properties for more complete and precise representation of the surfaces to be used in the object recognition and localization process. There is no attempt in this dissertation to try to understand the full structure of an object from vision alone, but to use low level vision processing to take what is useful and reliable from a range image and to supplement it with active touch sensing.

The vision processing consists of two levels. The first level is a procedure to extract surface points and their corresponding surface normals. The second level is a patch level in which points that belong to the same surface are grouped together to form a region on a surface called a surface patch. The output of these two levels is combined with
the output of the touch system. This chapter presents the techniques used in this research to extract object characteristics from vision range images.

4.2 Extraction of Object Properties

The vision processing system in this research extracts different geometric characteristics and combines object features from range images at two different levels: point and patch levels. The patch level processing extracts features provided by the point level and passes surface patches to the higher level processing for vision-touch integration. The block diagram of the extraction system is shown in Figure (4.1).

The extracted features represent a hierarchical feature space consisting of points (low level) and patches (high level). Each single pixel in the range image is a point that provides range information. A small area that contain smoothly connected points is a patch. Surface normals and their direction distributions are object characteristics to be calculated from the patches. Using these characteristics along with similar touch point characteristics, partial surfaces consisting of vision and touch points can be formed. These areas of surface are called *vision-touch patches*. Global object characteristics such as surface class and equation are also calculated for each partial surface. The function of each of the point and patch system levels is described in the following sections.
Figure 4.1 - Block diagram of the feature extraction process and the feature space
4.2.1 Point Level

To provide suitable points to process at the patch level, the point level algorithm scans range image points and calculates a surface normal at each point. The efficiency of this process can be improved by suspending certain image areas from processing. The heuristic used is that an image area enclosed by a rectangle whose corner points are four background points and whose area is less than a preset value is a region containing only background points and will not be processed. Since range images are formed using a laser sensor with a fixed background, background points are those points known when no 3D object is present. A range image containing only background surfaces is formed and processed first to generate descriptions of background surfaces. A point fit to any background surface is classified as a background point. Only points that are expected to have object information, not background points, are processed by the point level. The algorithm used by the point level is shown in Figure (4.2).

For each sampled point a surface normal is calculated using differential geometry and finite difference operators [46]. A similar technique has been used by Wu and Merat [72] with good results. The surface normal \( N(i,j) \) at the point \((i,j)\) can be calculated by Equation (4.1)

\[
N(i,j) = \frac{u_i(i,j) \times v_j(i,j)}{|u_i(i,j) \times v_j(i,j)|}
\] (4.1)
where \( u_i(i, j) \) and \( u_j(i, j) \) are partial derivative vectors of the surface function at \((i, j)\) and "\( \times \)" denotes the vector cross product.

<table>
<thead>
<tr>
<th><strong>Input:</strong></th>
<th>Range vision image</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Output:</strong></td>
<td>Array of sampled points with their normal direction</td>
</tr>
</tbody>
</table>

1. Find background points
2. Mask out background areas
3. Scan sampled points in image x and y directions
4. Calculate the surface normal at each point
5. Form the output array

**Figure 4.2 - Point level algorithm**

The calculation of these partial derivatives is done by convolving the surface points with the 3 by 3 matrices given by Equations (4.2) and (4.3), which is equivalent to calculating the coefficients of the three first order differences in the x and y direction.

\[
\begin{pmatrix}
-1 & 0 & 1 \\
-2 & 0 & 2 \\
-1 & 0 & 1
\end{pmatrix}
\]

(4.2)
\[ v_j = \frac{1}{4} \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 2 \end{bmatrix} \] (4.3)

To reduce the noise effect and the inaccuracy produced by depth discontinuities, the image should be filtered and the calculation procedure modified. Instead of using filtering techniques, the matrices (4.2) and (4.3) are decomposed into four operators such that the center element always has a value of (0, 0). If a depth discontinuity does not exist between image points within the windows of the four operators, these operators can be added to reproduce the original matrices. Figure (4.3) shows how the matrix in Equation (4.3) is decomposed. The depth discontinuity is checked by calculating the absolute difference between two points. If it is greater than a threshold value, a discontinuity exists between the two points and they don't belong to the same surface.

\[ \begin{bmatrix} -1 & -1 \\ 1 & 1 \\ -1 & -1 \\ 1 & 1 \end{bmatrix} \]

Figure 4.3 - Decomposition of \( v_j \)
4.2.2 Patch Level

At the patch level, the points extracted by the point level process are checked for similarity. Points with similar properties are grouped to form a surface patch and passed along with touch points to the higher level for further processing and touch integration.

Many researchers have used Gaussian and mean curvature to group points into surface patches [13,21,48,69]. There are two disadvantage to use curvature. First, the second order operators used to extract curvature are very noise-sensitive. Second, the difference between a zero and nonzero curvature is not recognizable. In this work, the similarity between image points is checked using the statistical properties between the dot products of the surface normals within a window. The advantage of this method is that no curvature estimation is required nor priori object knowledge is needed. Consequently, this measure is more reliable and eliminates sources of error associated with curvature calculations.

A 5 by 5 window is used to estimate these properties with satisfactory results. If the window is centered at an element, \( p \), located at \((x, y)\), the mean and standard deviation \( \mu_c \) and \( \sigma_c \) of the dot product between the surface normal at the center of the window, \( N_{xy} \), and a normal, \( N_{ij} \), at a point \( q \) in the window can be calculated from the following Equations:
\[ \mu_c = \frac{1}{25} \sum_{i=x-2}^{x+2} \sum_{i=y-2}^{y+2} N_{ij} \cdot N_{xy} \quad (4.4) \]

\[ \sigma_c = \left( \frac{1}{25} \sum_{i=x-2}^{x+2} \sum_{i=y-2}^{y+2} \left( N_{ij} \cdot N_{xy} \right)^2 - \mu_c^2 \right)^{1/2} \quad (4.5) \]

The value of \( \sigma_c \) determines if image points within the window belong to the same surface. An experimental threshold value is used to make such a decision. If \( \sigma_c \) is less than the threshold, these points are most likely in the same surface and are merged together. The extraction process is repeated in four directions around the same point. The central point to start with is chosen such that it is not a background point, has no depth discontinuity in its neighborhood, and is not an edge point. The accuracy of this method depends upon the accuracy of the surface normal estimated at the point level process. To improve the system accuracy, a voting scheme is used as follows: A point will be merged into the currently forming patch if many points within the window containing this point are already merged into the same patch. The algorithm used for the patch level process is shown in Figure (4.4).

These patches are passed to a higher processing level which integrates the vision data and points extracted from the touch data processing into a single set of data representing a surface. This level will be discussed in Chapter 6.
| **Input:** | Image points extracted from the point level process |
| **Output:** | Surface patches |

1. Select a surface point \( p \)
2. Center a window of width 5 at that point
3. For all points within this window calculate \( \sigma_c \) and \( \mu_c \)
4. If \( \sigma_c \leq \text{threshold} \), merge the window points into the set containing \( p \). Otherwise, end the process
5. Move the center of the window to a new point
6. Repeat steps 3 thru 5 in the four directions around \( p \)

Figure 4.4 - Patch level algorithm

4.3 Summary

The machine vision processing creates surface patches from range images. Thru two processing levels, point and patch, surface normals at surface points are extracted and points having similar properties are grouped into regions called surface patches. These patches will be used by the integration algorithm described in Chapter 6 to generate vision-touch patches. A new method to extract surface normals after masking certain image regions has been described. A new method for segmenting range images using the standard deviation of the dot product of surface normals has been presented.
CHAPTER 5

TOUCH SENSING

5.1 Introduction

Chapter 4 described the methods used to extract surface patches from vision range images. Each vision surface patch contained surface points and a surface normal at each point. Similar surface characteristics (in description) must be extracted from the touch system. A pair of tactile and force-torque sensors mounted on a robot manipulator arm is used to achieve this goal. This Chapter describes the algorithms used to analyze the data obtained from these sensors to organize and synchronize the movement of the robot manipulator with the data collected. The data reported by the robot manipulator is processed to extract surface points and surface normals. The procedures and algorithms used for this purpose are described in this Chapter.
5.2 Experimental Touch System

The experimental touch system used in this research used a two fingered Lord Corporation gripper\(^\dagger\). Each finger in a Lord corporation LTS-200 tactile sensor containing a 10x16 tactile array sensor and a force-torque sensor underneath the array. A mechanical probe, aligned with the manipulator Z axis, is grasped in the Lord gripper and used to probe the surface. The system is shown in Figure (5.1).

\[\text{Figure 5.1 - The touch configuration system}\]

\[^\dagger\text{Lord Corporation Gripper Model SE680}\]
The Lord gripper was mounted on an Intelledex model 605T six-axis robot manipulator arm. A microprocessor-based control system controls the movements of the robot arm. An auxiliary computer, an IBM AT, to control the gripper and the sensors. The robot can move freely anywhere within a pre-specified workspace area. Movements between any points in this workspace are made using the most efficient path as calculated by the robot controller.

The data returned by the robot controller represent the status of the tool tip in the form of a vector array containing seven elements given by Equation (5.1).

\[
v = [x, y, z, \theta, \phi, r, h]^T
\]  \hspace{1cm} (5.1)

The first five elements are of interest in this work. The three elements \(x\), \(y\), and \(z\) are the coordinates of the tool tip, the touch point, in the robot world coordinate frame. The fourth element \(\theta\) represents the angle of the tool's projection on the XY plane relative to the X axis while the angle \(\phi\) gives the tool's angle relative to the XY (horizontal) plane. The angle \(r\) represents the rotation of the tool while \(h\) represents the arm configuration to be used for reaching a point. Changing the two angles \(\theta\) and \(\phi\) will achieve all possible movements in the X and Y directions. Determination of the direction of the tool \(z\) axis is essential in this research because it will the surface normal direction and can be easily calculated from the values of \(\theta\) and \(\phi\) in the same coordinate
frame. The angels $\theta$ and $\phi$ are shown in Figure (5.2); The robot world coordinate frame is shown in Figure (5.3)

Figure 5.2 - The $\theta$ and $\phi$ angles
Figure 5.3 - The robot world coordinate frame
Movements and returned points are stored in a data base in the robot memory. The data base can be accessed by simple commands to get, change, and modify the data to establish the movement of the probe tip.

The Lord Corporation gripper has two motor-driven fingers. These motors are driven by the amplified output of D/A converters. The position of the fingers is monitored using a Sony optical encoder. The routines controlling the gripper reside in a UNIX-based microcomputer and can be used to send commands to close and open the gripper. When the touch system is in operation a command is sent to close the gripper tightly on the probe. The gripper then remains closed while data is collected.

The Lord LTS-200 finger sensors contain both a tactile array sensor that produces a tactile image and a vector sensor which measures force and torque. The array sensor is composed of sensitive sites imbedded in an elastomeric touch surface. There are 160 sensitive sites arranged in a 10 by 16 orthogonal array with 0.071 inches between sites. Mechanical deflection is read in 16 increments of 0.002 inch increments at each site. Thus the maximum normal deflection for the array is 0.03 inches. The basic mode of transduction from the mechanical to electrical signal is optoelectric. At each sensitive site within the array, a solid-state emitter and detector look directly at each other across a short distance. The light beam across that gap can be partially obscured by a projection from the touch surface layer.
Deflection of the touch surface cause more or less obscuration which, in turn, changes the electrical response. A cross section view of a typical transducer site is shown in Figure (5.4).

![Diagram of zero and moderate deflection](image-url)

**Figure 5.4** - Sectional view of a transduction site of the array sensor
The vector sensor measures forces and moments along and about the three coordinate axes at the touch surface. This sensor consists of a Maltese Cross arrangement of four beams with a set of strain gauges on either side of each beam. Maximum force capacity varies across the touch surface ranging from 20 lbs normal at the center of the touch sensor to 8 lbs at the edges.

A microprocessor-based data acquisition system interfaces the sensor with the host microcomputer system. This interface unit accepts special commands to control the sensors and perform the necessary steps to complete a desired operation.

Interaction with the sensors is accomplished through a series of commands which allow initializing, scanning the array sensor, and monitoring the overall load condition through the data provided by the vector sensor. These images are collected and passed to the IBM-AT for processing. Upon this decision, commands are issued to the robot to perform the desired operation. Figure (5.5) shows the Lord Gripper and the LTS-200 sensors.
Figure 5.5 - The Lord Gripper and the LTS-200 sensors.
5.3 Organization of Touch Data Processing

Before the touch data processing system can be described, we must define some notation that geometrically describes the system configuration as shown in Figure (5.6). Each of the tactile sensors generates an image array of sensor element mechanical deflections. Let $I_R$ denote the 10x16 data array from the right hand array sensor as viewed from the gripper wrist, and $I_L$ denote the corresponding array from the left hand array sensor. The two vector sensors provide a 1 by 6 vector array representing the forces and moments along and about the three coordinate axis of the touch sensor surface. Similar to the image arrays, let the vector arrays be designated $V_R$ and $V_L$. The origin of the coordinate system for $I_R$ and $V_R$ is chosen to be the center element of $I_R$ and is denoted as $O_R$. The origin $O_L$ for $I_L$ and $V_L$ is similarly chosen. The vector connecting the two origins $O_L$ and $O_R$ in the direction from L to R is considered the X axis of the touch coordinate system. The middle point of this line in the coordinate center "O". The direction of the vector originating at "O" and parallel to the short tactile surface is considered the y-axis of the touch coordinate system. The remaining perpendicular direction to both x and y vectors is the z-axis of the touch coordinate system. This is the same as the tool z-axis of the manipulator.

Touch data processing is done in two phases: an approach phase and an alignment phase. In the approach phase, the probe approaches the object while the tactile images are continuously monitored until its analysis detects a contact. Once contact is established, the processing
system enters its second phase. In this alignment phase, the tactile data is monitored and analyzed to align the probe perpendicular to the surface. These two phases are discussed in more detail in the following sections.

![Diagram](image)

(a) Perspective drawing  
(b) View looking at O along y axis

Figure 5.6 - Geometric structure of the touch system
5.4 The Approach Phase

The primary analysis of the touch data consists of the extraction of the surface normal at the contact point. In this phase, the robot probe approaches the surface until contact is established between the probe tip and a surface point. The tactile images $I_L$ and $I_R$ are initially determined for the gripper grasping the probe with no surface contact. These initial values are used as a reference for any future changes in the image arrays. After any incremental motion of the probe, the image arrays are analyzed for surface contact. Depending upon the results of the analysis, another movement could be done or a surface contact could be recorded.

Analysis of the tactile image data collected after each incremental movement consists of subtracting the final from the initial reference values and calculating the absolute value of the differences between these new arrays. Due to the noise involved in these images, local averaging is done [57]. Let the absolute value of the difference between these new arrays after averaging be $\varepsilon$. If $\varepsilon$ is greater than an experimentally determined value contact is established, otherwise a request for another incremental movement is initiated. Once a surface point is located the robot records the three coordinates of the surface point and the system goes into the second phase.
5.5 The Alignment Phase

Once a surface point is detected and while the probe is still in contact with the surface, side movement of the probe is needed in order to align it with the surface normal direction at that point. These side movements are referred to as surface probing. The vector data arrays $V_R$ and $V_L$ are read and compared in the $x$ and $y$ directions of the touch coordinate system to determine that the probe is perpendicular to the surface. The criteria used for alignment is as follows:

* Let $M_x$ and $M_y$ be the absolute value of the differences between the left and right vector array moment elements in the $x$ and $y$ directions respectively.

* If $M_x > M_y$, then motion is required in the $x$ direction otherwise motion is required in the $y$ direction. The sign of the motion is detected by comparing the corresponding force elements in $V_R$ and $V_L$.

* If the difference between $M_x$ and $M_y$ is less than a threshold value, the probe is aligned in the direction of the surface normal at that point.

Once this phase is complete, the two robot tool angles $\theta$ and $\phi$ are determined from the robot control system. The calculated surface normal is then rotated by these angles to determine its orientation in the robot world coordinate frame. This process is repeated for more
surface points as required. More touch points from other surfaces are obtained and the touch data analysis is carried out.

The number of touch points required for integration is determined by the number of surface patches extracted from the vision system. For each vision patch, a touch point is selected and touched. The location of these points are approximately the location of the central points used as starting points for the vision patch level processing. Therefore, the touch system is visually guided by the results of the vision system.

The results of this data processing stage are recorded in an array of touch surface points. Each element in the array contains the three coordinates of each point and the surface normal at that point. This array is passed to the integration stage of the object recognition and localization system.

The flow chart of the touch data processing system is shown in Figure (5.7)
Figure 5.7 - Flow chart for touch data processing
5.6 Summary

The use of tactile sensing in robotics has been limited. Previous approaches have emphasized static sensing using traditional pattern recognition techniques. The approach taken here is to use dynamic sensing of surfaces to extract surface normals. These surface normals are to be integrated with surface patches extracted from the vision system to generate vision-touch descriptions of object surfaces. The organization of the touch sensing involved hardware and software. The hardware included a robot manipulator, tactile sensors, force-torque sensors, a probe, and a controller. The software consisted of two phases: an approach phase in which the probe approaches a surface; and an alignment phase in which the probe is aligned in the surface normal direction. Once these surface normals are determined it is passed along with those extracted from vision for integration. The integration process is described in Chapter 6.
CHAPTER 6

INTEGRATING VISION AND TOUCH

6.1 Introduction

The data from the vision and tactile processing systems described in the Chapters 4 and 5 must be integrated to build descriptions of surfaces and features of objects that can be matched against the model characteristics in the model database. The procedures described in this chapter use both sensing modalities, then integrate the data from the sensors to build high level descriptions of what is sensed. These procedures are used to build surface features from the vision surface patches and touch extracted surface normals. These features are processed further and classified into three types of surfaces: planar, cylindrical or spherical. These are the same primitives presented in the model database. Therefore, the matching process is facilitated. The integration process is done in two steps: first, by compositing touch surface normals with vision surface patches and second, by classifying the surfaces into the three suggested types. Once the surface is classified a surface equation is developed for each type using a few surface points. These sensed surface equations are normalized and formulated to be in the same format as in the model database. These
methods and procedures are described in this Chapter.

6.2 Composing Tactile and Vision Surface Normals

The tactile data analysis, described in Chapter 5, calculates the surface normal \( N_t \) at each touch point \( P_t \). The vision data processing, described in chapter 4, provides surface patches with surface normals \( N_{xy} \) and the statistical properties of their dot products, \( \sigma \) and \( \mu \) as given by Equations (4.4) and (4.5). These properties are calculated at each surface point in the patch. To compose these surface normals, each touch point is checked with the patch points located within a window of width \( w \) for similarity existence. The correlation, \( \rho_{tv} \) between the dot product of surface normals is used for this purpose and is calculated using Equation (6.1).

\[
\rho_{tv} = \frac{1}{(2w + 1) \sigma^2} \sum_{x=-w}^{w} \sum_{y=-w}^{w} \left( N_{xy} \cdot N_t - \mu \right)^2
\]  

(6.1)

The value of \( \rho_{tv} \) determines if a vision surface patch and a touch point do belong to the same surface by comparing it with a pre-determined experimental threshold. They are considered to belong to the same surface if \( \rho_{tv} \) is greater than the threshold. The process is repeated for every touch point and all surface patches. All patches and touch points belonging to the same surface are grouped in a single patch and called a Vision-Touch Surface Patches, VTP. Since touch points are collected from surfaces than been imaged by the vision system, as
described in the previous chapter, each vision patch will correspond to some touch points. The VTP patches are passed to the next step for classification. Figure (6.1) shows the algorithm used to compose the data.

<table>
<thead>
<tr>
<th>Input: Surface patches extracted from vision and surface points extracted touch data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Output: Vision-Touch surface patches</td>
</tr>
</tbody>
</table>

1. Choose a touch point
2. Locate vision patch points inside a window centered at the touch point and of width w
3. Find the correlation of the dot product of surface normals within the window and the touch point
4. Determine if they belong to the same patch
5. Repeat for all other surface patches
6. Repeat steps 2 - 5 for all other touch points
7. Enter each group of similar touch and vision points in a queue and call it VTP(i); i = 1, 2, 3, . . .

Figure 6.1 Algorithm for composing vision and touch data
6.3 Surface Classification

After the VTPs are extracted, they are classified as planar, cylindrical or spherical. This classification is based on the mean and standard deviation of the unit vector from a touch point to a neighboring vision point in the same VTP and the normal at that point. When the surface is planar these statistics are zero, otherwise the surface is curved. Therefore, these statistics are used to distinguish a planar surface from a curved surface. The difference between a cylindrical surface and a spherical one is the existence of a central axis for cylindrical surfaces. The axis is perpendicular to the normal at every point in the surface. If the normal at a point is denoted as \( N_i \) and the central axis direction denoted as \( C_a \) then, from [46] the error of the central axis is calculated and defined by the following equation:

\[
e = \sum_{i=1}^{n} \left( C_a \cdot N_i \right)^2 \quad (6.2)
\]

The directional unit vector \( C_a \) is calculated by minimizing the error as given by Equation (6.2). This condition is satisfied if

\[
\nabla \left( e + \lambda C_a \cdot C_a \right) = 0 \quad (6.3)
\]

This condition can be rewritten as follows:

\[
\sum_{i=1}^{n} \left( N_i^T \cdot N_i \right) C_a + \lambda C_a = 0 \quad (6.4)
\]
The solution for $C_a$ is the eigenvector of the matrix $m$ given by (6.5)

$$m = \sum_{i=0}^{n} (N_i^T \cdot N_i)$$

with a minimum eigenvalue (which is the error of the vector $C_a$). If this error is less than a minimum threshold value, the surface is cylindrical, otherwise it is spherical. Figure (6.2) shows the flow chart for the surface classification algorithm.

![Flowchart](image)

**Figure 6.2 - VTP classification system flow chart**
6.4 Surface Equation Estimation

After the Vision-Touch Patches are classified as planar, cylindrical or spherical, a surface equation is developed for each type of surface. To be consistent with the model database, the coordinate system used in calculating the surface equation should have its origin at the surface centroid of a spherical surface, at a point on the central axis of a cylindrical surface, or at a touch point if the surface is planar. The surface equation calculated in this coordinate system is referred to as the Normalized Surface Equation. For numerical purposes, the surface equation is calculated in a coordinate frame that has its origin at a surface point. The equation calculated in this coordinate system is called the Calculated Surface Equation. Simple rotation and translation of the coordinate frame will transfer the calculated surface equation to its corresponding normalized surface equation.

Let the coefficient matrix, \( Q \), of the surface equation given in chapter 3 be rewritten here as Equation (6.6)

\[
Q = \begin{pmatrix}
q_1 & q_4/2 & q_5/2 & q_7/2 \\
q_4/2 & q_2 & q_6/2 & q_9/2 \\
q_5/2 & q_6/2 & q_3 & q_9/2 \\
q_7/2 & q_8/2 & q_9/2 & q_{10}
\end{pmatrix}
\]

To reduce the number of unknowns, the constant term \( q_{10} \) can be eliminated if the coordinate origin is chosen to be at a point on the surface. Since touch points must be on the surface, a touch point is
chosen as the origin. To further reduce the unknowns, an existing nonzero coefficient is chosen and normalized to one. The surface equation can now be calculated by solving eight equations in eight unknowns. Eight points are required to calculate the surface equation. These points are chosen as sparsely as possible from the VTP which was correlated to the touch point, the coordinate origin. The calculation of the surface equation for each surface type is presented in the following subsections followed by the method used to find the transformation to the normalized surface equation.

6.4.1 Estimation of Planar Surface Equation

A plane can be specified by a point on the surface and a surface normal at that point. In this case a tangent vector between any two surface points in the VTP is perpendicular to the surface normal at the chosen touch point as shown in Figure (6.3).

![Figure 6.3 - Geometry of calculating a planar surface equation](image-url)
The dot product between these two vectors should be zero. Thus, the least squared error for planar surface equation estimation is defined by Equation (6.7)

$$
e = \sum_{i=0}^{n} \left[ (P_i - P_t) \cdot N_t \right]^2 \quad (6.7)$$

Where $P_t$ is the origin, the touch point

$P_i$ is any point in the VTP

and $N_t$ is the surface normal at the touch point

For accurate estimation, this error must be minimized. This condition is achieved by normalizing the normal vector $N_t$. Let $N$ be the vector minimizing Equation (6.7) subject to the constraint $N_t \cdot N = 1$. Mathematically this can be written as:

$$\nabla \left( e + \lambda N_t \cdot N \right) = 0 \quad (6.8)$$

By expansion

$$\sum_{i=0}^{n} (P_i - P_t)^T (P_i - P_t) N + \lambda N = 0 \quad (6.9)$$

The solution $N$ for Equation (6.9) is the eigenvector of the matrix given by Equation (6.10) corresponding to the minimal eigenvalue $\lambda$.

$$\begin{bmatrix} \sum_{i=0}^{n} (P_i - P_t)^T (P_i - P_t) \end{bmatrix} \quad (6.10)$$
6.4.2 Estimation of Cylindrical Surface Equation

Since there exists a central axis for cylindrical surfaces its direction \( C_a \) is determined during the classification process. Any point along the central axis can be considered as the origin of the coordinate system for the normalized surface equation. As mentioned earlier, a touch point is chosen to be the origin for the sensed surface equation. Given that touch point, a plane containing that point such that its normal is in the direction of the central axis can be defined as shown in Figure (6.4).

\[ P_t : \text{The touch point and the origin for the sensed surface equation coordinate frame} \]

\[ P_i : \text{Image surface points} \]

\[ N_t : \text{touch surface normal} \]

\[ O : \text{Origin of the normalized surface equation (surface centroid)} \]

Figure 6.4 - Geometry for calculating a cylindrical surface equation
At least five other surface points, \(P_i; i = 1, 2, 3, \ldots\), must be known as belonging to the surface for its equation to be calculated. If the coordinate frame is rotated to have its \(z\) axis along the surface normal \(N_t\) and the \(y\) axis along the central axis \(C_a\), the coefficients of the terms that contain \(y\) (i.e. \(y^2, xy, yz\) and \(y\)) will be eliminated. The sensed surface equation is represented by Equation (6.11) while the \(Q\) matrix of the cylindrical surface equation is given by Equation (6.12)

\[
\begin{pmatrix}
  z_2^2 & x_2 z_2 & x_2 & z_2 & 1 \\
  z_3^2 & x_3 z_3 & x_3 & z_3 & 1 \\
  z_4^2 & x_4 z_4 & x_4 & z_4 & 1 \\
  z_5^2 & x_5 z_5 & x_5 & z_5 & 1 \\
  z_6^2 & x_6 z_6 & x_6 & z_6 & 1
\end{pmatrix}
\begin{pmatrix}
  q_2 \\
  q_3 \\
  q_4 \\
  q_5 \\
  q_6
\end{pmatrix}
=
\begin{pmatrix}
  -x_2^2 \\
  -x_3^2 \\
  -x_4^2 \\
  -x_5^2 \\
  -x_6^2
\end{pmatrix}
\quad (6.11)
\]

\[
Q = \begin{pmatrix}
  1 & 0 & q_{5/2} & q_{7/2} \\
  0 & 0 & 0 & 0 \\
  q_{5/2} & 0 & q_3 & q_{9/2} \\
  q_{7/2} & 0 & q_{9/2} & q_{10}
\end{pmatrix}
\quad (6.12)
\]

6.4.3 Estimation of Spherical Surface Equation

The centroid for this type surface is not a surface point and cannot coincide with the origin of the calculated surface equation coordinate frame. Therefore, the coefficients of the first order terms: \(xy, xz, yz\) and
the constant term can be much larger than the other terms producing a high degree of inaccuracy in calculating a surface equation. To make all the coefficients approximately of the same magnitude, the sensed surface equation is calculated in a coordinate frame with its origin at the center of the VTP, a touch point. The coordinate axes are then rotated such that the z axis is oriented along the surface normal direction. Since the origin is a surface point, \( q_{10} \) becomes zero while the coefficients of \( x^2 \) and \( y^2 \) are valid because the surface normal is not in the \( x \) or \( y \) directions. One of these coefficients can be normalized to one. In general, the coefficient of \( x^2 \) is normalized. Eight points are then required for the calculation. The coefficients of the cylindrical surfaces can be obtained by solving Equation (6.13).

\[
\begin{pmatrix}
  y_2^2 & z_2^2 & x_2 y_2 & x_2 z_2 & y_2 z_2 & x_2 & y_2 & z_2 \\
  y_3^2 & z_3^2 & x_3 y_3 & x_3 z_3 & y_3 z_3 & x_3 & y_3 & z_3 \\
  y_4^2 & z_4^2 & x_4 y_4 & x_4 z_4 & y_4 z_4 & x_4 & y_4 & z_4 \\
  y_5^2 & z_5^2 & x_5 y_5 & x_5 z_5 & y_5 z_5 & x_5 & y_5 & z_5 \\
  y_6^2 & z_6^2 & x_6 y_6 & x_6 z_6 & y_6 z_6 & x_6 & y_6 & z_6 \\
  y_7^2 & z_7^2 & x_7 y_7 & x_7 z_7 & y_7 z_7 & x_7 & y_7 & z_7 \\
  y_8^2 & z_8^2 & x_8 y_8 & x_8 z_8 & y_8 z_8 & x_8 & y_8 & z_8 \\
  y_9^2 & z_9^2 & x_9 y_9 & x_9 z_9 & y_9 z_9 & x_9 & y_9 & z_9
\end{pmatrix}
\begin{pmatrix}
  q_2 \\
  q_3 \\
  q_4 \\
  q_5 \\
  q_6 \\
  q_7 \\
  q_8 \\
  q_9
\end{pmatrix}
= \begin{pmatrix}
  x_2^2 \\
  x_3^2 \\
  x_4^2 \\
  x_5^2 \\
  x_6^2 \\
  x_7^2 \\
  x_8^2 \\
  x_9^2
\end{pmatrix}
\] (6.13)

Since the surface points are co-planar the inverse of the first matrix exists and the coefficients can be evaluated.
6.5 Estimation of the Normalized Surface Equation

In the case of cylindrical and spherical surfaces the sensed surface equation is calculated in a coordinate frame with its origin located at a surface point. It is necessary to rotate the axes to be in the proper direction then translate the origin of this system to the surface centroid or to a point on the central axis depending on the type of the surface. The normalized surface equation can be obtained by solving Equation (6.14) for $Q_N$

$$Q_N = T R Q R^T T^{-1} \quad (6.14)$$

where $T$ is the translation matrix.

In the case of spherical surfaces $R$ is the rotation matrix which rotates the $z$ axis to coincide with the surface normal, $N_t$, at the surface point $N_t$. If the surface is cylindrical, $R$ is the rotation matrix used to rotate the $y$ axis to coincide with the direction of the central axis $C_a$. For planar surfaces, the rotation aligns the surface normal with the $y$ axis. $R^T$ is the transpose of $R$ and $T^{-1}$ is the inverse of $T$.

Let the upper-left $3 \times 3$ submatrix of $Q$ defined in Equation (6.6) be denoted as $Q_{ul}$ and the vector $U$ be $(q_7, q_8, q_9)^T$. The surface orientation can be determined from the eigenvectors of $Q_{ul}$. Let $v_1$, $v_2$, and $v_3$ be the eigenvectors of $Q_{ul}$ and $\mu_1$, $\mu_2$ and $\mu_3$ be the corresponding eigenvalues. The rotational matrix $R$ is equal to $[v_1, v_2, v_3]$ and the three rotational angles are calculated. One angle about the $x$ axis and another angle
about the z axis are used to specify a new rotational axis. The third angle is used to specify the angle of rotation about the new rotational axis [55].

In the case of spherical surfaces all eigenvalues are equal and the rotation angles can take any value when rotating the sensed surface equation to the normalized surface equation. In the case of cylindrical surfaces two eigenvalues will be the same and the third angle defines the rotation about the new rotation axis. Again the value of this angle will not affect the rotation from the sensed surface equation to the normalized one since the other two angles are the same. If the surface is a plane the plane normal can be used as a rotational axis and the angle to rotate about the plane normal gives the same plane and is set to undefined.

The matrix $T$ defines the translation of the sensed surface equation to the centroid of a spherical surface or to the central axis of a cylindrical surface. For spherical surfaces the centroid can be calculated by Equation (6.15)

$$ T = U^T R \begin{pmatrix} -1/\mu_1 & 0 & 0 \\ 0 & -1/\mu_2 & 0 \\ 0 & 0 & -1/\mu_2 \end{pmatrix} R^T (6.15) $$

In the case of cylindrical surfaces, the surface centroid is any point on the central axis. This point is chosen to be on a plane whose normal is in the direction of the central axis and contains the origin of the coordinate frame used in calculating the sensed surface equation.
The point of intersection of this plane with the central axis defines the surface centroid. This point can be calculated from Equation (6.16)

\[ C_{ss} = U^T R \begin{pmatrix} -1/\mu_1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -1/\mu_2 \end{pmatrix} = (C_x, 0, C_z) \quad (6.16) \]

The middle term of the explicit matrix \((C_x, 0, C_z)\) in Equation (6.16) is set equal to zero because the length of the central axis (distance along the y axis) is infinite. The translation of the sensed surface equation to the surface centroid, a point on the central axis, can be calculated from \((C_x, 0, C_z)R^T\).

### 6.6 Summary

The integration of vision and touch is the cornerstone of the recognition process. This method generates surface description from vision and touch data. These surface description is in form of vision-touch surface patches, VTP, that contains correlated vision and touch surface points. The VTPs are then classified to planar, cylindrical or spherical. An equation is calculated for each VTP using a touch surface point and a few vision surface points using an efficient method based on least square minimization technique. These calculated surface equations are used to estimate normalized surface equations to match the form of the equations stored in the model database for faster and accurate recognition. Next chapter describes the methods used in
matching a model object to a sensed object based on the extracted surface descriptions described here.

As mentioned in Chapter 5, there exists a touch point for each corresponding patch. Since the touch data is collected using an accurate touch system, CMM machine, and the vision surface points are obtained from processing a lazer range sensor data (less accurate), the integration process provides a set of vision points correlated to a touch point. Few surface points from the vision-touch surface patch that include at least one touch point are used as basis for surface classification and surface equation calculations. If vision data alone had been processed for surface extraction, surface segmentation, edge detection and refining to the surface equation calculations must be done which are computationally costly. If touch data alone had been used, an intelligent feedback control scheme would be needed to determine the coordinates of points to be touched.
CHAPTER 7

MATCHING

7.1 Introduction

Integration of vision and touch provides a description of object surfaces in form of vision-touch surface patches, surface types, and normalized surface equations that will be used by the matching routines to determine what the object is and, simultaneously, to determine its location.

The object recognition system presented here has no way of knowing what surfaces will be viewed from a particular view point or touched at a specific point. Both viewed objects and data base models are described by a surface adjacency graph of the entire object. The surface adjacency graph contains geometrical structural information between the object surfaces and some of the surface characteristics. However, the sensed object descriptions only present surface characteristics and are not yet related to the models. The matching routines try to find an object in the model data base that is consistent with the surface characteristics discovered by the sensors.

If an object matches a model, each viewed surface should
correspond to a model surface with the same normalized surface equation. That these two surfaces match only ensures that a portion of the object has characteristics similar to a portion of the model. An incorrect match can happen if there is more than one model surface having the same normalized surface equation. Moreover, it is possible that more than one model contains surfaces with the same surface normalized equation. Therefore, it is complicated to search for a correct matched surface pair if the matched model is not known. To secure a correct matching, the object centroid is constrained with respect to each surface to certain orientations and positions. If all matched surface pairs are correct, there must be an object location which satisfies the constraint, referred to as the discriminant, on the object centroid.

The object location is specified relative to the robot manipulator world coordinate frame. The models are defined in a known coordinate frame referred to as the model coordinate frame which provides a fixed location for the objects. Such locations are easily transformed to the robot world coordinate frame to represent the real world location. The object location is specified by the location of the object centroid with respect to a touch point. The location of this point is directly determined from the touch data. Therefore, the problem is reduced to a matrix transformation. The object orientation is determined from the rotation used in rotating the model object to match the sensed object. These angles of rotations are valid for any vector in the model and used to represent the object orientation.
This chapter describes the matching process, referred to as the matcher, and the model discrimination.

7.2 Design of the Matcher

The matcher is a set of modules written in C, residing in a VAX11/780 computer running Unix. To execute the modules, data files that contain vision-touch surface patches are transferred and placed in the directory where the model database exists. Another data file containing all touch points is also transferred to the same directory. The block diagram of the matcher is shown in Figure (7.1).

Referring to the block diagram, the block labeled initial matching uses the sensed surface normal equation to match all the models that have surfaces with the same normalized equation and generates a set of possible matched models. Rejection of incorrectly matched models is done in the next series of tasks. The task labeled surface orientation determination calculates the orientation of a sensed surface relative to its matched surface, then rotates the model surface to have the same sensed surface orientation. Since both sensed and model surfaces have the same orientation, the object centroid is mapped from the model object to the sensed object using the same transformation done previously. This is done in the task block labeled Object Centroid Localization. The following task, Discrimination, checks if the transformed model object centroid satisfies the discriminant. If this
condition is true, the adjacency relation between the object surfaces is checked in the verification task block. Once a correct match is achieved the object location is calculated in the last task block in the matcher using the location of the touch data points.

7.3 Initial Matching

This step uses the normalized surface equation provided during the integration of sensed object data to generate an initial set of models that could possibly match the object. The selection of these matched models is based on the premise that: Given one normalized surface equation, the initial set of matched models is the set of models in which each model contains at least one surface with the same normalized surface equation. When more than one sensed surface is given, a matched model should contain all sensed surfaces.

The matching process is implemented by constructing a model surface table that has three fields for each entry: A surface index, a normalized surface equation, and a model list. Each model in this list contains at least one surface with the normalized surface equation given in the second field. An example of such a list is shown in Figure (7.2)
Figure 7.1 - The matcher block diagram
Searching by index for an entry having a normalized surface equation similar to the equation of the sensed surface, all models in the model list for that index are considered initial matched models. This initial set of models is reduced by intersecting other surfaces in the model list with model lists corresponding to other sensed surface equations. This new set of models is called the reduced model set.

<table>
<thead>
<tr>
<th>Index (number)</th>
<th>Normalized Surface Equation (equation)</th>
</tr>
</thead>
<tbody>
<tr>
<td>List of models</td>
<td></td>
</tr>
<tr>
<td>Model 1</td>
<td>(name)</td>
</tr>
<tr>
<td>Model 2</td>
<td>(name)</td>
</tr>
<tr>
<td>...</td>
<td></td>
</tr>
<tr>
<td>Model n</td>
<td>(name)</td>
</tr>
</tbody>
</table>

Figure 7.2 - An index entry for generating the initial set of matched models

Given a reduced model set, each model surface with the same normalized equation as the sensed one is used to form a matched surface pair. A matched surface pair is defined as a set of two surfaces, a sensed surface and a model surface, having the same normalized equation. These matched surface pairs may be incorrect for two reasons. First, the description of an object can contain multiple
surfaces with the same normalized equation but connected to different surfaces. Second, the matched surface pair may belong to two different objects. Therefore, a method is needed to refine the initial matched list of models. This is done by rejecting an incorrect match. The rejection of an incorrect match is achieved according to the following sequence: Calculate the surface orientation, search for a possible object centroid, impose the discriminant constraint and maintain the surface adjacency relations. Each of these steps is discussed in the following sections.

7.4 Sensed Surface Orientation

The sensed surface orientation is defined to be the orientation of the sensed surface relative to the matched model surface. Using this definition, a rotational matrix \( R = (\alpha, \beta, \gamma) \) can be constructed. By rotating the model surface with this rotation matrix the model surface and the sensed surface will have the same orientation. This section describes methods to calculate this rotation matrix for each surface type.

Let the orientation of the model surface be described by \( (\alpha_m, \beta_m, \gamma_m) \). The origin of the coordinate frame is at the model surface centroid. Its axes are predefined and remain the same for all object models. This orientation defines three rotational angles about the coordinate axes in the following sequence. First, rotation \( (\alpha_m) \) about the x axis. Second, rotation \( (\gamma_m) \) about the z axis of the original coordinate frame. Third, rotation \( (\beta_m) \) about the rotated y axis. Applying this rotation will align
the coordinate axes with the surface principle axes.

The relative surface orientation specifies a rotation angle $\alpha$ about the X axis and a rotation angle $\gamma$ about the Z axis which will rotate the Y axis to coincide with a rotation vector $r$. A rotation angle $\beta$ about $r$ is derived to align the X and Z axes with the corresponding surface principle axes. The direction of the rotational vector $r$ is different for each surface type. For spherical surfaces it can be in any direction. An arbitrary surface normal direction is chosen. Since the sphere has three equal principal axes, its rotation will not change the normalized surface equation. For cylindrical surfaces, the rotational vector $r$ is in the direction of the central axis of the cylinder while for planar surfaces it is in the surface normal direction.

Assume the coordinate axes of the model surface centered coordinate frame are X, Y, and Z, and the coordinate axes of the sensed surface are $X_s$, $Y_s$ and $Z_s$ as shown in Figure (7.3).

![Diagram of coordinate frames](image)

Figure 7.3 - The model object and sensed coordinate frames
To rotate $Y$ to $Y_s$, first rotate about the $X$ axis by $\alpha$ such that $Y$ is rotated to a vector $Y'$ maintaining $(Y' \cdot Z)$ equals $(Y_s \cdot Z)$. $\alpha$ is then equal to $\arcsin (Y_s \cdot Z)$. Since both vectors are known, $\alpha$ can be calculated. The axis $Y_s$ is the same as the rotation vector $r$. The rotational angle $\gamma$ about the $Z$ axis can be calculated from the projections of $r$ and $Y'$ onto the $XY$ plane. The projection of $Y'$ is $Y$ and the projection of $r$, denoted by $r_p$, is $[0, r_y, r_z]$. Then the rotation angle $\gamma$ about $Z$ equals to $\arccos (r_p \cdot Y)$. The geometry of these two angles is shown in Figure (7.4).

Figure 7.4 - The geometry of the rotational angles $\alpha$ and $\gamma$ about $X$ and $Z$ axes

The third rotational angle $\beta$ about $r$ is determined as follows: The description of the surface in the model data base defines a vector $V$. Its corresponding vector $V_s$ in the sensed surface is known. If the projections of the vectors $V_s$ and $V_s R_x(\alpha) R_z(\gamma)$ onto the $XZ$ plane are p
and $p_r$, the rotation angle $\beta$ about $r$ then equals $\arccos(p \cdot ps)$ similar to the calculations of $\gamma$ as shown in Figure (7.5).

![Diagram showing the geometry of the rotational angle $\beta$](image)

**Figure 7.5 - The geometry of the rotational angle $\beta$**

Given the relative surface orientation ($\alpha$, $\beta$, $\gamma$) a rotation matrix to rotate any vector in the model to its corresponding vector in the sensed coordinate is given by Equation (7.1)

$$R = R_Y(-\beta_m) R_z(-\gamma_m) R_x(-\alpha_m) \quad R_x(\alpha) R_z(\gamma) R_r(\beta)$$

$$R_x(\alpha_m) R_z(\gamma_m) R_Y(\beta_m). \quad (7.1)$$

The first three rotations are to rotate the vector from the model object coordinate frame to the model surface coordinate frame. The following three rotations are to rotate the vector to the sensed surface coordinate frame and the last three rotations rotate the vector back to the model object coordinate frame.
7.5 Sensed Object Centroid and the discriminant

For each object, there exists one position for the object centroid. Therefore, given a matched surface pair, calculated in the first step of the matching process, and a rotation matrix, calculated in the previous section, a possible location for the object centroid can be developed. This location is verified by checking the Centroid Geometric Primitive, CGP, stored in the model database. This constraint is referred to as the discriminant.

The discriminant can be calculated from the matched surface pair type and the rotation matrix $R$. If the matched surface pair is spherical, the discriminant specifies the distance from the surface centroid to the object centroid. If the two surfaces are cylindrical, the discriminant specifies the perpendicular distance from the cylinder central axis and the object centroid. If the matched surface pair is planar, the surface centroid is a point on the plane and the object centroid is a point on a constant distance away from that plan. This specifies the discriminant to be a constant distance between a point and a plane. The discriminant is calculated by methods described in the following subsections followed by the methods used in determining the object centroid.
7.5.1 Matching Spherical Surfaces

In this case a vector from the model surface centroid $P_m$ to the model object Centroid $P_o$ is specified in the model. This vector is rotated by the rotation matrix $R$ used in rotating this model surface to the matched sensed object surface. The discriminant is equal to the magnitude of this vector. If the discriminant matches the corresponding distance calculated using the sensed surface centroid and the rotated model centroid, the surface is correctly matched and the model object could match the sensed object. Figure (7.6) illustrates this case.

(a) Model Surface

$P_m$ : model surface centroid

$P_o$ : model object centroid

$v = (P_m - P_o)$

$d = |P_m - P_o|$ the discriminant

(b) Sensed Surface

$P_s$ : Sensed surface centroid

$P_{sc}$ : $P_o$ after rotation

$d_s$ : calculated distance

if $d_s = d$ -> correct match
7.5.2 Matching of Cylindrical Surfaces

Let the model object centroid be at \( C_{mo} \) and the model surface centroid be at \( C_{ms} \), a point on the cylinder central axis. The orientation matrix \( R \) of the sensed surface can be calculated by methods described in the previous chapter. The vector from \( C_{ms} \) to \( C_{mo} \) (\( C_{mo} - C_{ms} \)) is rotated by \( R \), the corresponding vector "\( V_r \)" can be calculated from \( V_r = (C_{mo} - C_{ms}) R \). The discriminant in this case is the perpendicular distance between the surface central axis and the CGP. This distance can be calculated by subtracting \( (C_{mo} - C_{ms}) R \) from \( C_{ms} \) where \( C_{ms} \) is the model surface centroid. This case is described in Figure (7.7)

![Diagram](image)

(a) Model Surface  
(b) Sensed Surface

- \( C_{ms} \): model surface centroid  
- \( C_{mo} \): model object centroid  
- \( C_{a} \): central axis  
- \( d \): the discriminant  
- \( Y_s \): Sensed surface central axis  
- \( C_{ss} \): Sensed surface centroid  
- \( v_r = v \) after rotation  
- if \( d_s = d \) -> correct match

Figure 7.7 Matching of cylindrical surfaces
7.5.3 Matching of Planar Surfaces

In this case, the rotation is about a plane surface normal and the object centroid is at a constant distance, \( d \), from the surface and located on the plane normal. This distance is a derivable invariant from the object model and is used as the discriminant. It specifies a plane, CGP, parallel to the surface and containing the object centroid. Figure (7.8) shows the CGP and the discriminant for planar surfaces.

The discriminant is calculated from \((C_{mo} - C_{ms}) \cdot N\) where

- \(C_{mo}\) is the model object centroid
- \(C_{ms}\) is the model surface centroid

\(\cdot\) indicates a dot product

and \(N\) is the surface normal.

![Figure 7.8 - The centroid geometric primitive and the discriminant for a planar surface](image-url)
7.6 Detecting Incorrect Matching

The matched object models generated by the discriminant may be incorrect because the description of an object can contain multiple surfaces with the same normalized equation but connecting to a different types of surfaces. Figure (7.9) shows an example for this case. The two objects are detected as matched objects because of the matched surface pairs \([S_{s1}, S_{m1}], [S_{s2}, S_{m2}]\) and \([S_{s3}, S_{m3}]\). This object match is incorrect because the adjacency relations are not satisfied.

![Figure 7.9 - An incorrect matching after the discriminant](image)

Therefore, the surface adjacency relations for a matched surface pair are checked to detect incorrect matching. Given two sensed surfaces and two model surfaces, the consistency of the surface adjacency relations is defined as: Two sensed surfaces are adjacent if and only if their corresponding model surfaces are adjacent.
7.7 Determination of Object Location

The object location is defined as the location of the object centroid in the robot world coordinate frame. In order to calculate this location, the relative location of the object centroid with respect to any surface should be determined first. The discriminant defines the relation between a specific point defined by the surface type [a centroid for spherical surfaces, a point on the central axis for cylindrical surfaces, a point in a plane for planar surfaces] and the object centroid. During the processing of the integrated data, a transformation from a touch point on the surface [the origin for the sensed surface equation coordinate frame] to the same specific point is calculated. The location of the touch point is determined in the robot coordinate frame while the robot is in contact with the surface point. Given the above mentioned information, the problem of locating the object reduced to a simple coordinate translation. The sequence of this translations is as follows: From the object centroid to the surface specific point, and from that point to the touch point on the same surface. Figure (7.10) is a graphical representation of these transformations.
7.8 Summary

Matching is the last step of the recognition process. It is accomplished by finding possible matched models based upon surface characteristics. These characteristics are the normalized surface equations which are stored in the model data base and also calculated after the integration procedure using few touch and vision surface points. Incorrect matched models are rejected by imposing a constraint on the relative location of the object centroid to each surface. This constraint was referred to as the "discriminant". After the object had been recognized its location in the robot world coordinate frame was determined. Since locations of surface points in the robot coordinate
frame are obtained from the touch system, the problem is reduced to finding the location of the object centroid relative to any touch point. This relative location was calculated during the recognition process using the discriminant. Simple coordinate transformation are then applied to calculate the location of the object centroid in the robot world coordinate frame.
CHAPTER 8

EXPERIMENTAL RESULTS

8.1 Introduction

This chapter details the experiments that were conducted to test the approach described in the previous chapters. The results of integrating vision and touch are presented as is the ability of correct matches to be made against the model data base. The experiments are intended to show working approaches to object recognition and localization. They show that integrating vision and touch is a valuable method for recognition. The experiments were conducted with real touch data with sensor noise, and synthesized (with added noise) range vision data. The tactile sensor being used is relatively crude in terms of spatial resolution compared to modern measurement robots, i.e., coordinate measuring machines. The main intent of these experiments is to show the utility of the methods presented in this dissertation and to show the ability of active touch and vision to succeed in situations that vision alone would find difficult such as smoothly connected surfaces or obscured surfaces.
8.1 Experiments for Surface Normal Extraction

This set of experiments is presented to show the ability of the operators and procedures developed in chapters three and four to extract surface normals. The vision and touch procedures were tested individually using the same type surfaces. Three surface types were used: planar, cylindrical and spherical. Synthetic surface vision range images for each surface were generated and used. Gaussian noise with a standard deviation error of 1 mm was added to each range image (the nominal range for a range image is about one meter). This noise is added randomly in the direction of viewing points, q.v. the camera and not necessary perpendicular to the sensed surface. This noise model is appropriate for the errors due to timing jitter in the range measurement electronics (triangulation sensing would have a different noise model). Real touch points are measured using actual sensors with accompanying real noise; hence, no separate noise model is needed for tactile data. Touch data was processed by the procedures given in Chapter five.

The first experiment examined the extraction of surface normals from a wooden circular plate image. The dynamic operators developed in Chapter four were used to successfully extract these planar surface normals. Figure (8.1a) shows the calculated surface normals verifying the plate's circular appearance. It also shows parallel surface normals which is typical for planar surfaces. When the same surface is sensed by the touch system, four points (Figure (8.1b)) are reported in terms of
their three coordinates and the values of the robot angles $\theta$ and $\phi$. The touch system performed well since all the surface normals are parallel as it is verified by the nearly constant value of the robot angle $\phi$; $\phi = 90^\circ \pm 0.031$

(a) Surface normals from vision analysis

<table>
<thead>
<tr>
<th>Points</th>
<th>Coordinates</th>
<th>$\theta$</th>
<th>$\phi$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>25.00, 37.00, 8.00</td>
<td>1.571</td>
<td>89.989</td>
</tr>
<tr>
<td>2</td>
<td>31.00, 43.00, 8.00</td>
<td>2.364</td>
<td>90.006</td>
</tr>
<tr>
<td>3</td>
<td>19.00, 31.00, 8.00</td>
<td>1.863</td>
<td>90.003</td>
</tr>
<tr>
<td>4</td>
<td>37.00, 39.00, 8.00</td>
<td>1.635</td>
<td>89.995</td>
</tr>
</tbody>
</table>

(b) Touch data for surface normal calculations

Figure 8.1 - Surface normals on a planar surface
The second experiment was conducted to test the surface normal extraction procedures for cylindrical surfaces. A synthetic surface range image for a cylinder 6 cm in diameter was generated and used. Gaussian noise with standard deviation error of 1 mm was added to the image. Extracted surface normals from the range vision processing are shown in Figure (8.2a). The vision operators proved that they give a good estimation of surface normals from cylindrical surfaces. The touch data from four surface points at different heights is presented in Figure (8.2b). They were reported in the same format as used for planar surface for comparison purposes. In this case, the robot angle $\phi$ is approximately zero because the cylinder was set upright and the touch plane is parallel to the XY plane of the robot coordinate system.

The third experiment used a synthetic range image for a sphere with a radius of 5 cm with the same added Gaussian noise. A cross section of the surface showing extracted surface normals is shown in Figure (8.3a). The results show that a spherical surface can be fit to that cross section. This proves that the surface normals were estimated correctly. Four touch points were selected to examine the behavior of the touch system at different approach angles. According, the touch points were selected at different planes and at different heights. The results of the touch system data analysis is recorded in Figure (8.3b).
(a) Surface normals from vision analysis

<table>
<thead>
<tr>
<th>Points</th>
<th>Coordinates</th>
<th>θ</th>
<th>φ</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>22.82, 31.02, 13.46</td>
<td>20.3</td>
<td>0.006</td>
</tr>
<tr>
<td>2</td>
<td>21.63, 32.52, 28.42</td>
<td>32.8</td>
<td>0.003</td>
</tr>
<tr>
<td>3</td>
<td>19.53, 32.96, 23.14</td>
<td>125.6</td>
<td>179.998</td>
</tr>
<tr>
<td>4</td>
<td>17.58, 31.78, 15.38</td>
<td>85.2</td>
<td>179.995</td>
</tr>
</tbody>
</table>

(b) Touch data for surface normals calculations

Figure 8.2 - Surface normals on a cylindrical surface
The recognition of a spherical surface was verified by examining the values of the robot angles $\theta$ and $\phi$. The angle $\theta$ widely varied because a spherical surface occupies all possible geometrical planes. The angle $\phi$ is neither constant as in the case of planar surfaces, nor equals to zero as in the case of cylindrical surfaces.

The above discussion leads to the conclusion that the surface normal estimation procedures developed in this dissertation have performed well in both estimating surface normals and in distinguishing between the three types of surfaces suggested in this research.

![Cross section showing surface normals from vision analysis](image)

<table>
<thead>
<tr>
<th>Points</th>
<th>Coordinates</th>
<th>$\theta$</th>
<th>$\phi$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>26.35, 38.38, 8.43</td>
<td>14.571</td>
<td>23.56</td>
</tr>
<tr>
<td>2</td>
<td>25.86, 37.47, 9.26</td>
<td>0.68</td>
<td>89.53</td>
</tr>
<tr>
<td>3</td>
<td>23.53, 39.51, 6.57</td>
<td>136.24</td>
<td>54.68</td>
</tr>
<tr>
<td>4</td>
<td>22.65, 39.36, 5.68</td>
<td>38.62</td>
<td>64.28</td>
</tr>
</tbody>
</table>

(b) Touch data for surface normal calculations

Figure 8.3 - Surface normals on a spherical surface
8.3 Experiments for Vision-Touch Surface Patches

A number of experiments have been done to test the ability of the described vision-touch integration technique to recognize a single surface type using 64x64 synthetic surface range images with added Gaussian noise (with a standard deviation error of 1 mm) and touch data acquired from the same surfaces. The threshold used in these experiments was experimentally determined by repeating the experiment several times and selecting the value of the threshold that gave optimal results for all cases. Touch points located at different places were selected for each surface type and used as a seed to grow a vision-touch patch.

The vision-touch integration algorithm performed well in all three cases. In all three cases, few points within the classified vision touch patches were not correctly identified as belonging to the patch. This is due to the added Gaussian noise. Since our goal is to use these patches as the starting points for a model based object recognition system, these errors are tolerable. All the points were correctly identified with no added noise.

Figure (8.4), shows a synthesized range image of a planar surface 5 cm in length and 3 cm in width. Figure (8.5) shows the vision-touch surface patch using a touch point at (32, 32). The touch point is chosen at the middle of the surface since this surface type is the easiest to recognize. There are several other touch points shown in Figure (8.4)
that could have been used for growing a vision-touch patch. Only one point was chosen for demonstration purposes.

The experiment is repeated for a cylindrical surface with radius 3 cm and a touch point at (32, 13). The touch point is randomly selected in this case. The image and the cylindrical vision-touch surface patch are shown in Figures (8.6) and (8.8).

An image for a sphere with a radius of 5 cm (with the added Gaussian noise) was synthesized and is shown in figure (8.9). The data from vision and touch are integrated using the procedures developed in chapter 5. The vision-touch spherical surface patch generated from these data integration procedures is shown in Figure (8.10). The touch point position used for this surface was chosen at (35, 40).
Figure 8.4 - Image of a plane with added Gaussian error with standard deviation of 1mm

Figure 8.5 - Planar vision - touch patch (within borders) using touch point at (32, 32)
T = touch points  S = vision surface points  . = image pixels
Figure 8.6 - Image of a cylinder with added Gaussian error with standard deviation of 1mm; the superimposed lines are contours of constant range.

Figure 8.7 - Cylindrical vision - touch patch (within borders) using touch point at (32, 13)

T = touch points  S = vision surface points  · = image pixels
Figure 8.3 - Image of a sphere with added Gaussian error with standard deviation of 1mm; the superimposed lines are contours of constant range

Figure 8.9 - Spherical vision: touch patch (within borders) using touch point at (35, 40)
T = touch points  S = vision surface points  = image pixels
8.4 Experiments for Object Recognition and Localization

The experiments described in the previous section were for a single surface. The experiments described in this section were performed on range vision images containing more than one surface to show the performance of the overall system. Two test objects were used.

The first object consists of a circular cylinder with radius equal to 4 cm and height equal to 12 cm. In the middle of this cylinder, there is a sphere of radius 5 cm. At each end of the cylinder there is a circular plane. The object configuration is shown in Figure (8.10a), its image is in Figure (8.10b), and its model description is in Figure (8.11). The result of the matching process and the object recognition and localization algorithm described in chapter seven is presented in Figures (8.12) and (8.13). The object is correctly recognized. The real location of the object is at (20.000, 30.000, 6.000) while the recognized object location is at (19.995, 30.002, 5.999). The absolute distance between the two locations is 0.003 cm, 0.085% of the total object length, which is very small and negligible for most applications.
(a) Object configuration

(b) Object range image

(the superimposed lines are contours of constant range)

Figure 8.10 - Test object 1 "cylinder & sphere"
/** Object information **/ 

object id: 2 
object name: Cylinder & Sphere 

/** Surface Information **/ 

surface id: 4 
surface class: PLANAR 

normalized equation: $Q_n = \begin{pmatrix} 0.000 & 0.000 & 0.000 & 0.000 \\ 0.000 & 0.000 & 0.000 & 0.500 \\ 0.000 & 0.000 & 0.000 & 0.000 \\ 0.000 & 0.500 & 0.000 & 0.000 \end{pmatrix}$ 

orientation: ( 0.000 , 0.000 , 0.000 ) 
position : type ( 0 ) 
   point ( 0.000 , 0.000 , 0.000 ) 
   vector ( 0.000 , 1.000 , 0.000 ) 
object Centroid : point ( 0.000 , 6.000 , 0.000 ) 
   vector ( 0.000 , -1.000 , 0.000 ) 
adjacency relation ( 4 , 5 ) 

surface id: 5 
surface class: CYLINDRICAL 

normalized equation: $Q_n = \begin{pmatrix} 1.000 & 0.000 & 0.000 & 0.000 \\ 0.000 & 0.000 & 0.000 & 0.000 \\ 0.000 & 0.000 & 1.000 & 0.000 \\ 0.000 & 0.000 & 0.000 & -16.000 \end{pmatrix}$ 

orientation: ( 0.000 , 0.000 , 0.000 ) 
position : type ( 1 ) 
   point ( 0.000 , 0.000 , 0.000 ) 
   vector ( 0.000 , 1.000 , 0.000 ) 
object Centroid : point ( 0.000 , 5.500 , 0.000 ) 
   vector ( 0.000 , -1.000 , 0.000 ) 
adjacency relation ( 4 , 5 ) ; ( 5 , 6 ) 

Continues into next page
surface id: 6
surface class: SPHERICAL

normalized equation: \[ Q_n = \begin{pmatrix} 1.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 1.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 1.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & -5.00 \end{pmatrix} \]

orientation: (-1.571, 0.000, -1.571)

position: type (2)
- point (0.000, 0.000, 0.000)
- vector (0.000, 0.000, 0.000)

object centroid: point (0.000, 0.000, 0.000)
- vector (0.000, 0.000, 0.000)

adjacency relation: (5, 6); (6, 7)

surface id: 7
surface class: CYLINDRICAL

normalized equation: \[ Q_n = \begin{pmatrix} 1.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 1.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & -16.000 \end{pmatrix} \]

orientation: (0.000, 0.000, 0.000)

position: type (1)
- point (0.000, 0.000, 0.000)
- vector (0.000, 1.000, 0.000)

object centroid: point (0.000, 5.500, 0.000)
- vector (0.000, 1.000, 0.000)

adjacency relation: (6, 7); (7, 8)

Continues into next page
surface id: 8
surface class: PLANAR

normalized equation: \( Q_n = \begin{pmatrix} 0.000 & 0.000 & 0.000 & 0.000 \\ 0.000 & 0.000 & 0.000 & 0.500 \\ 0.000 & 0.000 & 0.000 & 0.000 \\ 0.000 & 0.500 & 0.000 & 0.000 \end{pmatrix} \)

orientation: ( 0.000 , 0.000 , 0.000 )

position : type ( 0 )
  point ( 0.000 , 0.000 , 0.000 )
  vector ( 0.000 , 1.000 , 0.000 )

object centroid : point ( 0.000 , 6.000 , 0.000 )
  vector ( 0.000 , 1.000 , 0.000 )

adjacency relation ( 4 , 5 )

Figure 8.11 - Model description for test object 1
"cylinder & sphere"
<table>
<thead>
<tr>
<th>Surface ID: 1</th>
<th>Surface Class: cylindrical</th>
</tr>
</thead>
<tbody>
<tr>
<td>Normalized Equation: $Q_n = \begin{pmatrix} 1.000 &amp; 0.000 &amp; 0.000 &amp; 0.000 \ 0.000 &amp; 0.003 &amp; 0.000 &amp; 0.000 \ 0.000 &amp; 0.000 &amp; 0.989 &amp; 0.000 \ 0.000 &amp; 0.000 &amp; 0.000 &amp; -16.012 \end{pmatrix}$</td>
<td></td>
</tr>
<tr>
<td>Orientation: (-1.572, 0.000, 0.000)</td>
<td></td>
</tr>
<tr>
<td>Position: type (1)</td>
<td></td>
</tr>
<tr>
<td>point (19.984, 32.002, 5.996)</td>
<td></td>
</tr>
<tr>
<td>vector (0.000, 1.000, 0.000)</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Surface ID: 2</th>
<th>Surface Class: spherical</th>
</tr>
</thead>
<tbody>
<tr>
<td>Normalized Equation: $Q_n = \begin{pmatrix} 1.000 &amp; 0.000 &amp; 0.000 &amp; 0.000 \ 0.000 &amp; 0.983 &amp; 0.000 &amp; 0.000 \ 0.000 &amp; 0.000 &amp; 0.988 &amp; 0.000 \ 0.000 &amp; 0.000 &amp; 0.000 &amp; -5.032 \end{pmatrix}$</td>
<td></td>
</tr>
<tr>
<td>Orientation: (0.000, 0.000, 0.000)</td>
<td></td>
</tr>
<tr>
<td>Position: type (1)</td>
<td></td>
</tr>
<tr>
<td>point (19.995, 30.002, 5.999)</td>
<td></td>
</tr>
<tr>
<td>vector (0.000, 0.000, 0.000)</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Surface ID: 3</th>
<th>Surface Class: cylindrical</th>
</tr>
</thead>
<tbody>
<tr>
<td>Normalized Equation: $Q_n = \begin{pmatrix} 1.000 &amp; 0.000 &amp; 0.000 &amp; 0.000 \ 0.000 &amp; 0.000 &amp; 0.000 &amp; 0.000 \ 0.000 &amp; 0.000 &amp; 0.987 &amp; 0.000 \ 0.000 &amp; 0.000 &amp; 0.000 &amp; -16.006 \end{pmatrix}$</td>
<td></td>
</tr>
<tr>
<td>Orientation: (6.286, 0.000, -1.572)</td>
<td></td>
</tr>
<tr>
<td>Position: type (1)</td>
<td></td>
</tr>
<tr>
<td>point (19.983, 24.621, 5.988)</td>
<td></td>
</tr>
<tr>
<td>vector (0.000, -1.000, 0.000)</td>
<td></td>
</tr>
</tbody>
</table>

**Recognized Object:** Cylinder & Sphere

**Object Location:** (19.995, 30.002, 5.999)

Figure 8.12 - Object recognition and localization results for Test Object 1
The second experiment used the object labeled "Cylinder & Block" shown in Figure (8.13). The modeled object contains a block with dimensions 5 x 7 x 9 cm and is described by the format given in Figure (8.14). A synthesized range vision image containing a planar surface (8 x 9) and the cylindrical surface. Gaussian noise with standard deviation of 1mm is generated and added randomly to the image. The result of object recognition and localization is shown in Figure (8.15). The exact object centroid in the robot world coordinate frame is at (0.000, 35.000, 4.500) and the estimated centroid is at (-0.012, 34.996, 4.513). The absolute difference in distance between them is 0.018 cm for an object maximum length of 9 cm. The error in percentage is .02%. The error percentage in this case is smaller than in the first experiment since planar surfaces can be more accurately located because of the parallel nature of their surface normals. Also the smooth edge between the planar surfaces and the cylindrical surface did not impose any problems for recognition. These edges need intensive image processing techniques and complex operators if it would have been recognized using vision only [8].

![Figure 8.13 - Test object 2 "Cylinder & Block"](image_url)
/** Object information **/

object id: 3
object name: Cylinder & Block

/** Surface Information **/

surface id: 10
surface class: PLANAR
normalized equation: \( Q_n = \begin{pmatrix} 0.000 & 0.000 & 0.000 & 0.000 \\ 0.000 & 0.000 & 0.000 & 0.500 \\ 0.000 & 0.000 & 0.000 & 0.000 \\ 0.000 & 0.500 & 0.000 & 0.000 \end{pmatrix} \)
orientation: ( -1.752 , 0.000 , 0.000 )
position : type ( 0 )
    point ( 0.000 , 4.500 , 0.000 )
    vector ( 0.000 , 1.000 , 0.000 )
object Centroid : point ( 0.000 , 4.5 , 0.000 )
    vector ( 0.000 , -1.000 , 0.000 )
adjacency relation ( 10 , 11 ) ; ( 10 , 13 ) ; ( 10 , 14 )

surface id: 11
surface class: PLANAR
normalized equation: \( Q_n = \begin{pmatrix} 0.000 & 0.000 & 0.000 & 0.000 \\ 0.000 & 0.000 & 0.000 & 0.500 \\ 0.000 & 0.000 & 0.000 & 0.000 \\ 0.000 & 0.500 & 0.000 & 0.000 \end{pmatrix} \)
orientation: ( 0.000 , -1.752 , 0.000 )
position : type ( 0 )
    point ( 2.500 , 0.000 , 1.000 )
    vector ( 1.000 , 0.000 , 0.000 )
object Centroid : point ( 2.500 , 0.000 , 1.000 )
    vector ( -1.000 , 0.000 , -1.000 )
adjacency relation ( 11 , 10 ) ; ( 11 , 12 ) ;
    ( 11 , 14 ) ; ( 11 , 15 )
Continues into next page

surface id: 12
surface class: PLANAR

normalized equation: \( Q_n = \begin{pmatrix} 0.000 & 0.000 & 0.000 & 0.000 \\ 0.000 & 0.000 & 0.000 & 0.500 \\ 0.000 & 0.000 & 0.000 & 0.000 \\ 0.000 & 0.500 & 0.000 & 0.000 \end{pmatrix} \)

orientation: (1.752, 0.000, 0.000)

position: type (0)

point (0.000, -4.500, 0.000)
vector (0.000, -1.000, 0.000)

object Centroid: point (0.000, 4.5, 0.000)
vector (0.000, 1.000, 0.000)

adjacency relation (12, 11); (12, 13); (12, 14)

surface id: 13
surface class: PLANAR

normalized equation: \( Q_n = \begin{pmatrix} 0.000 & 0.000 & 0.000 & 0.000 \\ 0.000 & 0.000 & 0.000 & 0.500 \\ 0.000 & 0.000 & 0.000 & 0.000 \\ 0.000 & 0.500 & 0.000 & 0.000 \end{pmatrix} \)

orientation: (0.000, 1.752, 0.000)

position: type (0)

point (-2.500, 0.000, 1.000)
vector (-1.000, 0.000, 0.000)

object Centroid: point (2.500, 0.000, 1.000)
vector (1.000, 0.000, 1.000)

adjacency relation (13, 10); (13, 12);
(13, 14); (13, 15)

Continues into next page
surface id: 14
surface class: PLANAR

normalized equation: \( Q_n = \begin{pmatrix} 0.000 & 0.000 & 0.000 & 0.000 \\ 0.000 & 0.000 & 0.000 & 0.500 \\ 0.000 & 0.000 & 0.000 & 0.000 \\ 0.000 & 0.500 & 0.000 & 0.000 \end{pmatrix} \)

orientation: (0.000, 3.143, 0.000)
position: type (0)
  point (0.000, 0.000, 4.500)
  vector (-1.000, 0.000, -1.000)

object Centroid: point (0.000, 0.000, 4.5)
  vector (0.000, 0.000, 1.000)

adjacency relation (14, 10); (11, 11);
(11, 12); (11, 13)

surface id: 15
surface class: CYLINDRICAL

normalized equation: \( Q_n = \begin{pmatrix} 1.000 & 0.000 & 0.000 & 0.000 \\ 0.000 & 0.000 & 0.000 & 0.000 \\ 0.000 & 0.000 & 1.000 & 0.000 \\ 0.000 & 0.000 & 0.000 & -6.250 \end{pmatrix} \)

orientation: (0.000, 0.000, 0.000)
position: type (1)
  point (0.000, 0.000, 3.560)
  vector (0.000, 1.000, 0.000)

object centroid: point (0.000, 0.000, 3.560)
  vector (0.000, 0.000, -1.000)

adjacency relation (15, 11); (15, 13)

Figure 8.14 - Model description for object "cylinder & block"
<table>
<thead>
<tr>
<th>Surface id: 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Surface class: cylindrical</td>
</tr>
<tr>
<td>Normalized equation: $Q_n = \begin{pmatrix} 1.000 &amp; 0.000 &amp; 0.000 &amp; 0.000 \ 0.000 &amp; -0.001 &amp; 0.000 &amp; 0.000 \ 0.000 &amp; 0.000 &amp; 0.994 &amp; 0.000 \ 0.000 &amp; 0.000 &amp; 0.000 &amp; -6.248 \end{pmatrix}$</td>
</tr>
<tr>
<td>Orientation: (-3.560, 0.000, 0.000)</td>
</tr>
<tr>
<td>Position: type (1)</td>
</tr>
<tr>
<td>point (-0.012, 34.992, 7.847)</td>
</tr>
<tr>
<td>vector (0.004, 0.999, -0.007)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Surface id: 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Surface class: planar</td>
</tr>
<tr>
<td>Normalized equation: $Q_n = \begin{pmatrix} 0.000 &amp; 0.000 &amp; 0.000 &amp; 0.000 \ 0.000 &amp; 0.000 &amp; 0.000 &amp; 0.500 \ 0.000 &amp; 0.000 &amp; 0.000 &amp; 0.000 \ 0.000 &amp; 0.500 &amp; 0.000 &amp; 0.000 \end{pmatrix}$</td>
</tr>
<tr>
<td>Orientation: (-0.034, 1.752, 0.023)</td>
</tr>
<tr>
<td>Position: type (0)</td>
</tr>
<tr>
<td>point (2.488, 34.996, 5.500)</td>
</tr>
<tr>
<td>vector (0.023, -0.998, -0.034)</td>
</tr>
</tbody>
</table>

**Recognized Object**: Cylinder & Block

**Object Location**: (19.995, 30.002, 5.999)

8.12 - Object recognition and localization results for Test Object 2
8.5 Summary

The experiments reported in this chapter show the ability of vision and touch sensing systems to sense, recognize and locate objects. Different experiments were carried out to test the different components of the system at various levels of complexity. The degree of complexity involved the type of the object surfaces and the way that they were connected. These objects were chosen to include many different types of surfaces in the same object. The same surfaces were included in different objects to test the ability of the matcher to correctly recognize and locate the object under investigation.

The primary vision and touch data analysis to extract surface normals has performed well. Small errors in estimating surface normals were caused by the poor resolution of the touch sensors, inaccuracy in the force-torque sensor data, and added noise in the vision images. Since the data was collected for the same surface using two independent systems (vision and touch), the overall effect of these errors was tolerable and negligible.

The geometric descriptions of objects used in this dissertation provide a strong matching criteria that can lead to a unique matching strategy based upon a combination of partial surface descriptions and centroid constraints. The matcher correctly rejected different objects as shown by the correctly recognized objects in recognizing test objects one and two among five objects described in the data base.
Determination of the object location was a powerful and straightforward approach to determine location of objects in a robot coordinate frame. The combination of recognition and localization provides a strong, useful and applicable approach to apply in the real world.
CHAPTER 9

DISCUSSION AND CONCLUSIONS

9.1 Introduction

A new system for recognizing objects based upon their geometrical structure as sensed by vision and touch sensors has been presented. New methods have been developed to extract object descriptions from vision range images and touch data, to integrate these descriptions into a common model data base, and to recognize and locate objects using these descriptions and a model data base. Through point, patch, and vision-touch data integration an object can be described in terms of surfaces and their adjacency relations. New procedures based upon the statistical variation of the surface normals have been developed to generate surface patches and segment noisy vision range images. New algorithms have been developed to analyze touch data obtained from a pair of tactile and force-torque sensors to extract surface normals by the aid of active touch. A data structure based upon geometric primitives was developed to integrate the vision and touch information into surface primitives that are used in a matching phase.
The surface primitives used in this research are vision-touch surface patches generated from surface normals (extracted from vision and touch) and contain a calculated surface equation. Surface equations were calculated with an efficient method that needs only few touch and vision surface points. This method has the ability to extract the description of new and unknown objects as no a priori knowledge about the objects is required prior to the extraction process.

The paradigm of model based recognition was used, where the models are arranged in such ways to ease the matched model selection through the processing. The models also include spatial relations between each surface and the object centroid to help recognize the object from partial surface descriptions.

Matching was accomplished by finding possible matched models based upon surface characteristics. These characteristics are determined by the normalized surface equations which are stored in the model data base and also calculated from touch and vision surface points. After the integration procedure the set of possible matched models is reduced by matching additional surfaces, if possible, and by imposing constraints upon the relative location of the object centroid to that surface. These constraints were referred to as the "discriminant" and were defined as a geometric primitive that related each surface to its object centroid.
After the object had been recognized its location in the robot world coordinate frame was determined. Since the locations of surface points were obtained from the touch system the problem was reduced to finding the location of the object centroid relative to any touch point. This relative location was determined during the recognition process using the discriminant. Simple coordinate transformations were then applied to calculate the absolute location of the object centroid in the real world coordinate frame.

Finally, a series of experiments was run to test the ability of these methods to recognize and locate objects. This chapter is an attempt to put this work into context by discussing what was successful, what were the drawbacks, and what needs further development. Ideas for extensions to this work and possible future approaches are also presented.

9.2 Touch Sensing

The first success of this research was the use of active exploratory tactile sensing to obtain surface characteristics. Touch has been considered a poor sensing method to use with robots when compared to vision. This is due in part to the fact that most previous research was based on static touch which is local in nature. Active touch provides more useful information but it requires synchronization of the sensor motion with the collected data. This synchronization has been directed by analysis of data provided by force-torque sensors. An important
conclusion is that supplementary information must be used to guide the touch data collection scheme to eliminate the time waste in collecting unneeded touch data points and their corresponding processing. Vision is a good source of such supplementary information if the same object descriptions can be extracted from both sensing media. This makes touch and vision integration an important approach in the task of object recognition and localization.

A drawback to using touch data is the data collection process. Touch data is collected serially (point by point) using a probe. This method of data collection is very slow when compared to vision data collection that gives many points simultaneously. On the other hand, touch data gives the required surface features (surface points and surface normals) without further extraction processing. Another drawback to using touch sensors is the low resolution of the tactile sensors and the noisy data of the force-torque sensors. Errors were introduced due to hysteresis in the tactile sensor array. These errors were corrected by filtering the tactile data and performing many experiments to understand the behavior of the tactile pad. Since this data was only used to detect that the probe is touching the surface it did not impose serious problems on the recognition process itself but led to a slow collection of touch data. Accordingly, the recognition process was designed to use very few vision points on each surface to reduce the data acquisition time while only one touch point on a surface was required to calculate the location of the object.
9.3 Multi-sensor Data Integration

The success of this research was the ability to construct common surface characteristics using a combination of vision and touch sensory information. This gave an added advantage to the use of point based methods in building surface descriptions. In this research single pixels were integrated with touch points that had the same three dimensional information. Then, points with the same statistical properties were grouped into a single surface description. The similarity estimation developed in this dissertation to integrate the vision and touch data works much better than previous methods used a surface growing technique from vision alone as described in [17, 19, 61]. However, this similarity measures can not be applied to points which are far from each other because of the possibility of local noise in the vision data. The coordinates of the point and the surface normal at that point were integrated with similar visual data to form vision-touch surface patches that are directly used in the matching process. No complete surface description was required so the segmentation process was not a necessary step in this research. Consequently the speed of the system was increased and less calculation and vision processing was required.

9.4 Model Based Object Recognition and Localization

The object recognition and localization algorithm recognizes objects through matching object characteristics to a model data base.
The model data base is based on surface characteristics organized to speed up the recognition process. The objects were arranged in the model in two different ways to ease the initial selection of possible matched models. Certain regions from the object may not be easily sensed either due to their complexity or the inability of the vision system to understand a region. In this research all vision-touch patches must contain vision and touch data. Initial matching of surface characteristics allows recognition to continue, searching for possible surfaces with the same characteristics. Three dimensional objects can be complex; however the experiments (test object 2) have shown that if each surface can be sensed, it can lead to full and fast recognition. A drawback of this method is it can only be applied to objects bounded by planar, cylindrical and spherical surfaces. Determining the location of the object is done following the recognition process and used some of its results. It would be more faster if the recognition and localization were done simultaneously.
9.5 Future research

There are a number of directions in which this research could lead. The first direction is to integrate touch data from hidden surfaces with vision data from viewed surfaces to extract complete three-dimensional object descriptions. Two issues must be addressed for this particular extension. First, how to design an intelligent touch strategy to acquire data from unknown surface(s) such that every surface can be sensed. Second, how to match surface descriptions acquired from the same surface such that all object descriptions can be integrated into one. In order to achieve this goal a modification to the calculation of the surface equation should be done to include a procedure for calculating the equation from sparse touch points only. A related problem is how to integrate this equation with the surface equations of only viewed surfaces into a single object description.

The second direction is recognition of occluding edges. This problem can be solved by visually guiding the touch sensor to trace the object after recognition to verify the recognized edge. An extension in the same direction is the recognition of multiple objects. In this case surfaces that may belong to different objects can be considered as an occluding edge problem. The solution of this problem is to develop a procedure to guide the touch system based on vision and touch data simultaneously.
The third direction is to widen the domain of objects used to include other quadric surfaces such as ellipsoids, paraboloids, and hyperboloids. The axis of symmetry of these surfaces require a special representation that needs more constraints for recognition. The way to use the touch sensor in a visually guided exploratory manner to recognize objects that include holes and cavities is still an open question.

Another avenue of research is incorporating information from sensor systems other than vision or active tactile sensors used in this work. The benefit of many sensors is shown in this work, and there is no reason to stop at two. The control problem becomes much larger, distributed processing techniques will be needed, and reasoning from many (perhaps conflicting) sources will also be needed.

The touch sensor used here is a single active probe, and robots will need multiple fingers to do grasping and manipulation. Extending this work to multiple sensing elements (multi-fingers) is possible. This will help collect more data simultaneously and rapidly.

The application of this research can be extended to current design and manufacturing problems such as parts assembly and inspection.
References


